# Structure of the M31 Satellite System: 

## Bayesian Distances from the Tip of the Red

## Giant Branch

By<br>Anthony Rhys Conn

A thesis completed under a cotutelle agreement between
Macquarie University and the Université de Strasbourg
FOR THE DEGREE OF
Doctor of Philosophy
May 2013


UNIVERSITÉ DE STRASBOURG

(C) Anthony Rhys Conn, 2013.

Typeset in ${ }^{A T} T_{E} X{ }_{\varepsilon}$.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Anthony Rhys Conn

## Acknowledgements

I have been very fortunate throughout my PhD candidature to work with some very dedicated and talented astronomers and astrophysicists who have had a huge impact on the course my research has taken. I have worked most closely with my supervisors Dr Rodrigo Ibata (Université de Strasbourg) and Prof. Geraint Lewis (University of Sydney) who I would like to thank for the large amount of their time and resources they have invested in my project. Rodrigo is a renowned figure in the field of Galactic Archaeology and I have greatly benefited from his expertise. He has also been a great host on my many visits to Strasbourg and always available to discuss the problem at hand. Likewise Geraint has had a prolific impact on the field and is a veritable source of programming knowhow. He devotes an enormous amount of energy to the various projects of his many students and his input has been invaluable to my research. In addition, I would also like to thank my supervisors Prof. Quentin Parker and A/Prof. Dan Zucker (both from Macquarie University) for their constant support and encouragement. In particular I am indebted to Quentin for the great lengths he went to in securing a funded position for me at Macquarie and for his efforts in establishing and maintaining the cotutelle agreement with the Université de Strasbourg. Whilst my PhD project was somewhat external to the fields of expertise of both Quentin and Dan, they have always made themselves available to discuss my work at length and advise on all matters of administration.

In addition to the contributions of my 4 supervisors, there are a number of other people who have played an important role in my progress toward completion. I would especially
like to thank all the members of the Pan Andromeda Archaeological Survey (PAndAS) collaboration for providing a wonderful repository of data to work with and particularly those members included as coauthors on my papers for their detailed reviews of my drafts. I would also like to add that I had many meetings with Dr Nicolas Martin (Université de Strasbourg) whilst in Strasbourg which had a large impact on the work presented in Paper II. In addition I must thank Neil Ibata for his French translation of my thesis summary (Résumé de Thèse). I also received much support from my fellow PhD student Dr Anjali Varghese on my many trips to Strasbourg and relied on her superior French more than once.

There are also a number of institutions that I must acknowledge for the support they have provided. Firstly, I would like to thank Macquarie University for their financial support via the Macquarie University Research Excellence Scholarship (MQRES) and also for paying all my airfares and providing assistance with other expenses incurred during my candidature. I would also like to thank the Observatoire Astronomique of the Université de Strasbourg for providing me with accommodation on each visit to Strasbourg and I would like to thank both the aforementioned universities as well as the University of Sydney for use of computational and all other facilities. In addition, I thank the American Astronomical Society (AAS) for granting permission to reproduce the published versions of Conn et al. (2011), Conn et al. (2012) and Conn et al. (2013) in Chapters 3, 4 and 5 respectively.

Finally I would like to thank all of my family for their ceaseless support and encouragement. Without my family it is inconceivable that I could ever have made it this far. I would therefore like to dedicate this work to all of them: Mum, Dad, Nanna, Pa, Mardi, Pop, Andrew, Crysta, Sophie, Kristin, Remo, Alex, Chris, Matthew, Trixie and all my extended family.

## A Note on Paper Co-Author Contributions

As a thesis by publication, much of the work contained in the main body of this thesis has been published in the form of three separate papers submitted to the Astrophysical Journal. Each of these publications acknowledges contributions from a relatively large group of coauthors. This is an inevitable consequence of the use of data that is the property of a large collaboration (i.e. the Pan-Andromeda Archaeological Survey - PAndAS), and it has in no way diminished the amount of work that I have put into each of these papers. In fact, rather the opposite is true, due to the enormous amount of code adjustments and paper editing that has had to take place to accommodate the suggestions of a large group of people. The work contained in these papers has been carried out principally with the direction of the second and third co-authors, my supervisors Prof. Geraint Lewis and Dr Rodrigo Ibata, with all other contributions being in the form of written critiques of the paper drafts. This does not detract from the aid I have received from my other two supervisors Prof. Quentin Parker and $\mathrm{A} /$ Prof. Dan Zucker, who as well as providing similar critiques of my drafts, have been a great help with all administrational matters that I have contended with during the course of my PhD candidature. I should also note that I had many in-person discussions with Dr Nicolas Martin whilst in Strasbourg, which further influenced the way I carried out the analysis contained in the second paper.

All of this said, all three papers have been written entirely by me, and all figures they contain have been generated by me using the PGPLOT plotting package (Pearson, 2011), with the sole exception of the pole-density plots in Paper III; Ch. 5 (Figs. 2, 4, 8, 10 (right-hand column of plots), 11 and 14), which were produced using a Gaussian-smoothing
program by Geraint Lewis from the pole count analysis data that I sent him. All of the analysis contained in the papers has been undertaken using Fortran code that I have written from scratch, with examples of most of the principal versions of the code provided in the appendix. These programs do make occasional use of subroutines written by others, as now stated:

I The "PolyS elect" subroutine is built around code written by Rodrigo Ibata, which uses the functions "in_poly" and "fimag" written by him. Given a series of points defining the corners of a polygon, it determines whether or not a given point is inside this polygon. I use this routine to make colour-cuts on the Colour-Magnitude Diagram (see calls of this subroutine in MF_TRGB.f95; Appendix D for example) and also to reject random satellites drawn outside of the PAndAS survey area in my "RandomPoints" subroutine, used in the analysis of Paper III.

II Part of the SVDFitter subroutine called in MF_TRGB.f95 (and earlier versions of this program) has been modified by Geraint Lewis.

III The "func_i" function called in MF_TRGB. $f 95$ (and earlier versions of this program) gives the photometric error as a function of magnitude in CFHT i-band for the PAndAS data and was provided by Rodrigo Ibata.

IV The "k_verse_alpha" subroutine called in SatDensity_SampCont.f95 (Appendix D) makes use of data generated by Geraint Lewis (k_vs_alpha.dat), which accounts for the volume of space covered by the PAndAS survey area as a function of distance from M31.

V Several subroutines from Numerical Recipes (Press et al., 1992) have been used to perform standard functions by various programs I have written, in particular MF_TRGB.f95. In the programs listed in the appendices, use of these algorithms are generally noted at the end of the program with commented double lines and the words 'Libpress Algorithms' embedded. The Numerical Recipes subroutines themselves are not shown for the sake of brevity, but any subroutine or function that is called from the code as printed
in the appendices and yet is apparently absent from this printed code can be assumed to be from Numerical Recipes.

VI The SLALIB Positional Astronomy Library (Wallace, 1994) has been used to convert object RA and Dec into tangent plane coordinates in MF_TRGB. $f 95$ and earlier TRGB programs so that the user can access various parts of the PAndAS Survey directly by entering the objects Celestial coordinates. This library is also made use of in many of the programs written for Paper III to convert back and forth between tangent plane angles and real angles, and also to measure the angle between two given objects on the sky.

I believe this to be a complete disclosure of all coding aspects that I make use of that I have not written personally. They are few in number compared to the number I have written and in general, their function has been to perform tasks secondary to the principal operation of the programs that use them. Their inclusion is however, nevertheless vital to the correct functioning of the programs. In summary, whilst many have contributed to the work presented in these three papers, my contribution to each of them is exactly the same as any dedicated student would make to the work contained in a major chapter of their thesis were it not a journal publication.

## Other General Comments

This thesis has been typeset in ETEX using a template developed by Paul Cochrane, Alexei Gilchrist and Johann-Heinrich Schönfeldt. The template has been modified slightly to better suit the particular form of this thesis. Note that the digital version contains clickable hypertext providing links to relevant figures, sections and references. As a thesis by publication, the included papers have been generated independently and hence adhere to a different format. Whilst they are integrated in terms of the page numbering and access from the table of contents, their internal hyperlinks are inactive in the digital version of the thesis. Note also that whilst the references contained in the papers appear again in the the thesis 'References' section, the pages on which they are cited will only be available for citations external to the papers.

This thesis has been submitted under a cotutelle agreement between Macquarie University and the Université de Strasbourg, whereby the thesis is examined twice, under the independent procedures of both universities. For the Macquarie University examination, the three examiners (all external) remained anonymous. For the examination at the Université de Strasbourg, the thesis was defended before a jury of examiners and other faculty members as per the prevailing French custom. This took place at 3 pm on 7 Feb 2013 at the Observatoire Astronomique de Strasbourg. The examiner members of the jury were: Prof. Christian M. Boily (Université de Strasbourg); Dr Michele Bellazzini (INAF - Osservatorio Astronomico di Bologna); Dr Annie C. Robin (Observatoire de Besançon) and Dr Andreas Koch (Universität Heidelberg - External Examiner).

## List of Publications

- Conn, A. R.; Lewis, G. F.; Ibata, R. A.; Parker, Q. A.; Zucker, D. B.; McConnachie, A. W.; Martin, N. F.; Irwin, M. J.; Tanvir, N.; Fardal, M. A.; Ferguson, A. M. N.

A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. I.
The Astrophysical Journal, Volume 740, Issue 2, article id. 69 (2011)

- Conn, A. R.; Ibata, R. A.; Lewis, G. F.; Parker, Q. A.; Zucker, D. B.; Martin, N. F.; McConnachie, A. W.; Irwin, M. J.; Tanvir, N.; Fardal, M. A.; Ferguson, A. M. N.; Chapman, S. C.; Valls-Gabaud, D.

A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. II.
Distances to the Satellites of M31
The Astrophysical Journal, Volume 758, Issue 1, article id. 11 (2012)

- Conn, A. R.; Lewis, G. F.; Ibata, R. A.; Parker, Q. A.; Zucker, D. B.; McConnachie, A. W.; Martin, N. F.; Valls-Gabaud, D.; Tanvir, N.; Irwin, M. J.; Ferguson, A. M. N.; Chapman, S. C.

The Three-Dimensional Structure of the M31 Satellite System;
Strong Evidence for an Inhomogeneous Distribution of Satellites
The Astrophysical Journal, Volume 766, Issue 2, article id. 120 (2013)

- Ibata, R. A.; Lewis, G. F.; Conn, A. R.; Irwin, M. J.; McConnachie, A. W.; Chapman, S. C.; Collins, M. L.; Fardal, M.; Ferguson, A. M. N.; Ibata, N. G.; Mackey, A. D.; Martin, N. F.; Navarro, J.; Rich, R. M.; Valls-Gabaud, D.; Widrow, L. M. A vast, thin plane of corotating dwarf galaxies orbiting the Andromeda galaxy

Nature, Volume 493, Issue 7430, pp. 62-65 (2013)

## Abstract

The satellite system of a large galaxy represents the ideal laboratory for the study of galactic evolution. Whether that evolution has been dominated by past mergers or in situ formation, clues abound within the structure of the satellite system. This study utilizes recent photometric data obtained for the halo of M31 via the Pan-Andromeda Archaeological Survey (PAndAS), to undertake an analysis of the spatial distribution of the M31 satellite system. To do this, a new Bayesian algorithm is developed for measuring the distances to the satellites from the tip of their Red Giant Branch. The distances are obtained in the form of posterior probability distributions, which give the probability of the satellite lying at any given distance after accounting for the various spatial and photometric characteristics of the component stars. Thus robust distances are obtained for M31 and 27 of its satellite galaxies which are then transformed into three-dimensional, M31-centric positions yielding a homogenous sample of unprecedented size in any galaxy halo. A rigorous analysis of the resulting distribution is then undertaken, with the homogeneity of the sample fully exploited in characterizing the effects of data incompleteness. This analysis reveals a satellite distribution which as a whole, is roughly isothermal and no more planar than one would expect from a random distribution of equal size. A subset of 15 satellites is however found to be remarkably planar, with a root-mean-square thickness of just $12.34_{-0.43}^{+0.75} \mathrm{kpc}$. Of these satellites, 13 have subsequently been identified as co-rotating. This highly significant plane is all the more striking for its orientation. From the Earth we view it perfectly edge on and it is almost perpendicular to the Milky Way's disk. Furthermore, it is roughly orthogonal to the disk-like structure commonly reported for the Milky Way's satellite galaxies. The distribution is also found to be highly asymmetric, with the majority of satellites lying on the near side of M31. These findings point to a complex evolutionary history with possible links to that of our own galaxy.

## Résumé de Thèse

Étude de la structure tridimensionnelle du système de satellites de M31 au moyen d'une méthode Bayesienne de localisation de la pointe de la branche des Géantes Rouges

Les étoiles de basse masse pauvres en métaux qui ont consommé tout l'hydrogène présent dans leur noyau et dont celui-ci n'a plus une densité suffisante pour fusionner de l'hélium, entrent dans la phase de la branche des géantes rouges (RGB). Après un certain temps, l'étoile devient plus lumineuse et les cendres dhélium produites par cette réaction retombent sur le noyau, accroissant sa densité jusqu'à celle-ci soit suffisante pour remettre en marche la fusion de l'hélium. L'étoile, qui n'appartiendra bientôt plus à la branche d'étoiles RGB est dite du tip of the Red Giant Branch (TRGB). Du fait des propriétés similaires du noyau de toutes les étoiles qui arrivent à ce state de leur évolution dans une gamme spécifique de masse et de métallicité (voir Iben and Renzini 1983), leur radiation énergétique et donc leur luminosité est constante. Le TRGB pour de telles populations stellaires donne donc une mesure de la distance à cette population.

Avant le développement de la méthode de la détection d'un bord de Lee et al. (1993), la TRGB était déterminée par des Diagrammes Couleur-Magnitude (CMD) à l'oeil nu et les distances dérivées manquaient donc de précision et d'uniformité requis pour une utilisation fiable pour de nombreux objets. On a développé de nombreuses méthodes depuis celle-ci mais elles se basent toutes sur l'idée de convoluer la fonction de luminosité (LF) du RGB avec un kernel de détection de bord, afin de créer un maximum à la magnitude correspondant à la plus grande discontinuité dans la LF, qui devrait correspondre à la magnitude du TRGB. Malheureusement, de telles méthodes donnent de mauvais résultats dans la présence de bruits - notamment lorsque le RGB est noyé par des étoiles contaminantes. Pour cette raison, plusieurs alternatives d'ajustement de modèles qui utilisent toute la LF ont été proposées
(par exemple Méndez et al. 2002). Malgré cela, pour ces méthodes, les incertitudes de mesures sont souvent très grandes et mal définies et n'ont pas la possibilité d'incorporer nos informations à priori sur le système étudié. C'est pour cela que la première grande partie de cette thèse aura pour but de créer un algorithme robuste et versatile pour mesurer des distances en utilisant la magnitude du TRGB.

Les premiers chapitres décrivent le développement d'un algorithme Bayesien qui utilise une approche de maximum de vraisemblance. Les paramètres du modèle (magnitude du TRGB, pente de la LF, propriétés de contamination) sont ajustés par l'algorithme suivant une simulation Markov Chain Monte Carlo (MCMC). Cela donne accès aussi aux incertitudes sur ces paramètres. Malgré sa simplicité, cette méthode est robuste, et donne des sorties intuitives et visuelles des probabilités de paramètres et il reste facile d'ajouter de l'information à priori. La première version de cet algorithme a été publiée dans le Astrophysical Journal (Paper I), et est à la base du chapitre 3. Cette publication présente également des tests qui caractérisent la performance de cette méthode pour des LFs de différentes qualités, ainsi que son application à trois galaxies naines sphéroïdales, satellites de M31, et donne les meilleures incertitudes de toutes les méthodes basées sur le TRGB publiées jusqu'à ce jour.

Les données physiques analysées dans cette thèse viennent du Pan-Andromeda Archaeological Survey (PAndAS - McConnachie et al. 2009), un relevé ambitieux qui couvre plus de 300 degrés carrés autour de la galaxie d'Andromèdre, la galaxie géante la plus proche de la Voie Lactée. Ce relevé donne accès à la photométrie profonde en bande $\mathrm{g}^{\prime}$ (centré sur 487 nm ) et la bande i' (centré sur 770 nm ), et qui couvre plus de 25 satellites galactiques qui sont idéaux pour des mesures de distance par la méthode TRGB. L'algorithme présenté en chapitre 3 a été amélioré pour utiliser ces donnes spécifiques. La contamination du fond étant la plus grande source du détriment de la qualité des distances TRGB, j' ai mis au point une routine << matched filter >> (voir Rockosi et al. 2002) pour donner des poids à chaque étoile en fonction de sa position spatiale dans le profil de densité du satellite. L'effet de l'application de cet algorithme sur la LF est de réduire la contamination du fond et ainsi d'augmenter le contraste de la troncature du RGB au TRGB. Visuellement, le changement du LF est souvent suffisant pour révéler de façon très claire la position du TRGB qui était avant à peine plus que du bruit Poissonnien.

Cette méthode améliorée, appliquée à tous les satellites (27 en total) détectés dans le relevé PAndAS est présentée dans un deuxième article soumis à l'Astrophysical Journal et constitue l'essentiel du chapitre 4 (Paper II). Cet article apporte les premières mesures de distances pour une grande partie de ces satellites et se révèle être l'analyse la plus compréhensive des distances du système de satellites de M31. Cette investigation contient également une analyse brève du profil de la densité du halo en utilisant ces nouvelles distances, que nous avons comparées aux valeurs trouvées avec l'aide d'autres méthodes.

Le grand nombre de satellites autour de M31 pour lesquels j'ai obtenu de bonnes mesures de distances donne ainsi une excellente occasion d'analyser le degré de planarité et d'asymétrie du système de satellites. Cela a des fortes répercussions sur la distribution de matire dans le halo de la galaxie hôte ainsi que sur l'histoire de formation des satellites mêmes. Plusieurs études du système satellitaire de la Voie Lactée (par exemple Lynden-Bell 1982; Zentner et al. 2005; Pawlowski et al. 2012b), trouvent des plans fortement significatifs, souvent inclinés par rapport au disque Galactique. Des résultats similaires ont été publiés pour le système de M31 (par exemple Koch and Grebel 2006). Les études du système de M31 ont été faits avec de petits échantillons de satellites et les mesures de distances proviennent donc de plusieurs auteurs (et méthodes) différentes. C'est ainsi que le chapitre 5 et une troisième publication donnent à voir une analyse détaillée du système de satellites de M31 en se basant sur les données du chapitre 4. La planarité du système de satellites est explorée par le biais du plan de meilleur ajustement en utilisant plusieurs méthodes (moindre rms, moindre distance, ajustement à un modele Gaussien). La vraisemblance de ces alignements est analysée à l'aide de simulations où chaque satellite est tiré au hasard a partir de sa distribution de distance. L'analyse de l'asymétrie est effectuée de façon similaire, en utilisant des statistiques d'asymétrie, notamment le nombre de satellites qui se trouvent sur un hémisphère du halo. Les positions 3-D présentées au chapitre 4 montrent que le pôle du plan d'asymétrie maximal se trouve très près du vecteur Terre-M31 ; la probabilité d'un tel alignement est étudiée dans cette thèse.

## Contents

Acknowledgements ..... v
A Note on Paper Co-Author Contributions ..... vii
Other General Comments ..... xi
List of Publications ..... xiii
Abstract ..... xv
Résumé de Thèse ..... xvii
1 An Introduction to Galactic Archaeology ..... 1
1.1 Overview ..... 1
1.2 A Portrait of a Galaxy ..... 2
1.3 Galactic Archaeology - The Means and the Motives ..... 4
1.4 Completed and Future Surveys - What can they tell us? ..... 8
1.4.1 Photometric ..... 8
1.4.2 Astrometric ..... 10
1.4.3 Kinematic ..... 13
1.5 Dark Matter and the Predictions of $\Lambda$ CDM Cosmology ..... 15
1.6 Resolving the Matter - Methods for Measuring the Dark Matter Distribution ..... 19
1.7 The Pan-Andromeda Archaeological Survey ..... 23
1.8 The Importance of Position ..... 27
2 Building the Framework for a new TRGB Algorithm ..... 31
2.1 The RGB Tip Finding Problem ..... 31
2.2 Early Trials of TRGB Finding Algorithms ..... 33
2.3 A Simple Maximum Likelihood Test ..... 36
2.4 The Markov Chain Monte Carlo Method ..... 42
2.5 The Bayesian and the Frequentist ..... 45
2.6 Prior Information ..... 48
3 Paper I: A Bayesian Approach to Locating the Red Giant Branch Tip Magni- tude. I. ..... 51
Paper I Preface ..... 52
Abstract ..... 53

1. Introduction ..... 53
2. Method ..... 54
2.1 The MCMC Method ..... 54
2.2 A Note on Distance Errors ..... 58
2.3 Initial Tests ..... 59
2.4 Algorithm Behavior for Composite Luminosity Functions ..... 59
3. Distances to Two More Satellites ..... 60
3.1 Andromeda II ..... 61
3.2 Andromeda XXIII ..... 61
4. Conclusions ..... 62
Acknowledgements ..... 62
References ..... 63
4 Paper II: A Bayesian Approach to Locating the Red Giant Branch Tip Magni- tude. II. Distances to the Satellites of M31 ..... 65
Paper II Preface ..... 66
Abstract ..... 70
5. Introduction ..... 70
6. A Recap of the Base Method ..... 71
7. Addition of a Matched Filter ..... 72
3.1 Matched Filtering using Radial Density Profiles ..... 72
3.2 A Test for the Refined Algorithm ..... 73
3.3 An Additional Prior ..... 74
8. A New Perspective on the Companions of M31 ..... 77
4.1 Galaxy Distances ..... 77
4.2 Determining the Distances from M31 ..... 80
4.3 A First Approximation of the Satellite Density Profile within the Halo ..... 82
9. Conclusions ..... 88
Acknowledgements ..... 88
References ..... 88
5 Paper III: The Three Dimensional Structure of the M31 Satellite System; Strong Evidence for an Inhomogeneous Distribution of Satellites ..... 89
Paper III Preface ..... 90
Abstract ..... 97
10. Introduction ..... 97
11. Method ..... 99
2.1 Plane Fitting ..... 99
2.2 Generating Random Satellite Samples ..... 100
2.3 A note on Satellite Detection Bias ..... 100
12. Results ..... 101
3.1 Best Fit Plane to the Entire Satellite Sample ..... 101
3.2 The Plane of Maximum Asymmetry ..... 102
3.3 Subsets of Satellites ..... 102
3.4 A Great Plane of Satellites ..... 109
13. Discussion ..... 109
14. Conclusions ..... 111
Acknowledgements ..... 112
References ..... 112
6 Conclusions ..... 113
An Introduction to the Appendices ..... 117
A An Article in nature ..... 119
B Chapter Two Programs ..... 139
C Chapter Three Programs ..... 161
D Chapter Four Programs ..... 189
E Chapter Five Programs ..... 263
List of Abbreviations ..... 315
References ..... 317

## An Introduction to Galactic Archaeology

### 1.1 Overview

Large galaxies like the Milky Way and its neighbor the Andromeda Galaxy (M31) are complex, evolved structures when studied on any scale. They are a plethora of countless billions of stars and the condensing clouds of gas and dust from which they form, all in motion, all evolving since time immemorial. But far removed though their origins may be, their very structure preserves their past. However, even the structure of the Milky Way, our own galaxy, is not obvious from our vantage point deep within it and while the general structure of its basic components have been constrained, there is an underlying labyrinth of substructure remaining to be identified and interpreted with respect to its bearing on Galactic Evolution. Hence we must begin our study with an overview of the large scale structure of our own
galaxy a structure which, as might be expected, is shared by many of our galactic neighbours and indeed by nearly all those galaxies near and far of a similar type.

### 1.2 A Portrait of a Galaxy

The Milky Way (henceforth 'the Galaxy') is a late-type barred spiral galaxy. It is known to consist of both a thin and a thick disk component, a central bulge and an enormous halo, encompassing the whole system (Freeman and Bland-Hawthorn, 2002). The thin disk has been determined to have a scale length of 2600 pc and a scale height of 300 pc (Jurić et al., 2008) with an overall radius of $15 \pm 2 \mathrm{kpc}$ (Ruphy et al., 1996). It is within the thin disk that both the solar neighbourhood and the spiral arms reside. The spiral arms have been traced by various methods, notably by Georgelin and Georgelin (1976), who used HII regions to trace their extent. They found two symmetrical pairs of arms with a pitch angle of $12^{\circ}$. The four arms in total were identified as the Sagittarius-Carina Arm, the Scutum-Crux Arm, the Norma Arm and the Perseus Arm, with the Sun residing in a spur between the inner Sagittarius-Carina Arm and the outer Perseus Arm. This is represented schematically in Figure 1.1. Based on their findings they suggest a morphological type for the Galaxy closest to Sc .

Enveloping the thin disk is a somewhat more diffuse, ancient haze of stars termed the Galactic 'thick disk' (Gilmore and Reid, 1983). It has been calculated from the Sloan Digital Sky Survey I (SDSS I) to have a scale length of 3600 pc and a scale height of 900 pc (Jurić et al., 2008). Freeman and Bland-Hawthorn (2002) describe it as a 'snap frozen relic of the heated early disk' and allocate some $10 \%$ of the Galaxy's baryonic matter to its confines. A metallicity of $-2.2<[\mathrm{Fe} / \mathrm{H}]<-0.5$ is quoted for the thick disk stars in contrast to the -0.5 $<[\mathrm{Fe} / \mathrm{H}]<0.3$ determined for the younger thin disk, and its luminosity is specified as $10 \%$ that of the thin disk.

In the inner regions of the Galaxy is a denser conglomeration of what are generally considered to be older, metal poor stars termed 'the bulge.' Freeman and Bland-Hawthorn (2002) caution however that a study of bulge red giant stars (McWilliam and Rich, 1994) suggests


Figure 1.1: A Schematic of the observable portion of the Milky Way’s spiral arms. (Vallée, 2005)
a metallicity much closer to the older stars of the thin disk than to the truly ancient stars in the Galactic halo. They further describe the Milky Way's bulge as appearing significantly smaller than that of M31 and somewhat 'boxy,' typical of an Sb to Sc spiral. Also of particular note, the Galaxy has long been suspected of containing a bar at its centre which has, as of 2005 , been proven. Benjamin et al. (2005) find the bar to have a length of $8.8 \pm 1.0 \mathrm{kpc}$ with orientation such that it is rotated $44 \pm 10^{\circ}$ from a line connecting the Sun and Galactic Centre.

Finally, the halo of the Milky Way is easily its largest and arguably its oldest major constituent. It is an enormous, roughly spherical (Ibata et al., 2001b) cocoon of ancient field stars, and approximately 150 similarly ancient globular clusters (Freeman and BlandHawthorn, 2002). It is also known to extend out well beyond the Small Magellanic Cloud to a distance of 100 kpc from Galactic centre and it contains more than 25 known satellite galaxies (McConnachie, 2012). Perhaps most remarkable is that it contains $1.0_{-0.2}^{+0.3} \times 10^{12} M_{\odot}$ (Xue et al., 2008) of dark matter, which amounts to at least $90 \%$ of the total mass of the

Galaxy (Freeman and Bland-Hawthorn, 2002). The substructure within this dark matter halo is of great interest as it lies at the heart of our current understanding of galaxy formation.

Our current knowledge of the Galaxy as presented in the above paragraphs represents some of the fruits of Galactic Archaeology. This knowledge is however fairly coarse in scope and Galactic Archaeology may still be regarded as a burgeoning field. Nevertheless, it is our means to unravel the Galaxy's past and our best hope for predicting its future.

### 1.3 Galactic Archaeology - The Means and the Motives

The field of Galactic Archaeology is in a sense a toolkit providing the necessary tools to wind back the cosmic clock and provide us with a high resolution view of our galaxy and its immediate neighbours in a way that might otherwise have been restricted to the poorly resolved galaxies of the high-redshift universe. It is not a single method but rather a collection of techniques making use of large sky photometric, astrometric and kinematic surveys to study the positions, motions and chemical compositions of groups of stars in an effort to link them to ancient progenitor structures and then simulate the evolution of these structures through time to the present and beyond. In other words, if stars are found to be grouped together in 6D phase space (i.e. 3 dimensions in position and 3 dimensions of velocity) they may be members of a present day cluster whereas stars grouped together only in velocity space may be termed a moving group and be members of a since-dispersed cluster. Stars grouped together in chemical space might similarly be 'tagged' to an ancient progenitor structure. Some of these possibilities are further investigated in the following paragraphs.

With the advent of Galactic Archaeology, the discovery of moving groups has become common. In an early example, Eggen and Sandage (1959) identified the nearby moving group Groombridge 1830 and associated it with the Galaxy's globular clusters, providing an early detection of nearby halo stars. In the intervening decades, numerous further examples have been discovered associated with the halo alone, but Freeman and Bland-Hawthorn (2002) caution that the validity of some of these groups is questionable.

The tagging of stars to progenitor groups based on their chemical composition is perhaps an even more powerful technique. It relies on the assumption that the progenitor cloud
be uniformly well mixed before the formation of the surviving stars (Freeman and BlandHawthorn, 2002) which is conceivable if McKee and Tan (2002)'s model of cluster formation is accepted whereby all stars form at a similar time. Such a method has interesting implications not only for the origins of structure formation in the Galaxy at large, but also at a more local level, as it presents the real possibility of identifying Solar siblings - those stars that formed out of the same cloud as our Sun. Indeed Reipurth (2005) lists possible evidence supporting the idea that the Sun did in fact form in a cluster and Portegies Zwart (2009) goes so far as to provide mass and radius constraints for the cluster of $500-3000 \mathrm{M}_{\odot}$ and $1-$ 3 pc respectively. They further concur that with accurate chemical abundances and phase space information, the identity of the cluster members may be recovered. A direct test of the feasibility of chemical abundance tagging is seen in De Silva et al. (2007) where of the 18 supposed members of the comoving group HR1614, 14 were found to have very little scatter in chemical abundances across a wide range of elements with the non-conforming stars conceivably 'pollution' from the non-cluster background. Thus it seems that, at least in some cases, this powerful technique proposed for Galactic Archaeology should be applicable.

So far we have encountered the means to re-construct ancient Galactic components but the question remains - how ancient? A time frame is needed to accurately model the Galaxy's evolution, as evolution is after all time dependent. There are various methods proposed to fulfill this function, all relating to the determination of stellar age, of which Freeman and Bland-Hawthorn (2002) gives a concise summary. Since we are generally concerned with stars long since removed from their parent clusters, determining age from the main sequence turnoff is obviously not an option. Instead, such methods as nucleo-cosmochronology, astero-seismology and age-metallicity relations are suggested. Nucleo-cosmochronology is concerned with ageing the elements in a star based on their remaining radioactive isotope strengths, given a certain radioactive decay rate (see Ludwig et al. 2010). Since the original elemental abundances are not known, the method compares the radioactive isotope strengths to stable r-process elements. Some studies based on this technique have already been highly successful. Astero-seismology takes advantage of the evolving mean molecular weight in the cores of stars to ascertain age and has been used to provide an age for the Sun of 4.57 Gyr $\pm 0.12 \mathrm{Gyr}$ (Gough, 2001), which matches well with the ages determined for the oldest


Figure 1.2: An example of the Age-Luminosity Relation. Here, the ages of the oldest globular clusters have been plotted as a function of the absolute visual magnitude of component RR Lyrae stars. The best fit median is represented by the solid line while the dashed lines represent $1 \sigma$ limits. (Chaboyer et al., 1998)
meteorites by more direct means. An age-metallicity relationship would provide a more direct measure of stellar age, if indeed one could be established but alas, such a relationship only applies to a small subset of stars. Freeman and Bland-Hawthorn (2002) find such a relationship to exist only for a small range of young, hot, metal-rich stars. More useful however is the age-luminosity relationship (Figure 1.2) found to apply to the much older RR Lyrae stars, provided their distances may be accurately determined. This principle has been applied to constrain the ages of the Galaxy's globular clusters (Chaboyer et al., 1996). Still, there is a large age interval over which the latter two methods are not applicable, thus emphasizing the importance of the former two methods.

Having discussed the tools of Galactic Archaeology, what are its goals and to what extent have these goals already been met? The ultimate goal of Galactic Archaeology is to be able to trace the current structures of the Galaxy back to their progenitor structures in the protocloud
from which it formed. In so doing, the histories of the various components of the Galaxy are uncovered, spanning from the epoch of formation to the present day. As outlined in section 1.2, the basic structure of the Galaxy has already been established and based on the stellar ages and metallicities/ elemental abundances across the various components an hypothesis for galaxy formation has been formulated, again summarized in Freeman and Bland-Hawthorn (2002). It is suggested that the Galactic Protocloud began to form at a similar time to the epoch of reionization. At this time the Galaxy, like those around it, appeared in the form of a dark matter halo, with its central black hole and possibly its stellar bulge forming first. The prominent disk structure where most of the baryons reside did not develop until the beginning of the main epoch of baryon dissipation at a redshift of $z \sim 1-5$. This also coincides with the ages of the thick disk and the globular clusters. The populating of the halo with globular clusters and field stars is thought to have also begun very early in the formation process, the result of tidal interactions with small neighbouring dwarf galaxies. The thin disk comprises the youngest stars of the Galaxy while the thick disk is likely the dynamically heated remnant of an ancient thin disk - in fact, Galactic Archaeology may provide some clue as to the particular interaction responsible. One popular theory is that the globular cluster $\omega$ Cen is the remnant core of a small galaxy, stripped of its outer stars in an interaction precipitating the heating of the original thin disk (Bekki and Freeman, 2003). It is also believed that the current Galactic bulge is not of the ancient origin of more pronounced bulges such as that found in M31, but rather a later formation in the established inner disk. This is consistent with the relatively high metallicities in the Galactic core, although it must be stressed that metallicity is a better measure of the number of supernova events rather than of actual age and the density of the Galactic core is bound to influence this number profoundly.

The formation sequence presented above owes little to observations of high-redshift galaxies or even to computer simulations based on Cold Dark Matter (CDM) Cosmology, but rather it is a construction based on observations of our own galaxy and those nearby. Our focus has so far been centered on the Milky Way, but it must be stressed that any galaxies close enough to have their individual stars mapped into phase space or chemical space are within the reach of Galactic Archaeology. It should also be stressed that the methods associated
with Galactic Archaeology described above form the basis for such study but such methods provide for mere data acquisition - the possible applications for the data are enormous, and hence so is the scope of Galactic Archaeology. These points should be kept in mind as some of the various sky surveys available to the 'Galactic Archaeologist' are discussed in the next section.

### 1.4 Completed and Future Surveys - What can they tell us?

Modern Galactic Archaeology draws heavily on a small number of ambitious, wide field surveys focused, at least in part, on the acquisition of either photometric, astrometric or kinematic data for large numbers of stars. While there are many smaller data sets such as Hubble Space Telescope (HST) pointings and those from major ground telescopes which are also utilized, our focus here shall be limited to these major surveys.

### 1.4.1 Photometric

Among those surveys with the broadest scope are the photometric surveys, although the data they include is often more restrictive for Galactic Archaeology than that from the astrometric and kinematic surveys. Photometry is of particular usefulness in determining the distance to large numbers of objects. The two most recent major photometric catalogues are those from the Sloan Digital Sky Survey (SDSS) and the Two Micron All Sky Survey (2MASS). SDSS is an ongoing survey, begun in 2000, using the dedicated 2.5 m wide-field, modified RitcheyChrétien telescope at Apache Point Observatory and an array of $30 \times 4$ megapixel CCDs. The survey provides photometry in the $\mathrm{u}, \mathrm{g}, \mathrm{r}, \mathrm{i}$ and z bands (see York et al. 2000 for a technical summary) as well as spectroscopy of select targets. As of the ninth data release (SDSS-III Collaboration et al., 2012), the survey had covered some 14555 square degrees of sky or more than $\frac{1}{3}$ of the entire celestial sphere, with spectra obtained for 668054 stars. The survey also features stellar positions accurate to within 150 mas for each coordinate and metallicity as well as phase space information are determinable for the observed stars. The stellar coverage is however, relatively small owing to the survey's greater emphasis on obtaining photometry for galaxies.

The 2MASS survey (Skrutskie et al., 2006) in contrast covers an enormous quantity of stars, with some 471 million point sources extracted from the data. The survey covers the entire sky and includes photometry in the J, H and Ks near-infrared bandpasses. For entire sky coverage, two ground based telescopes were required, one in each hemisphere, and hence two 1.3 m telescopes were constructed for the task, one at Mount Hopkins, Arizona and the other at Cerro Tololo in Chile. The 7.8 second exposure for each field results in limiting magnitudes of $15.8,15.1$ and 14.3 in the $\mathrm{J}, \mathrm{H}$ and $\mathrm{K}_{s}$ bands respectively. A $1 \sigma$ error of $<0.03$ magnitudes is determined for the photometry with an estimated error of 100 mas in the source positions. With stellar positions as well as metallicity being determinable from the data, stellar tagging is a possibility from this data set. Indeed, this survey is a useful archive of data for isolating ancient structures, especially since such structures may be expected to be delineated by luminous red giant stars which would remain visible out to great distances due to their strong emission in the near infra-red. This merit of the survey has in fact been exploited by previous studies, as exemplified by Ibata et al. (2002a) where M giants were used to trace substructure in the outer Galactic halo. Still, the lack of kinematic data obtainable from the survey does present some limitations for reconstructing ancient structures that have since dispersed.

The Skymapper Telescope (see Keller et al. 2007) is currently working to improve on the 2MASS data set, at least for the southern celestial hemisphere. Skymapper is a 1.33 m telescope operated by the Australian National University (ANU) at Siding Spring mountain. It features an array of $32 \times 8$ megapixel CCDs mounted at the Cassegrain focus of the telescope to provide a 5.7 square degree field of view. Six coloured glass filters allow photometry in the $\mathrm{u}, \mathrm{v}, \mathrm{g}, \mathrm{r}, \mathrm{i}$ and z bands with peak throughput in the r band at around 650 nm . A proposed 'Five-Second Survey' consisting of at least 3 images of every field per filter is capable of providing photometry for stars of magnitude 8.5 through to 15.5 with a minimum accuracy in the g and r bands of $\sigma=0.1 \mathrm{mag}$, thus providing comparable sensitivity and accuracy to the 2MASS survey but with a wider wavelength coverage. With 36 observation epochs over a five year period, astrometry will also be possible from the Skymapper data, with proper motions as small as 4 mas year ${ }^{-1}$ detectable and position information accurate to within 50 mas. Hence in using the Skymapper data, Galactic Archaeologists have at their disposal 5
dimensions of phase space data as well as basic metallicity information for each surveyed star. It might therefore be argued that Skymapper represents one of the greatest leaps forward in the field of Galactic Archaeology to date and indeed the probing of the evolution and structure of the Galaxy ranks highly as one of the projects chief science goals.

### 1.4.2 Astrometric

Astrometry is essentially concerned with the determination of the 5 dimensions of phase space excluding radial velocity. Two data sets stand out as major contributions to the bulk of astrometry information currently available - that from the HIPPARCOS mission (ESA, 1997) and the data contained in the United States Naval Observatory (USNO) catalogues.

HIPPARCOS is actually an acronym for HIgh Precision PARallax COllecting Satellite, chosen in honour of the Greek astronomer Hipparchus whose main contribution to astronomy was astrometry, albeit in only two dimensions of phase space! The satellite operated from 1989 to 1993 providing high precision positional and proper motion data for more than 100000 stars. The final HIPPARCOS Catalogue consists of 118218 stars within a limiting magnitude of 12.4. The stars' positions on the celestial sphere, parallaxes and proper motions were determined to within median precisions of 0.77 mas, 0.97 mas and $0.88 \mathrm{mas} \mathrm{yr}^{-1}$ respectively. Additionally, photometry was determined for each star using an HIPPARCOSspecific visible pass band. The measurements were based on $\sim 110$ independent observations and are accurate to a mean value of 0.0015 mag. Based on these parameters, it is clear that the HIPPARCOS Catalogue represents an extraordinarily high precision source for phase space information and some photometry applications. The fundamental drawback to the data for Galactic Archaeology however is the small number of surveyed stars. This is remedied to some extent by the addition of the Tycho Catalogue (named in honour of Tycho Brahe's significant contributions to astrometry) wherein phase space data and photometry are presented for 1058332 stars with a median astrometric precision of 25 mas for all stars and photometry accurate to within 0.07 mag for B band photometry and 0.06 mag for V band photometry for all stars. It should also be noted that a new catalogue, Tycho 2 (Høg et al., 2000) has been released, based on the same raw data as the original Tycho Catalogue but with astrometry
available for 2.5 million stars and slightly higher parameter accuracy owing to a different reduction technique, yielding proper motions as small as $2.5{\mathrm{mas} \mathrm{yr}^{-1} \text { detectable. In sum- }}_{\text {d }}$. mary, the quality of the proper motion data from these three surveys distinguish them from other surveys, yet still, astrometric parallax - the particular specialty of these surveys - is inevitably limited by distance, so this dimension of phase space is not going to be available for far-flung structures of the Galaxy or extragalactic targets.

One of the largest astrometric catalogues available to date is the United States Naval Observatory A2.0 (USNO-A2.0) Catalogue (Monet, 1998). The catalogue is based on the same raw data as the USNO-A1.0 Catalogue (Monet et al., 1998) which was compiled using measurements of the Palomar Observatory Sky Survey I (POSS I) O and E plates for declinations north of $-35^{\circ}$ and the UK Science Research Council (SRC-J) and European Southern Observatory (ESO-R) survey plates for declinations south of $-35^{\circ}$. The plates were scanned using the Precision Measuring Machine (PMM) at the U. S. Naval Observatory Flagstaff Station with precisions of 150 mas in positional information and 0.15 mag in the b and r band photometry afforded by using the ACT Catalogue over the Guide Star Catalogue (GSC) - as was used for USNO-A1.0 - for astrometric calibration. The ACT catalogue is based on the combination of the Astrographic Catalogue and the Tycho Catalogue and provides proper motion information about an order of magnitude more accurate than that contained in the original Tycho Catalogue (Urban et al., 1998). The final product is a catalogue of some 526280881 stars with RA, DEC and $b$ and $r$ band photometry to the accuracies already specified. The data is hence limited to the 3 positional coordinates of phase space (assuming distances are obtained from the photometry) and minimal photometric information but nevertheless, the sheer bulk of stars covered warrants the inclusion of the USNO-A2.0 Catalogue as a major source of raw data for Galactic Archaeology.

In addition to these surveys, there have been some noteworthy astrometry surveys in the intervening years, such as that utilized for the Second US Naval Observatory CCD Astrograph Catalogue or UCAC2 (Zacharias et al., 2004) wherein are presented position and proper motion data for 48330571 sources - mostly stars - with declination between - $90^{\circ}$ and $+40^{\circ}$. The precision in position is estimated between 15 and 70 mas, depending on source magnitude, and proper motions are determined to within $1-3 \mathrm{mas}_{\mathrm{yr}}{ }^{-1}$ for stars
brighter than 12th magnitude and 4-7 mas yr ${ }^{-1}$ for those between 12th and 16th magnitude. Another, more restrictive survey, is the Southern Proper Motion Program III (Girard et al., 2004) which has catalogued $\sim 10.7$ million objects in an area $3700^{\circ}$ or $1 / 11$ of the entire sky, with proper motions determined in some cases accurate to $4 \mathrm{mas} \mathrm{yr}^{-1}$.

The future for the collection of astrometric data is potentially an exciting one, but alas there are many setbacks faced by would-be missions. Already, two particularly promising, ~ 40 million star surveys - the Full-sky Astrometric Mapping Explorer (FAME) and the German Interferometer for Multichannel Photometry and Astrometry (DIVA) - have been cancelled due to escalating costs and logistic difficulties. Disappointingly, this leaves some time until a new major astrometric survey is released. Nevertheless, two even more ambitious missions are scheduled for the next decade, one - JASMINE (the Japanese Astrometry Satellite Mission for INfrared Exploration) - is purely astrometric with regard to the dimensions of phase space it is intended to explore, the other, Gaia, will provide a measure of radial velocity as well and so is discussed amongst the 'kinematic' surveys in the next subsection. The JASMINE mission (see Gouda et al. 2005 and Gouda 2011), due for launch in the first half of the 2020s, is a 1.5 m space-based telescope under preparation by JAXA (the Japanese Aerospace Exploration Agency), designed to peer through the gas and dust of the Galactic disk at a wavelength of 0.9 microns. The telescope will be sent into a Lissajous orbit around the Sun-Earth Lagrange point L2 from where it shall undertake astrometry of some 100 million Galactic disk and bulge stars (or such stars brighter than magnitude 14 in the z band) in the Galactic Latitude range $|b| \leq 4.0$. As such it is not an all sky survey and it is of limited use for studying any other Galactic structures but nevertheless, with an accuracy of $10 \mu a s$ for position and parallax data and $10 \mu{\mathrm{as} \mathrm{yr}^{-1} \text { for proper motions, the mission }}^{\text {a }}$ has the potential to produce a substantial catalogue of data, so far unequaled in depth, for the appropriate Galactic Archaeology work. Two preliminary missions, 'Nano-JASMINE' and 'Small-JASMINE' are also scheduled, with Nano-JASMINE set to launch by the end of 2013 or early 2014 (Kobayashi et al., 2012).

### 1.4.3 Kinematic

Kinematic surveys are perhaps the most useful survey type to the Galactic Archaeologist as they provide a complete description of each star's location in phase space and provide the best chance for the identification of those structures sharing a similar evolutionary history. When this information is coupled with elemental abundance data, which is sometimes available from the same survey, the Galactic Archaeologist is endowed with the astronomical equivalent of the Rosetta Stone - the key to piece together the ancient lives of the Galactic populace. The only concern then is that the 'Galactic census' is far enough reaching to register enough substructure to give a representative view of the Galaxy in its entirety. Kinematic surveys are a relatively recent addition to the available data but, as we shall see, plans are afoot to see the kinematic dataset explode by the end of the next decade. Bland-Hawthorn and Freeman (2006) identify the Geneva-Copenhagen Survey of the Solar Neighbourhood (Nordstrom et al., 2004) as the first major kinematic survey - a study featuring kinematic data for 16682 nearby K and G dwarfs, with full 6D phase space data available for 14139 stars after combination with HIPPARCOS parallax data and Tycho 2 proper motions. Combined with photometry and metallicity data, the survey represents the means to study the precise structure of the local stellar neighbourhood and perhaps even identify any solar siblings that have migrated along similar paths to the Sun. Still, if enough data is to be had for the Galaxy on the broadest scales, the surveyed stars are going to have to be much more numerous and include those much less luminous!

Several such projects have either been completed or are in their final or preparatory stages. The most important completed to date is SEGUE - the Sloan Extension for Galactic Understanding and Exploration (Yanny et al., 2009). It is a moderate-resolution ( $R=1800$ ) spectroscopic survey of 240000 stars, spanning the spectral range from 390 nm to 900 nm , with the principal aim of aiding the study of the kinematics and populations of the Galaxy. The survey concentrates on fainter Milky Way stars of various spectral and luminosity classes with $g$ band magnitudes between 14.0 and 20.3. The spectra it contains are from 212 regions of sky covering a total of 3500 square degrees, scattered over three quarters of the celestial
sphere, though with an emphasis on low Galactic latitudes. From the spectra, radial velocities have been obtained accurate to $4 \mathrm{kms}^{-1}$ for stars brighter than $\mathrm{g}=18$ and better than $15 \mathrm{kms}^{-1}$ for those brighter than $\mathrm{g}=20$. Photometries are also provided for $u, g, r, i$ and $z$ bands, as are astrometry data (accurate to 100 mas), and determinations of metallicity and other stellar atmosphere parameters where an SNR exceeding 10 per resolution element is available. All things considered, the SEGUE data represent an excellent resource for Galactic Archaeology in all of the major Galactic substructures and may be used as a stand-alone resource with 6 dimensions of phase space as well as metallicity data all available from the one dataset. Still the number of stars included and the region of sky surveyed are still quite restrictive, and particularly in the Galactic halo or in cases where rare spectral types are used as tracers, there may simply not be enough coverage to properly identify and characterize substructure.

The RAdial Velocity Experiment (RAVE - see Steinmetz et al. 2006) should provide a substantial compliment to the SEGUE data at least for Southern Hemisphere stars. The project aims to obtain mid-resolution $(R=7500)$ spectra of up to one million stars using the Six Degree Field Multi-Object Spectrograph on the 1.2 m UK Schmidt Telescope at Siding Spring. The spectra are concentrated on the Ca-triplet region (841.0 nm -879.5 nm ) in an effort to determine metallicity as well as temperature and surface gravity for the surveyed stars, which will be chosen to have a magnitude in $I$ band in the range from 9 to 12 . Radial velocities will be determined to better than $3.4 \mathrm{kms}^{-1}$, marking a small improvement over the SEGUE data, while proper motions are included from external sources such as Tycho-2. As of the third data release (Siebert et al., 2011), 77461 individual stars had been surveyed, so the quantity of data is still considerably smaller than that available from SEGUE.

As the 'crescendo' to this review of stellar surveys, one particular project in the preparatory stages is set to supersede all the others - the ambitious Gaia space mission. A review of the Gaia mission is found in de Bruijne (2012), wherein the basic capabilities of the Gaia satellite are discussed. Gaia is set to measure the parallaxes, positions and proper motions of the one billion brightest stars in the sky - a truly astronomic endeavor! The stellar parallax measurements obtained by the satellite are expected to be accurate to within $25 \mu a s$ for stars brighter than 15th magnitude. Accompanying this astrometric data will be low-resolution
spectroscopic ( $R \approx 11500$ ) and photometric data covering the range from 330 nm to 1000 nm allowing the radial velocity to be measured to within $1-15 \mathrm{kms}^{-1}$ and metallicity and other parameters of the stellar atmospheres to be determined. The mission is planned to launch in 2013 with final results expected by 2021. By comparison to the other surveys already discussed, this mission represents a new generation for Galactic Archaeology. Not only will cooler main sequence stars comparable to the Sun be visible out to beyond 10 kpc but more luminous stars will be visible throughout the Local Group and even in external galaxy clusters, taking Galactic archaeology to new places quite literally. This data will provide the representative survey of the Galaxy really needed to unravel its past and to study galactic evolution in a more general sense. In closing, it should however be cautioned that 2021 is almost a decade away and budget restraints may yet curtail the ambitious scale of Gaia, and even if they do not, the next ten years should be ones of productivity in Galactic Archaeology. Hence the more immediate, albeit less ambitious surveys will be the raw material utilized to push forward the boundary of knowledge in the mean time.

### 1.5 Dark Matter and the Predictions of $\Lambda$ CDM Cosmology

The requirement for the existence of dark matter was first identified observationally by Fritz Zwicky (Zwicky, 1933). Upon studying the high velocities of member galaxies of the Coma Cluster, he realized that their orbits must enclose substantially more matter than could be attributed to visible galaxies alone in order for them to remain bound, hence implying the existence of some unseen, yet significant component of matter (Sahni, 2004). Rotation curves for individual galaxies were also subsequently shown to imply significant amounts of matter not associated with the luminous component of the galaxies (see Figure 1.3). Studies of the Cosmic Microwave Background (CMB) and the Universe's abundance of deuterium have indicated that ordinary baryonic matter - matter made up of baryons (i.e. protons and neutrons) - constitute a mere $4 \%$ of the total mass/energy content of the Universe and that non-baryonic matter must contribute a much larger fraction, ~30\% (Sahni, 2004). Various properties and forms have been suggested for the elusive dark matter, of which the $\Lambda$ Cold


Figure 1.3: A schematic conveying the disparity between observed and expected galaxy rotation curves (Sahni, 2004)

Dark Matter ( $\Lambda \mathrm{CDM}-\Lambda$ being the cosmological constant) model has been the most successful at explaining the primordial 'power spectrum of density fluctuations' and its evolution to its present state.

In $\Lambda$ CDM Cosmology, the dark matter's constituent particles exhibit a small, non-relativistic velocity dispersion (hence they are termed cold), having decoupled from baryonic matter and energy after they had slowed to non-relativistic speeds (Sahni, 2004). Associated with the particles is a 'free-streaming distance' $\lambda_{f s}$ that relates the mean distance traveled by the particles while still relativistic, before they slow to non-relativistic velocities. Since CDM cosmology already assumes 'cool' particles, this distance is not very long and so free streaming can only disrupt the primordial density distribution on small scales - hence giving rise to small-scale structure soon after the big bang. The opposite to this scenario is borne out by the Hot Dark Matter model, in which density inhomogeneities first appear on larger scales before fragmenting into the building blocks of individual galaxies - i.e. a top-down cosmology. CDM Cosmology in contrast is a bottom-up or hierarchical cosmology in which
smaller structures appear first in the Universe and over time undergo gravitational clustering into larger structures such as clusters and eventually into the super-cluster-filament/ void frothy structure observed today. $\Lambda$ CDM cosmology differs from the earlier standard CDM cosmology in that the mass density $\Omega_{m}$ is chosen to be 0.3 of the total mass-energy density (as opposed to 1) with Hubble constant (at $\mathrm{z}=0$ ) $\mathrm{h} \sim 70 \mathrm{kms}^{-1} \mathrm{Mpc}^{-1}$, thus providing for a better fit to the shape of the current observed power spectrum.

With regard to the actual form of the dark matter, several possibilities have been proposed which can generally be summarized into two fundamental categories - the non-baryonic WIMPs (Weakly Interacting Massive Particles) and the baryonic MACHOs (MAssive Compact Halo Objects). In particular, the neutralino particle has been put forward as a strong contender for the CDM particle. The proposed neutralino is a WIMP with energy in the 100 - 1000 GeV range and is both stable and neutral so that it does not scatter light. Jungman et al. (1996) describes the neutralino as "the best motivated and most theoretically developed" of the WIMP particles and goes on to outline how it might be detected and how its abundance might be determined. Indeed, schemes are underway aimed at the direct detection of neutralinos on the Earth via their gamma-ray emitting interaction with nuclei in a detector - similar to the generation of x-rays in an x-ray tube. At least some of the missing matter however, is going to exist in the form of MACHOs such as distant white dwarfs, brown dwarfs and other low-luminosity bodies in the halo but there are theoretical and observational constraints on the percentage of dark matter made up of such baryonic matter. Theoretically, baryonic matter is not particularly successful at 'growing substructure' from the small primordial density fluctuations in the universe due to its strong coupling with radiation. On the other hand, if most of the dark matter is non-baryonic and thus not coupled to the radiation, this matter can clump together much earlier so that the comparatively small percentage of baryons simply fall into these ready made over-densities shortly afterward (Sahni, 2004). An example of observational constraints on the size of the baryonic component of dark matter is found in Alcock et al. (2000) where the low count rate of micro-lensing events in the direction of the Large Magellanic Cloud over a 5.7 year period is used to constrain the halo mass tied up in MACHOs to $\sim 20 \%$. Whether MACHO or WIMP, the fact remains that the matter is dark and will not be directly observable to the astronomer - with the exception
of the odd MACHO as more sensitive telescopes become available. Hence it would appear that, at least for the time being, the study of the Galactic dark matter will be restricted to the astronomical indirect measurement of the halo mass distribution (with several methods described in the next section) and the independent detection of WIMPs by particle physicists in the laboratory.

Before leaving this discussion of the $\Lambda$ CDM cosmology, it must be noted that this model is a 'best-fit' model only and is not without its own shortcomings. Two principal examples are outlined in Sahni (2004). Firstly, the model predicts an over abundance of halo substructure or subhalos which, if assumed to be accompanied by a luminous baryonic component, are not currently observed. Secondly, CDM predicts a so-called 'cuspy core,' with N-body simulations producing a halo density dropping off more steeply in the central regions than is observed such that $\rho$ is proportional to $r^{-1}$. With regard to the first problem, Diemand et al. (2007) describes the results of "Via Lactea," the highest resolution simulation of the Galaxy to date, which predicts that the Milky Way halo should possess 124 subhaloes with masses comparible to the Galaxy's dwarf satellite galaxies, yet according to McConnachie (2012), only $\sim 25$ such galaxies have been observed. This begs the question: where are the missing satellites? Diemand et al. (2007) goes on however to identify two studies which may hold the answer to this. A local group model by Kravtsov et al. (2004) suggests that galaxy formation will only initiate in the most massive ( $>10^{9} M_{\odot}$ ) subhalos while Moore et al. (2006) find that only those subhalos forming very early on in the galaxy assembly process (at redshifts $z>12 \pm 2$ i.e. before the epoch of reionization) with masses above the atomic cooling mass ${ }^{1}$ can support the formation of a luminous component. Diemand et al. (2007) subsequently found that when the "Via Lactea" simulation was run backward through time only two subhalos were found to comply with each of Krastov and Moore's requirements the same two in each case - which is a much better match to the number of satellites found to date in the Milky Way halo. Furthermore, Sahni (2004) highlights the fact that powerful winds from star formation and early supernovae may be responsible for clearing potential low mass satellites of what baryonic matter they might of had initially. With regard to the 'cuspy core' problem, Sahni (2004) goes on to draw attention to the fact that complex processes in galactic cores such as bar formation and baryon-dark-matter interactions are not
treated adequately in the simulations to date.

### 1.6 Resolving the Matter - Methods for Measuring the Dark Matter Distribution

There are a variety of methods available to ascertain a broad picture of the dark matter distribution in galactic halos, of which three principal techniques are now discussed. The first method is that of gravitational lensing. In the last section, discussion was made of the use of microlensing to determine the percentage of dark matter attributable to MACHOs. Here we are concerned with strong lensing, where for instance a quasi-stellar object (QSO) is lensed by a foreground galaxy, and the contribution of substructure in the lense galaxy to the resulting flux distribution. Lense galaxy substructure in the form of dark subhalos will manifest itself as flux anomalies and milliarcsecond distortions in the image of the source object (Metcalf and Madau, 2001). A study into the feasibility of using such phenomena to map the subhalo distribution in the halos of lens galaxies is made in Riehm et al. (2008). Here, a test is proposed where a QSO is already known to be lensed on the arc second scale so as to ensure a suitably well-aligned, massive halo as the lensing object. Conditions are then favourable for the detection of subhalos in the $10^{6}-10^{10} M_{\odot}$ range based on the milliarcsecond distortions to the imaged QSO. Still, the study finds that the most realistic models currently available for the density distribution within typical subhalos do not bode well for the likelihood of their detection. Their density drops off with distance from the core at a more gradual rate than earlier models, yielding separations in the source image too small to resolve with the current generation of telescopes. Even if some subhalos are detectable, this method is not strictly in the realm of local cosmology, with inferences having to be drawn from the distant lensing galaxies as to how the halos of more local galaxies should be structured.

[^0]A much more direct method is proposed in the detection of gamma rays from annihilation of WIMPs such as from the chief contender - the neutralino. Diemand et al. (2007) goes so far as to produce an all sky map of the possible annihilation flux based on the "Via Lactea" simulation. They find that halo substructure should provide an overall boost to the annihilation signal from a galaxy when compared to a smooth halo distribution. Since the annihilation rate is proportional to the square of the density, a map of halo substructure may soon be possible with the upcoming Gamma Ray Large Area Telescope (GLAST) which has a field of view covering approximately one sixth of the sky and sub-degree resolution at energies greater than 1 GeV . Based on the "Via Lactea" run, subhalo luminosity is predicted to be directly proportional to the mass of the subhalo, with even comparatively small examples visible to such a telescope when they are close to the Sun. The background noise from the Galactic centre is expected to hinder observations toward Sagittarius and in the Galactic Plane in general so observations may be best made looking away from these regions. Whatever the simulations may show, however, studies such as this are based heavily on assumptions and so, until such a time as observational evidence is available to support such ideas, it is important to focus on those methods that are independent of the precise nature of dark matter, relying only on its gravitational effects.

Such a method is found in the kinematic study of currently detectable halo structures such as the stellar streams found in the Andromeda halo and that of our own galaxy. Studies have been made into the feasibility of such methods for constraining the distribution of massive subhalos, notably by Ibata et al. (2002b) and Johnston et al. (2002) with some success predicted upon the availability of deeper 6D-phasespace surveys such as will be undertaken by Gaia. Ibata et al. (2002b) present the results of N-body simulations and their implications for the possibility of inferring the presence of dark matter clumps from their heating effects on stellar streams. Specifically, a $10^{6} M_{\odot}$ globular cluster is modeled with $10^{4}$ particles and placed in a variety of smooth and lumpy galactic potentials both spherical and oblate. It is found that, assuming a spherical potential, the tidal stream from the cluster after a 10 Gyr period remains dynamically cold if the potential is smooth, with a width at its narrowest similar to the tidal radius of the initial cluster model. If the smooth halo is populated with subhalos so that a mere $1 \%$ of the halo mass is tied up in this substructure, the emergent
stream from the model cluster over the same time interval becomes significantly dynamically heated and hence physically wider and more diffuse. If, contrary to an earlier study (Ibata et al., 2001b) that will be discussed shortly, the Galactic halo is not spherical, but rather significantly oblate, the effect of the resulting precession of the cluster orbit can be distinguished from that of heating from subhalo disruption when the integrals of motion of the stream - the total energy and angular momentum (particularly the z -component) per unit mass - are plotted with respect to each other. As a result, this particular method is indeed a possibility whatever the structure of the halo, but as is statistically determined using the 2MASS survey (Ibata et al., 2002a), too few stream members per disrupted globular cluster are available in the presently available data - and with incomplete phase space information - to make such streams detectable, with the stream from the disrupted Sgr Dwarf being the only one discernable from the data. Alas, the Sgr Dwarf is too large, with stellar velocities too dispersed for the subtle effects of heating from Galactic subhalos to be easily distinguishable within its stream. Hence, it is concluded that this method must wait for the Gaia data before the level of halo substructure can reasonably be determined. Johnston et al. (2002) concur with this conclusion but they do find that data for the Sgr Stream is sufficient to isolate some dynamical heating due to 'lumpiness' in the halo, although they point out that the observed scattering may be accounted for by the effects of the Large Magellanic Cloud (LMC) alone. Further, they predict that even an improved data sample for the stream is unlikely to improve on the deductive possibilities of the technique due to the alignments of the orbits of the two progenitor satellites. It is pointed out however, that future deep halo surveys may allow detection of colder extended streams from other Milky Way satellites that are relatively unaffected by the LMC and ideal for probing the halo substructure.

Whilst a study of the subhalo distribution in the Galactic halo may not yet be practical, initial investigations regarding the overall shape of the Milky Way halo and mass of the M31 halo have already taken place. Ibata et al. (2001b) determine with a high level of confidence that the Milky Way halo cannot be significantly oblate. The study used the Automatic Plate Measuring Facility halo carbon star (APM) survey (Totten and Irwin, 1998), which utilized Palomar Sky Survey plates and those from the UK Schmidt Telescope, to examine the distribution of carbon stars and their possible association with known halo structures. Carbon
stars were chosen as the structure tracers of choice owing to their high intrinsic luminosity, rarity, distinct photometry and intermediate age, all of which act to make such stars easily identifiable and useful markers of recent Galactic accretion. Of the 75 carbon stars identified, 38 were found to lie within $10^{\circ}$ of the great circle on the celestial sphere corresponding to the predicted Sgr Dwarf orbit and a further 28 within a similar proximity to the projected orbit of the Magellanic Clouds as represented in Figure 1.4 using a pole-count analysis. These represent $6 \sigma$ and $4 \sigma$ overdensities respectively with regard to the statistically expected value of $\sim 10$ counts. To further illustrate the significance of these results, simulations were run which factored in the sky coverage of the survey plates employed and randomly positioned stars accordingly, and despite 1000 runs, no such overdensities were produced. Because the Sagittarius Stream delineated by these stars is observed as an approximate great circle, Ibata et al. (2001b) suggest that the orbit traced by the Sgr Dwarf must occupy a region of spherically symmetric gravitational potential since orbital precession must otherwise take place with orbital angular momentum no longer conserved. To better understand the evolution of the Sgr Dwarf responsible for the presently observed stream, the team represented the progenitor galaxy first as compact and then as a more loosely bound structure, evolving it each time within a Galactic potential with mass distribution:

$$
\rho(R, z)=\rho_{0}\left(\frac{s}{r_{0}}\right)^{-\gamma}\left(1+\frac{s}{r_{0}}\right)^{\gamma-\beta} e^{\frac{\rho_{t}^{2}}{\tau_{t}^{2}}}
$$

where $r_{0}$ is the core radius, $r_{t}$ is the truncation radius and $\gamma$ and $\beta$ are the power law indices for in and outside of the core respectively. Two particular halo models were investigated based on observational constraints, each with slightly different parameters input into the mass distribution equation. Further to this, each of these halos was simulated for 3 different circular velocities ( $v_{c}-$ determined at 50 kpc ) and 11 different values of the halo density flattening $\left(q_{m}\right)$. In short, both progenitor models were evolved in 66 different versions of the Galactic potential in order to find the combination best fitting observations. It was found that those models with low flattening (e.g. $q_{m} \geq 0.9$ ) are a much better match to the observed carbon star distribution whilst those halos with $q_{m} \leq 0.7$ are refuted with high confidence. Hence Ibata et al. (2001b) conclude that the Galaxy cannot be significantly oblate throughout


Figure 1.4: A pole-count analysis of the APM survey carbon stars where the number of carbon stars lying within $10^{\circ}$ of a given great circle are represented at the pole of the respective great circle using contour lines. The poles of the Sgr Dwarf stream are at $1=90^{\circ} / 270^{\circ}, \mathrm{b}=15^{\circ} /-15^{\circ}$ and those for the Magellanic stream are at $1=170^{\circ} / 350^{\circ}, \mathrm{b}=-5^{\circ} / 5^{\circ}$. (Ibata et al., 2001b)
the Galactocentric radii occupied by the orbit.
As a further example of the utility afforded by the study of halo stream kinematics, Ibata et al. (2004) uses the detection of 'giant stellar stream' stars in a kinematic survey using the DEep Imaging Multi-Object Spectrograph (DEIMOS) on Keck2 to obtain a mass estimate for the dark matter halo of the Andromeda Galaxy (M31). The measurement is made using a realistic galaxy model (Klypin et al., 2002) incorporating the disc and bulge components of the galaxy in addition to the dark halo. From the radial velocity gradient of stream stars in the 9 surveyed fields with confirmed stream components, a mass of $7.5_{-1.3}^{+2.5} \times 10^{11} M_{\odot}$ is obtained for the halo component located within 125 kpc of galactic centre.

### 1.7 The Pan-Andromeda Archaeological Survey

Up to this point we have concentrated our discussion of Galactic Archaeology on the Milky Way Galaxy. Due to our position within it, it has long been the only galaxy for which
deep, comprehensive survey data has been available. But this is no longer the case, with the completion in 2011 of the Pan-Andromeda Archaeological Survey (PAndAS).

The origins of the PAndAS survey lie in the 25 -square-degree survey of the disk and inner halo of M31 undertaken with the 2.5 m Isaac Newton Telescope (INT). The survey sought to identify the transition between the disk and inner halo, but identified extensive substructure and culminated in the discovery of the Giant Stellar Stream (Ibata et al., 2001a). A comprehensive study of the stellar density and metallicity was undertaken by Ferguson et al. (2002) on a field-by-field basis using the INT data. In an effort to map the full spatial extent of the Giant Stellar Stream, the whole southern quadrant of the M31 halo out to 150 kpc was mapped using the 3.6 m Canada-France-Hawaii Telescope (CFHT) on Mauna Kea, with an extension out to M33 more than 200 kpc from M31 (Ibata et al., 2007). With the wealth of substructure discovered, and given the large window of the M31 halo already covered, it was then decided to map the remaining three quadrants out to 150 kpc with CFHT. This major undertaking marked the official birth of the PAndAS survey, with the initial results published in Nature in 2009 (McConnachie et al., 2009). In total, the survey incorporates some 400 square-degrees of sky covering most of the constellation of Andromeda, with extensions into Cassiopeia and Triangulum. It covers the entire halo of M31 out to 150 kpc as well as that of M33 out to 50 kpc . A map of the survey showing the extent of its coverage just prior to completion is presented in Fig. 1.5.

PAndAS is a deep photometric survey which has been undertaken in two bands, $g$ and $i$ using CFHT with the MegaCam instrument. MegaCam is an array of 36, $2048 \times 4612$ pixel CCD chips, covering approximately one square degree on the sky with a resolution perfectly matched to the 0.7 " median seeing atop Mauna Kea. The MegaCam $g$ and $i$ band filters have a very similar throughput to the SDSS filters, with $g$ spanning from approximately $4000 \AA$ to $5700 \AA$ and $i$ from $6700 \AA$ to $8500 \AA$ see Gwyn (2010). A comparison of the two filter sets with the corresponding SDSS filters is illustrated in Fig 1.6. Each of the PAndAS fields reaches a depth of approximately magnitude 25.5 in $g$ band and 24.5 in $i$ band, though data incompleteness is noticeable at these magnitudes.


Figure 1.5: The Pan-Andromeda Archaeological Survey. This map of the PAndAS survey was generated just prior to its completion. It is generated from the most metal poor stars only and thus highlights the location of the various satellite dwarf galaxies. Also visible is the complex network of tidal streams marking the trails of past galaxy interactions. (McConnachie, 2010)

Since the first PAndAS data has become available, a great many studies have been undertaken across a diverse range of topics concerning the structure of the M31 halo system. Possible tidal interactions have been investigated and numerous satellite galaxies, globular clusters and streams have been detected. McConnachie et al. (2009) details the discovery of "stars and coherent structures" that are very likely the remains of ancient dwarf galaxies long since cannibalized. They also identify the remnants of a recent encounter between M31 and M33, and conclude that the wealth of halo structure present in the survey provides excellent evidence for the validity of hierarchical galaxy formation. The globular cluster system of the outer halos of both M31(Mackey et al., 2010) and M33 (Cockcroft et al., 2011) have been investigated, with a strong correlation identified between prominent streams and the locations of the known globular clusters. The presence of a large number of dark matter haloes


Figure 1.6: MegaCam Filter Set response compared with corresponding SDSS filter responses. (Gwyn, 2010)
has also been suggested by Carlberg et al. (2011), after a study of the 120 kpc long North West Stream found density fluctuations that should not arise in a smooth galactic potential. The locations and masses of known dwarf galaxies are also insufficient to explain the density variations. Many new satellites have also been discovered from the PAndAS survey, including Andromedas XVIII, XIX and XX (McConnachie et al., 2008), XXI and XXII (Martin et al., 2009) XXIII-XXVII (Richardson et al., 2011) and XXX (Irwin, 2012).

In summary, PAndAS represents our first opportunity to study an entire galaxy halo system from an unobstructed view point outside the galaxy. Though the survey is now complete, its legacy has just begun as the many studies underway continue to unravel the secrets of Galaxy formation.

### 1.8 The Importance of Position

Due to the enormous distances separating us from all astronomical objects, with out considerable effort, the Universe remains a purely two-dimensional realm. For the local universe, we can use the Earth's orbit as a base line to measure the angular parallax of an object, and derive a distance accordingly. Further afield at the distance of M31 however, even the 300 million km diameter of the Earth's orbit is of little use in gaging distances, and hence we must turn to indirect means. Nevertheless, the prospect of PAndAS in three-dimensions is an exciting one which would allow us to constrain orbits much more accurately and fully explore the matter distribution of the M31 halo.

At the distance of M31, there are several Standard Candles that could potentially be used as our distance gage. Two that are commonly invoked are Cepheid Variable and RR Lyrae stars. Indeed it was the Cepheid Variable that provided the first measure of the distance to the "Spiral Nebulae" thus establishing them as "Island Universes" external to our own Milky Way. The "Spiral Nebulae" targeted were of course M31 and M33 (Hubble, 1925). Nevertheless, Cepheid Variable stars are rare and require multiple epochs of observation to determine their light curves and hence use the Period-Luminosity relation to derive a distance. RR Lyrae stars are much more common than Cepheids but also much fainter and still require multiple observation epochs for distance measurements. Hence we turn our attention to the Tip of the Red Giant Branch (TRGB) standard candle.

The Red Giant Branch forms the backbone of the average metal poor galaxy and at the distance of M31, given the photometric depth of the PAndAS survey, it accounts for almost all of the stars observed to form any given structure. The TRGB standard candle is therefore applicable to even the most sparsely populated object and can even be used to gage distances at multiple points along streams. It also requires only one epoch of observation and hence is readily applicable to a large scale survey such as PAndAS. A study by Salaris and Cassisi (1997) has shown that Cepheid and RR Lyrae determined distances are consistant with those obtained using the TRGB to within $5 \%$.

The TRGB standard candle arises due to the properties common to all Red Giant Branch stars in a particular mass and luminosity range as they approach the onset of core helium
fusion. Such stars first enter the Red Giant Branch toward the end of their life when their source of core hydrogen is depleted. To fuse the helium ash left over in their core requires an immense pressure which the core density is as yet insufficient to produce, and so hydrostatic equilibrium is instead maintained by hydrogen fusion in a shell around the core. This process continues for the duration of the star's life on the Red Giant Branch, with the star gradually becoming more luminous as more and more energy is produced in the hydrogen fusion shell. Due to a relationship between the core helium mass and the luminosity of the star, the rate at which the luminosity increases grows as the star continues its evolution (Salaris et al. 2002; Zoccali and Piotto 2000). This means that more stars will be observed at the fainter end of the Red Giant Branch than at the brighter end, with the result that the luminosity function of a particular object is observed to follow a power law trend (Méndez et al., 2002).

As the star continues its evolution toward the bright end of the Red Giant Branch, the increasing buildup of helium ash in the core steadily increases the core density. In the particular mass range applicable to the TRGB standard candle, the stellar core succumbs to electron degeneracy before helium fusion can ignite and so all such stars have very similar core properties which in turn yields very similar energy outputs in the surrounding hydrogen fusion shell (see Iben and Renzini 1983, particularly Fig. 7). At the very instant of core helium fusion, the stars are at their most luminous and hence lie at the bright tip of the Red Giant Branch before undergoing the Helium Flash as the core pressure becomes sufficient for helium fusion to ignite. At this point, the stars contract and their luminosity diminishes as they enter life on the horizontal branch, resulting in a sudden truncation at the bright end of the luminosity function - i.e. the TRGB. One of the earliest detailed studies of the evolution of Population II stars toward the TRGB can be found in Hoyle and Schwarzschild (1955). A schematic summarizing this evolution is presented in Fig. 1.7.

In order to make use of the near-constant luminosity of the TRGB as a distance gage, it is usual to take measurements in the near infra-red region of the spectrum where dependence on metallicity is minimal. Indeed, Lee et al. (1993) show that for metallicities in the range $-2.2<[\mathrm{Fe} / \mathrm{H}]<-0.7$, the absolute magnitude of the TRGB in Johnson-Cousins $I$ band is constant to within 0.1 magnitudes, where in $V$ band it varies by 1.3 magnitudes. This small


Figure 1.7: Schematic showing the evolution in temperature and luminosity of an intermediate mass, metal poor star. The star 'turns off' the Main Sequence onto the Red Giant Branch (RGB) after exhausting its core supply of hydrogen. The star expands and cools as it fuses hydrogen in a shell surrounding the core. At the onset of core helium fusion, the star has reached the Tip of the Red Giant Branch (TRGB) from which point it cools and contracts and enters life as a Horizontal Branch (HB) star. When it exhausts its core supply of helium it continues to fuse helium in a shell around the core once again becoming more luminous and following a path approaching the RGB asymptotically. At this stage in its evolution the star is hence known as an Asymptotic Giant Branch (AGB) star. Note that stars spend only a tiny fraction of their life time as an AGB star in comparison to the time they spend as RGB stars and hence AGB stars are much rarer and so do little to diminish the contrast of the TRGB in an object's luminosity function.
variation is a consequence of the near-constant absorption in the stellar atmosphere in nearinfrared wavelengths. Using very accurate $I$ band photometry for the globular cluster $\omega$ Cen, Bellazzini et al. (2001) derived the absolute magnitude of the TRGB as $M_{I}=-4.04 \pm 0.12$. The MegaCam $i$ bandpass is however significantly different to Johnson-Cousins $I$ band and so for our PAndAS photometry, it is more suitable to use $M_{i}=-3.44 \pm 0.12$ as derived in Bellazzini (2008) for SDSS $i$ band. This is justified given the similar throughputs of the MegaCam and SDSS $i$ band filters as illustrated in Fig. 1.6.

At this point, having now introduced the burgeoning field of Galactic Archaeology, the

PAndAS survey and the unique opportunity it has provided to explore galaxy evolution in action, we come to the specific aims of the research contained in this thesis. The highest ambition any research thesis can aspire to, is to make an original and significant contribution to the field furnished with clear and accurate results. This is indeed a major underlying motivation for this thesis, though of course, the contribution must inevitably be a specialized one in a field with such enormous scope. To this end, the focus is concentrated on the satellite system of M31. With the known satellite population of the M31 halo so greatly increased in the last 5 years, largely thanks to the PAndAS survey, the time is ripe for a renewed study of the three-dimensional spatial structure of the system. Such a study has the potential to shed light on the past evolution of the satellite system as well as the distribution of matter within the M31 halo. It will also provide for a much needed comparison with the satellite system of the Milky Way and, when combined with velocity information, will facilitate a new, more accurate determination of the M31 halo mass. But before such a study can be undertaken, accurate satellite positions derived consistently via a single method are paramount. Hence, we turn our attention toward the development of a brand new algorithm for locating the TRGB - in particular, one that takes into full account all prior information available about the object's luminosity function.

# Building the Framework for a new TRGB 

## Algorithm

### 2.1 The RGB Tip Finding Problem

Given the broad applicability of the Red Giant Branch tip magnitude as a standard candle, it is not surprising to find that it is invoked frequently for distance measurements within the Local Group. Identifying the magnitude of the TRGB accurately however is not without its challenges, and hence many have resorted to simple "eyeball" measurements, read off from the Luminosity Function (LF) of the object in question. Such an approach is acceptable perhaps for distance measurements to a single, well populated object, but it falls short of the task when a consistent measurement is desired for numerous objects within the same group, or when the LF is poorly populated and the bright edge of the Red Giant Branch (RGB) is
not clear-cut. There is also the problem of ascribing an accurate measurement uncertainty to such an approach. It is therefore desirable to have an automated routine which, given the object LF, returns the most likely position of the RGB truncation with a measure of the uncertainty in this position.

The development of such a TRGB-finding algorithm is not without its difficulties however. Binned luminosity functions by their very nature suffer from Poisson noise, and thus star counts in two neighboring bins may differ by a significant factor. This is a serious problem when the primary task of our algorithm is to locate a sudden jump in star counts that might signal the bright edge of the RGB. It is however, less problematic for those objects exhibiting well populated RGBs. There is also the question of how the LF is affected by the stellar "background" contribution. After all, if the background LF contribution can be isolated perfectly and subtracted from the net LF for the object field, the RGB tip magnitude is simply the brightest non-zero bin remaining.

These issues have been approached in various ways over the years, and a more detailed literature review is provided in Chapter 3 (see Paper I Introduction), but relevant developmental landmarks are discussed below.

The first attempt at an automated tip-finding routine was introduced by Lee et al. (1993), who employed what is essentially an edge-finding technique, similar to what one might encounter in image processing. Instead of a 2D matrix however, the 'image' is the onedimensional, binned luminosity function and the edge finding kernel is a one-dimensional Sobel kernel. The LF is convolved with this kernel, and peaks are produced at the locations where the discontinuity in star counts is greatest. With this method, they find that they can regularly recover the location of the tip, accurate to within 0.2 of a magnitude. Whilst this approach represents the first automated, repeatable TRGB finding method, the size of the uncertainties limits its usefulness. At the distance of M31 for instance, an uncertainty of 0.2 magnitudes corresponds to an uncertainty of approximately $\pm 70 \mathrm{kpc}$ in the distance. The edge-finding method of Sakai et al. (1996) improves on this technique significantly by addressing the luminosity function binning issue via Gaussian smoothing, so that stars no longer fall in one single bin but rather contribute to all bins.

As shall be seen in the next section, some similar techniques to these were experimented
with in the earliest days of the work contained in this thesis. Nevertheless, even with the inclusion of Gaussian smoothing of the LF, the issue of the Poisson noise is still a major concern with any pure edge-finding algorithm. Such techniques also ignore the distinction between object and background contributions to the LF and in so doing, throw away valuable information that might be used to constrain the location of the tip. Hence it is arguable that a model-fitting approach is superior to simple edge-fitting, in that it is less susceptible to the effects of Poisson noise, and more versatile with respect to the incorporation of prior knowledge.

The base method introduced in Chapter 3; Paper I makes use of these advantages by modeling both the background LF and signal or RGB LF separately. The RGB component of the model, whereby the RGB is approximated by a truncated power law, is inspired by Méndez et al. (2002). As in the base method of Chapter 3, they employ a maximum likelihood approach where the model parameters are updated at each iteration, and the likelihood of the model being correct given the data is evaluated. They assume a fixed functional form for the background bright-ward of the tip however, as well as a fixed value for the RGB slope. A more sophisticated approach is to fit the functional form of the background on a case-by-case basis using a suitable (and separate) background field. Likewise, the RGB slope can be set as a separate free parameter.

### 2.2 Early Trials of TRGB Finding Algorithms

During the preliminary, "pathfinding" phase of the development of the base algorithm of Chapter 3, various edge-finding algorithms were experimented with, some of which are now discussed. The relevant code can be found in Appendix B ('EdgeFinder7.f95' and 'RGBPeakFinder6.f95').

The very first algorithms tested made use of artificial luminosity functions, where a deliberate 'kink' could be placed in the LF, and various algorithms used to recover the location of that kink. The kink was generated by summing two luminosity functions together, one displaced toward fainter magnitudes relative to the first such that the brightest non-zero bin in the second LF would mark the beginning of the RGB. In effect, the first LF simulated
the 'background' contamination whilst the second represented the luminosity function of the actual RGB. The prominence of the discontinuity at the beginning of the RGB could be controlled by adding a constant value to the RGB component of the LF. With the model LF set up in this way, it can then be populated with the desired number of stars.

The first edge detecting algorithms implemented on this artificial data were comparable to the kernel convolution method of Lee et al. (1993). Starting from the bright edge of the luminosity function, the gradient between each consecutive pair of magnitude bins was measured and stored. Once the whole LF had been scanned, the magnitude of the induced kink (i.e. the TRGB) was taken to be the fainter of the two consecutive bins for which the maximum gradient was recorded. An equivalent method replaced the measure of gradient with that of the angle subtended by each consecutive set of three magnitude bins. The central bin of the set producing the smallest angle was then taken as the magnitude of the TRGB. Both of these approaches are of course susceptible to confusing a noise spike in the LF for the 'TRGB' if the RGB truncation is not suitably prominent.

In an effort to lessen the sensitivity of the algorithm to Poisson noise, the possibility of fitting either a single polynomial or polynomial splines to the LF was investigated. If a suitable polynomial interpolation of the LF magnitude bins could be found, one would effectively have a smoothed LF, hopefully devoid of problematic noise spikes. The location of the tip could then be determined from the turning points in the second derivative of the fitted polynomial. This approach is fraught with difficulties however, as the degree of the polynomial or number of splines required depends on how smooth the LF is to begin with. If the TRGB truncation lies amidst noise spikes of comparable prominence, it will be very difficult to choose a polynomial which preserves the discontinuity in star counts at the TRGB whilst simultaneously smoothing out the surrounding noise spikes. Furthermore, such an approach inevitably requires a case-specific setup by the user and thus introduces significant biases into the measurement process.

In addition to the above tests carried out on artificial data, further experimentation was carried out on the luminosity functions of real objects. One fairly successful test incorporated a Gaussian smoothing, similar to that employed by Sakai et al. (1996). The dependence of the LF on binning was removed by replacing each star with a normalized Gaussian of some
user-specified width and summing all Gaussians together. The width of the Gaussian was chosen so as to produce the desired level of smoothing in the resulting LF, but in practice, this value should be dependent on the photometric error at the magnitude in question (as is the case in the base method of Chapter 3). Having produced this smoothed version of the LF, the magnitudes at which significant star-count discontinuities occurred could be identified from the function's second derivative. The results of applying this process to the colour-magnitude diagram of the M31 satellite galaxy Andromeda I (illustrated in Fig. 2.1) is presented in Fig. 2.2. It shows the smoothed Andromeda I luminosity function, created by replacing each star with a Gaussian of width ${ }^{1} 0.2$ magnitudes. The superimposed function in red is the second derivative of the LF, weighted by the star counts at that magnitude. It denotes inflection points in the LF gradient. It is clear to the eye that the inflection point corresponding to the TRGB is that identified at $i_{0}=20.77$, which would correspond to a distance to Andromeda I of 695 kpc . This is roughly consistent with the distances in the literature, though as shall be clear from later measurements, the degree of smoothing applied to the LF has shifted the tip brightward by (predictably) $\sim 0.1$ magnitudes.

As is clear from Fig. 2.2, this approach does provide a useful complement to a simple "eyeball" measurement, providing the user with a computationally based readout of the most likely tip locations. Precisely which location is the correct location is left to the discretion of the user, and hence the method stumbles when the onset of the RGB is not clear to the naked eye. As was the case for the polynomial fitting, there also remains the problem that the degree of smoothing required will vary from object to object, and the measurement is thus biased by the user's choices and lacks consistency.

The results of all of these trials culminated in the realization that any method that is to perform consistently across all luminosity functions, will need to abandon the hope that the effects of Poisson noise can somehow be eliminated from the luminosity function. If the method is to perform consistently across all luminosity functions that might be encountered, the data contained therein should be accepted for what it is in its raw form, and a suitable model developed that best explains it. In this way, we can incorporate our expectations of

[^1]

Figure 2.1: Colour-Magnitude Diagram for the dwarf spheroidal galaxy Andromeda I. It contains all stars in the PAndAS survey with $i$-band magnitude $i_{0}$ in the range $18<i_{0}<26$, located within a circular field of radius $0.1^{\circ}$ centered on Andromeda I. The red lines indicate the colour-cut imposed on the data in order to improve the contrast between the RGB and background stars. The Red Giant Branch of the M31 Giant Stellar Stream is visible as a second, faint RGB on the red side of the Andromeda I RGB. It can be seen much more prominently in Paper I, Fig. 2 where a larger field size is plotted.
what form the LF might take were it so well populated that Poisson noise was no longer an issue, and then let the data decide which version of the model approximates it best.

### 2.3 A Simple Maximum Likelihood Test

In order to gain a thorough understanding of how the RGB tip magnitude might be ascertained via model fitting, it is essential to "start simple." As the primary objective of any such algorithm is to locate a sharp discontinuity in star counts, we can begin by approximating the luminosity function with a simple step function, normalized so as to contain unit area. We can then set the 'step' at a particular location, populate the resulting LF with the desired number of star magnitudes and then attempt to recover the position of the step from those


Figure 2.2: A smoothed version of the Andromeda I luminosity function, built using the stars plotted in Fig. 2.1 after rejecting those stars outside of the colour-cut indicated. The superimposed function in red is the second derivative of the smoothed LF, divided by the LF height, and indicates the location of inflection points in the LF gradient. A blue arrow points out the inflection point at $i_{0}=20.77$ corresponding to the RGB tip. Note that the falloff in star counts faint-ward of $i_{0}=23.5$ is due to data incompleteness
star magnitudes. The code pertaining to this section can be found in the program 'spikes.f95' in Appendix B.

In using a maximum likelihood approach, we note that our model LF - i.e. our normalized step function - can actually be considered a probability distribution. It tells us the probability of finding a star at any given magnitude. Bright-ward of the step where we have only background stars, the probability of finding a star is lower but faint-ward of the step we have both background stars and stars from the object's RGB and so we expect more counts at a given magnitude. We can make use of this probability distribution both to generate our random sample of stellar magnitudes, and to recover the original state of the model that produced them.

To produce our random magnitude sample, we make use of a random number generator but weight the likelihood of drawing a particular magnitude by our step function. This can be


Figure 2.3: Random realization of a model luminosity function. It contains 1000 stars whose magnitudes were drawn at random from a step function probability distribution (shown as a red dashed line). The step is located at a magnitude of 0.4 and the signal and background components are in the ratio $0.7: 0.3$ respectively.
done by taking the integral of the step function, which gives us the cumulative area under the step function. This is equivalent to the probability of finding a star bright-ward of a particular magnitude. We then multiply the total area under the step function by a number between 0 and 1 generated by the random number generator and then find the magnitude corresponding to this value of the integral. In so doing we generate a 'random realization' of the step function in question. In practice, any number of random realizations can be generated from a single step function but the larger the sample, the better it will resemble the model that was used to produced it. Fig. 2.3 shows one such random realization. It contains 1000 stars and was produced from a step function where the fraction of stars contained in the RGB and background components were 0.7 and 0.3 respectively. The step was located at a magnitude of 0.4.

To recover the state of the model luminosity function which produced our artificial data, we need to test the likelihood of the model reproducing that data for a range of different


Figure 2.4: Schematic showing the scaling of the normalized step function as the step location is moved to fainter magnitudes.
step positions. Since our model LF is a probability distribution, the likelihood of the model producing a star at a particular magnitude is simply the value of the model at that magnitude. Thus if we are given a single star at a particular magnitude, we can find the version of the model that is most likely to have produced it by sliding the step location from the bright to the faint end of the LF and then noting which step location produced a maximum likelihood at the magnitude of the star. This process is illustrated in Fig. 2.4.

As can be seen in the figure, it is critical to the procedure that an equal area be preserved under the model for all step positions tried. The model represents all possible magnitudes at which a star can be observed and so the area underneath it should be unity. Likewise, the ratio of the background and RGB contributions to the model should remain constant. Given these requirements, the form of the probability distribution for the location of the step, as determined from a luminosity function containing a single star can be understood. As the step slides to fainter magnitudes, the RGB height rises to preserve equal area under the model. Hence the likelihood of the model producing the star at the observed magnitude grows at an increasing rate. Eventually however, the step slides past the magnitude at which the star is
observed and the likelihood for any subsequent step positions being correct therefore drops immediately to the background height.

In reality of course, one star does not constitute a useful luminosity function. We must therefore understand how probability distributions for the step location are produced when more than one star is present. To determine the likelihood of a particular state of the model producing two or more stars at their specifically observed magnitudes, we multiply together the likelihoods of the model separately producing each star. Equivalently, the final probability distribution for the step location is simply the product of the individual distributions generated for each star. This result is illustrated in Fig. 2.5 for a 10 star luminosity function.

Whilst the probability distribution of Fig. 2.5 (b) reveals a large uncertainty in the location of the step, little else can be expected from such a poorly populated luminosity function. More important is the fact that we have a reliable measure of the uncertainty in the most likely step position identified, assuming of course that our chosen model is a good approximation for the mechanisms responsible for producing the LF, as is the case here. Nevertheless, for well populated luminosity functions, the step location is generally locatable with considerable precision, as can be seen in Fig. 2.6 which has been generated from the luminosity function of Fig. 2.3.

From the simple tests presented in this section, the power, as well as the relative simplicity of the maximum likelihood model-fitting approach begin to emerge. Such an approach is particularly robust, as every single data point is taken into account every time a possible RGB tip location is considered. This greatly desensitizes any algorithm that implements it against the localized effects of Poisson noise. The approach is also versatile, being applicable to any model no matter how simple or complex. If we are to advance from a simple 'step' luminosity function however, we shall also need other tools at our disposal.


Figure 2.5: Probability distributions for the location of the step (i.e. RGB tip) in a 10 star, 'step' luminosity function. Fig. (a) shows the individual probability distributions resulting from each star whilst Fig. (b) shows the product of these individual distributions which forms the probability distribution given the whole luminosity function. As in Figures 2.3 and 2.4, the model which produced the stars consisted of RGB and background contributions in the ratio of $0.7: 0.3$ and the step was located at a magnitude of 0.4 . Note that the total area under all distributions in both (a) and (b) is unity, with all possible locations of the step position represented. This follows from the normalization of the model for all step positions.


Figure 2.6: Probability distribution for the location of the step in the 1000 star 'step' luminosity function of Fig. 2.3.

### 2.4 The Markov Chain Monte Carlo Method

The tests of §2.3 approximated an object's luminosity function with a single-parameter step function. All possible states of such a model are obtainable by changing only the step location. As such, it was not computationally intensive to calculate the likelihood for the model at every possible step location, even for very small increments in the step location and for very densely populated luminosity functions. A better model however, would take into consideration other aspects of the LF, such as the background to RGB contribution ratio and the slope of the RGB component. But the number of possible states of the model increases exponentially with the number of free parameters, and hence so too does the computation time. Hence the deterministic approach used above quickly becomes impractical as we add increased complexity to our model. For this reason it is advantageous to adopt a Monte Carlo method, which builds up a picture of the likelihood of the state space of the model by taking samples of the model likelihood at randomly chosen parameter values. Our method of choice is the Markov Chain Monte Carlo (MCMC) Method.

The MCMC works via the construction of a Markov Chain. Named for the Russian Mathematician Andrey Markov, it is essentially a statistically representative sample of all possible states of a model, given a specific set of data. The sample is a chain, in the sense that each newly chosen parameter set is affected by the previous. The creation of the chain is however a 'memoryless' procedure, with each newly chosen state having no dependence on past states in the chain, with the exception of the state that immediately preceded it. To properly represent the differing likelihoods of various states, the chain should be created in such a way as not to prohibit the possible recurrence of certain states. The extent to which the chain explores the full state space of the model is of course dependent on the length of the chain. One must therefore be careful to insure that the chain is indeed long enough to be a true representation of the model states and their likelihoods. A detailed overview of Markovian models with examples can be found in the online reference work Meyn and Tweedie (1993).

There are various ways that a Markov Chain can be constructed, but the one employed for the analysis contained in this thesis makes use of the Metropolis-Hastings algorithm (Metropolis et al. 1953, Hastings 1970). An excellent introduction to this algorithm, with examples can be found in Gregory (2005). The algorithm essentially provides an 'intelligent' random walk through the state space of the model by preferentially choosing steps toward model parameter sets that have a higher likelihood. To initiate the algorithm, one must first choose a 'jumping distribution' as well as a starting value, for each of the model parameters. The jumping distribution is a probability distribution that defines how likely a jump to any given parameter value is, given the present value of the parameter. For our implementation of the MCMC, a Gaussian jumping distribution is chosen due to its symmetry and preference for jumps to nearby parameter values. The appropriate width of the distribution for each parameter is chosen through experimentation.

With the jumping distributions and initial parameter values defined, new parameter values are proposed and the model likelihood for those values is calculated. If the likelihood is greater with these new parameter values than the current model likelihood, the proposed parameter values are automatically accepted and thus define the next model state in the chain.


Figure 2.7: The random walk in the RGB tip magnitude over 1000 iterations, resulting from the application of the MCMC to the Andromeda I luminosity function. The model LF being sampled consisted of two free parameters, the RGB tip magnitude and the slope $a$ of the power law chosen to approximate the RGB component of the LF. (Generated using 'BayesianTRGB_ANDI.f95' - see Appendix C)

If the model likelihood is smaller with these new parameter values, the probability of accepting them is weighted by the ratio $r$ of the two likelihoods, with $r=\frac{\mathcal{L}_{\text {proposed }}}{\mathcal{L}_{\text {current }}}$. Specifically, a random number $d$ is drawn between 0 and 1 and the proposed step is taken only if $d \leq r$. An example of the random walk that results from many iterations of this process is provided in Fig. 2.7, which shows the sampled values of the RGB tip parameter in the Andromeda I LF in a Markov Chain of 1000 iterations. Note that this figure has been generated with prior knowledge of the approximate magnitude of the tip, and hence there is no obvious lead-in period. In practice, it is advantageous to excise the first thousand or so iterations (depending on the step size and initial starting value) from the sample, to remove the initial walk to the general location of the best-fit parameter value.

The single-parameter random walk exemplified by Fig. 2.7 is of course, only a partial description of the Markov Chain. It ignores what any other parameters are doing at each
iteration. It is therefore useful to plot each pair of parameters against each other to observe any correlation between them. This said, whilst many parameters might be coupled, our ultimate goal is to obtain the absolute probability of particular values of each parameter without regard for what any other parameters are doing, thus creating a probability distribution for the parameter. This is achieved by marginalizing over the other parameters. If our model were to have $n$ free parameters, the probability of one of the parameters having some particular value is equal to the integral under the $n-1$ dimensional surface relating the probability distributions of all the other parameters at that value. In practice, it is actually very straightforward to create the probability distributions of individual parameters of a Markov Chain, by simply representing the number of occurrences of each parameter value as a histogram. This process is illustrated in Fig. 2.8.

It is clear from Fig. 2.8 (b) that 1000 iterations of our MCMC algorithm is insufficient to fully explore the likelihoods of the various model states. Indeed, for the probability distributions presented in Papers I and II, hundreds of thousands of iterations were found necessary before the distributions were as smooth as one would expect to achieve via a deterministic approach. With the underlying model luminosity function used for the analysis in each of these papers being defined by only two free parameters, the MCMC approach might be described as 'overkill' for the model used. It must be stressed however that this approach has been built into the algorithm from the outset in order to facilitate added model complexity with only minimal coding changes.

### 2.5 The Bayesian and the Frequentist

Statisticians are of two minds with regard to the nature of probability and how it should be calculated. The traditional 'Frequentist' view, as the name suggests, holds that the probability of an event is related to the frequency with which it occurs over a large number of samples. Strictly speaking, it is the limit in that frequency as the number of samples goes to infinity and therefore it can never be calculated exactly. Frequentists hence speak of confidence intervals, an interval over which they have some degree of confidence that an event


Figure 2.8: Generating the probability distribution for a single parameter via marginalization. Both Figures (a) and (b) are generated from the same Markov Chain as Fig. 2.7. Fig. (a) shows the value of the RGB slope $a$ corresponding to each value of the RGB tip. As the fitted model LF is defined solely by these two parameters, the figure portrays all information contained in the Markov Chain. Fig. (b) portrays the binned probability distribution in the RGB tip position. The height of each bin reflects the number of data points in Fig. (a) recorded in the magnitude range represented by each bin. It is clear that a longer chain is warranted if it is to form a truly representative sample of the state space of the model. (Generated using 'BayesianTRGB ANDI.f95' - see Appendix C)
will be observed. Beginning with the work of the mathematician Thomas Bayes (Bayes and Price, 1763), a new perspective on the nature of probability began to emerge however. It essentially regarded probability as a subjective construct, the distribution of which depended on the prior knowledge available to the investigator. Herein lies the power of the Bayesian technique. The probability of an event can be weighted by our knowledge of the laws that govern it, thus producing a probability distribution which characterizes the credibility of the measurement.

Central to Bayesian Inference is Bayes theorem. The modern form of Bayes Theorem has its origin in the work of Pierre-Simon Laplace (Laplace, 1812). It is commonly written as follows:

$$
\begin{equation*}
P(A \mid B)=\frac{P(B \mid A) \times P(A)}{P(B)} \tag{2.1}
\end{equation*}
$$

where $P(A \mid B)$ is the probability of $A$ being true, given observation $B ; P(B \mid A)$ is the probability of $B$ being true, given $A$; and $P(A)$ and $P(B)$ are the absolute probabilities of $A$ and $B$ respectively. As an example, suppose someone comes across a set of 10 old coins in the attic, apparently all identical. Upon closer inspection however, they note that one of them has a particular mintmark. After some investigation they determine this mintmark to be quite rare, being found on only $1 \%$ of coins of that type. They also learn however, that $30 \%$ of all coins of the type are fakes, and that $90 \%$ of the fakes bear the mintmark. So what is the probability that they have found an authentic example? We have

$$
\begin{align*}
P(\text { real } \mid \text { mintmark }) & =\frac{P(\text { mintmark } \mid \text { real }) \times P(\text { real })}{P(\text { mintmark })} \\
& =\frac{0.01 \times 0.7}{0.01 \times 0.7+0.9 \times 0.3}  \tag{2.2}\\
& =2.5 \% .
\end{align*}
$$

Note however that if the finder of the coin had not done their research, they would have had to assume that approximately 1 in 10 coins of the type bear the mintmark, and that the authenticity of the coin was not in question. Prior knowledge has thus completely changed their perspective.

We can equally well apply Bayes Theorem to our model-fitting RGB tip finding algorithm
thus:

$$
\begin{equation*}
f\left(M_{p} \mid D\right)=\frac{f\left(D \mid M_{p}\right) \times f\left(M_{p}\right)}{f(D)} \tag{2.3}
\end{equation*}
$$

where $f\left(M_{p} \mid D\right)$ is the 'posterior' distribution in model parameter $p$ after taking into account the data $D$. The function $f\left(D \mid M_{p}\right)$ is the distribution of likelihoods obtained from the model for the various values of $p, f\left(M_{p}\right)$ is our 'prior' or initial assumption as to the distribution in $p$ before accounting for the data, and $f(D)$ is the probability distribution for the data points - i.e. the luminosity function. Since $f(D)$ is constant regardless of the model parameters, it scales all probabilities by the same amount and hence is dealt with the by the normalization of the posterior distribution. In the examples of $\S 2.3$ and $\S 2.4$, a 'uniform' prior has always been assumed, such that $f\left(M_{p}\right)=c$, a constant. If we are to better constrain our posterior distributions however, we should incorporate all prior information we have on the object studied.

### 2.6 Prior Information

As shall be seen in the next two chapters, prior information has been incorporated into our TRGB algorithm in a variety of ways. Some of our prior knowledge of the objects under study has been applied in the form of an independent 'prior' distribution which is multiplied by the likelihood distribution as per Eq. 2.3, while other prior information has been built into the model luminosity function directly. The model LF by its very nature reflects our assumptions as to the form it would take were it populated with an infinite number of stars, thereby eradicating any Poisson noise. We assume a truncated power law for the RGB and fit the background component directly with a polynomial. This process is discussed in detail in Paper I, but it is important to realize that the model LF applied to each object is 'custom built' for that object. The likelihood distribution generated for each parameter thus already incorporates prior information we have for the object.

To expand on this, it shall be seen in the next two chapters that the areas under the RGB and background components of the model LF are set based on the average stellar density in the object field as compared with that of a suitable 'background' field. The background field is chosen so that it lies very close to the object field, and is usually in the form of a large
rectangle centered on the object, but with the object field subtracted so that the object RGB contributes negligibly to the LF of the background field ${ }^{2}$. In so doing, we are effectively assigning a prior constraint on how many stars we expect to find at a given magnitude on either side of the RGB tip. If the average stellar density in the background field is very low compared with that of the object field, than we do not expect to find very many stars that are not true members of the object RGB.

With the base method presented in Paper I, the ratio of the RGB to background model contributions is held constant for all stars. When the density matched filter is applied in Paper II however, individual stars are assigned a weight that reflects our prior expectation as to the probability of their being true object members. Using this information, we can tailor our model LF not only to the object in question, but to the specific star in question. The ratio of the RGB to background model contributions is calculated for each star individually, based on its position within the object's density profile.

In addition to the prior information that has been incorporated directly into the model LF, additional prior information has been incorporated in the conventional sense, i.e. as a prior probability distribution. The default prior distribution is a uniform prior, an assumption of equal probability for all parameter values. Various prior distributions were experimented with in the initial development of the base algorithm, each one devised so as to put some constraint on the distance at which the object can be found. A Gaussian distribution could be chosen for example, with the center of the distribution corresponding to the distance of M31 and the width reflecting our assumptions as to the extent of the halo. Alternatively, a flattened Gaussian could be chosen so as to yield equal probability over some desired distance range whilst cutting off sharply outside of that range. The priors on the tip magnitude $m_{T R G B}$ and RGB slope $a$ for both papers have been simple top hat functions, such that $19.5 \leq m_{\text {TRGB }} \leq$ 23.5 and $0<a<2$ are predicted with equal probability whilst values outside of those ranges are assumed impossible. A more subjective prior is assumed in Paper II for the object

[^2]distance however, namely the density profile of a simple spherical halo along the line of sight passing through the object (see Paper II, Fig. 6).

The net result of the inclusion of the prior information discussed above, is that we transform our simple maximum likelihood technique of $\S 2.3$ into an 'educated' tip finding algorithm. We effectively combine the best of both worlds by automating the tip finding process, but at the same time imparting some of the intuition to the tip finding algorithm that we would use if estimating the tip magnitude from the LF by eye. Such is the power of Bayesian Inference, that we can combine the information obtainable from one lone data sample with all other knowledge we can possibly infer about the circumstances which produced it.
"Every statistician would be a Bayesian if he took the trouble to read the literature thoroughly..."
D. V. Lindley (1986)

## Paper I: A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. I.

## Paper I Preface

This chapter presents the first of three papers which, together, represent the very heart of the thesis. They are perhaps best thought of as a 'trilogy,' as each one flows naturally into the next and, though written to stand as independent contributions in their own right they are best understood in the light of their companion papers. Unlike papers II and III however, Paper I is primarily a 'methods' paper. It lays the foundations for a new approach to the long standing Tip of the Red Giant Branch problem which is further developed in Paper II where it is applied to the majority of the M31 satellites. A preliminary analysis of the satellite distribution is provided in that paper with a study of the halo density profile but the real 'fruits of the labour' follow in Paper III which contains a thorough analysis of the satellite spatial distribution which has led to some very interesting results. Whilst papers II and III shall likewise be introduced with their own preface, chapters 1 and 2 arguably form the real 'preface' for this paper and hence it seemed apt that this paper should be introduced with a discussion of its place within a broader picture. Note also that the principal programing code pertinent to the material presented in this paper can be found in Appendix C.
(The published version of Conn et al. 2011 is reproduced by permission of the AAS)

# A BAYESIAN APPROACH TO LOCATING THE RED GIANT BRANCH TIP MAGNITUDE. I. 

A. R. Conn ${ }^{1,2}$, G. F. Lewis ${ }^{3}$, R. A. Ibata ${ }^{2}$, Q. A. Parker ${ }^{1,4}$, D. B. Zucker ${ }^{1,4}$, A. W. McConnachie ${ }^{5}$, N. F. Martin ${ }^{6}$, M. J. Irwin ${ }^{7}$, N. TanVir ${ }^{8}$, M. A. Fardal ${ }^{9}$, and A. M. N. Ferguson ${ }^{10}$<br>${ }^{1}$ Department of Physics \& Astronomy, Macquarie University, Sydney 2109, Australia<br>${ }^{2}$ Observatoire Astronomique, Université de Strasbourg, CNRS, 67000 Strasbourg, France<br>${ }^{3}$ Institute of Astronomy, School of Physics, A29, University of Sydney, Sydney, NSW 2006, Australia<br>${ }^{4}$ Australian Astronomical Observatory, Epping, NSW 2121, Australia<br>${ }^{5}$ NRC Herzberg Institute of Astrophysics, Victoria, British Columbia V9E 2E7, Canada<br>${ }^{6}$ Max-Planck-Institut für Astronomie, D-69117 Heidelberg, Germany<br>${ }^{7}$ Institute of Astronomy, University of Cambridge, Cambridge CB3 0HA, UK<br>${ }^{8}$ Department of Physics and Astronomy, University of Leicester, Leicester LE1 7RH, UK<br>${ }^{9}$ Department of Astronomy, University of Massachusetts, LGRT 619-E, Amherst, MA 01003-9305, USA<br>${ }^{10}$ Institute for Astronomy, University of Edinburgh, Royal Observatory, Edinburgh EH9 3HJ, UK<br>Received 2011 March 1; accepted 2011 July 15; published 2011 September 30


#### Abstract

We present a new approach for identifying the tip of the red giant branch (TRGB) which, as we show, works robustly even on sparsely populated targets. Moreover, the approach is highly adaptable to the available data for the stellar population under study, with prior information readily incorporable into the algorithm. The uncertainty in the derived distances is also made tangible and easily calculable from posterior probability distributions. We provide an outline of the development of the algorithm and present the results of tests designed to characterize its capabilities and limitations. We then apply the new algorithm to three M31 satellites: Andromeda I, Andromeda II, and the fainter Andromeda XXIII, using data from the Pan-Andromeda Archaeological Survey (PAndAS), and derive their distances as $731_{(-4)-17}^{(+5)+18} \mathrm{kpc}, 634_{(-2)-14}^{(+2)+15} \mathrm{kpc}$, and $733_{(-11)-22}^{(+13)+23} \mathrm{kpc}$, respectively, where the errors appearing in parentheses are the components intrinsic to the method, while the larger values give the errors after accounting for additional sources of error. These results agree well with the best distance determinations in the literature and provide the smallest uncertainties to date. This paper is an introduction to the workings and capabilities of our new approach in its basic form, while a follow-up paper shall make full use of the method's ability to incorporate priors and use the resulting algorithm to systematically obtain distances to all of M31's satellites identifiable in the PAndAS survey area.


Key words: galaxies: general - galaxies: stellar content - Local Group
Online-only material: color figures

## 1. INTRODUCTION

The tip of the red giant branch (TRGB) is a very useful standard candle for gauging distances to extended, metal-poor structures. The tip corresponds to the very brightest members of the first ascent red giant branch (RGB), at which point stars are on the brink of fusing helium into carbon in their cores and hence contracting and dimming to become horizontal branch stars. The result is a truncation to the RGB when the color-magnitude diagram (CMD) for an old stellar population is generated, beyond which lie only the comparatively rare asymptotic giant branch (AGB) stars and sources external to the system of interest. The (highly variable) contamination from such objects provides the principal obstacle to simply "reading off" the tip position from the RGB's luminosity function (LF) and the truncation of the AGB can even masquerade as the TRGB in certain instances. The $I$ band is the traditionally favored region of the spectrum for TRGB measurements, minimizing the interstellar reddening that plagues shorter wavelengths, while keeping dependence on metallicity lower than it would be at longer IR wavelengths. It should also be remembered that stars approaching the TRGB generally exhibit peak emission in this regime. Iben \& Renzini (1983) determined that low-mass ( $<1.6 M_{\odot}$ for Population I, $<1 M_{\odot}$ for Population II), metalpoor $([\mathrm{Fe} / \mathrm{H}]<-0.7$ dex) stars older than 2 Gyr produce a TRGB magnitude that varies by only 0.1 mag . More recently,

Bellazzini et al. (2001) determined the tip magnitude to lie at an $I$-band magnitude of $M_{\text {TRGB }}=-4.04 \pm 0.12$. This low variation can be attributed to the fact that all such stars have a degenerate core at the onset of helium ignition and so their cores have similar properties regardless of the global properties of the stars. The result is a standard candle that is widely applicable to the old, metal-poor structures that occupy the halos of major galaxies. Distances derived from the TRGB, unlike those from a Cepheid variable or RR Lyrae star, for example, can be determined from a single epoch of observation, making it very useful for wide-area survey data. Furthermore, Salaris \& Cassisi (1997) confirmed agreement between Cepheid and RR Lyrae distances and TRGB distances to within $\sim 5 \%$.
Until Lee et al. (1993) published their edge-finding algorithm, the tip had always been found by eye, but clearly if the widereaching applications of the TRGB standard candle were to be realized, a more consistent, repeatable approach was in order. The aforementioned paper shows that, if a binned LF for the desired field is convolved with a zero sum Sobel kernel $[-2,0,+2]$, a maximum is produced at the magnitude bin corresponding to the greatest discontinuity in star counts, which they attribute to the tip. Using this method, they were able to obtain accuracies of better than 0.2 mag. Sakai et al. (1996) set out to improve on this approach by replacing the binned LF and kernel with their smoothed equivalents. To do this, they equate each star with a Gaussian probability distribution whose FWHM
is determined by the photometric error at the magnitude actually recorded for the star. Then, rather than each star falling within a particular bin, it contributes to all bins via a normalized Gaussian centered on the magnitude recorded for it. This is illustrated in Equation (1):

$$
\begin{equation*}
\Phi(m)=\sum_{i=1}^{N} \frac{1}{\sqrt{2 \pi \sigma_{i}^{2}}} \exp \left[-\frac{\left(m_{i}-m\right)^{2}}{2 \sigma_{i}^{2}}\right] \tag{1}
\end{equation*}
$$

where $m$ is the magnitude of the bin in question and $m_{i}$ and $\sigma_{i}^{2}$ are the central magnitude and variance, respectively, of the Gaussian probability distribution for the $i$ th star. This method halved the error associated with the non-smoothed version of the algorithm and an identical smoothing is hence just incorporated into the model LF for our Bayesian approach.

In a more recent variation on the edge detection methods, Madore et al. (2009) once again applied a Sobel kernel, but fit to an LF built from composite stellar magnitudes $T \equiv$ $I-\beta\left[(V-I)_{0}-1.50\right]$ where $\beta$ is the slope of the TRGB as a function of color. This, they argued, results in a sharper output response from the filter, and allows all stars, regardless of color, to contribute equally to the derived tip position. Rizzi et al. (2007) derived a value of $0.22 \pm 0.02$ for $\beta$ after a study of five nearby galaxies, and showed that it is quite consistent from one galaxy to another.

Méndez et al. (2002) made a departure from the simple "edgefinding" algorithms above by adapting a maximum likelihood model fitting procedure into their technique. They pointed out that the LF faintward of the tip is well modeled as a power law:

$$
\begin{equation*}
L\left(m \geqslant m_{\mathrm{TRGB}}\right)=10^{a\left(m-m_{\mathrm{TRGB}}\right)}, \tag{2}
\end{equation*}
$$

where $m \geqslant m_{\text {TRGB }}$ and $a$ is fixed at 0.3 . They then ascribed the location of the tip to the magnitude at which this power law truncates, i.e., $m=m_{\text {TRGB }}$. Brightward of the tip they assumed a functional form

$$
\begin{equation*}
L\left(m<m_{\mathrm{TRGB}}\right)=10^{b\left(m-m_{\mathrm{TRGB}}\right)-c}, \tag{3}
\end{equation*}
$$

where $b$ is the slope of the power law brightward of the tip and $c$ is the magnitude of the step at the RGB tip.

Such a model, though simplistic, is robust against the strong Poisson noise that is inevitable in more sparsely populated LFs, making it a significant improvement over the previous, purely "edge-finding" methods.

Makarov et al. (2006) followed in a similar vein, demonstrating the proven advantages of a maximum likelihood approach over simple edge detection techniques, despite a model dependence. Unlike Méndez et al. (2002) however, they allowed $a$ as a free parameter, arguing its notable variance from 0.3 , and importantly, they smoothed their model LF using a photometric error function deduced from artificial star experiments. One shortcoming of both of these methods, however, is that the most likely parameter values alone are obtained, without their respective distributions or representation of their dependence on the other parameters. Also, with regard to the background contamination, the RGB LF in fact sits on top of non-system stars in the field and so rather than model the background exclusively brightward of the TRGB, the truncated power law of Equation (2) can be added onto some predefined function of the contamination.

Arguably the most successful method developed so far has been that devised by McConnachie et al. (2004). It has been
used to ascertain accurate distances to 17 members of the Local Group (McConnachie et al. 2005). It combines aspects of both "edge-finding" and model fitting to zero in more accurately on the tip. They argued that as the precise shape of the LF at the location of the tip is not known, a simple Sobel Kernel approach that assumes a sharp edge to the RGB does not necessarily produce a maximum at the right location. They instead used a least-squares model-fitting technique that fits to the LF in small windows searching for the portion best modeled by a simple slope function. This, they reasoned, marks the location of the steepest decline in star counts which is attributable to the tip location. This method is capable of finding the tip location accurate to better than 0.05 mag , although is still susceptible to being thrown off by noise spikes in a poorly populated LF.

Despite the merits of previous methods such as these, none of them work particularly well when confronted with the high levels of Poisson noise that abound in the more poorly populated structures of galaxy halos. Furthermore, in such conditions as these where the offset between detected and true tip position will likely be at its greatest, it is of great use to have a full picture of likelihood space, as opposed to merely the determined, most probable value. This has led us to develop a new, Bayesian approach to locating the TRGB, specifically, one that incorporates a Markov Chain Monte Carlo (MCMC) algorithm. As shall become apparent in the next section, such a method is very robust against noise spikes in the LF and allows all prior knowledge about the system to be incorporated into the tip-finding process-something lacking in the previous approaches. Further to this, the MCMC provides for a remarkably simple, yet highly accurate error analysis. It also makes it possible to marginalize over parameters to provide posterior probability distributions (PPDs) of each parameter, or to obtain plots of the dependence of each parameter on every other. In Section 2, a detailed explanation of our approach and its limitations is given. Section 2.1 introduces the method by applying the algorithm to one of M31's brightest dwarf spheroidals, Andromeda I. Section 2.2 discusses the nature of systematic errors that apply to the method. Section 2.3 investigates the accuracy that the basic method (before addition of priors) is capable of given the number of stars populating the LF for the field and the strength of the non-RGB background while Section 2.4 deals with its performance when faced with a composite LF. Section 3 then applies our new approach to two additional M31 dwarf satellite galaxies and Section 4 summarizes the advantages of the method and outlines the expected applicability of the method in the immediate future.

## 2. METHOD

### 2.1. The MCMC Method

The MCMC method is an iterative technique that, given some model and its associated parameters, rebuilds the model again and again with different values assigned to each parameter, in order that a model be found that is the best fit to the data at hand. It does this by comparing the likelihood of one model, built from newly proposed parameter values, being correct for the data, as opposed to the likelihood for the model built from the previously accepted set of model parameters. The MCMC then accepts or rejects the newly proposed parameter values weighted by the relative likelihoods of the current and proposed model parameter values. At every iteration of the MCMC, the currently accepted value of each parameter is stored so that the number of instances of each value occurring can be used to build


Figure 1. Position of Andromeda I relative to the M31 disk. The saturated disk dominates the northwest corner of the field while Andromeda I itself appears as an overdensity within the Giant Stellar Stream (GSS). The GSS in actuality lies well behind Andromeda I, as is evidenced by the CMD in Figure 2. A strict color-cut was imposed on the data to highlight the location of the satellite and the extent of the stream with greatest contrast.
(A color version of this figure is available in the online journal.)
a likelihood distribution histogram-which can be interpreted as a PPD-for each model parameter. Hence, the MCMC is a way of exploring the likelihood space of complicated models with many free parameters or possible priors imposed, where a pure maximum likelihood method would be quickly overwhelmed. With the PPD generated, the parameter values that produce the best-fit model to the data can simply be read off from the peak of the PPD for each parameter. Similarly, the associated error can be ascertained from the specific shape of the distribution. A detailed description of the MCMC with worked examples can be found in Gregory (2005, Chap. 12).

To illustrate the precise workings of our MCMC tip-finding algorithm, its application to a well-populated dwarf galaxy in the M31 halo is described. Andromeda I was discovered by van den Bergh (1971) and at a projected distance of $\sim 45 \mathrm{kpc}$ from M31 (Da Costa et al. 1996), it is one of its closest satellites. Da Costa et al. (1996) ascribed to it an age of $\sim 10 \mathrm{Gyr}$ and a relatively low metallicity of $\langle\mathrm{Fe} / \mathrm{H}\rangle=-1.45 \pm 0.2$ dex which is clearly exemplified in the CMD for Andromeda I presented in Figure 2. Here the RGB of Andromeda I lies well to the blue side of that of the Giant Stellar Stream (GSS) which lies behind Andromeda I but in the same field of view. Mould \& Kristian (1990) provide the first TRGB-based distance measurement to Andromeda I, which they deduce as $790 \pm 60 \mathrm{kpc}$, based solely on a visual study of the RGB. McConnachie et al. (2004) improve on this significantly, producing a distance determination of $735 \pm$ 23 kpc , based on a tip magnitude of $20.40_{-0.02}^{+0.03}$ in the $I$ band.

Andromeda I's position with respect to M31 and the GSS is presented in Figure 1, where the red circle indicates the precise field area fed to our MCMC algorithm. An object-to-background


Figure 2. Color-magnitude diagram for a circular field of radius $0^{\circ} .2$ centered on Andromeda I. Two red giant branches are clearly visible, that of Andromeda I (within the red rectangle color-cut) and that of the Giant Stellar Stream which lies behind Andromeda I in the same line of sight.
(A color version of this figure is available in the online journal.)
ratio (OBR) of 11.0 was recorded for this field with the color-cut applied, based on comparisons of the signal field stellar density with that of an appropriate background field. The data presented in this figure, as with all other data discussed in this paper, were obtained as part of the Pan-Andromeda Archaeological Survey (PAndAS; McConnachie 2009), undertaken by the 3.6 m Canada-France-Hawaii Telescope (CFHT) on Mauna Kea equipped with the MegaCam imager. CFHT utilizes its own unique photometric bandpasses $i$ and $g$ based on the AB system. We work directly with the extinction-corrected CFHT $i$ and $g$ magnitudes and it is these that appear in all relevant subsequent figures. The extinction-correction data applied to each star have been interpolated using the data from Schlegel et al. (1998).

At the heart of our tip-finding algorithm is the model LF that the MCMC builds from the newly chosen parameters at every iteration. The LF is a continuous function which we subsequently convolve with a Gaussian kernel to account for the photometric error at each magnitude. This is achieved by discretizing both functions on a scale of 0.01 mag . Like Méndez et al. (2002), we assume the LF faintward of the tip to follow a simple power law, of the form given in Equation (2); however, we set $a$ as a free parameter. The bin height at each magnitude is then calculated by integrating along this function setting the bin edges as the limits of integration. The value for the bin which is set to contain the RGB tip for the current iteration is calculated by integrating along the function from the precise tip location to the faint edge of the bin. All other bins are then set at 0 . A bin width of 0.01 mag for our model was found to provide a good balance between magnitude resolution, which is limited by the photometric error in the MegaCam data ( $\sim 0.01 \mathrm{mag}$ at $\mathrm{m}=$ 20.5), and the computational cost for a higher number of bins. We stress here, however, that each star's likelihood is calculated from the model independently, so that the actual data LF is "fed" to the MCMC in an unbinned state. A faint edge to the model LF was imposed at $m=23.5$ to remove any significant effects from data incompletion and increasing photometric error.
Further to this, we add a background function to this truncated power law. While the scaling of the background strength relative to the RGB signal strength could be set as another free parameter, and indeed was initially, it makes better use
of our prior information to instead determine the fraction of background stars or "background height" $(f)$ manually. This is achieved simply by calculating the average density of stars in the background field $D_{\text {BG }}$ and in the "signal" field $D_{\text {SIG }}$ with $f$ then being the ratio of the two, i.e., $f=D_{\mathrm{BG}} / D_{\mathrm{SIG}}$. Note that this is not directly the inverse of the object-to-background star ratio, $\mathrm{OBR}=\left(D_{\mathrm{SIG}}-D_{\mathrm{BG}}\right) / D_{\mathrm{BG}}$, as $f$ represents the percentage of all stars lying inside the signal field that can be expected to be external to the object of interest. Hence, when we normalize the area under the model LF so that it may be used by the MCMC as a probability distribution, the background component will have area $f$ while the RGB component will have area $1-f$. Now, with $f$ known, what we then have is a simplified two-parameter model, allowing for faster convergence of the MCMC algorithm.

We have thus devised our model so that the MCMC is tasked with the problem of finding just two parameters, namely the slope of the RGB LF (a) and of course the location of the RGB tip magnitude ( $m_{\text {TRGB }}$ ). For simplicity in this first paper, we impose uniform priors on each of these parameters, where $19.5 \leqslant m_{\text {TRGB }} \leqslant 23.5$ and $0 \leqslant a \leqslant 2$. We also do not account for the color dependence of the tip magnitude which is only slight in the $I$ band (see Rizzi et al. 2007) and for the metalpoor targets examined here, but these effects will be dealt with in future publications. While it is true that two parameters are tractable analytically, we apply the numerical MCMC in order to set the framework for computationally more challenging models with non-uniform priors that will become necessary for the more sparsely populated structures presented in future contributions. There are, however, several more complexities to the model that have yet to be discussed. First, the choice of background function is not arbitrary. It has been found that the best way to model the background is to fit it directly by taking the LF of an appropriate "background" field. The best choice of background field is arguably one that is at similar galactic latitude to the structure of interest, as field contamination is often largely Galactic in origin, and hence closely dependent on angular distance from the Galactic plane. Furthermore, the field should be chosen so that the presence of any substructure is minimal, so as to prevent the signature of another halo object interfering with the LF for the structure of interest.

In addition to these constraints, owing to the low stellar density of the uncontaminated halo, it is preferable that the background field be as large as possible to keep down the Poisson noise and hence it will of necessity be much larger than that of the field of interest. As a result, the main error in the background fit will arise from background mismatching and is not random. In addition, the large background field size may inevitably contain some substructure, requiring removal. This may be done by physically subtracting contaminated portions of the background area, but this is often unnecessary as the CMD color-cut imposed on the signal field must also be applied to the background field, usually ridding the sample of any substantial substructure that may be present. In the case of our Andromeda I background field, however, we have removed a large 2.4 portion crossing numerous streams (as shown in Figure 3) as these streams do trespass into the chosen Andromeda I colorcut. Nevertheless, this is just a precaution, because for wellpopulated systems such as Andromeda I and Andromeda II, the algorithm is impervious to small discrepancies in the functional form of the background.

Once an appropriate background field has been selected, its LF can be fitted by a high-order polynomial. This polynomial then becomes the function added to our model and scaled


Figure 3. Map of the entire PAndAS survey area, with color-cut chosen to favor the low metallicities exhibited by many of M31's satellite galaxies. The three dwarf spheroidal companions of M31 studied in this paper are labeled, along with the signal fields (small circles of radius 0.2 ) and their respective background fields fed to our algorithm. Note that the background fields are chosen to be as narrow as possible in Galactic latitude while retaining as large an area as possible. In each case, the signal field areas are subtracted from their respective background fields to prevent contamination.
(A color version of this figure is available in the online journal.)
by $f$ as described earlier. Our choice of background field for Andromeda I (along with Andromeda II and Andromeda XXIII) and the polynomial fit to its LF are presented in Figures 3 and 4 , respectively.

The other major consideration that has yet to be addressed is the effect of photometric error on the LF. This is dealt with by convolving the initial binned model with a normalized Gaussian whose width is adjusted as a function of magnitude in accordance with the error analysis conducted on the PAndAS data. This is equivalent to the method of Sakai et al. (1996) described in Equation (1). As described earlier, this procedure has the added advantage of making the model independent of binning. It is also important in this stage, as it is at every stage, that the model and all constituent parts are normalized so that the model can be used as a probability distribution.

With these issues addressed, the MCMC algorithm can be set in motion. The $i$-band magnitudes and $(g-i)_{0}$ data for the desired field is read into data arrays, spurious sources are rejected, and a color-cut is imposed to remove as many non-members of the structure's RGB as possible. The same constraints are of course applied to the background field as well. The MCMC then applies preset starting values of $a$ and $m_{\text {TRGB }}$ and builds the corresponding model for the first iteration. Within this iteration, the MCMC proposes new values for each parameter, displaced by some random Gaussian deviate from the currently set values and re-constructs the appropriate model. The step size, or width of the Gaussian deviate is chosen so as to be large enough for the MCMC to explore the entire span of probability space, while small enough to provide a high-resolution coverage of whatever features are present. The ratio of the likelihoods of the two models is then calculated (the Metropolis Ratio $r$ ) and a swap of accepted parameter values made if a new, uniform random deviate drawn from the


Figure 4. Top: CMD for the Andromeda I background field (see Figure 3). The same color-cut is applied as in the CMD for the signal field (Figure 2). Bottom: The binned luminosity function for the background with the fitted polynomial superimposed. A polynomial of degree seven was found adequate to represent the luminosity function.
(A color version of this figure is available in the online journal.)
interval $[0,1]$, is less than or equal to $r$. The calculation of the Metropolis Ratio for our model is exemplified in Equations (4) and (5):

$$
\begin{equation*}
r=\frac{\mathcal{L}_{\text {proposed }}}{\mathcal{L}_{\text {current }}} \tag{4}
\end{equation*}
$$

with the value for each of the likelihoods $\mathcal{L}$ being calculated thus

$$
\begin{equation*}
\mathcal{L}=\prod_{n=1}^{\text {ndata }} M\left(m_{\mathrm{TRGB}}, a, m_{n}\right) \tag{5}
\end{equation*}
$$

with

$$
\begin{aligned}
& M\left(m_{n} \geqslant m_{\mathrm{TRGB}}\right)=\mathrm{RGB}\left(m_{n}\right)+\mathrm{BG}\left(m_{n}\right) \\
& M\left(m_{n}<m_{\mathrm{TRGB}}\right)=\mathrm{BG}\left(m_{n}\right)
\end{aligned}
$$

where

$$
\begin{equation*}
\operatorname{RGB}\left(m_{n}\right)=10^{a\left(m_{n}-m_{\text {TRGB }}\right)} \tag{6}
\end{equation*}
$$

and $\quad \int_{m=m_{\text {TRGB }}}^{m=23.5} \mathrm{RGB} d m=1-f$
and $\int_{m=19.5}^{m=23.5} \mathrm{BG} d m=f$,


Figure 5. Posterior probability distribution for three million iterations of the MCMC on the Andromeda I CMD color-cut presented in Figure 2. The peak probability is located at $i_{0}=20.88$. The distribution is color coded, with red indicating tip magnitudes within $68.2 \%$ (Gaussian $1 \sigma$ ) on either side of the distribution mode, green those within $90 \%$, and blue those within $99 \%$.
(A color version of this figure is available in the online journal.)
where $m_{\text {TRGB }}$ and $a$ are the parameters currently chosen for the model by the MCMC, ndata is the number of stars and $m_{n}$ is the $i$-band magnitude of the $n$th star. BG represents the fitted background function (see Figure 4). The MCMC then stores the new choice for the current parameter values and cycles to the next iteration. In order to ascertain a reasonable number of iterations, the chains for each parameter were inspected to insure that they were well mixed, resulting in posterior distributions that appeared smooth (by eye).
When the MCMC has finished running, the PPD for each parameter is generated. By binning up the number of occurrences of each parameter value over the course of the MCMC's iterations, the probability of each value is directly determined and the most probable value can be adopted as the correct model value for the data. If one assumes a Gaussian probability distribution, then the $1 \sigma$ errors associated with each parameter value can be obtained simply by finding the value range centered on the best-fit value that contains $68.2 \%$ of the data points. As our PPDs are not always Gaussian, our quoted $1 \sigma$ errors in the tip magnitude represent more strictly a $68.2 \%$ credibility interval. We do not fit a Gaussian to our PPDs to obtain $1 \sigma$ errors. Our $1 \sigma$ errors in tip magnitude are obtained by finding the magnitude range spanning $68.2 \%$ of the PPD data points, on one side of the distribution mode and then the other. It must be stressed that these quoted errors are merely an indicator of the span of the parameter likelihood distribution and are no substitute for examining the PPDs themselves. Figures 5 and 6 present the PPD for the RGB tip magnitude based on the Andromeda I CMD (Figure 2) and the best-fit model to the LF for the field, respectively. The PPD for the LF slope $a$ is presented in Figure 7 and a contour map of the distribution of the tip magnitude versus $a$ is presented in Figure 8.

Upon the completion of the algorithm, the RGB tip for Andromeda I was identified at $m=20.879_{-0.012}^{+0.014}$. This corresponds to an extinction-corrected distance of $731_{(-4)-17}^{(+5)+18} \mathrm{kpc}$, where the final errors include contributions from the extinction and the uncertainty in the absolute magnitude of the TRGB (see Section 2.2). The $i$-band extinction in the direction of Andromeda I is taken as $A_{\lambda}=0.105 \mathrm{mag}$ (Schlegel et al.


Figure 6. Four-magnitude segment of the Andromeda I luminosity function fitted by our MCMC algorithm. It is built from 3355 stars. The best-fit model is overlaid in red. The bin width for the LF is 0.01 mag .
(A color version of this figure is available in the online journal.)


Figure 7. Posterior probability distribution obtained for the slope $a$ of the Andromeda I luminosity function. The distribution is a clean Gaussian with the distribution mode at 0.273 .


Figure 8. Contour map of the distribution of the tip magnitude vs. the LF slope $a$. It is noteworthy that there is little correlation between the two parameters, with the peak of the distribution of $a$ more or less independent of tip magnitude. Regardless of any correlation, the respective PPDs of each parameter are the result of marginalizing over the other parameter, and thus take into account any covariance between parameters.


Figure 9. Plot of the distribution of possible distances to Andromeda I obtained through the application of our method. Once again, the colors red, green, and blue denote distances within $68.2 \%, 90 \%$, and $99 \%$ credibility intervals, respectively. (A color version of this figure is available in the online journal.)
1998). The parameters $a$ and $f$ were derived as $0.273 \pm$ 0.011 and 0.083 , respectively. This distance measurement is in excellent agreement with the distance determined by McConnachie et al. (2004). It is noteworthy, however, that our method searches for the TRGB itself as distinct from the RGB star closest to the TRGB as sort out by the method of McConnachie et al. (2004), which would contribute to our slightly smaller distance measurement. A similar discrepancy arises in the case of Andromeda II (see Section 3).

### 2.2. A Note on Distance Errors

Despite the small errors in the tip magnitude afforded by our approach, there are a number of factors that contribute to produce a somewhat larger error in the absolute distance. These arise due to uncertainties both in the extinction corrections applied and in the absolute magnitude of the TRGB in the $i$ band. Both of these contributions are assumed to be Gaussian, where the $1 \sigma$ error in the extinction correction, $\Delta\left\{A_{\lambda}\right\}$, is taken as $10 \%$ of the correction applied, and the error in the absolute magnitude of the tip is expressed in Equation (7) below

$$
\begin{align*}
& \Delta\left\{M_{i}^{\mathrm{TRGB}}\right\} \\
& \quad=\sqrt{\Delta^{2}\left\{m_{i}^{\mathrm{TRGB}}\right\}_{\omega \mathrm{Cen}}+\Delta^{2}\left\{A_{\lambda}\right\}_{\omega \mathrm{Cen}}+\Delta^{2}\{m-M\}_{\omega \mathrm{Cen}}} \\
& \quad=\sqrt{\{0.04\}^{2}+\{0.03\}^{2}+\{0.11\}^{2}} \\
& \quad= \pm 0.12 . \tag{7}
\end{align*}
$$

As we are working in the native CFHT $i$ and $g$ bands, we adopt this magnitude as $M_{i}^{\mathrm{TRGB}}=-3.44 \pm 0.12$, where the conversion from $M_{I}^{\text {TRGB }}$ is based on the absolute magnitude for the TRGB identified for the Sloan Digital Sky Survey (SDSS) $i$ band (Bellazzini 2008). This is justified by the color equations applying to the new MegaCam $i$-band filter (Gwyn 2010). Noting that the largest contribution to this error is that from the distance modulus to $\omega$ Cen, $(m-M)_{\omega \text { Cen }}$, derived from the eclipsing binary OGLEGC 17, we consider only the contributions from the extinction $\left\{A_{\lambda}\right\}_{\omega \mathrm{Cen}}$, which is taken as $10 \%$ of the Schlegel et al. (1998) values, and the apparent tip magnitude determination $\left\{m_{i}^{\mathrm{TRGB}}\right\}_{\omega \text { Cen }}$ and note that our
derived distance modulus may be systematically displaced by up to 0.1 of a magnitude. This then gives us $M_{i}^{\text {TRGB }}=$ $-3.44 \pm \sqrt{0.04^{2}+0.03^{2}}=-3.44 \pm 0.05$. Since our principal motive is to obtain relative distances between structures within the M31 halo rather than the absolute distances to the structures, this offset is not important. Furthermore, as measurements for the $\omega$ Cen distance modulus improve, our distances are instantly updatable by applying the necessary distance shift.

While these external contributions to our distance uncertainties may be taken as Gaussian, the often non-Gaussian profile of our TRGB ( $m_{i}^{\text {TRGB }}$ ) posterior distributions necessitates a more robust treatment then simply adding the separate error components in quadrature. Hence to obtain final distance uncertainties, we produce a distance distribution obtained by sampling combinations of $m_{i}^{\text {TRGB }}, A_{\lambda}$ and $M_{i}^{\text {TRGB }}$ from their respective likelihood distributions, thus giving us a true picture of the likelihood space for the distance. The result of this process for Andromeda I is illustrated in Figure 9. From this distribution, we determine not only the quoted $1 \sigma$ errors but also that Andromeda I lies at a distance between 703 and 761 kpc with $90 \%$ credibility and between 687 and 778 kpc with $99 \%$ credibility.

### 2.3. Initial Tests

In order to gain a better understanding of the capabilities of our method when faced with varying levels of LF quality, a series of tests were conducted on artificial "random realization" data, as well as on sub-samples of the Andromeda I field utilized above. There are two major factors that affect the quality of LF available to work with, namely, the number of stars from which it is built and the strength of the background component relative to the RGB component. Hence to simulate the varying degrees of LF quality that are likely to be encountered in the M31 halo, artificial LFs were built for 99 combinations of background height versus number of stars. Specifically, background heights of $f=0.1,0.2, \ldots, 0.9$ were tested against each of ndata $=10,20,50,100,200,500,1000,2000,5000,10,000$, and 20,000 stars populating the LF.

To achieve this, a model was built as discussed in Section 2.1, with a constant tip magnitude and RGB slope of $m_{\text {TRGB }}=20.5$ and $a=0.3$, respectively, and a background height $f$ set to one of the nine levels given above. The functional form of the background was kept as a horizontal line for the sake of the tests. An LF was then built from the model, using one of the 11 possible values for the number of stars listed above. This was achieved by assigning to each of the ndata stars a magnitude chosen at random, but weighted by the model LF probability distribution-a "random realization" of the model. The MCMC algorithm was then run on this artificial data set as described in the previous subsection with $m_{\text {TRGB }}$ and $a$ as free parameters to be recovered. The tests also assume the photometric errors of the PAndAS survey and further assume that incompleteness is not an issue in the magnitude range utilized. The error in the recovered tip position and the offset of this position from the known tip position in the artificial data $(I=20.5)$ were then recorded. The results are presented in Figures 10 and 11 below. Each pixel represents the average result of ten 200,000 iteration MCMC runs for the given background height versus number of stars combination. Note that the kpc distances given correlate to an object distance of 809 kpc -i.e., $m_{I}^{\mathrm{TRGB}}=20.5$-which is in keeping with distances to the central regions of the M31 halo. Furthermore, all stars of the random realization were generated within a 1 mag range centered on this tip value.


Figure 10. Gray-scale map of the $1 \sigma$ error in tip magnitude obtained for different combinations of background height and number of sources. The actual value recorded for the error (in kpc) is overlaid on each pixel in red. For these tests, we approximate the $1 \sigma$ error as the half-width of the central $68.2 \%$ of the PPD span.
(A color version of this figure is available in the online journal.)

Figures 10 and 11 are intended to serve as a reference for future use of the basic method, with regard to the number of stars required to obtain the distance to within the desired uncertainty for the available signal-to-noise ratio. The results follow the inevitable trend of greater performance when the background height is small and there are many stars populating the LF. There are some minor deviations from this trend but these result from single outlying values whose effects would diminish if a higher number of samples were averaged. It is also noteworthy that the offsets recorded clearly correlate with the $1 \sigma$ errors and are consistently less than their associated errors.
The results of these random realization tests are borne out by similar tests conducted on subsamples of the Andromeda I field. Random samples were drawn containing 335 ( $10 \%$ of the total sample), 200, 100, and 50 stars. These correspond approximately to $10,20,50$, and 100 stars in the 1 mag range centered on the tip. In no case was the derived tip location more than 80 kpc from that identified from the full sample, and the offset grew steadily less as the number of stars in the sample was increased. Furthermore, the offsets were almost always less than the $1 \sigma$ errors.

### 2.4. Algorithm Behavior for Composite Luminosity Functions

When a field is fed to any RGB tip finding algorithm, it must be remembered that field is in fact three dimensions of space projected onto two, and therefore it is possible that two structures at very different distances may be present within it. Such a scenario becomes especially likely when dealing with the busy hive of activity that the PAndAS Survey has come to reveal around M31. The result of such an alignment along the line of sight is an LF built from two superimposed RGBs with two different-possibly widely separated-tip magnitudes. Hence it is important to understand how the TRGB algorithm applied to such a field will respond.
Unlike other algorithms that have been developed, our Bayesian approach provides us with a measure for the probability of the tip being at any given magnitude (the PPD). But this also leads to an important caveat-the selection criteria imposed


Figure 11. Gray-scale map of the offset from the true tip magnitude obtained for different combinations of background height and number of sources. The actual (absolute) value recorded for the offset (in kpc) is overlaid on each pixel in red. These values convey the discrepancy between the true object distance and that recovered by the MCMC . It was necessary to remove the direction of the individual offsets before averaging as the values would otherwise largely cancel out. Examination of the individual offsets shows no significant bias toward either direction however
(A color version of this figure is available in the online journal.)
on the data that is fed to the algorithm biases it strongly toward the structure whose distance we are trying to measure. Taking the Andromeda I measurement of Section 2.1 for example, this satellite sits on top of the GSS which contributes prominently to the field CMD, yet its contribution to the LF fed to the MCMC is almost eradicated by our choice of color-cut. Yet if this stringent color-cut is removed, the algorithm remains surprisingly insensitive to the GSS tip. This is because of another prior constraint we impose on the routine-the background height. With this fixed background imposed on our fitted model, the MCMC looks for the first consistent break of the data from the back-ground-i.e., the tip of the Andromeda I RGB. It is therefore necessary to reinstate the background height as a free parameter of the MCMC to give it any chance of finding the tip of the GSS's RGB. By this stage, enough of our prior constraints have been removed to give the method freedom to choose the best fit of the unrestricted model to the entire data set from the field. Nevertheless, the more (correct) prior information we can feed the algorithm, the better the result we can expect to receive.

Still, while the method has not been tailored toward composite LFs, it is worth noting that it can be used successfully to identify more than one object in the line of sight-a useful ability when the two structures are poorly separated in color-magnitude space. The model used assumes only one RGB and thus one tip; to do otherwise would increase computation times. If two distinct structures are identified by this method and cannot be separated using an appropriate color-cut or altered field boundaries, an appropriate double RGB model should be built to accurately locate the tip for each structure. But even with the basic single-RGB model (which will suffice for the vast majority of cases), at least the presence of a second structure is indicated. If we take the example of Andromeda I again, the ideal way to obtain a distance measurement to the portion of the GSS that sits behind it would be to make a color-cut that favors it and removes Andromeda I, but we can force the algorithm


Figure 12. Posterior probability distribution for the cold sampler chain of a fourchain parallel-tempering regime. The MCMC was run for 1.5 million iterations. The strong peak at $m=20.93$ results from the tip of the Andromeda I RGB, but it has been shifted faintward by the presence of the Giant Stellar Stream, responsible for the peaks at $m=21.29$ and $m=21.35$. Without the addition of parallel tempering, the MCMC is liable to spend an inordinate amount of time stuck in the first major probability peak it encounters.
to consider both structures to demonstrate the extreme case of what might be encountered in a general halo field. The result is two broad bumps in the PPD well separated in magnitude. The nature of the MCMC however is to converge straight onto the nearest major probability peak, seldom venturing far from that peak. This is remedied by the addition to the algorithm of Parallel Tempering.
While an infinite number of iterations of the MCMC would accurately map probability space in its entirety, Parallel Tempering is a way of achieving this goal much more quickly. Parallel Tempering involves a simple modification to the MCMC algorithm, whereby multiple chains are run in parallel. One chain, the "cold sampler" runs exactly as before, but additional chains have their likelihoods weighted down producing a flatter PPD that is more readily traversed by the MCMC. The further the chain is from the cold sampler chain, the heavier the weight that is applied. Every so many iterations, a swap of parameters is proposed between two random but adjacent chains so that even the "hottest" chains eventually affect the cold sampler chain and allow it to escape any local maximum it may be stuck in. The result is a cold sampler chain PPD that is more representative of the full extent of the LF (see Gregory 2005, Chap. 12 for a more detailed discussion). The result of applying a four-chain MCMC to the region of Andromeda I is summarized in the PPD of Figure 12.

While the Andromeda I TRGB is found much less accurately by this method as a result of the removal of our prior constraints for illustrative purposes, it is nevertheless clear that the addition of Parallel Tempering adds to our algorithm the facility to identify other structures in the field that may require separate analysis. Even given a properly constrained model and data set, the safeguard it provides against a poorly explored probability space arguably warrants its inclusion.

## 3. DISTANCES TO TWO MORE SATELLITES

To further illustrate the capabilities of our basic method as outlined in Section 2, we have applied it to two more of M31's brighter satellites, whose distances have been determined in


Figure 13. CMD for a circular field of radius 0.2 centered on Andromeda II. It is more densely populated than the Andromeda I CMD (Figure 2) and is very well defined against the stellar background. The RGB tip is clearly visible at $i_{0} \sim 20.6$.
(A color version of this figure is available in the online journal.)


Figure 14. Posterior probability distribution for three million iterations of the MCMC on a 4 mag interval (see Figure 15) of the Andromeda II CMD selection presented in Figure 13. The peak probability of the distribution is well defined at $i_{0} \approx 20.57$. The distribution is again color coded as in Figure 5, with red, green, and blue corresponding to $68.2 \%, 90 \%$, and $99 \%$ credibility intervals, respectively.
(A color version of this figure is available in the online journal.)
past measurements using a range of methods, including TRGBfinding algorithms. The additional satellites chosen for this study are the relatively luminous dwarf spheroidal Andromeda II and the somewhat fainter, newly discovered Andromeda XXIII dwarf. The location of both satellites within the M31 halo can be seen in Figure 3.

### 3.1. Andromeda II

Andromeda II was discovered as a result of the same survey as Andromeda I using the 1.2 m Palomar Schmidt telescope (van den Bergh 1971). Da Costa et al. (2000) deduce a similar age for Andromeda II as for Andromeda I but with a wider spread of metallicities centered on $\langle\mathrm{Fe} / \mathrm{H}\rangle=-1.49 \pm 0.11$ dex. Our Andromeda II LF was built from a circular field of radius 0.2 centered on the dwarf spheroidal with an OBR of 34.0 recorded.


Figure 15. Four-magnitude segment of the Andromeda II luminosity function fitted by our MCMC algorithm. It is built from 4409 stars. The best-fit model is overlaid in red. The bin width for the LF is again 0.01 mag.
(A color version of this figure is available in the online journal.)


Figure 16. Color-magnitude diagram for a circular field of radius 0.1 centered on Andromeda XXIII. It is much more sparsely populated than those of Andromeda I and Andromeda II. The RGB tip appears to lie just brightward of $i_{0}=21$.
(A color version of this figure is available in the online journal.)

This high OBR is not unexpected with Andromeda II arguably the best populated of M31's dwarf spheroidal satellites. The CMD for this field is presented in Figure 13.

Application of our algorithm to Andromeda II yields a tip magnitude of $i_{0}=20.572_{-0.006}^{+0.005}$ for the RGB which corresponds to an extinction-corrected distance to Andromeda II of $634_{(-2)-14}^{(+2)+15} \mathrm{kpc}$, where the $i$-band extinction is taken as $A_{\lambda}=0.121 \mathrm{mag}$ (Schlegel et al. 1998). This is in good agreement with McConnachie et al.'s (2004) derived distance of $645 \pm 19 \mathrm{kpc}$. Values for $a$ and $f$ were recovered as $0.276 \pm 0.009$ and 0.028 , respectively. The $m_{i}^{\text {TRGB }}$ PPD and best-fit model found by our method are illustrated in Figures 14 and 15, respectively.

### 3.2. Andromeda XXIII

Despite its relative brightness among the other satellites of the M31 system, Andromeda XXIII was only discovered with the undertaking of the outer portion of the PAndAS survey in 2009/ 2010, being too faint at $M_{V}=-10.2 \pm 0.5$ to identify from


Figure 17. Posterior probability distribution for three million iterations of the MCMC on a 4 magnitude interval (see Figure 18) of the Andromeda XXIII CMD selection presented in Figure 16. There are several probability peaks in this instance but the preferred peak lies at 20.885 . The distribution is again color coded as in Figure 5, with red, green, and blue corresponding to $68.2 \%, 90 \%$, and $99 \%$ credibility intervals, respectively.
(A color version of this figure is available in the online journal.)


Figure 18. Four-magnitude segment of the Andromeda XXIII luminosity function fitted by our MCMC algorithm. It is built from 328 stars. The best-fit model is overlaid in red. While the model LF tested by the MCMC retained the resolution of 100 bins per magnitude described in Section 2.1, the data LF is re-produced here at the lower resolution of 0.04 mag per bin to better reveal its structure to the eye.
(A color version of this figure is available in the online journal.)
the SDSS (Richardson et al. 2011). The said paper presents its vital statistics along with those for the other newly discovered satellites Andromeda XXIV-XXVII. It is a dwarf spheroidal galaxy and has the lowest recorded metallicity of the satellites we present with $\langle\mathrm{Fe} / \mathrm{H}\rangle=-1.8 \pm 0.2$. Making use of the deeper coverage of PAndAS in the $g$ band, Richardson et al. (2011) obtain a distance measurement of $767 \pm 44 \mathrm{kpc}$ from the horizontal branch of the CMD.

Andromeda XXIII is a more challenging target for our algorithm in its current form, with less than $\sim 50$ stars lying within the 1 mag range centered on the tip and an OBR of 8.4 for the field and color-cut employed. The CMD for this circular field of radius 0.1 is presented in Figure 16. We find the RGB tip at an $i$-band magnitude of $20.885_{-0.032}^{+0.038}$, which, given an $i$-band extinction of 0.112 mag in the direction of Andromeda XXIII (Schlegel et al. 1998), corresponds to a
distance of $733_{(-11)-22}^{(+13)+23} \mathrm{kpc}$. We derive the values of $a$ and $f$ as $0.270 \pm 0.039$ and 0.105 , respectively. Curiously, the MCMC finds several peaks very close to the major peak in the PPD (see Figure 17), but these are attributable to the lower star counts available in the LF around the tip. This has the effect of creating large magnitude gaps between the stars that are just brightward of the tip so that each individual star can mimic the sudden increase in star counts associated with the beginning of the RGB. As a result, there is a range of likely locations for the tip, but the PPD shows that the object cannot be more distant than 802 kpc nor closer than 601 kpc with $99 \%$ confidence. The best fit model determined by the MCMC is overlaid on the LF in red in Figure 18.

## 4. CONCLUSIONS

The versatility and robustness of our new method can be appreciated from Section 2 and its high level of accuracy is evident from the measurement errors which are consistently smaller than those in the literature to date. In addition, it is our hope that with the correct priors imposed, this new approach carries with it the ability to gauge distances to even the most poorly populated substructures, bringing a whole new range of objects with in reach of the TRGB standard candle. In the case of the M31 halo alone, it will be possible to obtain distances to all of the new satellites discovered by the PAndAS survey-a feat previously impractical using the TRGB. Furthermore, PAndAS has revealed a complicated network of tidal streams that contain valuable information as to the distribution of dark matter within the M31 halo. With our new method, it will be possible to systematically obtain distances at multiple points along these streams, thus providing vital information for constraining their orbits.
The great advantage of our new Bayesian method over a pure maximum likelihood method is the ease with which prior information may be built into the algorithm, making it more sensitive to the tip. Herein lies the great power of the Bayesian approach, whereby the addition of a few carefully chosen priors can reduce the measurement errors 10 fold. The result is an algorithm that is not only very accurate but highly adaptable and readily applicable to a wide range of structures within the distance (and metallicity) limitations of the TRGB standard candle. With instruments such as the 6.5 m infrared James Webb Space Telescope and the 42 m European Extremely Large Telescope expected to be operational within the decade, these distance limitations will soon be greatly reduced. This will bring an enormous volume of space within reach of the TRGB method, including the region of the Virgo Cluster. A tool with which it is possible to apply the TRGB standard candle to small, sparsely populated structures and small subsections of large structures alike is hence, needless to say, invaluable.
A.R.C. thanks Sydney University for allowing him the use of their computational and other resources. In addition, A.R.C. thanks fellow student Anjali Varghese for her assistance with and practical insights with regard to the implementation of parallel tempering. A.R.C. also thanks Neil Conn for assistance in proofreading the document. G.F.L. thanks the Australian Research Council for support through his Future Fellowship (FT100100268) and Discovery Project (DP110100678). N.F.M. acknowledges funding by Sonderforschungsbereich SFB 881 "The Milky Way System" (subproject A3) of the German Research Foundation (DFG).

## REFERENCES

Bellazzini, M. 2008, Mem. Soc. Astron. Ital., 79, 440
Bellazzini, M., Ferraro, F. R., \& Pancino, E. 2001, ApJ, 556, 635
Da Costa, G. S., Armandroff, T. E., Caldwell, N., \& Seitzer, P. 1996, AJ, 112, 2576
Da Costa, G. S., Armandroff, T. E., Caldwell, N., \& Seitzer, P. 2000, AJ, 119, 705
Gregory, P. C. 2005, Bayesian Logical Data Analysis for the Physical Sciences (Cambridge: Cambridge Univ. Press)
Gwyn, S. 2010, http://cadcwww.dao.nrc.ca/megapipe/docs/filters.html
Ibata, R., Martin, N. F., Irwin, M., et al. 2007, ApJ, 671, 1591
Iben, I., Jr., \& Renzini, A. 1983, ARA\&A, 21, 271
Lee, M. G., Freedman, W. L., \& Madore, B. F. 1993, ApJ, 417, 553
Letarte, B., Chapman, S. C., Collins, M., et al. 2009, MNRAS, 400, 1472

Madore, B. F., Mager, V., \& Freedman, W. L. 2009, ApJ, 690, 389
Makarov, D., Makarova, L., Rizzi, L., et al. 2006, AJ, 132, 2729
McConnachie, A. W. 2009, BAAS, 41, 278
McConnachie, A. W., Irwin, M. J., Ferguson, A. M. N., et al. 2004, MNRAS, 350, 243
McConnachie, A. W., Irwin, M. J., Ferguson, A. M. N., et al. 2005, MNRAS, 356, 979
Méndez, B., Davis, M., Moustakas, J., et al. 2002, AJ, 124, 213
Mould, J., \& Kristian, J. 1990, ApJ, 354, 438
Richardson, J. C., Irwin, M. J., McConnachie, A. W., et al. 2011, ApJ, 732, 76
Rizzi, L., Tully, R. B., Makarov, D., et al. 2007, ApJ, 661, 815
Sakai, S., Madore, B. F., \& Freedman, W. L. 1996, ApJ, 461, 713
Salaris, M., \& Cassisi, S. 1997, MNRAS, 289, 406
Schlegel, D. J., Finkbeiner, D. P., \& Davis, M. 1998, ApJ, 500, 525
van den Bergh, S. 1971, ApJ, 171, L31

# Paper II: A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. II. 

Distances to the Satellites of M31

## Paper II Preface

The first tip of the red giant branch paper (Paper I) was written with the intention that a second paper would soon follow which would further develop the method and apply it to the entire satellite sample of M31 ${ }^{1}$. As it came to pass, Paper II would not be accepted as an Astrophysical Journal publication until one year and three days after the acceptance of the first paper, despite being begun well before the first paper was accepted. This paper therefore represents a significant portion of my PhD candidature.

The method employed to gain the satellite distance distributions presented in Paper II, differs from that introduced in Paper I, chiefly in the way that prior information is taken into account. Most notably, 'matched filtering' is introduced to weight stars in accordance with their likelihood of being true object members. The object's density profile (as a function of radius) is treated as a probability distribution of object membership such that stars found in the densest central regions of the object are given more weight when fitting the object's luminosity function. In many cases, the contrast between the luminosity function with and without the matched filtering switched on is profound, with the RGB tip becoming clearly visible to the eye where before it was lost in a mass of masquerading background stars. In addition to the matched filtering, a prior is also imposed on the expected object distance in the form of a halo density prior. A cross-section through the (expected) M31 halo density profile along the line of sight to the object is used to weight the probability of finding the object at any distance along its distance probability distribution.

The reason for the rather lengthy time interval between the publication of the two papers was not due to any major issues with the method in this new paper, but rather the amount of feedback I received from those interested in the satellite distances. It became clear very early on that a lot of people had a vested interest in having access to accurate distances accompanied by accurate uncertainty distributions in those distances. It was also clear that many held clear-cut views as to how the distances should be obtained and presented. As a result I had to incorporate a particularly large amount of changes into the method and in turn

[^3]the draft of the paper which inevitably meant a large number of complete re-runs on all of the data and analysis it contained. After making the necessary changes to the method and re-writing various parts of the paper, it was finally ready for submission to Astrophysical Journal. All of this said, there is no doubt that the method is more robust as a result of this lengthy process.

One of the most important changes that arose from this scrutiny concerned the way the density profiles of the target objects were generated. Originally, the density profiles were being produced simply by drawing a series of evenly spaced concentric circles (or bands) about the object center and determining the density of stars in each band. The resulting binned profile was then fit in log space by a straight line (i.e. approximating the profiles as exponential). This of course assumes spherical symmetry which is not always a fair assumption, with some of M31's satellites being strongly elliptical. It was therefore decided to take this ellipticity fully into account which required a substantial re-write of the code for the density matched filter. These changes also warranted a second look at the luminosity function of each object and extra care was taken to insure that the CMD colour-cuts and the inner and outer cutoff radii for each object combined to produce luminosity functions with the greatest tip contrast possible. Other shortcomings in the algorithm code (see 'MF_TRGB.f95' in Appendix D) were also subsequently identified as the need arose for faster processing times and so provisions were made for feeding in the necessary object parameters in the command line and other portions of the code were altered to run more efficiently. Improvements to the PAndAS photometry calibration at the beginning of 2012 also required another re-run on the M31 satellites which further improved the quality of the distance measurements.

In many respects, the real climax of Paper II is the application of the distances to produce a new 3D view of the M31 system, as is presented in Fig. 10. This represents the true beginning of our study of the three dimensional structure of the satellite system, at the heart of which is the trigonometry necessary to convert the earth distances into an M31-centric coordinate system. Fig. 4.1 was created to aid in the determination of the necessary conversions. The coordinate system used here is that which arises most naturally from an Earth based perspective, with $z$ pointing along the line of sight to the center of M31, and $x$ and y pointing along lines of constant Declination and Right Ascension respectively. A more
typical orientation of the coordinate system is later adapted in Paper III by implementing rotations about the x and z axes so that z points toward the M 31 galactic north pole, with the Earth at a longitude of $0^{\circ}$.

Now, with M31 and its satellites represented by a series of points in three dimensions, we are in a position to begin an analysis of the distribution. This analysis is begun in Paper II with a study of the satellite density profile within the M31 halo. Of particular note, this study takes into full account the uneven coverage of the PAndAS survey, whereby certain radii from the center of M31 receive better coverage than others. The study also gives full account to the distance uncertainty distributions for each satellite by sampling possible positions from each distribution over many iterations. A more thorough study of the satellite distribution then follows in Paper III. Note that all of the principal code used throughout the analysis in Paper II can be found in Appendix D, along with a brief summary of what each program does.
(The published version of Conn et al. 2012 is reproduced by permission of the AAS)


Figure 4.1: Conversion of Earth-to-object distances into an M31-centric cartesian coordinate system. This figure was created to help visualize the geometry of Earth-M31-object alignments. The top part of the diagram shows the projection of the target object (satellite) on to the M31 tangent plane and the $\mathrm{x}, \mathrm{y}, \mathrm{z}$ of the coordinate system used. Positive $x$ points East (toward increasing $\xi$ ), positive $y$ points North (toward increasing $\eta$ ) and positive $z$ points along the line of sight (to M31) away from Earth. The three triangles in the lower half of the figure show how each coordinate can be determined from the Earth-to-M31 (a) and Earth-to-object (b) distances.

# A BAYESIAN APPROACH TO LOCATING THE RED GIANT BRANCH TIP MAGNITUDE. II. DISTANCES TO THE SATELLITES OF M31 

A. R. Conn ${ }^{1,2,3}$, R. A. Ibata $^{3}$, G. F. Lewis ${ }^{4}$, Q. A. Parker ${ }^{1,2,5}$, D. B. Zucker ${ }^{1,2,5}$, N. F. Martin ${ }^{3}$, A. W. McConnachie ${ }^{6}$, M. J. Irwin ${ }^{7}$, N. Tanvir ${ }^{8}$, M. A. Fardal ${ }^{9}$, A. M. N. Ferguson ${ }^{10}$, S. C. Chapman ${ }^{7}$, and D. Valls-Gabaud ${ }^{11}$<br>${ }^{1}$ Department of Physics \& Astronomy, Macquarie University, NSW 2109, Australia<br>${ }^{2}$ Research Centre in Astronomy, Astrophysics, and Astrophotonics (MQAASTRO), Macquarie University, NSW 2109, Australia<br>${ }^{3}$ Observatoire Astronomique, Universite de Strasbourg, CNRS, F-67000 Strasbourg, France<br>${ }^{4}$ Sydney Institute for Astronomy, School of Physics, A28, University of Sydney, Sydney, NSW 2006, Australia<br>${ }^{5}$ Australian Astronomical Observatory, P.O. Box 296, Epping, NSW 2121, Australia<br>${ }^{6}$ NRC Herzberg Institute of Astrophysics, 5071 West Saanich Road, Victoria, British Columbia V9E 2E7, Canada<br>${ }^{7}$ Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK<br>${ }^{8}$ Department of Physics and Astronomy, University of Leicester, Leicester LE1 7RH, UK<br>${ }^{9}$ University of Massachusetts, Department of Astronomy, LGRT 619-E, 710 N. Pleasant Street, Amherst, MA 01003-9305, USA<br>${ }^{10}$ Institute for Astronomy, University of Edinburgh, Royal Observatory, Blackford Hill, Edinburgh EH9 3HJ, UK<br>${ }^{11}$ Observatoire de Paris, LERMA, 61 Avenue de l'Observatoire, F-75014 Paris, France Received 2012 March 1; accepted 2012 July 18; published 2012 September 20


#### Abstract

In "A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude (Part I)," a new technique was introduced for obtaining distances using the tip of the red giant branch (TRGB) standard candle. Here we describe a useful complement to the technique with the potential to further reduce the uncertainty in our distance measurements by incorporating a matched-filter weighting scheme into the model likelihood calculations. In this scheme, stars are weighted according to their probability of being true object members. We then re-test our modified algorithm using random-realization artificial data to verify the validity of the generated posterior probability distributions (PPDs) and proceed to apply the algorithm to the satellite system of M31, culminating in a three-dimensional view of the system. Further to the distributions thus obtained, we apply a satellite-specific prior on the satellite distances to weight the resulting distance posterior distributions, based on the halo density profile. Thus in a single publication, using a single method, a comprehensive coverage of the distances to the companion galaxies of M31 is presented, encompassing the dwarf spheroidals Andromedas I-III, V, IX-XXVII, and XXX along with NGC 147, NGC 185, M33, and M31 itself. Of these, the distances to Andromedas XXIV-XXVII and Andromeda XXX have never before been derived using the TRGB. Object distances are determined from high-resolution tip magnitude posterior distributions generated using the Markov Chain Monte Carlo technique and associated sampling of these distributions to take into account uncertainties in foreground extinction and the absolute magnitude of the TRGB as well as photometric errors. The distance PPDs obtained for each object both with and without the aforementioned prior are made available to the reader in tabular form. The large object coverage takes advantage of the unprecedented size and photometric depth of the Pan-Andromeda Archaeological Survey. Finally, a preliminary investigation into the satellite density distribution within the halo is made using the obtained distance distributions. For simplicity, this investigation assumes a single power law for the density as a function of radius, with the slope of this power law examined for several subsets of the entire satellite sample.


Key words: galaxies: general - galaxies: stellar content - Local Group
Online-only material: color figures, machine-readable table

## 1. INTRODUCTION

The tip of the red giant branch (TRGB) is a well-established standard candle for ascertaining distances to extended, metalpoor structures containing a sufficient red giant population. Its near constant luminosity across applicable stellar mass and metallicity ranges (see Iben \& Renzini 1983) arises due to the prevailing core conditions of these medium-mass stars as core helium fusion ensues. Their cores lack the necessary pressure to ignite immediate helium fusion on the depletion of their hydrogen fuel and so they continue to fuse hydrogen in a shell around an inert, helium ash core. This core is supported by electron degeneracy and grows in mass as more helium ash is deposited by the surrounding layer of hydrogen fusion. On reaching a critical mass, core helium fusion ignites, and the star undergoes the helium flash before fading from its position at the TRGB, to begin life as a horizontal branch star. Due to the very similar core properties of the stars at this point, their energy output is almost independent of their total mass, resulting
in a distinct edge to the RGB in the color-magnitude diagram (CMD) of any significant red giant population.
With the TRGB standard candle applicable wherever there is an RGB population, it is an obvious choice for obtaining distances to the more sparsely populated objects in the Local Group and other nearby groups where Cepheid variables seldom reside. Even when Cepheids are available, the TRGB often remains a more desirable alternative, requiring only one epoch of observation, and facilitating multiple distance measurements across an extended structure. Good agreement between TRGBobtained distances and those obtained using Cepheid variables as well as the much fainter RR Lyrae variables have been confirmed by Salaris \& Cassisi (1997), with discrepancies of no more than $\sim 5 \%$ (see also Tammann et al. 2008 for an extensive list of distance comparisons utilizing the three standard candles). Of the satellites of M31, many are very faint and poorly populated and thus have poorly constrained distances which propagate on into related measurements concerning the structure of the halo system. Hence, a technique for refining
the distances that can be applied universally to all halo objects, while accurately conveying the associated distance errors has been a long sought goal.

In "A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude (Part I)"-Conn et al. (2011), hereafter Paper I, we reviewed the challenges of identifying the TRGB given the contamination to the pure RGB luminosity function (LF) typically encountered. We also outlined some of the methods that have been devised to meet these challenges since the earliest approach, put forward by Lee et al. (1993). We then introduced our own unique Bayesian approach, incorporating Markov Chain Monte Carlo (MCMC) fitting of the LFs. This approach was essentially the base algorithm, designed to easily incorporate priors to suit the task at hand. Here we present the results of an adaptation of that algorithm, intended for use on small, compact objects-specifically the dwarf spheroidal companions of M31. Once again, we utilize the data of the PanAndromeda Archaeological Survey (PAndAS; McConnachie et al. 2009), a two-color ( $i^{\prime}=770 \mathrm{~nm}, g^{\prime}=487 \mathrm{~nm}$ ) panoramic survey of the entire region around M31 and M33 undertaken using the Canada-France-Hawaii Telescope (CFHT). The tip is measured in the $i^{\prime}$ band where dependence on metallicity is minimal. Following a recap of the base method in Section 2, we introduce the aforementioned new adaptations to the method in Section 3.1 and in Section 3.2 we describe the results of tests intended to characterize the modified algorithms performance as well as check the accuracy of its outputs. In addition, Section 3.3 outlines the application of a further prior on the satellite distances. Section 4.1 presents the results of applying the modified algorithm to the companions of M31, while Section 4.2 details the method by which the object-to-M31 distances are obtained and Section 4.3 uses the obtained distances to analyze the density profile of these objects within the halo. Conclusions follow in Section 5.

## 2. A RECAP OF THE BASE METHOD

In Paper I, we introduced our "base" method, whereby the LF of a target field was modeled by a single, truncated power law (the RGB of the object of interest) added to a representative background polynomial. The location of the truncation (the TRGB) and the slope of the power law were set as free parameters of the model, with the best fit derived using an MCMC algorithm. The functional form of the background component was modeled by directly fitting a polynomial to the LF of an appropriate background field, and then scaling the polynomial to reflect the expected number of background stars in the target field. The resulting model was then convolved with a Gaussian of width increasing in proportion to the photometric error as a function of magnitude. The posterior distribution in the tip magnitude returned by the MCMC, which thus already incorporates the photometric error, is then sampled together with Gaussian distributions representing the distribution in the absolute magnitude of the tip $\left(M_{i}^{\text {TRGB }}=-3.44 \pm 0.05\right)$ and the distribution in the extinction $\left(A_{\lambda} \pm 0.1 A_{\lambda}\right)$ to give a final posterior distribution in the distance. The mode of this distribution is then adopted as the distance to the object, with the $\pm 1 \sigma$ error calculated from the portion of the distribution lying on the far and near side of the mode, respectively.

A more detailed discussion of the assumptions and rationale behind the base method is provided in paper I, but the reader should again be made aware of the most fundamental assumptions it entails. At the heart of the calculations of course is the choice of the absolute magnitude of the tip and its
associated uncertainty. We adopt the values of this parameter stated above based on the value derived for the SDSS $i$ band in Bellazzini (2008), noting the near-identical bandpass characteristics of the MegaCam i-band filter as detailed by Gwyn (2010). We adopt somewhat smaller uncertainties than those derived by Bellazzini (2008) following the same argument as McConnachie et al. (2004) that the quoted uncertainty in the absolute magnitude of the tip is conservative and it is a systematic error effecting all distance measurements in an identical way. As almost all applications of the distances to the satellites are concerned with their relative positions to one another and M31, this component of the error is of minimal importance. Nevertheless, it often forms the major component of the quoted errors in our distances.
Mention should also be made as to the effects of metallicity and internal reddening within the objects under study as well as the zero-point uncertainty in the PAndAS photometry. While there is a metallicity dependence of $M_{i}^{\text {TRGB }}$ (though minimal when compared with other bands), it is only really an issue for more metal-rich targets (e.g., $[\mathrm{Fe} / \mathrm{H}]>-1$; see Bellazzini 2008, Figure 6) and thus will primarily affect measurements to the large, diverse systems such as M31 itself and M33. But the TRGB for more metal-rich populations is fainter than that for their metal-poor counterparts and thus it is this metal-poor population component which dominates the measurement. A similar situation is encountered with the internal reddening present in the objects under study, where the vast majority of objects, chiefly the dwarf spheroidal galaxies, are almost completely devoid of such effects. Those objects most strongly affected are the large, well-populated systems which will provide ample signal from the least affected stars on the near side of the system, for a good distance determination. The uncertainty in the zero point of the photometry is consistent throughout the survey at approximately 0.02 mag. (R. A. Ibata et al. 2012, in preparation).
Lastly, a brief discussion of the distance posterior distributions themselves is warranted. As noted above, they are produced by the sampling of the distribution of possible tip positions (as generated by the MCMC and with photometric errors incorporated) along with sampling of the Gaussian distributions representing the uncertainties in the foreground extinction $\left(A_{\lambda}\right)$ and in the absolute magnitude of the tip ( $M_{i}^{\mathrm{TRGB}}$ ). Specifically, 500,000 possible distances are drawn to form the distance PPD, where for each draw $\kappa$, the distance modulus $\mu$ is

$$
\begin{equation*}
\mu(\kappa)=m_{i}^{\mathrm{TRGB}}(\kappa)-A_{\lambda}(\kappa)-M_{i}^{\mathrm{TRGB}}(\kappa), \tag{1}
\end{equation*}
$$

where each of $m_{i}^{\mathrm{TRGB}}(\kappa), A_{\lambda}(\kappa)$, and $M_{i}^{\mathrm{TRGB}}(\kappa)$ is the values drawn from the uncertainty distributions in the tip position, foreground extinction, and absolute magnitude of the tip, respectively. The foreground extinction and its uncertainty vary from object to object but the error in the absolute magnitude of the tip is a systematic error as already discussed. In using this method, there are two situations that can be encountered. The first is that the object is very well populated and the tip position is thus well constrained with a narrow PPD. In such instances, the uncertainty in $M_{i}^{\mathrm{TRGB}}$ far outweighs any other contributions to the error budget and is almost solely responsible for the width of the distance PPD. In the second situation, the object is poorly populated and the tip position PPD is very wide and typically asymmetric. If the LF population is not extremely low, the uncertainty in $M_{i}^{\text {TRGB }}$ will contribute noticeably to the distance PPD, otherwise the distance PPD will essentially depend solely on the uncertainty in the determined tip positions. Hence while
some of the smaller contributions to the distance uncertainties are omitted from the calculations, their overall effects will be washed out by the contributions from these two principal sources of error.

## 3. ADDITION OF A MATCHED FILTER

### 3.1. Matched Filtering using Radial Density Profiles

With the introduction of our method in Paper I, it was stressed that one of its greatest attributes was its adaptability to the prior knowledge available for the object of interest. When applying the method to compact satellites, there is one very conspicuous attribute that can be incorporated into the prior information constraining the model fit-namely, the object's density as a function of radius. The simplest way to achieve this is with the addition to the algorithm of a matched-filter weighting scheme, wherein the weighting is matched to the specific data by accounting for the data within the filter itself.

The successes of Rockosi et al. (2002) using a matched filter in color-magnitude space to identify member stars of globular cluster Palomar 5 amidst the stellar background provide the inspiration for our technique. They make use of the characteristic RGB of the globular cluster to weight stars as to their likelihood of being cluster members. To achieve such a goal, a matched filter can be created by binning the CMD of the field in which the cluster lies into a two-dimensional matrix and then dividing that matrix by a similarly created background matrix. Stars found in the densest regions of the resulting matched filter CMD are then assigned the highest weight, being the most likely cluster members. In this way, they can greatly improve the signal-to-noise ratio ( $\mathrm{S} / \mathrm{N}$ ) with respect to that of their original, unmodified data and are able to trace tidal streams from the globular cluster well into the surrounding background. Hence we have applied a similar approach to weight field stars fed to the MCMC in terms of their probability of being object members. In our case, however, the stars proximity to the object's center provides the basis for the weighting scheme, with the innermost stars being the most likely to be actual object members as opposed to background stars, and so a one-dimensional matched filter is sufficient.
The first step in implementing our weighting scheme is to ascertain a model of stellar density as a function of radius specific to the object of interest. For this purpose, we employ the best fits presented in N. F. Martin et al. (2012a, in preparation) for the dwarf spheroidal satellites, wherein the optimal ellipticity $\epsilon$, position angle (P.A.), half-light radii ( $r_{h}$ ), and object centers are given for exponential density profiles fitted to each satellite. For the two dwarf ellipticals, in the case of NGC 147 we assume $\epsilon=0.44$ and P.A. $=28^{\circ}$ as specified by Geha et al. (2010) and we derive the $r_{h}$ manually as $10^{\prime}$, which produces the best-fit profile to the data when coupled with the other two parameters. For NGC 185, we adopt $\epsilon=0.26$ and P.A. $=41^{\circ}$ based on the findings of Hodge (1963) and once again derive the $r_{h}$ manually, this time as $6^{\prime}$. For both NGC 147 and 185, we employ the object centers derived from the Two Micron All Sky Survey (2MASS; Skrutskie et al. 2006). With the ellipticity, P.A., half-light radius and object center know, we can proceed to produce a weighting scheme proportional to the density profile $\rho$ of the object, where $\rho$ is of the form

$$
\begin{equation*}
\rho\left(r_{\epsilon}\right)=e^{\frac{-r_{\epsilon}}{R}}, \tag{2}
\end{equation*}
$$

where $R=\left(r_{h} / 1.678\right)$ is the scale radius and $r_{\epsilon}$ is the elliptical radius at which the star lies, as now defined. With the P.A. and
object center of the object known, a rotation of coordinates is used to define each star's position ( $x^{\prime}, y^{\prime}$ ) with respect to the center of the ellipse. The projected elliptical radius $r_{\epsilon}$ of the ellipse on which the star lies is then

$$
\begin{equation*}
r_{\epsilon}=\left(\left(y^{\prime}\right)^{2}+\left(\frac{x^{\prime}}{1-\epsilon}\right)^{2}\right)^{1 / 2} \tag{3}
\end{equation*}
$$

where the $y^{\prime}$ axis is assumed as the major axis of the ellipse.
While Equation (2) gives us the functional form of our weighting scheme, it is further necessary to define the absolute values of the weights given to each star, so as to scale them appropriately with respect to the background density $\rho_{\mathrm{bg}}$. This is achieved by insuring that the area under the function $\rho\left(r_{\epsilon}\right)$ between any imposed inner and outer radius limits is set equal to the number of signal stars in the observed region. Hence, our weighting scheme is ultimately defined by

$$
\begin{equation*}
W\left(r_{\epsilon}\right)=S e^{\frac{-r_{\epsilon}}{R}} \tag{4}
\end{equation*}
$$

with

$$
\begin{equation*}
S=\frac{\left(\rho_{\text {total }}-\rho_{\mathrm{bg}}\right) \times A}{2 \pi R(1-\epsilon)\left[\left(e^{\frac{- \text { rinner }_{R}}{R}}\right)\left(R+r_{\mathrm{inner}}\right)-\left(e^{\frac{r_{\text {outer }}}{R}}\right)\left(R+r_{\text {outer }}\right)\right]}, \tag{5}
\end{equation*}
$$

where $\rho_{\text {total }}$ is the density of stars in the observed region before subtraction of the background density and $A$ is the area of the observed region which is either an ellipse in the (usual) case that $r_{\text {inner }}=0$ or an elliptical annulus otherwise. $r_{\text {inner }}$ and $r_{\text {outer }}$ are the inner and outer cutoffs respectively of the range of $r_{\epsilon}$ values observed.

In Figure 1, the result of our fitting procedure as applied to the sparsely populated dwarf spheroidal Andromeda X is presented. In this case, stars out to $r_{\epsilon}=0.15$ are fitted, with no inner cutoff radius imposed. While most of the satellites are too poorly populated for blending to be an issue, in the case of several, the stellar density counts at the innermost radii drop off in spite of the predicted counts from the fitted density profile. This is a good indicator of blending or overcrowding in those radii which can hinder the accuracy of the photometry for the affected stars and so in such cases, these inner radii are omitted. This was the case with Andromeda III ( $r_{\text {inner }}=0.0175$ ), Andromeda V $\left(r_{\text {inner }}=0.011\right)$, and Andromeda XVI $\left(r_{\text {inner }}=0.005\right)$. For the dwarf ellipticals NGC 147 and NGC 185, it was found beneficial to avoid the inner regions altogether, with the presence of a wider range of metallicities in these regions degrading the contrast of the RGB tip. Similarly, an outer cutoff radius was chosen for these objects inside of $3 r_{h}$ to help sharpen the tip discontinuity, so that for NGC 147, $r_{\text {inner }}=0.28$ and $r_{\text {outer }}=0.33$ and for NGC 185, $r_{\text {inner }}=0.18$ and $r_{\text {outer }}=0.26$. M31 and M33 are treated similarly to the dwarf ellipticals but with still thinner annuli so that any weighting is unnecessary. They are discussed in more detail in Section 4.1.
With regard to the actual likelihood calculations used at each iteration of the MCMC, these are undertaken not by simple multiplication of the likelihood for each star by the respective weight, but by physically adjusting the relative proportions of the RGB and background components of the LF. Up until now, we have assumed a generic LF and calculated the likelihood contributions from each star from this single LF. But in reality, the outer regions of the field are more accurately represented by a shallow-signal/high-background LF while the innermost stars obey an LF which has almost no background component.


Figure 1. Radial density profile (proportional to object membership probability) for Andromeda X. The error bars represent the Poisson error in the density for each bin, with each bin representing an elliptical annulus at the stated radius. Hence the innermost annuli have the smallest areas and thus the largest error bars. Note that this binned density distribution is for comparison only and has no bearing on the fit. The background level is marked "BG."
(A color version of this figure is available in the online journal.)

Hence using the radial density profile obtained above, we can essentially build an individual LF for each star, tailored to suit its position within the object. In practice, this is achieved with almost no extra computational effort, as the background and signal can be normalized separately and only the signal component is changed by the MCMC at each iteration so that the background component need only be generated once. The two components are normalized to contain an area of unity and then the bin of each corresponding to the star's magnitude is scaled according to the ratios of the star's weight and the background level when its contribution to the model likelihood is calculated by the MCMC.

The result of the incorporation of this extra prior information is a marked improvement in the performance of the algorithm for the more sparsely populated targets. In such objects, the RGB component is typically overwhelmed by non-system stars, even with the most carefully chosen field size. This can greatly diminish the prospects of obtaining a well-constrained tip measurement. This is apparent from Figures 2 and 3 which show the LF and corresponding posterior distributions before and after the application of the matched filter to the dwarf spheroidal Andromeda X. With the matched filtering applied, the great majority of non-system stars are severely suppressed, revealing clearly the RGB component, which in turn provides much stronger constraints on the location of the tip, as evidenced by Figure 3. Herein lies an example of the power of the Bayesian approach, where a single prior can cast the available data in a completely different light.

### 3.2. A Test for the Refined Algorithm

In Section 2.3 of Paper I, the results of a series of tests were presented that characterized the performance of our original algorithm given a range of possible background density levels
and LF populations. Here we present the results of similar tests applied to our new, matched-filter-equipped algorithm, but with some important differences. Most fundamentally, the way our artificial test data are generated is quite different. As we are now concerned with the position of each star in the field, a distance from field center must be generated for each star. To do this, we have randomly assigned a radial distance to each star, but weighted by a circularly symmetric $(\epsilon=0)$ exponential density profile. Further to this, the magnitudes of our stars are now generated directly from our convolved LF, so that photometric error as a function of stellar magnitude is incorporated.
The other important change from the previous tests concerns the way in which the artificial LFs are populated. Whereas in the former tests all of the sampled stars were drawn from the model LF within the one magnitude range $20 \leqslant m_{\text {star }} \leqslant 21$, in the current tests the stars are drawn from within the much larger magnitude range actually utilized for our satellite measurements, namely $19.5 \leqslant m_{\text {star }} \leqslant 23.5$. Hence a 100 star LF in these tests for example corresponds to a much smaller sample of stars than in the tests described in Section 2.3 of Paper I. Aside from these critical differences, the current tests are undertaken and presented as per the previous publication, with measurements of the average sigma and tip offset given for each combination of background level $(f)$ versus number of stars (ndata) where $f=0.1,0.2, \ldots, 0.9$ and ndata $=$ $10,20,50,100,200,500,1000,2000,5000,10,000,20,000$. The results are presented in Figures 4 and 5, respectively.
Examination of the figures reveals the expected trend of increased $1 \sigma$ error and tip offset with increasing background height and decreasing LF population levels. Once again, there is very good agreement between the derived errors and the actual offsets obtained. Most importantly, it is clear by comparing these results with those of Paper I that the matched filtering has


Figure 2. Best-fit model to the luminosity function of Andromeda $X$, obtained with the addition of matched filtering. The top figure shows the best fit overlaid on the unmodified LF (i.e., histogram created without the weighting afforded by the matched filter). The bottom figure shows the same best-fit model after applying the weighting. A field radius of 0.15 was used to generate the LF histograms, wherein each star contributes between 0 and 1 "counts," depending on its proximity to the field center and the density profile of the object.
(A color version of this figure is available in the online journal.)
greatly diminished the effects of the background contamination, as exemplified by the much gentler increase in $1 \sigma$ errors and offsets with increasing background star proportion.

### 3.3. An Additional Prior

In addition to our density matched filter, a further prior may be devised so as to constrain our distance posterior probability
distributions (PPDs) in accordance with our knowledge of the M31 halo dwarf density profile. The expected falloff in density of subhalos within an M31-sized galaxy halo is not well constrained. The largest particle simulation of an M31sized dark matter halo to date, the Aquarius Project (Springel et al. 2008), favored the density of subhalos to fall off following an Einasto profile with $r_{-2}=200 \mathrm{kpc}$ and $\alpha=0.678$, and furthermore identified no significant dependence of the


Figure 3. Posterior distributions obtained for Andromeda $X$ before (top) and after (bottom) the application of the matched filter. For the "before" case, a circular field of radius $0.05\left(2.143 \times r_{h}\right)$ has been chosen, specifically to provide the most possible signal with the least possible background contamination. For the "after" case, the same LF as presented in Figure 2 is used.
(A color version of this figure is available in the online journal.)
relationship on subhalo mass. For the specific case of the satellites within the M31 halo, Richardson et al. (2011) found a relation of $\rho \propto r^{-\alpha}$ where $\alpha=1$ a better fit to the data, drawing largely from the PAndAS survey, although this does not take into account the slightly irregular distribution of the survey area. We adopt this more gentle density falloff with radius giving us a
more subtle prior on the satellite density distribution and note that $\alpha$ may be changed significantly without great effect on our measured distances.
So in effect, we assume a spherical halo centered on M31, such that $\rho(\mathrm{sat}) \propto r^{-1}$ and integrate along a path through the halo at an angle corresponding to the angular displacement on

Sigma (kpc) - One Sigma Error


Figure 4. Gray-scale map of the $1 \sigma$ error in tip magnitude obtained for different combinations of background height and number of sources. The actual value recorded for the error (in kpc ) is overlaid on each pixel in red. Each value is the average of twenty 50,000 iteration runs for the given background height/LF population combination.
(A color version of this figure is available in the online journal.)

Offset (kpc) - Radial Distance Offset


Figure 5. Gray-scale map of the offset of the measured tip value from the true tip value obtained for different combinations of background height and number of sources. The actual value recorded (in kpc ) is overlaid on each pixel in red. Each value is the average of twenty 50,000 iteration runs for the given background height/LF population combination.
(A color version of this figure is available in the online journal.)


Figure 6. Distance prior applied to Andromeda XIII (solid line; $\alpha=1$ ). The distribution gives the likelihood of the satellite existing at a particular distance, given an angular separation on the sky of 8.5 from M31 (the halo center) and assuming a distance of 779 kpc for M31. The distribution peaks where the line of sight traverses the innermost region of the halo, and flattens out at large distances due to the increasing volume of the halo subtended by the unit of solid angle observed. The same prior with $\alpha=2$ is shown as a dashed line for reference. While this value for alpha is in closer agreement with the results of Section 4.3, we deliberately adopt the less restrictive $\alpha=1$ prior, so as not to suppress the probability of satellites in the outer halo too greatly.
the sky of the satellite from M31. This yields an equation of the form

$$
\begin{equation*}
P(d) \propto \frac{d^{2}}{\left(\left(d^{2}+779^{2}-2 d \times 779 \times \cos (\theta)\right)^{\alpha}\right)^{1 / 2}} \tag{6}
\end{equation*}
$$

where $\alpha=1,779 \mathrm{kpc}$ is the distance to M31, and $P(d)$ is the relative probability of the satellite lying at distance $d$ (in kpc ) given an angular separation of $\theta$ degrees from M31. Note that this produces a peak where the line of sight most closely approaches M31, and that $P(d \gg 779)$ is approximately proportional to $d$. The equation is normalized between limits appropriate to the size of the halo.

We thus generate a separate prior for the probability as a function of distance for each satellite, tailored to its specific position with respect to M31. The effect of the prior is to suppress unlikely peaks in the multi-peaked posterior distributions obtained for certain satellites, while leaving the peak positions unaffected. As such, the prior has very little effect on singlepeaked distributions, whatever the angular position and distance of the satellite it represents. The distance prior applied to the Andromeda XIII distance PPD is shown in Figure 6 for illustration.

## 4. A NEW PERSPECTIVE ON THE COMPANIONS OF M31

### 4.1. Galaxy Distances

The PAndAS survey provides us with a unique opportunity to apply a single method to a homogeneous data sample encompassing the entire M31 halo out to 150 kpc . The data encompass many dwarf spheroidals, along with the dwarf
ellipticals NGC 147 and NGC 185, and of course the M31 disk itself with additional fields bridging the gap out to the companion spiral galaxy M33, some $15^{\circ}$ distant. Of these objects, the vast majority have metallicities $[\mathrm{Fe} / \mathrm{H}] \leqslant-1$, so that any variation in the absolute magnitude of the tip is slight. Indeed, Bellazzini (2008) suggests that for such metallicities, the variation in the region of the spectrum admitted by the CFHT $i^{\prime}$ filter is perhaps less than in Cousins' I. Perhaps of greatest concern are the cases of M31 and M33, which will contain substructure at a variety of metallicities. In this case, however, the more metal-rich portions will exhibit a fainter TRGB than those in the regime $[\mathrm{Fe} / \mathrm{H}] \leqslant-1$, such that the brightest RGB stars will fall within this regime.
In this section we present distance measurements to these many halo objects, culminating in Figure 10 below, a threedimensional map of the satellite distribution, and Table 2, which presents the satellite data pertinent to our distance measurements. Figures 11 and 12 below present the distance posterior distributions obtained for every object in this study. It has been common practice in the majority of TRGB measurements to quote simply the most likely distance and estimated $1 \sigma$ uncertainties, but this throws away much of the information, except in the rare case that the distance distribution is actually a perfect Gaussian. On account of this, as well as providing the actual distance PPDs themselves for visual reference, we also provide the same information in condensed tabular form, where the object distance is given at $1 \%$ increments of the PPD, both for the prior-inclusive cases (as in Figures 11 and 12) and for the case in which no prior is invoked on the halo density. Note that for M31, no halo density prior is applied and so this column is set to zero. A sample of this information, as provided for

Table 1
Tabulated Distance Posterior Distribution

| Tabulated Distance Posterior Distribution |  |  |
| :--- | :---: | :---: |
| Percentage | Distance <br> (kpc, no density prior) | Distance <br> $(\mathrm{kpc})$ |
| 1 | 684 | 687 |
| 2 | 688 | 692 |
| 3 | 691 | 695 |
| 4 | 693 | 697 |
| 5 | 695 | 699 |
| 6 | 697 | 701 |
| 7 | 698 | 702 |
| 8 | 699 | 703 |
| 9 | 700 | 704 |
| 10 | 701 | 705 |
| $:$ | $:$ | $:$ |
| $:$ | $:$ | $:$ |
| 100 | 820 | 820 |

Notes. Distance posterior probability distributions for Andromeda I given at $1 \%$ intervals for the case of no halo density prior (Column 2) and with the angle-specific prior outlined in Section 3.3 applied (Column 3).
(This table is available in its entirety in a machine-readable form in the online journal. A portion is shown here for guidance regarding its form and content.)

Andromeda I, is presented in Table 1. The reader may then sample from these distributions directly rather than use the single quoted best-fit value, thus taking into account the true uncertainties in the measurements.

Due to the large number of objects studied, it is not practical to discuss each in detail within this paper. For this reason, Andromeda I will be discussed in further detail below as a representative example, followed by two of the more problematic cases for completeness. First, however, we describe the exceptional cases of M31 itself and M33.

M31 and M33 due to their large extent on the sky and the variety of substructure in their disks require a slightly different approach to that used for the other objects in this study. As was the case for NGC 147 and NGC 185, it was necessary to define a thin elliptical annulus so as to limit as much as possible the amount of substructure from other radii contaminating the LF. For both M31 and M33 such a thin annulus was used that any weighting with respect to the elliptical radius of the stars was trivial and so no weighting was used. For M31, an ellipticity of 0.68 was adopted, with P.A, $=37^{\circ}$. The inner and outer elliptical cutoff radii were set to 2.45 and 2.5 , respectively. To check for any inconsistencies in the TRGB location across the whole annulus, it was divided up into NE, NW, SE, and SW quarters and then the distance measured from each quarter, giving distances of $782_{-19}^{+19}, 782_{-18}^{+18}, 775_{-18}^{+20}$, and $781_{-19}^{+19} \mathrm{kpc}$, respectively. It is tempting to associate the slightly lower distance to the SE quadrant with the effects on the LF of the Giant Stellar Stream, though the distance is still within close agreement with the other three quadrants, such that all four are perfectly consistent. Hence, the distance was remeasured using the whole annulus to give $779_{-18}^{+19} \mathrm{kpc}$. This is in good agreement both with the findings of McConnachie et al. (2005) ( $785_{-25}^{+25}$ ) utilizing the TRGB and the more recent determination by Riess et al. (2012) using Cepheid variables $\left(765_{-28}^{+28}\right)$.

For M33, we employ an ellipticity of 0.4 as used by McConnachie et al. (2005), but find a position angle of P.A. $=$ $17^{\circ}$ in closest agreement with the data. Inner and outer elliptical radii of $r_{\text {inner }}=0.75$ and $r_{\text {outer }}=0.9$ were adopted to
give a very sharp discontinuity at the location of the tip. After applying an appropriate color-cut, the qualifying stars were fed into our algorithm to give a distance of $820_{-19}^{+20} \mathrm{kpc}$. This distance is in good agreement with that of $809_{-24}^{+24} \mathrm{kpc}$ obtained by McConnachie et al. (2005) and yields an M33-to-M31 distance of $214_{-5}^{+6} \mathrm{kpc}$. It is interesting to note that a variety of quite different M33 distances exist in the literature, with derived distance moduli ranging from 24.32 ( 730 kpc , water masers; Brunthaler et al. 2005) through 24.92 ( 964 kpc , detached eclipsing binaries; Bonanos et al. 2006). Indeed, the variety of standard candles utilized would suggest that M33 provides an ideal environment for calibrating the relative offsets between them. McConnachie (2005) suggests that the dispersion of M33 distances in the literature is tied to an inadequate understanding of the extinction in the region of M33. Most measurements, including those presented here, use the Galactic extinction values derived by Schlegel et al. (1998), although these do not account for extinction within M33 itself and are calculated via an interpolation of the extinction values for the surrounding region. Nevertheless, the elliptical annulus employed in our approach will act to smooth out the field-to-field variation that might exist between smaller regional fields.

### 4.1.1. Andromeda I: Example of an Ideal Luminosity Function

It would seem prudent to illustrate the performance of our new method by presenting the results for a range of the dwarf spheroidals from the most populated to the least populated. Hence Andromeda I, the first discovered and one of the two most highly populated of these objects, is the obvious place to start. The field employed for our Andromeda I distance measurement incorporated stars at elliptical radii between $0^{\circ} \leqslant r_{\epsilon} \leqslant 0.3$ and, after removal of stars outside of the range $19.5 \leqslant i_{0} \leqslant 23.5$ and beyond our chosen color-cut, yielded a star count of 4375 . The CMD for this field is presented in Figure 7(a). This figure colorcodes the stars in the CMD as per the color distribution in the inset field and plots them so that those innermost within the field (and hence those accorded the highest weight) are represented by the largest dots. In the case of Andromeda I, the RGB is so dominant over the background that our density matched filter is hardly necessary and hence does little to improve the already stark contrast. It is not surprising therefore that the distance and uncertainty obtained are almost identical to those obtained by the base method as presented in Paper I. Andromeda I is thus confirmed at a distance of $727_{-17}^{+18}$, which allows us to derive a similarly accurate separation distance from M31 of $68_{-16}^{+22} \mathrm{kpc}$.

### 4.1.2. Andromeda XV: Example of a Multi-peaked Distance PPD

As an example of a dwarf spheroidal of intermediate size, we present the comparatively compact Andromeda XV. Far from being the tidiest example of the many intermediatesized objects covered in this study, Andromeda XV provides something of a challenge. Examination of Figure 8 reveals a gradual rise in star counts when scanning from the top of the CMD color-cut faintward toward the Andromeda XV RGB and a correspondingly broad range in the possible tip locations in the tip magnitude PPD. Indeed, two peaks are prominent in the distance PPD of Figure 8(c), with the distribution mode at 626 kpc (our adopted distance) and the $1 \sigma$ credibility interval spanning from 591 kpc to 705 kpc as a consequence of the second peak. Ibata et al. (2007) determine this object to lie at a distance of $630_{-60}^{+60} \mathrm{kpc}$, which would correspond to a tip magnitude of approximately $m_{i}^{\text {TRGB }}=20.56$ assuming


Figure 7. Andromeda I: (a) color-coded CMD representing the weight given to each star in the field. Only stars within the red selection box with magnitudes $19.5 \leqslant i_{0} \leqslant 23.5$ were fitted and hence color-coded. The second, fainter RGB lying toward the redder end of the CMD is that of the giant stellar stream which passes behind our Andromeda I field. The inset at top right shows the field with the same color-coding and acts as a key. The field is divided into 20 radii bins following a linear decrease in density from the core (blue) to the field edge (purple). Stars marked as a purple " $\times$ " lie outside of the outer elliptical cutoff radius $r_{\text {outer }}$. Stars marked as a black " + " are artificial stars used in the estimation of the background density and are ignored by the MCMC; (b): posterior probability distribution for the TRGB magnitude. The distribution is color-coded, with red indicating tip magnitudes within $68.2 \%$ (Gaussian 1 -sigma) on either side of the distribution mode, green those within $90 \%$, and blue those within $99 \%$; (c) weighted LF of satellite with superimposed best-fit model in red. A star at the very center of the satellite contributes 1 count to the luminosity function while those further out are assigned some fraction of 1 count in proportion with the satellite's density profile. (A color version of this figure is available in the online journal.)
$M_{i}^{\text {TRGB }}=-3.44$. This is in excellent agreement with the $m_{i}^{\text {TRGB }}=20.57_{-0.14}^{+0.23}$ recovered by this study. Letarte et al. (2009) however derive a distance of $770_{-70}^{+70} \mathrm{kpc}$ which places it toward the far edge of our $99 \%$ credibility interval on the distance (see Figure 8(c)). This measurement was derived after three stars that had been found to lie close to the Andromeda XV RGB tip in the former investigation were identified as Galactic foreground stars, following measurements of their radial velocities obtained with the Deep Imaging Multi-Object Spectrograph on Keck II. Of these stars, however, none lies within $2^{\prime}$ from our object center by which point the maximum possible weighting has already dropped to below $10 \%$, meaning
that even the highest weighted of these three stars will have minimal effect on the likelihood calculation. This would then suggest that each of these three stars has magnitude consistent with belonging to the Andromeda XV RGB.

### 4.1.3. Andromeda XIII: Example of a very Poorly Populated Luminosity Function

Andromeda XIII is among the most sparsely populated objects targeted by the current study and it is important to realize that it is impossible to obtain distances to such objects with small uncertainties using the TRGB standard candle, unless of course one of the few member stars can be positively identified as


Figure 8. Andromeda XV: (a) same as Figure 7(a) but for Andromeda XV; (b) same as Figure 7(b) but for Andromeda XV; (c) sampled distance posterior probability distribution, obtained by calculating the distance 3 million times, each time randomly drawing on the tip magnitude, absolute magnitude of the tip, and extinction from their respective probability distributions. The distribution is color-coded, with red indicating possible distances within $68.2 \%$ (Gaussian $1 \sigma$ ) on either side of the distribution mode, green those within $90 \%$, and blue those within $99 \%$. Note that the large uncertainty in the absolute magnitude of the RGB tip is primarily responsible for the much smoother appearance of the distance PPD (c) compared with the tip PPD (b).
(A color version of this figure is available in the online journal.)
being right on the brink of core helium fusion. Nevertheless, though large uncertainties are inevitable, an accurate estimation of those uncertainties is still achievable, and this is the aspiration of the method here presented. Distances to Andromeda XI and XIII have been obtained with higher accuracy using RR Lyrae stars as a standard candle with photometry from the Hubble Space Telescope (Yang \& Sarajedini 2012). In the case of Andromeda XI, the tip magnitude identified by our method agrees well with the distance identified by that study, but in the case of Andromeda XIII, a brighter star in the central regions of the field causes some confusion. Indeed in such a sparsely populated field it is quite difficult to apply any effective density-based weighting scheme. Nevertheless, after sampling the tip magnitude PPD (Figure 9(b)), together with those for the absolute magnitude of the tip and the extinction in this
region of sky to obtain a sampled distance PPD, and multiplying that distribution with the angle-specific halo density prior as is standard for all our measurements, we are able to produce a distance PPD (Figure 9(c)) in good agreement with the findings of Yang \& Sarajedini (2012).

### 4.2. Determining the Distances from M31

Once a satellite's distance from Earth is determined, it is straightforward to determine the distance from M31 using the cosine rule:

$$
\begin{equation*}
r=\left(d^{2}+\left(d_{M 31}\right)^{2}-2 d d_{M 31} \cos (\theta)\right)^{1 / 2} \tag{7}
\end{equation*}
$$

where $r$ is the satellite's distance from M31, $d$ is the distance of the satellite from Earth, $d_{M 31}$ is the distance of M31 from Earth,


Figure 9. Andromeda XIII: All figures as per Figure 8, but for Andromeda XIII. The distance derived by Yang \& Sarajedini (2012) is plotted in (c) along with error bars for comparison.
(A color version of this figure is available in the online journal.)
and $\theta$ is the angle on the sky between M31 and the satellite. For convenience, we use a small angle approximation equating $\theta$ with its M31 tangent plane projection and note that any displacement of $r$ is insignificant due to the size of the $1 \sigma$ errors. If the uncertainty in distance to both M31 and the satellite takes on a Gaussian distribution, it is straightforward to determine the error in the satellite-M31 separation by adding the individual errors in quadrature. While it is reasonable to approximate the M31 distance uncertainty distribution as a Gaussian, the same cannot be said for each of the companion satellites. Hence once again it is more appropriate to sample values from the individual distance probability distributions. Thus, a histogram of $r$ values for the satellite is built up by sampling $d$ and $d_{M 31}$ from their respective distributions over many iterations. This brings to the fore an important consideration: there is an integrable singularity in the resulting distribution at the closest approach distance to M31 $\left(r_{c}=d_{M 31} \sin (\theta)\right)$ as shown below.

The probability distribution for the satellite-to-Earth distance $P(d)$ is related to that of the satellite-to-M31 distance $P(r)$ as follows:

$$
\begin{equation*}
P(r)=\frac{\delta d}{\delta r} P(d) \tag{8}
\end{equation*}
$$

From Equation (7), and further noting that the satellite-to-Earth distance corresponding to $r_{c}$ is $d_{c}=d_{M 31} \cos (\theta)$, we have

$$
\begin{equation*}
\frac{\delta d}{\delta r}=\frac{r}{d-d_{c}}, \tag{9}
\end{equation*}
$$

which allows us to derive

$$
\begin{equation*}
P(r)=\frac{r}{\left(r^{2}-r_{c}^{2}\right)^{1 / 2}} P(d), \tag{10}
\end{equation*}
$$

thus producing the singularity at $r=r_{c}$. In practice, after factoring in the Gaussian distribution in $d_{M 31}$, this results in a sharp
peak at the minimum possible satellite-to-M31 distance when dealing with the more asymmetric satellite-to-Earth distance probability distributions. Hence when considering the distribution of satellites as a function of distance from M31, one can either take the distances as determined directly from Equation (7) using solely the most likely distance from the satellite-to-Earth distance distributions or the whole distance probability distribution for a satellite can be allowed to influence the calculations, as accomplished via sampling. The final result can be quite different, depending on the choice.

### 4.3. A First Approximation of the Satellite Density Profile within the Halo

In the completed PAndAS survey, we have for the first time a comprehensive coverage of a galaxy halo, with a uniform photometric depth sufficient to identify even the comparatively faint satellite companions. In addition, in this paper we have provided distances to every one of these objects, all obtained via the same method. We are thus presented with an excellent opportunity to study the density of satellites as a function of radius within a Milky Way like halo.

As hinted at in the previous section, obtaining an accurate picture of the satellite density profile (SDP) is not a trivial task. The first major consideration is to devise a way of factoring in the selection function. Comprehensive though the survey coverage is, it is not symmetric and not infinite. Second, the choice of model for the SDP is not arbitrary. Whether a simple, unbroken power law is sufficient is not immediately clear. Furthermore, does it even make any sense to treat the halo as a radially symmetric, isotropic distribution? A glance at the obvious asymmetry in Figure 10(a) would suggest otherwise. Nevertheless, for a first approximation it is reasonable to consider what the best-fitting radially symmetric, unbroken power law to the SDP would be.

The PAndAS survey covers approximately $400 \mathrm{deg}^{2}$ of sky and is roughly symmetric about the center of the M31 disk but with a major protrusion in the southeast to encompass the M33 environs. For the purpose of obtaining an accurate measure of the survey coverage of the halo as a function of radius, as well as factoring in the actual survey borders, an inner ellipse was also subtracted where the presence of the M31 disk has made satellite detection more difficult. Both the outer survey borders and the inner cutoff ellipse are plotted in Figure 10. The inner cutoff ellipse has an eccentricity $\epsilon=\sqrt{0.84}$ and is inclined with the semi-major axis angled 51.9 with respect to the $x$-axis ( $\eta=0$ ). The dwarf galaxies M32 and M110 lie inside this ellipse as do the somewhat dubious satellite identifications Andromeda VIII and Andromeda IV (see Ferguson et al. 2000), hence their omission from the data presented in Table 2. With the inner and outer boundaries suitably delineated, the procedure then was to determine what fraction of halo volume at a given radius $f(r)$ would fall within these boundaries once projected onto the M31 tangent plane. This was achieved by implementing the even-odd rule on the projections of uniformly populated halo shells.

Having determined $f(r)$, we can proceed to determine the required normalization for a power law of any given $\alpha$, allowing us to use the power law directly as a probability distribution. Setting the problem out in terms of probabilities, we require to determine the probability of each tested M31-to-object distance (henceforth simply "radius") $r$ given a power law with slope $\alpha$ :

$$
\begin{equation*}
P(r \mid \alpha)=\frac{k}{r^{\alpha}}, \tag{11}
\end{equation*}
$$

where $k$ is the normalization constant and $r_{\text {min }} \leqslant r \leqslant r_{\text {max }}$. Using the assumed spherical symmetry, we then have

$$
\begin{equation*}
f(r) \int_{r_{\min }}^{r_{\max }} P(r \mid \alpha) \int_{0}^{2 \pi} \int_{0}^{\pi} r^{2} \sin \theta d \theta d \phi d r=1 \tag{12}
\end{equation*}
$$

so that

$$
\begin{equation*}
4 \pi f(r) \int_{r_{\min }}^{r_{\max }} k r^{2-\alpha} d r=1 \tag{11}
\end{equation*}
$$

Hence, for a given radius at a given $\alpha$, we have

$$
\begin{equation*}
k(r, \alpha)=\left[4 \pi f(r)\left(\frac{r^{3-\alpha}}{3-\alpha}\right)_{r_{\min }}^{r_{\max }}\right]^{-1} . \tag{14}
\end{equation*}
$$

The calculation of the likelihood for a power law of a given slope $\alpha$ may be simplified by noting that for any given radius, $f(r)$ and hence $k$ act to scale the probability in an identical way whatever the value of $\alpha$. Thus, the dependence of $k$ on $r$ is effectively marginalized over when the posterior distribution for $\alpha$ is calculated, so long as any sampling of radii utilizes the same radii at every value of $\alpha$. The likelihood for a given power law (i.e., a given $\alpha$ ) is thus

$$
\begin{equation*}
\mathcal{L}(\alpha)=\prod_{n=i}^{\mathrm{nsat}} k r_{i}^{2-\alpha}, \tag{15}
\end{equation*}
$$

where nsat is the number of satellites-i.e., the 27 companions of M31 listed in Table 2. As discussed in Section 4.2, there are essentially two ways we can determine the likelihood of a given $\alpha$. The most straightforward is to use single values of $r_{i}$ as determined directly from the mode in the posterior distribution for each satellite using Equation (7). The second and arguably more robust method is to use the entire radius probability distribution (RPD) for each satellite. In the case of this second approach, the likelihood for the power law determined for each satellite becomes a convolution of the power law with the satellite's RPD, so that the likelihoods of the individual samples are summed. The final likelihoods determined for each satellite can then be simply multiplied as before, giving a total likelihood as follows:

$$
\begin{align*}
\mathcal{L}(\alpha)= & \left(k r_{1,1}^{2-\alpha}+k r_{2,1}^{2-\alpha}+\cdots+k r_{\text {nsam, } 1}^{2-\alpha}\right) \\
& \times\left(k r_{1,2}^{2-\alpha}+k r_{2,2}^{2-\alpha}+\cdots+k r_{\text {nsam,2 }}^{2-\alpha}\right) \times \cdots \\
& \times\left(k r_{1, \text { nsat }}^{2-\alpha}+k r_{2, \text { nsat }}^{2-\alpha}+\cdots+k r_{\text {nsam,nsat }}^{2-\alpha}\right)  \tag{16}\\
= & \prod_{n=i}^{\text {nsat }}\left[\sum_{n=j}^{\text {nsam }} k r_{j, i}^{2-\alpha}\right]
\end{align*}
$$

where $r_{j, i}$ is the $j$ th sampled radius of the $i$ th satellite, and nsam is the total number of samples.
The resulting distribution achieved by implementing the first approach is presented in Figure 13(a) from which a value for $\alpha$ of $1.92_{-0.30}^{+0.32}$ is obtained. It is interesting to note that this value is consistent with an isothermal satellite distribution with uniform velocity dispersion. Replacing the individual best-fit radii with 500,000 samples from the respective RPD for each satellite as per the second approach, the result is substantially different, as demonstrated by Figure 13(b). Here a value for $\alpha$ of $1.52_{-0.32}^{+0.35}$ provides the best fit to the data. This discrepancy is presumably a consequence of the non-Gaussian RPD profiles for


Figure 10. Three views of the M31 neighborhood: (a) a view of the satellites of M31 along the $y$ - $z$ plane. The conic section illustrates the extent of volume covered by the PAndAS footprint as a function of distance from Earth; (b) a view of the satellites of M31 in the $x-y$ plane, revealing their true positions on the $x-y$ plane after removing the effects of perspective (assuming the distances quoted in Column 4 of Table 2). Note that Andromeda XXVII lies directly behind NGC 147 in this plot and is not labeled; (c) a three-dimensional view of the satellites of M31. The satellite positions on the PAndAS footprint are indicated (i.e., with perspective conserved) along with the $z$-vector giving distance from the M31-centered tangent plane. The central ellipse indicates the approximate area of the survey where satellite detection is hindered by the M31 disk; note that the perpendicular bars on relevant axes indicate 100 kpc intervals.
(A color version of this figure is available in the online journal.)
the more poorly populated satellites, as noted in Section 4.2. In fact, if the 15 most Gaussian-like distributions are taken alone, namely Andromedas I, II, III, V, X, XVI, XVII, XVIII, XX, XXI, XXIII, XXIV, NGC 147, NGC 185, and M33, the results are in much closer agreement, with $\alpha=1.87_{-0.42}^{+0.46}$ with sampling and $\alpha=2.02_{-0.41}^{+0.43}$ without.

Given the obvious asymmetry in the satellite distribution in Figure 10, it is interesting to consider the effects of isolating various other satellites from the calculations. The stark asymmetry between the number of satellites on the near side as opposed to the far side of the M31 tangent plane for instance (as had been initially reported by McConnachie \& Irwin 2006) is echoed in the respective density profiles, with an $\alpha$
of $2.37_{-0.37}^{+0.42}$ (no sampling) recorded when only the near-side satellites are considered, and that of $0.93_{-0.49}^{+0.56}$ (no sampling) when instead the far-side galaxies alone are included. When the individual satellite RPDs are sampled, the corresponding values are $1.87_{-0.40}^{+0.43}$ and $0.78_{-0.46}^{+0.61}$, respectively. Despite the large uncertainties, the results clearly do not support symmetry of any kind about the tangent plane. It is important to note, however, that this asymmetry may not be physical, but rather an effect of incompleteness in the data at the fainter magnitudes of the satellites on the far side of M31. McConnachie \& Irwin (2006) do however observe this asymmetry even when only the more luminous satellites are considered. In time, it is hoped that the nature of the data incompleteness will be better understood and

Table 2
M31 Satellite Parameters: Distance and Associated Parameters of M31 and its Companions

| Source | Distance Modulus | $E(B-V)$ | Distance (kpc) | M31 Distance (kpc) | Literature Distance Values (kpc) |
| :---: | :---: | :---: | :---: | :---: | :---: |
| M31 | $24.46_{-0.05}^{+0.05}$ | 0.062 | $779{ }_{-18}^{+19}$ | ... | $\begin{gathered} 785_{-25}^{+25} \text { TRGB; McConnachie et al. (2005) } \\ 784_{-17}^{+17} \text { RC; Stanek \& Garnavich (1998) } \\ 765_{-28}^{+28} \text { Ceph; Riess et al. (2012) } \\ \hline \end{gathered}$ |
| And I | $24.31_{-0.05}^{+0.05}$ | 0.054 | $727_{-17}^{+18}$ | $68_{-17}^{+21}$ | $731_{-17}^{+18}$ TRGB; Conn et al. (2011) $735_{-23}^{+23}$ TRGB; McConnachie et al. (2004) |
| And II | $24.00_{-0.05}^{+0.05}$ | 0.062 | $630_{-15}^{+15}$ | $195_{-17}^{+20}$ | $634_{-14}^{+15}$ TRGB; Conn et al. (2011) $645_{-19}^{+19}$ TRGB; McConnachie et al. (2004) |
| And III | $24.30_{-0.07}^{+0.05}$ | 0.057 | $723_{-24}^{+18}$ | $86_{-15}^{+25}$ | $749_{-24}^{+24}$ TRGB; McConnachie et al. (2005) |
| And V | $24.35_{-0.07}^{+0.06}$ | 0.125 | $742_{-22}^{+21}$ | $113_{-6}^{+9}$ | $774_{-28}^{+28}$ TRGB; McConnachie et al. (2005) |
| And IX | $23.89_{-0.08}^{+0.31}$ | 0.076 | $600_{-23}^{+91}$ | $182_{-66}^{+38}$ | $765_{-24}^{+24}$ TRGB; McConnachie et al. (2005) |
| And X | $24.13_{-0.13}^{+0.08}$ | 0.126 | $670_{-39}^{+24}$ | $130_{-17}^{+60}$ | $667-738$ TRGB; Zucker et al. (2007) |
| And XI | $24.41_{-0.32}^{+0.08}$ | 0.080 | $763_{-106}^{+29}$ | $102_{-1}^{+149}$ | $\begin{gathered} 740-955 \text { TRGB; Martin et al. (2006) } \\ 735_{-17}^{+17} \text { RR Ly; Yang \& Sarajedini (2012) } \end{gathered}$ |
| And XII | $24.84_{-0.34}^{+0.09}$ | 0.111 | $928{ }_{-136}^{+40}$ | $181_{-87}^{+19}$ | $825_{-159}^{+85}$ TRGB; (MCMC without MF) <br> 740 - 955 TRGB; Martin et al. (2006) |
| And XIII | $24.40_{-0.49}^{+0.33}$ | 0.082 | $760_{-154}^{+126}$ | $115_{-2}^{+207}$ | $890_{-361}^{+360}$ TRGB; (MCMC without MF) $740-955$ TRGB; Martin et al. (2006) $839_{-19}^{+20}$ RR Ly; Yang \& Sarajedini (2012) |
| And XIV | $24.50_{-0.56}^{+0.06}$ | 0.060 | $793+23$ | $161_{-3}^{+81}$ | $630-850$ TRGB; Majewski et al. (2007) |
| And XV | $23.98_{-0.12}^{+0.26}$ | 0.046 | $626_{-35}^{+79}$ | $1744_{-32}^{+46}$ | $630_{-60}^{+60}$ TRGB; Ibata et al. (2007) $770_{-70}^{+70}$ TRGB; Letarte et al. (2009) |
| And XVI | $23.39_{-0.14}^{+0.19}$ | 0.066 | $476{ }_{-29}^{+44}$ | $3199_{-27}^{+43}$ | $525_{-50}^{+50}$ TRGB; Ibata et al. (2007) $525_{-50}^{+50}$ TRGB; Letarte et al. (2009) |
| And XVII | $24.31_{-0.08}^{+0.11}$ | 0.075 | $727_{-25}^{+39}$ | $67_{-24}^{20}$ | $794_{-40}^{+40}$ TRGB; Irwin et al. (2008) |
| And XVIII | $25.42_{-0.08}^{+0.07}$ | 0.104 | $1214_{-43}^{+40}$ | $4577_{-47}^{+39}$ | $1355_{-88}^{+88}$ TRGB; McConnachie et al. (2008) |
| And XIX | $24.57_{-0.43}^{+0.08}$ | 0.062 | $821_{-148}^{+32}$ | $115_{-9}^{+96}$ | $933_{-61}^{+61}$ TRGB; McConnachie et al. (2008) |
| And XX | $24.35_{-0.16}^{+0.12}$ | 0.058 | $741_{-52}^{+42}$ | $128{ }_{-5}^{+28}$ | 802-96 ${ }_{-96}^{+297}$ TRGB; McConnachie et al. (2008) |
| And XXI | $24.59_{-0.07}^{+0.06}$ | 0.093 | $827_{-25}^{+23}$ | $135_{-10}^{+8}$ | $859_{-51}^{+51}$ TRGB; Martin et al. (2009) |
| And XXII (Tri I) | $24.82_{-0.36}^{+0.07}$ | 0.075 | $920_{-139}^{+32}$ | $2755_{-60}^{+8}$ | $794{ }_{-0}^{+239}$ TRGB; Martin et al. (2009) |
| And XXIII | $24.37_{-0.06}^{+0.09}$ | 0.066 | $748_{-21}^{+31}$ | $127_{-4}^{+7}$ | $733_{-22}^{+23}$ TRGB; Conn et al. (2011) $767_{-44}^{+44} \mathrm{HB}$; Richardson et al. (2011) |
| And XXIV | 24.77 ${ }_{-0.10}^{+0.07}$ | 0.083 | $898{ }_{-42}^{+28}$ | $169_{-29}^{+29}$ | $600_{-33}^{+33} \mathrm{HB}$; Richardson et al. (2011) |
| And XXV | $24.33_{-0.21}^{+0.07}$ | 0.101 | $736_{-69}^{+23}$ | $90_{-10}^{+57}$ | $812_{-46}^{+46} \mathrm{HB}$; Richardson et al. (2011) |
| And XXVI | $24.39_{-0.53}^{+0.55}$ | 0.110 | $754_{-164}^{+218}$ | $103_{-3}^{+234}$ | $762_{-42}^{+42} \mathrm{HB}$; Richardson et al. (2011) |
| And XXVII | $25.49_{-1.03}^{+0.07}$ | 0.080 | $1255_{-474}^{+42}$ | $482_{-425}^{+0}$ | $827_{-47}^{+47} \mathrm{HB}$; Richardson et al. (2011) |
| And XXX ${ }^{\text {a }}$ (Cass II) | $24.17_{-0.26}^{+0.10}$ | 0.166 | $681_{-78}^{+32}$ | $145_{-4}^{+95}$ | $565_{-25}^{+25}$ TRGB g-band; M. J. Irwin (2012, in preparation) |
| NGC 147 | $24.26_{-0.06}^{+0.06}$ | 0.173 | $712_{-19}^{+21}$ | $118_{-15}^{+15}$ | $675_{-27}^{+27}$ TRGB; McConnachie et al. (2005) |
| NGC 185 | $23.96_{-0.06}^{+0.07}$ | 0.182 | $620_{-18}^{+19}$ | $181_{-20}^{+25}$ | $616_{-26}^{+26}$ TRGB; McConnachie et al. (2005) |
| M33 | $24.57_{-0.05}^{+0.05}$ | 0.042 | $820_{-19}^{+20}$ | $210_{-5}^{+6}$ | $809_{-24}^{+24}$ TRGB; McConnachie et al. (2005) $964_{-54}^{+54}$ DEB; Bonanos et al. (2006) |

Notes. All distance measurements utilize the data from the Pan-Andromeda Archaeological Survey (McConnachie et al. 2009) and have been obtained using the method presented in this paper. A value of $M_{i}^{\text {TRGB }}=-3.44 \pm 0.05$ is assumed for the absolute magnitude of the RGB tip in CFHT MegaCam $i$ band, based on the value identified for the SDSS $i$ band (Bellazzini 2008) and justified for use here by the color equations applicable to the new MegaCam $i$-band filter (Gwyn 2010). Values for the extinction in MegaCam $i$ band have been adopted as $A_{\lambda}=2.086 \times E(B-V)$ for the same reasons, with uncertainties taken as $\pm 10 \%$. The extinction values quoted are for the object centers, though the actual calculations apply individual corrections to each member star according to their coordinates. Note that the uncertainties in the M31 distance are based on the sampled distributions while the quoted value is that derived directly from the Earth distance as per Equation (7). The last column gives alternative distances from the literature. TRGB-derived distances are quoted wherever possible. Distance derivation methods: TRGB, tip of the red giant branch; Ceph, Cepheid period-luminosity relation; RR Ly, RR Lyrae period-luminosity relation; RC, red clump; HB, horizontal branch; DEB, detached eclipsing binary.
${ }^{\text {a }}$ Andromeda XXX is a new discovery, and will also be known as Cassiopeia II, being the second dwarf spheroidal satellite of M31 to be discovered in the constellation of Cassiopeia (M. J. Irwin 2012, in preparation).


Figure 11. Distance posterior distributions for dwarf spheroidal satellites And I-III, And V and And IX-XIX. The distributions are color-coded with red, green, and blue denoting $1 \sigma(68.2 \%), 90 \%$, and $99 \%$ credibility intervals, respectively. The credibility intervals are measured from either side of the highest peak.
(A color version of this figure is available in the online journal.)


Figure 12. Distance posterior distributions for dwarf spheroidal satellites And XX-XXVII and And XXX, dwarf elliptical satellites NGC 147 and NGC 185, and major galaxies M31 and M33. The distributions are color-coded with red, green, and blue denoting $1 \sigma(68.2 \%), 90 \%$, and $99 \%$ credibility intervals, respectively. (A color version of this figure is available in the online journal.)


Figure 13. Probability distributions for the slope $\alpha$ of a single power law used to model the M31 halo satellite distribution, given the entire set of 27 M31 companions presented in Table 2. Panel (a) gives the distribution assuming a single best-fit radius for each of the satellites as determined from the mode in the satellite's distance posterior distribution (as given in Column 4 of Table 2). Panel (b) shows the same distribution when the entire radius probability distribution for each satellite is sampled 500,000 times.
effort is underway to determine the completeness functions for dwarf galaxy detection in the PAndAS survey (N. F. Martin et al. 2012b, in preparation). In the mean time, it would seem prudent to regard the contribution to the density profile of the far-side satellites with caution, instead taking the density profile measured from the near-side satellites alone as the best measurement.

On a final note with regard to near-side-far-side asymmetry, it is important to realize that the uncertainty in the distance to M31 has a large effect on how many satellites will lie on either side of the M31 tangent plane, and indeed on the density measurement as a whole. Where the individual PPDs are sampled, this is taken into account as the M31 PPD is sampled for each measurement. Nevertheless, it is interesting to consider the specific (nonsampled) case where M31 is measured at a closer distance, while all best-fit satellite distances remain unchanged. From the M31 PPD in Figure 12, it can be seen that there is a 5\%
chance that M31 lies at 750 kpc or closer. If M31 is taken to lie at 750 kpc , Andromedas XI, XIII, and XIV move onto the far side of the M31 tangent plane, going someway to even out the asymmetry. However, if the distances of all the satellites from M31 are re-measured for this new M31 position, the same stark contrast between the density profiles for the near and far sides remains and in fact grows. Using only those satellites on the near side of the new M31 tangent plane, an $\alpha$ of $2.87_{-0.45}^{+0.50}$ is determined whereas if only those satellites on the far side are considered, an $\alpha$ of $1.22_{-0.47}^{+0.47}$ is obtained. Hence it would seem unlikely that the observed near-side-far-side asymmetry is primarily a consequence of an overestimated M31 distance.
Recent research, such as that presented by Koch \& Grebel (2006) and Metz et al. (2007) point toward highly significant planar alignments of various collections of satellites within the M31 halo, even though as a whole, no such distribution is prominent. Interestingly, the former investigation finds that it is
predominantly the objects morphologically similar to the dwarf spheroidals in their sample that can be constrained to a relatively thin disk, which also includes NGC 147 and M33. While our sample is considerably larger, it nevertheless consists nearly entirely of such objects, so it will be interesting to determine what degree of symmetry may be found within and on either side of the best-fit plane. We intend to investigate this in an upcoming publication, though it must still be noted that outliers from the planar trend have already been noted in this small sample, such as Andromeda II and NGC 185. Furthermore, other members are known not to conform to the norm of M31 satellite dynamics, with Andromeda XIV for instance apparently at the escape velocity for the M31 system for its determined distance (Majewski et al. 2007). Indeed, it would seem that whatever model is assumed, a few outliers are inevitable.

## 5. CONCLUSIONS

With the ready applicability of the TRGB standard candle to almost any of our galactic neighbors, there can be no question that its role will continue to be an important one. As the world's premier telescopes grow in size, so too will the radius of the "neighborhood" of galaxies to which the TRGB can be applied. Hence a technique which accurately characterizes the true probability space of the TRGB distances determined is a great asset. Indeed this quality comes to play an increasingly important role as more and more sparsely populated objects are found to frequent the environs of our larger nearby neighbors. The differences in the results achieved in the previous section with and without sampling of the actual distance distributions illustrate this fact.

Where in Paper I the foundations were laid for a TRGB method with such desirable qualities, its full value only becomes apparent when one actually employs its full Bayesian potential. It only requires a brief glance at Figures 2 and 3 to see how powerful a single data-specific prior can be. Similarly, the simple distance weighting prior outlined in Section 3.3 can make a poorly constrained model quite workable, as illustrated in the case of Andromeda XIII. Both tools will likely prove very useful when the method is used further afield.

It should also be remembered that the TRGB standard candle is in many ways, just the "first assault." When photometric data of sufficient depth are obtained, the horizontal branch can often pin down the distance with still greater accuracy. With a simple adjustment to the model LF, the techniques outlined in this paper and its predecessor commute quite readily to implementation on the horizontal branch.

Last, it must also be said that the distances presented herein provide an excellent opportunity to provide a new, updated analysis of the asymmetry and density of the M31 halo satellite distribution, one only touched on here. With such comprehensive and consistent coverage, there is great potential in these distances to further constrain the possible evolution and dynamical history of the M31 halo system.
A.R.C. would like to thank Macquarie University for their financial support through the Macquarie University Research Excellence Scholarship (MQRES) and both the University of Sydney and Université de Strasbourg for the use of computational and other facilities. R.A.I. and D.V.G. gratefully acknowledge support from the Agence Nationale de la Recherche though the grant POMMME (ANR 09-BLAN-0228). G.F.L. thanks the Australian Research Council for support through his Future Fellowship (FT100100268) and Discovery Project (DP1 10100678). M.A.F. acknowledges support from NSF grant AST-1009652. The analysis contained in this publication uses photometric data obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-FranceHawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institute National des Sciences de l'Univers of the Centre National de la Recherche Scientifique of France, and the University of Hawaii. Our thanks go to the entire staff at CFHT for their great efforts and continuing support throughout the PAndAS project.

## REFERENCES

Bellazzini, M. 2008, Mem. Soc. Astron. Ital., 79, 440
Bonanos, A. Z., Stanek, K. Z., Kudritzki, R. P., et al. 2006, ApJ, 652, 313
Brunthaler, A., Reid, M. J., Falcke, H., Greenhill, L. J., \& Henkel, C. 2005, Science, 307, 1440
Conn, A. R., Lewis, G. F., Ibata, R. A., et al. 2011, ApJ, 740, 69
Ferguson, A. M. N., Gallagher, J. S., \& Wyse, R. F. G. 2000, AJ, 120, 821
Geha, M., van der Marel, R. P., Guhathakurta, P., et al. 2010, ApJ, 711, 361
Gwyn, S. 2010, http://cadcwww.dao.nrc.ca/megapipe/docs/filters.html
Hodge, P. W. 1963, AJ, 68, 691
Ibata, R., Martin, N. F., Irwin, M., et al. 2007, ApJ, 671, 1591
Iben, I., J.., \& Renzini, A. 1983, ARA\&A, 21, 271
Irwin, M. J., Ferguson, A. M. N., Huxor, A. P., et al. 2008, ApJ, 676, L17
Koch, A., \& Grebel, E. K. 2006, AJ, 131, 1405
Lee, M. G., Freedman, W. L., \& Madore, B. F. 1993, ApJ, 417, 553
Letarte, B., Chapman, S. C., Collins, M., et al. 2009, MNRAS, 400, 1472
Majewski, S. R., Beaton, R. L., Patterson, R. J., et al. 2007, ApJ, 670, L9
Martin, N. F., Ibata, R. A., Irwin, M. J., et al. 2006, MNRAS, 371, 1983
Martin, N. F., McConnachie, A. W., Irwin, M., et al. 2009, ApJ, 705, 758
McConnachie, A. W. 2005, PhD thesis, Univ. Cambridge
McConnachie, A. W., Huxor, A., Martin, N. F., et al. 2008, ApJ, 688, 1009
McConnachie, A. W., \& Irwin, M. J. 2006, MNRAS, 365, 902
McConnachie, A. W., Irwin, M. J., Ibata, R. A., et al. 2009, Nature, 461, 66
McConnachie, A. W., Irwin, M. J., Ferguson, A. M. N., et al. 2004, MNRAS, 350, 243
McConnachie, A. W., Irwin, M. J., Ferguson, A. M. N., et al. 2005, MNRAS, 356, 979
Metz, M., Kroupa, P., \& Jerjen, H. 2007, MNRAS, 374, 1125
Richardson, J. C., Irwin, M. J., McConnachie, A. W., et al. 2011, ApJ, 732, 76
Riess, A. G., Fliri, J., \& Valls-Gabaud, D. 2012, ApJ, 745, 156
Rockosi, C. M., Odenkirchen, M., Grebel, E. K., et al. 2002, AJ, 124, 349
Salaris, M., \& Cassisi, S. 1997, MNRAS, 289, 406
Schlegel, D. J., Finkbeiner, D. P., \& Davis, M. 1998, ApJ, 500, 525
Skrutskie, M. F., Cutri, R. M., Stiening, R., et al. 2006, AJ, 131, 1163
Springel, V., Wang, J., Vogelsberger, M., et al. 2008, MNRAS, 391, 1685
Stanek, K. Z., \& Garnavich, P. M. 1998, ApJ, 503, L131
Tammann, G. A., Sandage, A., \& Reindl, B. 2008, A\&AR, 15, 289
Yang, S.-C., \& Sarajedini, A. 2012, MNRAS, 419, 1362
Zucker, D. B., Kniazev, A. Y., Martínez-Delgado, D., et al. 2007, ApJ, 659, L21

## Paper III: The Three Dimensional Structure of the M31 Satellite System;

Strong Evidence for an Inhomogeneous<br>Distribution of Satellites

## Paper III Preface

This paper in many respects represents the climax of the thesis and the culmination of the work presented in the three previous chapters. Specifically, it takes the satellite distance distributions presented in the previous chapter (i.e. Paper II), converts them into threedimensional positions and then proceeds with a thorough analysis of the resulting distribution, leading to some exciting revelations as to the structure of the satellite system. Central to the work presented in Paper III are a variety of tools utilized throughout the analysis. These tools are discussed in the method section of the paper, but they are elaborated upon here in an effort to help the reader visualize the processes described and thus provide a suitable preface to the forthcoming material.

The first consideration in our analysis of the M31 satellite system must be to devise a suitable means for viewing the distribution in a clear and consistent way. Since we are concerned with a system completely external to our own Milky Way, it is intuitive to depart from our Earth-bound view and instead view the system as it would appear from the center of M31. To do this, we shift to M31-centric galactic coordinates. The convention in this regard has been to orient the north galactic pole $\left(b=+90^{\circ}\right)$ in the reverse direction to that of the net angular momentum of the disk (i.e. perpendicular to the disk of the galaxy) with the meridian of longitude $l=0^{\circ}$ aligned so as to pass through the Milky Way (or specifically Earth). Hence, we take the $x, y$ and $z$ coordinates derived as per Fig. 4.1 and perform a rotation of coordinates (quantified in the method section of the paper - see 'PlaneSigRMS.f95' in Appendix E for implementation) so as to bring our satellite positions into the new coordinate system. For plotting purposes, we can then simply convert from cartesian to spherical coordinates and plot each object's latitude and longitude in an Aitoff-Hammer projection (see aitoff_hammer.f95 in Appendix E), where positive $x$ points in the direction of $l=0^{\circ}, b=0^{\circ}$; positive $y$ toward $l=-90^{\circ}, b=0^{\circ}$; and positive $z$ toward $l=0^{\circ}, b=+90^{\circ}$.

With a suitable method for visualizing the satellite distribution devised, the next consideration in our analysis concerns plane fitting. The whole of Paper III is essentially built around plane fitting, whether it be to identify planes of satellites within the distribution, or the asymmetry of the distributions as a whole. Where we are interested in identifying
physically significant planes or disks of satellites, we need to seek out the plane that most closely approximates the constituent satellites as determined by any one of many possible 'goodness-of-fit' statistics. When instead we wish to find the magnitude and direction of the asymmetry of the distribution, we need simply find that plane which divides the sample most unequally. What ever the application, for ease of implementation and versatility, a simple scanning routine was chosen to accomplish the task.

The algorithm devised for the plane fitting is best understood by visualizing a plane pivoted at the center of our coordinate system (i.e. the center of M31) with its normal vector projecting out from this point. This plane is then rotated such that its normal vector or 'pole' scans a complete hemisphere of the sky. In so doing, every possible orientation of the plane is passed through exactly once. In practice, discrete pointings of the normal vector are used and at each one the goodness-of-fit statistic or the asymmetry is measured and compared with the best-fit value encountered so far. It is then either stored or discarded accordingly. Thus by the completion of the scan, the true best-fit pole has been identified and retained. In order to scan the hemisphere at a suitably high resolution whilst retaining computational efficiency, a low resolution scan is made first and then a localized high resolution search initiated about the best-fit pole. This process is illustrated in Fig. 5.1, where the poles tested in a single instance of plane fitting have been plotted using the TOPCAT graphics program (Taylor, 2005). Once the pole to the best-fit plane has been identified via this method, its orientation can then be converted into M31-centric latitude and longitude and plotted on an aitoff-hammer projection. This may be done once, as in the case for a particular set of satellites, or many times, in order to produce a pole distribution plot representing all possible combinations of a particular number of satellites for instance.

We are now equipped to identify the best-fit plane for a particular set of satellites, but this is only part of our analysis. We now need to ascertain the significance of the identified plane; is it likely to be a chance alignment or is it a real physical structure? To answer this question, we need a method by which the identical measurement can be performed repeatedly on a large number of 'realizations' of randomly distributed artificial satellites, each one subject to the same constraints as the real data. Our first requirement in producing



Figure 5.1: Plane Fitting; Poles of Tested Planes. This figure illustrates the plane fitting method utilized for most of the analysis contained in Paper III. Figure (a) shows a hemisphere of equally spaced points, each one the pole of a tested plane. This plot represents the low resolution scan undertaken for every instance of plane fitting to a particular set of satellites. Once the scan is complete, a high resolution scan is undertaken about the best-fit pole. Figure (b) is a close up of several of these high resolution scans, indicating the effective resolution of each scan. Note that several adjacent scans are shown to indicate the overlapping coverage. For a given instance of plane fitting, only one high resolution search (i.e. square) need be made. Plotted using TOPCAT.
such 'random realizations' is a tool by which a given satellite position vector can be spun around to any random position on the M31 sky. Development of such a tool is not a trivial task, as lines of constant latitude on a sphere are not great circles, but decrease in diameter toward the poles. Thus simply drawing a latitude and longitude at random will produce a disproportionate number of satellites at high latitude. We therefore weight the probability of drawing a particular latitude in proportion to the cosine of the latitude. This is illustrated in Fig. 5.2 (a). Fig. (b) shows 10,000 unit vectors distributed truly randomly following this procedure. Note that this same process is not only applicable to positioning satellites randomly on the sky but is also another means of generating random poles for plane fitting (see $\S 3.4$ of Paper III for instance).

With the means to spin satellite position vectors to random orientations now in place, we can proceed to build our random satellite realizations. The general procedure then is as follows. First, the desired number of satellites to be included in the random realization is chosen. This is always the total number of satellites in the real distribution i.e. usually 27, but 25 where NGC147, NGC185 and Andromeda XXX are grouped together as a single
point (see $\S 3.3$ of Paper III). Then to generate a position for each artificial satellite, one of the real satellites is chosen at random and a distance is drawn from its associated distance distribution. The M31-centric position vector is then calculated and spun around to a new random orientation as described above. As we wish to subject our random sample to the same constraints as the real sample, it is very important at this point that we verify that the new orientation does not place the object outside of the utilized region of the PAndAS survey area. We therefore project the new satellite position back onto the sky and determine whether it meets this criterion. If it does, we proceed to generate a position for the next satellite following the same procedure, if it does not, we reject the new position and likewise repeat the process, until we get an acceptable position for the current satellite. The process repeats until the desired number of satellites are produced.

By this point, we have generated our random realization with one possible position for each satellite. If we are to mimic the data most closely, we should have in effect a full line-of-sight distance distribution for each artificial satellite. Hence once we have determined a single set of acceptable three-dimensional positions for the satellites, the positions of each on the sky as viewed from Earth are stored, as are the new Earth-to-satellite distances. The original satellite distance distributions used for each one can then be sampled and appropriately mapped to the new positions. Fig. 5.3 below illustrates the procedure for the random generation of satellites. Figure (a) illustrates the positions on the sky of 1000 accepted satellite positions as viewed from Earth. As with most of the figures in this preface, this figure was generated to verify the correct behavior of the algorithm. Hence a very large number of satellites were generated in order to insure that all accepted satellites did indeed fall inside the utilized portion of the PAndAS survey area. Figure (b) likewise was generated to insure that the final random realizations had the correct appearance. It shows 1000 possible positions drawn for each of 27 artificial satellites.

With all of the above tools in place, we are now in a position to embark on our analysis of the three dimensional structure of the M31 satellite system. As shall become apparent in the paper, the analysis reveals some very interesting results. There can, for instance, be very little doubt that the distribution is significantly inhomogeneous, with degrees of planarity
and asymmetry observed which are shown to be very unlikely to arise by chance. Most striking is a very thin plane or disk made up of 15 satellites from the total sample of 27. The orientation of this 'great plane' is also of particular note. Indeed, the identification of this plane has culminated in a publication in nature (Ibata et al., 2013), wherein velocity measurements are added to the spatial measurements presented in this chapter to further characterize what is a most remarkable and important structure. The final accepted version is reproduced immediately after the thesis conclusions in Appendix A.
(The published version of Conn et al. 2013 is reproduced by permission of the AAS)


Figure 5.2: Method for rotating vectors to random angles. Fig. (a) illustrates the calculation of the necessary weighting factor as a function of latitude. Fig. (b) shows 10,000 unit vectors spun to random angles after incorporating this weighting factor. Note that had this weighting factor not been included, the density of poles would be greater at higher latitudes. Fig. (b) plotted using TOPCAT.


Figure 5.3: Generating random satellite realizations. Fig (a) illustrates the acceptable on-sky positions in which artificial satellites are allowed to appear, via the creation of a large 1000 -satellite random realization. Note that satellites are less likely to be positioned at larger distances from M31 due to the small number of satellites in the real sample lying at equivalent distances. Figure (b) plots a single random realization of 27 satellites complete with sampled distance distributions for each. This is the form of the random realizations used in the actual analysis. Fig. (b) plotted using TOPCAT.

# THE THREE-DIMENSIONAL STRUCTURE OF THE M31 SATELLITE SYSTEM; STRONG EVIDENCE FOR AN INHOMOGENEOUS DISTRIBUTION OF SATELLITES 

A. R. Conn ${ }^{1,2,3}$, G. F. Lewis ${ }^{4}$, R. A. Ibata $^{3}$, Q. A. Parker ${ }^{1,2,5}$, D. B. Zucker ${ }^{1,2,5}$, A. W. McConnachie ${ }^{6}$, N. F. Martin ${ }^{3,7}$, D. Valls-Gabaud ${ }^{8}$, N. Tanvir ${ }^{9}$, M. J. Irwin ${ }^{10}$, A. M. N. Ferguson ${ }^{11}$, and S. C. Chapman ${ }^{12}$<br>${ }^{1}$ Department of Physics \& Astronomy, Macquarie University, NSW 2109, Australia<br>${ }^{2}$ Research Centre in Astronomy, Astrophysics and Astrophotonics (MQAAAstro), Macquarie University, NSW 2109, Australia<br>${ }^{3}$ Observatoire Astronomique de Strasbourg, Université de Strasbourg, CNRS, UMR 7550, 11 rue de l'Université, F-67000 Strasbourg, France<br>${ }^{4}$ Sydney Institute for Astronomy, School of Physics, A28, University of Sydney, Sydney NSW 2006, Australia<br>${ }^{5}$ Australian Astronomical Observatory, P.O. Box 296, Epping, NSW 2121, Australia<br>${ }^{6}$ NRC Herzberg Institute of Astrophysics, 5071 West Saanich Road, Victoria, British Columbia, V9E 2E7, Canada<br>${ }^{7}$ Max-Planck-Institut für Astronomie, Königstuhl 17, D-69117 Heidelberg, Germany<br>${ }^{8}$ Observatoire de Paris, LERMA, 61 Avenue de l'Observatoire, F-75014 Paris, France<br>${ }^{9}$ Department of Physics and Astronomy, University of Leicester, Leicester LE1 7RH, UK<br>${ }^{10}$ Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK<br>${ }^{11}$ Institute for Astronomy, University of Edinburgh, Royal Observatory, Blackford Hill, Edinburgh EH9 3HJ, UK<br>${ }^{12}$ Department of Physics and Atmospheric Science, Dalhousie University, 6310 Coburg Road, Halifax, Nova Scotia, B3H 4R2, Canada<br>Received 2012 November 8; accepted 2013 January 29; published 2013 March 15


#### Abstract

We undertake an investigation into the spatial structure of the M31 satellite system utilizing the distance distributions presented in a previous publication. These distances make use of the unique combination of depth and spatial coverage of the Pan-Andromeda Archaeological Survey to provide a large, homogeneous sample consisting of 27 of M31's satellites, as well as M31 itself. We find that the satellite distribution, when viewed as a whole, is no more planar than one would expect from a random distribution of equal size. A disk consisting of 15 of the satellites is however found to be highly significant, and strikingly thin, with an rms thickness of just $12.34_{-0.43}^{+0.75} \mathrm{kpc}$. This disk is oriented approximately edge-on with respect to the Milky Way and almost perpendicular to the Milky Way disk. It is also roughly orthogonal to the disk-like structure regularly reported for the Milky Way satellite system and in close alignment with M31's Giant Stellar Stream. A similar analysis of the asymmetry of the M31 satellite distribution finds that it is also significantly larger than one would expect from a random distribution. In particular, it is remarkable that 20 of the 27 satellites most likely lie on the Milky Way side of the galaxy, with the asymmetry being most pronounced within the satellite subset forming the aforementioned disk. This lopsidedness is all the more intriguing in light of the apparent orthogonality observed between the satellite disk structures of the Milky Way and M31.


Key words: galaxies: dwarf - galaxies: halos - galaxies: individual (M31) - galaxies: structure
Online-only material: color figures

## 1. INTRODUCTION

The possibility that irregular distributions of satellite galaxies may be a common feature of large galaxy halos was originally bolstered by several studies of the anisotropic distribution of our own galaxy's satellites. Lynden-Bell (1976) found that the Magellanic Stream along with Sculptor and the Draco-Ursa Minor Stream and their associated dwarf spheroidal galaxies all appear to lie in the orbital plane of the Magellanic Clouds. In Lynden-Bell (1982), all the then-known dwarf spheroidal companions of the Milky Way are identified as lying in one of two streams. Kroupa et al. (2005) examined the likelihood of producing the observed disk-like distribution of Milky Way satellites from a spherical or oblate dark matter halo. From comparisons with theoretical isotropic satellite distributions produced from such a halo, they find that the chance of producing the observed distribution from the dark-matter subhalos of cold-dark-matter (CDM) cosmology is less than $0.5 \%$. They examine various combinations of the inner most satellites and find a best-fit plane that is almost perpendicular to the plane of the Milky Way with an rms height ranging from only about $10-30 \mathrm{kpc}$. Zentner et al. (2005), while finding a similar plane to Kroupa et al. (2005) for the Milky Way satellites, disagree with their assumption that such a plane is unlikely to arise from a conventional CDM dark matter halo. They argue that that the
most luminous satellites cannot be taken for granted as forming randomly from the isotropic sub-halo distribution but instead, lie preferentially at smaller distances from the halo center and co-planar with the major axis of the host halo. Coupled with the finding that galaxies preferentially align themselves with their major axis highly inclined or even perpendicular to that of the surrounding matter (e.g., Navarro et al. 2004; Hartwick 2000), this would provide a good explanation for the observed orientation of the best-fit plane. The arguments of Zentner et al. (2005) are contended by Metz et al. (2007), however, and it should be noted that in contrast to Zentner et al. (2005), Libeskind et al. (2005) found that the distribution of the most massive sub-halos is not as flattened as the distribution of the Milky Way's satellites.

More recently, Lovell et al. (2011), using the six halo models in the Aquarius Simulations (Springel et al. 2008), find that all six halos produce a significant population of sub-halos with quasi-planar orbits aligned with the main halo spin. This, they argue, is a natural explanation for the observed satellite distribution of the Milky Way. Pawlowski et al. (2012a) argue against this, however. With the calculation of the angular momenta of 8 Milky Way Satellites (Metz et al. 2008) revealing a strong alignment between six of the orbital poles, Pawlowski et al. (2012a) examine the likelihood of randomly drawing six sub-halos from each of the six Aquarius simulations (among other halo
simulations) and finding a similar degree of alignment. More precisely, they draw $10^{5}$ sets of eight satellites from each of the six simulations, and select the six with the highest degree of alignment between their orbits, thus emulating the findings of Metz et al. (2008). They then look at the degree of clumping of the orbital poles $\Delta_{\text {sph }}$ as well as the angular distance of the average of the orbital pole inclinations from the model equator $d$ and find that the actual degree of planarity observed for the six satellites identified by Metz et al. (2008) $\left(\Delta_{\text {sph }}^{\mathrm{MW}}=35^{\circ} 4\right.$ and $d_{\mathrm{MW}}=9.4$ ) are equaled or exceeded in the random draws in less than $10 \%$ of cases when $\Delta_{\text {sph }}$ is considered and less than $15 \%$ of cases for $d$. Starkenburg et al. (2013) also find that the degree of planarity observed for the Milky Way satellites is uncommon in all six of the Aquarius halos (see Figure 7 of that study).

In addition to the revelation that the Milky Way's satellites appear to inhabit highly inclined great planes, they also appear to corroborate the finding of Holmberg (1969), namely that the companions of Spiral Galaxies preferentially congregate at high galactic latitudes (the Holmberg Effect), as observed in his study of 174 galaxy groups. It is not clear why this should be the case, or even if it truly is the case, although if the apparent adherence of satellite systems to polar great planes is typical of galaxies in general, then the Holmberg Effect seems to be an extension of this. Quinn \& Goodman (1986) proposed that dynamical friction may be responsible for the observed polar great planes, with those orbits spending the most time in close proximity to the galactic disk experiencing the fastest decay, while those that take the most direct route through the disk environs, namely the polar orbits, experience the slowest orbital decay. It should be noted however that dynamical friction, while producing more polar orbits, would not produce planes of satellites. Nor would it be effective on the young globular clusters which are shown to be co-planar with the "Vast Plane of Satellites" identified around the Milky Way by Pawlowski et al. (2012b). Indeed, Angus et al. (2011) show that dynamical friction did not play a role in the formation of the Milky Way satellite orbits.

Besides the conjecture that satellite great planes trace the major axis of the dark matter halo in which the parent galaxy resides, there are other proposed mechanisms for their creation. One hypothesis is that these planes trace the orbits of ancient galaxies that have been cannibalized by the host galaxy. Palma et al. (2002) have investigated this hypothesis by looking for planes among groups of satellite galaxies and globular clusters in the Milky Way's outer halo and find various members to be co-planar with either the Magellanic or Sagittarius streams. The findings of Lynden-Bell \& Lynden-Bell (1995) are also consistent with such a hypothesis. Indeed, it is this hypothesis which is most strongly supported by Pawlowski et al. (2012a), wherein the $\Delta_{\text {sph }}$ and $d$ of satellites drawn from various tidal models equal or exceed $\Delta_{\mathrm{sph}}^{\mathrm{MW}}$ and $d_{\mathrm{MW}}$ in over $80 \%$ of draws in some cases. A similar hypothesis, which in some regards links the galaxy-cannibalization and dark-matter hypotheses, proposes that the observed planes result from the orientation of the large-scale filamentary structure of galaxy clusters (e.g., Knebe et al. 2004), an orientation traced out by those minor galaxies which fall into the halo of a major galaxy. Metz et al. (2009b) argue however that extragalactic associations of dwarf galaxies are too extended to account for the high degree of planarity observed for the Milky Way satellites. This argument is supported by the findings of Vera-Ciro et al. (2011) based on the Aquarius Simulations.

The great obstacle to a conclusive resolution of these issues is the lack of systems for which reliable spatial (and kinematic)
data exists. While some such data does exist for large galaxy clusters such as Virgo and Coma, accurate 3D distributions of galaxies within their halo have for a long time been known only for our own galaxy's halo, ascertainable due to our central position within it. It has only been in recent times that a second system has opened up to us-that of our counterpart in the Local Group, M31. While various databases of photometry and other data have been available for M31 and some of its brighter companions for over a decade, it is the PanAndromeda Archaeological Survey (PAndAS-McConnachie et al. 2009)—a deep photometric, two-color survey providing a uniform coverage of the M31 halo out to approximately 150 kpc - that has provided a new level of detail for this system. It is from this survey that we obtained our distances to M31 and 27 of its companions, following the method developed in Conn et al. (2011, henceforth CLI11) and further adapted for this purpose in Conn et al. (2012, henceforth CIL12). The distances themselves and their associated uncertainty distributions are presented in CIL12 and it is these distributions that are utilized for all analysis contained in this paper.
With regard to previous studies of the anisotropy in the M31 satellite distribution, two investigations warrant consideration at this point. McConnachie \& Irwin (2006), making use of Wide Field Camera photometry from the Isaac Newton Telescope in what was essentially the forerunner to the PAndAS Survey, focus on "Ghostly Streams" of satellite galaxies following a similar approach as Lynden-Bell \& Lynden-Bell (1995) used for the Milky Way. In addition, they characterize the large degree of asymmetry in the satellite distribution, a feature also noted in CIL12, and examine the radial distribution of the satellites, noting a (statistically insignificant) larger average distance from M31 than that observed between the Milky Way and its satellites. They find a large number of candidate satellite streams, with some favoring the dwarf spheroidal members. Koch \& Grebel (2006) utilize distance measurements from a variety of sources and focus particularly on planes of satellites and, while they do not find a particularly significant best-fit plane when their whole satellite sample is considered, it is rather interesting that they find a $99.7 \%$ statistical significance to their best-fit plane when the then-known dwarf spheroidal galaxies dominate their sample. Furthermore, this plane is near-polar-as has been observed for the Milky Way, although they find little support for the Holmberg Effect. Koch \& Grebel (2006) utilize a particularly robust method in their search for high-significance planar fits to subsets of galaxies by considering every possible combination of a given number of satellites from their sample.
In the current study we employ a similar approach, but with the great advantage of having a considerably extended sample of galaxies in our sample, with all distances derived by the same method and from the same data as described in CLI11 and CIL12. (Distances are sampled from the distributions without the density prior applied-e.g., CIL12, Table 1, Column 2-while the best-fit distances are drawn from CIL12, Table 2, Column 4.) As a result, we are able to give full consideration to the effects of selection bias on the observed satellite distribution. This then presents an excellent opportunity to greatly improve our knowledge of the three-dimensional structure of the M31 satellite distribution, with important implications regarding the recent evolution of the system.
A breakdown of the structure of the paper is as follows. In Section 2, we outline our method for plane fitting (Section 2.1) and locating significant planes of satellites as well as the orientation, magnitude and significance of the asymmetry of
the distribution. A method for generating random realizations of satellites subject to the same selection biases as the real data is also discussed in this section (Section 2.2) as is the selection bias itself (Section 2.3). Section 3 then presents the results of applying these methods, first to the sample as a whole, and then to subsets of galaxies. Specifically, Section 3.1 presents a study of planarity within the satellite system when all satellites contribute to the determination of the best-fit plane; Section 3.2 examines the asymmetry in a similar way; Section 3.3 examines the orientations of planes of smaller subsets of satellites within the distribution; and Section 3.4 concludes this section with a determination of the significance of a "Great Plane" of satellites emerging from the preceding sections. Sections 4 and 5 then follow with discussion and conclusions.

Note that this paper was written in conjunction with a shorter contribution (Ibata et al. 2013, hereafter ILC13) which announced some of the key discoveries resulting from the analysis we present here. In particular, the process of identifying the member satellites of the "great plane" discussed in ILC13 is described here in more detail. In this analysis, however, we concern ourselves with the spatial structure of the satellite system only and so the reader should refer to ILC13 for the interesting insight provided by the addition of the velocity information.

## 2. METHOD

### 2.1. Plane Fitting

In order to find planes of satellites within the M31 satellite system, our first concern is to convert the satellite distances as presented in CIL12 into three-dimensional positions. To do this, we begin with an M31-centered, Cartesian coordinate system oriented such that the $x$ and $y$ axes lie in the M31 tangent plane with the $z$-axis pointed toward the Earth. Specifically, the $x$-axis corresponds to $\eta_{\text {tp }}=0$ which is the projection of M31's Declination onto the tangent plane. The $y$-axis then corresponds to $\xi_{\text {tp }}=0$-the projection of M31's right ascension onto the tangent plane. The $z$-axis then points along the Earth-to-M31 vector, with magnitude increasing with distance from Earth. This orientation can be seen in Figure 10(c) of CIL12. Thus,

$$
\begin{align*}
& x=D_{\text {sat }} \cos (\theta) \tan (\xi) \\
& y=D_{\text {sat }} \sin (\eta) \\
& z=D_{\text {sat }} \cos (\theta)-D_{\mathrm{M} 31}, \tag{1}
\end{align*}
$$

where $D_{\mathrm{M} 31}$ and $D_{\text {sat }}$ are the distances from Earth to M31 and from Earth to the satellite, respectively, $\theta$ is the angular separation on the sky between M31 and the satellite, and $\eta$ and $\xi$ are the real-angle equivalents of the tangent plane projection angles $\eta_{\mathrm{tp}}$ and $\xi_{\mathrm{tp}}$, respectively.

Next, we rotate this reference frame to the conventional M31 reference frame such that the positive $z$-axis points toward M 31 's north galactic pole ${ }^{13}$ (i.e., $b_{\mathrm{M} 31}=+90^{\circ}$ ) and the $l_{\mathrm{M} 31}=$ $0^{\circ}$ meridian passes through the Earth. So as to be consistent with the earlier work of McConnachie \& Irwin (2006), we have adopted the same values for M31's position angle (39.8) and inclination ( 77.5 -de Vaucouleurs 1958). Each object is hence rotated by 39.8 about the $z$-axis to counter the effect of its position angle, and then 77.5 about the $x$-axis to account for M31's inclination. A final rotation of $90^{\circ}$ about the $z$-axis is then

[^4]necessary to bring $l_{\mathrm{M} 31}=0^{\circ}$ into alignment with the direction of Earth (which hence lies at $l_{\mathrm{M} 31}=0^{\circ}, b_{\mathrm{M} 31}=-12.5$ ). The resulting spherical coordinates for each object in the sample are plotted onto an Aitoff-Hammer projection in Figure 1. This same figure also shows the uncertainties in position associated with each object, generated via sampling of the respective distance posterior probability distributions (PPDs) of each object and subsequent conversion of each drawn distance into a threedimensional position.
With the satellites' positions determined in Cartesian coordinates, it is straightforward to determine the minimum distance of each satellite from a given plane as follows:
\[

$$
\begin{equation*}
D_{\text {plane }}=|a x+b y+c z+d|, \tag{2}
\end{equation*}
$$

\]

where $D_{\text {plane }}$ is the distance of a satellite at a point $(x, y, z)$ from a plane whose normal vector is $(a, b, c)$ and is of unit length. For simplicity, we invoke the reasonable requirement that all planes must pass through the center of M31 and so in our case, $d=0$ and the plane normal vector points out from the center of M31. Hence, in order to find the best-fit or maximum significance plane to a set of satellites, we need simply minimize $D_{\text {plane }}$ for the satellites to be fitted. This can be done via a variety of means, some of which are compared in the following section, but perhaps the most robust and the predominant method employed in this study, is that of minimizing the rms of the distances to the fitted satellites.
In order to measure the asymmetry of the satellite distribution about a given plane, we need only count the number of satellites on one side of the plane. To do this, we can simply remove the absolute value signs from Equation (2), so that the side of the plane on which a satellite lies can be determined by whether $D_{\text {plane }}$ is positive or negative. The plane of maximum asymmetry is then taken to be that which divides the sample such that the difference in satellite counts for opposite sides of the plane is greatest.
Whether we wish to determine the best-fit plane through a sample of satellites or the plane of maximum asymmetry, we require a system by which a large number of planes can be tested on the sample so that the goodness of fit (or asymmetry) can be calculated for each. To do this, we define each tested plane by its normal vector or pole ( $a, b, c$ ) so that Equation (2) can be applied directly. We then rotate this pole to different orientations around the sky in such a way as to "scan" the whole sphere evenly and at a suitably high resolution. In practice, we need to be able to apply this routine many thousands of times for a large number of samples and so a fast computational time is of the essence. To this end, for a given sample, our algorithm determines the desired plane following a two step procedure.
First, a low-resolution scan of the sphere is made to determine the approximate direction on the sky of the pole to the bestfit plane. Only half the sphere actually needs to be scanned since poles lying on the opposite hemisphere correspond to the identical planes flipped upside down. The low-resolution scan tests 2233 different poles across the hemisphere. A near-uniform coverage is achieved by decreasing the number of planes tested in proportion to the cosine of the latitude of the planes' pole. This prohibits what would otherwise be an increased coverage at the higher latitudes of the coordinate system. With the pole to the best-fit plane determined in low-resolution, a high-resolution search is then made around the identified coordinates at 10 times the resolution. In this way a pole can effectively be found at any of approximately 250,000 evenly spread locations on the hemisphere.


Figure 1. An Aitoff-Hammer Projection showing the positions of M31's satellites, along with their associated $1 \sigma$ uncertainties. The positions show where each object would appear in the sky if viewed from the center of M31, and are given in M31's inherent galactic latitude and longitude. The position of the Milky Way is also shown for reference. The position uncertainties trace single arcs across the M31 sky, rather than two-dimensional patches on account of the restriction of the uncertainty to lie solely along the line of sight to the object from Earth. For this same reason, all the lines point radially outward from the Milky Way. These uncertainties also take into account the uncertainty in M31's distance. The existence of a prominent plane, broadly consisting of Andromedas I, XI, XII, XIII, XIV, XVI, XVII, XXV, XXVI, XXVII, XXX, and NGC 147 and NGC 185, is hinted at by the close proximity of their respective arcs.
(A color version of this figure is available in the online journal.)

### 2.2. Generating Random Satellite Samples

While we are now equipped to identify best-fit planes to our sample and subsamples thereof, it is necessary to have some means of determining the significance of these planes in an absolute sense. The most intuitive way to do this is to perform the same analysis on a randomly generated sample of equal size. In particular, when we are concerned with all possible combinations of a particular number of satellites that can be produced from the whole sample, we are often dealing with a very large number of subsamples and so it is inevitable that some of these subsets of satellites will exhibit a very high degree of planarity. Identical analysis must therefore be performed on random distributions, to see if there are similar numbers of subsets with equal degrees of planarity.

For this reason, considerable care was taken to design an algorithm capable of providing a unique random realization of the desired number of satellites whenever it is called. The algorithm makes use of the distance PPD for each satellite, and also takes into account the irregular window function (i.e., useable portion) of the PAndAS survey. Each time a satellite is to be added to the random realization, one of the 27 actual satellites is chosen at random and a distance is drawn from its associated PPD. This distance ( $D_{\text {sat }}$ ) is then converted into a three-dimensional position $(x, y, z)$ following equation set 1 and this satellite-to-M31 separation vector is then spun around to a new, random location in the M31 sky. Note that for each random realization, a new value of $D_{\mathrm{M} 31}$ is similarly drawn from the M31 distance PPD.

Once again, care must be taken in this step to ensure that the whole sphere is given equal weight, otherwise there is a higher likelihood for the artificial satellites to be positioned at high latitude. Again, this is remedied by weighting the likelihood by the cosine of the latitude.

With the new, random location for the satellite chosen, it is then projected back onto the sky as it would appear from Earth and a check is made to ensure that it does indeed lie within the boundaries of the PAndAS survey area, and outside of the
central ellipse ( $5^{\circ}$ major axis, $2^{\circ}$ minor axis-see Figure 10 (c) of CIL12) where the disk of M31 inhibits reliable measurements. If the satellite does not meet these requirements, it is rejected and the satellite drawing process is repeated until a suitable position is generated. By repeating this process until the desired number of satellites are produced, a new, random comparison sample is generated which gives full account to the constraints on the actual data.

In order for the random satellite realizations to mimic the actual data most closely, it is necessary that each artificial satellite is represented not by just one point, but rather a string of points reflecting the uncertainty in the Earth-to-Object distance. Hence once acceptable positions for each satellite are drawn as described above, the distance distributions for each object are sampled and projected to their equivalent positions along the line of sight about the initially placed point. For Sections 3.1, 3.2, and 3.4 each artificial satellite's distance distribution is represented by 1000 points such that each planefitting measurement is made for 1000 possible positions of the object and then the average value of the measurements is taken. The only exception to this number is where the maximumlikelihood approach is used in Section 3.1. Due to the inclusion of a second fitting parameter in this case, only 100 samples are taken for each satellite. For Section 3.3, as we are not concerned with comparisons of plane significance between the real sample and the random realizations, it is sufficient to use a single drawn position for each artificial satellite.

### 2.3. A Note on Satellite Detection Bias

By employing a similar method to that described above, it is also possible to explore the effect of the PAndAS survey area boundaries on the satellite detection bias as viewed from the center of M31. It is intuitive that more satellites are likely to be detected along the line of sight to Earth, since even satellites at a large distance from M31 will still appear within the survey boundaries if they lie along this line. We can visualize this effect by generating a large number of randomly distributed satellites


Figure 2. An Aitoff-Hammer projection illustrating the satellite detection bias resulting from the PAndAS survey boundaries and M31 disk obstruction. Note that this figure utilizes a Gaussian blurring of radius $5^{\circ}$, as do all of the subsequent pole-density plots.
(A color version of this figure is available in the online journal.)
and plotting them on the M31 sky after first rejecting those satellites that would appear outside the survey area "mask" if viewed from Earth. To do this, one million satellites were drawn from a spherically symmetric halo potential with density falling off as a function of the square of the distance from the halo center. Satellites were hence drawn at distances between 0 and 700 kpc from M31 with equal probability. The satellites were then projected onto the M31 tangent plane and those satellites lying outside the survey area or inside the M31 disk obstruction area were excised from the density map. The resulting anisotropy of the satellites on the M31 sky is presented in Figure 2.

As can be seen from the figure, the probability of detection is indeed higher along a great circle oriented edge-on with respect to the direction of Earth, and perpendicular to the M31 disk $\left(b_{\mathrm{M} 31}=0^{\circ}\right)$. This great circle has its pole/anti-pole at $l_{\mathrm{M} 31}= \pm 90^{\circ}, b_{\mathrm{M} 31}=0^{\circ}$ and hence we would expect a predisposition toward finding planes of satellites with a pole in this vicinity. We would also expect, though to a lesser extent, to find an excess of satellite planes oriented edge-on with respect to Earth at any inclination. Such planes would have poles lying anywhere on the great circle whose normal is directed toward Earth. The drop in the satellite density at $l_{\mathrm{M} 31}=0^{\circ}, b_{\mathrm{M} 31}=-12.5$ and $l_{\mathrm{M} 31}= \pm 180^{\circ}, b_{\mathrm{M} 31}=12.5$ is a consequence of the hindrance to detection caused by the M31 disk. Due to the increased volume of space covered by the survey at greater distances from Earth, unhindered satellite detection is possible over a larger range of angles on the far side of M31 in comparison to the Earth-ward side.

## 3. RESULTS

### 3.1. Best-fit Plane to the Entire Satellite Sample

In order to find the best-fit plane to the satellite system as a whole, the procedure of Section 2.1 is applied to the whole sample of 27 satellites presented in CIL12. The rms thickness of the sample is used here, as in subsequent sections, as the statistic of planarity; we find it to be a robust measure and it has the convenient property of being computationally inexpensive. Since we are dealing with only one sample in this case, two other measures are also used for comparison. The first calculates the sum of the absolute values of the distances of each of the satellites from the tested plane. The second is essentially a maximum-likelihood approach and replaces the plane of zerothickness with a "Gaussian Plane" such that a satellite's position within the Gaussian determines the plane's goodness of fit to that satellite. This second approach requires that different Gaussian


Figure 3. An Aitoff-Hammer Projection showing the best-fit plane to the satellite system as a whole. The pole and anti-pole of the plane are denoted by " + " and " $\times$ " symbols, respectively. Only the best-fit satellite positions were incorporated into the fit for this figure. The distribution of poles obtainable from other possible realizations of the satellite distribution is presented in Figure 4. Note that the plane is near-polar, similar to the preferred plane orientations identified for the Milky Way Satellite System.
(A color version of this figure is available in the online journal.)
widths $\sigma$ be tested for each plane orientation in order to find the width that best matches the satellite distribution. Values between 5 kpc and 150 kpc were tested at 5 kpc intervals for each tested plane orientation. Hence an additional characteristic of the satellite distribution is obtained, but at the expense of a considerably longer computation time.

For each of the three measures of goodness of fit described above, the first step is to find the best-fit plane to the satellite positions with their positions determined from their bestfit distances. When either the rms or maximum-likelihood approach is used, the same best-fit plane is found as $0.153 x+$ $0.932 y+0.329 z=0$ with pole at $\left(l_{\mathrm{M} 31}, b_{\mathrm{M} 31}\right)=\left(-80^{\circ} .7,19^{\circ} 2\right)$. This plane is plotted as a great circle on the M31 sky in Figure 3 with the poles of the plane indicated. When the absolute distance sum is used instead, the pole is found farther from the plane of the galaxy, at $\left(l_{\mathrm{M} 31}, b_{\mathrm{M} 31}\right)=(-74.9,24.3)$. Nevertheless, the polar plane described by Koch \& Grebel (2006) is supported by either measurement, and is reminiscent of the satellite streams identified in the Milky Way satellite system. In light of the detection biases imposed by the PAndAS survey area as illustrated in Figure 2, the result in this case must clearly be treated with suitable caution, however. Like Koch \& Grebel (2006), we find little evidence for the Holmberg Effect, with only three best-fit satellite positions falling within $30^{\circ}$ of the M31 galactic poles, and only six of the $1 \sigma$ error trails from Figure 1 pass beyond $b_{\mathrm{M} 31}= \pm 60^{\circ}$.

To determine the uncertainty in the plane's goodness of fit, we need to repeat the procedure for a large number of realizations of the satellite sample, with the best-fit satellite distances replaced with a distance drawn at random from their respective satellite distance PPDs. A density map of the best-fit plane poles identified from 200,000 such realizations is presented in Figure 4. This figure was generated using the distribution rms as the goodness of fit statistic, and contains $71.1 \%$ of all poles within a $5^{\circ}$ radius of the best-fit pole stated above. When the sum of absolute distances is used in place of the rms, this fraction falls to $68.3 \%$, or to $70.9 \%$ when the maximum-likelihood approach is used. It should be noted that the distribution of poles lies in close proximity to the pole of maximum detection bias at $l_{\mathrm{M} 31}=-90^{\circ}, b_{\mathrm{M} 31}=0^{\circ}$, again suggesting that the detection bias is having a strong influence on the polar orientation of the best-fit plane.


Figure 4. A pole density map showing the effective uncertainty in the location of the best-fit plane to the whole satellite sample. The poles of the best-fit planes derived for 200,000 possible realizations of the data are plotted, along with their corresponding anti-poles.
(A color version of this figure is available in the online journal.)

In order to determine whether the goodness of fit of the best-fit plane is really physically significant, similar analysis should be performed on a large number of random realizations of satellites, to see how often distributions of satellites arise with a comparable degree of planarity. Figure 5 presents probability distributions of the plane significance for possible realizations of the real satellite sample along with average values from random realizations of the satellites (as per Section 2.2), obtained using the three measures of goodness of fit stated above.

It is immediately clear from Figure 5 that regardless of the choice of the measure of goodness of fit, the range of values obtainable from possible realizations of the real satellite positions are similar to the most likely values to be expected from completely random realizations of the satellites. Hence, while a prominent plane of satellites comprising roughly half of the sample is suggested in Figure 1, it would seem that the sample as a whole is no more planar than would be expected from a strictly random distribution. Again, this is in keeping with the findings of Koch \& Grebel (2006), and detracts from any physical significance that should be attributed to the plane's polar orientation.

Further to this finding, the overall width of the "plane" is again in keeping with that expected from a purely random satellite distribution. From fitting the Gaussian Plane to the best-fit satellite positions, a $1 \sigma$ width of 60 kpc is found to produce the best fit to the data. When the 200,000 PPD-sampled realizations were tested, a $1 \sigma$ of 60 kpc was found preferential in $66.3 \%$ of cases, with a $1 \sigma$ of 55 kpc being preferred in $32.7 \%$ of cases. Values of 50 kpc make up the remaining $1 \%$ almost entirely. The average value for the actual satellite distribution was thus determined as 58.3 kpc . This value is similar to the most likely width identified from the 10,000 random realizations, as can be seen in Figure 6.

### 3.2. The Plane of Maximum Asymmetry

To determine the plane of maximum asymmetry and its significance, we employ an identical approach as in the preceding section, but with the goodness of fit statistic replaced with a count of the number of satellites on each side of the plane as per Section 2.1. As was suggested by the three-dimensional satellite distribution generated in CIL12, the asymmetry about the M31 tangent plane is close to a maximum, with 19 satellites on the near-side of the plane but only eight on the other when the best-fit satellite positions are assumed. The highest asymmetry plane possible from this same distribution has 21 satellites on
one side and six on the other, with the equation of the plane identified by the algorithm as $-0.797 x-0.315 y+0.515 z=0$. The anti-pole of this plane lies 27.2 away from the Milky Way at $\left(l_{\mathrm{M} 31}, b_{\mathrm{M} 31}\right)=(-21.6,-31.0)$. This plane is plotted as a great circle on the M31 sky in Figure 7.
When 200,000 realizations of the satellite sample are generated using the satellite's respective distance probability distributions, the most likely asymmetry of the sample is actually found to be greater than this, with 23 satellites on one side and only 4 on the other. Such a scenario is more than twice as likely as the 21:6 scenario. In one realization, a plane was identified which could divide the sample such that all 27 satellites lay in a single hemisphere, while an asymmetry of $26: 1$ was found possible for $815(0.4 \%)$ of the realizations. The distribution of maximumasymmetry poles on the sky, as determined from realizations of possible satellite positions, is illustrated in Figure 8, while Figure 9 (a) plots the probability distribution for the greatest number of satellites that can be found in one hemisphere for a given realization of the observed satellite sample. The average value of this distribution is 22.7 (shown as a dashed line in Figure 9(b)), a value which is equaled or exceeded for 422 out of the 10,000 random realizations represented in Figure 9(b). A maximum asymmetry ratio of 21:6, as was observed for the best-fit satellite distribution plotted in Figure 7, is more common, however, falling inside the $1 \sigma$ credibility interval.
What is particularly striking about the satellite distribution, however, is the orientation of the asymmetry, with the majority of satellites lying on the near-side of the M31 tangent plane. From Figure 9(c), it is clear that the effect of the distance uncertainties lying along the line of sight is to create quite a broad distribution in the level of asymmetry about the tangent plane, though the average is markedly high at 20.3. To investigate the likelihood of this scenario arising from a random satellite distribution, we measure the average number of satellites on either side of the M31 tangent plane for each of 10,000 random realizations as per Section 2.2. The results are illustrated in Figure 9(d). The observed profile is more or less as expected, with a maximum probability close to the minimum possible asymmetry at 14 and then a rapid fall off toward higher asymmetries. It is therefore clear that the distance uncertainties lying along the line of sight have no significant bearing on the orientation of the asymmetry. Yet the observed degree of asymmetry about the M31 tangent plane is equaled or exceeded in only 46 of the 10,000 random satellite realizations and hence is very significant. The possibility that this asymmetry may be a consequence of data incompleteness is currently being examined more closely (see N. F. Martin 2013, in preparation), although it seems very unlikely. The high degree of asymmetry is still observed even when only the brightest satellites are considered. Furthermore, the data incompleteness appears to be dominated by the boundaries of the PAndAS survey area and obstructed regions which are already taken into account by our analysis. Indeed, one would expect more satellites to be observed on the far side of the M31 tangent plane on account of the increased volume of space covered by the survey at greater distances, an effect clearly visible in Figure 2.

### 3.3. Subsets of Satellites

It is perhaps not surprising that the satellite system of M31, when treated as a whole, is no more planar than one would expect from a random sample of comparable size. Indeed, a similar result was noted for the M31 system by Koch \& Grebel (2006). The existence of outliers in our satellite sample was already clear


Figure 5. Probability distributions for the planarity of the entire satellite sample, as determined from three different measures of the plane goodness of fit. The left-hand column of figures gives the distribution of the goodness of fit statistic as obtained via plane fitting to 200,000 separate samplings of the real satellite sample. The right-hand column of figures summarizes the same procedure performed for 1000 separate samplings of each of 10,000 random realizations of the satellites (as per Section 2.2). It is important to note that each histogram in this column has been generated by plotting the average values from the 10,000 individual histograms corresponding to each of the random realizations and hence they should only be compared with the average of the histograms in the left-hand column. The goodness of fit statistic for (a) and (b) is the distribution rms; for (c) and (d) is the absolute distance sum and; for (e) and (f) is the sum of satellite likelihoods. The average of the histograms in (a), (c), and (e) are shown in (b), (d), and (f), respectively, as dashed lines. Red, green, and blue lines denote the extent of $1 \sigma$ ( $68.2 \%$ ), $90 \%$, and $99 \%$ credibility intervals, respectively.
(A color version of this figure is available in the online journal.)
from Figure 1 and furthermore, if multiple planes of differing orientation are present as has been suggested for both the Milky Way's satellite system (e.g., Lynden-Bell 1982; Pawlowski et al. 2012b) and the M31 system (McConnachie \& Irwin 2006), then
the goodness of fit of the best-fit plane to the entire distribution is of little consequence. For this reason, we now concentrate our analysis on subsets or combinations of satellites. Specifically, we perform a pole-count analysis by determining the pole of


Figure 6. The probability distribution for the average $1 \sigma$ width as determined from 10,000 random distributions of 27 satellites. This figure is generated from the same run as Figure 5(f) and is the result of marginalizing over the planeorientation model parameters.
(A color version of this figure is available in the online journal.)


Figure 7. An Aitoff-Hammer projection showing the plane of maximum asymmetry identified from the full sample of best-fit satellite positions. It divides the distribution such that 21 satellites lie in one hemisphere, but only 6 in the other. The anti-pole of the maximum asymmetry plane lies just 28.1 from the Milky Way as viewed from the center of M31.
(A color version of this figure is available in the online journal.)
the best-fit plane to every possible satellite combination of a particular size that can be drawn from the entire sample.

A pole-count analysis is an excellent way of mapping the degree of prominence of various planes that exist within the distribution as a whole, whatever their orientation may be. The choice of combination size is not trivial however. The number of combinations $s$ of a particular number of satellites $k$ that can be drawn from the entire sample of $n$ satellites can be determined as follows:

$$
\begin{equation*}
s=\frac{n!}{k!(n-k)!} \tag{3}
\end{equation*}
$$

For reasons that shall be discussed shortly, we will effectively be working with a sample of 25 satellite positions. It is clear from this equation, however, that with 25 satellites forming the entire sample, the total number of combinations that can be drawn may be very large, depending on the number of satellites forming the combinations. For instance, if $n=25$ and $k=$ 13 , there are over 5.2 million possible combinations that can be drawn. Additionally, if we are to properly account for the uncertainties in the satellite positions, it will be necessary to sample from the distance distributions of each satellite a large number of times for every combination. Given that we must


Figure 8. A pole density map showing the effective uncertainty in the location of the maximum asymmetry plane to the whole satellite sample. The poles of the maximum asymmetry planes derived for 200,000 possible realizations of the data are plotted, along with their corresponding anti-poles. The elongated distributions that run through the pole and anti-pole determined from the best-fit distribution (see Figure 7) arise due to the orientation of the uncertainty trails of the individual satellite positions, as presented in Figure 1. Note that the probability of the anti-pole of the asymmetry lying within a couple of degrees of the direction of the Milky Way is close to a maximum.
(A color version of this figure is available in the online journal.)
test every possible plane orientation (as per Section 2.1) for every rendition of every combination, the computation times can become impracticable. It is therefore necessary to limit our combination sizes as much as possible. We note, however, that the final pole-plot distribution showing the poles of the best-fit planes to each combination, is not so dependent on the combination size as might at first be thought.

With all the planes tested as per Section 2.1 having to pass through the center of M31, the minimum number of satellites that cannot be fitted exactly is three. This is therefore the smallest combination size we consider. There are 2300 combinations of three satellites that can be drawn from the full sample of 25 satellites. If we increase the combination size considerably to seven satellites, there are 480,700 satellite combinations that can be drawn. Due to an excessive number of combinations beyond this point, this is the largest combination size we consider. But it is critical to note that even if we produce our pole-plot map from combinations of only three satellites we do not exclusively find planes consisting of three satellites. If a plane of seven satellites exists for instance, then by Equation (3), such a plane will produce 35 poles at the same location on the pole plot, where a plane consisting of only three satellites would contribute only one pole. Conversely if we take combinations of seven satellites, despite the larger number of possible combinations in total, we become less sensitive to planes made up of less than seven satellites. So in a sense, the combination size we choose depends on the satellite planes we wish to be most sensitive to. In practice, we have found that the smaller combination sizes of three and four satellites are particularly useful for identifying the lowest rms planes congregating around the band of satellites visible in Figure 1. The larger combination sizes of five, six, and seven satellites gradually shift toward finding planes closer to the best-fit plane to the entire satellite sample illustrated in Figure 3.

Noting these points, we proceed as follows. First, the number of satellites per combination $k$ is chosen $(3 \leqslant k \leqslant 7)$ and then for each combination, distances are drawn for each of the satellites from their respective posterior distance distributions as provided in CIL12. To give a satisfactory representation of the form of the distributions, each combination is sampled 100 times. As such, each satellite combination contributes not 1 pole to the pole density map for the chosen combination size but 100 , with


Figure 9. Asymmetry probability distributions. The top two histograms plot probability distributions for the greatest number of satellites that can be found in one hemisphere, as generated from (a) 200,000 samplings of satellite positions possible from the data and; (b) the average of 1000 samplings from each of 10,000 random realizations of the satellites generated as per Section 2.2. Figures (c) and (d) give the equivalent distributions when the maximum asymmetry plane is replaced with the fixed M31 tangent plane. As for Figure 5, the histograms in the right-hand column should only be compared with the average of the corresponding histogram in the left column. The average value of the histograms of (a) and (c) are shown in (b) and (d), respectively, as a dashed line.
(A color version of this figure is available in the online journal.)
the spread of poles relating the possible orientations of the bestfit plane to the combination, given the error in the individual satellite positions. The contribution of each pole to the density map is also weighted by dividing it by the rms of the bestfit plane it represents. Thus each pole does not contribute one count, but rather some fraction, depending on how good a fit the plane it represents is to the satellites in the combination. This fraction is also further divided by 100 , since it represents only $1 \%$ of the samples for the combination, as just discussed.

As stated above, it should also be noted that we effectively limit the total number of satellites in our sample to 25 for all analysis in this subsection. This is to account for the bound group of satellites consisting of NGC 147, NGC 185, and And XXX (henceforth the NGC 147 group-see M. J. Irwin et al. 2013, in preparation). Since we suspect that these satellites orbit M31 as a group and since they all lie along the apparent plane identified in Figure 1, it is preferable to treat the group as a single object when we are not concerned with measurements of the significance of particular planes. To do this, we take the luminosity weighted center as an approximation for the center of mass, and treat this determined position as though it were the location of a single satellite. To calculate the luminosity
weighted center, we can ignore the contribution from And XXX since it is negligible compared with the contributions of the two dwarf ellipticals. From the Third Reference Catalogue of Bright Galaxies (de Vaucouleurs et al. 1991), NGC 185 is 0.2 magnitudes brighter than NGC 147 in the $V$ band. Each time the NGC 147 group is chosen as one of the "satellites" for a combination, distances to each of NGC 147 and NGC 185 are sampled from their respective distributions and the luminosity weighted center of the group is determined. As for any other combination, this position, along with all other satellites in the combination, is sampled 100 times.
The results of applying the above procedure to all combinations of three, four, five, six, and seven satellites that can be drawn from the total sample is presented in Figure 10. The lefthand column shows the fit to the most planar combination determined from the best-fit positions while the right-hand column shows the corresponding pole-density plots for all combinations of that particular number of satellites, based on 100 samples of each combination as per the discussion above. It is noteworthy that the best-fit planes to the most planar combinations are almost identical in every case, except for that of the three satellite combinations, where the rms values are so small for so many


Figure 10. Best-fit planes and pole density maps for combinations of three through seven satellites. The left-hand column shows the best-fit plane through the combination of satellites that can be fit with the lowest rms. Satellites included in the best-fit combination are colored red. The center of the NGC 147 group is marked with a circle, and lies on the best-fit plane in every case. The three members of this group are colored orange. Only the best-fit satellite positions are considered for these plots. The right-hand column shows the corresponding pole-density plot for the poles of all satellite combinations. These plots have been weighted by the rms of each pole and fully account for the uncertainty in the satellite positions.
(A color version of this figure is available in the online journal.)
combinations as to make this result not particularly important. It should also be noted that these best-fit planes trace out the same approximate great circle as the prominent plane indicated in Figure 1, a result that shall be investigated a little later in Section 3.4. It is particularly interesting that the pole shared by each of these planes, located at $l_{\mathrm{M} 31} \approx-80^{\circ}, b_{\mathrm{M} 31} \approx 40^{\circ}$ corresponds to a pole count maximum in each of the pole plots. This indicates that many of the satellite combinations are aligned along this plane, hence further suggesting that the plane applies to more satellites than the combination sizes tested here. The other, lower latitude principle maximum in the pole plots is that corresponding approximately to the best fit to all the satellites and hence it grows more prominent in the plots made from larger combination sizes as discussed earlier.

Besides the pole count maxima that are strongly indicative of a highly planar subset of satellites, the other principle feature of the pole plots in Figure 10 is the great circle along which the pole count density is highest. This great circle is very prominent but great caution must be exercised in attributing any significance to it. It is centered on the Milky Way, indicating that the constituent poles result from a majority of satellites lying along the Earth to M31 line of site. But this reflects the anisotropy predicted from Figure 2, the result of the bias incurred by the finite area of the PAndAS survey. Hence it would seem that the progenitor of this prominent great circle is not physical but rather the result of selection effects. To further investigate the significance of the patterns observed in the pole plots, 1000 random realizations of 25 satellites were generated as per Section 2.2, and a similar pole count analysis performed on each of them. Specifically, the pole density distribution resulting from the best-fit planes to all combinations of five satellites was generated for each of them. The resulting pole plots for 8 of the 1000 random realizations (chosen at random) are presented in Figure 11 along with an enlarged version of the equivalent plot from Figure 10 generated from the real distribution. A bias toward a similar high-density great circle is indeed observed in these plots, but the plot generated from the actual data features a conspicuously narrower great circle, and a much more constrained distribution in general. This appears to be primarily the result of the large fraction of satellites that lie along the prominent plane that is repeatedly identified and plotted in the left-hand column of Figure 10. It should also be noted that this plane, while being oriented perfectly edge-on with respect to the Earth, contains a significant fraction of satellites lying well outside the region of the M31 sky where the detection bias is large, and hence it is unlikely that its prominence is due to our observational constraints.

Figure 12 provides for a comparison between the concentration of poles around the principle maximum in the pole distributions of the actual satellite distribution and the average of the 1000 random satellite distributions. From line (a) in Figure 12 we see that $21.5 \%$ of all combinations of the actual satellite positions are fitted by a best-fit plane with pole within $15^{\circ}$ of the principal maximum (located at $l_{\mathrm{M} 31}=-78.7, b_{\mathrm{M} 31}=38.4$ ). This is in stark contrast to the $12.0 \%$ that lie within $15^{\circ}$ of the principal maximum for the average random realization of satellite positions (Figure 12, line (b)). Furthermore, we find that only 117 of the 1000 random realizations exhibited the degree of concentration of poles within $15^{\circ}$ of the principal maximum that was observed for the actual satellite distribution. Hence it would seem that a large percentage of satellite combinations are fitted by best-fit planes that all have strikingly similar orientations when compared with what one could expect from a
random distribution of satellites. Again, this points toward a significant plane of satellites that includes a large fraction of the whole satellite sample.

In order to obtain a better understanding of the satellites that this plane consists of, it is of particular interest to explore the number of times each satellite is included in a combination that is best fit by a plane with pole in close proximity to the principal maximum in the pole distribution for the entire sample. Once again, we use the pole distribution for all combinations of five satellites, and we count the number of times each satellite contributes to a pole within $3^{\circ}$ of the principal maximum at $l_{\mathrm{M} 31}=-78^{\circ} .7, b_{\mathrm{M} 31}=38^{\circ} .4$. The counts are divided by 100 to account for the 100 samples that are taken of each combination. The result can be seen in Figure 13. From this figure, it can be seen that the main contributors to the principal maximum in pole counts are those same satellites identified as forming a prominent plane in Figure 1, namely Andromedas I, XI, XII, XIII, XIV, XVI, XVII, XXV, XXVI, XXVII, and the NGC 147 group, along with Andromeda III and Andromeda IX. Hence the conclusion of our analysis thus far must be that there is indeed a significant plane in the satellite distribution of M31 and that it broadly consists of the aforesaid satellites. We therefore investigate the numerical significance of the best-fit plane to these satellites in Section 3.4. As yet there is still more to be gleaned from a study of the pole density distribution, however.

From Figure 13 we have been able to determine the principle contributing satellites to the principal maximum in the pole density distribution, but what of the remaining satellites? Do the positions of these satellites follow any particular trend? The best way to determine this is to construct pole-density plots of the two halves of the complete sample, namely the major contributors to the principal maximum and the minor contributors. The resulting pole plots are presented in Figure 14.

The left-hand plot of Figure 14 shows the pole density distribution generated from the major contributing satellites to the principal maximum at $l_{\mathrm{M} 31}=-78.7, b_{\mathrm{M} 31}=38.4$. This half-sample includes Andromedas I, III, IX, XI, XII, XIII, XIV, XVI, XVII, XXV, XXVI, XXVII, and the NGC 147 group. As expected, this plot reflects the existence of the aforementioned plane with all combination poles lying in the vicinity of the principal maximum. The right-hand plot, with poles generated from the remaining 12 satellites, namely Andromedas II, V, X, XV, XVIII, XIX, XX, XXI, XXII, XXIII, XXIV, and M33, paints a very different picture, however. There is a much greater spread in the distribution of poles, with the great circle induced by the survey area bias once again conspicuous. Also prominent in this plot are two density maxima with their corresponding mirror images in the opposite hemisphere. The maximum lying midway between Andromedas XIX and XX lies very close to the pole of maximum detection bias at $l_{\mathrm{M} 31}=-90^{\circ}, b_{\mathrm{M} 31}=0^{\circ}$ and so it is not unexpected, now that the prominent plane of satellites is effectively removed from the distribution. The elongated maximum passing through $l_{\mathrm{M} 31} \approx 45^{\circ}, b_{\mathrm{M} 31} \approx 45^{\circ}$ is more interesting, however, and suggests the possibility of a second plane, roughly orthogonal to the major plane represented in the left-hand plot, though much less conspicuous. The planes represented by this maximum pass close to the error trails on the M31 sky of Andromedas II, III, XIX, XX, XXIII, and XXIV. This maximum is faintly discernible in the pole distribution for combinations of six satellites presented in Figure 10 but is no more pronounced than anywhere else along the high-density great circle in any of the other pole plots. On account of this, it would appear that this plane is likely no more significant than


Figure 11. Pole density maps for eight random realizations of 25 satellites. The maps plot the poles for the best-fit planes to all combinations of five satellites. The contribution of each pole is weighted by the rms of the plane it represents. The map resulting from all combinations of five satellites drawn from the real data is shown again at the top for comparison. Great circle overdensities are evident to varying degrees in the plots and are a result of the satellite detection bias induced by the finite PAndAS survey area. Note that the density scale for the random realizations is much smaller than for the real data on account of the many realizations utilized for each combination of satellites from the real data.
(A color version of this figure is available in the online journal.)


Figure 12. Radial density profiles showing the percentage of all poles lying within $n$ degrees of the densest point in the pole count distributions (the principal maximum) for (a) the actual satellite distribution and (b) the average of 1000 random satellite distributions. The profile for the actual satellite distribution is generated from the same pole distribution as illustrated for five satellites in Figure 10 and at the top of Figure 11. Note that the relative linearity of (b) compared with (a) is simply a result of the averaging of a large number of individual profiles undertaken to produce the former.
one would expect to find from a random satellite distribution subject to the same detection biases, such as those illustrated in Figure 11.

### 3.4. A Great Plane of Satellites

Throughout the investigation undertaken thus far, all evidence has repeatedly pointed toward a conspicuously planar subset of satellites consisting of roughly half the total sample of satellites. Andromedas I, XI, XII, XIII, XIV, XVI, XVII, XXV, XXVI, XXVII, and XXX as well as the dwarf ellipticals NGC 147 and NGC 185 all appeared to lie along a plane in Figure 1. The reality of this co-planarity was verified in Section 3.3 and in particular Figure 13, which also suggested that Andromeda III and Andromeda IX should be considered as plane members. Hence it is of great interest to ascertain whether this "great plane" is in fact significant. To do this, it is necessary to determine how likely such a plane is to arise from a random satellite distribution subject to the same selection biases. The plane itself and the satellites of which it is constituted are illustrated in Figure 15. The plane shown is that calculated from the best-fit satellite positions and has equation of the form: $0.158 x+0.769 y+0.620 z=0$ with pole at $\left(l_{\mathrm{M} 31}, b_{\mathrm{M} 31}\right)=(-78.4,38.3)$. Note that for this section, we reinstate NGC 147, NGC 185, and Andromeda XXX as separate objects since we are again concerned with measurements of the significance of the planarity of the distribution. Our "great plane" thus consists of 15 satellites out of the entire sample of 27.

Using the method of Section 2.2, we again generate 10,000 independent random realizations of 27 satellites and seek the most planar combination of 15 satellites from each. For each random realization, we sample 1000 possible positions for each satellite as in previous sections and take the average value for the rms of the best-fit plane through the most planar combination. Since there are more than 17 million ways that 15 satellites can be drawn from 27, and since we are not concerned with the orientation of each fitted plane as we have been in all previous sections, we depart from the plane-fitting method of Section 2.1 for this section and instead proceed as follows. For each sample


Figure 13. Histogram showing the relative contribution of each satellite to the pole density within $3^{\circ}$ of the principal maximum at $l_{\mathrm{M} 31}=-78^{\circ} .7, b_{\mathrm{M} 31}=$ 38.4. The histogram is generated from the same pole distribution as illustrated for five satellites in Figure 10 and at the top of Figure 11.
of satellite positions from each realization, 10,000 randomized planes are generated and the 15 closest satellites of the 27 to the plane are stored in each case and the rms recorded. The lowest rms achieved is hence taken to be that for the most planar combination of 15 satellites in the sample. These minimum rms values from each of the 1000 samples of the particular random realization are then averaged to provide the best representation for the realization, given the distance uncertainties. Figure 16 provides probability distributions in the rms for the observed "great plane" (a) together with those for the average rms for the most planar combination from each random realization $(b)$. The average rms for the observed plane is plotted in (b) for comparison.
As can be seen from Figure 16, the rms for the observed plane is very low compared to what one could reasonably expect from a chance alignment. Indeed, the average rms of 12.58 kpc for the observed plane is found to be equaled or exceeded in only 36 out of the 10,000 random realizations. The chances of obtaining such a planar group of 15 satellites from a sample of 27 at random is thus estimated as $0.36 \%$. Hence we can conclude from this test that the observed plane is very unlikely to be a chance alignment, but rather the result of some underlying physical mechanism. Note that an independent but equivalent investigation is presented in ILC13 where such an alignment is found to occur in only $0.15 \%$ of instances. This is due to the larger central obstruction adopted in that analysis (19.6 versus $7.9 \mathrm{deg}^{2}$ ) which rejects more satellites in close proximity to the plane pivot point (M31) where small plane distances are most likely.

## 4. DISCUSSION

Throughout the analysis conducted in Section 3, the presence of a prominent plane of satellites has been a consistent feature. This is not the first time that a significant plane of satellites has been identified from among the denizens of the M31 halo however. As early as 1995, Fusi Pecci et al. (1995) had noted that the majority of the then known satellites lay conspicuously in a plane oriented edge-on with respect to the Milky Way. Koch \& Grebel (2006) identified a highly significant plane lying within $5^{\circ}-7^{\circ}$ of being polar. Furthermore, they identify a subset of nine


Figure 14. Pole density distributions generated from all combinations of five satellites possible from: (left) the satellites contributing significantly to the principal maximum at $l_{\mathrm{M} 31}=-78.7, b_{\mathrm{M} 31}=38.4$ as per Figure 13 and (right) the remaining 12 satellites.
(A color version of this figure is available in the online journal.)
satellites from this plane lying within a thin disk with an rms of 16 kpc . Metz et al. (2007) and later Metz et al. (2009a) similarly identify a disk of satellites, this time not so markedly polar, with pole (in our coordinate system) at $\left(l_{\mathrm{M} 31}, b_{\mathrm{M} 31}\right)=\left(-70.2,32^{\circ} .9\right)$. They find this disk to have an rms height of 39.2 kpc . This disk is clearly the same structure that we identify here, being tilted by only 8.6 with respect to our "great plane." Our plane is found to have a much smaller rms of just $12.34_{-0.43}^{+0.75} \mathrm{kpc}$, however, despite including a comparable number of satellites. It is particularly noteworthy, however, that their satellite sample is significantly different from that used here, with their disk including M32, NGC 205, IC10, LGS3, and IC1613-all of which lie outside the portion of the PAndAS survey region used in this study (see Figure 10(c) of CIL12). Indeed, it is clear from Figure 4 of McConnachie \& Irwin (2006) that the galaxies M32, IC10, LGS3, and IC1613 all lie along the same great circle as our "great plane" in Figure 15, as do their entire error trails. Their conformity along with Andromeda I to a thin disk is noted in the said paper as one of eight possible "streams of satellites," thus providing another early detection of the plane identified by this study. Both Majewski et al. (2007) and Irwin et al. (2008) also draw attention to the linear distribution of many of the plane-member satellites on the sky, a consequence of the edgeon orientation of the plane as indicated by the present study. The plane of Metz et al. (2009a) does however include a significant number of satellites that, while included in our sample, we exclude due to their looser association with our plane. This then accounts for the much smaller rms height observed in our study.

Unlike previous studies of the M31 satellite system, we have a significant advantage in this study on account of the greatly improved sample of satellites available to us. Our sample is not only more numerous, but the positions are all determined via the same method applied to the same data as per CLI11 and CIL12. We are thus afforded unprecedented knowledge of the satellite detection biases, as well as the uncertainties in the object positions, and have factored this knowledge into the analysis. An understanding of this bias is of particular importance when it comes to ascertaining the significance of any substructure identified, since a physically homogeneous satellite distribution will inevitably appear anisotropic after "folding in" the selection function and it is important that we do not attribute physical significance to this anisotropy.

Even after taking these effects into account, however, there can be little doubt that the plane described in Section 3.4 is a real physical object. The component satellites extend well into the regions of low detection bias in Figure 2 and the analysis of the last section makes it clear that such a thin disk


Figure 15. A Great Plane of Satellites consisting of Andromedas I, III, IX, XI, XII, XIII, XIV, XVI, XVII, XXV, XXVI, XXVII, XXX, NGC 147, and NGC 185. The plane shown is that derived from the best-fit satellite positions. The pole is located at $\left(l_{\mathrm{M} 31}, b_{\mathrm{M} 31}\right)=(-78.4,38.3)$.
(A color version of this figure is available in the online journal.)
of satellites has very little chance of arising within a random satellite distribution of the same size, even when subject to the same observation biases. Furthermore, it should be noted that the study of the plane's significance in Section 3.4 is likely to be conservative, given that if the satellites M32, IC10, LGS3, IC1613, and NGC 205 were to be included in the analysis, the significance of our observed plane would likely grow still further. What is also particularly interesting is that subsequent research has shown 13 of the 15 objects to be co-rotating. This result is discussed in more detail in ILC13.
What then could be the progenitor of this "great plane"? The polar orientation one might expect to arise had the satellites formed within the dark matter halo or had the dynamical friction proposed by Quinn \& Goodman (1986) had sufficient time to take effect is not observed. Similarly, the findings of Metz et al. (2009b) seemingly preclude the possibility that the structure might be the result of the accretion of an external galactic association. Furthermore, there is apparently no marked distinction in the metallicities of the disk members compared with the non-disk members as one might expect from this scenario, though it is possible that some of the non-disk satellites may share the same origin as the disk members. There remains, however, the possibility that the satellites trace out the tidal debris of a galaxy merger. This is a particularly interesting possibility, especially since the plane, when projected onto the M31 tangent plane, is in close alignment with the Giant Stellar Stream. Indeed, Hammer et al. (2010) show that the Giant Stellar Stream could feasibly be the product of a major merger event


Figure 16. Determining the significance of the observed "great plane" of satellites (see Figure 15). Figure (a) gives the distribution of possible values of the rms obtainable from 200,000 realizations of possible positions of the 15 plane members, given their respective distance probability distributions. Figure (b) plots the average rms of the best-fit plane through the most planar combination of 15 satellites for each of 10,000 random realizations of 27 satellites. These satellites are subject to the same selection biases as the real data. As for Figure 5, histogram (b) should only be compared with the average of histogram (a), which is plotted in (b) as a dashed line. It is thus clear that the planarity observed for our "great plane" of satellites is very unlikely to arise by chance. The $1 \sigma(68.2 \%)$, $90 \%$, and $99 \%$ credibility intervals are shown as red, green, and blue lines, respectively.
(A color version of this figure is available in the online journal.)
that began around 9 Gyr ago, sustained by the returning stars from a tidal tail oriented similarly to our "great plane."

If a link is to be established, the observed asymmetry of the system must also be compatible with the tidal scenario however. It is of particular interest that, of the 13 co-rotating satellites in the plane, all but one lie on the near side of the M31 tangent plane. Indeed, if we removed all of the plane membersatellites from the system, the remaining satellite distribution would no longer be significantly asymmetric. With almost all of the satellites currently on the near side of M31, it would seem on first consideration that the progenitor event could not have occurred substantially more than a typical orbital time ago or else the satellites would have had sufficient time to disperse. This suggests the event responsible must have occurred within the last 5 Gyr. Other studies, however, have supported the proposition that a group of tidal dwarf galaxies could survive for extended periods while retaining the asymmetry inherent from
the time of their formation. For instance, Duc et al. (2011) find three moderately old tidal dwarf candidates with this asymmetry preserved. It must also be remembered that the precise orbits of the satellites are undetermined and so it is not clear how close they have come to M31 in the past. As discussed by Pawlowski et al. (2011), there is also evidence that tidal material can survive in a "bridge" between the interacting galaxies for an extended period. Nevertheless, it must be cautioned that there is no established precedent for tidal dwarfs with the longevity implied by the stellar populations of the M31 satellites, and the fact that they continue to adhere to such a thin structure is even more perplexing under such a scenario.

There is also another striking characteristic of the observed plane. As one will note from examination of Figure 15 (and indeed the left-hand column of plots in Figure 10), it is oriented perfectly edge-on with respect to the Milky Way. While there is a noted bias toward detection of satellites positioned along planes oriented in this way, it must be remembered that this bias arises primarily due to the propensity for detecting satellites close to the line of sight passing through M31. Many of the satellites observed to lie on our plane are located a good distance from this line of sight, however, and well into the low-bias portions of the M31 sky. In any case, the random realizations of Section 3.4 suffer from the same biases and yet show unequivocally that the observed plane is very unlikely to arise by chance. Hence if we are to accept these results, we must also accept the plane's orientation.

Further to this strikingly edge-on orientation, it is also noteworthy that the plane is approximately perpendicular to the Milky Way disk. This fact can be easily seen if the constituent satellites are traced out in Galactic coordinates (i.e., all lie on approximately the same Galactic longitude). This of course raises the question-how does the orientation of the Milky Way's polar plane of satellites compare with this plane? Noting that the average pole of the "Vast Polar Structure" described by Pawlowski et al. (2012b) points roughly in the direction of M31, the two planes are approximately orthogonal. These precise alignments are discussed in more detail in ILC13, but suffice to say here that this alignment is particularly interesting and suggests that the Milky Way and M31 halos should not necessarily be viewed as fully isolated structures. It is entirely conceivable that our current ignorance as to the coupling between such structures may be to blame for our inability to pin down the precise mechanism by which such planes arise.

Finally, in consequence of these findings, most particularly with respect to the highly significant, very thin disk of satellites that has been identified, it is clear that if $\Lambda C D M$ is to remain the standard model of cosmology, the occurrence of such structures has to be explicable within it. The possible deeper implications of the satellite anisotropies are discussed in Kroupa et al. (2010), Angus et al. (2011), Fouquet et al. (2012), and Kroupa (2012), wherein other alternative cosmologies are also highlighted.

## 5. CONCLUSIONS

It is clear that while the satellites of M31 when taken as a whole are no more planar than one can expect from a random distribution, a subset consisting of roughly half the sample is remarkably planar. The presence of this thin disk of satellites has been conspicuous throughout the analysis contained in this paper. The degree of asymmetry determined from the satellite distribution is also found to be relatively high. Of particular note, the orientation of the asymmetry is very significant, being aligned very strongly in the direction of the Milky Way. When
this fact is combined with the apparent orthogonality observed between the Milky Way and M31 satellite distributions and the Milky Way disk, it appears that the two halos may in fact be coupled. Regardless, the great plane of satellites identified in this study, and its clear degree of significance, is not directly expected from $\Lambda C D M$ cosmology. This finding provides strong evidence that thin disks of satellites do indeed exist in galaxy halos, and whether or not the standard model can account for such structures remains to be seen.
A.R.C. thanks Macquarie University for their financial support through the Macquarie University Research Excellence Scholarship (MQRES) and both the University of Sydney and Université de Strasbourg for the use of computational and other facilities. G.F.L. thanks the Australian Research Council for support through his Future Fellowship (FT100100268) and Discovery Project (DP110100678). R.A.I. and D.V.G. gratefully acknowledge support from the Agence Nationale de la Recherche though the grant POMMME (ANR 09-BLAN-0228). Based on observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-France-Hawaii Telescope (CFHT), which is operated by the National Research Council (NRC) of Canada, the Institut National des Science de l'Univers of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii.

## REFERENCES

Angus, G. W., Diaferio, A., \& Kroupa, P. 2011, MNRAS, 416, 1401 Conn, A. R., Ibata, R. A., Lewis, G. F., et al. 2012, ApJ, 758, 11 (CIL12) Conn, A. R., Lewis, G. F., Ibata, R. A., et al. 2011, ApJ, 740, 69 (CLI11) de Vaucouleurs, G. 1958, ApJ, 128, 465
de Vaucouleurs, G., de Vaucouleurs, A., Corwin, H. G., Jr., et al. 1991, Third Reference Catalogue of Bright Galaxies (Berlin: Springer)

Duc, P.-A., Cuillandre, J.-C., Serra, P., et al. 2011, MNRAS, 417, 863
Fouquet, S., Hammer, F., Yang, Y., Puech, M., \& Flores, H. 2012, in SF2A-2012 Proc. Annual Meeting of the French Society of Astronomy and Astrophysics, ed. S. Boissier et al. (Paris: SF2A), 333
Fusi Pecci, F., Bellazzini, M., Cacciari, C., \& Ferraro, F. R. 1995, AJ, 110, 1664
Hammer, F., Yang, Y. B., Wang, J. L., et al. 2010, ApJ, 725, 542
Hartwick, F. D. A. 2000, AJ, 119, 2248
Holmberg, E. 1969, ArA, 5, 305
Ibata, R. A., Lewis, G. F., Conn, A. R., et al. 2013, Natur, 493, 62 (ILC13)
Irwin, M. J., Ferguson, A. M. N., Huxor, A. P., et al. 2008, ApJL, 676, L17
Knebe, A., Gill, S. P. D., Gibson, B. K., et al. 2004, ApJ, 603, 7
Koch, A., \& Grebel, E. K. 2006, AJ, 131, 1405
Kroupa, P. 2012, PASA, 29, 395
Kroupa, P., Famaey, B., de Boer, K. S., et al. 2010, A\&A, 523, A32
Kroupa, P., Theis, C., \& Boily, C. M. 2005, A\&A, 431, 517
Libeskind, N. I., Frenk, C. S., Cole, S., et al. 2005, MNRAS, 363, 146
Lovell, M. R., Eke, V. R., Frenk, C. S., \& Jenkins, A. 2011, MNRAS, 413, 3013
Lynden-Bell, D. 1976, MNRAS, 174, 695
Lynden-Bell, D. 1982, Obs, 102, 202
Lynden-Bell, D., \& Lynden-Bell, R. M. 1995, MNRAS, 275, 429
Majewski, S. R., Beaton, R. L., Patterson, R. J., et al. 2007, ApJL, 670, L9
McConnachie, A. W., \& Irwin, M. J. 2006, MNRAS, 365, 902
McConnachie, A. W., Irwin, M. J., Ibata, R. A., et al. 2009, Natur, 461, 66 Metz, M., Kroupa, P., \& Jerjen, H. 2007, MNRAS, 374, 1125
Metz, M., Kroupa, P., \& Jerjen, H. 2009a, MNRAS, 394, 2223
Metz, M., Kroupa, P., \& Libeskind, N. I. 2008, ApJ, 680, 287
Metz, M., Kroupa, P., Theis, C., Hensler, G., \& Jerjen, H. 2009b, ApJ, 697, 269
Navarro, J. F., Abadi, M. G., \& Steinmetz, M. 2004, ApJL, 613, L41
Palma, C., Majewski, S. R., \& Johnston, K. V. 2002, ApJ, 564, 736
Pawlowski, M. S., Kroupa, P., Angus, G., et al. 2012a, MNRAS, 424, 80 Pawlowski, M. S., Kroupa, P., \& de Boer, K. S. 2011, A\&A, 532, A118
Pawlowski, M. S., Pflamm-Altenburg, J., \& Kroupa, P. 2012b, MNRAS, 423, 1109
Quinn, P. J., \& Goodman, J. 1986, ApJ, 309, 472
Springel, V., Wang, J., Vogelsberger, M., et al. 2008, MNRAS, 391, 1685
Starkenburg, E., Helmi, A., De Lucia, G., et al. 2013, MNRAS, 429, 725
Vera-Ciro, C. A., Sales, L. V., Helmi, A., et al. 2011, MNRAS, 416, 1377
Zentner, A. R., Kravtsov, A. V., Gnedin, O. Y., \& Klypin, A. A. 2005, ApJ, 629, 219

## 6

## Conclusions

The contribution to the field of galactic archaeology embodied in this work has essentially been twofold. Firstly, a robust new technique has been developed for ascertaining distances via the tip of the red giant branch, a technique which is stand-alone in terms of its diverse applicability. Secondly, this technique has been applied to the satellite system of M31 to reveal an inhomogeneous structure which is somewhat at odds with our current understanding of galaxy formation. The key outcomes of the thesis are summarized as follows:

I A powerful new Bayesian technique has been developed for determining the distance probability distribution of an object from the tip of its red giant branch. The technique is best suited to older, metal poor structures that are sufficiently close as to facilitate accurate photometric measurements to a depth exceeding that of the RGB tip by at least 0.5 magnitudes.

II A 'density' matched-filter has been developed to complement the technique of I. This matched filter was developed specifically for the satellite galaxies of M31 and as such, is not applicable to extended structures (such as streams). It effectively acts to improve the contrast of the RGB tip in the object's luminosity function by weighting the component stars with respect to their position within the object's density profile.

III An angle-specific density prior has been devised specifically for the M31 halo and incorporated into the technique of $\mathbf{I}$. It effectively equates each position along the line of sight to one of the satellites with some probability, based on the assumed sub-halo density at the associated radius from the center of the halo.

IV Accurate distance probability distributions have been obtained for 27 of the satellites of M31 as well as for M31 itself via incorporating the priors of II and III into the technique of $\mathbf{I}$.

V The distance distributions of IV have been converted into M31-centric 3D positions, providing the largest homogeneous sample of satellite galaxy positions for any galaxy halo.

VI The M31 satellite distribution has been found to be approximately isothermal. When the 15 most Gaussian distance distributions are considered, the satellite density profile is found to follow a power law with $\rho(r) \propto r^{-\alpha}$ where $\alpha=1.87_{-0.42}^{+0.46}$.

VII The satellite distribution as a whole has been shown to be no more planar than one would expect from a random distribution of points.

VIII A large subset of the satellites, 15 out of the total sample of 27 , has been found to be remarkably planar, with a root-mean-square thickness of just $12.34_{-0.43}^{+0.75} \mathrm{kpc}$. The probability of obtaining such a large, thin structure in a random distribution of equal size is found to be only $0.36 \%$. The orientation of this plane is intriguing. It is found to lie almost perfectly edge-on with respect to the Milky Way, and approximately perpendicular to the Milky Way disk. It is also roughly orthogonal to the planar distribution of satellites regularly reported for the Milky Way, and 13 of the 15 satellites have subsequently been identified as co-rotating (see Ibata et al. 2013 and Appendix A).

IX The asymmetry of the distribution as a whole has been shown to be considerably larger than one would expect by chance. After factoring in the uncertainty in the satellite positions, it is most likely that the sample can be divided such that 23 of the 27 satellites all lie in a single hemisphere. The probability of the observed asymmetry arising in a random distribution is $4.22 \%$.

X The asymmetry about the M31 tangent plane has been found to be particularly high, with 20 of the 27 satellites most likely lying on the Milky Way side. The probability of the observed degree of asymmetry about this plane arising by chance is just $0.46 \%$. It is noteworthy that if the 15 satellites belonging to the plane of VIII are omitted, the asymmetry about the M31 tangent plane is no longer significant.

In light of the above outcomes, there are several avenues of future investigation that warrant attention. The first concerns the future application of the RGB tip finding technique in its current form. The PAndAS survey region is awash with the relics of past accretion events and is the obvious starting place. There are many streams of stars that are well within reach of the technique. Furthermore, it should be possible to divide most of these streams into segments and obtain distance measurements to each individually. The result would be the effective conversion of the key structures of the PAndAS survey into a three dimensional network of streams and interspersed satellite galaxies. This would facilitate a study of unprecedented scale into the distribution of mass within the M31 halo.

The RGB tip finding technique is of course also readily applicable to the denizens of the Milky Way halo. There are more than 25 satellite galaxies and more than 150 globular clusters that orbit within the halo of our own galaxy. Distance measurements already exist for almost all of them, but there would be significant advantages to a sample of distances obtained via the systematic application of a single measurement technique.

In addition to the possible future applications of the tip finding technique, there are also a variety of means by which it might be improved. In particular, the method in its current form is most suited to more metal poor structures where there is minimum variation of the i-band tip luminosity with metallicity. The method could be made more versatile by replacing the one-dimensional model of the object's luminosity function with a two-dimensional model
of its CMD. Following this approach, stellar isochrones could be incorporated to model the correct form of the object's red giant branch in colour-magnitude space, making the method more robust in its treatment of more metal rich objects, and those containing more than one stellar population. Along similar lines, one could also apply a matched-filter to the object's CMD. This could be achieved by fitting a 2D surface to the object's CMD and dividing it by that fitted to the CMD of a suitable comparison (background) field, creating an effective 'flat field' tailored to suit the object. Each star in the object's CMD could thus be weighted by its probability of being a true member of the object's red giant branch.

The structure of the M31 satellite distribution as revealed by this study, also presents a number of opportunities to further our understanding of the local universe. The most immediate course to pursue would seem to lie in the application of this knowledge to a new study of the galaxy's mass distribution via modeling of its rotation curve. The plane of corotating satellites (VIII) provides the perfect starting point for such a study. With the plane fortuitously aligned edge-on with respect to the Earth, it will be possible to calculate the tangential component of the satellites' orbital velocities directly from the radial velocities via simple trigonometry. Thus, if the satellites are approximated to follow circular orbits, it will be relatively straightforward to obtain probability distributions for the mass enclosed by each satellite orbit, and in so doing, extend the known M31 rotation curve significantly. It would also be possible to obtain estimates of the enclosed mass for the non-plane members via a maximum likelihood approach, after marginalizing over the two tangential components of the orbital velocity.

More than anything else, this study has highlighted the limitations of existing theories of galaxy formation and evolution. It is very difficult to explain how such a large, thin structure as that identified in VIII can remain intact for any length of time, let alone how it came to exist in the first place. Add to that the bizarre orientation and the high degree of asymmetry, and we are left with an intriguing enigma. The onus then is on unlocking this enigma, for in so doing we shall undoubtedly learn a great deal about galaxy evolution and M31's past, as well as that of our own galaxy.

## An Introduction to the Appendices

The appendices are divided into 5 sections labelled A through E. Appendix A is devoted to the article 'A Vast Thin Plane of Co-rotating Dwarf Galaxies Orbiting the Andromeda Galaxy' published in nature which was written in conjunction with Paper III (Ch 5). A brief introduction to the article can be found on the Appendix A title page. The remaining 4 appendices contain examples of some of the principal code used for the analysis contained in the thesis. The format of these appendices are now discussed.

As a PhD student, perhaps $90 \%$ of my time has been occupied with the development of programs (principally in Fortran) designed to perform the analysis necessary for my research. In this sense, the written component of the thesis really is just the tip of the iceberg, and it therefore seemed both fitting and rather useful to record some of the source code for reference. When I embarked to do this however, I did not realize the shear volume of code I had amassed and subsequently found it necessary to condense the code substantially. The code that is presented in these appendices therefore represents only a fraction of all the programs written during my PhD candidature. Nevertheless, I have endeavored to reproduce here the most important programs and subroutines in as logical a way as possible and with minimum repetition. Each appendix is devoted to code pertinent to a particular chapter, and each program is introduced with a brief description of its purpose and functionality as well as a link to the thesis content to which it relates. Note that some of the programs make a very localized contribution to the material presented in the thesis while others are much broader in scope and may apply to a number of chapters. Many subroutines have also been omitted either to avoid repetition or to remove portions of code which are secondary to the principal
functionality of the parent program. In summary, it is intended that the programs presented in these appendices serve to provide further clarification of the precise way in which the analysis discussed in the thesis has been implemented. The code is not intended for ready implementation on other systems and hence may in many instances require substantial modification to be usable. All code is however well commented and should be reasonably intuitive even for those not well acquainted with Fortran.

## A

## An Article in nature

The work undertaken for Paper III (Ch. 5) quickly revealed a remarkable structure at odds with current models of galaxy formation. Almost half of the 27 satellites studied were found to populate a remarkably thin plane which could be shown to be highly statistically significant. When the radial velocities obtained for the satellites using DEIMOS on Keck were investigated, 13 of the 15 co-planer satellites were found to also be co-rotating. This presents a stark contrast to the predictions of $\Lambda C D M$ whereby the satellites are expected to occupy random orbits. When the intriguing orientation of the plane is considered, along with the high asymmetry in the direction of the Milky Way, the structure is remarkable indeed. The discovery was thus submitted to, and accepted for publication in nature - Ibata et al. (2013). We were very fortunate (owing largely to the beautiful astrophotography of Jean-Charles Cuillandre) to secure the cover of the first issue for 2013 (3rd January). The paper provides a fitting epilogue to the work presented in the main body of the thesis. The final accepted version is reproduced below (note that p15 containing a rotatable 3D figure has been omitted).


Figure A.1: Cover image for the 3 Jan 2013 issue of nature. This image has been reproduced with the permission of the Nature Publishing Group (NPG). This issue is available online at http: //www.nature.com/nature/journal/v493/n7430/.

# A Vast Thin Plane of Co-rotating Dwarf Galaxies Orbiting the Andromeda Galaxy 

Rodrigo A. Ibata ${ }^{1}$, Geraint F. Lewis ${ }^{2}$, Anthony R. Conn ${ }^{3}$, Michael J. Irwin ${ }^{4}$, Alan W. McConnachie ${ }^{5}$, Scott C. Chapman ${ }^{6}$, Michelle L. Collins ${ }^{7}$, Mark Fardal ${ }^{8}$, Annette M. N. Ferguson ${ }^{9}$, Neil G. Ibata ${ }^{10}$, A. Dougal Mackey ${ }^{11}$, Nicolas F. Martin ${ }^{1,7}$, Julio Navarro ${ }^{12}$, R. Michael Rich ${ }^{13}$, David VallsGabaud ${ }^{14}$, and Lawrence M. Widrow ${ }^{15}$
${ }^{1}$ Observatoire astronomique de Strasbourg, 11, rue de l'Université, F-67000 Strasbourg, France.
${ }^{2}$ Sydney Institute for Astronomy, School of Physics, A28, The University of Sydney, NSW 2006, Australia.
${ }^{3}$ Department of Physics and Astronomy, Macquarie University, NSW 2109, Australia.
${ }^{4}$ Institute of Astronomy, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK.
${ }^{5}$ NRC Herzberg Institute of Astrophysics, 5071 West Saanich Road, Victoria, British Columbia, V9E 2 E7 Canada.
${ }^{6}$ Department of Physics and Atmospheric Science, Dalhousie University, 6310 Coburg Rd., Halifax, NS B3H 4R2 Canada
${ }^{7}$ Max-Planck-Institut für Astronomie, Königstuhl 17, 69117 Heidelberg, Germany.
${ }^{8}$ University of Massachusetts, Department of Astronomy, LGRT 619-E, 710 N. Pleasant Street, Amherst, Massachusetts 01003-9305, USA.
${ }^{9}$ Institute for Astronomy, University of Edinburgh, Royal Observatory, Blackford Hill, Edinburgh EH9 3HJ, UK.
${ }^{10}$ Lycée international des Pontonniers, 1 rue des Pontonniers, F-67000 Strasbourg, France.
${ }^{11}$ The Australian National University, Mount Stromlo Observatory, Cotter Road, Weston Creek, ACT 2611, Australia
${ }^{12}$ Department of Physics and Astronomy, University of Victoria, 3800 Finnerty Road, Victoria, British Columbia, Canada V8P 5C2.
${ }^{13}$ Department of Physics and Astronomy, University of California, Los Angeles, PAB, 430 Portola Plaza, Los Angeles, California 90095-1547, USA.
${ }^{14}$ LERMA, UMR CNRS 8112, Observatoire de Paris, 61 Avenue de l'Observatoire, 75014 Paris, France
${ }^{15}$ Department of Physics, Engineering Physics, and Astronomy Queen's University, Kingston, Ontario, Canada K7L 3N6.

Dwarf satellite galaxies are thought to be the remnants of the population of primordial structures that coalesced to form giant galaxies like the Milky Way ${ }^{1}$. An early analysis ${ }^{2}$ noted that dwarf galaxies may not be isotropically distributed around our Galaxy, as several are correlated with streams of HI emission, and possibly form co-planar groups ${ }^{3}$. These suspicions are supported by recent analyses ${ }^{4-7}$, and it has been claimed ${ }^{7}$ that the apparently planar distribution of satellites is not predicted within standard cosmology ${ }^{8}$, and cannot simply represent a memory of past coherent accretion. However, other studies dispute this conclusion ${ }^{9-11}$. Here we report the existence ( $\mathbf{9 9 . 9 9 8} \%$ significance) of a planar subgroup of satellites in the Andromeda galaxy, comprising $\approx 50 \%$ of the population. The structure is vast: at least $\sim 400 \mathrm{kpc}$ in diameter, but also extremely thin, with a perpendicular scatter $<14.1 \mathrm{kpc}$ ( $99 \%$ confidence). Radial velocity measurements ${ }^{12-15}$ reveal that the satellites in this structure have the same sense of rotation about their host. This finding shows conclusively that substantial numbers of dwarf satellite galaxies share the same dynamical orbital properties and direction of angular momentum, a new insight for our understanding of the origin of these most dark matter dominated of galaxies. Intriguingly, the plane we identify is approximately aligned with the pole of the Milky Way's disk and is co-planar with the Milky Way to Andromeda position vector. The existence of such extensive coherent kinematic structures within the halos of massive galaxies is a fact that must be explained within the framework of galaxy formation and cosmology.

We undertook the Pan-Andromeda Archaeological Survey ${ }^{16}$ (PAndAS) to obtain a largescale panorama of the halo of the Andromeda galaxy (M31), a view that is not available to us for the Milky Way. This Canada-France-Hawaii Telescope survey imaged $\sim 400$ square degrees around M31, the only other giant galaxy in the Local Group apart from the Milky Way. Stellar objects are detected out to a projected distance of $\sim 150 \mathrm{kpc}$ from M31, and $\sim 50 \mathrm{kpc}$ from M33, the most massive satellite of M31. The data reveal a substantial population of dwarf spheroidal galaxies that accompany Andromeda ${ }^{17}$.

The distances to the dwarf galaxies can be estimated by measuring the magnitude of the

Tip of the Red Giant Branch (TRGB) ${ }^{18}$. Improving on earlier methods, we have developed a Bayesian approach that yields the probability distribution function (PDF) for the distance to each individual satellite ${ }^{19}$. In this way we now have access, for the first time, to homogenous distance measurements (typical uncertainties $20-50 \mathrm{kpc}$ ) to the 27 dwarf galaxies (filled circles in Figure 1) visible within the PAndAS survey area $^{20}$, that lie beyond the central $2^{\circ} .5$.

In Figure 2 these distance measurements are used to calculate the sky positions of the homogenous sample of 27 dwarf galaxies as they would appear from the centre of the Andromeda galaxy. Visually, there appears to be a correlation close to a particular great circle (red line): this suggests that there is a plane, centred on M31, around which a subsample of the satellites have small scatter. This is confirmed by the Monte Carlo analysis presented in the Supplementary Information, where we show that the probability of the alignment of the subsample of $n_{\text {sub }}=15$ satellites marked red in Figures 1 and 2 occurring at random is $0.13 \%$ (see Figure S1).

Following this discovery, we sought to investigate whether the sub-sample displayed any kinematic coherence. The radial velocity of each satellite is shown in Figure 3, corrected for the bulk motion of the Andromeda system towards us: what is immediately striking is that 13 of the 15 satellites possess coherent rotational motion, such that the southern satellites are approaching us with respect to M31, while the northern satellites recede away from us with respect to their host galaxy.

The probability that 13 or more out of 15 objects should share the same sense of rotation is $1.4 \%$ (allowing for right-handed or left-handed rotation). Thus the kinematic information confirms the spatial correlation initially suspected from a visual inspection of Figure 2. The total significance of the planar structure is approximately $99.998 \%$.

Thus we conclude that we have detected, with very high confidence, a coherent planar structure of 13 satellites with a root-mean-square (rms) thickness of $12.6 \pm 0.6 \mathrm{kpc}(<14.1 \mathrm{kpc}$ at $99 \%$ confidence), that co-rotate around M31 with a (right-handed) axis of rotation that points
approximately East. The three-dimensional configuration can be assessed visually in Figure 3. The extent of the structure is gigantic, over 400 kpc along the line of sight and nearly 300 kpc North-South. Indeed, since And XIV and Cas II lie at the Southern and Northern limits of the PAndAS survey, respectively, it is quite probable that additional (faint) satellites belonging to this structure are waiting to be found just outside the PAndAS footprint. Although huge in extent, the structure appears to be lopsided, with most satellites populating the side of the halo of M31 nearest the Milky Way. The completeness analysis we have undertaken shows that this configuration is not due to a lowered detection sensitivity at large distance, but reflects a true paucity of satellites in the more distant halo hemisphere ${ }^{21}$.

The existence of this structure had been hinted at in earlier work ${ }^{22}$, thanks to the planar alignment we report on here being viewed nearly edge-on, although the grouping could not be shown to be statistically significant from the information available at that time. Other previouslyclaimed alignments do not match the present plane, although they share some of the member galaxies: the most significant alignment of Koch \& Grebel ${ }^{23}$ has a pole $45^{\circ}$ away from that found here, and the (tentative) poles of the Metz et al. ${ }^{4}$ configuration are $23^{\circ} .4$ away, while those of a later contribution ${ }^{6}$ have poles $34^{\circ} .1$ and $25^{\circ} .4$ distant from ours. Without the increased sample size, reliable three-dimensional positions and radial velocities, and most importantly, a spatially unbiassed selection function resulting from the homogenous panoramic coverage of PAndAS, the nature, properties and conclusive statistical significance of the present structure could not be inferred.

The present detection proves that in some giant galaxies, a significant fraction of the population of dwarf satellite galaxies, in this case $\approx 50 \%$ ( 13 out of 27 over the homogeneouslysurveyed PAndAS area), are aligned in coherent planar structures, sharing the same direction of angular momentum. The Milky Way is the only other giant galaxy where we have access to high-quality three-dimensional positional data, and the existence of a similar structure around our Galaxy is strongly suggested by current data ${ }^{2,5,7}$. The implications for the origin and dynamical history of dwarf galaxies are profound. It also has a strong bearing on the analyses of dark matter in these darkest of galaxies, since one cannot now justifiably assume such objects to
have evolved in dynamical isolation.

Intriguingly, the Milky Way lies within $1^{\circ}$ of the plane reported here, the pole of the plane and the pole of the Milky Way's disk are approximately perpendicular ( $81^{\circ}$ ), and furthermore this plane is approximately perpendicular to the claimed plane of satellites that surrounds the Galaxy (since its pole points approximately towards Andromeda ${ }^{7}$ within the uncertainties). While these alignments may be chance occurrences, it is nevertheless essential information about the structure of the nearby Universe that will have to be taken into account in future simulations aimed at modelling the dynamical formation history of the Local Group.

The formation of this structure around M31 poses a difficult puzzle. For discussion, we envisage two broad classes of possible explanations: accretion or in-situ formation. In either type of model, the small scatter of the satellites out of the plane is a challenge to explain, even though the orbital timescales for the satellites are long ( $\sim 5 \mathrm{Gyr}$ for satellites at 150 kpc ). All the galaxies in the plane are known to have old, evolved, stellar populations, and so in-situ formation would additionally imply that the structure is ancient.

In an accretion scenario, the dynamical coherence points to an origin in a single accretion of a group of dwarf galaxies. However, the spatial extent of the progenitor group would have to be broadly equal to or smaller than the current plane thickness ( $<14.1 \mathrm{kpc}$ ), yet no such groups are known. Interpreting the coherent rotation as a result of our viewing perspective ${ }^{24}$ requires a bulk tangential velocity for the in-falling group of order $\sim 1000 \mathrm{~km} \mathrm{~s}^{-1}$, which seems unphysically high. A further possibility is that we are witnessing accretion along filamentary structures that are fortuitously aligned. In-situ formation may be possible if the planar satellite galaxies formed like tidal-dwarf galaxies (TDG) in ancient gas-rich galaxy mergers ${ }^{7}$, but then the dwarf galaxies should be essentially devoid of dark matter. If the planar M31 dwarfs are dynamically-relaxed, the absence of dark matter would be greatly at odds with inferences from detailed observations ${ }^{25}$ of Milky Way satellites assuming the standard theory of gravity. An alternative possibility is that gas was accreted preferentially onto dark matter sub-halos that were already orbiting in this particular plane, but then the origin of the plane of sub-haloes would still require explanation. We conclude
that it remains to be seen whether galaxy formation models within the current cosmological framework can explain the existence of this vast, thin, rotating structure of dwarf galaxies within the halo of our nearest giant galactic neighbour.

1. Klypin, A., Kravtsov, A. V., Valenzuela, O. \& Prada, F. Where Are the Missing Galactic Satellites? Astrophysical Journal 522, 82-92 (1999).
2. Lynden-Bell, D. Dwarf galaxies and globular clusters in high velocity hydrogen streams. Monthly Notices of the Royal Astronomical Society 174, 695-710 (1976).
3. Lynden-Bell, D. \& Lynden-Bell, R. M. Ghostly streams from the formation of the Galaxy's halo. Monthly Notices of the Royal Astronomical Society 275, 429-442 (1995).
4. Metz, M., Kroupa, P. \& Jerjen, H. The spatial distribution of the Milky Way and Andromeda satellite galaxies. Monthly Notices of the Royal Astronomical Society 374, 1125-1145 (2007). arXiv:astro-ph/0610933.
5. Metz, M., Kroupa, P. \& Libeskind, N. I. The Orbital Poles of Milky Way Satellite Galaxies: A Rotationally Supported Disk of Satellites. Astrophysical Journal 680, 287-294 (2008). 0802.3899.
6. Metz, M., Kroupa, P. \& Jerjen, H. Discs of satellites: the new dwarf spheroidals. Monthly Notices of the Royal Astronomical Society 394, 2223-2228 (2009).
7. Pawlowski, M. S., Pflamm-Altenburg, J. \& Kroupa, P. The VPOS: a vast polar structure of satellite galaxies, globular clusters and streams around the Milky Way. Monthly Notices of the Royal Astronomical Society 423, 1109-1126 (2012).
8. Komatsu, E. et al. Seven-year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Cosmological Interpretation. Astrophysical Journal Supplement 192, 18 (2011).
9. Zentner, A. R., Kravtsov, A. V., Gnedin, O. Y. \& Klypin, A. A. The Anisotropic Distribution of Galactic Satellites. Astrophysical Journal 629, 219-232 (2005). arXiv:astro-ph/ 0502496.
10. Lovell, M. R., Eke, V. R., Frenk, C. S. \& Jenkins, A. The link between galactic satellite orbits and subhalo accretion. Monthly Notices of the Royal Astronomical Society 413, 3013-3021 (2011). 1008.0484.
11. Wang, J., Frenk, C. S., Navarro, J. F., Gao, L. \& Sawala, T. The missing massive satellites of the Milky Way. Monthly Notices of the Royal Astronomical Society 424, 2715-2721 (2012). 1203.4097.
12. Collins, M. L. M. et al. The scatter about the 'Universal' dwarf spheroidal mass profile: a kinematic study of the M31 satellites And V and And VI. Monthly Notices of the Royal Astronomical Society 417, 1170-1182 (2011).
13. Tollerud, E. J. et al. The SPLASH Survey: Spectroscopy of 15 M31 Dwarf Spheroidal Satellite Galaxies. Astrophysical Journal 752, 45 (2012).
14. McConnachie, A. W. The Observed Properties of Dwarf Galaxies in and around the Local Group. Astronomical Journal 144, 4 (2012).
15. Collins, M. L. et al. The non-universal dSpH mass profile? - A kinematic study of the Andromeda dwarf spheroidal system. Monthly Notices of the Royal Astronomical Society (2012). Submitted.
16. McConnachie, A. W. et al. The remnants of galaxy formation from a panoramic survey of the region around M31. Nature 461, 66-69 (2009).
17. Richardson, J. C. et al. PAndAS' Progeny: Extending the M31 Dwarf Galaxy Cabal. Astrophysical Journal 732, 76 (2011).
18. Lee, M. G., Freedman, W. L. \& Madore, B. F. The Tip of the Red Giant Branch as a Distance Indicator for Resolved Galaxies. Astrophysical Journal 417, 553-559 (1993).
19. Conn, A. R. et al. A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. I. Astrophysical Journal 740, 69 (2011).
20. Conn, A. R. et al. A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. II. Distances to the Satellites of M31. Astrophysical Journal 758, 11 (2012). 1209.4952.
21. McConnachie, A. W. \& Irwin, M. J. The satellite distribution of M31. Monthly Notices of the Royal Astronomical Society 365, 902-914 (2006). arXiv: astro-ph/05 10654.
22. Majewski, S. R. et al. Discovery of Andromeda XIV: A Dwarf Spheroidal Dynamical Rogue in the Local Group? Astrophysical Journal Letters 670, L9-L12 (2007). arXiv:astro-ph/ 0702635.
23. Koch, A. \& Grebel, E. K. The Anisotropic Distribution of M31 Satellite Galaxies: A Polar Great Plane of Early-type Companions. Astronomical Journal 131, 1405-1415 (2006). arXiv:astro-ph/0509258.
24. van der Marel, R. P. \& Guhathakurta, P. M31 Transverse Velocity and Local Group Mass from Satellite Kinematics. Astrophysical Journal 678, 187-199 (2008). 0709.3747.
25. Walker, M. G. Dark Matter in the Milky Way's Dwarf Spheroidal Satellites. ArXiv e-prints (2012). 1205.0311.
26. Barnes, D. G., Fluke, C. J., Bourke, P. D. \& Parry, O. T. An Advanced, Three-Dimensional Plotting Library for Astronomy. Publications of the Astronomical Society of Australia 23, 82-93 (2006). arXiv:astro-ph/0608245.

Acknowledgements We thank the staff of the Canada-France-Hawaii Telescope for taking the PAndAS data, and for their continued support throughout the project. We would like to thank one of our referees, Prof. B. Tully, for pointing out that IC 1613 could also be associated to the planar structure. R.A.I. and D.V.G. gratefully acknowledges support from the Agence Nationale de la Recherche though the grant POMMME, and would like to thank Benoit Famaey for insightful discussions. G.F.L thanks the Australian research council for support through his Future Fellowship and Discovery Project. Based on observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA/DAPNIA, at the Canada-France-Hawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institut National des Sciences de l'Univers of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii. Some of the data presented herein were obtained at the W.M. Keck Observatory, which is operated as a scientific partnership among the California Institute of Technology, the University of California and the National Aeronautics and Space Administration. The Observatory was made possible by the generous financial support of the W.M. Keck Foundation.

Competing Interests The authors have no competing financial interests.

Author contributions All authors assisted in the development and writing of the paper. In addition, the structural and kinematic properties of the dwarf population, and the significance of the Andromeda plane were determined by R.A.I., G.F.L. and A.R.C., based on distances determined by the same group (as part of the PhD research of A.R.C.). In addition, A.W.M. is the Principal Investigator of PAndAS; M.J.I. and R.A.I led the data processing effort; R.A.I. was the Principal Investigator of an earlier CFHT MegaPrime/MegaCam survey, which PAndAS builds on (which included S.C.C., A.M.N.F., M.J.I., G.F.L., N.F.M. and A.W.M.). R.M.R. is Principal Investigator of the spectroscopic followup with the Keck Telescope. M.L.C. and S.C.C. led the analysis of the kinematic determination of the dwarf population, and N.F.M. led the detection of the dwarf population from PAndAS data. N.G.I. performed the initial analysis of the satellite kinematics.

Correspondence Reprints and permissions information is available at www.nature.com/reprints. Correspondence and requests for materials should be addressed to R.A.I. (rodrigo.ibata@astro.unistra.fr).


Galactic Longitude
Figure 1: Map of the Andromeda satellite system. The homogenous PAndAS survey (irregular polygon) provides the source catalogue for the detections and distance measurements of the 27 satellite galaxies ${ }^{20}$ (filled circles) used in this study. Near M31 (ellipse), the high background hampers the detection of new satellites and precludes reliable distance measurements for M32 and NGC 205 (black open circles); we therefore exclude the region inside 2.5 (dashed circle) from the analysis. The seven satellites known outside the PandAS area (green circles/arrows) constitute a heterogenous sample, discovered in various surveys with non-uniform spatial coverage, and their distances are not measured in the same homogenous way. Since a reliable spatial analysis requires a dataset with homogenous selection criteria, we do not include these objects in the sample either. The analysis shows that satellites marked red are confined to a highly planar structure. Note that this structure is approximately perpendicular to lines of constant Galactic latitude, so it is therefore aligned approximately perpendicular to the Milky Way's disk (the grid squares are $4^{\circ} \times 4^{\circ}$ ).


Figure 2: Satellite galaxy positions as viewed from Andromeda. The (Aitoff-Hammer) projection shows the sample of 27 satellites $^{20}$ (filled circles from Figure 1) as they would be seen from the centre of the Andromeda galaxy. In these coordinates the disk of Andromeda lies along the equator. The background image represents the probability density function of the poles derived from $10^{5}$ iterations of resampling the 27 satellites from their distance PDFs, and finding the plane of lowest rms from a sub-sample of 15 (the colour wedge shows the relative probability of the poles). A clear narrow peak at $\left(l_{M 31}=100^{\circ} .9 \pm 0^{\circ} 9, b_{M 31}=-38^{\circ} .2 \pm 1^{\circ} .4\right)$ highlights the small uncertainty in the best-fit plane. The solid red line, which passes within less than $1^{\circ}$ of the position of the Milky Way (yellow circle), represents the plane corresponding to this best pole location.

Figure 3: (Next page) Three-dimensional view of the planar, rotating structure (can be visualised and manipulated in 3D with Adobe Reader). The coordinate system is such that the $z$ direction is parallel to the vector pointing from the Milky-Way to M31, $x$ increases eastwards and $y$ northwards. Only the radial component of velocity of the satellites is measured, these are shown as vectors pointing either towards or away from the Milky Way. As in Figs. 1 and 2, red spheres mark the planar satellites, while blue spheres represent the "normal" population. The coherent kinematic behaviour of the spatially very thin structure (red) is clearly apparent viewed from the $y-z$ plane. With the exception of And XIII and And XXVII, the satellites in the planar structure that lie to the North of M31 recede from us, while those to the South approach us; this property strongly suggests rotation. Our velocity measurements ${ }^{15}$ (supplemented by literature values ${ }^{14}$ ), have very small uncertainties, typically $<5 \mathrm{~km} \mathrm{~s}^{-1}$. The irregular green polygon shows the PAndAS survey area, the white circle indicates a projected radius of 150 kpc at the distance of M31, and the white arrow marks a velocity scale of $100 \mathrm{~km} \mathrm{~s}^{-1}$. (And XXVII is not shown in this diagram as its most likely distance is 476 kpc behind M31). This figure is interactive (allowing the reader to change the magnification and viewing angle), and was constructed with the S2PLOT programming library ${ }^{26}$.

## Supplementary Information

## 1 Monte Carlo tests for statistical significance

To determine the best plane that fits the data it is necessary to take the distance uncertainties into account. This is accomplished as follows: first, we generate a set of 27 (three-dimensional) positions, by randomly drawing from the distance PDFs of each satellite. Subsequently, we find the plane that has the lowest root-mean-square (rms) distance to any sub-sample of $n_{\text {sub }}=15$ satellites. By repeating 1000 times the procedure of drawing 27 satellites from their distance PDFs and calculating the lowest rms plane, we obtain a probability density function for the root mean square thickness of a possible planar sub-structure of $n_{\text {sub }}=15$ satellites given the data. The distribution is approximately Gaussian with $\sigma=0.6 \mathrm{kpc}$, and mean 12.6 kpc . The 15 satellites that are closest to the lowest rms plane are marked with red circles in Figures 1 and 2.

The next question we need to assess is: given the M31-centric distance distribution to the satellites, what is the chance that they could be arranged at random to form a planar structure with equal or lower rms? To answer this question we performed a careful Monte Carlo simulation using random realisations of satellite configurations that are isotropically-distributed in the Andromedan sky, but preserve the same distribution of radial (not projected) distances to the centre of M31 as the observed one. To achieve this, an artificial satellite is generated by selecting a satellite at random from the real set, giving it a random 3-dimensional orientation with respect to Andromeda, and assigning the same line of sight distance PDF as the real satellite has (although shifted to the new distance). If the resulting artificial satellite is located outside of the PAndAS area, or within a projected distance of 2.5 , it is discarded. In this way we draw a random sample of 27 artificial satellites (with replacement) which we process in an identical way as we processed the real sample: creating a new set of 1000 rms values, from which we construct a probability density function for the rms for that particular sample of 27 artificial satellites. The whole procedure is repeated $10^{5}$ times. The corresponding histogram of mean rms values in the artificial satellite samples is shown in Figure S1. We find that a mean rms scatter of 12.6 kpc or less occurs with probability $0.13 \%$.

Thus the sample of 15 satellites that is closely aligned on the great circle in Figure 2 (red circles) is indeed a highly surprising and statistically significant detection.

The above choice of $n_{\text {sub }}=15$ members for the sub-sample size was made by examining the effect that this parameter has on the rms thickness and on the position of the best-fit plane. We find that for $5 \leq n_{\text {sub }} \leq 15$ the position of the pole to the lowest rms plane remains very stable, while the plane rms thickness increases only slowly with increasing $n_{\text {sub }}$. However, for values of $n_{\text {sub }} \geq 16$ the plane rms increases rapidly, and also there is no well-defined plane solution, unlike the tight solution for $n_{\text {sub }}=15$ displayed as the background image in Figure 2. A slightly higher significance is derived by choosing an $n_{\text {sat }}$ value of 13 or 14, as the smaller rms scatter about the plane more than compensates for the lower statistics in the sub-sample.

## 2 The planar satellites

The co-rotating satellites are: And I, And III, And IX, And XI, And XII, And XIV, And XVI, And XVII, And XXV, And XXVI, Cas II, NGC 147 and NGC 185. The two satellites that do not partake in this motion are And XIII and And XXVII; it seems likely that these are interlopers from the normal non-planar population, although they may nevertheless be members of the planar subgroup if the population has significant velocity dispersion. Figure 1 shows that of the known M31 satellite galaxies, M32, NGC 205, LGS 3, IC 10 and IC 1613 (the latter three are situated up to 40 degrees from M31 in the directions indicated) also lie along the same axis as the red-coloured objects from our sample, and published distance estimates ${ }^{28}$ are consistent with all five satellites being at the same distance as their host within the uncertainties. However, only LGS 3, IC 1613 and NGC 205 share the same sense of rotation as the 13 planar satellites listed above, and may plausibly be associated with the structure.

No obvious differences in the physical properties (metallicity, stellar populations or velocity dispersion) were found between the 13 kinematically-coherent coplanar satellites and the remainder of the sample.


Figure S1: Statistical significance of the spatial alignment. The histogram presents the distribution in $10^{5}$ random trials of the average root-mean-square distances from a best-fit plane of sub-samples of 15 satellites. The average rms value of 12.6 kpc , derived from the real configuration, is extremely rare, occurring with probability $0.13 \%$ in random realisations. This demonstrates that the observed spatial alignment of dwarf galaxies is very unlikely to be a random coincidence. If we had chosen a smaller sub-sample size, the chance alignment becomes even more extreme: e.g. $0.06 \%$ for $n_{\text {sub }}=13$.

## References

28. McConnachie, A. W. The Observed Properties of Dwarf Galaxies in and around the Local Group. Astronomical Journal 144, 4, July (2012).

## B

## Chapter Two Programs

EdgeFinder7.f95 ..... 140
RGBPeakFinder6.f95 ..... 149
spikes.f95 ..... 155

## Program: EdgeFinder7.f95

Creation Date: 3 September 2009
Relevant Section: 2.2
Notes: This program represents one of my earliest investigations into potential TRGBfinding algorithms. It is really a number of stand alone algorithms rolled into one program. An artificial 'kink' is induced in a simple luminosity function and this kink is sort out by a number of methods: 1 . By fitting a polynomial to the data and finding where the second derivative of that polynomial has the largest absolute value; 2. By Finding the largest positive gradient between two neighboring bins of the luminosity function and; 3. By taking the angle subtended by each subsequent set of 3 luminosity function bins. The identified location of the tip is outputted along with the value of the particular measurement statistic.

```
PROGRAM EdgeFinder
IMPLICIT NONE
!A polynomial of degree ma - 1 is fitted to the read-in data and a '.p' file is
!generated so that the read-in data can be instantly plotted using gnuplot along
!with the fitted polynomial. The RandReal subroutine then generates a mock data
!set based on the fitted polynomial and the d2ydx2max subroutine finds where the
!maximum rate of change of the gradient occurs which is symbolic of an 'edge,
!or sudden discontinuity in the fitted polynomial.
!Later adapted to use pgplot
INTEGER :: ma,mp,ndata, ndat,np
parameter ( np = 20)
parameter (mp = 1000)
parameter (ndat = 1000)
parameter (ma = np)
DOUBLE PRECISION :: chisq,a(ma), sig(ndat),u(mp,np),v(np,np),w(np),x(ndat)
DOUBLE PRECISION :: y(ndat), z(ndat), integral_max, dummy, e(ndat), f(ndat)
EXTERNAL :: funcs
INTEGER :: ios, i, j
integer :: ma_max,ma_used
parameter (ma_max=100)
DOUBLE PRECISION :: pass_a (ma_max)
common/pass_block2/pass_a, ma_used
ma_used=ma
x=0. ; y=0. ; z=0. ; sig=0.00001
OPEN (unit = 1, file = 'luminosity_function2.dat', status ='old')
OPEN (unit = 2, file = 'lffit.dat', status = 'unknown')
i=0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
    i=i+1
    READ (1, *, IOSTAT = ios) dummy,x(i),dummy, dummy, dummy, dummy, y(i)
    if (ios == 0) then ;
```

```
    else if (ios == -1) then ;
            i=i - 1
            exit ;
    else if (ios > 0) then ;
        i=i - l
        cycle
    end if
x(i) = x(i)/5.
IF(x(i) > -0.2) then
    i = i-1
else if (abs(x(i)).1t.0.1) then
    i = i-1
end if
END DO
DO j = 1, i !Outputs exclusively the chosen data to 'lffit.dat,
    WRITE (2, '(2ES20.5)') x(j), y(j)
END DO
ndata=i
CALL svdfit(x,y,sig,ndata,a,ma,u,v,w,mp,np,chisq,funcs) !SVD fitting program
do j=1,ma
    pass_a(j)=a(j)
end do
PRINT *, a
OPEN(unit = 3, file = "lffit.p", status = 'unknown')
CALL RandReal(ma, a, x, i, sig, ndata, u, v, w, mp, np, chisq, funcs)
!CALL d2ydx2max (ma, a)
END PROGRAM EdgeFinder
SUBROUTINE RandReal(ma, a, x, i, sig, ndata, u, v, w, mp, np, chisq, funcs)
!Random realization mock data generator
INTEGER :: ma, i, q, l, val, ndat
PARAMETER (val = 200)
PARAMETER (ndat = 1000)
INTEGER :: idum = 0
DOUBLE PRECISION :: a(ma), area, max_x (ma + 1), min_x (ma + 1), x(i)
DOUBLE PRECISION :: ran1, randnum, sig(ndat), u(mp,np),v(np,np),w(np)
DOUBLE PRECISION :: b(ma + 1), rtr(ma), rti(ma), chisq
DOUBLE PRECISION :: mock_x(val), mock_y(val), mock_y_at_l(ma)
OPEN (unit = 1, file = "mockdata.dat", status = 'unknown')
OPEN(unit = 2, file = "mockdata.p", status = 'unknown')
DO q = 1, ma
    b(q+1)=a(q)/q !Transfers from coefficients of p(x) to those of integral
END DO
b(1) = 0
```

```
DO q = ma + 1, 1, -1 !Calculates the variable 'area' - the
    min_x(q) = b(q) * x(i)**(q-1) !area under the polynomial between
    max_x(q) = b(q)*x(1)**(q-1) !x(1) and x(i) - i.e. the range of the
END DO
area = SUM(max_x) - SUM(min_x ) !domain.
DO q = 1, val
    randnum = ran 1(idum)
    b(1) = -(SUM(min_x) + randnum * area) !Generates a random y value between the value of the integral at x(1) and at x(i).
    CALL zrhqr(b,ma, rtr, rti) ! Finds roots of integral for given y value
    DO l = 1, ma
            IF(rti(1) == 0.) THEN ! Only use the real roots
                IF(rtr(1) .gt. MINVAL(x)) THEN !Make sure the chosen root
                    IF(rtr(1) . lt. MAXVAL(x)) THEN ! is in the domain used
                    mock_x(q) = rtr(1)
                    END IF
            END IF
            END IF
        END DO
END DO
DO q = 1, val
    DO 1 = 1, ma
            mock_y_at_l(1) = a(1) * mock_x (q) **(1-1)
    END DO
    mock_y(q) = SUM(mock_y_at_1)
    WRITE (1, '(2ES20.5)') mock_x(q), mock_y(q)
END DO
WRITE (2,*) 'plot-\', !Prints fitted
DO j = ma, 2, -1 !polynomial to
    WRITE (2,*) a(j), '*_x**', j - 1, ''+_\, !a,'p' file for
END DO !plotting with
WRITE (2,*) a(1), 'title \lrcorner','svdfit',,,`mockdata.dat", !gnuplot
CALL Kink(ma,a,mock_x,mock_y,val, sig,ndata,u,v,w,mp,np,chisq,funcs)
END SUBROUTINE RandReal
SUBROUTINE Kink(ma,a,mock_x, mock_y, val, sig, ndata,u,v,w,mp, np,chisq, funcs)
!Generates a new set of mock data points with a kink at offset of 0.25
INTEGER :: val, q, ma, l, h, ndat, ndata, ios = 0
PARAMETER (ndat = 1000)
DOUBLE PRECISION :: a(ma), mock_x(val), mock_y(val), e(ndat), f(ndat)
DOUBLE PRECISION :: shift_x(val), shift_y(val), shift_y_at_l(ma), new_y(val)
INTEGER :: mp, np
DOUBLE PRECISION :: chisq, sig(ndat),u(mp,np),v(np,np),w(np)
EXTERNAL :: funcs
DOUBLE PRECISION :: yp1,ypn,ya2(ndat),x,y, der_abs(ndat)
INTEGER :: indx (ndat)
DOUBLE PRECISION :: xa(ndat), ya(ndat)
OPEN (unit = 1, file = "kink.dat", status = 'unknown')
DO q}=1, va
    shift_x(q) = mock_x(q) + 0.25 !Offsets mockdata
    DO 1 = 1, ma !along x-axis by 0.25
```

```
            shift_y_at_l(1)= a(1)* shift_x(q)**(l-1) !and then adds these
    END DO
    hift_y(q) = SUM( shift-y-at-l)
    new_y (q) = (mock_y (q) + 5.0*MINVAL(mock_y)) + shift_y(q) ! to the previous ones.
    WRITE (1, *) shift_x (q), new_y(q) !5*MINVAL(mock_y)
END DO
    !makes large kink.
DO q = 1, val
    IF (mock_x(q) .lt. MINVAL(shift_x)) THEN !Outputs original mockdata points
        WRITE (1, *) mock_x(q), mock_y(q) !for mock_x points less than the
    END IF !minimum shift_x value
END DO
REWIND (1)
e = 0. ; f = 0. ; h = 0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
    h=h+1
    READ (1, *, IOSTAT = ios) e(h), f(h) !i.e. Read shift_x(q), new_y(q)
    if (ios == 0) then ;
    else if (ios == -1) then ;
        h=h-1
        exit ;
    else if (ios > 0) then ;
        h=h-1
        cycle
    end if
END DO
ndata = h
CALL indexx(ndata,e,indx) !Creates array indx(1:ndata) whose elements are
                                    !indicies to the elements of e in chronological order
DO j=1,ndata !
    xa(j)=e(indx(j)) !Makes xa and ya equal to the ordered versions of
    ya(j)=f(indx(j)) !e and f respectively
    write (*,*) xa(j), ya(j)
END DO
WRITE (*,*) MINVAL(xa), MAXVAL(xa ), MINVAL(ya), MAXVAL(ya)
call pgbegin(0,'?', 1, 1)
call pgenv(REAL(MINVAL(xa)), REAL(MAXVAL(xa)), &
    0., REAL(MAXVAL(ya) ), 0, 0)
call pgpt(ndata, REAL(xa), REAL(ya), 1)
call pgend
!CALL PolyTest(xa,ya, sig, ndat, ndata,a,ma,u,v,w,mp,np,chisq, funcs)
CALL SplineTest(xa,ya,ndata,mock_x, val)
CALL GradTest(xa, ya, ndata, mock_x, val)
CALL AngleTest(xa, ya, ndata, mock_x, val)
CALL LegPrint(ma, a)
END SUBROUTINE Kink
```

```
SUBROUTINE PolyTest(xa,ya, sig, ndat, ndata,a,ma,u,v,w,mp,np,chisq, funcs)
!Use to test the polynomials ability to find the kink
INTEGER :: ndat, ndata, ma, mp, np
DOUBLE PRECISION :: sig(ndat), a(ma), u(mp,np), v(np,np), w(np), chisq, x(ndata)
DOUBLE PRECISION :: y(ndata), xa(ndata), ya(ndata)
EXTERNAL :: funcs
x = xa; y = ya !Don't want high precission here
OPEN(unit = 1, file = "kinkpoly.p", status = 'unknown')
a}=0
CALL svdfit(x,y,sig, ndata,a,ma,u,v,w,mp,np,chisq,funcs) !SVD fitting program
WRITE (1,*), set_xr_[-1.0_:_-0.2],
WRITE (1,*) 'plot-\', !Prints fitted
DO j = ma, 2, -1 !polynomial to
    WRITE (1,*) a(j), ', *_x**', j - 1, ',+\\, !a'.p' file for
END DO !plotting with gnuplot
WRITE (1,*) a(1), 'title 弓','svdfit',,, „"kink.dat",
CALL LegPlot2(xa, ya, ndata)
END SUBROUTINE PolyTest
SUBROUTINE SplineTest(xa,ya, ndata,mock_x,val)
!Use to test the spline functions ability to find the kink
INTEGER :: ndata, indx(ndata), val
DOUBLE PRECISION :: yp1, ypn, ya2(ndata), x, y, der_abs(ndata), a(ndata)
DOUBLE PRECISION :: b(ndata), mock_x(val), xa(ndata), ya(ndata)
a = xa; b = ya !Don't want high precision here
yp1=1.1e30 !Makes 2nd derivative (ya2) equal to zero at either
ypn=1.1e30 !end of the spline-interpolated function
CALL spline_NR(a,b,ndata,yp1,ypn,ya2) ! Finds 2nd der. of tabulated fn f(e)
der_abs = 0.
DO j = 1, ndata
    IF (xa(j) .gt. -0.8) THEN
        IF(xa(j) . lt. -0.2) THEN !Find location and
            der_abs(j) = ABS(ya2(j)) !absolute value of
            END IF !the maximum
    END IF !second derivative
END DO
PRINT *, 'Max._abs._val._of_2nd_der._is', MAXVAL(der_abs) !
```



```
PRINT *, 'Giving_offset:', xa(MAXLOC(der_abs)) - MINVAL(mock_x)
OPEN (1, file= 'spline.dat', status = 'unknown')
DO j=1,1000
    x=-0.9+j*0.7/1000.0 !x range and interval size for outputted spline data
    CALL splint_NR(a,b,ya2,ndata, x,y)!Returns cubic-spline interpolated value y
    WRITE(1,*) x,y
END DO
CLOSE (1)
```

284
285
DO $\mathbf{j}=1$, ndata -1
IF ( $\mathrm{xa}(\mathrm{j}) \cdot \mathrm{gt} .-1.0)$ THEN
IF ( $\mathrm{xa}(\mathrm{j}$ ) . lt. - 0.2 ) THEN
$\operatorname{grad}(j)=(y a(j+1)-y a(j)) /(x a(j+1)-x a(j))$ !adjacent
END IF ! points
END IF !in the range
END DO $\quad!-1.0<\mathrm{x}<-0.2$
graddiff $=0 . \quad$ !Find difference
DO $\mathbf{j}=1$, ndata $-2 \quad$ !between each
IF (grad (j) .ne. 0.) THEN !two adjacent
IF ( $\operatorname{grad}(\mathrm{j}+1)$.ne. 0.) THEN ! gradients so
$\operatorname{graddiff}(j)=\operatorname{grad}(j+1)-\operatorname{grad}(j) \quad$ !long as neither
END IF !of the two
END IF ! gradients are
END DO !equal to zero.
! Note: absolute values are not used in either of the above loops as we are
!specifically looking for positive gradients and for the edge where the
! gradient goes from small to large with increasing $x$.
PRINT *, 'Largest_gradient_difference_is_at_x_=, $\quad x a($ MAXLOC(graddiff) +1$)$
PRINT *, 'With_magnitude', MAXVAL(graddiff)

END SUBROUTINE GradTest
SUBROUTINE AngleTest (xa, ya, ndata, mock_x, val)
! This subroutine is designed to find the angle between lines connecting
!adjacent data points. Starting from the smallest $x$ value (say point 1), the
! line joining point 2 and point 3 is denoted length 'a', that joining point 1
! and point 2 is denoted length 'b, and that connecting points 1 and 3 is
! denoted length 'c,' thus setting up a triangle. The trigonometric rule
$!c=a+b-2 a b \cos C$ is then used to find angle C. The algorithm then shifts
! to concentrate on the triangle made by points 2,3 and 4 and so on until the
!end of the data. The sharpest gradient change occurs where angle $C$ is smallest.

```
!Note that only positive gradient changes from left to right are considered
INTEGER :: ndata, j, val
DOUBLE PRECISION :: mock_x (val)
DOUBLE PRECISION :: xa(ndata), ya(ndata), grad(ndata-1)
DOUBLE PRECISION :: a(ndata - 2), b(ndata - 2), c(ndata - 2), angle_c (ndata - 2)
grad = 0. !Find
DO j = 1, ndata - 1 !gradient
    IF (xa(j) .gt. -1.0) THEN !between
        IF (xa(j) . 1t. -0.2) THEN !each two
            grad(j) = (ya(j + 1) - ya(j))/(xa(j +1) - xa(j)) !adjacent
        END IF !points
    END IF !in the range
END DO
angle_c = 7. !Makes angle_c larger than 2* pi for grad(j+1)<grad(j)
DO j = 1, ndata-2 !Populates
    a(j) = SQRT((xa(j + 2) - xa(j +1)) **2 + (ya(j +2) - ya(j +1))**2) !array
    b(j) = SQRT((xa(j+1) - xa(j))**2 + (ya(j+1) - ya(j))**2) !'angle_c,
    c(j) = SQRT((xa(j+2) - xa(j))**2 + (ya(j+2)-ya(j))**2) ! with the
    IF (grad (j+1) .gt.grad(j)) THEN !angles
        angle_c(j) = ACOS(-(cc(j)**2-a(j)**2-b(j)**2)/(2*a(j)*b(j))) ! between
    END IF !each set
END DO !of points
```



```
PRINT *, 'With\lrcornermagnitude', MINVAL(angle_c), 'radians'
PRINT *, 'This_recovers_an_offset_of,, xa(MINLOC(angle_c) + 2) - MINVAL(mock_x)
END SUBROUTINE AngleTest
SUBROUTINE LegPrint(ma, a) !For printing Legendre polynomials
implicit none
REAL*8 :: fac1, fac2, fac3, fac4
INTEGER :: ma, n, k, j
INTEGER :: ma_max,ma_used
PARAMETER (ma_max = 100)
INTEGER :: LegEx (0:ma_max - 1, 0:(ma_max/2))
DOUBLE PRECISION :: a (ma)
DOUBLE PRECISION :: LegCo(0:ma_max - 1, 0:(ma_max/2)), PolyCo(ma_max)
CHARACTER(LEN=100), DIMENSION(0:ma_max - 1, 0:(ma_max/2)) :: P
CHARACTER(LEN=100), DIMENSION(ma_max) :: PolyElement
CHARACTER(LEN=1000) :: Polynomial = , ,
COMMON /PASS/ PolyCo, ma_used
PolyCo = 0.
if (ma.gt.ma_max) then
    write(*,*) 'ma\_too\iotalarge' ; stop
end if
ma_used=ma
!|Calculate Legendre polynomial
!\/exponents and coefficients
DO n = 0, ma-1
    DO k = 0, n/2
        fac1 = 1; fac2=1; fac3=1; fac4 = 1
```

406
407

DO $\mathrm{j}=1,2 * \mathrm{n}-2 * \mathrm{k}$

$$
\mathrm{fac} 1=\mathrm{fac} 1 * \mathrm{j}
$$

END DO
DO $\mathrm{j}=1, \mathrm{k}$ fac2 $=$ fac $2 * j$
END DO
DO $\mathrm{j}=1, \mathrm{n}-\mathrm{k}$
fac3 $=$ fac $3 * j$
END DO
DO $\mathrm{j}=1, \mathrm{n}-2 * \mathrm{k}$

$$
\mathrm{fac} 4=\mathrm{fac} 4 * \mathrm{j}
$$

END DO
$\operatorname{LegCo}(\mathrm{n}, \mathrm{k})=\mathrm{a}(\mathrm{n}+1) *(((-1 . \mathrm{d} 0) * * \mathrm{k}) *(\mathrm{fac} 1 /((2 \cdot \mathrm{~d} 0 * * \mathrm{n}) * \mathrm{fac} 2 * \mathrm{fac} 3 * \mathrm{fac} 4)))$
$\operatorname{LegEx}(\mathrm{n}, \mathrm{k})=\mathrm{n}-2 \cdot \mathrm{~d} 0 * \mathrm{k}$
$\operatorname{WRITE}(\mathrm{P}(\mathrm{n}, \mathrm{k}), *) \operatorname{LegCo}(\mathrm{n}, \mathrm{k}),{ }^{\prime} *_{\llcorner } \mathrm{X}_{\llcorner }{ }^{*}{ }^{\prime}, \quad \operatorname{LegEx}(\mathrm{n}, \mathrm{k})$
PRINT *, $\operatorname{TRIM}(\mathrm{P}(\mathrm{n}, \mathrm{k}))$
END DO
PRINT *, ',
END DO

DO $\mathrm{j}=1$, ma
DO $\mathrm{n}=0$, ma-1
DO $\mathrm{k}=0, \mathrm{n} / 2$
IF (LegEx $(\mathrm{n}, \mathrm{k})==\mathrm{j}-1)$ THEN
$\operatorname{PolyCo}(\mathrm{j})=\operatorname{PolyCo}(\mathrm{j})+\operatorname{LegCo}(\mathrm{n}, \mathrm{k})$
END IF
END DO
END DO
END DO

OPEN (unit $=1$, file = "lffit $2 . \mathrm{p} "$, status $=$ 'unknown')
WRITE ( $1, *$ ) 'plot_<br>,

```
DO j = ma, 1, -1
```

IF ( j . ne . 1) THEN

IF (PolyCo(j - 1) .gt. 0.d0) THEN
WRITE(PolyElement(j), *) PolyCo(j), ' ${ }_{*-x-* * ', ~ j-1, ~ '+' ~}^{\text {, }}$
END IF
IF (PolyCo(j - 1) . lt. 0.d0) THEN
WRITE(PolyElement(j), *) PolyCo(j), '*_x_**', $\mathrm{j}-1$
END IF
END IF
IF ( $\mathrm{j}==1$ ) THEN
WRITE (1,*) PolyCo(j), 'title_,'svdfit,', -" lffit.dat",
WRITE(PolyElement (j) ,*) PolyCo(j)
END IF
Polynomial $=$ TRIM(Polynomial) // TRIM(PolyElement(j))
END DO

PRINT *, TRIM(Polynomial)

END SUBROUTINE LegPrint

SUBROUTINE LegPlot2(xa, ya, ndata) ! For plotting Legendre polynomials with pgplot IMPLICIT NONE

INTEGER :: ndata, j

```
DOUBLE PRECISION :: xa(ndata), ya(ndata)
REAL :: xasp(ndata), yasp(ndata), Legendre_new, dummy
EXTERNAL :: Legendre_new
xasp = xa ; yasp = ya
dummy = Legendre_new( -0.5)
DO j= 1, ndata
    PRINT *, j, xasp(j), yasp(j)
END DO
CALL pgbegin(0,'?', 1,1)
CALL pgfunx(Legendre_new , 100, -0.9, -0.2,0)
CALL pgpt(ndata, xasp, yasp, 228)
CALL pgend
END SUBROUTINE LegPlot2
```

$\qquad$

```
SUBROUTINE d2ydx2max(ma, a) ! Finds the turning pts of the 2nd derivative of p(x)
INTEGER :: ma, j
DOUBLE PRECISION :: a(ma), c(ma-2), d(ma-3), rtr(ma-3), rti(ma-3)
DO j = 1, ma-2
    c(j) = a(j+2)*(j+1)*(j) ! Finds the coefficients for d2ydx2
END DO
DO j = 1, ma-3
    d(j) = c(j+1) * j ! Finds the coefficients for the 3rd derivative (d3ydx3)
END DO
CALL zrhqr(d,1, rtr, rti) ! Finds the roots for the maximum rate of change of the
DO j=1, ma - 3 !gradient (d3ydx3 = 0) and d2ydx2's magnitude there
    IF(rti(j) == 0) THEN !Include only real roots
```



```
    END IF
END DO
END SUBROUTINE d2ydx2max
```



Program: RGBPeakFinder6.f95
Creation Date: 7 December 2009
Relevant Section: 2.2
Notes: This is another RGB finder which combines elements of 'edge-finding' and model fitting. Stars in a small region around Andromeda I are read in and a smoothed 'Luminosity Probability Distribution' is produced from the individual stellar magnitudes via a Gaussian smoothing of the luminosity function. The second derivative of this distribution is produced with the peaks denoting inflection points in the gradient of the smoothed luminosity function. See Figs. 2.1 and 2.2.

```
MODULE Global3 !Define all
IMPLICIT NONE ! Variables
INTEGER :: ndata_max, ndata, ndata_t, i, j, k, n, ios, idum = 0, randnum
INTEGER :: ndata2, ndata_sub, div_per_mag = 100
PARAMETER (ndata_max = 100000)
REAL :: xmin, xmax, min_mag = 19.5d0, max_mag = 21.5d0
REAL*8 :: temp_x(ndata_max), temp_y(ndata_max), temp_e(ndata_max), dummy
REAL*8 :: mag(10000), phi(10000), mag2(10000), phi2(10000)
REAL*8 :: mag3(10000), phi3(10000), phi4(10000), phi4_max
END MODULE Global3
PROGRAM RGBPeakFinder6 !Finds greatest peak of (d2phi/dm2)/(phi) between min_mag and max_mag.
USE GLOBAL3
IMPLICIT NONE
INTEGER :: check(100000)
REAL*8 :: xh1(ndata_max), yh1(ndata_max), eh1(ndata_max)
temp_x = 0.d0 ; temp_y = 20.d0
OPEN (unit = 1, file =, //m31_fields_stellar/ANDI_box_small.dat,, status = 'old')
i = 0 ; ios = 0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
    i=i+1
    READ (1, *, IOSTAT = ios) temp_x (i), temp_y(i) !x: g magnitude
                                    !y: i magnitude
    if (ios == 0) then ;
    else if (ios == - 1) then ;
        i=i-1
        exit ;
    else if (ios > 0) then ;
        i=i-1
        cycle
    end if
    IF (temp_x (i) == 0. .or. temp_y(i) == 0.) THEN
        i=i-1
        cycle
    END IF
```

```
END DO
ndata = i
PRINT *, "Number_of_sources_=", ndata
DO j = 1, ndata
    temp_x(j) = temp_x(j) - temp_y(j)
END DO
CALL CutPlot
CALL Smooth
END PROGRAM RGBPeakFinder6
l-2,
SUBROUTINE CutPlot !Produce colour cuts
USE Global3
IMPLICIT NONE
REAL :: div, var(1000)
REAL :: uppercut(1000), lowercut(1000)
div = 2*(INT(MAXVAL(temp_x)) - INT(MINVAL(temp_x)))/1000
DO j = 1, 1000
    var(j) = INT(MINVAL(temp_x)) + div*j
    uppercut(j) = 24.5 - 3*(var(j))
    lowercut(j) = 27.0 - 3*(var(j))
END DO
!------------------------------------------
CALL pgbegin(0,'temp1.ps/CPS',1,1)
CALL pgenv(MINVAL(REAL(temp_x)), MAXVAL(REAL(temp_x )), MAXVAL(REAL(temp_y)), MINVAL(REAL(temp_y)), 0, 0)
CALL pgpt(ndata, REAL(temp_x), REAL(temp_y), 1)
CALL pgsci(2)
CALL pgline(1000, var, uppercut)
CALL pgline(1000, var, lowercut)
CALL pgsci(1)
CALL pglab('(g_-_i)\d0\u',, 'i\d0\u', ',')
CALL pgend
!-_-_-_-_
END SUBROUTINE CutPlot
1-2,
SUBROUTINE Smooth !Apply Gaussian smoothing to LF and
USE Global3 !find inflection points
IMPLICIT NONE
REAL*8 :: x(ndata), y(ndata), xa(ndata), ya(ndata), pi = 2*ACOS(0.d0)
REAL*8 :: e(ndata), err(ndata), ave_phi, ave_mag, ave_phi4
REAL*8 :: xa_sel(ndata), ya_sel(ndata), err_sel(ndata)
REAL :: yasp(ndata), phisp(10000), magsp(10000)
INTEGER :: indx(ndata), place, place2, place3, counts, denominator
DO j = 1, ndata
```

```
    x(j) = temp_x(j)
    y(j) = temp_y(j)
END DO
CALL indexx(ndata,y,indx) ! Creates array indx(1:ndata) whose elements are
                                    !indicies to the elements of y in chronological order
DO j=1, ndata
    xa(j)=x(indx(j)) !Makes xa, ya & err equal to the ordered versions of
    ya(j)=y(indx(j)) !x, y & e respectively
    err(j) = 0.1d0
END DO
counts = 0 ; xa_sel = 0.d0 ; ya_sel = 0.d0
DO j=1, ndata
    IF (ya(j) .gt. 24.5 - 3*(xa(j))) THEN !Throw away stars
        IF (ya(j) . 1t. 27.0 - 3*(xa(j))) THEN !outside of colour cut
            counts = counts + 1 !counts is the total number of accepted stars
            xa_sel(counts) = xa(j)
            ya_sel(counts) = ya(j)
            err_sel(counts)= err(j)
        END IF
    END IF
END DO
```



```
CALL pgbegin(0,'?',1,1)
CALL pgenv (MINVAL(REAL(temp_x )), MAXVAL(REAL(temp_x )), MAXVAL(REAL(temp_y)), MINVAL(REAL(temp_y )), 0, 0)
CALL pgpt(counts, REAL(xa_sel), REAL(ya_sel), 1)
CALL pglab('(g--ьi)\d0\u', ', i\d0\u', ',')
CALL pgend
!Produce Luminosity Probability Distribution (LPD)
place = 0 ; phi = 0.d0 ; mag = 0.d0
DO j = 100*(INT(MINVAL(ya)) - 1), 100*(INT(MAXVAL(ya)) + 1), (100/div_per_mag)
    place = place + 1
    mag(place) = j/100.d0
    DO k = 1, counts !Perform Gaussian smoothing at magnitude by summing contributions
        phi(place) = phi(place) +& !of each star represented by a normalized Gaussian
        (1.d0/(SQRT(2.d0 * pi) * err_sel(k))) * EXP(-((ya_sel(k) - j/100.d0)**2.d0)/ (2.d0*(err_sel(k) **2.d0)))
    END DO
END DO
!Produce derivative of LPD
place2 = 0 ; phi2 = 0.d0 ; mag2 = 0.d0
DO j = 100*(INT(MINVAL(ya)) - 1), 100*(INT(MAXVAL(ya)) + 1), (100/div_per_mag)
    place2 = place2 + 1
    mag2(place2) = j/100.d0
    DO k = 1, counts
        phi2(place2) = phi2(place2) + &
        (1.d0/(SQRT(2.d0 * pi) * (err_sel(k))*3)) * EXP(-((ya_sel(k) - j/100.d0)**2.d0)/ (2.d0*(err_sel(k)**2.d0))) * (ya_sel(k) - (j
        /100.d0))
    END DO
END DO
!Produce second derivative of LPD
place3 = 0 ; phi3 = 0.d0 ; mag3 = 0.d0
```

```
DO j = 100*(INT(MINVAL(ya)) - 1), 100*(INT(MAXVAL(ya)) + 1), (100/div_per_mag)
    place3 = place3 + 1
    mag3(place3) = j/100.d0
    DO k = 1, counts
        phi3(place3) = phi3(place3) + &
        (1.d0/(SQRT(2.d0 * pi) * (err_sel(k))*5)) * EXP(-((ya_sel(k) - j/100.d0)**2.d0)/ (2.d0*(err_sel(k) **2.d0))) * (ya_sel(k)**2 - 2*
            ya_sel(k)*(j/100.d0) - err_sel(k)**2 + (j/100.d0)**2)
    END DO
END DO
phi4 = (phi3/(ABS(phi)+0.1)) !Use 2nd der. of LPD divided by LPD to locate potential TRGBs
ndata2 = div_per_mag*(INT(MAXVAL(ya)) - INT(MINVAL(ya)) + 2) + 1
ndata_sub = (div_per_mag * (max_mag - min_mag)) + 1
ave_phi = SUM(phi)/ ndata2
ave_mag = SUM(mag)/ ndata2
ave_phi4 = 0. ; denominator = 0; phi4_max = 0.d0
DO j = 1, ndata2
    IF (mag(j) .ge. min_mag) THEN
            IF (mag(j) . le. max_mag) THEN
            ave_phi4 = ave_phi4 + phi4(j) !Find maximum value of array phi4 and
            denominator = denominator + 1 !the averagage value of array phi4 in
            IF (phi4(j) .gt. phi4_max) THEN !the region where the TRGB could
                phi4_max = phi4(j) !feasibly be located. These can then
            END IF !be used for scaling graphs and
            END IF !determining the strenghts of possible
    END IF !TRGBs.
END DO
ave_phi4 = ave_phi4/ denominator !
yasp = ya ; magsp = mag
phisp = phi*(phi4_max/ MAXVAL(phi)) !Scale REAL(phi) for plotting with phi4
DO j = ndata 2 + 1, 10000
    phi(j) = ave_phi !Since SIZE(phi) = SIZE(phisp) = SIZE(mag) = SIZE(magsp)
    mag(j) = ave_mag ! = 10000 .not. ndata2
    phisp(j) = 0. !it is necessary to make all array elements outside of
    magsp(j) = ave_mag !ndata2 equal to some intermediate value so that the
END DO !minval & maxval functions are still useable.
xmin = INT(MINVAL(yasp)) - 1. ; xmax = INT(MAXVAL(yasp)) + 1.
CALL TurningPoints
                                    !Main Plot (i.e. smoothed luminosity
                                    !function with inflection points)
CALL pgbegin(0,'temp2.ps/CPS',1,1)
CALL pgenv(18., 26., MINVAL(phisp), REAL(1.5*phi4_max), 0, 0)
CALL pgsci(1)
CALL pgline(ndata2, magsp, phisp)
CALL pgsci(2)
CALL pgline(ndata2, REAL(mag3), REAL(phi4))
CALL pgsci(4)
CALL pgsch(4.0)
CALL pgpt(1, 20.77, 0.001, 2264)
CALL pgsch(1.0)
CALL pgsci(1)
```

```
CALL pglab('i\d0\u', 'relative_probability,, ',')
CALL pgend
END SUBROUTINE Smooth
SUBROUTINE TurningPoints !Prints location of potential RGB tips (in magnitudes)
USE Global3 !and assigns to them a stength, based on the change in
IMPLICIT NONE !LF slope at that magnitude
INTEGER :: TRGB_found = 0
PRINT *,
DO j = 1, ndata2
    IF (mag(j) .ge. min_mag) THEN
        IF (mag(j) . le. max_mag) THEN
            IF (phi4(j) .gt. phi4(j-1) . and. phi4(j) .gt. phi4(j + 1)) THEN
                PRINT *, "Potential_TRGB_at", mag(j), &
                    "Strength_=", REAL(phi4(j)/ phi4_max)
                TRGB_found = 1
            END IF
        END IF
    END IF
END DO
IF (TRGB_found == 0) THEN
    PRINT *, "No_TRGB_could_be_located."
END IF
PRINT *,
END SUBROUTINE TurningPoints
SUBROUTINE RandSplit(check)
USE Global3
IMPLICIT NONE
!Subroutine creates a randomized index to the read-in data set. It is called
!by subroutine, TestSeparate, to split the data in two halves and process them.
INTEGER :: check(100000), count, num = 0
REAL*8 :: ran 1
check(100000)=0; j=0 ! Insure array check has all elements 0
DO WHILE (j . lt. ndata_t - 1) !
    j = j + 1
    randnum = INT(ndata_t *ran 1(idum))
    count = 0
    DO k = 1, j
        IF (randnum .eq. check(k)) THEN
            count = count + 1
        END IF !Loop generates randomised index with
    END DO !number of entries equal to ndata. All
    IF (count == 0) THEN !entries are unique integers from 1 to
        check(j) = randnum ; !ndata. Index is outputted to 'check'
    END IF
    IF (count . ne. 0) THEN
        j = j - 1
```

```
    cycle
    END IF
END DO
check(ndata_t) = ndata_t !Final element of 'check' is ndata
DO j = 1, ndata_t ! Check that
    DO k = 1, ndata_t ! all integer
        IF (check(k) == j) THEN ! values between
            num = num + 1 ! 1 and ndata
        END IF ! can be
    END DO ! found in the
END DO ! array 'check
PRINT *, "number_of_unique_integers_=", num
END SUBROUTINE RandSplit
```

$\qquad$


Program: spikes.f95
Creation Date: 1 February 2011
Relevant Section: 2.3
Notes: I created this program to illustrate the precise way in which the posterior probability distribution of a parameter is produced via maximum likelihood model fitting. An artificial luminosity function is created from a step function, where the user specifies the number of stars to be produced as well as the position of the step and the relative proportions of the 'background' and 'signal' components. Various types of priors can then be applied to the resulting posterior distributions. Fig. 2.4 illustrates the way in which the posterior distributions in the tip position are created. Figs. 2.3, 2.5 and 2.6 were created using this code.

```
MODULE Global !Define all
IMPLICIT NONE ! variables
INTEGER :: i, j, indx(20001)
REAL*8 :: like(1000), posterior (1000,2), likex1(1000,20000), data(20001), rand_num, ylim = 0.d0, x(1000)
REAL*8 :: hist(101,2) = 0.d0, temp(101,2), ord_data(20001)
INTEGER :: ndata = 1000 !<= Enter number of stars
REAL*8 :: TRGB ; PARAMETER (TRGB = 0.4d0) !<= Enter tip position 0< TRGB < 1
REAL}*8 :: f = 0.3d0 !<= Enter fraction background
!------------------------
REAL*8 :: u(1000)
!---------For Gaussian Prior on tip ---------
REAL*8 :: tip_expt = TRGB !Tip magnitude expected for structure
REAL*8 :: gauss_hwhm = 0.25d0 !Magnitudes on either side of expected tip magnitude to explore
REAL*8 :: gauss_expo = 4.d0 !Sharpness of edges of Gaussian prior profile
REAL*8 :: prior_sig, g(1000)
--
REAL*8 :: gap, maxgap, counts, d(1000)
REAL :: BG_counts, RGB_counts
END MODULE Global
PROGRAM spikes
USE Global ! Main Program
IMPLICIT NONE
CALL random_seed ! Insures stars are at different magnitudes each time
DO i = 1, ndata
    CALL random_number(rand_num)
    IF (rand_num .gt. TRGB * f) THEN
        CALL random_number(rand_num) !Draw ndata stars at
        rand_num = TRGB + (1.00d0 - TRGB) * rand_num!random from a luminosity
    ELSE !function with tip at TRGB
```

```
    CALL random_number(rand_num) !and background height = f.
    rand_num = TRGB * rand_num
    END IF
    rand_num = NINT(rand_num * 100.d0)
    data(i) = rand_num/ 100.d0
END DO
DO i = 1, 101
    hist (i,1) = REAL}(\textrm{i}-1)/100.d
    DO j = 1, ndata
        IF (data(j) .eq. hist(i,1)) THEN
            hist(i,2) = hist(i,2) + 1.d0
        END IF
    END DO
END DO
hist(101,2) = hist (101,2)* 2.e0 !Account for the bin width of the last bin
BG_counts = (REAL(ndata) * REAL(f))/100.e0 !Function height before step
RGB_counts =(\boldsymbol{REAL}(ndata) * REAL(1.d0 - f))/(\boldsymbol{REAL}(1.d0-TRGB) * 100.e0)
!/\ Function height
!|| after step
```



```
CALL pgbegin(0,'tempLF.ps/CPS',1,1)
CALL pgenv(0.0, 1.0, 0., 1.1*MAXVAL(REAL(hist (:,2))), 0, 0)
CALL pgbin (101, REAL(hist (:,1)), REAL(hist (:,2)), .true.)
CALL pglab('star_magnitude', 'counts,, ,,)
!|Plot model
!\/over LF
CALL pgsci(2)
CALL pgsls(3)
CALL pgslw(5)
CALL pgline(2, (/0.0 e0, REAL(TRGB) /), (/ BG_counts, BG_counts /))
CALL pgline(2, (/REAL(TRGB), 1.e0/), (/ BG_counts + RGB_counts, BG_counts + RGB_counts /))
CALL pgline(2, (/REAL(TRGB), REAL(TRGB)/), (/ BG_counts, BG_counts + RGB_counts /))
CALL pgslw(1)
CALL pgsls(1)
CALL pgsci(1)
!/\
!|
CALL pgend
1----------------------------------------------------------------
IF (ndata .ge. NINT(1/f)) THEN
    !CALL remove
END IF
```



```
CALL pgbegin(0,'?', 1, 1)
CALL pgenv(0.0, 1.0, 0., 1.1*MAXVAL(REAL(hist (:,2))), 0, 0)
CALL pgbin (101, REAL(hist (:,1)), REAL(hist (:,2)), .true.)
CALL pglab('star_magnitude', 'counts', 'BG-removed_Luminosity_Function')
CALL pgend
```

```
CALL pgbegin(0,'ind_like.ps/CPS', 1, 1)
```

DO i $=1$, ndata !
IF (MAXVAL(likex1(:,i)) .gt. ylim) THEN
ylim $=\operatorname{MAXVAL}($ likex $1(\cdot$, i $))$ !
END IF ! Change stars plotted here
END DO
CALL pgenv (0., 1., 0., $1.1 * \operatorname{REAL}(y l i m), 0,0)$ !
DO $\mathrm{i}=1$, ndata !
CALL pgsci $(\mathrm{i}+1)$
CALL pgbin (1000, REAL(posterior (: , 1)), REAL(likex1 (:,i)), .true.)
END DO
CALL pgsci (1)
CALL pglab('Proposed_tip_magnitude', 'Probability', ,')
CALL pgend

CALL pgbegin(0, post_dis.ps/CPS', 1, 1)
CALL pgenv $(0.0,1.0,0 ., \quad 1.1 * \operatorname{MAXVAL}(\operatorname{REAL}($ posterior $(:, 2))), 0,0)$
CALL pgbin (1000, $\mathbf{R E A L}($ posterior $(:, 1)), \mathbf{R E A L}($ posterior (:,2)), , true.)
CALL pglab('Proposed_tip_magnitude', 'Probability', ,')
CALL pgend
END PROGRAM spikes

```
SUBROUTINE u_prior !
USE Global !Generates Uniform prior function -> i.e. u = 1
IMPLICIT NONE
DO i = 1, 1000
    x(i) = REAL(i) / 1000.d0
END DO
!---_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_
DO i = 1, 1000
    u(i) = 1.d0
END DO
CALL pgbegin(0,'?', ,1,1)
CALL pgenv(0., 1., 0., 1.1, 0, 0)
CALL pgbin (1000, REAL(x), REAL(u), .true.)
CALL pglab('x', 'y', 'Uniform_Prior_Applied')
CALL pgend
END SUBROUTINE u_prior
```

$\qquad$

```
SUBROUTINE g_prior !
USE Global !Generates Gaussian prior function -> parameters of Gaussian changed in MODULE Global.
IMPLICIT NONE !
DO i = 1, 1000
    x(i) = REAL(i) / 1000.d0
END DO
!-_-_-_-_-_-_-_Gaussian Prior --_-_-_-_-_-_-_
prior_sig = ABS(gauss_hwhm ** (0.5d0 * gauss_expo))
DO i = 1, 1000
    g(i) = exp((-(\boldsymbol{REAL}(x(i)) - tip_expt)**gauss_expo)/(2.d0*(prior_sig)**2.d0))
END DO
CALL pgbegin(0,'?', , , 1)
CALL pgenv(0., 1., 0., 1.1, 0, 0)
CALL pgbin (1000, REAL(x), REAL(g), .true.)
CALL pglab('magnitude', 'weight', 'Gaussian」Prior\iotaApplied')
CALL pgend
END SUBROUTINE g_prior
SUBROUTINE d_prior !For applying a density prior. A window is chosen with width equal to the greatest separation between any
USE Global !2 data points in the array "data." The window then slides across the LF at 1-bin increments and the number
IMPLICIT NONE !of stars lying within the window is placed in the bin corresponding to either the LHS, RHS
!or middle of the window.
DO i = 1, 1000
    x(i) = REAL(i) / 1000.d0
END DO
```

```
CALL indexx(ndata, REAL(data), indx)
DO i = 1, ndata
    ord_data(i) = data(indx(i))
END DO
maxgap = 0.d0
DO i = 2, ndata
    gap = ord_data(i) - ord_data(i - 1)
    IF (gap . gt . maxgap) THEN
        maxgap = gap
    END IF
END DO
maxgap = NINT(maxgap *100.d0)
!-----Density applied to left edge of window
DO i = 1, 101 - NINT(maxgap)
    counts = 0.d0
    DO j = i, i + NINT(maxgap)
        counts = counts + hist(j,2)
    END DO
    d(10*i - 9:10*i) = counts
    IF (i .eq. 101 - NINT(maxgap)) THEN
        d(10*i:1000) = counts
    END IF
END DO
d = d / (MAXVAL(d))
--------Plot density prior ---------
CALL pgbegin(0,'?', 1, 1)
CALL pgenv(0., 1., 0., 1.1 * MAXVAL(REAL(d)), 0, 0)
CALL pgbin (1000, REAL(x), REAL(d), .true.)
CALL pglab('magnitude', 'weight', 'Density_Prior_Applied')
CALL pgend
END SUBROUTINE d_prior
SUBROUTINE remove !Remove f*ndata data points
USE Global !spread randomly over magnitude
IMPLICIT NONE !space from data array.
i = 0
DO
    CALL random_number(rand_num)
    rand_num = NINT(rand_num * 100) + 1.d0
    IF (hist(NINT(rand_num), 2) .ge. 1.d0) THEN
        i = i +1
        hist(NINT(rand_num), 2) = hist(NINT(rand_num), 2) - 1.d0
    END IF
    IF (i .ge. NINT(f * ndata)) THEN
        exit;
    END IF
END DO
temp = hist
```

```
ndata = ndata - (INT(f * ndata) + 1)
data = 0.d0
j = 0
DO i = 1, 101
    DO WHILE (temp(i,2) .gt . 0.d0)
        j = j + 1
        data(j) = temp(i, 1)
        temp(i,2) = temp(i,2) - 1.d0
    END DO
END DO
END SUBROUTINE remove
```


# Chapter Three Programs 

BayesianTRGB_ANDI.f95 ..... 162
MCMCTRGBTester2.f95 ..... 182
BayesianTRGBTestPlotterMCMC.f95 ..... 186

## Program: BayesianTRGB_ANDI.f95

Creation Date: 23 September 2010 (first version 6 Mar 2010)
Relevant Sections: 2.4, 3
Notes: This program lies at the heart of the material presented in Paper I (Ch. 3) and hence is described there in much more detail. Note also that $\S 2$ of Paper II (presented in Ch . 4) also provides a useful summary as to its workings. The program is an example of my original TRGB finding algorithm. At this stage I had not yet generalized the code so that the parameters for different objects could be fed in, and hence each object had its own separate code - that shown here is for Andromeda I. In summary, a circular field is taken centered on user-specified coordinates and all stars within that field are then plotted on a ColourMagnitude Diagram. The user then provides the coordinates for the corners of a polygon to be used as a colour-cut to isolate the stars of the object's Red Giant Branch. The colour-cut should extend at least half a magnitude brightward of the estimated TRGB magnitude to give a reasonable portion of pure background luminosity function (LF) for the algorithm to fit, and it should span an equal colour range as a function of magnitude so as not to distort the LF. The user must also provide a 'background field' from which the algorithm generates a LF which it then fits with a polynomial to give the functional form of the background component of the model LF. The same colour-cut is imposed on the background field as was chosen for the object or 'signal' field. By calculating the number of stars in both the signal and background fields and dividing by their respective areas, the expected ratio of the two components in the signal field's LF is determined and the components are scaled accordingly in the model LF generated. An MCMC algorithm is then used to find the parameters of the model which best fit the data and posterior distributions of these parameters are plotted. The object's distance posterior distribution can then be determined by sampling the distribution in the tip magnitude along with those for the uncertainty in the absolute magnitude of the tip and the extinction. This is done using another, purpose-written program, a version of which can be found in Appendix D ('Multi_MCMC_ResultPlotter.f95').

```
MODULE Global !Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, j, k, l, eval, idum = -9999, it, nit
INTEGER :: ndata_max, nsamples, binspm, nbins, cmod_nbins, ghw, mm, ios
```

```
PARAMETER (ndata_max = 20000000, nsamples = 100)
PARAMETER (binspm = 100)
PARAMETER (nbins = 8* binspm + 1)
PARAMETER (nit = 3000000)
INTEGER :: ndata, ndata2
INTEGER :: d1, d2, d3, d4
REAL*8 :: blim, flim, pi
PARAMETER (blim = 19.5 d0)
PARAMETER ( flim = 23.5d0)
PARAMETER (pi = ACOS(-1.e0))
INTEGER :: blimBins = INT(REAL((blim - 18.d0) * binspm)) + 1
INTEGER :: flimBins = INT(REAL((flim - 18.d0) * binspm)) + 1
REAL*8 :: randnum1, randnum2, randnum3, r1, r2, spotR, hb = 0.005 d0
REAL*8 :: model(nbins,2), cmodel(nbins,2), magnitude(ndata_max)
REAL*8 :: histo_fine(nbins,2), histo_coarse(INT(0.25*(nbins - 1.d0)) + 1,2)
REAL*8 :: data(ndata_max), cumulative_cmodel(nbins,2), f, f_hold, bfm(nbins)
REAL*8 :: mag_tip, mag, mag_cutoff = 24.e0, a
REAL*8 :: area, area2
REAL*8 :: modelnoise(nbins,2), noise(nbins) = 0.d0
REAL*8 :: kernel(nbins,2) = 0.e0, scale, uplim, lowlim, gx
REAL}*8 :: temp(nbins,2) = 0.e0, 
REAL*8 :: logL, prob, LikeA, LikeB
REAL*8 :: tip(nsamples), tip_ord(nsamples), maxlogL(nsamples) = -999999999999.
REAL*8 :: tip_rec, tip_offset, tip_psigma, tip_msigma, Toffset_kpc, Tsigma_kpc
REAL*8 :: f_offset, tip_kpc, kpc_perr, kpc_merr, f_sigma, a_offset, a_sigma
REAL*8 :: f_rec, a_rec, tip_counts, f_counts, a_counts
REAL*8 :: tipminsig, tiplusig, fminsig, fplusig, aminsig, aplusig
REAL*8 :: mcounts, pcounts
REAL*8 :: x1(nit), x2(nit), x3(nit), p(3), time(nit), r
REAL*8 :: post_y1(10*(nbins-1)+1) = 0.d0, post_x 1 (10*(nbins - 1)+1), mlim
REAL*8 :: d_blim, bg_blim, d_flim, bg_flim
REAL*8 :: post-y2(nbins) = 0.d0, post.x2(nbins)
REAL*8 :: post_y3(2*nbins - 1) = 0.d0, post_x 3 (2*nbins - 1)
REAL*8 :: PPD_peak, Best_Combo(6)
CHARACTER :: argv*10, field * 30, ch 1*9, ch2*9, ch 3*9, ch4*9, ch5*9, string*60
!----------------------or reading in PAndAS data
INTEGER :: iCCDt, clsg, clsi, ifieldt, iacc_t
REAL*4 :: xgt, ygt, g, dg, im, dim, xki_t, eta_t, FeH_phot_t, diff_tip_t, E_BV_t
REAL*8 :: ra_t, de_
REAL*4 :: dummy
REAL*4 :: mag_g(ndata_max), mag_i(ndata_max), xki(ndata_max), eta(ndata_max)
REAL*4 :: g_min_i(ndata_max), mag_i_poly(ndata_max), g_min_i_poly(ndata_max)
REAL}*4 :: gmi
!-- Additional parameters for calculating background stats --
INTEGER :: bg_ndata, bg_ndata2, bg_ndata3
REAL*4 :: bg_mag_g(ndata_max), bg_mag_i(ndata_max), bg_xki(ndata_max), bg_eta(ndata_max)
REAL*4 :: bg_g_min_i(ndata_max), bg_mag_i_poly(ndata_max), bg_g_min_i_poly(ndata_max)
REAL*4 :: bg_gmi
REAL*8 :: bg_data(ndata_max)
    !--SVD fitting of background--
    INTEGER ma, mp, np, ndat
    PARAMETER (ndat = INT (0.25*(nbins - 1.d0)) + 1)
    PARAMETER ( np = 8
    PARAMETER (mp = ndat)
    PARAMETER (ma = np)
    REAL :: chisq, ay(ma), sig(ndat), u(mp,np), v(np,np), w(np), xa(ndat), ya(ndat)
REAL :: xt(ndat), yt(ndat)
```

```
    REAL*8 :: bg_histo_coarse(ndat,2)
    EXTERNAL :: func
!--Additional parameters for specifying object coordinates --
INTEGER :: Jop
REAL*8 :: XIop, ETAop
REAL*8 :: RAh, RAm, RAs, DecD, DecM, DecS, RA_rad, Dec_rad
REAL*8 :: tpRAh, tpRAm, tpRAs, tpDecD, tpDecM, tpDecS, tpRA_rad, tpDec_rad
!---------------------------------------------------
INTEGER :: bg_stars, sig_stars
REAL*8 :: bg_area, sig_area
REAL*8 :: known_f, bg_stars_in_sig_field
REAL*8 :: sig_field_radius = 0.1d0, bg_low_xi = - 10.d0, bg_up_xi = 10.d0
END MODULE Global
PROGRAM BayesianTRGBsatellite !Master program
USE Global
IMPLICIT NONE
WRITE (field,*) 'AndromedaI'
string = TRIM(ADJUSTL(field)) // '/results.dat,
OPEN(3, file=TRIM(ADJUSTL(string)), status = 'unknown')
WRITE (3,*) "Field_Name:", field
CALL positionFinder !
CALL random_seed !
CALL M31DataReader !
CALL SVDFitter
CALL NoiseMake
!CALL NoisePlot !CALL
CALL MCMC !
!CALL PosteriorPlot !SUBROUTINES
CALL TipAndSigma
CALL PosteriorPlot
CALL OtherPlots
CALL DataHist
WRITE (3, '(3a11)') "ьььtip \lrcornermag:", " \lrcornerь+ьsigma:ьь", "„ь-ьsigma:\iota" !
WRITE (3, '(3F10.3)') tip_rec, tip_psigma, tip_msigma! !
WRITE (3, '(2a11)') ",_-_f:__-_-_-"," „___sigma:__""
WRITE (3, '(2F10.3)') f_rec, f_sigma !Write results
WRITE (3, '(2a11)') "___a:__-_-_",",__ssigma:__">
WRITE (3, '(2F10.3)') a_rec, a_sigma !
WRITE (3,*) "Distance}\lrcorner=", REAL(tip_kpc), "kpc"
WRITE (3,*) "Error\iota=ь+", kpc_perr, "kpc\_-", kpc_merr, "kpc" !
END PROGRAM BayesianTRGB satellite
SUBROUTINE PositionFinder ! Converts object's position in RA and Dec into
USE Global !its position on the M31 tangent plane
IMPLICIT NONE
```



```
RAh = 0.d0 !
```

RAh = 0.d0 !
RAm = 45.d0 !
RAm = 45.d0 !
RAs = 39.8d0 ! Object coordinates
RAs = 39.8d0 ! Object coordinates
DecD = 38.d0 !in RA and Dec
DecD = 38.d0 !in RA and Dec
DecM = 2.d0 !
DecM = 2.d0 !
DecS = 28.d0 !
DecS = 28.d0 !
tpRAh = 0.d0
tpRAh = 0.d0
tpRAm = 42.d0 !
tpRAm = 42.d0 !
tpRAs = 44.33d0 !Tangent point (i.e. M31)
tpRAs = 44.33d0 !Tangent point (i.e. M31)
tpDecD = 41.d0 !coordinates in RA and Dec
tpDecD = 41.d0 !coordinates in RA and Dec
tpDecM = 16.d0 !
tpDecM = 16.d0 !
tpDecS = 7.5 d0
tpDecS = 7.5 d0
!|Perform
!|Perform
! \/Conversion
! \/Conversion
RA_rad = (pi/180.d0) * (RAh * 15.d0 + RAm * (15.d0/60.d0) + RAs * (15.d0/3600.d0))
RA_rad = (pi/180.d0) * (RAh * 15.d0 + RAm * (15.d0/60.d0) + RAs * (15.d0/3600.d0))
Dec_rad =(pi/180.d0)*(DecD + DecM/60.d0 + DecS/3600.d0)
Dec_rad =(pi/180.d0)*(DecD + DecM/60.d0 + DecS/3600.d0)
tpRA_rad = (pi/180.d0) *(tpRAh * 15.d0 + tpRAm * (15.d0/60.d0) + tpRAs * (15.d0/3600.d0))
tpRA_rad = (pi/180.d0) *(tpRAh * 15.d0 + tpRAm * (15.d0/60.d0) + tpRAs * (15.d0/3600.d0))
tpDec_rad = (pi/180.d0) *(tpDecD + tpDecM/60.d0 + tpDecS/3600.d0)
tpDec_rad = (pi/180.d0) *(tpDecD + tpDecM/60.d0 + tpDecS/3600.d0)
CALL sla_DS2TP (RA_rad, Dec_rad, tpRA_rad, tpDec_rad, XIop, ETAop, Jop)
CALL sla_DS2TP (RA_rad, Dec_rad, tpRA_rad, tpDec_rad, XIop, ETAop, Jop)
!/\
!/\
!|
!|
XIop = XIop * (180.d0/pi) !tangent plane coordinates
XIop = XIop * (180.d0/pi) !tangent plane coordinates
ETAop = ETAop * (180.d0/pi) !(i.e. PAndAS xi and eta)
ETAop = ETAop * (180.d0/pi) !(i.e. PAndAS xi and eta)
WRITE (3,*) "C.O.F.\&Xi\lrcorner=", XIop, "C.O.F.\curvearrowleftEta ="", ETAop
WRITE (3,*) "C.O.F.\&Xi\lrcorner=", XIop, "C.O.F.\curvearrowleftEta ="", ETAop
END SUBROUTINE PositionFinder
END SUBROUTINE PositionFinder
SUBROUTINE M31DataReader !The field to be analysed is specified here
SUBROUTINE M31DataReader !The field to be analysed is specified here
USE Global
USE Global
IMPLICIT NONE
IMPLICIT NONE
OPEN(1,file='../../../PANDAS/M31_unique_con.dat',form='unformatted',
OPEN(1,file='../../../PANDAS/M31_unique_con.dat',form='unformatted',
i = 0; j = 0
i = 0; j = 0
DO WHILLE (.true.)
DO WHILLE (.true.)
READ(1,IOSTAT=ios) ra_t, de_t,iCCDt,xgt,ygt, \& !Read in data
READ(1,IOSTAT=ios) ra_t, de_t,iCCDt,xgt,ygt, \& !Read in data
g,dg,clsg,im,dim,clsi,ifieldt,xki_t,eta_t, \& !from binary
g,dg,clsg,im,dim,clsi,ifieldt,xki_t,eta_t, \& !from binary
dummy, FeH_phot_t, diff_tip_t, E_BV_t,iacc_t !format data file
dummy, FeH_phot_t, diff_tip_t, E_BV_t,iacc_t !format data file
IF (ios.ne.0) exit
IF (ios.ne.0) exit
g=g-3.793*E_BV_t ! Extinction
g=g-3.793*E_BV_t ! Extinction
im=im-2.086*E_BV_t ! Corrections
im=im-2.086*E_BV_t ! Corrections
gmi = g - im
gmi = g - im
if (clsi.ne.-1 .and. clsi.ne.-2) cycle !Rejects
if (clsi.ne.-1 .and. clsi.ne.-2) cycle !Rejects
if (clsg.ne.-1 .and. clsg.ne.-2) cycle !non stars
if (clsg.ne.-1 .and. clsg.ne.-2) cycle !non stars
if (18.0.le.im.and.im.le.24.) then !Specifies
if (18.0.le.im.and.im.le.24.) then !Specifies
else !magnitude

```
    else !magnitude
```

```
    cycle !range to
    end if !include
    if (-2.5.le.FeH_phot_t.and.FeH_phot_t.le. - 1.5) then!
    else !Specifies metallicity
        !cycle !range to include
    end if !
    spotR = SQRT((ABS(eta_t - (ETAop))) **2 + (ABS(xki_t - (XIop))) **2)
    IF (spotR .lt. sig_field_radius) THEN
        i = i + 1
        IF (i . gt. ndata_max) exit
        mag_g(i)=g
        mag_i (i)=im
        g_min_i(i)=gmi !If all conditions are met, add star data to signal arrays
        xki(i)=xki_t !
        eta(i)=eta_t !
    ELSE IF (xki_t .ge. bg_low_xi .and. xki_t .le. bg_up_xi) THEN
        IF (xki_t . 1t. - 3.5 .or. xki_t .gt. 2.5) THEN
        IF (eta_t .ge. ETAop - 0.2d0 .and. eta_t .le. ETAop + 0.2 d0) THEN
            j = j + 1
            IF (j .gt. ndata_max) exit
            bg_mag_g(j)=g !
            bg_mag_i(j)=im !
            bg_g_min_i(j)=gmi !If all conditions are met, add star data to bckgrnd arrays
            bg_xki(j)=xki_t !
            bg_eta(j)=eta_t !
        END IF
        END IF
    END IF
END DO
ndata = i ; bg_ndata = j
sig_area = pi * (sig_field_radius ** 2.d0) !Calculate area of signal field
bg_area = 0.4d0 * (bg_up_xi - bg_low_xi) - 2.4d0 !Calculate area of bacground field
DO i = 1, ndata
    data(i) = mag_i(i) !Object stars before colour cut
END DO
DO j = 1, bg_ndata
    bg_data(j) = bg_mag_i(j) !Background stars before colour cut
END DO
CALL M31DataPlotter
END SUBROUTINE M31DataReader
```

$\qquad$

```
SUBROUTINE M31DataPlotter !Plot object and backgound field star positions
USE Global !on the sky and CMDs for each object (g-i vs. i)
IMPLICIT NONE
```

```
251
252
54
255
```

string = TRIM(ADJUSTL(field)) // '/ sig_field.ps/CPS,

```
string = TRIM(ADJUSTL(field)) // '/ sig_field.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MAXVAL(xki, mask = xki . ne. 0.), MINVAL(xki, mask = xki . ne. 0.), &
CALL pgenv(MAXVAL(xki, mask = xki . ne. 0.), MINVAL(xki, mask = xki . ne. 0.), &
MINVAL(eta, mask = eta .ne. 0.), MAXVAL(eta, mask = eta .ne. 0.), 1, 0)
MINVAL(eta, mask = eta .ne. 0.), MAXVAL(eta, mask = eta .ne. 0.), 1, 0)
CALL pgpt (ndata, xki, eta, -1)
CALL pgpt (ndata, xki, eta, -1)
CALL pglab('\(0640)\bullet(degrees)', '\(0633)\bullet(degrees)', ,')
CALL pglab('\(0640)\bullet(degrees)', '\(0633)\bullet(degrees)', ,')
CALL pgend
CALL pgend
----------------------------------------------------------
----------------------------------------------------------
string = TRIM(ADJUSTL(field)) // ,/sig_cmd.ps/CPS'
string = TRIM(ADJUSTL(field)) // ,/sig_cmd.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MINVAL(g_min_i, mask = g_min_i .ne. 0.), MAXVAL(g_min_i), &
CALL pgenv(MINVAL(g_min_i, mask = g_min_i .ne. 0.), MAXVAL(g_min_i), &
MAXVAL(mag_i), MINVAL(mag_i, mask = mag_i . ne. 0.), 0, 0)
MAXVAL(mag_i), MINVAL(mag_i, mask = mag_i . ne. 0.), 0, 0)
CALL pgpt (ndata, g_min_i, mag_i, -1)
CALL pgpt (ndata, g_min_i, mag_i, -1)
CALL pglab('(g--_i)\d0\u', 'i\d0\u',,',')
CALL pglab('(g--_i)\d0\u', 'i\d0\u',,',')
CALL PolySelect !Apply colour cut to signal CMD
CALL PolySelect !Apply colour cut to signal CMD
CALL pgend
CALL pgend
-_----------------------Bckgrnd-Field
-_----------------------Bckgrnd-Field
string = TRIM(ADJUSTL(field)) // '/bg_field.ps/CPS'
string = TRIM(ADJUSTL(field)) // '/bg_field.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MAXVAL(bg_xki, mask = bg_xki .ne. 0.), &
CALL pgenv(MAXVAL(bg_xki, mask = bg_xki .ne. 0.), &
    MINVAL(bg_xki, mask = bg_xki .ne. 0.), &
    MINVAL(bg_xki, mask = bg_xki .ne. 0.), &
    MINVAL(bg_eta, mask = bg_eta .ne. 0.), &
    MINVAL(bg_eta, mask = bg_eta .ne. 0.), &
    MAXVAL(bg_eta, mask = bg_eta .ne. 0.), 1, 0)
    MAXVAL(bg_eta, mask = bg_eta .ne. 0.), 1, 0)
CALL pgpt (bg_ndata, bg_xki, bg_eta, -1)
CALL pgpt (bg_ndata, bg_xki, bg_eta, -1)
CALL pglab('\(0640)_(degrees)', '\(0633)_(degrees)',,',)
CALL pglab('\(0640)_(degrees)', '\(0633)_(degrees)',,',)
CALL pgend
CALL pgend
                                    Bckgrnd-CMD
                                    Bckgrnd-CMD
string = TRIM(ADJUSTL(field)) // ,/bg_cmd.ps/CPS,
string = TRIM(ADJUSTL(field)) // ,/bg_cmd.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MINVAL(bg_g_min_i, mask = bg_g_min_i .ne. 0.), &
CALL pgenv(MINVAL(bg_g_min_i, mask = bg_g_min_i .ne. 0.), &
    MAXVAL(bg_g_min_i), MAXVAL(bg_mag_i), &
    MAXVAL(bg_g_min_i), MAXVAL(bg_mag_i), &
    MINVAL(bg_mag_i, mask = bg_mag_i .ne. 0.), 0, 0)
    MINVAL(bg_mag_i, mask = bg_mag_i .ne. 0.), 0, 0)
CALL pgpt (bg_ndata, bg_g_min_i, bg_mag_i, -1)
CALL pgpt (bg_ndata, bg_g_min_i, bg_mag_i, -1)
CALL pglab ('(g_-_i)\d0\u', 'i\d0\u',,',')
CALL pglab ('(g_-_i)\d0\u', 'i\d0\u',,',')
CALL PolySelect
CALL PolySelect
CALL pgend
CALL pgend
---------------Input selected data into 'data'-------------
---------------Input selected data into 'data'-------------
!|Removes stars from the signal and backround fields that
!|Removes stars from the signal and backround fields that
\/lie outside of the chosen colour cut
\/lie outside of the chosen colour cut
data = 0.d0 ; xki = 0.d0 ; eta = 0.d0 ; d_blim = 100.d0 ; d_flim = 0.d0
data = 0.d0 ; xki = 0.d0 ; eta = 0.d0 ; d_blim = 100.d0 ; d_flim = 0.d0
DO i = 1, ndata2
DO i = 1, ndata2
    data(i) = mag_i_poly(i)
    data(i) = mag_i_poly(i)
    IF (data(i) . lt. d_blim) THEN
    IF (data(i) . lt. d_blim) THEN
        d_blim = data(i)
        d_blim = data(i)
    END IF
```

    END IF
    ```
```

    IF (data(i) .gt. d_flim) THEN
        d_flim = data(i)
    END IF
    END DO
bg_data = 0.d0 ; bg_blim = 100.d0 ; bg_flim = 0.d0
DO i = 1, bg_ndata2
bg_data(i) = bg_mag_i_poly(i)
IF (bg_data(i) .lt. bg_blim) THEN
bg_blim = bg_data(i)
END IF
IF (bg_data(i) .gt. bg_flim) THEN
bg_flim = bg_data(i)
END IF
END DO
!/\
!|
!---Set parameters for calculation of background height ---
sig_stars = ndata2 !Total number of stars in signal field
bg_stars = bg_ndata3 !Number of stars in background field
bg_stars_in_sig_field= REAL(bg_stars) * (sig_area/bg_area)
!Number of Background stars in signal field
WRITE (3,*) "Number_of_data_points:", sig_stars
WRITE (3,*) "SNR:", (REAL(sig_stars) - bg_stars_in_sig_field) / bg_stars_in_sig_field
!-_-_-----_
DO i = 1, INT(0.25*(nbins - 1.d0)) + 1
bg_histo_coarse(i,1) = 18.d0 + (i - 1.d0)/REAL}(0.25* binspm)
END DO
DO i = 1, bg_ndata2
bg_histo_coarse(INT((bg_data(i) - 18.d0) *0.25*binspm) + 1, 2) = \&
bg_histo_coarse(INT((bg_data(i) - 18.d0) *0.25*binspm) + 1, 2) + 1.d0
END DO
!|| Fill empty bright edge of array with
!\/ artificial data for improved fitting
DO i = 1, INT((bg_blim - 18.d0) * REAL(binspm/4.d0)) + 4
bg_histo_coarse(i,2) = bg_histo_coarse(INT((bg_blim - 18.d0) * REAL(binspm/4.d0)) + 4, 2)
END DO
END SUBROUTINE M31DataPlotter
SUBROUTINE SVDFitter !Fits a polynomial to the background luminosity function
USE Global
IMPLICIT NONE
INTEGER :: ntmp
xa = bg_histo_coarse(:,1)
ya}=\mp@code{bg_histo_coarse (:,2)
xt = xa
yt = ya
sig}=1.e

```
```

3 7 3
374
*
376

```
! Shift the array in steps of 1 until the first element does not contain a zero
```

! Shift the array in steps of 1 until the first element does not contain a zero
shiftloop: do
shiftloop: do
xt = cshift(xt,1)
xt = cshift(xt,1)
yt}=\operatorname{cshift(yt,1)
yt}=\operatorname{cshift(yt,1)
if ( yt(1) > 0.1 ) exit shiftloop
if ( yt(1) > 0.1 ) exit shiftloop
end do shiftloop
end do shiftloop
ntmp = 0
ntmp = 0
countloop: do i = 1 , ndat
countloop: do i = 1 , ndat
if ( yt(i) > 0.1 ) then
if ( yt(i) > 0.1 ) then
ntmp = ntmp + 1
ntmp = ntmp + 1
else
else
exit countloop
exit countloop
end if
end if
end do countloop
end do countloop
xt = xt - 21.
xt = xt - 21.
CALL svdfit(xt,yt,sig,ntmp-1,ay,ma,u,v,w,mp,np,chisq,funcs)
CALL svdfit(xt,yt,sig,ntmp-1,ay,ma,u,v,w,mp,np,chisq,funcs)
CALL BG_DataHist
CALL BG_DataHist
END SUBROUTINE SVDFitter
END SUBROUTINE SVDFitter
SUBROUTINE BG_DataHist ! Plots background luminosity function together with
SUBROUTINE BG_DataHist ! Plots background luminosity function together with
USE Global !fitted polynomial
USE Global !fitted polynomial
IMPLICIT NONE
IMPLICIT NONE
bfm = 0.d0
bfm = 0.d0
DO i = 1, ndat
DO i = 1, ndat
DO j = 1 , np
DO j = 1 , np
bfm(i)}=\operatorname{bfm}(\textrm{i})+\textrm{ay}(\textrm{j})*(xa(i)-21.)**(j -1
bfm(i)}=\operatorname{bfm}(\textrm{i})+\textrm{ay}(\textrm{j})*(xa(i)-21.)**(j -1
END DO
END DO
END DO

```
END DO
```




```
string = TRIM(ADJUSTL(field)) // '/bckgrdfit.ps/CPS,
```

string = TRIM(ADJUSTL(field)) // '/bckgrdfit.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(19.5, 23.5, 0., 1.1*MAXVAL(REAL(bg_histo_coarse(:,2))), 0, 0)
CALL pgenv(19.5, 23.5, 0., 1.1*MAXVAL(REAL(bg_histo_coarse(:,2))), 0, 0)
CALL pgbin (ndat, REAL(bg_histo_coarse (:, 1)), REAL(bg_histo_coarse (:, 2)), . true.)
CALL pgbin (ndat, REAL(bg_histo_coarse (:, 1)), REAL(bg_histo_coarse (:, 2)), . true.)
CALL pgsci(2)
CALL pgsci(2)
CALL pgline (ndat, xa, REAL(bfm))
CALL pgline (ndat, xa, REAL(bfm))
CALL pgsci(1)
CALL pgsci(1)
CALL pglab('i\d0\u', 'counts',,',)
CALL pglab('i\d0\u', 'counts',,',)
CALL pgend
CALL pgend
END SUBROUTINE BG_DataHist
END SUBROUTINE BG_DataHist
SUBROUTINE MCMC !The master Markov Chain MonteCarlo routine
SUBROUTINE MCMC !The master Markov Chain MonteCarlo routine
USE Global !creates a new model at each iteration and then compares
USE Global !creates a new model at each iteration and then compares
IMPLICIT NONE ! the quality of the fit to the data

```
IMPLICIT NONE ! the quality of the fit to the data
```

```
    !*** Most subroutines are called from 'MCMC' ***
REAL*8 :: gasdev
known_f = (REAL(bg_stars) * sig_area)/(REAL(sig_stars) * bg_area)
x1(1) = 20.88d0; x2(1) = known_f; x3(1) = 0.27d0 ; time(1) = 1
mag_tip = x1(1) ; f = x2(1) ; a = x3(1)
CALL ModelMake !Make model and
CALL Convolution !evaluate goodness of fit
CALL Loglike !for initial parameter choices
LikeA = logL
LikeB = 0.d0
x1(2) = x1(1) ; x2(2) = x2(1); x3(2) = x3(1)
Best_Combo(6) = -9.d99
string = TRIM(ADJUSTL(field)) // '/MCMC_steps.dat,
OPEN(2, file=TRIM(ADJUSTL(string)), status = 'unknown')
```



```
DO it = 2, nit
    time(it) = it
    p(1) = x1(it) + 0.03d0*gasdev(idum) !Gaussian weighted steps from initial
    p(2) = x2(it)! + 0.02d0*gasdev(idum) !parameters for the tip magnitude (p (1))
    p(3) = x3(it) + 0.02d0*gasdev(idum) !noise ratio (p(2)) and slope (p(3))
    IF (p(1) .lt. blim .or. p(1) .gt. flim) THEN
        r = 0.d0
    else IF (p(2) .le. 0.d0 .or. p(2) .ge. 1.d0) THEN !Restrictions on
        r = 0.d0 !whether proposed step
    else IF (p(3) . le. 0.d0 .or. p(3) .ge. 2.d0) THEN !
        r = 0.d0
    else
        mag_tip = p(1) ; f = p(2) ; a = p(3)
        CALL ModelMake !Make model and
        CALL Convolution !evaluate the
        CALL Loglike !goodness of fit
        LikeB = logL
        r = 10**(LikeB-LikeA)
    end IF
    CALL random_number(randnum3)
    IF (it . lt. nit) THEN
    IF (randnum3 . le. r) THEN
        x1(it+1) = p(1); x2(it+1)=p(2); x3(it+1)= p(3)
        likeA = likeB !Decide whether
    ELSE !to take step
        x1(it+1) = x1(it) ; x2(it+1) = x2(it) ; x3(it+1) = x3(it) !or not
        likeA = likeA
    END IF
    END IF
    post_y1(INT((x1(it) - 18.d0)*10*binspm + 1)) = &
    post_y1(INT((x1(it) - 18.d0)*10*binspm + 1)) + 1.d0
    post_y2(INT(x2(it) * (nbins - 1)) + 1) = & !Generate posterior plot
    post_y2(INT(x2(it) * (nbins - 1)) + 1) + 1.d0 !for mag_tip, f and a
    post_y3(INT(x3(it) * (nbins - 1)) + 1) = &
    post_y3(INT(x3(it) * (nbins - 1)) + 1) + 1.d0 !
    WRITE (2, '(6F16.5)') time(it), x1(it), x2(it), x3(it), LikeA, LikeB
```

```
    IF (LikeB .gt. Best_Combo(6)) THEN
        Best_Combo(1) = time(it) ; Best_Combo(2) = p(1)
        Best_Combo(3) = p(2) ; Best_Combo(4) = p(3)
        Best_Combo(5) = LikeA ; Best_Combo(6) = LikeB
    END IF
END DO
WRITE (2, '(6F16.5)') Best_Combo(1), Best_Combo(2), Best_Combo(3), &
        Best_Combo(4), Best_Combo(5), Best_Combo(6)
DO i = 1, 10*(nbins - 1)+1
    post_x1(i) = 18.d0 + (REAL(i) - 1.d0)/REAL(10*binspm)!
END DO
DO i = 1, nbins !x-values of posterior
    post_x2(i)=(\mathbf{REAL}(i) - 1.d0)/REAL(nbins - 1) !histograms created above
END DO
DO i = 1, 2*nbins - 1
    post_x3(i) = (REAL(i) - 1.d0)/REAL(nbins - 1)
END DO
END SUBROUTINE MCMC
SUBROUTINE PosteriorPlot
USE Global
IMPLICIT NONE
post_yl = post_y1/nit ; post-y2 = post_y 2/nit ; post_y 3 = post_y 3/nit
!-----------------------------Plots mag_tip posterior plot
string = TRIM(ADJUSTL(field)) // ,/mag_tip_postplot.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(REAL(MINVAL(x1)) -0.01, REAL(MAXVAL(x 1)) +0.01,&
    0., 1.1*REAL(MAXVAL(post_y1)), 0, 0)
CALL pgbin (10*(nbins-1)+1, REAL(post_x1), REAL(post_y1),.true.)
CALL pglab('Proposed_i \d0\u_tip_magnitude',, Probability,,,',)
CALL pgend
-------------------------------->ots f and a posterior plots
string = TRIM(ADJUSTL(field)) // '/f_and_a_postplot.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
IF (MAXVAL(post_y 3) .ge. MAXVAL(post_y2)) THEN
    CALL pgenv(0., 2., 0., 1.1*REAL(MAXVAL(post_y3)), 0, 0)
ELSE
    CALL pgenv(0., 2., 0., 1.1*REAL(MAXVAL(post-y2)), 0, 0)
END IF
CALL pgsci(2)
CALL pgbin (nbins, REAL(post_x 2), REAL(post-y2),.true.)
CALL pgsci(3)
CALL pgbin (2*nbins - 1, REAL(post_x 3), REAL(post-y 3),.true.)
CALL pgsci(1)
CALL pglab('Proposed_value', 'Probability:_f_(red)_a_(green)', ',')
```

```
556
```

CALL pgend

```
CALL pgend
post_y1 = post_y1*nit ; post_y2 = post_y 2*nit ; post_y3 = post_y 3*nit
post_y1 = post_y1*nit ; post_y2 = post_y 2*nit ; post_y3 = post_y 3*nit
END SUBROUTINE PosteriorPlot
END SUBROUTINE PosteriorPlot
SUBROUTINE OtherPlots
SUBROUTINE OtherPlots
USE Global
USE Global
IMPLICIT NONE
IMPLICIT NONE
--------------------------Variation of 'mag_tip' with iteration #
--------------------------Variation of 'mag_tip' with iteration #
string = TRIM(ADJUSTL(field)) // '/mag_tip_val_vs_it.ps/CPS'
string = TRIM(ADJUSTL(field)) // '/mag_tip_val_vs_it.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(0., REAL(nit), REAL(MINVAL(x1)) - 0.01, REAL(MAXVAL(x1)) +0.01, 0, 0)
CALL pgenv(0., REAL(nit), REAL(MINVAL(x1)) - 0.01, REAL(MAXVAL(x1)) +0.01, 0, 0)
CALL pgline (nit, REAL(time), REAL(x1))
CALL pgline (nit, REAL(time), REAL(x1))
CALL pglab('Iteration_number', 'Proposed_i\d0\u_tip_magnitude',,',)
CALL pglab('Iteration_number', 'Proposed_i\d0\u_tip_magnitude',,',)
CALL pgend
CALL pgend
------------------------- ariation of 'f' and 'a' with iteration #
------------------------- ariation of 'f' and 'a' with iteration #
string = TRIM(ADJUSTL(field)) // '/f_and_a_val_vs_it.ps/CPS,
string = TRIM(ADJUSTL(field)) // '/f_and_a_val_vs_it.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(0., REAL(nit), 0., 2., 0, 0)
CALL pgenv(0., REAL(nit), 0., 2., 0, 0)
CALL pgsci(2)
CALL pgsci(2)
CALL pgline (nit, REAL(time), REAL(x2))
CALL pgline (nit, REAL(time), REAL(x2))
CALL pgsci(3)
CALL pgsci(3)
CALL pgline (nit, REAL(time), REAL(x3))
CALL pgline (nit, REAL(time), REAL(x3))
CALL pgsci(1)
```

CALL pgsci(1)

```


```

CALL pgend

```
CALL pgend
!-_-_-_-_-_-_-_-_-__-_malues of 'f' for each value of 'mag_tip'
!-_-_-_-_-_-_-_-_-__-_malues of 'f' for each value of 'mag_tip'
string = TRIM(ADJUSTL(field)) // '/f_vs_mag_tip.ps/CPS'
string = TRIM(ADJUSTL(field)) // '/f_vs_mag_tip.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)), 1, 1)
CALL pgbegin(0,TRIM(ADJUSTL(string)), 1, 1)
CALL pgenv (0.99*REAL(MINVAL(x 1)), 1.01*REAL(MAXVAL(x 1)), 0.9*REAL(MINVAL(x2 )), 1.1*REAL(MAXVAL(x2 )), 0, 0)
CALL pgenv (0.99*REAL(MINVAL(x 1)), 1.01*REAL(MAXVAL(x 1)), 0.9*REAL(MINVAL(x2 )), 1.1*REAL(MAXVAL(x2 )), 0, 0)
CALL pgpoint (nit, REAL(x1), REAL(x2), -1)
CALL pgpoint (nit, REAL(x1), REAL(x2), -1)
CALL pglab('Proposed_i\d0\u_tip_magnitude', 'Proposed_value_of_f',,',')
CALL pglab('Proposed_i\d0\u_tip_magnitude', 'Proposed_value_of_f',,',')
CALL pgend
```

CALL pgend

```


```

string = TRIM(ADJUSTL(field)) // '/a_vs_mag_tip .ps/CPS,

```
string = TRIM(ADJUSTL(field)) // '/a_vs_mag_tip .ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(REAL(MINVAL(x1)) - 0.01, REAL(MAXVAL(x 1) ) +0.01, REAL(MINVAL(x3)) - 0.01, REAL(MAXVAL(x3)) +0.01, 0, 0)
CALL pgenv(REAL(MINVAL(x1)) - 0.01, REAL(MAXVAL(x 1) ) +0.01, REAL(MINVAL(x3)) - 0.01, REAL(MAXVAL(x3)) +0.01, 0, 0)
CALL pgslw(3)
CALL pgslw(3)
CALL pgpoint (nit, REAL(x1), REAL(x3), -1)
CALL pgpoint (nit, REAL(x1), REAL(x3), -1)
CALL pgslw(1)
CALL pgslw(1)
CALL pglab('Proposed_i \d0\u_tip_magnitude', 'Proposed_value_of_a',,',')
CALL pglab('Proposed_i \d0\u_tip_magnitude', 'Proposed_value_of_a',,',')
CALL pgend
CALL pgend
!-_-_-_-_-_-_-_-_-_
!-_-_-_-_-_-_-_-_-_
string = TRIM(ADJUSTL(field)) // '/a_vs_f.ps/CPS'
string = TRIM(ADJUSTL(field)) // '/a_vs_f.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
```

CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)

```
```

CALL pgenv(0.9*REAL(MINVAL(x2)), 1.1*REAL(MAXVAL(x2)), 0.9*REAL(MINVAL(x3)), 1.1*REAL(MAXVAL(x3)), 0, 0)
CALL pgpoint (nit, REAL(x2), REAL(x3), -1)
CALL pglab('Proposed_value_of_f,, 'Proposed_value_of_a', ',')
CALL pgend
END SUBROUTINE OtherPlots
SUBROUTINE NoiseMake !Generates a polynomial of degree 7 that follows the
USE Global !functional form of the GSS background LF. The polynomial
IMPLICIT NONE !coefficients were derived in 'BackgroundPolyFit' using
!'svdfit, from Numerical Recipes
area2 = 0.d0
DO i = 1, 8 * binspm + 1
modelnoise(i,1) = 18.d0 + (i - 1.d0)/REAL(binspm)
modelnoise(i,2) = 0.d0
DO j = 1, np !Set background counts
modelnoise(i,2) = modelnoise(i,2) + ay(j) * (modelnoise(i,1) - 21.d0) ** (j - 1)
END DO
IF (modelnoise(i,2) . 1t. 0.d0) THEN !
modelnoise(i,2) = 0.d0 !Insure no negative counts
END IF
IF (i .ge. (blim-18.d0)*binspm + 1 .and. i . le. flimBins) THEN
area2 = area2 + modelnoise(i,2) !Used for normalization in 'ModelMake,
END IF
END DO
END SUBROUTINE NoiseMake
SUBROUTINE NoisePlot !Plots the unscaled form of the background LF
USE Global
IMPLICIT NONE
CALL pgbegin(0,'?', 1, 1)
CALL pgenv(18., REAL(mag_cutoff), 0., 1.1*REAL(MAXVAL(modelnoise(:,2), mask = modelnoise(:,1) . le. 23.5)), 0, 0)
CALL pgbin (nbins - INT(2.5*binspm), REAL(modelnoise (:, 1)), REAL(modelnoise (:,2)), .true .)
CALL pglab('i\d0\u', 'Counts', ',')
CALL pgend
END SUBROUTINE NoisePlot
SUBROUTINE ModelMake !Initial Model (i.e. model before convolution)
USE Global
IMPLICIT NONE
REAL*8 :: func_i
noise = modelnoise(:,2) * (f/area2) !Calculate background height
area = 0.d0
DO i = 1, nbins
model(i,1) = 18.d0 + (i 1.d0)/REAL(binspm)
IF (model(i,1) + hb .gt. mag_tip .and. model(i,1) - hb .le. mag_tip) THEN

```
```

    model(i,2)=((10.d0**(a*(model(i,1) + hb - mag_tip)))/(a*LOG(10.))) - &
                        (1.d0/(a*LOG(10.)))
    area = area + model(i,2) !Used to calculate noise in master program
    ELSE IF (model(i,1) .gt. mag_tip) THEN !Model value faintward of tip
    model(i,2) = ((10.d0**(a*(model(i,1) + hb - mag_tip)))/(a*LOG(10.))) - &
            ((10.d0**(a*(model(i,1) - hb - mag_tip)))/(a*LOG(10.)))
    IF (i .ge. (blim-18.d0)*binspm + 1 .and. i .le. flimBins) THEN
        area = area + model(i,2) !Used to calculate noise in master program
    ELSE
        cycle;
    END IF
    ELSE
    model(i,2)=0.d0 !Model value brightward of tip
    END IF
    END DO
model(:,2) = (model(:,2)/area) * (1.d0-f) !Normalize
model(:,2) = model(:,2) + noise !Add noise
END SUBROUTINE ModelMake
SUBROUTINE ModelPrint ! Prints model before convolution
USE Global
IMPLICIT NONE
CALL pgbegin(0,'?', 1,1)
CALL pgenv(REAL(mag_tip ) - 3., REAL(mag_cutoff), 0., 1.1*REAL(model(INT(5.5*binspm),2)), 0, 0)
CALL pgbin (nbins - INT(2.5*binspm), REAL(model(:,1)), REAL(model(:,2)), .true.)
CALL pglab('i\d0\u',, 'Counts',',')
CALL pgend
END SUBROUTINE ModelPrint
SUBROUTINE GaussianKernel !Generates a Gaussian kernel 'kernel' with
USE Global !HWHM (sigma) changing with magnitude in
IMPLICIT NONE !accordance with func_i. Kernel is defined from
!gx = -5*sigma to gx = +5*sigma
REAL*8 :: func_i
temp = 0.d0 ; kernel = 0.d0
gx=0.
j=0
DO WHILE (gx . le. 5.e0*func_i (t))
j=j+1
gx = 0.e0 + (j-1.e0)/binspm !Creates half of
temp(j,1)=gx !the kernel ('temp')
temp(j, 2) = exp(-((gx)**2.e0)/(2.e0*(func_i(t)**2.e0)))!
END DO
ghw = j - 1.d0 !The first non-zero bin of 'cmodel' will be the first
!non-zero bin of 'model' minus ghw
DO k = 1, j
kernel(k,:) = temp(j - (k-1),:) !Create 'kernel' by concatenating
kernel(j+k,2) = temp(k+1,2) !'temp' with a reflected version

```
```

    kernel(j+k,1) = -temp(k+1,1) !of itself
    END DO
!Note: temp (2*j,2) = 0.d0 ; temp (2*j,1) = -0.d0
kernel(:,2) = kernel(:,2)/SUM(kernel(:,2))
END SUBROUTINE GaussianKernel
SUBROUTINE GaussianKernelPrint !Prints Gaussian Kernel at given magnitude
USE Global
IMPLICIT NONE
REAL*8 :: func_i
CALL pgbegin(0,'?',1,1)
CALL pgenv(-5.5 * REAL(func_i(t)), 5.5 * REAL(func_i(t)), 0., 1.1*MAXVAL(REAL(kernel(:,2))), 0, 0)
CALL pgbin (2*ghw+1, REAL(kernel(:,1)), REAL(kernel(:,2)), .true.)
CALL pglab('Magnitude_offset', 'Strength', ',)
CALL pgend
END SUBROUTINE GaussianKernelPrint
SUBROUTINE Convolution ! Convolves initial model with a Gaussian kernel
Use Global !whose width is equal to the photometric error
IMPLICIT NONE !and hence expands with increasing magnitude
cmodel = 0.d0
DO i = 1, nbins
t = 18.d0 + (i - 1.d0)/REAL(binspm) !Convert bin number to magnitude
cmodel(i,1)=t !This then derives the current
CALL GaussianKernel !width of the Gaussian kernel
DO j = -ghw, ghw, +1
IF (i .gt. ghw . and. i .lt. nbins - ghw) THEN !Convolve
cmodel (i+j,2) = cmodel (i+j,2) + kernel(ghw+j+1,2)*model(i,2)!model with
END IF
END DO
END DO
DO i = nbins, flimBins+1, -1 !Set the faint limit
cmodel(i,2) = 0.d0 !of the final convolved
END DO !model at flim.
cmod_nbins = flimBins
!Normalize the convolved model
cmodel(:,2) = cmodel(:,2)/SUM(cmodel(:,2), mask = cmodel(:,1) .ge. blim)
!Note the above step is very important - normalization must only be over the
!range of magnitudes in the 'data' array - i.e. down to mlim -> not right the
!way to }18\mathrm{ unless mlim = 18. This was a difficult bug to find!
END SUBROUTINE Convolution

```
```

800

```
SUBROUTINE ConvolutionPrint ! Prints convolved version of model
```

SUBROUTINE ConvolutionPrint ! Prints convolved version of model
USE Global
USE Global
IMPLICIT NONE
IMPLICIT NONE
CALL pgbegin(0,'?', 1,1)
CALL pgbegin(0,'?', 1,1)
CALL pgenv(REAL(mag_tip) - 0.5, 25., 0., 1.1*MAXVAL(REAL}(\operatorname{cmodel}(:,2))), 0, 0
CALL pgenv(REAL(mag_tip) - 0.5, 25., 0., 1.1*MAXVAL(REAL}(\operatorname{cmodel}(:,2))), 0, 0
CALL pgbin (nbins, REAL(cmodel(:,1)), REAL(cmodel(:,2)), .true.)
CALL pgbin (nbins, REAL(cmodel(:,1)), REAL(cmodel(:,2)), .true.)
CALL pglab('i\d0\u', 'Relative_probability,, ',')
CALL pglab('i\d0\u', 'Relative_probability,, ',')
CALL pgend
CALL pgend
END SUBROUTINE ConvolutionPrint
END SUBROUTINE ConvolutionPrint
SUBROUTINE DataHist ! Generates finely and coarsely binned histograms and
SUBROUTINE DataHist ! Generates finely and coarsely binned histograms and
USE Global !overlays them with the best fit model determined by
USE Global !overlays them with the best fit model determined by
IMPLICIT NONE ! the MCMC
IMPLICIT NONE ! the MCMC
REAL*8 :: scaled_f_rec
REAL*8 :: scaled_f_rec
histo_fine(:,1) = model(:,1)
histo_fine(:,1) = model(:,1)
DO i = 1, INT (0.25*(nbins - 1.d0)) + 1
DO i = 1, INT (0.25*(nbins - 1.d0)) + 1
histo_coarse(i,1) = 18.d0 + (i - 1.d0)/REAL(0.25*binspm)
histo_coarse(i,1) = 18.d0 + (i - 1.d0)/REAL(0.25*binspm)
END DO
END DO
DO i = 1, ndata2 !
DO i = 1, ndata2 !
histo_fine(INT(REAL((data(i) - 18.d0)*binspm) + 1.d0),2)=\& !Generates
histo_fine(INT(REAL((data(i) - 18.d0)*binspm) + 1.d0),2)=\& !Generates
histo_fine(INT(REAL((data (i) - 18.d0)*binspm) + 1.d0),2) + 1.d0 !
histo_fine(INT(REAL((data (i) - 18.d0)*binspm) + 1.d0),2) + 1.d0 !
histo_coarse(INT(REAL((data (i) - 18.d0)*0.25*binspm) + 1.d0),2)=\& ! Histograms
histo_coarse(INT(REAL((data (i) - 18.d0)*0.25*binspm) + 1.d0),2)=\& ! Histograms
histo_coarse(INT(REAL}((\boldsymbol{data}(\textrm{i})-18.d0)*0.25*binspm) + 1.d0),2) + 1.d0 !
histo_coarse(INT(REAL}((\boldsymbol{data}(\textrm{i})-18.d0)*0.25*binspm) + 1.d0),2) + 1.d0 !
END DO
END DO
histo_coarse(INT(5.5*REAL(binspm/4.d0)) + 1,2) = \& !See paragraph
histo_coarse(INT(5.5*REAL(binspm/4.d0)) + 1,2) = \& !See paragraph
histo_coarse(INT (5.5*REAL(binspm/4.d0)) + 1,2)*2.d0 !below
histo_coarse(INT (5.5*REAL(binspm/4.d0)) + 1,2)*2.d0 !below
!For graphing purposes, the last bin of the coarse histogram is doubled since
!For graphing purposes, the last bin of the coarse histogram is doubled since
!this bin lies half outside the range of interest and so is depleted by
!this bin lies half outside the range of interest and so is depleted by
!roughly one half. This is for graphing only and has no bearing on the
!roughly one half. This is for graphing only and has no bearing on the
!determined best fit model
!determined best fit model
!| Plot Best Fit Model
!| Plot Best Fit Model
!\/ over histogram
!\/ over histogram
mag_tip = tip_rec ; f = f_rec ; a = a_rec !
mag_tip = tip_rec ; f = f_rec ; a = a_rec !
CALL ModelMake !Generate best fit function
CALL ModelMake !Generate best fit function
CALL Convolution !
CALL Convolution !
bfm = cmodel(:,2) !bfm = best fit model
bfm = cmodel(:,2) !bfm = best fit model
bfm = bfm * (SUM(histo_fine(:,2))/SUM(bfm)) !Scale bfm to match histogram
bfm = bfm * (SUM(histo_fine(:,2))/SUM(bfm)) !Scale bfm to match histogram
--_--c----_-------------Plots best fit model over fine histogram
--_--c----_-------------Plots best fit model over fine histogram
string = TRIM(ADJUSTL(field)) // '/model_fit_vs_data_fine.ps/CPS'
string = TRIM(ADJUSTL(field)) // '/model_fit_vs_data_fine.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(REAL(blim), REAL(flim), 0., 1.1*MAXVAL(real(histo_fine (:,2))), 0, 0)
CALL pgenv(REAL(blim), REAL(flim), 0., 1.1*MAXVAL(real(histo_fine (:,2))), 0, 0)
CALL pgbin (nbins, REAL(histo_fine (:, 1)), REAL(histo_fine(:,2)), .false.)
CALL pgbin (nbins, REAL(histo_fine (:, 1)), REAL(histo_fine(:,2)), .false.)
CALL pgsci(2)
CALL pgsci(2)
CALL pgslw(5)
CALL pgslw(5)
CALL pgline (nbins, REAL(histo_fine(:,1)), REAL(bfm))

```
CALL pgline (nbins, REAL(histo_fine(:,1)), REAL(bfm))
```

```
CALL pgsci (1)
CALL pgslw(1)
CALL pglab ('i\d0\u', 'Counts',',')
CALL pgend
bfm = bfm * 4.d0 !Scale bfm to match coarse histogram
!--------------------------ots best fit model over coarse histogram
string = TRIM(ADJUSTL(field)) // '/model_fit_vs_data_coarse.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(REAL(blim), REAL(flim), 0., 1.1*MAXVAL(real(histo_coarse (:,2))), 0, 0)
CALL pgbin (INT(0.25*(nbins - 1.d0)) + 1, REAL(histo_coarse(:,1)), &
    REAL(histo_coarse (:,2)), .false.)
CALL pgsci(2)
CALL pgline (nbins, REAL(histo_fine(:,1)), REAL(bfm))
CALL pgsci(1)
CALL pglab('i\d0\u', 'Counts',',')
CALL pgend
END SUBROUTINE DataHist
SUBROUTINE LogLike ! Generates the log of the likelihood fn
USE Global ! - gives the likelihood of the tip being at
IMPLICIT NONE !each magnitude given the dataset
logL= 0.d0
DO i = 1, ndata2
    prob = (data(i) - 18.d0)*binspm + 1.d0
    prob = cmodel(INT(prob),2)
    logL = logL + LOG10(prob)
END DO
END SUBROUTINE LogLike
SUBROUTINE TipAndSigma !Identifies the best parameter values and
USE Global !their associated 1 sigma errors from the
IMPLICIT NONE !respective posterior plots.
PPD_peak = 0.d0
DO i = 1, 10*(nbins - 1) +1
    IF (post_y1(i) .gt. PPD_peak) THEN
        PPD_peak = post_y1(i) !Find best fit TRGB value
        tip_rec = post_x1(i) !
    END IF
END DO
PPD_peak = 0.d0
DO i = 1, nbin
    IF (post-y2(i) .gt. PPD_peak) THEN !
        PPD_peak = post_y2(i) !Find best fit f value
        f_rec = post_x2(i) !
    END IF
END DO
```

```
922
```

PPD_peak = 0.d0

```
PPD_peak = 0.d0
DO i = 1, 2*nbins - 1
DO i = 1, 2*nbins - 1
    IF (post_y3(i) .gt. PPD_peak) THEN
    IF (post_y3(i) .gt. PPD_peak) THEN
        PPD_peak = post_y3(i) !Find best fit a value
        PPD_peak = post_y3(i) !Find best fit a value
        a_rec = post_x3(i)
        a_rec = post_x3(i)
    END IF
    END IF
END DO
END DO
tip_kpc = (100.d0**((tip_rec + 3.44d0)/10.d0))/100.d0 !Distance inferred from
tip_kpc = (100.d0**((tip_rec + 3.44d0)/10.d0))/100.d0 !Distance inferred from
                                    !tip magnitude in kpc
                                    !tip magnitude in kpc
tip_counts = 0.d0 ; mcounts = 0.d0
tip_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(post_y1, DIM = 1), 1, -1
DO i = MAXLOC(post_y1, DIM = 1), 1, -1
    mcounts = mcounts + post-y1(i)
    mcounts = mcounts + post-y1(i)
END DO
END DO
DO i = MAXLOC(post-y 1, DIM = 1), 1, -1
DO i = MAXLOC(post-y 1, DIM = 1), 1, -1
    tip_counts = tip_counts + post_yl(i) !Finds negative one sigma
    tip_counts = tip_counts + post_yl(i) !Finds negative one sigma
    IF (tip_counts .ge. 0.682*mcounts) THEN !error in magnitudes
    IF (tip_counts .ge. 0.682*mcounts) THEN !error in magnitudes
        tip_msigma = ((\boldsymbol{REAL}(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
        tip_msigma = ((\boldsymbol{REAL}(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
        tip_msigma = tip_rec - tip_msigma
        tip_msigma = tip_rec - tip_msigma
        exit
        exit
    END IF
    END IF
END DO
END DO
tip_counts = 0.d0 ; pcounts = 0.d0
tip_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(post-y1, DIM = 1), 10*(nbins - 1)+1
DO i = MAXLOC(post-y1, DIM = 1), 10*(nbins - 1)+1
    pcounts = pcounts + post_yl(i)
    pcounts = pcounts + post_yl(i)
END DO
END DO
DO i = MAXLOC(post-y 1, DIM = 1), 10*(nbins -1) +1
DO i = MAXLOC(post-y 1, DIM = 1), 10*(nbins -1) +1
    tip_counts (post-y1, DIM = 1), 10*(nbins - 1)+1,
    tip_counts (post-y1, DIM = 1), 10*(nbins - 1)+1,
            tip_psigma = ((\boldsymbol{REAL}(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
            tip_psigma = ((\boldsymbol{REAL}(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
            tip_psigma = tip_psigma - tip_rec
            tip_psigma = tip_psigma - tip_rec
            exit
            exit
    END IF
    END IF
END DO
END DO
d1 = 0 ; d2 = 0 ; d3 = 0 ; d4 = 0
d1 = 0 ; d2 = 0 ; d3 = 0 ; d4 = 0
f_counts = 0.d0 ; a_counts = 0.d0
f_counts = 0.d0 ; a_counts = 0.d0
DO i = 1, nbins
DO i = 1, nbins
    f_counts = f_counts + post_y2(i)
    f_counts = f_counts + post_y2(i)
    a_counts = a_counts + post_y3(i)
    a_counts = a_counts + post_y3(i)
    IF (f_counts .ge. 0.159*nit . and. d1 .eq. 0) THEN
    IF (f_counts .ge. 0.159*nit . and. d1 .eq. 0) THEN
            fminsig = post_x2(i)
            fminsig = post_x2(i)
            d1 = 1
            d1 = 1
    END IF
    END IF
    IF (f_counts .ge. 0.841*nit .and. d2 .eq. 0) THEN !For f and a:
    IF (f_counts .ge. 0.841*nit .and. d2 .eq. 0) THEN !For f and a:
        fplusig = post_x2(i) !Finds upper and lower
        fplusig = post_x2(i) !Finds upper and lower
            d2 = 1 !bounds for posterior
            d2 = 1 !bounds for posterior
    END IF !distribution within one
    END IF !distribution within one
    IF (a_counts .ge. 0.159*nit .and. d3 .eq. 0) THEN !sigma of maximum.
    IF (a_counts .ge. 0.159*nit .and. d3 .eq. 0) THEN !sigma of maximum.
            aminsig = post_x 3(i)
            aminsig = post_x 3(i)
            d3 = 1
            d3 = 1
    END IF
    END IF
    IF (a_counts .ge. 0.841*nit .and. d4 .eq. 0) THEN !
    IF (a_counts .ge. 0.841*nit .and. d4 .eq. 0) THEN !
        aplusig = post_x3(i)
        aplusig = post_x3(i)
        d4 = 1
        d4 = 1
    END IF
    END IF
END DO
```

END DO

```

983
984
```

f_sigma = 0.5d0*(fplusig - fminsig) !Hence calculates 1 sigma error
a_sigma = 0.5d0*(aplusig - aminsig) !for f and a
kpc_merr = tip_kpc*100.d0**(tip_msigma/10.d0) - tip_kpc !minus tip error in kpc
kpc_perr = tip_kpc*100.d0**(tip_psigma/10.d0) - tip_kpc !plus tip error in kpc
END SUBROUTINE TipAndSigma
FUNCTION func_i(m) !This function feeds the photometric error as a function
USE Global !of magnitude to the 'GaussianKernel' subroutine.
IMPLICIT NONE
REAL*8 :: func_i,m, c1, c2, c3
c1 = 0.001
c3 = log(0.24) - log(0.11)
c2 = c3*25.0 - log(0.24)
func_i = c1 + exp(c3*m - c2)
END FUNCTION
SUBROUTINE PolySelect !Used for selection of appropriate colour cut
USE Global !in colour-magnitude space
IMPLICIT NONE
integer MAXPT, ipol
integer NPT_ggr,NPT_spatial
parameter (MAXPT=100)
real*4 XCOL_ggr(MAXPT),YMAG_ggr(MAXPT)
real*4 X_spatial (MAXPT), Y_spatial (MAXPT)
logical refine_CMDsel_ggr, refine_spatialsel
!parameter (refine_CMDsel_ggr=.true.)
parameter (refine_CMDsel_ggr=.false.)
! parameter (refine_spatialsel =.true.)
parameter (refine_spatialsel=.false.)
logical in_poly
external in_poly
npt_ggr=0
if (refine_CMDsel_ggr) then
call pgsls(2)
call pgmove(0.2, 26.0)
call pgdraw (0.2,15.0)
call pgsls(1)
call pglcur(MAXPT,NPT_ggr,XCOL_ggr,YMAG_ggr)
open(2,file='ANDI.CMD', status='unknown')
write(2,*) NPT_ggr
do ipol=1,NPT_ggr
write(2,*) XCOL_ggr(ipol),YMAG_ggr(ipol)
end do
close(2)
call pgsci(1)
call pgadvance

```
```

else
open(2,file='ANDI.CMD',status='old')
read(2,*) NPT_ggr
do ipol=1,NPT_ggr
read(2,*) XCOL_ggr(ipol),YMAG_ggr(ipol)
end do
close(2)
call pgsci(2)
call pgslw(5)
call pgline(NPT_ggr,XCOL_ggr,YMAG_ggr)
call pgsci(1)
call pgslw(1)
end if

```

```

j=0
DO i = 1, ndata !Makes new
IF (in_poly(g_min_i(i),mag_i(i),NPT_ggr, XCOL_ggr, YMAG_ggr)) THEN!arrays for
IF (mag_i(i) . le. flim .AND. mag_i(i) .ge. blim) THEN !i and g-i
j = j+1 !containing
mag_i_poly(j) = mag_i(i) !only stars
g_min_i_poly(j) = g_min_i(i) !within
END IF !polygon
END IF
END DO
ndata2 = j !New number of stars in dataset after colour cut
!---------------------------------
j=0 ; k = 0
DO i = 1, bg_ndata
IF (in_poly(bg_g_min_i(i),bg_mag_i(i),NPT_ggr, XCOL_ggr,YMAG_ggr)) THEN
IF (bg_mag_i(i) .le. 24.d0) THEN
IF (bg_mag_i(i) .le. flim .AND. bg_mag_i(i) .ge. blim) THEN !arrays for
k = k+1
END IF
j = j+1
bg_mag_i_poly(j) = bg_mag_i(i)
bg_g_min_i_poly(j) = bg_g_min_i(i)
END IF
END IF
END DO
bg_ndata2 = j ; bg_ndata 3 = k !Stars in bckgrnd ; Stars in bckgrnd between blim \& flim
END SUBROUTINE PolySelect

```

```

logical function in_poly(x,y,np,xp,yp) !Used by PolySelect subroutine
implicit none
real*4 x,y
integer np
real*4 xp(np),yp(np)
real*4 tiny, xs, xe,ys,ye
parameter (tiny=1.e-5)
real*4 simag, fimag
external fimag
integer j

```
```

1105
106 simag=0.0
do j=1,np
if (j.lt.np) then
xe=xp(j+1)
xs=xp(j)
ye=yp(j+1)
ys=yp(j)
else
xe=xp(1)
xs=xp(j)
ye=yp(1)
ys=yp(j)
end if
simag=simag+ fimag (x,xs,xe,y,ys,ye)
end do
if (abs(simag).gt.tiny) then
in_poly=.true.
else
in_poly=.false.
end if
end

```

```

real*4 function fimag( }x0,xs,xe,y0,ys,ye) !Used by PolySelect subroutine
implicit none
real*4 x0, xs, xe,y0,ys,ye
real*4 top, bot
top= -(xe-x0)*(ys-y0)+(ye-y0)*(xs-x0)
bot=(xe-x0)*(xs-x0) +(ye-y0)*(ys-y0)
fimag=atan2(top,bot)
end

```

```

!---------------------------------

```

\section*{Program: MCMCTRGBTester2.f95}

Creation Date: 30 July 2010 (first version 25 Mar 2010)
Relevant Section: §2.3 of Paper I (Ch. 3)
Notes: This program was written to test the performance of the TRGB algorithm (i.e. BayesianTRGB_ANDI.f95) for different luminosity functions (LFs) that might be encountered. A model LF is created with both the tip magnitude and RGB slope constant at mag_tip \(=20.5\) and \(a=0.3\) respectively. The fraction of background stars \((f)\) in the LF is varied however as is the number of stars populating the LF (ndata). In practice, a perl script was written to run this code for all combinations of \(f\) and ndata, where \(f \in\{0.1,0.2, \ldots 0.9\}\) and ndata \(\in\{10,20,50,100,200,500,1000,2000,5000,10000,20000\}\). Many of the subroutines in this program are omitted for the sake of brevity, but their form can be ascertained from the 'BayesianTRGB_ANDI.f95' program. The subroutine that actually generates the artificial stars from the model LF is however shown - 'DataMake.'
```

MODULE Global ! Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, j, k, l, eval, idum = -9999, it, nit
INTEGER :: ndata_max, nsamples, binspm, nbins, cmod_nbins, ghw, mm, ios
PARAMETER (ndata_max = 20000000, nsamples = 100)
PARAMETER (binspm = 100)
PARAMETER (nbins = 8* binspm + 1)
PARAMETER (nit = 200000)
INTEGER :: ndata, ndata2
INTEGER :: d1, d2, d3, d4, d5, d6, field_num
INTEGER :: flimBins, blimBins
REAL*8 :: blim, flim, array(ndata_max)
REAL*8 :: c_o_f(2)
REAL*8 :: randnum1, randnum2, randnum3, randnum4, r1, r2, B1, B2, B3, B4, hb = 0.005d0
REAL*8 :: model(nbins,2), cmodel(nbins,2), magnitude(ndata_max)
REAL*8 :: histo_fine(nbins,2), histo_coarse(INT(0.25*(nbins - 1.d0)) + 1,2)
REAL*8 :: data(ndata_max), cumulative_cmodel(nbins,2), f, f_hold, bfm(nbins)
REAL*8 :: mag_tip, mag, mag_cutoff = 24.e0, a, inputs(4)
REAL*8 :: area, area2
REAL*8 :: modelnoise(nbins,2), noise(nbins) = 0.d0, pi = ACOS(-1.e0)
REAL*8 :: kernel(nbins,2) = 0.e0, scale, uplim, lowlim, gx
REAL*8 :: temp(nbins,2) = 0.e0, t
REAL*8 :: logL, prob, LikeA, LikeB
REAL*8 :: tip(nsamples), tip_ord(nsamples), maxlogL(nsamples) = -999999999999
REAL*8 :: tip_rec, tip_offset, tip_sigma, Toffset_kpc, Tsigma_kpc
REAL*8 :: f_offset, tip_kpc, kpc_err, f_sigma, a_offset, a_sigma
REAL*8 :: f_rec, a_rec, tip_counts, f_counts, a_counts
REAL*8 :: tipminsig, tiplusig, fminsig, fplusig, aminsig, aplusig
REAL*8 :: x1(nit), x2(nit), x3(nit), p(3), time(nit), r
REAL*8 :: post_y1(10*(nbins-1)+1) = 0.d0, post_x1 (10*(nbins - 1)+1), mlim
REAL*8 :: post-y2(nbins) = 0.d0, post_x2(nbins)
REAL*8 :: post_y 3 (2*nbins - 1) = 0.d0, post_x 3 (2*nbins - 1)
REAL*8 :: offset_kpc, PPD_peak
CHARACTER :: }\operatorname{argv*10, test * 40, ch 1 *9, ch 2*9, ch 3*9, ch4*9, ch5*9, string*80

```
\[
\begin{aligned}
& 79 \\
& 80
\end{aligned}
\]
```

```81
```

INTEGER :: iCCDt, clsg, clsi, ifieldt, iacc_t

```
INTEGER :: iCCDt, clsg, clsi, ifieldt, iacc_t
REAL*4 :: xgt, ygt, g, dg, im, dim, xki_t, eta_t, FeH_phot_t, diff_tip_t, E_BV_t
REAL*4 :: xgt, ygt, g, dg, im, dim, xki_t, eta_t, FeH_phot_t, diff_tip_t, E_BV_t
REAL*8 :: ra_t, de_t
REAL*8 :: ra_t, de_t
REAL*4 :: mag_g(ndata_max), mag_i(ndata_max), xki(ndata_max), eta(ndata_max)
REAL*4 :: mag_g(ndata_max), mag_i(ndata_max), xki(ndata_max), eta(ndata_max)
REAL*4 :: g_min_i(ndata_max), mag_i_poly(ndata_max), g_min_i_poly(ndata_max)
REAL*4 :: g_min_i(ndata_max), mag_i_poly(ndata_max), g_min_i_poly(ndata_max)
REAL*4 :: gmi
REAL*4 :: gmi
END MODULE Global
END MODULE Global
PROGRAM MCMCTRGBTester2 !Master program
PROGRAM MCMCTRGBTester2 !Master program
USE Global
USE Global
IMPLICIT NONE
IMPLICIT NONE
mm = IARGC()
mm = IARGC()
IF (mm==4) THEN
IF (mm==4) THEN
    CALL GETARG(1, argv )
    CALL GETARG(1, argv )
    READ (argv,*,iostat=ios) mag_tip
    READ (argv,*,iostat=ios) mag_tip
    CALL GETARG(2, argv)
    CALL GETARG(2, argv)
    READ (argv,*,iostat=ios) a
    READ (argv,*,iostat=ios) a
    CALL GETARG(3, argv)
    CALL GETARG(3, argv)
    READ (argv,*,iostat=ios) ndata !Indicates the arguments to be
    READ (argv,*,iostat=ios) ndata !Indicates the arguments to be
    CALL GETARG(4, argv) !set in the command line
    CALL GETARG(4, argv) !set in the command line
    READ (argv,*, iostat=ios) f
    READ (argv,*, iostat=ios) f
ELSE
ELSE
    WRITE(*,*) "You}\lrcorner\mathrm{ mustьenter }\lrcorner4\_\mathrm{ arguments :"
    WRITE(*,*) "You}\lrcorner\mathrm{ mustьenter }\lrcorner4\_\mathrm{ arguments :"
    stop ;
    stop ;
END IF
END IF
WRITE (*,*) "-"
WRITE (*,*) "-"
WRITE (*,*) "Model_mag_tip/slope/#_sources/background_height_=", mag_tip, a, ndata, f
WRITE (*,*) "Model_mag_tip/slope/#_sources/background_height_=", mag_tip, a, ndata, f
WRITE (ch1,*) mag_tip !
WRITE (ch1,*) mag_tip !
WRITE (ch2,*) a !
WRITE (ch2,*) a !
WRITE (ch3,*) ndata !
WRITE (ch3,*) ndata !
IF (f .eq. 0.d0) THEN !Generate test identifying character string
IF (f .eq. 0.d0) THEN !Generate test identifying character string
    WRITE (ch4,*) , 0, !to become file name using mag_tip, ndata and f
    WRITE (ch4,*) , 0, !to become file name using mag_tip, ndata and f
ELSE !e.g. 'MCMC_Test/ T_20.5-0.3-1000-0.2'
ELSE !e.g. 'MCMC_Test/ T_20.5-0.3-1000-0.2'
    WRITE (ch4,*) f !
    WRITE (ch4,*) f !
END IF
END IF
inputs(1) = mag_tip
inputs(1) = mag_tip
inputs(2) = a
inputs(2) = a
inputs(3) = ndata
inputs(3) = ndata
inputs(4) = f
inputs(4) = f
ndata2 = 0
ndata2 = 0
WRITE (test,*) 'MCMC_Test/T_' // TRIM(ADJUSTL(ch1)) &
WRITE (test,*) 'MCMC_Test/T_' // TRIM(ADJUSTL(ch1)) &
                    // '_, // TRIM(ADJUSTL(ch2)) &
                    // '_, // TRIM(ADJUSTL(ch2)) &
                    // '_, // TRIM(ADJUSTL(ch3)) &
                    // '_, // TRIM(ADJUSTL(ch3)) &
                        // ,-, // TRIM(ADJUSTL(ch4))
                        // ,-, // TRIM(ADJUSTL(ch4))
CALL random_seed !
CALL random_seed !
flimBins = INT(\boldsymbol{REAL}((23.5d0 - 18.d0)) * binspm) + 1
```

flimBins = INT(\boldsymbol{REAL}((23.5d0 - 18.d0)) * binspm) + 1

```
```

blim = 18.0d0 ; flim = 23.5d0 ; mlim = blim
CALL NoiseMake !Generates the convoluted model for the inputted
CALL ModelMake !tip magnitude and then uses to generate sets
CALL Convolution !of data points in 'DataMake' subroutine
blim = 20.0d0 ; flim = 21.0d0 ; mlim = blim
flimBins = INT(\boldsymbol{REAL}((flim - 18.d0)) * binspm) + 1
blimBins = INT(REAL((blim - 18.d0)) * binspm) + 1
CALL DataMake
DO i = 1, ndata
IF (data(i) .ge. blim .AND. data(i) . le. flim) THEN
ndata2 = ndata2 + 1
data(ndata2) = data(i)
END IF
END DO
WRITE (*,*) "Number_of_stars_in_fitted_range:", ndata2
CALL NoiseMake !
CALL MCMC !CALL
CALL TipAndSigma !MCMC
CALL PosteriorPlot !SUBROUTINES
CALL OtherPlots !
CALL DataHist !

```

```

WRITE (*, '(3F11.3)') tip_rec, tip_sigma, tip_offset !
WRITE (*, '(3a11)') "___f:______-","___sigma:", "____ooffset:" ! Write results
WRITE (*, '(3F11.3)') f_rec, f_sigma, f_offset !to file

```

```

WRITE (*, '(3F11.3)') a_rec, a_sigma, a_offset !
WRITE (*,*) "Distance }=>=, REAL(tip_kpc), "kpc\lrcornerp/m", REAL(kpc_err), "kpc"
WRITE (*,*) "Distance_Offset_=", REAL(offset_kpc), "kpc"
END PROGRAM MCMCTRGBTester2
SUBROUTINE DataMake !Generates data points from the convolved model
USE Global
IMPLICIT NONE
real*8 :: ran1
cumulative_cmodel(:,1) = cmodel (:, 1)
cumulative_cmodel(1,2)= cmodel(1,2) !Effective
DO i = 2, cmod_nbins !integral of
cumulative_cmodel(i,2) = cumulative_cmodel(i-1,2) + cmodel(i, 2)!convolved
END DO
!model
DO i = 1, ndata
CALL random_number(randnum4) !
randnum4 = cumulative_cmodel(blimBins,2) + \& !
randnum4 * (cumulative_cmodel(flimBins,2) - cumulative_cmodel(blimBins,2))
DO j = flimBins, blimBins, -1
IF (randnum4 .le. cumulative_cmodel(j,2)) THEN !Generates 'ndata'
IF (randnum4 .gt. cumulative_cmodel(j - 1,2)) THEN !datapoints from

```

158
159
160
161
162
163
163
164
165
166
167
168
data \((\mathrm{i})=\) cumulative_cmodel \((\mathrm{j}, 1)\)

\section*{exit;}

END IF
END IF
END DO
END DO

END SUBROUTINE DataMake


\section*{Program: BayesianTRGBTestPlotterMCMC.f95}

Creation Date: 29 April 2010
Relevant Section: Figs. 10 \& 11 of Paper I (Ch. 3) and Figs. \(4 \& 5\) of Paper II (Ch. 4)
Notes: This program is used to plot the results returned by 'MCMCTRGBTester2.f95,' namely the one sigma uncertainties and the offset of the recovered tip magnitude from mag_tip \(=20.5\) for each combination of \(f\) and ndata. Pixels are created with bounds \(X 1, X 2, Y 1, Y 2\) with the added complication of a log scale for the x -axis. These pixels are then assigned a shade of grey based on the magnitude of the quantity they represent.
```

PROGRAM BayesianTRGBTestPlotterMCMC !Plots results of tests for different combinations of
IMPLICIT NONE !f vs. ndata (generates two plots: tip offset for each
INTEGER :: i, ios, j, k, x (11,9) = 3
REAL :: A1 (11,9), A2(11,9)
REAL :: ndata(1000), ndata_actual(1000), f(1000), offmag(1000), v
REAL :: offkpc(1000), sigmag(1000), sigkpc(1000), temp, X1, X2, Y1, Y2
REAL :: ALEV(100), TR(6), stars(11,9), noise(11,9)
REAL :: grey, xmin, xmax
CHARACTER(LEN=15) :: C1(11,9), C2(11,9)
OPEN (unit = 1, file = './ summary.dat', status = 'old')
i = 0 ; ios = 0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
i=i+1
READ (1, *, IOSTAT = ios) ndata(i), f(i), sigkpc(i), offkpc(i), temp, temp, temp
if (ios == 0) then ;
else if (ios == - 1) then ;
i=i-1
exit
else if (ios > 0) then ;
i=i-1
cycle
end if
END DO
DO j = 1, 11
DO k = 1, 9
Al(j,k) = sigkpc(((j-1)*9)+k)
IF (A1(j,k) .eq. 0.d0) THEN
A1(j,k) = 0.001d0
END IF !Puts results
A2(j,k) = offkpc(((j - 1)*9)+k)
IF (A2(j,k) .eq. 0.d0) THEN !into ndata x f
A2(j,k) = 0.001 d0
END IF !arrays for
stars(j,k) = ndata (((j - 1)*9)+k) !
noise(j,k) = f(((j - 1)*9)+k) !easy plotting
WRITE (C1(j , k),*) NINT(A1(j,k)) !
WRITE (C2(j , k),*) NINT(A2(j , k)) !
END DO
END DO

```
```

CALL pgbegin(0,'MF_TRGBSigmaMCMC.ps/CPS', 1,1) !Generates sigma plot
CALL pgenv (0.8, 4.5, 0., 1., 0, 10)
xmin = MINVAL(A1)
xmax = MAXVAL(A1)
DO i = 1, 11
DO j = 1, 9
IF (MOD(i,3) .eq. 1) THEN
X1 = LOG10(10.**(((i-1)/3)+1)) - 0.13
X2 = LOG10(10.**(((i-1)/3)+1))+0.13
END IF
IF (MOD( i , 3) .eq. 2) THEN !x boundaries for
X1 = LOG10(2.* 10.**(((i-1)/3)+1)) - 0.13 !
X2 = LOG10(2.* 10.**(((i-1)/3)+1)) + 0.13 ! log x axis
END IF
!(3 different width
IF (MOD(i,3) .eq. 0) THEN !calculations required)
X1 = LOG10(5.* 10.**(((i-1)/3)+1)) - 0.13
X2 = LOG10(5.* 10.**(((i-1)/3)+1)) + 0.13
END IF
Y1 = j * 0.1e0 - 0.04 !Generate
Y2 = j * 0.1e0 + 0.04
grey = (xmax - Al(i,j)) / (xmax - xmin) ! Determine shade of grey
CALL pgscr(3, grey, grey, grey)
CALL pgpixl(x, 11, 9, i, i, j, j, X1, X2, Y1, Y2) !make pixels
CALL pgsci(2)
CALL pgptxt(LOG10(stars(i,j)), noise(i,j), 0., 0.5, C1(i,j)) ! put value in pixels
END DO
END DO
CALL pgsci(1)
CALL pglab('number_of\&stars', 'proportion_of^background\iotastars', \&
'Sigma_(kpc)_-_One_Sigma_Error')
CALL pgend
CALL pgbegin(0,'MF_TRGBOffsetMCMC.ps/CPS',1,1) !Generates offset plot
CALL pgenv(0.8, 4.5, 0., 1., 0, 10)
A2 = ABS(A2)
xmin = MINVAL(A2)
xmax = MAXVAL(A2)
DO i = 1, 11
DO j = 1, 9
IF (MOD(i , 3) .eq. 1) THEN
X1 = LOG10(10.**(((i-1)/3)+1)) - 0.13

```
```

    X2 = LOG10(10.**(((i-1)/3)+1))+0.13
    END IF
    IF (MOD(i,3) .eq. 2) THEN
        X1 = LOG10(2.* 10.**(((i-1)/3)+1)) - 0.13
        X2 = LOG10(2.* 10.**(((i-1)/3)+1)) + 0.13
    END IF
    IF (MOD(i,3) .eq. 0) THEN
        X1 = LOG10(5.* 10.**(((i-1)/3)+1)) - 0.13
        X2 = LOG10(5.* 10.**(((i-1)/3)+1)) + 0.13
    END IF
    Y1 = j * 0.1e0 - 0.04 !Generate
    Y2 = j * 0.1e0 + 0.04 !y boundaries
    grey = (xmax - A2(i,j)) / (xmax - xmin) !Determine shade of grey
    CALL pgscr(3, grey, grey, grey)
    CALL pgpixl(x, 11, 9, i, i, j, j, X1, X2, Y1, Y2) !make pixels
    CALL pgsci(2)
    CALL pgptxt(LOG10(stars(i,j)), noise(i,j), 0., 0.5, C2(i,j)) ! put value in pixels
    END DO
    END DO
CALL pgsci(1)

```

```

    'Offset_(kpc)_-_Radial_Distance_Offset')
    CALL pgend
END PROGRAM BayesianTRGBTestPlotterMCMC

```

\section*{Chapter Four Programs}
MF_TRGB.f95 ..... 190
MF_TRGB_Feed.pl ..... 222
MF_TRGB_Tester.f95 ..... 224
Multi_MCMC_Result_Plotter.f95 ..... 229
SatPlot.f95 ..... 245
SatDensity_SampCont.f95 ..... 256

Program: MF_TRGB.f95
Creation Date: 31 August 2011 (first version 8 Dec 2010) Many modifications.
Relevant Section: Ch. 4
Notes: This program is the successor of 'BayesianTRGB_ANDI.f95' in Appendix C and similarly lies at the heart of the material presented in Paper II (Ch. 4). The principal difference between the two is that this new version incorporates a density matched filter weighting scheme, where by stars are given a weight based on their position with in the object's density profile. In this way, stars that are more likely to be true object member's are given greater consideration during the luminosity function fitting. The actual weighting itself is taken care of by the 'Weighter' subroutine, but other subroutines have been modified significantly to handle it. The background component of the LF (built in 'NoiseMake') for instance is no longer added to the model LF in the 'ModelMake' subroutine. This is because with the weighting switched on, each star effectively has its own model LF with the ratio of background to RGB component tailored to suit the star's probability of being a true object member. Hence, these ratios are now taken into account in the 'LogLike' subroutine on a star-by-star basis. There are many other additions. A new LF plotting subroutine 'w_DataHist' plots the weighted LF and a plot of the object density profile is created in the 'Weighter' subroutine. Parallel tempering has been added to the 'MCMC' subroutine and the run-speed of the whole algorithm has be greatly improved by fixing up a design flaw in the way the convolution step was being done (The 'GaussianKernel' subroutine is now called just once and the values are saved). The program also now takes command line input so that the one set of code can be used for all objects. Due to the large number of changes made to most of the subroutines originally written for 'BayesianTRGB_ANDI.f95,' I have reproduced the whole program here rather than omitting the duplicate subroutines. Note that the command line inputs for each satellite are provided as the next item in this appendix for completeness. For a more in depth description of the workings of the program in general, see Paper II - particularly §2 and §3.1.
```

MODULE Global ! Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, j, k, l, eval, idum = -9999, it, nit
INTEGER :: ndata_max, nsamples, binspm, nbins, cmod_nbins, mm, ios

```
```

PARAMETER (ndata_max = 10000000, nsamples = 100)
PARAMETER (binspm = 100)
PARAMETER (nbins = 8*binspm + 1)
PARAMETER (nit = 500000)
INTEGER :: ndata, ndata2
INTEGER :: d1, d2, d3, d4
INTEGER :: ghw(nbins)
REAL*8 :: blim, flim, pi
PARAMETER (blim = 19.5d0)
PARAMETER ( flim = 23.5d0)
PARAMETER (pi = ACOS(-1.e0))
INTEGER :: blimBins = INT(REAL((blim - 18.d0) * binspm)) + 1
INTEGER :: flimBins = INT(REAL((flim - 18.d0) * binspm)) + 1
REAL*8 :: randnum1, randnum2, randnum3, randnum4, randnum5
INTEGER :: randint
REAL*8 :: r1, r2, spotR, hb = 0.005d0
REAL*8 :: model(nbins ,2), cmodel(nbins,2), magnitude(ndata_max)
REAL*8 :: histo_fine(nbins,2), histo_coarse(INT(0.25*(nbins-1.d0)) + 1,2)
REAL*8 :: w_histo_fine(nbins,2), w_histo_coarse(INT(0.25*(nbins-1.d0)) + 1,2)
REAL*8 :: data(ndata_max), cumulative_cmodel(nbins,2), f, f_hold, bfm(nbins)
REAL*8 :: mag_tip, mag, mag_cutoff = 24.e0, a
REAL*8 :: area, area2
REAL*8 :: modelnoise(nbins,2), noise(nbins) = 0.d0 , bg(nbins) = 0.d0
REAL*8 :: kernel(nbins,2,nbins) = 0.e0, scale, uplim, lowlim, gx
REAL*8 :: temp(nbins,2) = 0.e0, t
INTEGER :: starbin
REAL*8 :: tip(nsamples), tip_ord(nsamples), maxlogL(nsamples) = -999999999999.
REAL*8 :: tip_rec, tip_offset, tip_psigma, tip_msigma, Toffset_kpc, Tsigma_kpc
REAL*8 :: f_offset, tip_kpc, kpc_perr, kpc_merr, f_sigma, a_offset, a_sigma
REAL*8 :: f_rec, a_rec, tip_counts, f_counts, a_counts
REAL*8 :: tipminsig, tiplusig, fminsig, fplusig, aminsig, aplusig
REAL*8 :: mcounts, pcounts
INTEGER :: num_chains, cn, chain_compare, swap_count
PARAMETER (num_chains = 4)
REAL*8 :: swaprate = 1.d0/ 30.d0, logL(num_chains), LikeA(num_chains), LikeB(num_chains)
REAL*8 :: prob, sig_prob, bg_prob
REAL*8 :: beta, betaholder(num_chains) = (/ 1.d0, 0.25d0, 0.111d0, 0.001d0 /)
REAL*8 :: m_step(num_chains) = (/ 0.03d0, 0.06d0, 0.12d0, 0.3d0 /)
REAL*8 :: f_step(num_chains) = (/ 0.02d0, 0.04d0, 0.08d0, 0.2d0 /)
REAL*8 :: a_step(num_chains) = (/ 0.02d0, 0.04d0, 0.08d0, 0.2d0 /)
REAL*8 :: PTAR, par_hold(4)
REAL*8 :: x1(nit,num_chains), x2(nit,num_chains), x3(nit,num_chains), p(3), time(nit), r
REAL*8 :: post_y1(10*(nbins-1)+1) = 0.d0, post_x 1 (10*(nbins -1)+1), mlim
REAL*8 :: d_blim, bg_blim, d_flim, bg_flim
REAL*8 :: post_y2(nbins) = 0.d0, post_x2(nbins)
REAL*8 :: post-y3(2*nbins - 1) = 0.d0, post_x3(2*nbins - 1)
REAL*8 :: PPD_peak, Best_Combo(6)
CHARACTER :: argv*30, field*30, colcut*30, ch1*9, ch2*9, ch 3*9, ch4*9, ch5*9, string*60, command*200
INTEGER :: scout_counts
LOGICAL :: not_scout
INTEGER :: iCCDt, clsg, clsi, ifieldt, iacc_t
REAL*4 :: xgt, ygt, g, dg, im, dim, xki_t, eta_t, FeH_phot_alan_t, FeH_phot_t, diff_tip_t, E_BV_t
REAL*8 :: ra_t, de_t
REAL*4 :: mag_g(ndata_max), mag_i(ndata_max), xki(ndata_max), eta(ndata_max)
REAL*4 :: g_min_i(ndata_max), mag_i_poly(ndata_max), g_min_i_poly(ndata_max)
REAL*4 :: mag_g_poly(ndata_max), xi_poly(ndata_max), eta_poly(ndata_max)
REAL*4 :: gmi, xi_all(ndata_max), eta_all(ndata_max)

```
```

LOGICAL :: truestar(ndata_max), truestar_poly(ndata_max)
!-----Additional parameters for calculating background stats -------
INTEGER :: bg_ndata, bg_ndata2, bg_ndata3
REAL*4 :: bg_mag_g(ndata_max), bg_mag_i(ndata_max), bg_xki(ndata_max), bg_eta(ndata_max)
REAL*4 :: bg_g_min_i(ndata_max), bg_mag_i_poly(ndata_max), bg_g_min_i_poly(ndata_max)
REAL*4 :: bg_mag_g_poly(ndata_max), bg_gmi
REAL*8 :: bg_data(ndata_max)
!--SVD fitting of background--
INTEGER ma, mp, np, ndat
PARAMETER (ndat = INT(0.25*(nbins - 1.d0)) + 1)
PARAMEIER ( }\textrm{np}=8\mathrm{ )
PARAMETER (mp = ndat)
PARAMETER (ma = np)
REAL :: chisq, ay(ma), sig(ndat), u(mp,np), v(np,np), w(np), xa(ndat), ya(ndat)
REAL :: xt(ndat), yt(ndat)
REAL*8 :: bg_histo_coarse(ndat,2)
EXTERNAL :: funcs
!-_-_-_Additional parameters for specifying object coordinates _-_-_
INTEGER :: Jop
REAL*8 :: XIop, ETAop
REAL*8 :: RAh, RAm, RAs, DecD, DecM, DecS, RA_rad, Dec_rad
REAL*8 :: tpRAh, tpRAm, tpRAs, tpDecD, tpDecM, tpDecS, tpRA_rad, tpDec_rad
!--Additional parameters for Matched Filters Subroutine 'Weighter'--
INTEGER :: rhobins, rhobins2
PARAMETER (rhobins = 40)
REAL*4 :: C_O_F_dist(ndata_max), Density(rhobins,2), Den_sig(rhobins), rhofit(rhobins,2), weightplot(500,2)
REAL*8 :: weight(ndata_max), scaled_a(ndata_max), scaled_a_all(ndata_max), crowded_rad, ellipse_stars, ellipse_area
REAL*8 :: ellip, HLR, PA, xdash, ydash, maxweight, maxa, SR, den_prof_scale, outer_rad
REAL*8 :: weightsum, Densitysum

```

```

INTEGER :: bg_stars, sig_stars
REAL*8 :: bg_area, sig_area
REAL*8 :: known_f, bg_stars_in_sig_field
REAL*8 :: sig_field_radius, bg_low_xi, bg_up_xi
END MODULE Global
PROGRAM BayesianTRGBsatellite !Master program
USE Global
IMPLICIT NONE
mm}=\operatorname{IARGC()
IF (mm==16) THEN
CALL GETARG(1, argv )
READ (argv,*,iostat=ios) field
CALL GETARG(2, argv )
READ (argv,*,iostat=ios) RAh
CALL GETARG(3, argv)
READ (argv,*,iostat=ios) RAm
CALL GETARG(4, argv)
READ (argv,*,iostat=ios) RAs
CALL GETARG(5, argv)
READ (argv,*,iostat=ios) DecD

```
```

    CALL GETARG(6, argv )
    READ (argv,*,iostat=ios) DecM
    CALL GETARG(7, argv)
    READ (argv,*,iostat=ios) DecS
    CALL GETARG(8, argv)
    READ (argv,*,iostat=ios) ellip
    CALL GETARG(9, argv)
    READ (argv,*, iostat=ios) HLR
    CALL GETARG(10, argv)
    READ (argv,*,iostat=ios) PA
    CALL GETARG(11, argv)
    READ (argv,*,iostat=ios) crowded_rad
    CALL GETARG(12, argv)
    READ (argv,*,iostat=ios) outer_rad
    CALL GETARG(13, argv)
    READ (argv,*,iostat=ios) sig_field_radius
    CALL GETARG(14, argv)
    READ (argv,*, iostat=ios) bg_low_xi
    CALL GETARG(15, argv)
    READ (argv,*,iostat=ios) bg_up_xi
    CALL GETARG(16, argv)
    READ (argv,*, iostat=ios) colcut
    ELSE
WRITE(*,*) "You_must_enter_16_arguments:" !
stop ;
END IF
string = TRIM(ADJUSTL(field)) // ,/results.dat,
OPEN(3, file=TRIM(ADJUSTL(string)), status = 'unknown')
WRITE (3,*) "Field_Name:", field
CALL positionFinder !
CALL random_seed !
CALL M31DataReader !
CALL Weighter
CALL SVDFitter
CALL GaussianKernel !
CALL NoiseMake !CALL
CALL MCMC !
!CALL PosteriorPlot !SUBROUTINES
CALL TipAndSigma
CALL PosteriorPlot
CALL OtherPlots
CALL DataHist
CALL w_DataHist
IF (num_chains .ne. 1) THEN
WRITE (3,*) "Proposed_Swaps_with_Cold_Sampler_Chain:", chain_compare
WRITE (3,*) "Accepted_Swaps\_with乞Cold_Sampler_Chain:", swap_count
WRITE (3,*) "Parallel\_Tempering„Acceptance\lrcornerRate:", REAL(swap_count)/ REAL(chain_compare)
END IF

```

```

WRITE (3, '(3F10.3)') tip_rec, tip_psigma, tip_msigma!
WRITE (3, '(2a11)') "-_-f:_-_-_--",">-_sigma:__"" !
WRITE (3, '(2F10.3)') f_rec, f_sigma !Write results

```

```

WRITE (3, '(2F10.3)') a_rec, a_sigma !
WRITE (3,*) "Distance_=", REAL(tip_kpc), "kpc"

```
```

WRITE (3,*) "Error_=_+", kpc_perr, "kpc_-", kpc_merr, "kpc"
END PROGRAM BayesianTRGB satellite
SUBROUTINE PositionFinder !Converts object RA and Dec into M31 tangent plane coordinates xi and
USE Global !eta. These are used in the next subroutine for reading in all stars
IMPLICIT NONE !from the PAndAS survey with in some radius of the object center
tpRAh = 0.d0 !
tpRAm = 42.d0 !
tpRAs = 44.33d0 !RA and Dec of tangent point
tpDecD = 41.d0 !(i.e. M31)
tpDecM = 16.d0 !
tpDecS = 7.5d0 !
!|Perform
!\/Conversion
RA_rad = (pi/180.d0) * (RAh * 15.d0 + RAm * (15.d0/60.d0) + RAs * (15.d0/3600.d0))
Dec_rad = (pi/180.d0)*(DecD + DecM/60.d0 + DecS/3600.d0)
tpRA_rad = (pi/180.d0) * (tpRAh * 15.d0 + tpRAm * (15.d0/60.d0) + tpRAs * (15.d0/3600.d0))
tpDec_rad =(pi/180.d0) *(tpDecD + tpDecM/60.d0 + tpDecS/3600.d0)
CALL sla_DS2TP (RA_rad, Dec_rad, tpRA_rad, tpDec_rad, XIop, ETAop, Jop)
!/\
! |
XIop = XIop * (180.d0/pi) !tangent plane coordinates
ETAop = ETAop * (180.d0/pi) !(i.e. PAndAS xi and eta)
WRITE (3,*) "C.O.F.\_Xi_=", XIop, "C.O.F._Eta_=", ETAop
END SUBROUTINE PositionFinder
SUBROUTINE M31DataReader !The field to be analysed is specified here
USE Global
IMPLICIT NONE
OPEN(1,file='../../../PANDAS/M31_unique_con.dat',form='unformatted,},\mathrm{ status='old')
i = 0 ; j = 0
DO WHILE (.true.)
READ(1,IOSTAT=ios) ra_t, de_t,iCCDt,xgt,ygt, \& !Read in data
g,dg,clsg,im,dim,clsi,ifieldt,xki_t,eta_t, \& !from binary
FeH_phot_alan_t, FeH_phot_t, diff_tip_t,E_BV_t,iacc_t !format data file
IF (ios.ne.0) exit
g=g-3.793*E_BV_t !Extinction
im=im-2.086*E_BV_t !Corrections
gmi = g - im
if (clsi.ne.-1 .and. clsi.ne.-2) cycle !Regects
if (clsg.ne.-1 .and. clsg.ne.-2) cycle !non stars
if (iacc_t .ne. 1) cycle

```
```

    if (18.0.1e.im.and.im.le.24.0) then !Specifies
    else !magnitude
        cycle !range to
    end if !include
    if (-2.5.le.FeH_phot_alan_t.and.FeH_phot_alan_t.le.-1.5) then !
    else !Specifies metallicity
    ! cycle !range to include
    end if
    spotR = SQRT((ABS(eta-t - (ETAop))) **2 + (ABS(xki_t - (XIop))) **2)
    IF (spotR .lt. sig_field_radius) THEN
        i = i + 1
        IF (i .gt. ndata_max) exit
        mag_g(i)=g
        mag_i (i)=im !
        g_min_i(i)=gmi !If all conditions are met, add star data to signal arrays
        xki(i)=xki_t
        eta(i)=eta_t !
        IF (ifieldt .ge. 0) THEN !
            truestar(i) = true. ! Distinguish between
        ELSE !real data and artificial
            truestar(i) = .false. !background
    END IF
    ELSE IF (xki_t .ge. bg_low_xi .and. xki_t .le. bg_up_xi) THEN
    IF (eta_t .ge. ETAop - 0.5d0 .and. eta_t .le. ETAop + 0.5d0) THEN
        j = j + 1
        IF (j .gt. ndata_max) exit
            bg_mag_g(j)=g
            bg_mag_i(j)=im
            bg_g_min_i(j)=gmi !If all conditions are met, add star data to bckgrnd arrays
            bg_xki(j)=xki_t !
            bg_eta(j)=eta_t !
        END IF
    END IF
    END DO
ndata = i ; bg_ndata = j
sig_area = pi * (sig_field_radius ** 2.d0) !Calculate area of signal field
bg_area = 1.d0* (bg_up_xi - bg_low_xi) - (pi * (sig_field_radius ** 2.d0)) !Calculate area of BG field
DO i = 1, ndata
data(i) = mag_i(i) !
xi_all(i) = xki(i) !Object stars before applying colour cut
eta_all(i) = eta(i)
END DO
DO j = 1, bg_ndata
bg_data(j) = bg_mag_i(j) !BG stars before applying colour cut
END DO
CALL M31DataPlotter
END SUBROUTINE M31DataReader

```
```

312
313
CALL pgslw(1)
CALL pglab('\(0640)_(degrees)', '\(0633)_(degrees)',,',)
CALL pgend
WRITE (command,*) 'convert_-rotate _90_./' // TRIM(ADJUSTL(field)) // \&
'/sig_field.ps_./, // TRIM(ADJUSTL(field)) // \&
'/ sig_field.jpg'
call system(command)

```

```

string = TRIM(ADJUSTL(field)) // '/sig_cmd.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MINVAL(g_min_i, mask = g_min_i . ne. 0.), MAXVAL(g_min_i), \&
MAXVAL(mag_i), MINVAL(mag_i, mask = mag_i .ne. 0.), 0, 0)
CALL pgslw(3)
DO i = 1, ndata
IF (truestar(i)) THEN
CALL pgpt (1, g_min_i(i), mag_i(i), -1)
ELSE
CALL pgpt (1, g_min_i(i), mag_i(i), 225)
END IF
END DO
CALL pgslw(1)
CALL pglab('(g_-_i)\d0\u',,'i\d0\u', ,')
CALL PolySelect ! For CMD colour-cut
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // \&
'/sig_cmd.ps_./' // TRIM(ADJUSTL(field)) // \&
,/sig_cmd.jpg,
call system(command)
!------------------------Bckgrnd-Field
string = TRIM(ADJUSTL(field)) // '/bg_field.ps/CPS'

```
```

CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MAXVAL(bg_xki, mask = bg_xki .ne. 0.), \&
MINVAL(bg_xki, mask = bg_xki .ne. 0.), \&
MINVAL(bg_eta, mask = bg_eta .ne. 0.), \&
MAXVAL(bg_eta, mask = bg_eta .ne. 0.), 1, 0)
CALL pgslw (2)
CALL pgpt (bg_ndata, bg_xki, bg_eta, -1)
CALL pgslw(1)
CALL pglab('\(0640)_(degrees)', '\(0633)_(degrees)', ,')
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // \&
,/bg_field.ps_./' // TRIM(ADJUSTL(field)) // \&
,/bg_field.jpg,
call system(command)
!-----------------------------------------------------
string = TRIM(ADJUSTL(field)) // ,/bg_cmd.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MINVAL(bg_g_min_i, mask = bg_g_min_i .ne. 0.), \&
MAXVAL(bg_g_min_i), MAXVAL(bg_mag_i), \&
MINVAL(bg_mag_i, mask = bg_mag_i .ne. 0.), 0, 0)
CALL pgslw (3)
CALL pgpt (bg_ndata, bg_g_min_i, bg_mag_i, -1)
CALL pgslw(1)
CALL pglab ('(g_-_i)\d0\u,, ' i \ d0 \u', ,',)
CALL PolySelect ! For CMD colour-cut
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // \&
,/bg_cmd.ps_./' // TRIM(ADJUSTL(field)) // \&
'/bg_cmd.jpg'
call system(command)

```

```

string = TRIM(ADJUSTL(field)) // '/i_and_g_in_cut.dat,
OPEN(7, file=TRIM(ADJUSTL(string)), status = 'unknown')

```

```

data = 0.d0 ; xki = 0.d0 ; eta = 0.d0 ; d_blim = 100.d0 ; d_flim = 0.d0
DO i = 1, ndata2
data(i) = mag_i_poly(i)
xki(i) = xi_poly(i)
eta(i) = eta_poly(i)
IF (data(i) . lt. d_blim) THEN
d_blim = data(i)
END IF
IF (data(i) .gt. d_flim) THEN
d_flim = data(i)
END IF
WRITE (7, '(2F16.5)') mag_i_poly(i), mag_g_poly(i)
END DO
!-----------------------------
string = TRIM(ADJUSTL(field)) // '/sig_field_cc.ps/CPS,

```
```

CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MAXVAL(xki, mask = xki .ne. 0.), MINVAL(xki, mask = xki . ne. 0.), \&
MINVAL(eta, mask = eta .ne. 0.), MAXVAL(eta, mask = eta .ne. 0.), 1, 0)
CALL pgslw(3)
DO i = 1, ndata2
IF (truestar_poly(i)) THEN
CALL pgpt (1, xki(i), eta(i), -1)
ELSE
CALL pgpt (1, xki(i), eta(i), 225)
END IF
END DO
CALL pgslw(1)
CALL pglab('\(0640)_(degrees)', '\(0633)_(degrees)', ',')
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // \&
,/sig_field_cc.ps_./' // TRIM(ADJUSTL(field)) // \&
\prime/}\mathrm{ sig_field_cc.jpg'
call system(command)
!-------Input selected background data into 'bg_data'-------
bg_data = 0.d0 ; bg_blim = 100.d0 ; bg_flim = 0.d0
DO i = 1, bg_ndata2
bg_data(i) = bg_mag_i_poly(i)
IF (bg_data(i) .lt. bg_blim) THEN
bg_blim = bg_data(i)
END IF
IF (bg_data(i) .gt. bg_flim) THEN
bg_flim = bg_data(i)
END IF
END DO
!---Set parameters for calculation of background height--
sig_stars = ndata2 !Total number of stars in signal field
bg_stars = bg_ndata3 !Number of stars in background field
bg_stars_in_sig_field= REAL(bg_stars) * (sig_area/bg_area)! ; bg_stars = 0.d0
!Number of Background stars in signal field
WRITE (3,*) "Number_of_data\iotapoints:", sig_stars
WRITE (3,*) "Average_Field_SNR:", (REAL(sig_stars) - bg_stars_in_sig_field) / bg_stars_in_sig_field
!-------Make coarse data histogram for bckgrnd---------
DO i = 1, INT(0.25*(nbins - 1.d0)) + 1
bg_histo_coarse(i, 1) = 18.d0 + (i - 1.d0)/REAL}(0.25* binspm)
END DO
DO i = 1, bg_ndata2
bg_histo_coarse(INT((bg_data(i) - 18.d0) *0.25*binspm) + 1, 2) = \&
bg_histo_coarse(INT((bg_data(i) - 18.d0) *0.25*binspm) + 1, 2) + 1.d0
END DO
!|| Fill empty bright edge of array with
!\/ artificial data for improved fitting
DO i = 1, INT((bg_blim - 18.d0) * REAL(binspm/4.d0)) + 4
bg_histo_coarse(i,2) = bg_histo_coarse(INT((bg_blim - 18.d0) * REAL(binspm/4.d0)) + 4, 2)
END DO

```

\section*{END SUBROUTINE M31DataPlotter}
```

SUBROUTINE Weighter !Implements an elliptical weighting scheme using the
USE Global !ellipticity, half-light radius and position angle
IMPLICIT NONE !of the satellite
HLR = HLR * (1.d0/ 60.d0) !Convert half-light radius from minutes to degrees
SR = HLR / 1.678 ! Convert from half light radius to exponential scale radius
PA = PA * (pi/180.d0) !Convert position angle from degrees to radians
ellipse_area = pi * outer_rad * outer_rad * (1.d0 - ellip) ! Area of outer ellipse
ellipse_stars = 0.d0 ; maxweight = 0.d0 ; maxa = 0.d0
ellipse_area = ellipse_area - pi * crowded_rad * crowded_rad * (1.d0 - ellip) !Subtract area of inner ellipse

```
DO i \(=1\), ndata2 ! Rotation of coordinates to match orientation
    \(x d a s h=(x k i(i)-(X I o p)) * \cos (P A)-(e t a(i)-(E T A o p)) * \sin (P A) \quad\) ! of satellite. The new coordinates are then used
    \(y d a s h=(x k i(i)-(X I o p)) * \sin (P A)+(e t a(i)-(E T A o p)) * \cos (P A) \quad!\) to find the major axis of the ellipse a given
    scaled_a (i) \(=\operatorname{SQRT}(y d a s h * * 2 . d 0+(x d a s h * * 2 . d 0 /(1 . d 0-e l l i p) * * 2 . d 0))!\) star lies on - this equates to circular radius.
    IF (scaled_a(i) . le. outer_rad .and. scaled_a(i) .ge. crowded_rad) THEN !
        ellipse_stars = ellipse_stars + 1.d0 ! Count stars in ellipse_area
    END IF
    IF (scaled_a (i) .gt. maxa) THEN
        maxa \(=\) scaled_a(i) \(\quad\) Find major-axis of largest ellipse that passes through field
    END IF ! (weighting will be smallest for stars on this ellipse)
END DO
WRITE ( \(3, *\) ) "Number_of_data_points_in_annulus:", ellipse_stars
WRITE ( \(3, *\) ) "Average_annulus_SNR:", (ellipse_stars - (REAL(bg_stars) * (ellipse_area/bg_area))) / (REAL(bg_stars) * (ellipse_area/
        bg_area) )
den_prof_scale = ((ellipse_stars/ellipse_area) - (bg_stars/bg_area)) * ellipse_area
den_prof_scale \(=\) den_prof_scale \(/\left(2 . d 0 * p i * S R *\left(\left(\exp \left(-c r o w d e d \_r a d / S R\right) *\left(S R+c r o w d e d \_r a d\right)\right)-\left(e x p\left(-o u t e r \_r a d / S R\right) *(S R+\right.\right.\right.\)
    outer_rad)))
den_prof_scale \(=\) den_prof_scale \(/(1 . d 0-\) ellip)
\(!/ \backslash\) Calculate scaling factor of elliptical function of shape defined by HLR, ellipticity and PA.
! \| This is calculated by insuring the total number of stars under the curve matches the number of stars in ellipsearea
! Where there is an inner or outer cutoff radius, it is not absolutely necessary to account for this in the scaling
!as the exponential profile should account for the variations in density across annuli but for completeness, scaling is achieved
! by only measuring the density and number of stars in the annulus used (this becomes very important if a huge HLR is set to remove
! weighting as the profile will no longer account for the variation in density in this case!)
DO i \(=1\), ndata2 !
    IF (truestar_poly (i) .and. scaled_a (i) .ge. crowded_rad.and. scaled_a(i). le. outer_rad) THEN
        weight (i) \(=\exp (-1 . d 0 * \operatorname{scaled} a(i) / S R) * d e n_{-}\)prof_scale ! Apply weights to stars based on elliptical
    ELSE
        weight (i) \(=\exp (-1 . d 0 * \operatorname{maxa} / S R) *\) den_prof_scale \(\quad\) Artificial stars are given the lowest possible
    END IF !weight in this step.
    IF (weight(i) .gt. maxweight) THEN
        maxweight \(=\) weight (i) !Determine maximum weight for use in
    END IF ! plotting the weighted LF
END DO
! \| All subsequent lines in this subroutine are for plotting the density function
! \/ "density.ps" and for checking the scaling relative to the data.
```

Density = 0.e0
maxa = sig_field_radius
DO i = 1, ndata2 !Calculate number of stars in each density bin (i.e. elliptical annulus)
IF (scaled_a(i) . lt . maxa) THEN
Density(INT((scaled_a(i)/maxa) * (rhobins)) + 1, 2) = \&
Density(INT((scaled_a(i)/maxa) * (rhobins)) + 1, 2) + 1.e0
END IF
IF (scaled_a (i) .eq. maxa) THEN
Density(rhobins,2) = Density(rhobins,2) + 1.d0
END IF
END DO
Densitysum = 0.d0
DO i = 1, rhobins !Calculate density of stars in each density bin
Density(i,1)=(\boldsymbol{REAL}(i)/REAL(rhobins)) * REAL(maxa) ! radius of bin
IF (i .eq. 1) THEN
Den_sig(i) = SQRT(Density (i,2))/ (pi*(Density(i,1)**2.e0)*(1.e0 - ellip)) !<---Error bars for bin ---------------------------

```

```

    Densitysum = (Density(i,2) - (bg_stars/bg_area)) * (pi*(Density(i,1)**2.e0)*(1.e0 - ellip))
    ELSE
    Den_sig(i) = SQRT(Density(i,2))/ ((pi*(Density(i,1)**2.e0)*(1.e0 - ellip)) - (pi*(Density(i - 1,1)**2.e0)*(1.e0 - ellip)))!<-! !
    Density (i,2) = Density (i,2)/ ((pi*(Density (i,1)**2.e0)*(1.e0 - ellip)) - (pi*(Density (i-1,1)**2.e0)*(1.e0 - ellip)))!<-------!
    Densitysum = Densitysum + (Density(i,2) - (bg_stars/bg_area)) * ((pi*(Density (i,1)**2.e0)*(1.e0 - ellip)) - (pi*(Density(i-1,1)**2.
        e0)*(1.e0 - ellip)))
    END IF
    END DO
weightsum = 0.d0
DO i = 1, 500 !Calculate values of fitted density profile
weightplot(i,1)=(\boldsymbol{REAL}(\textrm{i})/500.e0) * maxa
weightplot(i,2) = exp(-1.e0*(weightplot(i,1))/SR)* den_prof_scale
IF (i . eq. 1) THEN
weightsum = weightplot(i,2) * (pi*(weightplot(i,1)**2.e0)*(1.e0 - ellip))
ELSE
weightsum = weightsum + weightplot(i,2) * ((pi*(weightplot(i,1)**2.e0)*(1.e0 - ellip)) - (pi*(weightplot(i-1,1)**2.e0)*(1.e0 -
ellip)))
END IF
END DO
weightplot(:,2) = weightplot(:,2) + (bg_stars/bg_area)
WRITE (3,*) "stars_in_model/stars_in_largest_field_ellipse:", weightsum/ Densitysum
WRITE (3,*) "stars_in_annulus/stars_in_largest_field_ellipse:", \&
(((ellipse_stars/ellipse_area) - (bg_stars/bg_area)) * ellipse_area)/ Densitysum
CALL WeighterPlots
END SUBROUTINE Weighter
SUBROUTINE WeighterPlots ! Plots (log) binned density profile of object and
USE Global !superimposes the best fit model to the
IMPLICIT NONE ! density profile
INTEGER :: lw
!--_-_-_-_-_-_-_-_-_-_-_ots Density Profile histogram
string = TRIM(ADJUSTL(field)) // '/density.ps/CPS,

```

\section*{613} 614 615
```

CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(0., 1.1 * REAL(maxa), 0., 1.1*MAXVAL((/MAXVAL(weightplot (:,2)),MAXVAL(Density (:,2) + Den_sig)/)), 0, 0)
CALL pgslw (10)
CALL pgsci (1)
DO i = 1, rhobins ! Plot density bin values
IF (Density(i,1) . le. crowded_rad .or. Density(i,1) .gt. outer_rad) THEN
CALL pgsci(5) !Outside of fitted region
CALL pgpt (1, Density(i,1), Density(i,2), -1)
ELSE
CALL pgsci(1) !In fitted region
CALL pgpt (1, Density(i,1), Density(i,2), -1)
END IF
END DO
CALL pgsci(1)
CALL pgslw (1)
DO i = 1, rhobins !Plot error bars for density bin values
IF (Density(i,1) . le. crowded_rad . or. Density(i,1) .gt. outer_rad) THEN
CALL pgsci(5) !Outside of fitted region
CALL pgerry(1, Density(i,1), Density(i,2) + Den_sig(i), Density(i,2) - Den_sig(i), 5.)
ELSE
CALL pgsci(1) !In fitted region
CALL pgerry(1, Density(i,1), Density(i,2) + Den_sig(i), Density(i,2) - Den_sig(i), 5.)
END IF
END DO
CALL pgsci(2) ! Plot fit to density bins of object
CALL pgline (500, weightplot(:,1), weightplot(:,2))
CALL pgsci(3)
CALL pgline (2, (/0., REAL(sig_field_radius)/), (/REAL(bg_stars/bg_area), REAL(bg_stars/bg_area)/))
CALL pgptxt(0.05*REAL(sig_field_radius), 1.2*REAL(bg_stars/bg_area), 0., 0.5, 'BG')
CALL pgsci(1)
CALL pglab('Elliptical_Radius_(degrees)', 'Object_stars_per_sq._degree', ',')
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./, // TRIM(ADJUSTL(field)) // \&
,/density.ps_./, // TRIM(ADJUSTL(field)) // \&
,/density.jpg,
call system(command)
-----------------------------------------------
string = TRIM(ADJUSTL(field)) // '/ sig_field_cc_w.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MAXVAL(xki, mask = xki . ne. 0.), MINVAL(xki, mask = xki . ne. 0.), \&
MINVAL(eta, mask = eta .ne. 0.), MAXVAL(eta, mask = eta .ne. 0.), 1, 0)
DO i = 1, 20
CALL pgscr(i, 0.5*(SIN (1.0*REAL}((\textrm{i}+10)*\textrm{pi}/10))+1.),\&
0.5*(SIN (1.0*REAL}((\textrm{i}-5)* pi/10)) + 1.),\& !Assign colours to indicie
0.5*(SIN (1.0*REAL}((i)*pi/10))+1.)
END DO
DO i = 1, ndata2
lw = nint((weight(i)/maxweight) * 20.d0) + 1
CALL pgslw(lw+5)

```
```

    CALL pgsci(22 - lw)
    IF (truestar_poly(i)) THEN !If star is real (not artifical) ...
        IF (scaled_a(i) .ge. crowded_rad .and. scaled_a(i).le. outer_rad) THEN !Fitted stars
            CALL pgpt (1, xki(i), eta(i), -1)
        ELSE
            CALL pgslw (1)
            IF (scaled_a(i) . lt. crowded_rad) THEN !stars inside crowded region
            CALL pgsci(3)
            CALL pgpt (1, xki(i), eta(i), 2779)
            END IF
            IF (scaled_a(i) .gt. outer_rad) THEN !stars outside fitted region
            CALL pgsci(20)
            CALL pgpt (1, xki(i), eta(i), 227)!2281)
            END IF
        END IF
    ELSE !If star is artificial...
        CALL pgslw(1)
        CALL pgsci(21)
        CALL pgpt (1, xki(i), eta(i), 225)
    END IF
    END DO
CALL pgscr(1, 0., 0., 0.)
CALL pgslw (1)
CALL pgsci(1)
CALL pglab('$0640)-(degrees)', '\(0633)-(degrees)',,',)
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // &
    '/sig_field_cc_w.ps_./' // TRIM(ADJUSTL(field)) // &
    '/sig_field_cc_w.jpg,
call system(command)
!--------------------------------------------
string = TRIM(ADJUSTL(field)) // '/sig_cmd_w.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)), 1, 1)
CALL pgenv(MINVAL(g_min_i, mask = g_min_i .ne. 0.), MAXVAL(g_min_i), &
MAXVAL(mag_i), MINVAL(mag_i, mask = mag_i .ne. 0.), 0, 0)
DO i = 1, 20
    CALL pgscr(i, 0.5*(SIN (1.0*REAL}((\textrm{i}+10)* pi/10)) + 1.),&
                0.5*(SIN(1.0*REAL}((\textrm{i}-5)*\textrm{pi}/10))+1.),&\quad!Assign colours to indicie
                0.5*(SIN(1.0*REAL}((\textrm{i})*\textrm{pi}/10))+1.))
END DO
DO i = 1, ndata2
    lw = nint((weight(i)/maxweight) * 20.d0) + 1
    CALL pgslw(lw +5)
    CALL pgsci(22 - lw)
    IF (truestar_poly(i)) THEN !If star is real (not artifical) ...
        IF (scaled_a(i) .ge. crowded_rad . and. scaled_a(i).le. outer_rad) THEN !Fitted stars
            CALL pgpt (1, REAL(g_min_i_poly(i)), REAL(data(i)), -1)
        ELSE
            CALL pgslw(1)
            IF (scaled_a(i) . lt. crowded_rad) THEN !stars inside crowded region
            CALL pgsci(3)
            CALL pgpt (1, REAL(g_min_i_poly(i)), REAL(data(i)), 2779)
            END IF
            IF (scaled_a(i) .gt. outer_rad) THEN !stars outside fitted region
```
```
            CALL pgsci(20)
            CALL pgpt (1, REAL(g_min_i_poly(i)), REAL(data(i)), 227)
            END IF
            END IF
        ELSE !If star is artificial...
            CALL pgslw(1)
            CALL pgsci(21)
            CALL pgpt (1, REAL(g_min_i_poly(i)), REAL(data(i)), 225)
    END IF
END DO
CALL pgscr(1, 0., 0., 0.)
CALL pgscr(2, 1., 0., 0.)
CALL pgslw (3)
CALL pgsci(1)
CALL pgpt (ndata, g_min_i, mag_i, -1)
CALL pgslw(1)
CALL pglab('(g_-_i)\d0\u',, 'i\d0\u',',')
CALL PolySelect
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./, // TRIM(ADJUSTL(field)) // &
            '/sig_cmd_w.ps_./' // TRIM(ADJUSTL(field)) // &
            ,/sig_cmd_w.jpg,
call system(command)
                                    Signal-CMD-(included stars only)----------------------
DO i = 1, ndata
    xdash = (xi_all(i) - (XIop)) * cos(PA) - (eta_all(i) - (ETAop)) * sin(PA)
    ydash = (xi_all(i) - (XIop)) * sin(PA) + (eta_all(i) - (ETAop)) * cos(PA)
    scaled_a_all(i) = SQRT(ydash**2.d0 + (xdash **2.d0 / (1.d0 - ellip) **2.d0))
END DO
string = TRIM(ADJUSTL(field)) // ,/sig_cmd_used.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(MINVAL(g_min_i, mask = g_min_i . ne. 0.), MAXVAL(g_min_i), &
MAXVAL(mag_i), MINVAL(mag_i, mask = mag_i .ne. 0.), 0, 0)
DO i = 1, ndata
    IF (scaled_a_all(i) .ge. crowded_rad .and. scaled_a_all(i) .le. outer_rad) THEN
        CALL pgpt (1, REAL(g_min_i(i)), REAL(mag_i(i)), -1)
    END IF
END DO
CALL pglab('(g_--i)\d0\u',, 'i\d0\u',',')
CALL PolySelect
CALL pgend
WRITE (command,*) 'convert^-rotate_90_./' // TRIM(ADJUSTL(field)) // &
    ,/sig_cmd_used.ps_./, // TRIM(ADJUSTL(field)) // &
    '/sig_cmd_used.jpg,
call system(command)
```
```
END SUBROUTINE WeighterPlots
SUBROUTINE SVDFitter !For fitting polynomial to the
USE Global !background field luminosity function
IMPLICIT NONE
INTEGER :: ntmp
xa = bg_histo_coarse(:,1)
ya = bg_histo_coarse(:,2)
xt = xa
yt = ya
sig = 1.e0
! Shift the array in steps of 1 until the first element does not contain a zero
shiftloop: do
    xt = cshift(xt,1)
    yt = cshift(yt,1)
    if ( yt(1) > 0.1 ) exit shiftloop
end do shiftloop
ntmp = 0
countloop: do i = 1 , ndat
    if ( yt(i) > 0.1 ) then
            ntmp = ntmp + 1
    else
            exit countloop
    end if
end do countloop
    xt = xt - 21.
    CALL svdfit(xt,yt, sig,ntmp-1,ay,ma,u,v,w,mp,np,chisq,funcs)
CALL BG_DataHist
END SUBROUTINE SVDFitter
SUBROUTINE BG_DataHist ! Produces plot of background field luminosity
USE Global !function and polynomial fit
IMPLICIT NONE
bfm = 0.d0
DO i = 1, ndat
    DO j = 1 , np
        bfm(i) = bfm(i) + ay(j) * (xa(i) -21.) ** (j -1)
    END DO
END DO
!-----------------------Plots best fit model over coarse histogram
string = TRIM(ADJUSTL(field)) // '/bckgrdfit.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
```

CALL pgbin (ndat, REAL(bg_histo_coarse (: , 1) ), REAL(bg_histo_coarse (: , 2) ), . true.)
CALL pgsci(2)
CALL pgline (ndat, xa, REAL(bfm))
CALL pgsci(1)
CALL pglab ('i \(\backslash \mathrm{d} 0 \backslash \mathrm{u}$, , 'counts, , ', $)$
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./, // TRIM(ADJUSTL(field)) // \&
,/bckgrdfit.ps_./' // TRIM(ADJUSTL(field)) // \&
,/bckgrdfit.jpg,
call system (command)
END SUBROUTINE BG_DataHist
SUBROUTINE MCMC ! The master Markov Chain MonteCarlo routine
USE Global !creates a new model at each iteration and then compares
IMPLICIT NONE ! the quality of the fit to the data
!*** Most subroutines are called from 'MCMC' ***
REAL* 8 :: gasdev
string $=$ TRIM(ADJUSTL(field)) // ,/MCMC_steps.dat ${ }^{\text {, }}$
$\operatorname{OPEN}(2, \mathbf{f i l e}=\operatorname{TRIM}(\operatorname{ADJUSTL}($ string $))$, status = 'unknown')
$\operatorname{OPEN}(8$, file=TRIM(ADJUSTL(field)) // '/MCMC_steps_unf.dat', form = 'unformatted, status = 'unknown')

not_scout $=$.false. ; scout-counts $=0$
known_f $=\left(\boldsymbol{R E A L}\left(b g_{-}\right.\right.$stars $) *$ sig_area) $/(\boldsymbol{R E A L}($ sig_stars $) *$ bg_area $)$
$x 1(1,:)=20.5 \mathrm{~d} 0 ; x 2(1,:)=$ known_f; $x 3(1,:)=0.3 \mathrm{~d} 0$; time $(1)=1$
1 IF (not_scout) THEN ! Set values
$\mathrm{x} 1(1,:)=\mathrm{x} 1(200,:) ; \mathrm{x} 2(1,:)=\mathrm{x} 2(200,:) ; \mathrm{x} 3(1,:)=\mathrm{x} 3(200,:) ;$ time $(1)=1 \quad$ !after
END IF !scout run
mag_tip $=\mathrm{x} 1(1,1) ; \mathrm{f}=\mathrm{x} 2(1,1) ; \mathrm{a}=\mathrm{x} 3(1,1)$
$\mathrm{cn}=1$; beta $=$ betaholder (1)
CALL ModelMake ! Make model and
CALL Convolution !
DO $\mathbf{j}=1$, num_chains !evaluate goodness of fit
$\mathrm{cn}=\mathrm{j} ;$ beta $=$ betaholder $(\mathrm{cn})$ !
CALL Loglike ! for initial parameter choices
$\operatorname{LikeA}(\mathrm{cn})=\log \mathrm{L}(\mathrm{cn}) \quad$ !
END DO
$\operatorname{LikeB}(:)=0 . d 0$
$x 1(2,:)=x 1(1,:) ; x 2(2,:)=x 2(1,:) ; x 3(2,:)=x 3(1,:)$
Best_Combo $(5)=-9 . d 99$
DO it $=2$, nit
time (it) $=\mathrm{it}$
DO $\mathrm{cn}=1$, num_chains
beta $=$ betaholder (cn)
$\mathrm{p}(1)=\mathrm{x} 1(\mathrm{it}, \mathrm{cn})+\mathrm{m}$ _step $(\mathrm{cn}) *$ gasdev(idum) ! Gaussian weighted steps from initial
$\mathrm{p}(2)=\mathrm{x} 2(\mathrm{it}, \mathrm{cn})!+\mathrm{f}_{\mathrm{s}} \mathrm{step}(\mathrm{cn}) * \operatorname{gasdev}(\mathrm{idum})$ ! parameters for the tip magnitude (p(1))
$\mathrm{p}(3)=\mathrm{x} 3(\mathrm{it}, \mathrm{cn})+\mathrm{a}$ step(cn) *gasdev(idum) ! noise ratio (p(2)) and slope (p(3))
IF ( $\mathrm{p}(1)$. lt. blim .or. $\mathrm{p}(1)$.gt. flim) THEN
$r=0 . d 0$
else IF ( $\mathrm{p}(2)$. le. $0 . \mathrm{d} 0$.or. $\mathrm{p}(2)$.ge. 1.d0) THEN ! Restrictions on
$\mathrm{r}=0 . \mathrm{d} 0 \quad$ !whether proposed
else IF ( $\mathrm{p}(3)$. le. $0 . \mathrm{d} 0$.or. $\mathrm{p}(3)$.ge. 2.d0) THEN ! step is taken
$r=0 . d 0$
else
mag_tip $=\mathrm{p}(1) \quad ; \mathrm{f}=\mathrm{p}(2) ; \mathrm{a}=\mathrm{p}(3)$
CALL ModelMake ! Make model and
CALL Convolution !evaluate the
CALL Loglike ! goodness of fit
$\operatorname{LikeB}(\mathrm{cn})=\log \mathrm{L}(\mathrm{cn})$
$\mathrm{r}=10 * *(\operatorname{LikeB}(\mathrm{cn})-\operatorname{LikeA}(\mathrm{cn}))$
end IF
IF (cn .eq. 1 .and. not_scout) THEN Only counts after the scouting run contribute to the ppds
post-y1(INT ((x1 (it, 1) $-18 . d 0) * 10 *$ binspm +1$))=\&$
post_yl(INT((x1 (it, 1) - 18.d0) $* 10 *$ binspm +1$))+1 . d 0$
post-y2 (INT $(x 2($ it, 1$) *(n b i n s-1))+1)=\& \quad$ ! Generate posterior plot
post-y2(INT $(x 2($ it, 1$) *($ nbins -1$))+1)+1 . d 0 \quad$ !for mag_tip, f and a
post_y3 (INT $(x 3($ it, 1$) *(n b i n s-1))+1)=\&$
post-y3(INT(x3(it,1)*(nbins -1$))+1)+1 . d 0$
WRITE (2, $\left.(6 \mathrm{~F} 16.5)^{\prime}\right)$ time (it), $\mathrm{x} 1(\mathrm{it}, \mathrm{cn})$, $\mathrm{x} 2(\mathrm{it}, \mathrm{cn}), \mathrm{x} 3(\mathrm{it}, \mathrm{cn})$, LikeA(cn), LikeB(cn)
WRITE (8) time(it), x1(it,cn), x2(it,cn), x3(it,cn), LikeA(cn), LikeB(cn)
$!/ \backslash$ Prints current parameter values and their likelihood (LikeA) as
! \| well as the likelihood of the current proposed swap (LikeB)
IF (LikeA (cn) .gt. Best_Combo (5)) THEN
Best_Combo(1) $=$ time (it) ; Best_Combo(2) $=x 1(i t, c n)$ !Update best likelihood
Best_Combo (3) = x2 (it, cn) ; Best_Combo (4) =x3(it, cn) !combination encountered
Best_Combo (5) $=$ LikeA(cn) ; Best_Combo (6) $=\operatorname{LikeB(cn)~!~}$
END IF
END IF
CALL random_number (randnum3)
IF (it . lt. nit) THEN
IF (randnum3 . le . r) THEN
$\mathrm{x} 1(\mathrm{it}+1, \mathrm{cn})=\mathrm{p}(1)$
$\mathrm{x} 2(\mathrm{it}+1, \mathrm{cn})=\mathrm{p}(2)$
$\mathrm{x} 3(\mathrm{it}+1, \mathrm{cn})=\mathrm{p}(3) \quad$ ! Decide
likeA(cn) $=\operatorname{likeB}(\mathrm{cn}) \quad$ ! whether
ELSE ! to take
$\mathrm{x} 1(\mathrm{it}+1, \mathrm{cn})=\mathrm{x} 1(\mathrm{it}, \mathrm{cn}) \quad$ !step
$\mathrm{x} 2(\mathrm{it}+1, \mathrm{cn})=\mathrm{x} 2(\mathrm{it}, \mathrm{cn}) \quad!$
$\mathrm{x} 3(\mathrm{it}+1, \mathrm{cn})=\mathrm{x} 3(\mathrm{it}, \mathrm{cn}) \quad!$
$\operatorname{likeA}(\mathrm{cn})=\operatorname{likeA}(\mathrm{cn}) \quad$ !
END IF
END IF
END DO
!/ \ RUN MULTIPLE
! || MCMC CHAINS
IF (scout_counts . 1t. 200) THEN !
scout_counts $=$ scout_counts $+1!$
cycle !
END IF $\quad!200$ iterations will be run at the beginning before the
IF ( scout_counts .eq. 200) THEN
not_scout $=$ true. !' nit, used iterations in order to remove the lead in trail.
```    scout_counts = 1000000000         goto 1     END IF     !| PARALLEL     !\/ TEMPERING     IF (num_chains .ne. 1 .and. it .ne. nit) THEN         CALL random_number(randnum4)         IF (randnum4 .le. swaprate) THEN         CALL random_number(randnum4)         randint = INT((num_chains - 1) * randnum4) + 1         IF (randint .eq. 1) THEN ! Count number of proposed             chain_compare = chain_compare + 1. !swaps with the         END IF !cold sampler chain             PTAR = (Betaholder (randint)/ Betaholder (randint + 1)) * LikeA(randint + 1) +&                 (Betaholder(randint + 1)/ Betaholder(randint)) * LikeA(randint) - &                 LikeA(randint) - LikeA(randint + 1)         CALL random_number(randnum5)         IF (randnum5 . le. 10 ** PTAR) THEN             IF (randint .eq. 1) THEN !Count number of accepted             swap_count = swap_count + 1 !swaps with the             END IF !cold sampler chain             par_hold(1) = x1(it+1, randint) !             par_hold(2) = x2(it+1, randint) !             par_hold(3)=x3(it+1, randint) !             par_hold(4) = LikeA(randint)             x1(it+1, randint) = x1(it+1, randint + 1)             x2(it+1, randint) = x2(it +1, randint + 1) !Swap the parameter             x3(it+1, randint) = x3(it + 1, randint + 1) !values and             LikeA(randint) = LikeA(randint + 1)*& !likelihoods             (Betaholder(randint)/Betaholder(randint + 1)) !between chains             x1(it+1, randint + 1) = par_hold(1)             x2(it+1, randint + 1) = par_hold(2)             x3(it+1, randint + 1) = par_hold(3)             LikeA(randint + 1) = par_hold(4) *&             (Betaholder(randint + 1)/Betaholder(randint))         END IF         END IF END IF END DO x1(1,:) = x1(200,:) ; x2(1,:) = x2(200,:) ; x3(1,:) = x3(200,:) !Remove initial x1(2,:) = x1(201,:) ; x2(2,:) = x2(201,:) ; x3(2,:) = x3(201,:) !parameter values WRITE (2, '(6F16.5)') Best_Combo(1), Best_Combo(2), Best_Combo(3), &         Best_Combo (4), Best_Combo(5), Best_Combo(6) DO i = 1, 10*(nbins - 1)+1     post_x 1(i) = 18.d0 +(REAL(i) - 1.d0)/REAL(10*binspm)! END DO DO i = 1, nbins !x-values of posterior     post_x 2(i) = (REAL(i) - 1.d0)/REAL(nbins - 1) !histograms created above END DO DO i = 1, 2*nbins - 1     post_x3(i) = (REAL(i) - 1.d0)/REAL(nbins - 1) END DO END SUBROUTINE MCMC```
$\qquad$
```SUBROUTINE PosteriorPlot !Produces histogram plots of the posterior distributions USE Global !in the tip magnitude and LF slope IMPLICIT NONE post_yl = post_y1/nit ; post_y2 = post_y 2/nit ; post-y3 = post-y3/nit```

```string = TRIM(ADJUSTL(field)) // '/mag_tip_postplot.ps/CPS' CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1) CALL pgenv(REAL(MINVAL(post_x 1, mask = post-y1 .ne. 0.)), &     REAL(MAXVAL(post_x 1, mask = post_yl .ne. 0.)), 0., &     1.1*REAL(MAXVAL(post_y 1)), 0, 0) CALL pgbin (10*(nbins-1)+1, REAL(post_x 1), REAL(post_y1),.false.) CALL pglab('Proposed,i\d0\u_tip_magnitude', 'Probability', ',') CALL pgend WRITE (command,*) 'convert^-rotate&90\_./' // TRIM(ADJUSTL(field)) // &     ,/mag_tip_postplot.ps_./' // TRIM(ADJUSTL(field)) // &     ,/mag_tip_postplot.jpg, call system(command)```

```string = TRIM(ADJUSTL(field)) // '/f_and_a_postplot.ps/CPS, CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1) IF (MAXVAL(post_y 3) .ge. MAXVAL(post_y2)) THEN     CALL pgenv(0., 2., 0., 1.1*REAL(MAXVAL(post_y 3)), 0, 0) ELSE     CALL pgenv(0., 2., 0., 1.1*REAL(MAXVAL(post_y 2)), 0, 0) END IF CALL pgsci(2) CALL pgbin (nbins, REAL(post-x2), REAL(post-y2),.false.) CALL pgsci(3) CALL pgbin (2*nbins - 1, REAL(post_x 3), REAL(post_y 3),.false.) CALL pgsci(1)```

```CALL pgend WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // &     '/f_and_a_postplot.ps_./' // TRIM(ADJUSTL(field)) // &     ,/f_and_a_postplot.jpg, call system(command) post_y1 = post_y1*nit ; post_y2 = post_y 2*nit ; post_y3 = post_y 3*nit END SUBROUTINE PosteriorPlot SUBROUTINE OtherPlots !MCMC related plots - i.e. plots each parameter vs USE Global !iteration number and vs. each other IMPLICIT NONE```
```1101 102 1 0 3 1104 106```
CALL pgpoint (nit, REAL(x1(:,1)), REAL(x3(:,1)), -1)
CALL pgslw(1)
``````
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // \&
//a_vs_mag_tip.ps_./' // TRIM(ADJUSTL(field)) // \&
,/a_vs_mag_tip.jpg,
call system(command)
---------_--_
string = TRIM(ADJUSTL(field)) // '/a_vs_f.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string)), 1, 1)
CALL pgenv (0.9*REAL(MINVAL(x2 (:, 1))), 1.1*REAL(MAXVAL(x2(:, 1))), 0.9*REAL(MINVAL(x3 (:, 1))), 1.1*REAL(MAXVAL(x3(:, 1))), 0, 0)
CALL pgslw(3)
CALL pgpoint (nit, REAL(x2(:,1)), REAL(x3(:,1)), -1)
CALL pgslw(1)
``````
CALL pgend
WRITE (command,*) 'convert\_-rotate «90_./' // TRIM(ADJUSTL(field)) // \&
'/a_vs_f.ps_./' // TRIM(ADJUSTL(field)) // \&
,/a_vs_f.jpg,
call system(command)
END SUBROUTINE OtherPlots
SUBROUTINE NoiseMake !Generates a polynomial of degree 7 that follows the
USE Global !functional form of the GSS background LF. The polynomial
IMPLICIT NONE !coefficients were derived in 'BackgroundPolyFit, using
!'svdfit, from Numerical Recipes.
area2 = 0.d0
DO i = 1, 8 * binspm + 1
modelnoise(i,1) = 18.d0 + (i 1 1.d0)/REAL(binspm)
modelnoise(i,2) = 0.d0
DO j = 1, np !Set background counts
modelnoise(i,2) = modelnoise(i,2) + ay(j) * (modelnoise(i,1) - 21.d0) ** (j - 1)
END DO
IF (modelnoise(i,2) . 1t . 0.d0) THEN !
modelnoise(i,2)=0.d0 !Insure no negative counts
END IF
IF (i .ge. blimBins .and. i .le. flimBins) THEN
area2 = area 2 + modelnoise(i,2) !Used for normalization in 'ModelMake'
END IF
END DO
model(:,2) = modelnoise(:,2) / area2
CALL Convolution
noise = cmodel(:,2)
END SUBROUTINE NoiseMake
``` ```
223
1224
1225
1226
1227
1228
1229
SUBROUTINE NoisePlot ! Plots the unscaled form of the background LF
USE Global
IMPLICIT NONE
``` ```
CALL pgbegin(0,'?', 1, 1)
``` ```
CALL pgbegin(0,'?', 1, 1)
``` ```
CALL pgenv(REAL(blim), REAL(mag_cutoff), 0., 1.1*REAL(MAXVAL(modelnoise(:,2), mask = modelnoise(:,1) . le. 23.5 .and. modelnoise(:,1)
```CALL pgenv(REAL(blim), REAL(mag_cutoff), 0., 1.1*REAL(MAXVAL(modelnoise(:,2), mask = modelnoise(:,1) . le. 23.5 .and. modelnoise(:,1)     .ge. bg_blim)), 0, 0) CALL pgbin (nbins - INT(2.5*binspm), REAL(modelnoise (:,1)), REAL(modelnoise (:,2)), .true.) CALL pglab('i\d0\u', 'Counts',,',) CALL pgend END SUBROUTINE NoisePlot SUBROUTINE ModelMake !Initial Model (i.e. model before convolution) USE Global IMPLICIT NONE REAL*8 :: func_i area = 0.d0 DO i = 1, nbins     model(i,1) = 18.d0 + (i - 1.d0)/REAL(binspm)     IF (model(i,1) + hb .gt. mag_tip .and. model(i,1) - hb .le. mag_tip) THEN         model(i,2) = ((10.d0**(a*(model(i,1) + hb - mag_tip)))/(a*LOG(10.))) - &                 (1.d0/(a*LOG(10.))) !Model value at tip         area = area + model(i,2) !Used for normalization     ELSE IF (model(i,1) .gt. mag_tip) THEN !Model value faintward of tip         model(i,2) = ((10.d0**(a*(model(i,1) + hb - mag_tip)))/(a*LOG(10.))) - &                 ((10.d0**(a*(model(i,1) - hb - mag_tip)))/(a*LOG(10.)))     ELSE         model(i,2) = 0.d0 !Model value brightward of tip     END IF     IF (i .ge. blimBins .and. i .le. flimBins) THEN         area = area + model(i,2) !Used for normalization     END IF END DO model(:,2) = model(:,2) / area !Normalize END SUBROUTINE ModelMake SUBROUTINE ModelPrint ! Prints model before convolution USE Global IMPLICIT NONE CALL pgbegin(0,'?',1,1) CALL pgenv(REAL(mag_tip) - 3., REAL(mag_cutoff), 0., 1.1*REAL(model(INT(5.5*binspm),2)), 0, 0) CALL pgbin (nbins - INT(2.5* binspm), REAL(model(:,1)), REAL(model(:,2)), .true.) CALL pglab('i\d0\u', 'Counts',,',) CALL pgend END SUBROUTINE ModelPrint```
```1283 128 1286```
SUBROUTINE GaussianKernel !Generates a Gaussian kernel 'kernel' with
```SUBROUTINE GaussianKernel !Generates a Gaussian kernel 'kernel' with USE Global !HWHM (sigma) changing with magnitude in USE Global !HWHM (sigma) changing with magnitude in IMPLICIT NONE !accordance with func_i. Kernel is defined from IMPLICIT NONE !accordance with func_i. Kernel is defined from     !gx = -5*sigma to gx = +5*sigma     !gx = -5*sigma to gx = +5*sigma REAL*8 :: func_i REAL*8 :: func_i kernel = 0.d0 kernel = 0.d0 DO i = 1, nbins DO i = 1, nbins     t = 18.d0 + (i - 1.d0)/REAL(binspm) !Convert bin number to magnitude     t = 18.d0 + (i - 1.d0)/REAL(binspm) !Convert bin number to magnitude     temp = 0.d0     temp = 0.d0     gx =0.     gx =0.     j=0     j=0     DO WHILE (gx .le. 5.e0*func_i(t)) !     DO WHILE (gx .le. 5.e0*func_i(t)) !         j=j+1 !         j=j+1 !         gx = 0.e0 + (j-1.e0)/binspm ! Creates half of         gx = 0.e0 + (j-1.e0)/binspm ! Creates half of         temp(j,1)=gx !the kernel ('temp')         temp(j,1)=gx !the kernel ('temp')         temp(j,2) = exp(-((gx) **2.e0)/(2.e0*(func_i(t)**2.e0)))!         temp(j,2) = exp(-((gx) **2.e0)/(2.e0*(func_i(t)**2.e0)))!     END DO     END DO     ghw(i) = j - 1.d0 !The first non-zero bin of 'cmodel' will be the first     ghw(i) = j - 1.d0 !The first non-zero bin of 'cmodel' will be the first             !non-zero bin of 'model' minus ghw             !non-zero bin of 'model' minus ghw     DO k = 1, j     DO k = 1, j             kernel(k,:,i) = temp(j - (k-1),:) !Create 'kernel' by concatenating             kernel(k,:,i) = temp(j - (k-1),:) !Create 'kernel' by concatenating             kernel(j+k,2,i) = temp(k+1,2) !'temp' with a reflected version             kernel(j+k,2,i) = temp(k+1,2) !'temp' with a reflected version             kernel(j+k,1,i) = -temp(k+1,1) !of itself             kernel(j+k,1,i) = -temp(k+1,1) !of itself     END DO     END DO     !Note: temp (2*j,2) = 0.d0 ; temp (2*j,1) = -0.d0     !Note: temp (2*j,2) = 0.d0 ; temp (2*j,1) = -0.d0     kernel(:,2,i) = kernel(:,2,i)/SUM(kernel(:,2,i))     kernel(:,2,i) = kernel(:,2,i)/SUM(kernel(:,2,i)) END DO END DO END SUBROUTINE GaussianKernel END SUBROUTINE GaussianKernel SUBROUTINE GaussianKernelPrint ! Prints Gaussian Kernel at given magnitude SUBROUTINE GaussianKernelPrint ! Prints Gaussian Kernel at given magnitude USE Global USE Global IMPLICIT NONE IMPLICIT NONE REAL*8 :: func_i REAL*8 :: func_i CALL pgbegin(0,'?', 1, 1) CALL pgbegin(0,'?', 1, 1) CALL pgenv(-5.5 * REAL(func_i(t)), 5.5 * REAL(func_i(t)), 0., 1.1*MAXVAL(REAL(kernel(:, 2,i))), 0, 0) CALL pgenv(-5.5 * REAL(func_i(t)), 5.5 * REAL(func_i(t)), 0., 1.1*MAXVAL(REAL(kernel(:, 2,i))), 0, 0) CALL pgbin (2*ghw(i) + 1, REAL(kernel(:,1,i)), REAL(kernel(:,2,i)), .true.) CALL pgbin (2*ghw(i) + 1, REAL(kernel(:,1,i)), REAL(kernel(:,2,i)), .true.) CALL pglab('Magnitude_offset','Strength', ',') CALL pglab('Magnitude_offset','Strength', ',') CALL pgend CALL pgend END SUBROUTINE GaussianKernelPrint```
END SUBROUTINE GaussianKernelPrint
``` ```
SUBROUTINE Convolution !Convolves initial model with a Gaussian kernel
Use Global !whose width is equal to the photometric error
IMPLICIT NONE !and hence expands with increasing magnitude
cmodel = 0.d0
DO i = 1, nbins
cmodel(i,1) = 18.d0 + (i - 1.d0)/REAL(binspm)
DO j = -ghw(i), ghw(i), +1 !
IF (i .gt.ghw(i) .and. i .lt. nbins - ghw(i)) THEN !Convolve
cmodel(i+j,2) = cmodel(i+j,2) + kernel(ghw (i)+j+1,2,i)*model(i,2) !model with
END IF
!gaussian
END DO
END DO
DO i = nbins, flimBins+1, -1 !Set the faint limit
cmodel(i,2) = 0.d0 !of the final convolved
END DO !model at flim
cmod_nbins = flimBins
!Normalize the convolved model
cmodel(:,2) = cmodel(:,2)/SUM(cmodel(:,2), mask = cmodel(:,1) .ge. blim)
!Note the above step is very important - normalization must only be over the
!range of magnitudes in the 'data, array - i.e. down to blim -> not right the
!way to blim - 1.d0. This was a difficult bug to find!
END SUBROUTINE Convolution
SUBROUTINE ConvolutionPrint ! Prints convolved version of model
USE Global
IMPLICIT NONE
CALL pgbegin(0,'?',1,1)
CALL pgenv(REAL(mag_tip) - 0.5, 25., 0., 1.1*MAXVAL(REAL(cmodel(:,2))), 0, 0)
CALL pgbin (nbins, REAL(cmodel(:,1)), REAL(cmodel(:,2)), .true.)
CALL pglab('i\d0\u', 'Relative cprobability,, ',')
CALL pgend
END SUBROUTINE ConvolutionPrint
SUBROUTINE DataHist !Generates finely and coarsely binned histograms and
USE Global !overlays them with the best fit model determined by
IMPLICIT NONE ! the MCMC
REAL}*8 :: scaled_f_re
histo_fine(:,1) = model(:,1)
DO i = 1, INT(0.25*(nbins - 1.d0)) + 1
histo_coarse(i,1) = 18.d0 + (i-1.d0)/REAL(0.25*binspm)
END DO
DO i = 1, ndata2
histo_fine(INT(REAL((data(i) - 18.d0)*binspm) + 1.d0),2)=\& !Generates
histo_fine(INT(REAL((data(i) - 18.d0)*binspm) + 1.d0),2) + 1.d0 !
``` ```
histo_coarse(INT(REAL((data(i) - 18.d0)*0.25*binspm) + 1.d0),2) = \& !Histograms
histo_coarse(INT(REAL((data(i) - 18.d0)*0.25*binspm) + 1.d0),2) + 1.d0 !
END DO
histo_coarse(INT(REAL(blimbins)/4.e0) + 1,2)=\& !See paragraph
histo_coarse(INT(REAL(blimbins)/4.e0) + 1,2) * 2.d0 !below
histo_coarse(INT(REAL}(flimbins)/4.e0) + 1,2)=\& !See paragraph
histo_coarse(INT(REAL(flimbins)/4.e0) + 1,2) * 2.d0 !below
!For graphing purposes, the first and last bins of the coarse histogram are doubled since
!these bin lies half outside the range of interest and so are depleted by
!roughly one half. This is for graphing only and has no bearing on the
!determined best fit model.
!| Plot Best Fit Model
!\/ over histogram
mag_tip = tip_rec ; f = f_rec ; a = a_rec !
CALL ModelMake !Generate best fit sig function
CALL Convolution
bfm = 0.d0 ; bg = 0.d0
bfm = cmodel(:,2) * (1.d0 - f) !bfm = best fit signal function
bg = noise * f !bg = back ground function
bfm = bfm + bg !Add bfm and background together
bfm = bfm * (SUM(histo_fine(:,2))/SUM(bfm, mask = cmodel(:,1).ge. blim)) !Scale bfm to match histogram
!---------------------------_ots best fit model over fine histogram
string = TRIM(ADJUSTL(field)) // '/model_fit_vs_data_fine.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(REAL(blim), REAL(flim), 0., 1.1*MAXVAL(real(histo_fine(:,2))), 0, 0)
CALL pgbin (nbins, REAL(histo_fine(:,1)), REAL(histo_fine(:,2)), false.)
CALL pgsci (2)
CALL pgslw(5)
CALL pgline (nbins, REAL(histo_fine(:,1)), REAL(bfm))
CALL pgsci(1)
CALL pgslw(1)
CALL pglab('i\d0\u', 'Counts', ',')
CALL pgend
WRITE (command,*) 'convert_-rotate_90_./' // TRIM(ADJUSTL(field)) // \&
'/model_fit_vs_data_fine.ps_./' // TRIM(ADJUSTL(field)) // \&
//model_fit_vs_data_fine.jpg,
call system(command)
bfm = bfm * 4.d0 !Scale bfm to match coarse histogram
!---------------------------ots best fit model over coarse histogram
string = TRIM(ADJUSTL(field)) // '/model_fit_vs_data_coarse.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL( string)),1,1)
CALL pgenv(REAL(blim), REAL(flim), 0., 1.1*MAXVAL(real(histo_coarse(:,2))), 0, 0)
CALL pgbin (INT(0.25*(nbins-1.d0)) + 1, REAL(histo_coarse(:,1)), \&
REAL(histo_coarse(:,2)), .false.)
CALL pgsci(2)
CALL pgline (nbins, REAL(histo_fine(:,1)), REAL(bfm))
``` ```
CALL pgsci(1)
CALL pglab('i\d0\u', 'Counts',',')
CALL pgend
WRITE (command,*) 'convert^-rotate_90_./' // TRIM(ADJUSTL(field)) // \&
'/model_fit_vs_data_coarse.ps_./' // TRIM(ADJUSTL(field)) // \&
,/model_fit_vs_data_coarse.jpg,
call system(command)
END SUBROUTINE DataHist
SUBROUTINE w_DataHist ! Generates finely and coarsely binned (weighted) histograms
USE Global !and overlays them with the best fit model determined by
IMPLICIT NONE ! the MCMC
REAL*8 :: scaled_f_rec
w_histo_fine(:,1) = model(:,1)
DO i = 1, INT (0.25*(nbins - 1.d0)) + 1
w_histo_coarse(i,1) = 18.d0 + (i-1.d0)/REAL(0.25*binspm)
END DO
DO i = 1, ndata2
!
IF (truestar_poly(i)) THEN
IF (scaled_a(i) .ge. crowded_rad . and. scaled_a(i) . le. outer_rad) THEN
w_histo_fine(INT(REAL((data(i) - 18.d0)*binspm) + 1.d0),2) = \&
W_histo_fine(INT(REAL((data(i) - 18.d0)*binspm) + 1.d0),2) + \&
(weight(i)/maxweight)
w_histo_coarse(INT(REAL}((\boldsymbol{data}(\textrm{i})-18.d0)*0.25*binspm) + 1.d0),2)=\& ! while stars at some radius give
w_histo_coarse(INT(REAL((data(i) - 18.d0)*0.25*binspm) + 1.d0),2) + \& !some fraction of 1.0 counts
(weight(i)/maxweight) !depending on the density profile
END IF
END IF
END DO
w_histo_coarse(INT(REAL(blimbins)/4.e0) + 1,2) = \& !See paragraph
w_histo_coarse(INT(REAL(blimbins)/4.e0) + 1,2)*2.d0 !below
w_histo_coarse(INT(REAL(flimbins)/4.e0) + 1,2)=\& !See paragraph
w_histo_coarse(INT(REAL(flimbins)/4.e0) + 1,2) * 2.d0 !below
!For graphing purposes, the first and last bins of the coarse histogram are doubled since
!these bin lies half outside the range of interest and so are depleted by
!roughly one half. This is for graphing only and has no bearing on the
!determined best fit model.
!| Plot Best Fit Model
!\/ over weighted histogram
mag_tip = tip_rec ; f = f_rec ; a = a_rec !
CALL ModelMake !Generate best fit signal function
CALL Convolution
!
bfm = 0.d0 ; bg = 0.d0 !Apply weights to best fit model for ||
DO i = 1, ndata2 !each star and sum together. \/
IF (truestar_poly(i)) THEN
IF (scaled_a(i) .ge. crowded_rad . and. scaled_a(i) . le. outer_rad) THEN
bfm = bfm + cmodel(:, 2) * (weight(i)/(weight(i) + (bg_stars / bg_area))) * weight(i) !sum together RGB LFs from each star
```
$\mathrm{bg}=\mathrm{bg}+$ noise * ((bg_stars / bg_area)/(weight(i) + (bg_stars / bg_area)))* weight(i) !sum together BG LFs from each star END IF
END IF
END DO
$\mathrm{bfm}=\mathrm{bfm}+\mathrm{bg} \quad$ !Add bfm and background together


string $=\operatorname{TRIM}(A D J U S T L(f i e l d)) / / \quad /$ model_fit_vs_data_fine_w.ps/CPS,
CALL pgbegin ( $0, \operatorname{TRIM}(\operatorname{ADJUSTL}($ string $)), 1,1)$

CALL pgbin (nbins, REAL(w_histo_fine (: , 1)), REAL(w_histo_fine (: , 2) ), . false.)
CALL pgsci(2)
CALL pgslw (5)
CALL pgline (nbins, REAL(w_histo_fine (: , 1)), REAL(bfm))
CALL pgsci(1)
CALL pgslw (1)
CALL pglab('i $\backslash \mathrm{d} 0 \backslash \mathrm{u}$ ', ' Weighted_Counts, , ', )

CALL pgend

,/model_fit_vs_data_fine_w.ps./, // TRIM(ADJUSTL(field)) // \&
'/model_fit_vs_data_fine_w.jpg,
call system (command)
bfm $=\mathrm{bfm} * 4 . \mathrm{d} 0 \quad$ ! Scale bfm to match coarse histogram
!--_-_-_-_-_-_-_-_-_Otots best fit model over coarse histogram
string $=$ TRIM(ADJUSTL(field)) // '/model_fit_vs_data_coarse_w.ps/CPS'
CALL pgbegin ( 0 , TRIM (ADJUSTL(string) ) $, 1,1$ )

CALL pgenv(REAL(blim), REAL(flim), 0., $1.1 * \operatorname{MAXVAL}($ real(w_histo_coarse (: , 2))), 0, 0)
CALL pgbin (INT $(0.25 *($ nbins $-1 . d 0))+1$, REAL(w_histo_coarse (: 1$)$ ), \&
REAL(w_histo_coarse (: , 2)), .false.)
CALL pgsci(2)
CALL pgline (nbins, $\operatorname{REAL}\left(w_{\text {_ }}\right.$ histo_fine (: 1$)$ ), $\left.\boldsymbol{R E A L}(b f m)\right)$
CALL pgsci(1)
CALL pglab('i\d0\u', 'Weighted_Counts, , ,')

CALL pgend

,/model_fit_vs_data_coarse_w.ps_./' // TRIM(ADJUSTL(field)) // \&
,/model_fit_vs_data_coarse_w.jpg,
call system (command)

END SUBROUTINE w_DataHist

SUBROUTINE LogLike ! Generates the log
USE Global ! of the likelihood
IMPLICIT NONE ! for a given model
$\log \mathrm{L}(\mathrm{cn})=0 . \mathrm{d} 0$
DO $\mathrm{i}=1$, ndata2

```
    IF (truestar_poly(i)) THEN
    IF (scaled_a(i) .ge. crowded_rad . and. scaled_a(i) .le. outer_rad) THEN
    starbin = INT((data(i) - 18.d0)*binspm) + 1
    sig_prob = cmodel(starbin , 2) * weight(i)/(weight(i) + (bg_stars/ bg_area)) !Determine likelihood for star given current ratio
    bg_prob = noise(starbin) * (bg_stars/ bg_area)/(weight(i) + (bg_stars/ bg_area))!of the RGB LF vs. the BG LF due to star's weight.
    prob = (sig_prob + bg_prob) * weight(i) !This also insures the total prob. of the model
    logL(cn) = logL(cn) + LOG10(prob) !is 1. Add likelihoods together in log space.
    END IF
    ELSE
        cycle
    END IF
END DO
logL(cn)}=\operatorname{log}\textrm{L}(\textrm{cn})*\mathrm{ beta
END SUBROUTINE LogLike
SUBROUTINE TipAndSigma !Identifies the best parameter values and
USE Global !their associated 1 sigma errors from the
IMPLICIT NONE !respective posterior plots.
PPD_peak = 0.d0
DO i = 1, 10*(nbins - 1)+1
    IF (post_y1(i) .gt. PPD_peak) THEN !
        PPD_peak = post_y1(i) !Find best fit TRGB value
        tip_rec = post_x (i) !
    END IF
END DO
PPD_peak = 0.d0
DO i = 1, nbin
    IF (post_y2(i) .gt. PPD_peak) THEN !
        PPD_peak = post_y2(i) !Find best fit f value
        f_rec = post_x (i)
    END IF
END DO
PPD_peak = 0.d0
DO i = 1, 2*nbins - 1
    IF (post_y3(i) .gt. PPD_peak) THEN
        PPD_peak = post_y3(i) !Find best fit a value
        a_rec = post_x3(i) !
    END IF
END DO
tip_kpc = (100.d0**((tip_rec + 3.44d0)/10.d0))/100.d0 ! Distance inferred from
            ! tip magnitude in kpc
tip_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(post_y 1, DIM = 1), 1, -1
    mcounts = mcounts + post_yl(i)
END DO
DO i = MAXLOC(post_y 1, DIM = 1), 1, -1
    tip_counts = tip_counts + post_yl(i) !Finds negative one sigma
    IF (tip_counts .ge. 0.682*mcounts) THEN !error in magnitudes
        tip_msigma = ((\boldsymbol{REAL}(\textrm{i})-1.d0)/REAL}(10*\mathrm{ binspm ) ) + 18.d0 !
        tip_msigma = tip_rec - tip_msigma
        exit
    END IF
END DO
```

```
1 6 4 9
```

tip_counts = 0.d0 ; pcounts = 0.d0

```
tip_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(post_y 1, DIM = 1), 10*(nbins -1)+1
DO i = MAXLOC(post_y 1, DIM = 1), 10*(nbins -1)+1
    pcounts = pcounts + post-y1(i)
    pcounts = pcounts + post-y1(i)
END DO
END DO
DO i = MAXLOC(post_y 1, DIM = 1), 10*(nbins -1)+1
DO i = MAXLOC(post_y 1, DIM = 1), 10*(nbins -1)+1
    tip_counts = tip_counts + post_yl(i) !Finds positive one sigma
    tip_counts = tip_counts + post_yl(i) !Finds positive one sigma
    IF (tip_counts .ge. 0.682* pcounts) THEN !error in magnitudes
    IF (tip_counts .ge. 0.682* pcounts) THEN !error in magnitudes
        tip_psigma = ((REAL(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
        tip_psigma = ((REAL(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
        tip_psigma = tip_psigma - tip_rec
        tip_psigma = tip_psigma - tip_rec
        exit
        exit
    END IF
    END IF
END DO
END DO
d1 = 0 ; d2 = 0 ; d3 = 0 ; d4 = 0
d1 = 0 ; d2 = 0 ; d3 = 0 ; d4 = 0
f_counts = 0.d0 ; a_counts = 0.d0
f_counts = 0.d0 ; a_counts = 0.d0
DO i = 1, nbins
DO i = 1, nbins
    f_counts = f_counts + post_y2(i)
    f_counts = f_counts + post_y2(i)
    a_counts = a_counts + post-y3(i)
    a_counts = a_counts + post-y3(i)
    IF (f_counts .ge. 0.159*nit .and. d1 .eq. 0) THEN !
    IF (f_counts .ge. 0.159*nit .and. d1 .eq. 0) THEN !
        fminsig = post_x2(i)
        fminsig = post_x2(i)
        d1 = 1
        d1 = 1
    END IF
    END IF
    IF (f_counts .ge. 0.841*nit .and. d2 .eq. 0) THEN !For f and a:
    IF (f_counts .ge. 0.841*nit .and. d2 .eq. 0) THEN !For f and a:
        fplusig = post_x2(i) !Finds upper and lower
        fplusig = post_x2(i) !Finds upper and lower
        d2 = 1 !bounds for posterior
        d2 = 1 !bounds for posterior
    END IF !distribution within one
    END IF !distribution within one
    IF (a_counts .ge. 0.159*nit . and. d3 .eq. 0) THEN !sigma of maximum.
    IF (a_counts .ge. 0.159*nit . and. d3 .eq. 0) THEN !sigma of maximum.
        aminsig = post_x3(i)
        aminsig = post_x3(i)
        d3 = 1
        d3 = 1
    END IF
    END IF
    IF (a_counts .ge. 0.841* nit .and. d4 .eq. 0) THEN !
    IF (a_counts .ge. 0.841* nit .and. d4 .eq. 0) THEN !
        aplusig = post_x3(i)
        aplusig = post_x3(i)
        d4 = 1
        d4 = 1
    END IF
    END IF
END DO
END DO
f_sigma = 0.5d0*(fplusig - fminsig) !Hence calculates 1 sigma error
f_sigma = 0.5d0*(fplusig - fminsig) !Hence calculates 1 sigma error
a_sigma = 0.5d0*(aplusig - aminsig) !for f and a
a_sigma = 0.5d0*(aplusig - aminsig) !for f and a
kpc_merr = tip_kpc*100.d0**(tip_msigma/10.d0) - tip_kpc !minus tip error in kpc
kpc_merr = tip_kpc*100.d0**(tip_msigma/10.d0) - tip_kpc !minus tip error in kpc
kpc_perr = tip_kpc*100.d0**(tip_psigma/10.d0) - tip_kpc !plus tip error in kpc
kpc_perr = tip_kpc*100.d0**(tip_psigma/10.d0) - tip_kpc !plus tip error in kpc
END SUBROUTINE TipAndSigma
END SUBROUTINE TipAndSigma
FUNCTION func_i(m) !This function feeds the photometric error as a function
FUNCTION func_i(m) !This function feeds the photometric error as a function
USE Global !of magnitude to the 'GaussianKernel' subroutine.
USE Global !of magnitude to the 'GaussianKernel' subroutine.
IMPLICIT NONE
IMPLICIT NONE
REAL*8 :: func_i, m, c1, c2, c3
REAL*8 :: func_i, m, c1, c2, c3
c1 = 0.001
c1 = 0.001
c3 = log(0.24) - log(0.11)
c3 = log(0.24) - log(0.11)
c2 = c3*25.0 - log(0.24)
c2 = c3*25.0 - log(0.24)
func_i = c1 + exp(c3*m - c2)
```

func_i = c1 + exp(c3*m - c2)

```
```

END FUNCTION

```

```

SUBROUTINE PolySelect !Used for selection of appropriate colour cut
USE Global !in colour-magnitude space
IMPLICIT NONE
integer MAXPT, ipol
integer NPT_ggr,NPT_spatial
parameter (MAXPT=100)
real*4 XCOL_ggr(MAXPT),YMAG_ggr(MAXPT)
real*4 X_spatial (MAXPT), Y_spatial (MAXPT)
logical refine_CMDsel_ggr, refine_spatialsel
!parameter (refine_CMDsel_ggr=.true.)
parameter (refine_CMDsel_ggr=.false.)
!parameter (refine_spatialsel=.true.)
parameter (refine_spatialsel=.false.)
logical in_poly
external in_poly
npt_ggr=0
if (refine_CMDsel_ggr) then
call pgsls(2)
call pgmove(0.2,26.0)
call pgdraw(0.2, 15.0)
call pgsls(1)
call pglcur(MAXPT, NPT_ggr,XCOL_ggr,YMAG_ggr)
open(2,file=TRIM(ADJUSTL(colcut)), status='unknown')
write (2,*) NPT_ggr
do ipol=1,NPT_ggr
write(2,*) XCOL_ggr(ipol),YMAG_ggr(ipol)
end do
close(2)
call pgsci(1)
call pgadvance
else
open(2,file=TRIM(ADJUSTL(colcut)), status='old')
read(2,*) NPT_ggr
do ipol=1,NPT_ggr
read(2,*) XCOL_ggr(ipol),YMAG_ggr(ipol)
end do
close(2)
call pgsci(2)
call pgslw(5)
call pgline(NPT_ggr,XCOL_ggr,YMAG_ggr)
call pgsci(1)
call pgslw(1)
end if

```

```

j=0
DO i = 1, ndata
IF (in_poly(g_min_i(i),mag_i(i),NPT_ggr,XCOL_ggr,YMAG_ggr)) THEN!
IF (mag_i(i) . le. flim .AND. mag_i(i) .ge. blim) THEN !Makes new
j = j+1 !arrays
mag_i_poly(j) = mag_i(i) !containing
mag_g_poly(j) = mag_g(i) !only
g_min_i_poly(j) = g_min_i(i) !stars

```
```

            xi_poly(j) = xki(i) !within
            eta_poly(j) = eta(i) !polygon
            truestar-poly(j) = truestar(i)
            END IF
    END IF
    END DO
ndata2 = j !New number of stars in dataset after colour cut
!----------------------------------------
j=0 ; k = 0
DO i = 1, bg_ndata
IF (in_poly(bg_g_min_i(i),bg_mag_i(i),NPT_ggr,XCOL_ggr, YMAG_ggr)) THEN
IF (bg_mag_i(i) .le. 24.d0) THEN
IF (bg_mag_i(i) . le. flim .AND. bg_mag_i(i) .ge. blim) THEN
k = k+1
END IF
j = j+1
bg_mag_i_poly(j) = bg_mag_i(i)
bg_mag_g_poly(j) = bg_mag_g(i)
bg_g_min_i_poly(j) = bg_g_min_i(i)
END IF
END IF
END DO
bg_ndata2 = j ; bg_ndata 3 = k !Stars in bckgrnd ; Stars in bckgrnd between blim \& flim
END SUBROUTINE PolySelect

```
\(\qquad\)
```

logical function in_poly(x,y,np,xp,yp) !Used by PolySelect subroutine
implicit none
real*4 x,y
integer np
real*4 xp(np),yp(np)
real*4 tiny,xs,xe,ys,ye
parameter (tiny=1.e-5)
real*4 simag, fimag
external fimag
integer j
simag=0.0
do j=1,np
if (j.lt.np) then
xe=xp(j+1)
xs=xp(j)
ye=yp(j+1)
ys=yp(j)
else
xe=xp(1)
xs=xp(j)
ye=yp(1)
ys=yp(j)
end if
simag=simag+ fimag(x,xs,xe,y,ys,ye)
end do
if (abs(simag).gt.tiny) then
in_poly=.true.

```
```

else
in_poly=.false.
end if
end

```

```

real*4 function fimag(x0,xs,xe,y0,ys,ye) !Used by PolySelect subroutine
implicit none
real*4 x0,xs,xe,y0,ys,ye
real*4 top, bot
top = -(xe-x0)*(ys-y0) +(ye-y0)*(xs-x0)
bot= (xe-x0) * (xs-x0) + (ye-y0) * (ys-y0)
fimag=atan2(top,bot)
end

```
\(\qquad\)
```

-Libpress Algorithms --_-_-_-_-_-_-_-_-_-_-_-_

```

\section*{Program: MF_TRGB_Feed.pl}

Creation Date: 23 January 2012
Relevant Section: Ch. 4
Notes: This Perl scrip shows the individual parameters for each satellite fed to the program 'MF_TRGB.f95.' I have included it as it provides information specific to each satellite that is not given in Ch. 4. For each satellite, there are 16 inputs in the order described below. Note that for the dwarf spheroidal satellites, values for parameters 2-10 were provided by Nicolas Martin (Observatoire Astronomique, Universite de Strasbourg) and are due for publication in the near future. As an aside, it is worth noting that the weighting can effectively be turned off by specifying a very large Half-Light Radius, which produces an essentially flat object density profile across the field of view.
1. Object Name
2. Right Ascension Coordinate (hours)
3. Right Ascension Coordinate (minutes)
4. Right Ascension Coordinate (seconds)
12. Outer Cutoff Radius
13. Object Field Radius
14. Background Field Right Edge (Xi)
15. Background Field Left Edge (Xi)
16. File Name for Colour-Cut Polygon
```

\#!/usr/bin/perl
system("./MF_TRGB.e_AndromedaIe_0.0_45.0_40.0_38.0_2.0_18.5_0.26_3.9_25.0_0.0_0.3_0.3_0.0_2.0_ANDI.CMD"); \#f final
print "AndromedaI}|\mathrm{ done. \n";
system("./MF_TRGB.e_AndromedaIIe_1.0_16.0_26.9_33.0_26.0_1.9_0.13_5.0_27.0_0.0_0.4_0.4_5.0_8.0_ANDII.CMD"); \#final
print "AndromedaII_done.\n";

```

```

print "AndromedaIII_done.\n";
system("./MF_TRGB.e_AndromedaVe_1.0_10.0_17.1_47.0_37.0_45.4_0.27_1.6_41.0_0.011_0.2_0.2_3.0_6.0_ANDV.CMD"); \#final
print "AndromedaV_done.\n";

```

```

print "AndromedaIX_done.\n";

```
```

system("./MF_TRGB.e_AndromedaXe_1.0_6.0_35.5_44.0_48.0_30.9_0.41_1.4_33.0_0.0_0.15_0.15_3.2_5.0_2_ANDX.CMD"); \#final
print "AndromedaX_done.\n";

```

```

print "AndromedaXI_done.\n";
system("./MF_TRGB.e_AndromedaXIIe_0.0_47.0_27.0_34.0_22.0_29.0_0.0_1.1_0.0_0.0_0.15_0.15_-0.5_2.5_ANDXII.CMD"); \#final
print "AndromedaXII_done.\n";
system("./MF_TRGB.e_AndromedaXIIIe_0.0_51.0_50.9_33.0_0.0_14.5_0.60_0.8_-25.0_0.0_0.15_0.15_0.4_3.4_ANDXIII.CMD"); \# final
print "AndromedaXIII_done.\n";

```

```

print "AndromedaXIV_done.\n";
system("./MF_TRGB.e_AndromedaXVe_1.0_14.0_18.8_38.0_7.0_18.0_0.23_1.4_33.0_0.0_0.2_0.2_4.7_7.7_ANDXV.CMD"); \#final
print "AndromedaXV_done.\n";
system("./MF_TRGB.e_AndromedaXVIe_0.0_59.0_30.1_32.0_22.0_32.9_0.27_0.92_104.0_0.005_0.2_0.2_1.7_4.7_ANDXVI.CMD"); \# final
print "AndromedaXVI_done.\n";

```

```

print "AndromedaXVII_done.\n";
system("./MF_TRGB.e_AndromedaXVIIIe_0.0_2.0_16.0_45.0_5.0_33.2_0.15_0.72_91.0_0.0_0_0.1_0.1_-8.5__6.0_ANDXVIII.CMD"); \# final
print "AndromedaXVIII_done.\n";
system("./MF_TRGB.e_AndromedaXIXe_0.0_19.0_32.1_35.0_2.0_37.1_0.17_6.2_37.0_0.0_0.2_0.2_-7.5_-2.0_ANDXIX.CMD");
print "AndromedaXIX_done.\n";

```

```

print "AndromedaXX done.\n";
system("./MF_TRGB.e_AndromedaXXIe_23.0_54.0_46.8_42.0_28.0_16.9_0.23_4.2_-36.0_0.0_0.3_0.3_-9.5_-6.5_ANDXXI.CMD"); \# final
print "AndromedaXXI_done.\n";

```

```

print "AndromedaXXII_done.\n";

```

```

print "AndromedaXXIII_done.\n";
system("./MF_TRGB.e_AndromedaXXIVe_1.0_18.0_31.4_46.0_22.0_19.3_0.0_2.3_-87.0_0.0_0.125_0.125_5.0_8.0_ANDXXIV.CMD"); \#final
print "AndromedaXXIV_done.\n";

```

```

print "AndromedaXXV_done.\n";
system("./MF_TRGB.e_AndromedaXXVIe_0.0_23.0_45.7_47.0_54.0_43.6_0.55_1.3_-31.0_0.0_0.15_0.15_-4.2_-2.2_ANDXXVI.CMD"); \#final
print "AndromedaXXVI_done.\n";
system("./MF_TRGB.e_AndromedaXXVIIe_0.0_37.0_36.4_45.0_22.0_19.0_0.75_15.8_-59.0_0.0_0.0_0_0.3_-2.5_0.5_ANDXXVII .CMD");\# final
print "AndromedaXXVII_done.\n";

```

```

print "AndromedaXXX_done.\n";
\#system("./MF_TRGB_NGC147e.e_NGC147e_outer_0.0_33.0_12.0_48.0_30.0_31.0_0.44_10.0_28.0_0.28_0.33_0.6_-4.2_-2.2_NGC147.CMD");
\#print "NGC147e_outer^done.\n";
\#system("./MF_TRGB_NGC147e.e_NGC147e_inner_0.0_33.0_12.0_48.0_30.0_31.0_0.44_10.0_28.0_0.12_0.18_0.6_-4.0_-2.1 _NGC147.CMD");
\#print "NGC147e_inner_done.\n";

```

```

\#print "NGC147stream_done.\n";
\#system("./MF_TRGB_NGC185e.e_NGC185e_outer_0.0_38.0_57.97_48.0_20.0_14.56_0.26_6.0_41.0_0.18_0.26_0.6_0.0_1.0_NGC185.CMD");
\#print "NGC185e_outer_done.\n";
\#system("./MF_TRGB_NGC205e.e_NGC205e_0.0_40.0_22.075_41.0_41.0_7.08_0.50_13.0_-35.0_0.38_0.4_0.4_-1.9_-0.9_NGC205.CMD");
\#print "NGC205e_done.\n";

```

```

\#print "M33_done.\n";

```

```

\#print "M31_done.\n";

```

```

\#print "M31e_NE_done.\n";

```

```

\#print "M31e_NW_done.\n";
\#system("./MF_TRGB_M31e_SE.e_M31e_SE_0.0_42.0_44.33_41.0_16.0_7.50_0.68_6000.0_37.0_2.45_2.5_2.5_9.0_10.0_M M 1 _ - ellipse .CMD");
\#print "M31e_SE_done.\n";

```

```

\#print "M31e_SW_done.\n";

```

\section*{Program: MF_TRGB_Tester.f95}

Creation Date: 8 December 2010
Relevant Section: §3.2 of Paper II (Ch. 4)
Notes: This program is the equivalent of 'MCMCTRGBTester2.f95' provided in Appendix C, but it has been updated for use with 'MF_TRGB.f95' and thus also provides the artificial stars with a radius representing their distance from the object's center. For simplicity, an ellipticity of 0 is assumed. For the sake of brevity, only the 'DataMaker' subroutine is shown, but the other subroutines called can be found in 'MF_TRGB.f95.'
```

MODULE Global ! Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, j, k, l, eval, idum = -9999, it, nit, trial
INTEGER :: ndata_max, nsamples, binspm, nbins, cmod_nbins, ghw, mm, ios
PARAMETER (ndata_max = 20000000, nsamples = 100)
PARAMETER (binspm = 100)
PARAMETER (nbins = 8*binspm + 1)
PARAMETER ( nit = 50000)
INTEGER :: ndata, ndata2
INTEGER :: d1, d2, d3, d4
REAL*8 :: blim, flim, pi
PARAMETER (blim = 19.5 d0)
PARAMETER (flim = 23.5d0)
PARAMETER (pi = ACOS(-1.e0))
INTEGER :: blimBins = INT(REAL((blim - 18.d0) * binspm)) + 1
INTEGER :: flimBins = INT(REAL((flim - 18.d0) * binspm)) + 1
REAL*8 :: randnum1, randnum2, randnum3, randnum4, randnum5, randnum6, randnum7
INTEGER :: randint
REAL*8 :: r1, r2, spotR, hb = 0.005 d0
REAL*8 :: model(nbins,2), cmodel(nbins,2), magnitude(ndata_max)
REAL*8 :: histo_fine(nbins,2), histo_coarse(INT(0.25*(nbins - 1.d0)) + 1,2)
REAL*8 :: w_histo_fine(nbins,2), w_histo_coarse(INT(0.25*(nbins - 1.d0)) + 1, 2)
REAL*8 :: data(ndata_max), cumulative_cmodel(nbins,2), f, f_hold, bfm(nbins)
REAL*8 :: cumulative_dist (2000,2)
REAL*8 :: mag_tip, mag, mag_cutoff = 24.e0, a
REAL*8 :: area, area2
REAL*8 :: modelnoise(nbins,2), noise(nbins) = 0.d0 , bg(nbins) = 0.d0
REAL*8 :: kernel(nbins,2) = 0.e0, scale, uplim, lowlim, gx
REAL*8 :: temp(nbins,2) = 0.e0, t
INTEGER :: starbin
REAL*8 :: tip(nsamples), tip_ord(nsamples), maxlogL(nsamples) = -999999999999
REAL*8 :: tip_rec, tip_offset, tip_psigma, tip_msigma, Toffset_kpc, Tsigma_kpc
REAL*8 :: f_offset, tip_kpc, kpc_perr, kpc_merr, f_sigma, a_offset, a_sigma
REAL*8 :: f_rec, a_rec, tip_counts, f_counts, a_counts
REAL*8 :: tipminsig, tiplusig, fminsig, fplusig, aminsig, aplusig
REAL*8 :: mcounts, pcounts
INTEGER :: num_chains, cn, chain_compare, swap_coun
PARAMETER (num_chains = 4)
REAL*8 :: swaprate = 1.d0/ 30.d0, logL(num_chains), LikeA(num_chains), LikeB(num_chains)
REAL*8 :: prob, sig_prob, bg_prob
REAL*8 :: beta, betaholder(num_chains) = (/ 1.d0, 0.25d0, 0.111d0, 0.001d0 /)
REAL*8 :: m_step(num_chains) = (/ 0.03d0, 0.06d0, 0.12d0, 0.3d0 /)
REAL*8 :: f_step(num_chains) = (/ 0.02d0, 0.04d0, 0.08d0, 0.2d0 /)

```
```

REAL*8 :: a_step(num_chains) = (/ 0.02d0, 0.04d0, 0.08d0, 0.2d0 /)
REAL*8 :: PTAR, par_hold(4)
REAL*8 :: x1(nit, num_chains), x2(nit,num_chains), x3(nit,num_chains), p(3), time(nit), r
REAL*8 :: post_y1(10*(nbins -1)+1) = 0.d0, post_x 1 (10*(nbins-1)+1), mlim
REAL*8 :: d_blim, bg_blim, d_flim, bg_flim
REAL*8 :: post_y2(nbins) = 0.d0, post_x 2(nbins)
REAL}*8 :: post_y 3 (2*nbins - 1) = 0.d0, post_x 3 (2*nbins - 1)
REAL*8 :: PPD_peak, Best_Combo(6)
CHARACTER :: argv * 10, field *60, ch 1*9, ch2*9, ch 3*9, ch4*9, ch5*9, ch6*9, string*90
!-___-__________-_-_-_For reading in PAndAS data_-___-_
INTEGER :: iCCDt, clsg, clsi, ifieldt, iacc_t
REAL*4 :: xgt, ygt, g, dg, im, dim, xki_t, eta_t, FeH_phot_t, diff_tip_t, E_BV_t
REAL*8 :: ra_t, de_t
REAL*4 :: mag_g(ndata_max), mag_i(ndata_max), xki(ndata_max), eta(ndata_max)
REAL*4 :: g_min_i(ndata_max), mag_i_poly(ndata_max), g_min_i_poly(ndata_max)
REAL*4 :: xi_poly(ndata_max), eta_poly(ndata_max)
REAL*4 :: gmi
!--_-- Additional parameters for calculating background stats -------
INTEGER :: bg_ndata, bg_ndata2, bg_ndata3
REAL*4 :: bg_mag_g(ndata_max), bg_mag_i(ndata_max), bg_xki(ndata_max), bg_eta(ndata_max)
REAL*4 :: bg_g_min_i(ndata_max), bg_mag_i_poly(ndata_max), bg_g_min_i_poly(ndata_max)
REAL*4 :: bg_gmi
REAL*8 :: bg_data(ndata_max)
!--SVD fitting of background--
INTEGER ma, mp, np, ndat
PARAMETER (ndat = INT (0.25*(nbins - 1.d0)) + 1)
PARAMETER ( np = 8)
PARAMETER (mp = ndat)
PARAMETER (ma = np)
REAL :: chisq, ay(ma), sig(ndat), u(mp,np), v(np,np), w(np), xa(ndat), ya(ndat)
REAL :: xt(ndat), yt(ndat)
REAL*8 :: bg_histo_coarse(ndat,2)
EXTERNAL :: funcs
!-----Additional parameters for specifying object coordinates ------
INTEGER :: Jop
REAL*8 :: XIop, ETAop
REAL*8 :: RAh, RAm, RAs, DecD, DecM, DecS, RA_rad, Dec_rad
REAL*8 :: tpRAh, tpRAm, tpRAs, tpDecD, tpDecM, tpDecS, tpRA_rad, tpDec_rad
!--Additional parameters for Matched Filters Subroutine 'Weighter'--
INTEGER :: rhobins, rhobins2
PARAMETER (rhobins = 10)
REAL*4 :: C_O_F_dist(ndata_max), Density(rhobins,2), rhofit(rhobins,2)
REAL*8 :: weight (ndata_max)
!--Fitting to Density Profile--
INTEGER :: mwt, ndat2
PARAMETER (ndat2 = rhobins)
REAL offset, gradient, chi2, q, siga, sigb, sigma(ndat2)
When f is known----------------------------------
INTEGER :: bg_stars, sig_stars
REAL*8 :: bg_area, sig_area, bg_density
REAL*8 :: known_f, bg_stars_in_sig_field
REAL}*8 :: sig_field_radius = 0.2d0, bg_low_xi = - 5.d0, bg_up_xi = 13.d

```
```

END MODULE Global
PROGRAM BayesianTRGBsatellite !Master program
USE Global
IMPLICIT NONE
mm = IARGC()
IF (mm==4) THEN
CALL GETARG(1, argv)
READ (argv,*,iostat=ios) mag_tip
CALL GETARG(2, argv)
READ (argv,*,iostat=ios) a
CALL GETARG(3, argv)
READ (argv,*,iostat=ios) ndata !Indicates the arguments to be
CALL GETARG(4, argv) !set in the command line
READ (argv,*,iostat=ios) f
ELSE
WRITE(*,*) "You^must\_enter_4\_arguments:"
stop ;
END IF
WRITE (ch1,*) mag_tip
WRITE (ch2,*) a
WRITE (ch3,*) ndata
IF (f .eq. 0.d0) THEN !Generate test identifying character string
WRITE (ch4,*) ,0, !to become file name using mag_tip, ndata and f
ELSE !e.g. 'MCMC_Test/ T_20.5-0.3-1000-0.2'
WRITE (ch4,*) f !
END IF
ndata2 = 0
WRITE (field,*) 'MF_MCMC_Test/T_' // TRIM(ADJUSTL(ch1)) \&
// ,_, // TRIM(ADJUSTL(ch2)) \&
// ,_, // TRIM(ADJUSTL(ch3)) \&
// '-' // TRIM(ADJUSTL(ch4))
string = TRIM(ADJUSTL(field)) // '/test.dat'
OPEN(3, file=TRIM(ADJUSTL(string)), status = 'unknown')
WRITE (3,*) "Field_Name:", field
WRITE (3,*) "ndata_=_", TRIM(ADJUSTL(ch3))
WRITE (3,*) " f_=_", TRIM(ADJUSTL(ch4))
CALL random_seed !
CALL NoiseMake
CALL ModelMake
CALL Convolution !
cmodel (:,2) = (1.d0 - f) * cmodel (:,2) + f * noise
CALL DataMaker
CALL Weighter
CALL NoiseMake !CALL
CALL MCMC
CALL TipAndSigma !SUBROUTINES
CALL PosteriorPlot !

```
```

CALL OtherPlots !
CALL DataHist !
CALL w_DataHist !
IF (num_chains .ne. 1) THEN
WRITE (3,*) "Proposed_Swaps_with_Cold_Sampler_Chain:", chain_compare
WRITE (3,*) "Accepted_Swaps_with_Cold_Sampler_Chain:", swap_count
WRITE (3,*) "Parallel_Tempering_Acceptance\lrcornerRate:", REAL(swap_count)/ REAL(chain_compare)
END IF
WRITE (3, '(3a11)') "__-_tip_mag:", "__+_sigma:__", "__--_sigma:_"
WRITE (3, '(3F10.3)') tip_rec, tip_psigma, tip_msigma!
WRITE (3, '(2a11)') "__-_f:-_---_", "-_-_sigma:__">
WRITE (3, '(2F10.3)') f_rec, f_sigma !Write results

```

```

WRITE (3, '(2F10.3)') a_rec, a_sigma !
WRITE (3,*) "Distance_=", REAL(tip_kpc), "kpc"
WRITE (3,*) "Error_=`+", REAL(kpc_perr), "kpc_-", REAL(kpc_merr), "kpc"
WRITE (3,*) "Average_Error_=", REAL((ABS(kpc_perr) + ABS(kpc_merr))/2.d0), "kpc"
WRITE (3,*) "Tip_Mag_\&_Error_=", tip_rec, REAL(tip_psigma), REAL(tip_msigma)
WRITE (3,*) "Offset_=", REAL(tip_rec - 20.5d0), "=", REAL(tip_kpc - (100.d0**((20.5d0 + 3.44d0)/10.d0))/100.d0),"kpc"
END PROGRAM BayesianTRGBsatellite
SUBROUTINE DataMaker !Generates artificial stars with magnitudes from
USE Global !model luminosity function and positions from
IMPLICIT NONE !model density profile
cumulative_cmodel(:,1) = cmodel(:, 1)
cumulative_cmodel(1,2) = cmodel(1,2) !Effective
DO i = 2, cmod_nbins !integral of
cumulative_cmodel(i,2) = cumulative_cmodel(i-1,2) + cmodel(i, 2)!convolved
END DO !model
DO i = 1, ndata
CALL random_number(randnum6)
randnum6 = cumulative_cmodel(blimBins,2) + \&
randnum6 * (cumulative_cmodel(flimBins,2) - cumulative_cmodel(blimBins,2))
DO j = flimBins, blimBins, -1 !Generates 'ndata'
IF (randnum6 .le. cumulative_cmodel(j,2)) THEN !magnitude datapoints
IF (randnum6 .gt. cumulative_cmodel(j-1,2)) THEN !from the convolved
data(i) = cumulative_cmodel(j-1,1) !model
exit ;
END IF
END IF
END DO
END DO
cumulative_dist = 0.d0 !
cumulative_dist (1,1) = 0.0001 !Generates model radial density
cumulative_dist (1,2) = 10.d0 ** (5.610696-0.0013362497) !profile based on that fitted
DO i = 2, 2000 !of zero ellipticity)
cumulative_dist(i,1)= i * 0.0001d0
cumulative_dist(i,2) = cumulative_dist (i-1,2) + (10.d0 ** (5.610696 - 13.362497 * cumulative_dist(i, 1)))
END DO

```
```

DO i = 1, ndata
CALL random_number(randnum6)
IF (data(i) .ge. mag_tip) THEN
CALL random_number(randnum7) !Draws random
IF (randnum7 .gt. f * (flim - mag_tip)/(flim - blim)) THEN
randnum6 = randnum6 * cumulative_dist (2000,2)
DO j = 2000, 2, -1
IF (randnum6 . le. cumulative_dist(j, 2) .and. randnum6 .gt. cumulative_dist(j - 1, 2)) THEN
C_O_F_dist(i) = cumulative_dist(j - 1, 1)
exit; !for each
END IF !
END DO !star based
ELSE
C_O_F_dist(i) = SQRT((randnum6 * (pi * sig_field_radius ** 2))/ pi)!on above
END IF
ELSE
C_O_F_dist(i) = SQRT((randnum6 * (pi * sig_field_radius ** 2))/ pi) !
END IF
END DO
sig_area = pi * (sig_field_radius ** 2.d0) !
sig_stars = ndata !For calculating ratio
ndata2 = ndata !of RGB to background
bg_density = f * REAL(sig_stars)/ sig_area !
plot magnitude vs. radius --------------------
string = TRIM(ADJUSTL(field)) // '/mag_vs_rad.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string)),1,1)
CALL pgenv(19.5, 23.5, 0., MAXVAL(REAL(C_O_F_dist)), 0, 0)
CALL pgslw(3)
CALL pgpt (ndata, REAL(data), REAL(C_O_F_dist), -1)
CALL pgslw(1)
CALL pglab('magnitude', 'radius', ',')
CALL pgend
END SUBROUTINE DataMaker

```

Program: Multi_MCMC_Result_Plotter.f95
Creation Date: 6 Feb 2012 (first version 10 Dec 2010 )
Relevant Sections: Ch. 3 \& Ch. 4
Notes: This program was created to take the posterior distributions generated by the TRGB algorithm (e.g. 'MF_TRGB.f95') and produce more polished versions of the figures for use in papers I and II. In particular, it colour-codes the distributions to indicate the \(1 \sigma, 90 \%\) and \(99 \%\) credibility intervals. It also generates a contour map of the distribution of the tip magnitude verses the RGB slope model parameters (see Fig. 8 of Paper I; Ch. 3) - i.e. a 3D surface from which the individual parameter posterior distributions are created by marginalizing over the other parameter. The actual distance posterior distributions for each object are also created by this program. This is achieved by sampling the posterior distribution in the tip magnitude along with the probability distributions for the absolute magnitude of the tip and the extinction along the line of sight (see the 'Dist_Error' subroutine). The halo density prior (see \(\S 3.3\) of Paper II; Ch. 4) is also generated and applied in this program, as are the hundredth-percentile tables of the object distance distributions published alongside Paper II (see Table 1 of Paper II for example).
```

MODULE Global ! Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, ios, j, k, ndata_max, ndata, ndata_M31, nbins, binspm, d1, d2, d3, d4, mm
PARAMETER (ndata_max = 7100000, binspm = 100)
PARAMETER (nbins = 8 * binspm)
REAL*8 :: pi
PARAMETER (pi = ACOS (-1.e0))
REAL :: it(ndata_max), mag_tip(ndata_max), f(ndata_max), a(ndata_max)
REAL :: LikeA(ndata_max), LikeB(ndata_max), M31_dist_ppd(ndata_max)
REAL :: tip_PPD (10*(nbins -1)+1, 2) = 0.d0
REAL :: f_PPD(nbins, 2) = 0.d0, a_PPD(2*nbins - 1, 2) = 0.d0
REAL :: mag_tip_ll, mag_tip_ul
REAL :: f_ll = 0., f_ul = 1.
REAL :: a_11, a_ul
REAL :: tip_rec, f_rec, a_rec, PPD_peak, tip_kpc, tip_counts, mcounts, pcounts, tip_msigma, tip_psigma
REAL :: f_sigma, a_sigma, kpc_merr, kpc_perr, f_counts, a_counts, fminsig, fplusig, aminsig, aplusig
REAL :: tip_m90, tip_p90, tip_m99, tip_p99, xpts(2), ypts (2)
REAL :: xi_coord, eta_coord
REAL*8 :: RA, DEC, xi_dble, eta_dble
CHARACTER :: argv*20, field*60, plot_dir *60, string*200, string2*200, command 1*200, command 2*300
INTEGER :: cont_bins, ncontours
PARAMETER (cont_bins = 75)
PARAMETER (ncontours = 20)
REAL :: Cont(cont_bins, cont_bins), clevels(ncontours), TR(6)
!-_____-_-_For Distance Distribution -_-_-_-_-_-_-_

```
```

INTEGER :: idum = -9999, nsamples, DistBins, prior-type
PARAMETER (nsamples = 500000)
REAL :: M_TRGB, Ext, Ext_0, Tip, Dist_PPD, Dist_PPDx(4000), Dist_PPDy(4000), Dist_PPD_min, Dist_PPD_max
REAL :: dist_rec, dist_counts, dist_msigma, dist_psigma, dist_m90, dist_p90, dist_m99, dist_p99
REAL :: DistPrior(4000), alpha, slope, flat, hwhm, theta
REAL :: M31_to_obj-x (4001), M31_to_obj-y(4001), M31_to_obj, m31_dist
REAL :: M31_dist_rec, M31_dist_psigma, M31_dist_msigma
REAL*8 :: randnum
!---For converting distances back to magnitudes ---
REAL :: dist2mag_rec, d2m_msigma, d2m_psigma, d2m_m90, d2m_p90, d2m_m99, d2m_p99
!-_-_-_-_For Hundredth Percentile Table________
REAL :: cum_dist, perc, dist_at_perc(100,2)
LOGICAL next
END MODULE Global
PROGRAM MCMC_Result_Plotter
USE Global
IMPLICIT NONE
DOUBLE PRECISION :: sla_DSEP
mm = IARGC()
IF (mm==9) THEN
CALL GETARG(1, argv)
READ (argv,*,iostat=ios) field
CALL GETARG(2, argv)
READ (argv,*,iostat=ios) Ext_0
CALL GETARG(3, argv )
READ (argv,*,iostat=ios) mag_tip_1l
CALL GETARG(4, argv)
READ (argv,*,iostat=ios) mag_tip_ul !Indicates the arguments to be
CALL GETARG(5, argv) !set in the command line
READ (argv,*,iostat=ios) a_ll
CALL GETARG(6, argv)
READ (argv,*,iostat=ios) a_ul
CALL GETARG(7, argv)
READ (argv,*,iostat=ios) xi_coord
CALL GETARG(8, argv)
READ (argv,*,iostat=ios) eta_coord
CALL GETARG(9, argv)
READ (argv,*,iostat=ios) plot_dir
ELSE
WRITE(*,*) "You_must_enter_9_arguments:"
stop ;
END IF
xi_coord = xi_coord * (pi/180.e0) !Convert angles from
eta_coord = eta_coord * (pi/180.e0) !degrees to radians
xi_dble = xi_coord ; eta_dble = eta_coord
CALL sla_DTP2S(xi_dble, eta_dble, 0.d0, 0.d0, RA, DEC) !Convert tangent plane
IF (xi_dble .lt. 0.d0) then !projection angles into
RA = RA - (2.e0 * pi) !their true angles using
END IF !sla_DTP2S

```
```

    xi_coord = RA
    eta_coord = DEC
    xi_dble = xi_coord
    eta_dble = eta_coord
    theta = sla_DSEP(0.d0, 0.d0, xi_dble, eta_dble)
                !
                !the sky between M31
                                    !and the object
                                    !(uses sla_DSEP)
    xi_coord = xi_coord * (180.e0/pi) !Convert back
    eta_coord = eta_coord * (180.e0/pi) !to degrees
    WRITE (*,*) xi_coord, eta_coord, theta * (180.e0/pi)
    WRITE (string,*) ,./, // TRIM(ADJUSTL(field)) // ,/, // TRIM(ADJUSTL(plot_dir))
WRITE (commandl,*) 'mkdir_' // TRIM(ADJUSTL(string))
call system(command1)
OPEN (unit = 1, file = './' // TRIM(ADJUSTL(field)) // '/MCMC_steps.dat', status = 'old') !Open input
OPEN (unit = 2, file = './' // TRIM(ADJUSTL(string)) // '/results.dat', status = 'unknown')!and output
OPEN (unit = 3, file = '/M31e/other_plots/M31_Distance_PPD.dat', status = 'old') !files
WRITE (2,*) "Field:\iota", TRIM(ADJUSTL(field))
WRITE (2,*) "Coordinates:_xi_=", xi_coord, ",_eta_=", eta_coord !
WRITE (2,*) "Plot_Directory:\lrcorner", TRIM(ADJUSTL(string)) ! Print basic object
WRITE (2,*) "„" !info to file
WRITE (2,*) "Extinction_in_SDSS_i:_", Ext_0
WRITE (2,*) "E(B-V) : „", Ext_0 / 2.086e0
i = 0 ; ios = 0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
i=i+1
READ (1, *, IOSTAT = ios) it(i), mag_tip(i), f(i), a(i), LikeA(i), LikeB(i)
if (ios == 0) then ;
else if (ios == -1) then ;
i=i-1
exit ;
else if (ios > 0) then ;
i=i - 1
cycle
end if
END DO
ndata = i - 1
i = 0 ; ios = 0
DO WHILE (.TRUE.) !Reads M31 distance sample data until end of input file and puts it into an array
i=i+1
READ (3, *, IOSTAT = ios) M31_dist_ppd(i)
if (ios == 0) then ;
else if (ios == -1) then ;
i=i - 1
exit ;
else if (ios > 0) then ;
i=i-1
cycle
end if
END DO
ndata_M31 = i - 1

```
```

CALL random_seed !
CALL PosteriorBuild !CALL
CALL PosteriorPlot
CALL OtherPlots !SUBROUTINES
CALL Dist_Error !
END PROGRAM MCMC_Result_Plotter
SUBROUTINE PosteriorBuild
USE Global
IMPLICIT NONE
DO i = 1, 10*(nbins -1)+1
tip_PPD (i, 1) = 18.d0 + (REAL(i) - 1.d0)/REAL(10*binspm) !
END DO
DO i = 1, nbins !x-values of
f_PPD (i, 1)=(REAL(i) - 1.d0)/REAL(nbins - 1) !PPD histograms
END DO
DO i = 1, 2* nbins - 1
a_PPD (i, 1) = (REAL(i) - 1.d0)/REAL(nbins - 1) !
END DO
DO i = 1, ndata
tip_PPD (INT ((mag_tip (i) - 18.d0)*10*binspm + 1), 2) = \&
tip_PPD(INT((mag_tip (i) - 18.d0)*10*binspm + 1), 2) + 1.d0 !
f_PPD(INT(f(i) * (nbins - 1)) + 1, 2) = \& !y-values of
f_PPD(INT(f(i) * (nbins - 1)) + 1, 2) + 1.d0 !PPD histograms
a_PPD(INT(a(i) * (nbins - 1)) + 1, 2) = \&
a_PPD(INT(a(i) * (nbins - 1)) + 1, 2) + 1.d0
END DO
CALL Confidence
END SUBROUTINE PosteriorBuild
SUBROUTINE PosteriorPlot ! Plots posterior distributions in tip magnitude and a
USE Global !tip magnitude PPD is plotted with credibility intervals
IMPLICIT NONE
tip_PPD(:,2) = tip_PPD(:,2)/ndata ; f_PPD(:,2) = f_PPD(:,2)/ndata
a_PPD (:,2) = a_PPD (:,2)/ndata

```

```

string2 = TRIM(ADJUSTL(string)) // '/bw_mag_tip_postplot.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(mag_tip_ll, mag_tip_ul, 0., 1.1*MAXVAL(tip_PPD (:,2)), 0, 0)
CALL pgbin (10*(nbins-1)+1, tip_PPD (:, 1), tip_PPD (:, 2),.false.)
CALL pglab('Proposed_i\d0\u_tip magnitude,, 'Probability,, ',')
CALL pgend
WRITE (command2,*) , convertu-rotate \& 90_, // TRIM(ADJUSTL(string)) // \&
'/bw_mag_tip_postplot.ps_' // TRIM(ADJUSTL(string)) // \&
'/bw_mag_tip_postplot.jpg,

```
```

call system(command2)

```

```

string2 = TRIM(ADJUSTL(string)) // ,/mag_tip_postplot.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(mag_tip_11, mag_tip_ul, 0., 1.1*MAXVAL(tip_PPD (:,2)), 0, 0)
DO i = 1, 10*(nbins - 1)+1
IF (tip_PPD(i,1) .ge. tip_rec - tip_msigma . and. tip_PPD(i,1).lt. tip_rec + tip_psigma) THEN
CALL pgsci(2)
CALL pgbin (2, tip_PPD(i,1), tip_PPD(i, 2),.false.)
IF (tip_PPD(i,1) .eq. tip_rec - tip_msigma) THEN
xpts = tip_PPD(i,1)
ypts(1) = 0.e0 ; ypts(2) = tip_PPD(i,2) !One Sigma
CALL pgline (2, xpts, ypts)
IF (tip_PPD(i+1,1) .eq. tip_rec + tip_psigma) THEN
xpts = tip_PPD(i+1,1) !Interval
ypts(1) = 0.e0 ; ypts(2) = tip_PPD(i,2) !
CALL pgline (2, xpts, ypts) !
END IF
ELSE IF (tip_PPD(i,1) .ge. tip_rec - tip_m90 .and. tip_PPD(i,1).lt. tip_rec + tip_p90) THEN
CALL pgsci (3)
CALL pgbin (2, tip_PPD(i,1), tip_PPD(i,2),.false.)
IF (tip_PPD(i,1) .eq. tip_rec - tip_m90) THEN !
xpts = tip_PPD(i,1) !
ypts(1) = 0.e0 ; ypts(2) = tip_PPD(i,2) !90 percent
CALL pgline (2, xpts, ypts) !
END IF !Credibility
IF (tip_PPD(i+1,1) .eq. tip_rec + tip_p90) THEN
xpts = tip_PPD (i+1,1)
!Interval
ypts(1) = 0.e0 ; ypts(2) = tip_PPD(i,2)
CALL pgline (2, xpts, ypts) !
!
END IF
ELSE IF (tip_PPD(i,1) .ge. tip_rec - tip_m99 . and. tip_PPD(i,1).lt. tip_rec + tip_p99) THEN
CALL pgsci(4)
CALL pgbin (2, tip_PPD(i,1), tip_PPD(i,2),.false.) !
IF (tip_PPD(i,1) .eq. tip_rec - tip_m99) THEN
xpts = tip_PPD(i,1)
ypts(1)=0.e0; ypts(2)=tip_PPD(i,2) !99 percent
CALL pgline (2, xpts, ypts) !
END IF !Credibility
IF (tip_PPD(i+1,1) .eq. tip_rec + tip_p99) THEN
xpts = tip_PPD(i+1,1)
!
ypts(1)=0.e0 ; ypts(2)=tip_PPD(i,2)
!Interval
!
CALL pgline (2, xpts, ypts)
!
END IF
ELSE
CALL pgsci(1) !Distribution
CALL pgbin (2, tip_PPD(i,1), tip_PPD(i,2),.false.) !outside of 99%
END IF !Cred. Interval
END DO
CALL pgsci(1)
CALL pglab('Proposed,i \d0\u_tip mmagnitude', 'Probability',',')
CALL pgend

```
```

WRITE (command2,*) 'convert_-rotate_90_' // TRIM(ADJUSTL(string)) // \&
'/mag_tip_postplot.ps_' // TRIM(ADJUSTL(string)) // \&
'/mag_tip_postplot.jpg,
call system(command2)

```

```

string2 = TRIM(ADJUSTL(string)) // ,/a_postplot.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(a_11, a_ul, 0., 1.1*MAXVAL(a_PPD (:,2)), 0, 0)
CALL pgbin (2*nbins - 1, a_PPD(:,1), a_PPD(:,2),.false.)
CALL pglab('Proposed_value_for_LF_slope_(a)', 'Probability', ,',)
CALL pgend
WRITE (command2,*) 'convert_-rotate_90_' // TRIM(ADJUSTL(string)) // \&
,/a_postplot.ps_, // TRIM(ADJUSTL(string)) // \&
,/a_postplot.jpg,
call system(command2)
tip_PPD(:,2) = tip_PPD (:,2)*ndata ; f_PPD (:,2) = f_PPD (:, 2)* ndata
a_PPD (:,2) = a_PPD (:,2) * ndata
END SUBROUTINE PosteriorPlot
SUBROUTINE OtherPlots
USE Global
IMPLICIT NONE
!--_-_-----Values of 'a' for each value of 'mag_tip' - contour plot
Cont = 0.e0
TR = 0.e0
TR(1) = mag_tip_ll ; TR(2) = (mag_tip_ul - mag_tip_ll)/REAL(cont_bins); TR(4) = a_ll ; TR(6) = (a_ul - a_ll)/REAL(cont_bins)
DO k = 1, ndata
i = INT((mag_tip(k) - TR(1))/TR(2)) + 1
j = INT((a(k) - TR(4))/TR(6)) + 1
if ( i>0 .and. i<=cont_bins .and. j>0 .and. j<=cont_bins) Cont(i,j) = Cont(i,j) + 1.e0
END DO
DO i = 1, ncontours
clevels(i) = 0.e0 + i*MAXVAL( Cont)/REAL(ncontours)
END DO
string2 = TRIM(ADJUSTL(string)) // '/m_vs_a_contour.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(mag_tip_ll, mag_tip_ul, a_1l, a_ul, 0, 0)
CALL PGCONT (Cont, cont_bins, cont_bins, 1, cont_bins, 1, cont_bins, clevels, ncontours, TR)
CALL pglab('Proposed_i \d0\u_tip \&magnitude', 'Proposed_value _of La',,',')
CALL pgend
WRITE (command 2,*) 'convert_-rotate \& 90^' // TRIM(ADJUSTL(string)) // \&
'/m_vs_a_contour.ps_' // TRIM(ADJUSTL(string)) // \&
,/m_vs_a_contour.jpg,

```
```

call system(command2)
END SUBROUTINE OtherPlots
SUBROUTINE Confidence !Identifies the best parameter values and
USE Global !their associated 1 sigma errors from the
IMPLICIT NONE !respective posterior plots.
PPD_peak = 0.d0
DO i = 1, 10*(nbins - 1)+1
IF (tip_PPD(i,2) .gt. PPD_peak) THEN !
PPD_peak = tip_PPD(i,2) !Find best fit TRGB value
tip_rec = tip_PPD(i,1) !
END IF
END DO
PPD_peak = 0.d0
DO i = 1, nbins
IF (f_PPD(i,2) .gt. PPD_peak) THEN
PPD_peak = f_PPD(i,2) !Find best fit f value
f_rec = f_PPD(i,1) !
END IF
END DO
PPD_peak = 0.d0
DO i = 1, 2*nbins - 1
IF (a_PPD(i,2) .gt. PPD_peak) THEN
PPD_peak = a_PPD(i,2) !Find best fit a value
a_rec = a_PPD (i,1)
END IF
END DO
tip_kpc = (100.d0**((tip_rec + 3.44d0)/10.d0))/100.d0 !Distance inferred from
!tip magnitude in kpc
tip_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 1, -1
mcounts = mcounts + tip_PPD(i,2)
END DO
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 1, -1
tip_counts = tip_counts + tip_PPD(i,2) !Finds negative one sigma
IF (tip_counts .ge. 0.682*mcounts) THEN !error in magnitudes
tip_msigma = ((REAL(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
tip_msigma = tip_rec - tip_msigma
exit
END IF
END DO
tip_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 10*(nbins - 1)+1 !
pcounts = pcounts + tip_PPD(i,2)
END DO
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 10*(nbins -1)+1
tip_counts = tip_counts + tip_PPD(i,2) !Finds positive one sigma
IF (tip_counts .ge. 0.682*pcounts) THEN !error in magnitudes
tip_psigma = ((REAL(i) - 1.d0)/REAL(10*binspm)) + 18.d0 !
tip_psigma = tip_psigma - tip_rec
exit

```
```

    END IF
    END DO
tip_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 1, -1
mcounts = mcounts + tip_PPD(i,2)
END DO
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 1, -1
tip_counts = tip_counts + tip_PPD(i, 2)
IF (tip_counts .ge. 0.9 d0*mcounts) THEN
tip_m90 = ((\boldsymbol{REAL}(i) - 1.d0)/REAL(10*binspm)) + 18.d0
tip_m90 = tip_rec - tip_m90
exit
END IF
END DO
tip_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 10*(nbins -1)+1
pcounts = pcounts + tip_PPD(i,2)
END DO
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 10*(nbins - 1)+1
tip_counts = tip_counts + tip_PPD(i,2)
IF (tip_counts .ge. 0.9d0*pcounts) THEN
tip_p90 = ((\boldsymbol{REAL}(i) - 1.d0)/REAL(10*binspm)) + 18.d0
tip_p90 = tip_p90 - tip_rec
exit
END IF
END DO
tip_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 1, -1
mcounts = mcounts + tip_PPD (i,2)
END DO
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 1, -1
tip_counts = tip_counts + tip_PPD(i,2)
IF (tip_counts .ge. 0.99d0*mcounts) THEN
tip_m99 = ((\boldsymbol{REAL}(i) - 1.d0)/REAL}(10*\mathrm{ binspm ) ) + 18.d0
tip_m99 = tip_rec - tip_m99
exit
END IF
END DO
tip_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 10*(nbins - 1)+1
pcounts = pcounts + tip_PPD(i,2)
END DO
DO i = MAXLOC(tip_PPD (:,2), DIM = 1), 10*(nbins - 1)+1
tip_counts = tip_counts + tip_PPD(i,2)
IF (tip_counts .ge. 0.99d0*pcounts) THEN
tip_p99 = ((\boldsymbol{REAL}(i) - 1.d0)/REAL(10*binspm)) + 18.d0
tip_p99 = tip_p99 - tip_rec
exit
END IF
END DO
d1 = 0 ; d2 = 0; d3 = 0 ; d4 = 0
f_counts = 0.d0 ; a_counts = 0.d0
DO i = 1, nbins
f_counts = f_counts + f_PPD(i, 2)
a_counts = a_counts + a_PPD(i,2)
IF (f_counts .ge. 0.159*ndata . and. d1 .eq. 0) THEN!

```
```

        fminsig = f_PPD(i, 1)
        d1 = 1
    END IF
    IF (f_counts .ge. 0.841*ndata . and. d2 .eq. 0) THEN! For f and a:
        fplusig = f_PPD(i,1) !Finds upper and lower
        d2 = 1 !bounds for posterior
    END IF ! distribution within one
    IF (a_counts .ge. 0.159*ndata .and. d3 .eq. 0) THEN!sigma of maximum.
        aminsig = a_PPD(i, 1)
        d3 = 1
    END IF
    IF (a_counts .ge. 0.841*ndata .and. d4 .eq. 0) THEN
        aplusig = a_PPD(i, 1)
        d4 = 1
    END IF
    END DO
f_sigma = 0.5d0*(fplusig - fminsig) !Hence calculates 1 sigma error
a_sigma = 0.5d0*(aplusig - aminsig) !for f and a
kpc_merr = tip_kpc*100.d0**(tip_msigma/10.d0) - tip_kpc !minus tip error in kpc
kpc_perr = tip_kpc*100.d0**(tip_psigma/10.d0) - tip_kpc !plus tip error in kpc
WRITE (2,*) "৩"
WRITE (2,*) "Distance_Modulus:", tip_rec + 3.44e0
WRITE (2,*) "+sigma_-sigma:", tip_psigma, tip_msigma
WRITE (2,*) "+90_- 90:", tip_p90, tip_m90
WRITE (2,*) "+99_-99:", tip_p99, tip_m99
WRITE (2,*) "tip_+sigma_-sigma:", tip_rec, tip_rec + tip_psigma, tip_rec - tip_msigma
WRITE (2,*) "tip_+90_-90:", tip_rec, tip_rec + tip_p90, tip_rec - tip_m90
WRITE (2,*) "tip_+99_-99:", tip_rec, tip_rec + tip_p99, tip_rec - tip_m99
END SUBROUTINE Confidence

```
```

SUBROUTINE Dist_Error !Samples Distance likelihood space

```
SUBROUTINE Dist_Error !Samples Distance likelihood space
USE Global !using samples of m_TRGB, A_lambda and M_TRGB
USE Global !using samples of m_TRGB, A_lambda and M_TRGB
IMPLICIT NONE !from their respective likelihood distributions
IMPLICIT NONE !from their respective likelihood distributions
REAL*8 :: gasdev
REAL*8 :: gasdev
!||Don't forget to reinstate
!||Don't forget to reinstate
!\/writing distances to files 13 & 14
!\/writing distances to files 13 & 14
string2 = TRIM(ADJUSTL(string)) // '/Sampled_MWy_Distances.dat
string2 = TRIM(ADJUSTL(string)) // '/Sampled_MWy_Distances.dat
OPEN (unit = 13, file = TRIM(ADJUSTL(string2)), status = 'unknown')
OPEN (unit = 13, file = TRIM(ADJUSTL(string2)), status = 'unknown')
string2 = TRIM(ADJUSTL(string)) // '/Sampled_M31_Distances.dat'
string2 = TRIM(ADJUSTL(string)) // '/Sampled_M31_Distances.dat'
OPEN (unit = 14, file = TRIM(ADJUSTL(string2)), status = 'unknown')
OPEN (unit = 14, file = TRIM(ADJUSTL(string2)), status = 'unknown')
Dist_PPDx = 0.e0 !Pre-set Distance likelihood
Dist_PPDx = 0.e0 !Pre-set Distance likelihood
Dist_PPDy = 0.e0 !distribution histogram to 0.
Dist_PPDy = 0.e0 !distribution histogram to 0.
Dist_PPD_min = (100.e0**((MINVAL(mag_tip, mask = mag_tip .ne. 0.) - 0.3e0*Ext_0 + 3.14)/10.e0))/100.e0
Dist_PPD_min = (100.e0**((MINVAL(mag_tip, mask = mag_tip .ne. 0.) - 0.3e0*Ext_0 + 3.14)/10.e0))/100.e0
Dist_PPD_max = (100.e0**((MAXVAL(mag_tip, mask = mag_tip .ne. 0.) + 0.3e0*Ext_0 + 3.74)/10.e0))/100.e0
Dist_PPD_max = (100.e0**((MAXVAL(mag_tip, mask = mag_tip .ne. 0.) + 0.3e0*Ext_0 + 3.74)/10.e0))/100.e0
M31_to_obj_x = 0.e0 !Pre-set M31 to object histogram x values to 0.
M31_to_obj_x = 0.e0 !Pre-set M31 to object histogram x values to 0.
M31_to_obj-y = 0.e0 !Pre-set M31 to object distance histogram values to 0.
M31_to_obj-y = 0.e0 !Pre-set M31 to object distance histogram values to 0.
DistBins = 0
DO i = NINT(Dist_PPD_min) - 1, NINT(Dist_PPD_max ) + 1
    DistBins = DistBins + 1
    Dist_PPDx(DistBins) = REAL(i)
                                    !Generate 'x' values (MWy distances)
                                    !for distribution histogram and count
                                    !number of bins
```

```
END DO
DO i = -2000, 2000
    M31_to_obj_x (i+2001) = REAL(i)
END DO
DO i = 1, nsamples
    M_TRGB = 3.44e0 + 0.05 e0 *gasdev (idum)
    Ext = Ext_0 + 0.1e0*Ext_0*gasdev(idum)
    CALL random_number(randnum)
    Tip = mag_tip (NINT(randnum *0.9999d0*ndata) +1) + Ext_0
    Dist_PPD = (100.e0**((Tip - Ext + M_TRGB)/10.e0))/100.e0
    m31_dist = M31_dist_ppd(NINT(randnum*0.9999d0*ndata_M31)+1)
    M31_to_Obj = ((Dist_PPD ** 2.e0) + (m31_dist ** 2.e0) - &
                2.e0* Dist_PPD * m31_dist * cos(theta)) ** 0.5e0 !
    WRITE (13,*) Dist_PPD
    WRITE (14,*) M31_to_Obj
    Dist_PPDy(NINT(Dist_PPD) - (NINT(Dist_PPD_min) - 2)) = & !Tally up number of counts
    Dist_PPDy(NINT(Dist_PPD) - (NINT(Dist_PPD_min) - 2)) + 1.e0 !in each Earth distance bin
    M31_to_obj-y(2001 + NINT(M31_to_Obj)) = & !Tally up number of counts
    M31_to_obj_y(2001 + NINT(M31_to_Obj)) + 1.e0 !in each M31 distance bin
END DO
!-___-_-_-_One Hundredth Percentiles before prior -_-_-_
dist_at_perc = 0.e0
cum_dist = 0.e0 ; perc = 0.e0 ; next = .true.
DO i = 1, 4000
    cum_dist = cum_dist + Dist_PPDy(i) !
1 IF (next) THEN
            perc = perc + 1.e0
            next = .false.
        END IF
        IF (cum_dist .ge. (perc/100.e0)*SUM(Dist_PPDy)) THEN
            dist_at_perc(NINT(perc),1) = Dist_PPDx(i)
            next = .true.
        goto 1
    END IF
END DO
!-----------------------------
CALL DistancePrior
Dist_PPDy = Dist_PPDy * DistPrior
Dist_PPDy = Dist_PPDy/ SUM(Dist_PPDy)
!----------------
cum_dist = 0.e0 ; perc = 0.e0 ; next = .true.
DO i = 1, 4000
    cum_dist = cum_dist + Dist_PPDy(i) !
2 IF (next) THEN 
            perc = perc + 1.e0
            next = .false. !data. i.e. - cum_dist does not progress until
                                    !Note, this routine now accounts for the fact that
                                    !a single bin can contain more than 1% of the
    END IF !the percentage of the PPD surpasses it. Otherwise
    IF (cum_dist .ge. (perc/100.e0)*SUM(Dist_PPDy)) THEN
                                    !
```

! Note, this routine now accounts for the fact that !a single bin can contain more than $1 \%$ of the !data. i.e. - cum_dist does not progress until ! the percentage of the PPD surpasses it. Otherwise ! cum_dist overtakes it and the second if statement !is always true.

```
2 IF (next) THEN !No
!cum_dist overtakes it and the second if statement
```

```
    dist_at_perc(NINT(perc),2) = Dist_PPDx(i) !is always true
    next = .true. !
        goto 2
    END IF
END DO
!-_-_---_Create table of One Hundredth Percentiles -------
string2 = TRIM(ADJUSTL(string)) // '/Hundredth_Percentiles.dat
OPEN (unit = 14, file = TRIM(ADJUSTL(string2)), status ='unknown')
DO i = 1, 100
    WRITE(14,'(3i7)') i, NINT(dist_at_perc(i,1)), NINT(dist_at_perc(i, 2))
END DO
CALL Confidence2 !Calculate 68.3%, 90% and 99% plus/ minus credibility intervals
Dist_PPDy = Dist_PPDy/ SUM(Dist_PPDy) !normalize distribution
CALL Confidence3 ! Calculate 68.3%credibility intervals
M31_to_obj_y = M31_to_obj_y/ SUM(M31_to_obj_y) !normalize distribution
```



```
string2 = TRIM(ADJUSTL(string)) // '/bw_dist_PPD.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(MINVAL(DIST_PPDx, mask = DIST_PPDx . ne. 0.) - 1, MAXVAL(DIST_PPDx ) + 1, 0., 1.1*MAXVAL(Dist_PPDy ), 0, 0)
CALL pgbin (DistBins, Dist_PPDx, Dist_PPDy, .false.)
CALL pglab('Proposed_Distance_(kpc)', 'Probability,',',)
CALL pgend
WRITE (command2,*) 'convert_-rotate_90_` // TRIM(ADJUSTL(string)) // &
    ,/bw_dist_PPD.ps_, // TRIM(ADJUSTL(string)) // &
    ,/bw_dist_PPD.jpg,
call system(command2)
!--_-_-_-_-_-_-_
string2 = TRIM(ADJUSTL(string)) // '/dist_PPD.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(MINVAL(DIST_PPDx, mask = DIST_PPDx .ne. 0.) - 1, MAXVAL(DIST_PPDx) + 1, 0., 1.1*MAXVAL(Dist_PPDy), 0, 0)
DO i = 1, DistBins
    IF (Dist_PPDx(i) .ge. dist_rec - dist_msigma . and. Dist_PPDx(i).lt. dist_rec + dist_psigma) THEN
        CALL pgsci(2)
        CALL pgbin (2, Dist_PPDx(i), Dist_PPDy(i),.false.) !
        IF (Dist_PPDx(i) .eq. dist_rec - dist_msigma) THEN
            xpts = Dist_PPDx(i)
            ypts(1)=0.e0 ; ypts(2)= Dist_PPDy(i) !One Sigma
            CALL pgline (2, xpts, ypts) !
        END IF !Credibility
        IF (Dist_PPDx(i+1).eq. dist_rec + dist_psigma) THEN
            xpts = Dist_PPDx(i+1) !Interval
            ypts(1) = 0.e0 ; ypts(2) = Dist_PPDy(i)
            CALL pgline (2, xpts, ypts)
        END IF
```

ELSE IF (Dist_PPDx(i) .ge. dist_rec - dist_m90 .and. Dist_PPDx(i).lt. dist_rec + dist_p90) THEN
CALL pgsci(3)
CALL pgbin (2, Dist_PPDx(i),Dist_PPDy(i),.false.))
.false.)
IF (Dist_PPDx(i) .eq. dist_rec - dist_m90) THEN
xpts = Dist_PPDx(i)
ypts(1)=0.e0 ; ypts(2)= Dist_PPDy(i) !90 percent
CALL pgline (2, xpts, ypts)
END IF !Credibility
IF (Dist_PPDx(i+1).eq. dist_rec + dist_p90) THEN
xpts = Dist_PPDx(i+1) !Interval
ypts(1) = 0.e0 ; ypts(2)= Dist_PPDy(i) !
CALL pgline (2, xpts, ypts) !
END IF !
ELSE IF (Dist_PPDx(i) .ge. dist_rec - dist_m99 .and. Dist_PPDx(i).lt. dist_rec + dist_p99) THEN
CALL pgsci(4)
CALL pgbin (2, Dist_PPDx(i), Dist_PPDy(i),.false.) !
IF (Dist_PPDx(i) .eq. dist_rec - dist_m99) THEN !
xpts = Dist_PPDx(i)
ypts(1)= 0.e0 ; ypts(2)= Dist_PPDy(i) !99 percent
CALL pgline (2, xpts, ypts)
END IF
IF (Dist_PPDx (i+1).eq. dist_rec + dist_p99) THEN !
xpts = Dist_PPDx(i+1) !Interval
ypts(1)= 0.e0 ; ypts(2)= Dist_PPDy(i) !
CALL pgline (2, xpts, ypts) !
END IF
ELSE
CALL pgsci(1) !Distribution
CALL pgbin (2, Dist_PPDx(i), Dist_PPDy(i),.false.) !outside of 99%
END IF !Cred. Interval
END DO
CALL pgsci(1)
CALL pglab('Proposed_Distance_(kpc)','Probability,,',')
CALL pgend
WRITE (command 2,*) 'convert_-rotate_90_, // TRIM(ADJUSTL(string)) // \&
,/dist_PPD.ps_, // TRIM(ADJUSTL(string)) // \&
'/dist_PPD.jpg'
call system(command2)
!--_-_-_-_-_-_-_-_-_ots M31 to Object Distance Distribution w/o credibility intervals
string2 = TRIM(ADJUSTL(string)) // '/bw_M31dist_PPD.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(MINVAL(M31_to_obj_x, mask = M31_to_obj_y . ne. 0.) - 1,\&
MAXVAL(M31_to_obj-x, mask = M31_to_obj-y .ne. 0.) + 1, \&
0., 1.1*MAXVAL(M31_to_obj-y), 0, 0)
CALL pgbin (4001, M31_to_obj_x, M31_to_obj_y, .false.)
CALL pglab('Proposed_Distance\&from_M31\_(kpc)', 'Probability,, ',')
CALL pgend
WRITE (command 2,*) 'convert_-rotate_90_, // TRIM(ADJUSTL(string)) // \&
,/bw_M31dist_PPD.ps_' // TRIM(ADJUSTL(string)) // \&
,/bw_M31dist_PPD.jpg,
call system(command2)

```
```

7 0 1720
741
END DO

```
END IF
IF (prior_type .eq. 4) Then !For a linear decreasing prior
slope = 2.e0 !Gradiant of diminishing probability
WRITE (2,*) "-"
WRITE (2,*) "Prior_Type:_Linear_Decreasing,_slope_=", slope
DO i = 1, DistBins
    DistPrior(i) = 779.e0 - abs(slope *(Dist_PPDx(i) - 779.e0))
END DO
END IF
IF (prior_type .eq. 5) Then !For a Gaussian Prior
flat = 1.e0 !Gaussian Flattening Factor
hwhm = 150.e0 !Gaussian Half Width Half Maximum
WRITE (2,*) "„"
```



```
DO i = 1, DistBins
```

```
    DistPrior(i) = exp(-((Dist_PPDx(i) - 779.e0) ** (2.e0 * flat)) / (2.e0 * hwhm ** (2.e0 * flat)))
END DO
END IF
DistPrior = DistPrior/SUM(DistPrior)
```



```
string2 = TRIM(ADJUSTL(string)) // '/dist_prior.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(MINVAL(DIST_PPDx, mask = DIST_PPDx .ne. 0.) - 1, MAXVAL(DIST_PPDx) + 1, 0., 1.1*MAXVAL(DistPrior), 0, 0)
CALL pgbin (DistBins, Dist_PPDx, DistPrior, .false.)
CALL pglab(`Proposed_Distance_(kpc)', 'Probability', '')
CALL pgend
WRITE (command2,*) 'convert_-rotate_90」' // TRIM(ADJUSTL(string)) // &
    '/dist_prior.ps_' // TRIM(ADJUSTL(string)) // &
    '/dist-prior.jpg,
call system(command2)
END SUBROUTINE DistancePrior
SUBROUTINE Confidence2 !Identifies the best parameter values and
USE Global !their associated 1 sigma errors from the
IMPLICIT NONE !respective posterior plots
PPD_peak = 0.d0
DO i = 1, DistBins
    IF (Dist_PPDy(i) .gt. PPD_peak) THEN
        PPD_peak = Dist_PPDy(i) !Find best fit TRGB value
        dist_rec = Dist_PPDx(i)
            !
    END IF
END DO
dist_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(Dist_PPDy, DIM = 1), 1, -1
    mcounts = mcounts + Dist_PPDy(i)
END DO
DO i = MAXLOC(Dist_PPDy, DIM = 1), 1, -1
    dist_counts = dist_counts + Dist_PPDy(i)
    IF (dist_counts .ge. 0.682*mcounts) THEN
        dist_msigma = dist_rec - Dist_PPDx(i)
        exit
    END IF
END DO
dist_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(Dist_PPDy, DIM = 1), DistBins
    pcounts = pcounts + Dist_PPDy(i)
END DO
DO i = MAXLOC(Dist_PPDy, DIM = 1), DistBins
    dist_counts = dist_counts + Dist_PPDy(i)
    IF (dist_counts .ge. 0.682*pcounts) THEN !error in distance
!Finds positive one sigma
        dist_psigma = Dist_PPDx(i) - dist_rec
        exit
    END IF
```

```
END DO
dist_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(Dist_PPDy, DIM = 1), 1, -1
    mcounts = mcounts + Dist_PPDy(i)
END DO
DO i = MAXLOC(Dist_PPDy, DIM = 1), 1, -1
    dist_counts = dist_counts + Dist_PPDy(i)
    IF (dist_counts .ge. 0.9*mcounts) THEN
        dist_m90 = dist_rec - Dist_PPDx(i)
        exit
    END IF
END DO
dist_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(Dist_PPDy, DIM = 1), DistBins
    pcounts = pcounts + Dist_PPDy(i)
END DO
DO i = MAXLOC(Dist_PPDy, DIM = 1), DistBins
    dist_counts = dist_counts + Dist_PPDy(i)
    IF (dist_counts .ge. 0.9*pcounts) THEN
        dist_p90 = Dist_PPDx(i) - dist_rec
        exit
    END IF
END DO
dist_counts = 0.d0 ; mcounts = 0.d0
DO i = MAXLOC(Dist_PPDy, DIM = 1), 1, -1
    mcounts = mcounts + Dist_PPDy(i)
END DO
DO i = MAXLOC(Dist_PPDy, DIM = 1), 1, -1
    dist_counts = dist_counts + Dist_PPDy(i)
    IF (dist_counts .ge. 0.99*mcounts) THEN
        dist_m99 = dist_rec - Dist_PPDx(i)
        exit
    END IF
END DO
dist_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(Dist_PPDy, DIM = 1), DistBins
    pcounts = pcounts + Dist_PPDy(i)
END DO
DO i = MAXLOC(Dist_PPDy, DIM = 1), DistBins
    dist_counts = dist_counts + Dist_PPDy(i)
    IF (dist_counts .ge. 0.99* pcounts) THEN
        dist_p99 = Dist_PPDx(i) - dist_rec
        exit
    END IF
END DO
WRITE (2,*) "-"
WRITE (2,*) "Most_Likely_Distance:", dist_rec
WRITE (2,*) "+sigma\_-sigma\_dist+sigma\_dist-sigma:", dist_psigma, dist_msigma, dist_rec + dist_psigma, dist_rec - dist_msigma
WRITE (2,*) "+90_-90_dist+90_dist - 90:", dist_p90, dist_m90, dist_rec + dist_p 90, dist_rec - dist_m90
WRITE (2,*) "+99_-99_dist+99_dist - 99:", dist_p99, dist_m99, dist_rec + dist_p 99, dist_rec - dist_m99
!| Convert distance profile mode and intervals
!\/ back into the equivalent in magnitudes
dist2mag_rec = 5.e0 * LOG10(dist_rec * 100.e0) - 3.44e0
d2m_psigma = (5.e0 * LOG10((dist_rec+dist_psigma) * 100.e0) - 3.44e0) - (5.e0 * LOG10(dist_rec * 100.e0) - 3.44e0)
d2m_msigma = (5.e0 * LOG10(dist_rec * 100.e0) - 3.44e0)-(5.e0* LOG10((dist_rec-dist_msigma) * 100.e0) - 3.44e0)
```

d2m_p90 $=\left(5 . e 0 * \operatorname{LOG10}\left(\left(d i s t \_r e c+d i s t \_p 90\right) * 100 . e 0\right)-3.44 \mathrm{e} 0\right)-\left(5 . e 0 * \operatorname{LOG10}\left(\mathrm{dist} \_\right.\right.$rec $\left.\left.* 100 . \mathrm{e} 0\right)-3.44 \mathrm{e} 0\right)$

d2m_p99 $=(5 . \mathrm{e} 0 * \operatorname{LOG10}((\mathrm{dist}$ _rec+dist_p99)$* 100 . e 0)-3.44 \mathrm{e} 0)-(5 . \mathrm{e} 0 * \operatorname{LOG10}(\mathrm{dist}$ _rec $* 100 . \mathrm{e} 0)-3.44 \mathrm{e} 0)$
d2m_m99 $=\left(5 . e 0 * \operatorname{LOG10}\left(d i s t \_r e c * 100 . e 0\right)-3.44 e 0\right)-\left(5 . e 0 * \operatorname{LOG} 10\left(\left(d i s t \_r e c-d i s t \_m 99\right) * 100 . e 0\right)-3.44 e 0\right)$
! /
! 11
WRITE $(2, *)$ "-"
WRITE (2,*) "Distance $\quad$ back $\quad$ to $\quad$ magnitude: ", dist2mag_rec
WRITE (2,*) "Hence,_distance_modulus_after_applying_prior:", dist2mag_rec + 3.44 e 0
WRITE ( $2, *$ ) "+sigma_-sigma:", d2m_psigma, d2m_msigma


END SUBROUTINE Confidence2
SUBROUTINE Confidence3
USE Global
IMPLICIT NONE
PPD_peak $=0$. d0
DO i $=1,3001$
IF (M31_to_obj-y (i) . gt . PPD_peak) THEN !
PPD_peak $=$ M31_to_obj-y (i) ! Find best fit TRGB value
M31_dist_rec $=$ M31_to_obj_x (i)
!
END IF
END DO
dist_counts $=0 . d 0$; mcounts $=0 . d 0$
DO i = MAXLOC(M31_to_obj-y, DIM = 1), $1,-1$
mcounts $=$ mcounts + M31_to_obj-y (i)
END DO
DO i = MAXLOC(M31_to_obj_y, DIM = 1), $1,-1$
dist_counts $=$ dist_counts + M31_to_obj_y (i) ! Finds negative one sigma
IF (dist_counts .ge. $0.682 *$ mcounts) THEN !error in M31 to object distance
M31_dist_msigma $=$ M31_dist_rec - M31_to_obj_x (i) !
exit
END IF
END DO
dist_counts $=0 . d 0 ;$ pcounts $=0 . d 0$
DO i = MAXLOC(M31_to_obj-y, DIM = 1), 3001
pcounts = pcounts + M31_to_obj-y (i)
END DO
DO i = MAXLOC(M31_to_obj-y, DIM = 1), 3001
dist_counts $=$ dist_counts + M31_to_obj_y (i) !Finds positive one sigma
IF (dist_counts ge. $0.682 *$ pcounts) THEN error in M31 to object distance
M31_dist_psigma $=$ M31_to_obj_x (i) - M31_dist_rec !
exit
END IF
END DO
WRITE (2,*)" "
WRITE (2,*) "Most」Likely_Distance_from_M31:", M31_dist_rec
WRITE (2,*) "+sigma_-sigma_dist+sigma_dist-sigma:", M31_dist_psigma, M31_dist_msigma, \&
M31_dist_rec + M31_dist_psigma, M31_dist_rec - M31_dist_msigma
END SUBROUTINE Confidence 3

## Program: SatPlot. 995

Creation Date: 17 February 2012 (first version 22 Sep 2011)

## Relevant Section: Fig. 10 of Paper II (Ch. 4)

Notes: I wrote this program with the sole objective of producing a three dimensional plot of the satellite system. The distances and positions on the sky of each satellite are read in and used to generate a set of M31-centric cartesian coordinates for each satellite. Rotation matrices are then used to spin the view angle. The use of rotation matrices here is actually not quite correct as the order of application is not given its due importance. This means that the operation is a little clumsy but the plots themselves are unaffected. The actual PAndAS survey area footprint visible in Fig. 10 (c) of Paper II was generated (painstakingly!) for use in 'SatDensity_SampCont.f95,' but I added it to this figure for illustration.

```
MODULE Global ! Define all
IMPLICIT NONE ! Variables
INTEGER :: i, j, k, ios, ndata, SAP_ndata, ICP_ndata, v_angle !SAP = Survey Area Point
PARAMETER (ndata = 28)
PARAMETER (SAP_ndata = 135)
REAL :: x(ndata), y(ndata), z(ndata), MWy_to_Obj(ndata), M31_to_Obj(ndata), m31_dist, m31_psig, m31_msig, x_pro(ndata), y_pro(ndata)
REAL :: SAP_xi(SAP_ndata), SAP_eta(SAP_ndata), SAP_x(SAP_ndata), SAP_y(SAP_ndata), SAP_theta, MWy_to_SAP(SAP_ndata)
REAL :: SAPn_x(SAP_ndata), SAPn_y(SAP_ndata), SAPf_x(SAP_ndata), SAPf_y(SAP_ndata), MWy_to_SAPn(SAP_ndata), MWy_to_SAPf(SAP_ndata)
REAL :: ICP_xi(300), ICP_eta(300), ICP_x(300), ICP_y(300), ICP_theta, MWy_to_ICP(300)
PARAMETER (m31_dist = 779.e0)
PARAMETER (m31_psig = 0.e0)
PARAMETER (m31_msig = 0.e0)
REAL :: xi(ndata), eta(ndata), theta(ndata), p_sig(ndata), m_sig(ndata), pi, dummy
PARAMETER (pi = ACOS(-1.e0))
REAL :: M31_to_Obj_psig(ndata), M31_to_Obj_msig(ndata), temp1, temp2
CHARACTER :: name(ndata) *20, string*200
!-_-_-_For Rotate subroutine --_-_
REAL :: obj_rot(ndata,3), obj_pro_rot(ndata,3), obj(ndata,5), obj_pro(ndata,3)
REAL :: obj_rotp(ndata,3), obj_rotm(ndata,3)
REAL :: SAP_rot(SAP_ndata, 3), SAP(SAP_ndata,3), SAPn_rot(SAP_ndata,3), SAPn(SAP_ndata,3), SAPf_rot(SAP_ndata,3), SAPf(SAP_ndata,3)
REAL :: ICP_rot (300,3), ICP (300,3)
REAL :: x_rot (3,3), y_rot(3,3), z_rot(3,3), rot_mat (3,3), x_axis(3), y_axis(3), z_axis(3)
REAL :: marker_x (3), marker_y (3), marker_z (3)
REAL :: alpha, beta, gamma
END MODULE Global
```

```
PROGRAM SatPlot !Master program
```

PROGRAM SatPlot !Master program
USE Global
IMPLICIT NONE
CALL GetData

```
```

4 0
0.e0 *(pi/180.e0)
beta = 0.e0 * (pi/180.e0) !Plots satellite
gamma = 0.e0 * (pi/180.e0) !positions on
v_angle = 1 !xy plane
CALL Rotate !
CALL Plot
string = 'M31_neighborhood_xz.ps/CPS' !
alpha = 90.e0 * (pi/180.e0)
beta = 0.e0 * (pi/180.e0) !Plots satellite
gamma = 0.e0 *(pi/180.e0) !positions on
v_angle = 2 !xz plane
CALL Rotate !
!CALL Plot !
string = 'M31_neighborhood_yz.ps/CPS' !
alpha = 0.e0 * (pi/180.e0)
beta = 270.e0*(pi/180.e0) !Plots satellite
gamma = 0.e0 * (pi/180.e0) !positions on
v_angle = 3 !yz plane
CALL Rotate
CALL Plot
string = 'M31_neighborhood_xyz.ps/CPS'!
alpha = 5.e0 * (pi/180.e0) !
beta = 5.e0 * (pi/180.e0) !Plots satellite positions
gamma = 0.e0 * (pi/180.e0) !in 3D.
v_angle = 4 !
CALL DistancePerspective !Remove Effects of distance on x/y positions of satellites
CALL Rotate
CALL Plot
END PROGRAM SatPlot
SUBROUTINE GetData !Get data :)
USE Global
IMPLICIT NONE
OPEN (unit = 1, file = './SatStats.dat', status = 'old')
i = 0 ; ios = 0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
i = i+1
READ (1, *, IOSTAT = ios) xi(i), eta(i), theta(i), Mwy_to_Obj(i), p_sig(i), m_sig(i), name(i)
if (ios == 0) then ;
else if (ios == - 1) then ;
i=i-1
exit ;
else if (ios > 0) then,
i=i-1
cycle
end if
END DO
DO i = 1, ndata-1

```
```

    xi(i)= xi(i)*(pi/180.e0) !
    eta(i)= eta(i) * (pi/180.e0) !Convert angles from degrees to radians
    theta(i) = theta(i)*(pi/180.e0)
x(i) = ABS(MWy_to_Obj(i) * cos(theta(i)) * tan(xi(i))) !Determine length of x vector for each satellite
IF (xi(i) .lt. 0.e0) THEN !
x(i) = -1.e0 * x(i) !Determine if x is positive or negative
END IF
y(i) = ABS(MWy_to_Obj(i) * sin(eta(i)))
IF (eta(i).1t. 0.e0) THEN
y(i)=-1.e0 * y(i)
END IF
z(i)= MWy_to_Obj(i) * cos(theta(i)) - m31_dist !Determine length and sign of z vector
M31_to_Obj(i)=(((x(i))** 2.e0) +((y(i)) ** 2.e0) + ((z(i)) ** 2.e0)) ** 0.5e0 !Determine distance between M31 and satellite
temp1 = ((2.e0 * MWy_to_obj(i)) - 2.e0 * m31_dist * cos(theta(i))) ** 2.e0 / \&
((MWy_to_Obj(i)** 2.e0) + (m31_dist ** 2.e0) - (2.e0 * MWy_to_Obj(i) * m31_dist * cos(theta(i))) )
temp2 = ((2.e0 * m31_dist) - 2.e0 * MWy_to_obj(i) * cos(theta(i))) ** 2.e0 / \&
((MWy_to_Obj(i)** 2.e0) + (m31_dist ** 2.e0) - (2.e0 * MWy_to_Obj(i)*m31_dist * cos(theta(i))) )
M31_to_Obj_psig(i) = SQRT((temp1) * (p_sig(i) ** 2.e0) + (temp2) * (12.e0 ** 2.e0))
M31_to_Obj_msig(i) = SQRT((temp1) * (m_sig(i) ** 2.e0) + (temp2) * (11.e0 ** 2.e0))
WRITE (*,*) name(i) ! Write x,y and z components of M31-to-satellite separation vector along with vector
WRITE (*, '(6F16.5)') x(i), y(i), z(i), M31_to_Obj(i), M31_to_Obj_psig(i), M31_to_Obj_msig(i) !magnitude and uncertainties

```

\section*{END DO}


```

OPEN(unit = 2, file = '../SurveyArea/Border_Coords_XiEta.dat', status = 'old'')

```
OPEN(unit = 2, file = '../SurveyArea/Border_Coords_XiEta.dat', status = 'old'')
i=0; ios = 0
i=0; ios = 0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
    i = i+1
    i = i+1
    READ (2,*, IOSTAT = ios) SAP_xi(i), SAP_eta(i) !SAP = Survey Area Point
    READ (2,*, IOSTAT = ios) SAP_xi(i), SAP_eta(i) !SAP = Survey Area Point
    if (ios == 0) then ;
    if (ios == 0) then ;
    else if (ios == -1) then ;
    else if (ios == -1) then ;
        i=i-1
        i=i-1
        exit ;
        exit ;
    else if (ios > 0) then ;
    else if (ios > 0) then ;
        i=i-1
        i=i-1
        cycle
        cycle
    end if
    end if
END DO
END DO
DO i = 1, SAP_ndata
DO i = 1, SAP_ndata
    SAP_xi(i) = SAP_xi(i)*(pi/180.e0) !
    SAP_xi(i) = SAP_xi(i)*(pi/180.e0) !
    SAP_eta(i) = SAP_eta(i) * (pi/180.e0) !Convert angles from degrees to radians
    SAP_eta(i) = SAP_eta(i) * (pi/180.e0) !Convert angles from degrees to radians
    SAP_theta = SQRT(SAP_xi(i)**2.e0 + SAP_eta(i)**2.e0) * (pi/180.e0)!
    SAP_theta = SQRT(SAP_xi(i)**2.e0 + SAP_eta(i)**2.e0) * (pi/180.e0)!
    MWy_to_SAP(i) = m31_dist / cos(SAP_theta) !Determine Survey Area Point Distance assuming plane at distance of M31
    MWy_to_SAP(i) = m31_dist / cos(SAP_theta) !Determine Survey Area Point Distance assuming plane at distance of M31
    SAP_x(i) = ABS(MWy_to_SAP(i) * cos(SAP_theta) * tan(SAP_xi(i))) !Determine length of x vector for each Survey
    SAP_x(i) = ABS(MWy_to_SAP(i) * cos(SAP_theta) * tan(SAP_xi(i))) !Determine length of x vector for each Survey
    IF (SAP_xi(i) . lt. 0.e0) THEN !Area Point, assuming plane at distance of M31
    IF (SAP_xi(i) . lt. 0.e0) THEN !Area Point, assuming plane at distance of M31
        SAP_x(i) = -1.e0 * SAP_x(i) !Determine if x is positive or negative
```

        SAP_x(i) = -1.e0 * SAP_x(i) !Determine if x is positive or negative
    ```
```

END IF
SAP_y(i) = ABS(MWy_to_SAP(i) * sin(SAP_eta(i)))
IF (SAP_eta(i) . 1t. 0.e0) THEN
SAP_y(i) = -1.e0 * SAP_y(i)
END IF
MWy_to_SAPn(i) = (m31_dist + z(12)) / cos(SAP_theta) !Determine Survey Area Point Distance assuming plane at distance of ANDXVI
SAPn_x(i) = ABS(MWy_to_SAPn(i) * cos(SAP_theta) * tan(SAP_xi(i))) !Determine length of x vector for each Survey
IF (SAP_xi(i).lt. 0.e0) THEN !Area Point, assuming plane at distance of ANDXVI
SAPn_x(i) = -1.e0 * SAPn_x(i) !Determine if x is positive or negative
END IF
SAPn_y(i) = ABS(MWy_to_SAPn(i) * sin(SAP_eta(i)))
IF (SAP_eta(i) .lt. 0.e0) THEN
CDetermine length of y vector for each Survey
!Area Point, assuming plane at distance of ANDXVI
SAPn_y(i) = -1.e0 * SAPn_y(i)
END IF
MWy_to_SAPf(i) = (m31_dist + z(23)) / cos(SAP_theta) ! Determine Survey Area Point Distance assuming plane at distance of ANDXVIII
SAPf_x(i) = ABS(MWy_to_SAPf(i) * cos(SAP_theta) * tan(SAP_xi(i))) !Determine length of x vector for each Survey
IF (SAP_xi(i).lt. 0.e0) THEN !Area Point, assuming plane at distance of ANDXVIII
SAPf_x(i) = -1.e0 * SAPf_x(i) !Determine if x is positive or negative
END IF !
SAPf_y(i) = ABS(MWy_to_SAPf(i) * sin(SAP_eta(i))) !Determine length of y vector for each Survey
IF (SAP_eta(i).lt. 0.e0) THEN !Area Point, assuming plane at distance of ANDXVIII
SAPf_y(i) = -1.e0 * SAPf_y(i) !Determine if y is positive or negative
END IF
END DO

```

```

OPEN (unit = 3, file = '../SurveyArea/M31CutOffEllipse.dat', status = 'old')
i = 0 ; ios=0
DO WHILE (.TRUE.) !Reads data until end of input file and puts it into arrays
i = i+1
READ (3, *, IOSTAT = ios) ICP_xi(i), ICP_eta(i) !ICP = Inner Cut-Off Point
if (ios == 0) then ;
else if (ios == - 1) then ;
i=i - 1
exit ;
else if (ios > 0) then ;
i=i-1
cycle
end if
END DO
ICP_ndata = i
DO i = 1, ICP_ndata
ICP_xi(i) = ICP_xi(i) *(pi/180.e0)
ICP_eta(i) = ICP_eta(i) * (pi/180.e0) !Convert angles from degrees to radians
ICP_theta = SQRT(ICP_xi(i)**2.e0 + ICP_eta(i)**2.e0) * (pi/180.e0)!
MWy_to_ICP(i) = m31_dist / cos(ICP_theta) !Determine Inner Cut-Off Point Distance assuming plane at distance of M31

```
```

277

```
```279

``` IF (ICP_xi(i).1t. 0.e0) THEN ! Cut-Off Point, assuming plane at distance of M31
        ICP_x(i) = -1.e0 * ICP_x (i)
    END IF
    ICP_y(i) = ABS(MWy_to_ICP(i) * sin(ICP_eta(i)))
    IF (ICP_eta(i) . 1t. 0.e0) THEN
        ICP_y(i) = -1.e0 * ICP_y (i)
    END IF
END DO
CLOSE(1)
CLOSE (2)
CLOSE (3)
END SUBROUTINE GetData
SUBROUTINE DistancePerspective ! This subroutine is for removing the effects of perspective on the 3D view.
USE Global !It scales x and y positions of satellites relative to the closest satellite
IMPLICIT NONE !(And XVI) to preserve the on sky view of the satellites.
DO i = 1, ndata
    x(i) = x(i) * ((m31_dist + z(12))/ (m31_dist + z(i)))
    y(i)}=y(i)*((m31_dist + z(12))/ (m31_dist + z(i)))
END DO
DO i = 1, SAP_ndata
    SAP_x(i) = SAP_x(i) * ((m31_dist + z(12))/m31_dist) !PAndAS survey border at distance of M31
    SAP_y(i) = SAP_y(i) * ((m31_dist + z(12))/ m31_dist) !
    SAPn_x (i) = SAPn_x (i) * ((m31_dist + z(12))/ (m31_dist + z(12))) !PAndAS survey border at distance of And XVI
    SAPn_y(i) = SAPn_y(i) * ((m31_dist + z(12))/ (m31_dist + z(12))) !(And XVI is the nearest satellite)
    SAPf_x(i) = SAPf_x(i) * ((m31_dist + z(12))/ (m31_dist + z(23))) !PAndAS survey border at distance of And XXVII
    SAPf_y(i) = SAPf_y(i) * ((m31_dist + z(12))/ (m31_dist + z(23))) !(And XXVII is the furthest satellite)
END DO
DO i = 1, ICP_ndata
    ICP_x(i) = ICP_x(i) * ((m31_dist + z(12))/ m31_dist) !Inner cutoff ellipse at
    ICP_y(i) = ICP_y(i) * ((m31_dist + z(12))/ m31_dist) !distance to M31.
END DO
END SUBROUTINE DistancePerspective
SUBROUTINE Rotate !Uses rotation matrices to shift
USE Global !the position on screen of the
IMPLICIT NONE !satellites based on the viewing angle
x_axis(1) = MAXVAL(abs(x)) ; x_axis(2)= 0.e0 ; x_axis(3) = 0.e0 !Generate coordinates of
y_axis(1) = 0.e0 ; y_axis(2)=MAXVAL(abs(y)) ; y_axis(3) = 0.e0 !the positive ends of the
z_axis(1) = 0.e0 ; z_axis(2) = 0.e0 ; z_axis(3) = MAXVAL(abs(z)) !x, y and z axes
marker_x (1) = 100.e0 ; marker_x (2) = 0.e0 ; marker_x (3) = 0.e0 ! Generate coordinates of
marker_y (1) = 0.e0 ; marker_y (2) = 100.e0 ; marker_y (3) = 0.e0 !the positive 100 kpc
marker_z (1) = 0.e0 ; marker_z (2) = 0.e0 ; marker_z (3) = 100.e0 !axis markers
x_rot(1,1) = 1.e0
x_rot(2,1) = 0.e0
x_rot (3,1) = 0.e0
```

```
x_rot (1,2)=0.e0 !
x_rot(2,2) = cos(alpha) !Rotation matrix for adjusting yaw - angle alpha
x_rot(3,2)=-1.e0*sin(alpha) !
x_rot (1,3) = 0.e0
x_rot (2,3) = sin(alpha) !
x_rot (3,3) = cos(alpha)!
y_rot(1,1)=\operatorname{cos(beta) !}
y_rot (2,1) = 0.e0
y_rot (3,1) = sin(beta) !
y_rot (1,2) = 0.e0 !
y_rot(2,2)=1.e0 !Rotation matrix for adjusting pitch - angle beta
y_rot(3,2) = 0.e0
y_rot(1,3)=-1.e0 * sin(beta) !
y_rot (2,3)=0.e0 !
y_rot (3,3) = cos(beta)
z_rot(1,1) = cos(gamma)
z_rot (2,1) = -1.e0 * sin(gamma)
z_rot(3,1) = 0.e0
z_rot(1,2) = sin(gamma)
z_rot(2,2)= cos(gamma) !Rotation matrix for adjusting roll - angle gamma
z_rot (3,2) = 0.e0 !
z_rot (1,3)=0.e0 !
z_rot (2,3)=0.e0 !
z_rot (3,3)=1.e0 !
rot_mat = MATMUL(x_rot, y_rot) !Generate rotation matrix to adjust object coordinates
rot_mat = MATMUL(rot_mat, z_rot) !for the chosen combination of yaw, pitch and roll
x_axis = MATMUL(rot_mat, x_axis) !
y_axis = MATMUL(rot_mat, y_axis) !Convert coordinates of positive ends of axes for new view angle
z_axis = MATMUL(rot_mat, z_axis)
marker_x = MATMUL(rot_mat, marker_x)
marker_y = MATMUL(rot_mat, marker_y)
marker_z = MATMUL(rot_mat, marker_z)
DO i = 1, ndata
    obj(i,1)=x(i) !
    obj(i,2)=y(i) !Convert object coordinates
    obj(i,3)=z(i) !for new viewing angle
    obj(i,4)=z(i) + p_sig(i) !For error
    obj(i,5) = z(i) - m_sig(i) !bars
    obj_rot(i,:) = MATMUL(rot_mat, obj(i,(/ 1,2,3 /))) !
    obj_rotp(i,:) = MATMUL(rot_mat, obj(i,(/ 1, 2,4 /))) !
    obj_rotm(i,:) = MATMUL(rot_mat, obj(i,(/ 1,2,5 /))) !
    obj_pro(i,1) = x(i)
    obj_pro(i,2)=y(i) !Convert z = 0 projection of object coordinates
    obj_pro(i,3)=0.e0 !for new viewing angle to form other end of plotted z vector
    obj_pro_rot(i,:) = MATMUL(rot_mat, obj_pro(i,:)) !
END DO
DO i = 1, SAP_ndata
    SAP(i,1) = SAP_x(i) !
    SAP(i,2) = SAP_y(i) !Convert SAP coordinates
    SAP(i,3)=0.e0 !for new viewing angle
    SAP_rot(i,:) = MATMUL(rot_mat, SAP(i,:))
    SAPn(i, 1) = SAPn_x(i)
```

```
    SAPn(i,2) = SAPn_y(i)
    !Convert SAPn coordinates
    SAPn(i,3) = z(12)
    !for new viewing angle
    SAPn_rot(i,:) = MATMUL(rot_mat, SAPn(i,:))
    SAPf(i,1) = SAPf_x(i)
    SAPf(i,2)= SAPf_y(i)
    SAPf(i,3) = z(23)
    SAPf_rot(i,:) = MATMUL(rot_mat, SAPf(i,:))
    !
END DO
DO i = 1, ICP_ndata
    ICP(i,1) = ICP_x (i)
    ICP(i,2)= ICP_y(i) !Convert ICP coordinates
    ICP(i,3)=0.e0 !for new viewing angle
    CP_rot(i,: ) = MATMUL(rot_mat, ICP(i,:))
END DO
END SUBROUTINE Rotate
```

SUBROUTINE Plot ! Plot the satellites for chosen view angle
USE Global
IMPLICIT NONE
CALL pgbegin(0,TRIM(ADJUSTL(string)) , 1, 1)
IF (v_angle .eq. 1 . or . v_angle .eq. 2) THEN
CALL pgenv (1.1*(MAXVAL(ABS (obj_rot (: 1) )) ), $-1.1 *\left(\operatorname{MAXVAL}\left(\operatorname{ABS}\left(o b j \_r o t(:, 1)\right)\right)\right), \& \quad!$ Set frame limits
$\left.-1.1 *(\operatorname{MAXVAL}(\operatorname{ABS}(\operatorname{obj} \operatorname{rot}(:, 2)))), \quad 1.1 *\left(\operatorname{MAXVAL}\left(\operatorname{ABS}\left(\operatorname{obj} \_\operatorname{rot}(:, 2)\right)\right)\right), 1,0\right)!$
ELSE IF (v_angle .eq. 3) THEN
CALL pgenv(1.1*(MINVAL(obj_rot $(:, 1))), \quad 1.1 *\left(\operatorname{MAXVAL}\left(\operatorname{ABS}\left(o b j \_r o t(:, 1)\right)\right)\right), \& \quad$ \& and parameters for
$1.1 *\left(\operatorname{MINVAL}\left(\operatorname{SAPf} \_\right.\right.$rot $\left.\left.(:, 2)\right)\right), \quad 1.1 *\left(\operatorname{MAXVAL}\left(\operatorname{SAPf} \_\right.\right.$rot $\left.\left.\left.(:, 2)\right)\right), \quad 1,0\right)$ !
ELSE !
CALL pgenv $\left(1.1 *\left(\operatorname{MAXVAL}\left(\operatorname{ABS}\left(\operatorname{SAP} \_r o t(:, 1)\right)\right), \quad-1.1 *(\operatorname{MAXVAL}(\operatorname{ABS}(\operatorname{SAP} \quad\right.\right.$ rot $(:, 1)))), \& \quad$ ! various viewing angles
$-1.1 *\left(\operatorname{MAXVAL}\left(\operatorname{ABS}\left(S A P \_r o t(:, 2)\right)\right)\right), \quad 1.1 *\left(\operatorname{MAXVAL}\left(\operatorname{ABS}\left(S A P \_r o t(:, 2)\right)\right), 1,-1\right)!$
END IF
CALL pgline (2, (/x-axis(1), -1.e0 * x-axis(1)/), (/x-axis(2), -1.e0*x-axis(2)/)) !Draw lines from positive
CALL pgline (2, (/y-axis(1), -1.e0 $*$ y-axis(1)/), (/y-axis (2), $-1 . e 0 * y-a x i s(2) /)) \quad$ !end to negative end of
CALL pgline (2, (/z_axis(1), -1.e0 * z_axis(1)/), (/z_axis(2), -1.e0* z-axis(2)/)) !x, y and z axes
IF (v_angle .eq. 1) THEN
CALL pgptxt(x-axis(1) - 5., x-axis (2) - 10., 0., 0.5, 'x')
CALL pgptxt(y_axis(1) - 10., y_axis(2) - 5., 0., 0.5, 'y')
END IF
IF (v_angle .eq. 3) THEN ! Print $x, y$ and $z$
CALL pgptxt(y-axis(1) - 10., y-axis (2) - 10., 0., 0.5, 'y') !
CALL pgptxt(z_axis (1) - 10., z_axis (2) $\left.-15 ., 0 ., 0.5,{ }^{\prime} z^{\prime}\right) \quad$ !at the positive ends of
END IF
IF (v_angle .eq. 4) THEN !their respective axes
CALL pgptxt (x_axis (1) - 5., x_axis (2) $-5 ., 0 ., 0.5,{ }^{\prime}{ }^{\prime}$ ) !
CALL pgptxt (y-axis(1) - 5., y-axis (2) - 5., 0., 0.5, 'y') !
CALL pgptxt(z_axis (1) - 5., z-axis (2) $-5 ., 0 ., 0.5,{ }^{\prime}{ }^{\prime}$ ) !
END IF
IF (v_angle .ne. 4) THEN
CALL pgpt(1, marker_x (1), marker_x (2), 0612)
CALL pgpt (1,-1.e0 $*$ marker_x $_{-1}$ ), $\left.-1 . \mathrm{e} 0 * \operatorname{marker}_{-x}(2), 0612\right)$
CALL pgpt(1, marker_y (1), marker_y (2), 0590)
CALL pgpt (1,-1.e0 * marker_y (1), $-1 . e 0 *$ marker_y (2), 0590)

```
406
```

END IF

```
END IF
IF (v_angle .eq. 2) THEN !Plot
IF (v_angle .eq. 2) THEN !Plot
CALL pgpt(1,marker_z(1), marker_z(2), 0590)
CALL pgpt(1,marker_z(1), marker_z(2), 0590)
CALL pgpt(1,-1.e0 * marker_z(1), -1.e0 * marker_z(2), 0590) !Markers
CALL pgpt(1,-1.e0 * marker_z(1), -1.e0 * marker_z(2), 0590) !Markers
ELSE
ELSE
CALL pgpt(1,marker_z(1), marker_z(2), 0612)
CALL pgpt(1,marker_z(1), marker_z(2), 0612)
CALL pgpt(1,-1.e0 * marker_z(1), -1.e0 * marker_z(2), 0612)
CALL pgpt(1,-1.e0 * marker_z(1), -1.e0 * marker_z(2), 0612)
END IF
END IF
CALL PGSCH(0.75)
CALL PGSCH(0.75)
DO i = 1, 24 !For satellites with prefix 'And'
DO i = 1, 24 !For satellites with prefix 'And'
    CALL pgslw (2)
    CALL pgslw (2)
    CALL pgsci(2)
    CALL pgsci(2)
    CALL pgline(2, (/obj_pro_rot(i,1), obj_rot(i,1)/), (/obj-pro_rot(i,2), obj_rot(i,2)/)) !Draw z vector of object for
    CALL pgline(2, (/obj_pro_rot(i,1), obj_rot(i,1)/), (/obj-pro_rot(i,2), obj_rot(i,2)/)) !Draw z vector of object for
    CALL pgslw(15) !chosen view angle
    CALL pgslw(15) !chosen view angle
    CALL pgpt(1, obj_rot(i,1), obj_rot(i,2), -1) !Draw large dot at the end of the z vector
    CALL pgpt(1, obj_rot(i,1), obj_rot(i,2), -1) !Draw large dot at the end of the z vector
    CALL pgslw(2)
    CALL pgslw(2)
    CALL pgsci(2)
    CALL pgsci(2)
END DO
END DO
CALL pgslw(8)
CALL pgslw(8)
CALL pgsci(3)
CALL pgsci(3)
DO i = 25, 26 !For NGC147 and NGC185
DO i = 25, 26 !For NGC147 and NGC185
    CALL pgslw(2)
    CALL pgslw(2)
    CALL pgsci(3)
    CALL pgsci(3)
    CALL pgline(2, (/obj_pro_rot(i,1), obj_rot(i,1)/), (/obj_pro_rot(i,2), obj_rot(i,2)/)) !Draw z vector of object for
    CALL pgline(2, (/obj_pro_rot(i,1), obj_rot(i,1)/), (/obj_pro_rot(i,2), obj_rot(i,2)/)) !Draw z vector of object for
    CALL pgslw(20) !chosen view angle
    CALL pgslw(20) !chosen view angle
    CALL pgpt(1, obj_rot(i,1), obj_rot(i,2), -1) !Draw large dot at the end of the z vector
    CALL pgpt(1, obj_rot(i,1), obj_rot(i,2), -1) !Draw large dot at the end of the z vector
    CALL pgslw(2)
    CALL pgslw(2)
    CALL pgsci(3)
    CALL pgsci(3)
END DO
END DO
DO i = 27, 27 !For M33
DO i = 27, 27 !For M33
    CALL pgslw(2)
    CALL pgslw(2)
    CALL pgsci(4)
    CALL pgsci(4)
    CALL pgline(2, (/obj_pro_rot(i,1), obj_rot(i,1)/), (/obj_pro_rot(i,2), obj_rot(i,2)/)) !Draw z vector of object for
    CALL pgline(2, (/obj_pro_rot(i,1), obj_rot(i,1)/), (/obj_pro_rot(i,2), obj_rot(i,2)/)) !Draw z vector of object for
    CALL pgslw(30) !chosen view angle
    CALL pgslw(30) !chosen view angle
    CALL pgpt(1, obj_rot(i,1), obj_rot(i,2), -1) !Draw large dot at the end of the z vector
    CALL pgpt(1, obj_rot(i,1), obj_rot(i,2), -1) !Draw large dot at the end of the z vector
    CALL pgslw(2)
    CALL pgslw(2)
    CALL pgsci(4)
    CALL pgsci(4)
END DO
END DO
IF (v_angle .eq. 4) THEN
IF (v_angle .eq. 4) THEN
DO i = 1, 27
DO i = 1, 27
    CALL pgslw (2)
    CALL pgslw (2)
    CALL pgsci(1) !Plot square on survey plane
    CALL pgsci(1) !Plot square on survey plane
    CALL pgpt(1, obj_pro_rot(i,1), obj_pro_rot(i,2), 0254) !
    CALL pgpt(1, obj_pro_rot(i,1), obj_pro_rot(i,2), 0254) !
END DO
END DO
END IF
END IF
CALL pgslw(30)
CALL pgslw(30)
CALL pgsci(4)
CALL pgsci(4)
CALL pgpoint (1, 0., 0., -1) !Plot large dot at origin for M31
CALL pgpoint (1, 0., 0., -1) !Plot large dot at origin for M31
CALL pgslw(2)
CALL pgslw(2)
CALL pgsci(4)
CALL pgsci(4)
IF (v_angle .eq. 4) THEN !Plot Survey Area and Inner Cut-Off Ellipse on plane at M31 distance
IF (v_angle .eq. 4) THEN !Plot Survey Area and Inner Cut-Off Ellipse on plane at M31 distance
CALL pgsci(1)
```

CALL pgsci(1)

```
```

DO i = 2, SAP_ndata - 1
CALL pgline(2, (/SAP_rot(i-1,1), SAP_rot(i,1)/), (/SAP_rot(i-1,2), SAP_rot(i,2)/))
END DO
CALL pgline(2, (/SAP_rot(SAP_ndata - 1,1), SAP_rot(1,1)/), (/SAP_rot(SAP_ndata - 1, 2), SAP_rot(1, 2)/))
DO i = 2, ICP_ndata - 1
CALL pgline(2, (/ICP_rot(i-1,1), ICP_rot(i,1)/),(/ICP_rot(i-1,2), ICP_rot(i,2)/))
END DO
CALL pgline(2, (/ICP_rot(ICP_ndata - 1,1), ICP_rot(1,1)/), (/ICP_rot(ICP_ndata - 1, 2), ICP_rot(1,2)/))
END IF
IF (v_angle .eq. 3) THEN !Plot Survey Area on planes at And XVI and And XXVII distances
CALL pgsci(1)
DO i = 2, SAP_ndata - 1
CALL pgline(2, (/SAPn_rot(i - 1,1), SAPn_rot(i,1)/), (/SAPn_rot(i-1,2), SAPn_rot(i,2)/))
END DO
CALL pgline(2, (/SAPn_rot(SAP_ndata - 1,1), SAPn_rot(1,1)/), (/SAPn_rot(SAP_ndata - 1, 2), SAPn_rot(1, 2)/))
DO i = 2, SAP_ndata - 1
CALL pgline(2, (/SAPf_rot(i-1,1), SAPf_rot(i,1)/), (/SAPf_rot(i-1,2), SAPf_rot(i,2)/))
END DO
CALL pgline(2, (/SAPf_rot(SAP_ndata-1,1), SAPf_rot(1,1)/), (/SAPf_rot(SAP_ndata - 1, 2), SAPf_rot(1, 2)/))
!| Plot lines from the top of the two survey areas
!\/ to the bottom of the two survey areas
CALL pgline(2, (/MAXVAL(SAPn_rot (:,1)), MAXVAL(SAPf_rot (:, 1))/), (/MAXVAL(SAPn_rot(:,2)), MAXVAL(SAPf_rot(:,2))/))
CALL pgline(2, (/MINVAL(SAPn_rot(:,1)), MINVAL(SAPf_rot(:,1))/), (/MINVAL(SAPn_rot(:,2)), MINVAL(SAPf_rot(:,2))/))
END IF
IF (v_angle .eq. 1) THEN ! Prints satellite labels for xy plane view
CALL pgsci(1)
CALL pgptxt(obj_pro_rot(1,1) + 7., obj_pro_rot(1,2), 0., 0.5, 'I')
CALL pgptxt(obj_pro_rot(2,1) + 7., obj_pro_rot(2,2), 0., 0.5, 'II')
CALL pgptxt(obj_pro_rot(3,1) - 9., obj_pro_rot(3,2), 0., 0.5, 'III')
CALL pgptxt(obj_pro_rot(4,1) - 8., obj_pro_rot(4,2) -4., 0., 0.5, 'V')
CALL pgptxt(obj_pro_rot(5,1) + 8., obj_pro_rot(5,2), 0., 0.5,'IX')
CALL pgptxt(obj_pro_rot (6,1) + 8., obj_pro_rot (6,2), 0., 0.5, 'X')
CALL pgptxt(obj_pro_rot(7,1) + 8., obj_pro_rot(7,2), 0., 0.5, 'XI')
CALL pgptxt(obj_pro_rot(8,1) - 6., obj_pro_rot(8,2) - 12., 0., 0.5, 'XII')
CALL pgptxt(obj_pro_rot (9,1) + 11., obj-pro_rot (9,2), 0., 0.5,'XIII')
CALL pgptxt(obj_pro_rot(10,1) + 11., obj_pro_rot(10,2), 0., 0.5, 'XIV')
CALL pgptxt(obj_pro_rot(11,1) - 9., obj_pro_rot(11,2), 0., 0.5, 'XV')
CALL pgptxt(obj_pro_rot(12,1) + 12., obj_pro_rot(12,2), 0., 0.5, 'XVI')
CALL pgptxt(obj_pro_rot(13,1) - 12., obj_pro_rot(13,2) - 3., 0., 0.5, 'XVII')
CALL pgptxt(obj_pro_rot(14,1), obj_pro_rot(14,2) - 11., 0., 0.5,'XVIII')
CALL pgptxt(obj_pro_rot(15,1) + 12., obj_pro_rot(15,2), 0., 0.5, 'XIX')
CALL pgptxt(obj_pro_rot (16,1), obj_pro_rot(16,2) + 6., 0., 0.5, 'XX')
CALL pgptxt(obj_pro_rot(17,1) + 14., obj_pro_rot(17,2) - 5., 0., 0.5, 'XXI')
CALL pgptxt(obj_pro_rot(18,1) - 13., obj_pro_rot(18,2) - 2., 0., 0.5, 'XXII')
CALL pgptxt(obj-pro_rot(19,1) - 14., obj_pro_rot(19,2), 0., 0.5, 'XXIII')
CALL pgptxt(obj_pro_rot(20,1) + 5., obj_pro_rot(20,2) - 13., 0., 0.5, 'XXIV')
CALL pgptxt(obj_pro_rot(21,1) - 14., obj-pro_rot(21,2) - 2., 0., 0.5, 'XXV')
CALL pgptxt(obj_pro_rot(22,1) - 15., obj_pro_rot(22,2) - 3., 0., 0.5, 'XXVI')
!CALL pgptxt(obj_pro_rot(23,1) - 16., obj_pro_rot(23,2) - 10., 0., 0.5, 'XXVII') ! Hidden behind NGC147

```

CALL pgptxt (obj-pro_rot (24,1) - 0., obj_pro_rot (24,2) + 5., 0., 0.5, 'XXX')
CALL pgptxt (obj_pro_rot (25,1) - 23., obj_pro_rot (25,2) + 7., 0., 0.5, 'NGC147') CALL pgptxt (obj-pro_rot \((26,1)+28 .\), obj_pro_rot \((26,2)-4 ., 0 ., 0.5, \quad\) NGC185') CALL pgptxt (obj_pro_rot \((27,1)-7 .\), obj-pro_rot \((27,2)-18 ., 0 ., 0.5, \quad\) M33') CALL pgptxt (obj_pro_rot (28,1) - 15., obj_pro_rot (28,2) - 15., 0., 0.5, 'M31') END IF

IF (v_angle .eq. 3) THEN ! Prints satellite labels for yz plane view CALL pgsci(1)

CALL pgptxt \(\left(0.8 *(\operatorname{obj} \operatorname{rot}(2,1))\right.\), obj_rot \(\left.(2,2)-15 ., 0 ., 0.5,{ }^{\prime} I^{\prime}{ }^{\prime}\right)\)
CALL pgptxt (0.5 * (obj_rot \((3,1))\), obj_rot \((3,2)+3 ., 0 ., 0.5,{ }^{\prime}\) III' \()\) CALL pgptxt (1.23 * (obj_rot \((4,1))\), obj_rot \(\left.(4,2)-4 ., 0 ., 0.5, \quad V^{\prime}\right)\) CALL pgptxt (1.08 * (obj_rot (5,1)), obj_rot (5,2) - 3., 0., 0.5, 'IX')
CALL pgptxt (1.1 * (obj_rot \((6,1))\), obj_rot \(\left.(6,2)-2 ., 0 ., 0.5, X^{\prime}\right)\) CALL pgptxt \((0.5 *(\operatorname{obj} \operatorname{rot}(7,1)), \operatorname{obj} \operatorname{rot}(7,2)+2 ., 0 ., 0.5, \quad\) XI' \()\) CALL pgptxt (0.5 * (obj_rot \((8,1))\), obj_rot \((8,2)-15 ., 0 ., 0.5, \quad\) XII') CALL pgptxt \(\left(0.5 *\left(\operatorname{obj} \_r o t(9,1)\right), \operatorname{obj}\right.\) _rot \((9,2)-15 ., 0 ., 0.5,{ }^{\prime}\) XIII' \()\) CALL pgptxt (7.0 * (obj_rot (10,1)), obj_rot (10,2) - 3., 0., 0.5, 'XIV') CALL pgptxt (0.75 * (obj_rot (11,1)), obj_rot (11,2) + 2., 0., 0.5, 'XV') CALL pgptxt (0.85 * (obj_rot \((12,1)), \operatorname{obj} \_\)rot \((12,2)+2 ., 0 ., 0.5, \quad\) XVI') CALL pgptxt (1.35 * (obj rot \((13,1))\), obj_rot (13,2)-10., 0., 0.5, 'XVII') CALL pgptxt (0.95 * (obj_rot \((14,1))\), obj_rot \((14,2)-12 ., 0 ., 0.5, \quad\) XVIII') CALL pgptxt (1.6 * (obj_rot \((15,1))\), obj_rot (15,2), 0., 0.5, 'XIX') CALL pgptxt (0.5 * (obj_rot \((16,1))\), obj_rot \((16,2), 0 ., 0.5, \quad \mathrm{XX}\) ') CALL pgptxt (1.4 * (obj_rot (17,1)), obj_rot(17,2)-3., 0., 0.5, 'XXI') CALL pgptxt \(\left(0.5 *\left(o b j \_r o t(18,1)\right), \operatorname{obj} \operatorname{rot}(18,2)-15 ., 0 ., 0.5, \quad\right.\) XXII') CALL pgptxt \(\left(0.5 *\left(o b j \_r o t(19,1)\right), \operatorname{obj} \operatorname{rot}(19,2)+2 ., 0 ., 0.5, ' X X I I I '\right)\) CALL pgptxt (0.75 * (obj_rot \((20,1))\), obj_rot \((20,2)-15 ., 0 ., 0.5, \quad\) XXIV') CALL pgptxt (0.45 * (obj_rot \((21,1))\), obj_rot \((21,2)-12 ., 0 ., 0.5, \quad\) 'XXV') CALL pgptxt (0.75 * (obj_rot \((22,1))\), obj_rot \((22,2)+2 ., 0 ., 0.5, \quad\) XXVI') CALL pgptxt \((0.95 *(\operatorname{obj} \operatorname{rot}(23,1))\), obj_rot \((23,2)+3 ., 0 ., 0.5, \quad\) XXVII' \()\) CALL pgptxt (0.5 * (obj_rot \((24,1))\), obj_rot \((24,2)+4 ., 0 ., 0.5\), 'XXX') CALL pgptxt (1.5 * (obj_rot \((25,1))\), obj_rot \((25,2)-4 ., 0 ., 0.5, \quad\) NGC」147') CALL pgptxt \((0.75 *(\operatorname{obj}-\operatorname{rot}(26,1))\), obj_rot \((26,2)-15 ., 0 ., 0.5\), 'NGC185') CALL pgptxt (3.0 * (obj_rot \((27,1))\), obj_rot \((27,2)-2 ., 0 ., 0.5, \quad\) M33') CALL pgptxt \((0.5 *(\operatorname{obj}-r o t(28,1))+30 ., \operatorname{obj}-\operatorname{rot}(28,2)-20 ., 0 ., 0.5, \quad\) M31') END IF

IF (v_angle .eq. 4) THEN ! Prints satellite labels for 3D view
CALL pgsci(1)
CALL pgptxt (obj-pro_rot (1, 1) + 5., obj_pro_rot (1, 2), 0., 0.5, ' I')
CALL pgptxt (obj-pro_rot \((2,1)+5 .\), obj_pro_rot \((2,2), 0 ., 0.5,{ }^{\prime} I^{\prime}\) ')
CALL pgptxt (obj-pro_rot (3,1)-7., obj_pro_rot (3,2), 0., 0.5, 'III') CALL pgptxt (obj_pro_rot (4, 1) - 6., obj_pro_rot (4,2) -2., 0., 0.5, 'V') CALL pgptxt (obj-pro_rot (5, 1) + 6., obj_pro_rot (5,2), 0., 0.5, 'IX') CALL pgptxt (obj_pro_rot \((6,1)+6 .\), obj_pro_rot \(\left.(6,2), 0 ., 0.5, \mathrm{X}^{\prime}\right)\) CALL pgptxt (obj_pro_rot (7,1), obj_pro_rot (7,2) - 8., 0., 0.5, 'XI')
 CALL pgptxt (obj-pro_rot \((9,1)+8 ., \operatorname{obj}_{-}\)pro_rot \((9,2), 0 ., 0.5,{ }^{\prime}\) XIII' \()\) CALL pgptxt (obj-pro_rot ( 10,1 ) - 6., obj_pro_rot ( 10,2 ), 0., 0.5, 'XIV') CALL pgptxt (obj_pro_rot (11,1) - 7., obj_pro_rot(11,2), 0., 0.5, 'XV') CALL pgptxt (obj_pro_rot \((12,1)+8 .\), obj_pro_rot \((12,2), 0 ., 0.5, \quad\) XVI') CALL pgptxt (obj-pro_rot (13,1) - 9., obj_pro_rot ( 13,2 ), 0., 0.5, 'XVII') CALL pgptxt (obj_pro_rot (14,1), obj_pro_rot(14,2)-8., 0., 0.5, 'XVIII') CALL pgptxt (obj_pro_rot ( 15,1 ) +8 ., obj_pro_rot \((15,2), 0 ., 0.5,{ }^{\prime}\) XIX') CALL pgptxt (obj-pro_rot ( 16,1 ), obj_pro_rot (16,2) + 4., 0., 0.5, ' XX ') CALL pgptxt (obj_pro_rot (17,1) + 9., obj_pro_rot (17,2) - 2., 0., 0.5, 'XXI') CALL pgptxt (obj_pro_rot \((18,1)+9 .\), obj_pro_rot \((18,2)+1 ., 0 ., 0.5\), \({ }^{\prime}\) XXII') CALL pgptxt (obj_pro_rot (19,1) - 9., obj_pro_rot (19,2), 0., 0.5, 'XXIII') CALL pgptxt (obj_pro_rot \((20,1)+6 .\), obj_pro_rot \((20,2)-8 ., 0 ., 0.5, \quad\) XXIV')
```

CALL pgptxt(obj-pro_rot(21,1) - 9., obj-pro_rot(21,2) - 2., 0., 0.5, 'XXV')
CALL pgptxt(obj-pro_rot(22,1) - 10., obj_pro_rot(22,2) - 2., 0., 0.5, 'XXVI')
CALL pgptxt(obj_pro_rot(23,1) - 11., obj-pro_rot(23,2) - 2., 0., 0.5, 'XXVII')
CALL pgptxt(obj-pro_rot(24,1) - 7., obj-pro_rot (24,2) + 2., 0., 0.5, 'XXX')
CALL pgptxt(obj_pro_rot(25,1) - 8., obj_pro_rot(25,2) + 4., 0., 0.5, 'NGC147')
CALL pgptxt(obj-pro_rot(26,1) + 18., obj-pro_rot(26,2) - 4., 0., 0.5, 'NGC185')
CALL pgptxt(obj-pro_rot(27,1), obj-pro_rot(27,2) - 8., 0., 0.5, 'M33')
CALL pgptxt(obj_pro_rot(28,1) - 7., obj_pro_rot(28,2) - 8., 0., 0.5, 'M31')
END IF

```

```

CALL pgslw(2)
CALL pgsci(1)
CALL PGSCH(1.0)
IF (v_angle .eq. 1 .or. v_angle .eq. 2 .or. v_angle .eq. 3) THEN
CALL pglab('kpc', 'kpc',',') !axis labels
END IF
CALL pgend
END SUBROUTINE PLOT

```

\section*{Program: SatDensity_SampCont.f95}

Creation Date: 20 Feb 2012
Relevant Section: \(\S 4.3\) of Paper II (Ch. 4)
Notes: The analysis presented in \(\S 4.3\) of Paper II concerning the density profile of the M31 halo was carried out using this program. It fits a spherically-symmetric unbroken power law to the satellite density profile, taking into account the uneven coverage of the PAndAS survey area. One of the most difficult tasks here was to actually generate the PAndAS survey polygon from the field centers specified by Mike Irwin (Institute of Astronomy, University of Cambridge) and this took me two days to complete. Using this polygon as the outer boundary, and using the inner cutoff ellipse around the M31 disk, it is possible to determine how much volume at each radius (i.e. distance from M31) would fall inside the utilized survey region. The resulting function can then be used to weight the density profile so that we can obtain an unbiased measure of the slope of the power law for the desired sample of satellites. Note that the program can perform the analysis using either the best-fit distances alone or the full distance distributions for each satellite. The code presented here is as applied to the whole satellite sample but desired satellites can be omitted from the sample by skipping over them in the 'MaxLike' subroutine.
```

MODULE Global !Define all
IMPLICIT NONE !variables
INTEGER :: i, j, h, n, nn, k_ndata
REAL :: Sat_Rad(27), alpha, alpha_hold(600), ML_logL(600), Rel_ML_logL(600), norm_fac
REAL :: k_vs_alpha(601,2), k(600)
REAL :: alpha_counts, alpha_psigma, alpha_msigma, pcounts, mcounts
DOUBLE PRECISION :: av_ML_logL
!|| For sampled distributions
!\/ with MCMC
INTEGER :: nit, it, nsamples, ndata_max, ios, idum = -9999
PARAMETER (nit = 3000000)
PARAMETER (nsamples = 10000)
PARAMETER (ndata_max = 3000000)
INTEGER :: time (ndata_max)
REAL :: Sat_Radii (27,500001), M33_dist, in_cut, out_cut, LikeA, LikeB, logL, randnum, r
REAL :: Radius(27,10000)
REAL :: x(ndata_max, 3), p(3)
REAL :: post-y1(600), post-y2(100), post_y3(300), Best_Combo(6)
REAL :: post_x1(600), post_x2(100), post_x3(300), pi
PARAMETER ( pi = 3.141592)
CHARACTER :: folder *100, string*200, string2*200, command*200, sample
PARAMETER (sample = 'y')
END MODULE Global

```

PROGRAM SatDensity_SampCont ! Master program
USE Global
IMPLICIT NONE
```

WRITE (folder,*) 'Sat_Density,

```
WRITE (string,*) , /, // TRIM(ADJUSTL(folder))
WRITE (command,*) 'mkdir_, // TRIM(ADJUSTL(folder))
CALL system (command)
CALL random_seed
CALL NonSampledRadii
IF (sample .eq. 'y') THEN ! If using many samples of possible
    CALL SampledRadii !radii for each object as opposed
    CALL SampleSelect ! to just the best fit radius
END IF
CALL k_verse_alpha
CALL MaxLike
string2 \(=\operatorname{TRIM}(\operatorname{ADJUSTL}(\) folder \()) / /\), results.dat ,
\(\operatorname{OPEN}(3, \mathbf{f i l e}=\operatorname{TRIM}(\operatorname{ADJUSTL}(\) string 2\())\), status = 'unknown')
WRITE ( \(3, *\) ) "Results \(\lrcorner\) for \(\lrcorner\) Maximum」Likelihoodュtest"
WRITE ( \(3, *\) ) "Most_Likely_alpha_(Max_Likelihood):", alpha_hold (MAXLOC(ML_logL))

END PROGRAM SatDensity_SampCont

\section*{SUBROUTINE NonSampledRadii}

\section*{USE Global}

\section*{IMPLICIT NONE}
Sat_Rad (2) \(=196 . \mathrm{e} 0 \quad\) !And II
Sat_Rad (3) \(=86 . \mathrm{e} 0 \quad!\) And III
Sat_Rad (4) \(=113 . \mathrm{e} 0 \quad\) :And V
Sat_Rad (5) \(=182 . \mathrm{e} 0 \quad\) !And IX
Sat_Rad (6) \(=130 . \mathrm{e} 0 \quad\) : And \(X\)
Sat_Rad (7) = 103.e0 \(\quad\) And XI
Sat_Rad (8) \(=182 . \mathrm{e} 0 \quad\) :And XII
Sat_Rad (9) \(=116 . \mathrm{e} 0 \quad\) !And XIII
Sat_Rad (10) \(=163 . \mathrm{e} 0 \quad\) ! And XIV
Sat_Rad (11) \(=174 . \mathrm{e} 0 \quad\) :And XV
Sat_Rad (12) \(=320 . \mathrm{e} 0 \quad\) ! And XVI
Sat_Rad (13) \(=67 . \mathrm{e} 0 \quad\) :And XVII
Sat_Rad (14) \(=457 . \mathrm{e} 0 \quad\) !And XVIII
Sat_Rad (15) \(=116 . \mathrm{e} 0 \quad\) !And XIX
Sat_Rad (16) \(=129 . \mathrm{e} 0 \quad\) : And XX
Sat_Rad (17) \(=136 . \mathrm{e} 0 \quad!\) And XXI
Sat_Rad (18) \(=280 . \mathrm{e} 0 \quad\) !And XXII
Sat_Rad (19) \(=128 . \mathrm{e} 0 \quad\) ! And XXIII
Sat_Rad (20) \(=169 . \mathrm{e} 0 \quad\) :And XXIV
Sat_Rad (21) \(=\) 91.e0 \(\quad\) :And XXV
Sat_Rad (22) \(=103 . \mathrm{e} 0 \quad\) :And XXVI
Sat_Rad (23) \(=482 . \mathrm{e} 0 \quad\) ! And XXVII
Sat_Rad (24) \(=146 . \mathrm{e} 0 \quad!\) And XXX
Sat_Rad (25) \(=118 . \mathrm{e} 0 \quad\) ! NGC147
```

```
Sat_Rad(26) = 181.e0 !NGC185
```

```
Sat_Rad(26) = 181.e0 !NGC185
Sat_Rad(27) = 214.e0 !M33
Sat_Rad(27) = 214.e0 !M33
END SUBROUTINE NonSampledRadii
END SUBROUTINE NonSampledRadii
SUBROUTINE k_verse_alpha ! This subroutine finds values of the power law normalization factor 'k' for
SUBROUTINE k_verse_alpha ! This subroutine finds values of the power law normalization factor 'k' for
USE Global !each power law (i.e. exponent 'alpha'). It simply interpolates based on values
USE Global !each power law (i.e. exponent 'alpha'). It simply interpolates based on values
IMPLICIT NONE !derived by Geraint (provided in 'k_vs_alpha.dat')
IMPLICIT NONE !derived by Geraint (provided in 'k_vs_alpha.dat')
OPEN(40, file='./k_vs_alpha.dat', status = 'unknown')
OPEN(40, file='./k_vs_alpha.dat', status = 'unknown')
i = 0
i = 0
DO WHILE (.TRUE.)
DO WHILE (.TRUE.)
i = i + 1
i = i + 1
READ (40, *, IOSTAT = ios) k_vs_alpha(i,1), k_vs_alpha(i, 2)
READ (40, *, IOSTAT = ios) k_vs_alpha(i,1), k_vs_alpha(i, 2)
IF (ios == -1) THEN
IF (ios == -1) THEN
    i = i - 1
    i = i - 1
    exit
    exit
ELSE IF (ios .gt. 0) THEN
ELSE IF (ios .gt. 0) THEN
    WRITE (*,*) i
    WRITE (*,*) i
    i= i - 1
    i= i - 1
    cycle
    cycle
END IF
END IF
END DO
END DO
CLOSE(36)
CLOSE(36)
k_ndata = i
k_ndata = i
DO i = 1, 599 !k as determined for alpha = 0.01, 0.02, \ldots, 5.99
DO i = 1, 599 !k as determined for alpha = 0.01, 0.02, \ldots, 5.99
    alpha = REAL(i)/100.e0
    alpha = REAL(i)/100.e0
    nn = INT((alpha - k_vs_alpha(1,1))/(k_vs_alpha(241,1) - k_vs_alpha(1,1)) * 241.e0) + 1
    nn = INT((alpha - k_vs_alpha(1,1))/(k_vs_alpha(241,1) - k_vs_alpha(1,1)) * 241.e0) + 1
    k(i) = (alpha - k_vs_alpha(nn,1))/(k_vs_alpha(nn+1, 1) - k_vs_alpha(nn, 1)) * (k_vs_alpha(nn+1,2) - k_vs_alpha(nn, 2)) + k_vs_alpha(
    k(i) = (alpha - k_vs_alpha(nn,1))/(k_vs_alpha(nn+1, 1) - k_vs_alpha(nn, 1)) * (k_vs_alpha(nn+1,2) - k_vs_alpha(nn, 2)) + k_vs_alpha(
        nn,2)
        nn,2)
    k(i) = 10.e0 ** k(i)
    k(i) = 10.e0 ** k(i)
END DO
END DO
DO i = 600, 600 !k for alpha = 6.00
DO i = 600, 600 !k for alpha = 6.00
    alpha = REAL(i)/100.e0
    alpha = REAL(i)/100.e0
    nn = 241
    nn = 241
    k(i) = (alpha - k_vs_alpha(nn,1))/(k_vs_alpha(nn+1, 1) - k_vs_alpha(nn, 1)) * (k_vs_alpha(nn+1,2) - k_vs_alpha(nn, 2)) + k_vs_alpha(
    k(i) = (alpha - k_vs_alpha(nn,1))/(k_vs_alpha(nn+1, 1) - k_vs_alpha(nn, 1)) * (k_vs_alpha(nn+1,2) - k_vs_alpha(nn, 2)) + k_vs_alpha(
        nn,2)
        nn,2)
    k(i) = 10.e0 ** k(i)
    k(i) = 10.e0 ** k(i)
END DO
END DO
END SUBROUTINE k_verse_alpha
```

```
END SUBROUTINE k_verse_alpha
```

```
\(\qquad\)
.
```

SUBROUTINE MaxLike ! Direct calculation of likelihoods for alpha
USE Global !assuming fixed values of inner and outer
IMPLICIT NONE !cutoff radii. See Eq. }16\mathrm{ of Paper II for operation.
string2 = TRIM(ADJUSTL(folder)) // ,/alpha_ML_dist.dat,
OPEN(9, file=TRIM(ADJUSTL(string2)), status = 'unknown')
ML_logL = 0.e0
DO i = 1, 600
print *, i
alpha = REAL(i)/100.e0
alpha_hold(i) = alpha
IF (sample .eq. 'y') THEN
DO j = 1, 27
av_ML_logL = 0.d0
DO h = 1, 500000 !nsamples
av_ML_logL = av_ML_logL + ((k(i) * Sat_Radii(j, h) ** (2.d0 - alpha)))
END DO
av_ML_logL = LOG10(av_ML_logL)
ML_logL(i) = ML_logL(i) + av_ML_logL
END DO
ELSE
DO j = 1, 27
ML_logL(i) = ML_logL(i) + LOG10((k(i) * Sat_Rad(j) ** (2.e0 - alpha))) !Determine likelihood of given alpha
END DO
END IF
END DO
!| Plot
!\/ Distribution
string2 = TRIM(ADJUSTL(string)) // ,/ml_alpha_loglike.ps/CPS,
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
CALL pgenv(0., 6., 0.9 * MINVAL(ML_logL), 1.1 * MAXVAL(ML_logL ), 0, 0)
CALL pgbin(600, alpha_hold, ML_logL, .false.)
CALL pglab('$0627)', 'Log_Likelihood', ,')
CALL pgend
WRITE (command,*) 'convert_-rotate_90_' // TRIM(ADJUSTL(string)) // &
    ,/ml_alpha_loglike.ps_' // TRIM(ADJUSTL(string)) // &
    '/ml_alpha_loglike.jpg,
call system(command)
!/\
!|
!| Plot
!\/ Distribution
string2 = TRIM(ADJUSTL(string)) // '/ml_alpha_PPD.ps/CPS'
CALL pgbegin(0,TRIM(ADJUSTL(string2)),1,1)
DO i = 1, 600
```
```
    Rel_ML_logL(i) = 10.** (ML_logL(i) - MAXVAL(ML_logL))
END DO
Rel_ML_logL = Rel_ML_logL/ SUM(Rel_ML_logL)
CALL pgenv(0., 3.5, 0., 1.1 * MAXVAL(Rel_ML_logL), 0, 0)
DO i = 1, 600
    WRITE (9, '(3F16.5)') alpha_hold(i), Rel_ML_logL(i), ML_logL(i)
END DO
CALL pgline(600, alpha_hold, Rel_ML_logL)
CALL pglab(`\(0627)', 'Probability', ',')
CALL pgend
WRITE (command,*) 'convert_-rotate _90」' // TRIM(ADJUSTL(string)) // &
    `/ml_alpha_PPD.ps_' // TRIM(ADJUSTL(string)) // &
    //ml_alpha_PPD.jpg,
call system(command)
!/\
!|
alpha_counts = 0.d0 ; mcounts = 0.d0 !
DO i = MAXLOC(REL_ML_logL, DIM = 1), 1, -1
    mcounts = mcounts + REL_ML_logL(i)
END DO
DO i = MAXLOC(REL_ML_logL, DIM = 1), 1, -1
    alpha_counts = alpha_counts + REL_ML_logL(i) !Finds negative
    IF (alpha_counts .ge. 0.682*mcounts) THEN !one sigma error
        alpha_msigma = alpha_hold(MAXLOC(REL_ML_logL, DIM = 1)) - alpha_hold(i)
        exit
    END IF
END DO
alpha_counts = 0.d0 ; pcounts = 0.d0
DO i = MAXLOC(REL_ML_logL, DIM = 1), 600
    pcounts = pcounts + REL_ML_logL(i)
END DO
DO i = MAXLOC(REL_ML_logL, DIM = 1), 600
    alpha_counts = alpha_counts + REL_ML_logL(i) !Finds positive
    IF (alpha_counts .ge. 0.682*pcounts) THEN !one sigma error
        alpha_psigma = alpha_hold(i) - alpha_hold(MAXLOC(REL_ML_logL, DIM = 1))
        exit
    END IF
END DO
END SUBROUTINE
!---------------------------------------------------------------------------------------------------
!MCMC subroutine omitted
!---------------------
&-_-
!LogLike subroutine omitted
SUBROUTINE SampledRadii !Read in sampled radii probability
```
```
USE Global ! Distributions for each satellite
IMPLICIT NONE
OPEN (unit = 11, file = './AndromedaIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 12, file ='./AndromedaIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 13, file = './AndromedaIIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 14, file = './AndromedaVe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 15, file ='./AndromedaIXe/other_plots2/Sampled_M31_Distances.dat', status ='old')
OPEN (unit = 16, file = './AndromedaXe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 17, file ='./AndromedaXIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 18, file = './AndromedaXIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 19, file = './AndromedaXIIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 20, file ='./AndromedaXIVe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 21, file = './AndromedaXVe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 22, file =,./AndromedaXVIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 23, file = './AndromedaXVIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 24, file =, ./AndromedaXVIIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 25, file = './AndromedaXIXe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 26, file = './AndromedaXXe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 27, file =, ./AndromedaXXIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 28, file ='./AndromedaXXIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 29, file =, ./AndromedaXXIIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 30, file ='./AndromedaXXIVe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 31, file = './AndromedaXXVe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 32, file ='./AndromedaXXVIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 33, file = './AndromedaXXVIIe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 34, file = './AndromedaXXXe/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 35, file =, ./NGC147e_outer/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN (unit = 36, file = './NGC185e_outer/other_plots2/Sampled_M31_Distances.dat', status = 'old')
OPEN(unit = 37, file =,./M33e/other_plots2/Sampled_M31_Distances.dat', status ='old')
```
! \| Read in the distribution of
\(!\backslash$ distances of Andromeda I from M31
$\mathrm{i}=0$
DO WHILE (.TRUE.)
$\mathrm{i}=\mathrm{i}+1$
READ ( $11, *, \operatorname{IOSTAT}=$ ios) $\operatorname{Sat\_ Radii}(1, i)$
IF (ios = - 1) THEN
$\mathrm{i}=\mathrm{i}-1$
exit
ELSE IF (ios .gt. 0) THEN
WRITE (*,*) i
$\mathrm{i}=\mathrm{i}-1$
cycle
END IF
END DO
!Repeat $*^{*}$ to read in the distance distribution
! of each satellite with respect to M31
$\operatorname{CLOSE}(11)$; CLOSE (12) ; CLOSE (13) ; CLOSE (14) ; CLOSE (15) ; CLOSE (16) ; CLOSE(17) ; CLOSE(18) ; CLOSE(19) ; CLOSE(20)
CLOSE (21) ; CLOSE (22) ; CLOSE (23) ; CLOSE (24) ; CLOSE (25) ; CLOSE (26) ; CLOSE(27) ; CLOSE (28) ; CLOSE(29) ; CLOSE(30)
CLOSE (31) ; CLOSE (32) ; CLOSE (33) ; CLOSE (34) ; CLOSE (35) ; CLOSE (36) ; CLOSE (37)
```END SUBROUTINE SampledRadii SUBROUTINE SampleSelect USE Global IMPLICIT NONE REAL :: gasdev DO i = 1, 27     DO j = 1, nsamples     1 CALL random_number (randnum)         randnum = (randnum * 500000) +1         IF (Sat_Radii(i, NINT(randnum)) .ge. 50.e0 .and. Sat_Radii(i, NINT(randnum)) . le. 600.e0) THEN             Radius(i,j) = Sat_Radii(i, NINT(randnum))             WRITE (*,*) i, j, Radius(i,j)         ELSE             goto 1         END IF     END DO END DO END SUBROUTINE SampleSelect```

## E

## Chapter Five Programs

PlaneSigRMS.f95 ..... 264
Alternative Plane Fitting Code Segments ..... 278
Subroutines for Processing Satellite Subsets ..... 280
PlaneSigS ubS ets_RandReal4_noGroup.f95 ..... 288
pole_vicinity_counts_satid_w.f95 ..... 294
aitoff_hammer.f95 ..... 299
RR_Histograms.f95 ..... 306

## Program: PlaneSigRMS.f95

Creation Date: 7 June 2012 (first version Feb 2012) Many modifications.
Relevant Section: Ch. 5
Notes: This program is representative of the many versions used to implement plane fitting on the satellite distribution. The satellite distance distributions (along with the best fit distances) are read in and stored for subsequent sampling in the 'SampledDist' subroutine. The 'Significance' subroutine then samples distances for each satellite (and M31 itself), converts these distances into 3D positions and calls 'MaxSigFind' to do the actual plane fitting (see the preface for Paper III (p. 90) and Fig. 5.1 for details). The program is currently set to repeat this process for 200,000 realizations of possible positions of the satellites as well as the particular realization where each satellite is in the position defined by its best-fit distance (mainly for plotting). The program is also set up to build 200, 000 random realizations (generated in the 'RandomPoints' subroutine) of the (27) satellites and perform equivalent plane fitting on each. The version of 'RandomPoints' included in this program includes only one possible position for each satellite, and is used only in $\S 3.3$ of Paper III. The version of this subroutine used in all other sections can be seen in the program 'PlaneS igSubSets_RandReal4_noGroup.f95' (p. 288). This version represents each satellite by a distance distribution containing 1,000 possible positions along the line of sight from Earth (i.e. an accurate representation of the real data). Note also that this program is designed to perform plane-fitting on the whole sample. The modified code segments designed to handle each satellite combination of a given size can be seen in Subroutines for Processing Satellite Subsets (p. 280). The 'goodness of fit statistic' used for the plane fitting by this program is the RMS. The code for alternative measures are given in Alternative Plane Fitting Code Segments (p. 278).

```
MODULE Global ! Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, j, k, s, mm, ios, idum = -9999, it, nit, err_samp
INTEGER :: ndata_max, nsats
PARAMETER (ndata_max = 10000000)
PARAMETER (nit = 200000)
PARAMETER (err_samp = 200000)
PARAMETER (nsats = 27)
REAL*8 :: pi
PARAMETER (pi = ACOS(-1.e0))
REAL :: randnum, sig(nit), err_sig(nit), norm(3), best_fit_vect(3), best_fit_sigma
```

```
REAL :: pos(3,ndata_max), temp_pos(3)
REAL :: a(ndata_max), b(ndata_max), c(ndata_max), d(ndata_max)
REAL :: a_hist(201,2), b_hist (201,2), c_hist(201,2), d_hist(2001, 2)
REAL :: logL, LikeA, LikeB, r, p(4), p_temp(4), min_sigma, max_sigma
REAL*8 :: sigma, planeDist, like, plane_sig, rms
REAL :: max_plane_sig, min_signif, Actual_sig, Actual_bfv(3), Actual_bfs
INTEGER :: dummy, sat_pick
REAL :: Sat_Dist(500000,27), Sat_Dist_Drawn(27), Sat_Pos(27,2), xi(27), eta(27), theta(27), M31_Dist_PPD(30000000)
REAL*8 :: RA, DEC, xi_dble, eta_dble
REAL :: xi_test, eta_test, theta_test, SAP_xi(134), SAP_eta(134), spotR
REAL :: Best_Sat_Dist(27)
REAL :: m31_dist
REAL :: alpha_set, beta_set, gamma_set, pole_alpha, pole_beta
REAL :: x_rot (3,3), y_rot (3,3), z_rot (3,3)
REAL :: par_like (180,6)
REAL :: theta_coord, phi_coord
CHARACTER :: argv*30, folder*100, string*200, string2*200, command *200, subsize* *
END MODULE Global
PROGRAM PlaneSignificance !Master program
USE Global
IMPLICIT NONE
WRITE (subsize,*) nsats
WRITE (folder,*) 'RMS_Plane_Stats_' // TRIM(ADJUSTL(subsize)) // ' -sats, !Primary output
WRITE (string,*) , /, // TRIM(ADJUSTL(folder)) !directory
WRITE (command,*) 'mkdir_' // TRIM(ADJUSTL(folder))
CALL system(command)
CALL random_seed ! Insure random seed for random numbers
CALL SampledDist !Get sampled satellite distances
CALL BorderGet !Get PAndAS survey boundary points
string2 = TRIM(ADJUSTL(folder)) // '/results.dat, !File for result summary e.g. plane orient-
OPEN(11, file=TRIM(ADJUSTL(string2)), status = 'unknown') !ation and RMS for real satellite distribution
string2 = TRIM(ADJUSTL(folder)) // ,/sat_pos.dat, !Positions of satellites in
OPEN(12, file=TRIM(ADJUSTL(string2)), status = 'unknown') !Random Realizations
string2 = TRIM(ADJUSTL(folder)) // '/significance.dat, !RMS distribution and poles from
OPEN(13, file=TRIM(ADJUSTL(string2)), status = 'unknown') !Random Realizations
string2 = TRIM(ADJUSTL(folder)) // '/real_sig_wth_err.dat' !RMS distribution and poles for realizations
OPEN(14, file=TRIM(ADJUSTL(string2)), status = 'unknown') !of the real satellite distribution
CALL Significance !The main subroutine which in turn calls the plane fitting subroutine
CALL Theta_Phi(Actual_bfv(1), Actual_bfv(2), Actual_bfv(3))
WRITE(11,*) "Best^fit^vector:^(", Actual_bfv(1), Actual_bfv(2), Actual_bfv(3), ")"
WRITE(11,*) "Theta\iota=", theta_coord, ";\lrcornerPhi\lrcorner=", phi_coord
WRITE(11,*) "LOG10(RMS)_of_best_fit:", Actual_sig
WRITE(11,*) "Minimum_RMS_from_random_samples:", min_signif
```

```
CLOSE(11) ; CLOSE(12) ; CLOSE(13) ; CLOSE(14)
END PROGRAM PlaneSignificance
SUBROUTINE RandomPoints ! Generates Random Realizations of Satellites
USE Global
IMPLICIT NONE
LOGICAL :: in_poly
CALL random_number (randnum)
randnum = randnum * 2999999.e0 + 1.e0
m31_dist = M31_Dist_PPD(NINT(randnum))
DO i = 1, nsats
2 CALL random_number(randnum)
    sat_pick = 1 + NINT(randnum*REAL(nsats - 1)) !Draw a random satellite
    CALL random_number(randnum)
    randnum = randnum * 499999.e0 + 1.e0
    pos(1,i) = ABS(Sat_Dist(NINT(randnum),sat_pick) * cos(theta(sat_pick)) * tan(xi(sat_pick))) !Determine length of x vector
    IF (xi(sat_pick).lt. 0.e0) THEN ! !for each satellite
        pos(1,i)=-1.e0 * pos(1,i) !Determine if x is positive or negative
    END IF !
    pos(2,i) = ABS(Sat_Dist(NINT(randnum),sat_pick)* sin(eta(sat_pick))) !Determine length of y vector for each satellite
    IF (eta(sat_pick) .lt. 0.e0) THEN
        pos(2,i)=-1.e0 * pos(2,i) !Determine if y is positive or negative
    END IF
    pos(3,i) = Sat_Dist(NINT(randnum),sat_pick) * cos(theta(sat_pick)) - m31_dist ! Determine length and sign of z vector
    pos(3,i)=SQRT((pos(1,i)**2.e0) + (pos(2,i)**2.e0) + (pos(3,i)**2.e0)) !Rotate position vector to point
    pos(1,i) = 0.e0 ; pos(2,i) = 0.e0 !along z-axis
    CALL random_number(randnum)
    alpha_set = randnum * 360.e0 * (pi/180.e0) !Pick random longitude
    CALL random_number(randnum) !Pick random latitude between 0 and 90 weighted
    beta_set = ASIN(randnum) !by area of a sphere as a function of latitude
    CALL random_number(randnum)
    IF (randnum .lt. 0.5e0) THEN !Re-assign latitude as
        beta_set = beta_set !
    ELSE !-1* latitude in
        beta_set = -beta_set
    END IF ! !50% 0f cases
    CALL Rotate
    pos(:,i) = MATMUL(y_rot, pos(:,i)) !Rotate to the chosen
    pos(:,i) = MATMUL(x_rot,pos(:,i)) !random angle
    xi_test = ATAN(abs(pos(1,i))/(m31_dist + pos(3,i))) !Convert
    IF (pos(1,i) . 1t. 0.e0) THEN !new random
        xi_test = -xi_test !position
    END IF !vector
    eta_test = ATAN(abs(pos(2,i))/SQRT(pos(1,i)**2 + (m31_dist + pos(3,i))**2))!into
    IF (pos(2,i) . lt. 0.e0) THEN !non t.p
        eta_test = -eta_test !eta and
```

```
    END IF
        !xi
    RA = xi_test
    DEC = eta_test !Use sla_DS2TP
                                    !to convert true
    CALL sla_DS2TP (RA, DEC, 0.d0, 0.d0, xi_dble, eta_dble, j) !eta amd xi to
                            !their tangent
                            !plane projections
    xi_test = xi_dble * (180.e0/pi)
    !
    eta_test = eta_dble * (180.e0/pi)
    IF (in_poly(xi_test, eta_test, 134,SAP_xi,SAP_eta)) THEN
    ELSE
        goto 2
    END IF
                                    !Re-generate
                                    !the new
                                    !randomized
                                    !satellite
                                    ! position if
                                    !the current
    spotR = ((xi_test*cos(51.9d0*pi/180.d0) + eta_test*sin(51.9d0*pi/180.d0))**2 / 6.25d0) + & !choice doesn't
    ((xi_test*sin(51.9d0*pi/180.d0) - eta_test*\operatorname{cos(51.9d0*pi/180.d0))**2 / 1.d0) !fall within}
                                    !fall within
                                    ! the PAndAS
    IF (spotR . le. 1.e0) THEN
    ! footprint
        goto 2 !as viewed
    END IF !from Earth
END DO
END SUBROUTINE RandomPoints
SUBROUTINE Rotate ! Rotation Matrices for rotations about x,y and z axes. 'alpha_set,
USE Global ! is the desired rotation angle about the x axis, 'beta_set, about
IMPLICIT NONE !the y axis and 'gamma_set' about the z axis
x_rot(1,1) = 1.e0
x_rot (2,1) = 0.e0
x_rot (3,1) = 0.e0
x_rot (1,2) = 0.e0
x_rot(2,2) = cos(alpha_set) !Rotation about x-axis - angle alpha
x_rot(3,2) = -1.e0 * sin(alpha_set)
x_rot (1,3) = 0.e0
x_rot}(2,3)= sin(alpha_set
x_rot (3,3)= cos(alpha_set) !
y_rot(1,1)= cos(beta_set) !
y_rot (2,1) = 0.e0
y_rot(3,1) = sin(beta_set) !
y_rot (1,2) = 0.e0 !
y_rot (2,2)=1.e0 !Rotation about y-axis - angle beta
y_rot (3,2)=0.e0 !
y_rot (1,3) = -1.e0 * sin(beta_set) !
y_rot (2,3) = 0.e0
y_rot (3,3)= cos(beta_set) !
z_rot(1,1) = cos(gamma_set)
z_rot (2,1) = -1.e0 * sin(gamma_set)
z_rot (3,1) = 0.e0
z_rot (1,2) = sin(gamma_set) !
z_rot (2,2) = cos(gamma_set) !Rotation about z-axis - angle gamma
z_rot (3,2) = 0.e0
z_rot (1,3) = 0.e0
z_rot (2,3) = 0.e0
z_rot (3,3) = 1.e0
```

```
END SUBROUTINE Rotate
```

SUBROUTINE Theta_Phi( $x, y, z$ ) ! Converts from cartesian $x, y, z$ into spherical coordinates theta
USE Global ! and phi (r is not required) for obtaining positions of objects
IMPLICIT NONE ! (and plane normal vector pointings) in M31 galactic coordinates
REAL : : x, y, z
theta_coord $=\operatorname{acos}(\mathrm{z} /(\operatorname{SQRT}(\mathrm{x} * * 2 . \mathrm{e} 0+\mathrm{y} * * 2 . \mathrm{e} 0+\mathrm{z} * * 2 . \mathrm{e} 0)))-(\mathrm{pi} / 2 . \mathrm{e} 0)$
theta_coord $=-$ theta_coord * (180.e0/pi)
IF ( x . gt. 0.e0) THEN
phi_coord $=\operatorname{atan}(y / x)$
ELSE IF (x . lt. 0.e0 . and. y .ge. 0.e0) THEN
phi_coord $=\operatorname{atan}(y / x)+p i$
ELSE IF (x . lt. 0.e0 .and. y . lt. 0.e0) THEN
phi_coord $=\operatorname{atan}(y / x)-p i$
ELSE IF (x .eq. 0.e0 .and. y.gt. 0.e0) THEN
phi_coord $=$ pi/2.e0
ELSE IF (x .eq. $0 . \mathrm{e} 0$. and. $\mathrm{y} . \mathrm{lt}$. $0 . \mathrm{e} 0$ ) THEN
phi_coord $=-\mathrm{pi} / 2 . \mathrm{e} 0$
ELSE IF (x .eq. 0.e0 .and. y .eq. 0.e0) THEN
phi_coord $=0 . e 0$
END IF
phi_coord $=-$ phi_coord $*(180 . e 0 / p i)$
END SUBROUTINE Theta_Phi
SUBROUTINE MaxSigFind ! Finds best fit plane for a satellite distribution by testing goodness of fit of each
USE Global !tested plane. The poles of the tested planes are all approximately equi-distant, taking
IMPLICIT NONE !into account the surface area of a shere as a function of latitude.
!A low resolution run finds the approximate location of the best fit plane's pole and then
! poles around this point are searched at higher resolution.
par_like $=0 . e 0$
max_plane_sig $=9999999 . \mathrm{e} 0$
! \| Low resolution
! \/ plane tests
DO i $=1,30$
beta_set $=\operatorname{REAL}(\mathrm{i} * 3) *(\mathrm{pi} / 180 . \mathrm{e} 0)$

```
    DO j = 1, NINT(120.e0 * cos(beta_set)) !The higher the latitude, the smaller the
                number of points
        alpha_set =(\boldsymbol{REAL}(\textrm{j})/\operatorname{NINT}(120.e0*\operatorname{cos}(beta_set)))}*360.e0*(pi/180.e0
    norm = (/ 0.e0, 0.e0, 1.e0 /)
    CALL Rotate
    norm = MATMUL(y_rot, norm)
    norm = MATMUL(x_rot, norm)
    plane_sig = 0.d0
    rms = 0.d0
    DO k = 1, nsats !RMS Calculation
            planeDist = norm(1)*pos(1,k) + norm(2)*pos(2,k) + norm(3)*pos(3,k)
            ms = rms + (planeDist)**2
    END DO
    rms = SQRT(rms/nsats)
    plane_sig = LOG10(rms)
    IF (plane_sig .lt. max_plane_sig) THEN !Most significant plane has lowest rms
            max_plane_sig = plane_sig !Store approx, low resolution values
            best_fit_vect = norm !of best fit pole and significance
            pole_alpha = alpha_set !Store best fit pole for
            pole_beta = beta_set !high resolution search
        END IF
    END DO
END DO
norm = (/ -1.e0, 0.e0, 0.e0 /) !Test at the actual pole (not included in above loop)
plane_sig = 0.d0
rms = 0.d0
DO k = 1, nsats !RMS Calculation
    planeDist = norm(1)*pos(1,k) + norm(2)*\operatorname{pos}(2,k) + norm(3)*\operatorname{pos(3,k)}
    ms = rms + (planeDist)**2
END DO
rms = SQRT(rms/nsats)
plane_sig = LOG10(rms)
!| High resolution search
!\/ around best fit pole
IF (plane_sig .lt. max_plane_sig) THEN !Condition not met unless the RMS at the actual pole
                                    !was better than anywhere else in the low res search
max_plane_sig = plane_sig
best_fit_vect = norm
DO i = 1, 15
    beta_set =(88.5e0 + (REAL(i)/10.e0)) *(pi/180.e0)
    DO j = 1, NINT(1200.e0 * cos(beta_set)) !The higher the latitude, the smaller the
                                    number of points
        alpha_set =(\operatorname{REAL}(j)/NINT(1200.e0 * cos(beta_set))) * 360.e0 * (pi/180.e0)
        norm = (/ 0.e0, 0.e0, 1.e0 /)
        CALL Rotate
        norm = MATMUL(y_rot, norm)
        norm = MATMUL(x_rot, norm
        plane.sig = 0.d0
        rms = 0.d0
```

```
    DO k = 1, nsats !RMS Calculation
        planeDist = norm(1)*pos(1,k) + norm(2)*pos(2,k) + norm(3)*pos(3,k)
    rms = rms + (planeDist )**2
    END DO
    rms = SQRT(rms/nsats)
    plane_sig = LOG10(rms)
    IF (plane_sig .lt. max_plane_sig) THEN !Most significant plane has lowest rms
        max_plane_sig = plane_sig !Store final, high resolution values
        best_fit_vect = norm !of best fit pole and significance
    END IF
    END DO
END DO
ELSE
DO i = 1, 11
    DO j = 1, 11
        beta_set = pole_beta + 2.e0 * REAL(j-6) * (0.15e0) * (pi/180.e0)
        alpha_set = pole_alpha + 2.e0* REAL(i-6)*(0.15e0)*(pi/180.e0)*(1.e0/cos(beta_set))
        norm = (/ 0.e0, 0.e0, 1.e0 /)
        CALL Rotate
    norm = MATMUL(y_rot, norm)
    norm = MATMUL(x_rot, norm)
    plane_sig = 0.d0
    rms = 0.d0
    DO k = 1, nsats !RMS Calculation
        planeDist = norm(1)*pos(1,k) + norm(2)*\operatorname{pos(2,k) + norm(3)*pos(3,k)}
        rms = rms + (planeDist)**2
    END DO
    rms = SQRT(rms/nsats)
    plane_sig = LOG10(rms)
    IF (plane_sig .lt. max_plane_sig) THEN !Most significant plane has lowest rms
            max_plane_sig = plane_sig !Store final, high resolution values
            best_fit_vect = norm !of best fit pole and significance
        END IF
    END DO
END DO
END IF
END SUBROUTINE MaxSigFind
SUBROUTINE Significance ! Principal subroutine which generates distributions of the
USE Global ! plane fitting statistic (RMS in this case). The plane fitting
IMPLICIT NONE !subroutine 'MaxSigFind' is called from this subroutine
!!||Determine best fit plane and significance for satellite
!!\/positions generated from best fit distances
m31_dist = 779.e0 !M31
Best_Sat_Dist(1) = 727.e0 !And I
Best_Sat_Dist (2) = 630.e0 !And II
Best_Sat_Dist(3) = 723.e0 !And III
Best_Sat_Dist(4) = 742.e0 !And V
```

```
Best_Sat_Dist(5) = 600.e0 !And IX
Best_Sat_Dist(6) = 670.e0 !And X
Best_Sat_Dist(7) = 763.e0 !And XI
Best_Sat_Dist (8) = 928.e0 !And XII
Best_Sat_Dist(9) = 760.e0 !And XIII
Best_Sat_Dist(10) = 793.e0 !And XIV
Best_Sat_Dist(11) = 626.e0 !And XV
Best_Sat_Dist(12) = 476.e0 !And XVI
Best_Sat_Dist(13) = 727.e0 !And XVII
Best_Sat_Dist(14) = 1214.e0 !And XVIII
Best_Sat_Dist(15) = 821.e0 !And XIX
Best_Sat_Dist(16) = 741.e0 !And XX
Best_Sat_Dist(17)=827.e0 !And XXI
Best_Sat_Dist(18) = 920.e0 !And XXII
Best_Sat_Dist (19) = 748.e0 !And XXIII
Best_Sat_Dist(20) = 898.e0 !And XXIV
Best_Sat_Dist(21) = 736.e0 !And XXV
Best_Sat_Dist(22) = 754.e0 !And XXVI
Best_Sat_Dist(23) = 1255.e0 !And XXVII
Best_Sat_Dist(24) = 681.e0 !And XXX
Best_Sat_Dist(25) = 712.e0 !NGC147
Best_Sat_Dist(26) = 620.e0 !NGC185
Best_Sat_Dist(27) = 820.e0 !M33
DO i = 1, nsats
    pos(1,i) = ABS(Best_Sat_Dist(i) * cos(theta(i)) * tan(xi(i))) !Determine length of x vector for each satellite
    IF (xi(i) . 1t. 0.e0) THEN
        pos(1,i)=-1.e0 * pos(1,i) !Determine if x is positive or negative
    END IF !
    pos(2,i) = ABS(Best_Sat_Dist(i) * sin(eta(i)))\quad !Determine length of y vector for each satellite
    IF (eta(i) .lt. 0.e0) THEN
        pos(2,i) = -1.e0 * pos(2,i) !Determine if y is positive or negative
    END IF !
    pos(3,i) = Best_Sat_Dist(i) * cos(theta(i)) - m31_dist ! Determine length and sign of z vector
END DO
CALL MaxSigFind
Actual_sig = max_plane_sig
Actual_bfv = best_fit_vect
alpha_set = - (90.e0 - 12.5e0) * (pi/180.e0) !Rotate to bring back out of M31's inclination !
gamma_set = + (90.e0-39.8e0)* (pi/180.e0) !angle and PA (i.e. to view from above the M31 pole) !
                                    ! Change
CALL Rotate
                                    !
                                    ! to
Actual_bfv = MATMUL(z_rot, Actual_bfv) !Convert vectors back to how they would appear! !
Actual_bfv = MATMUL(x_rot, Actual_bfv) !in M31 reference frame !M31
gamma_set = 90.e0*(pi/180.e0) ! !coordinate
CALL Rotatem!
Additional rotation in M31 galactic longitude
!system
CALL Rotate !Aditional rotation in M31 galactic longitude,
Actual_bfv = MATMUL(z_rot, Actual_bfv) !
!!/\Determine best fit plane and significance for satellite
!!| positions generated from best fit distances
!!|Determine best fit plane and significance for "err_samp" samples of
```

```
!!\/possible satellite positions generated from sampled distances and plot
DO it = 1, err_samp
    CALL random_number(randnum)
    m31_dist = M31_Dist_PPD(NINT(randnum)) !to generate err_samp
    !posible x,y,z coords
    DO i = 1, 27
        CALL random_number (randnum)
        randnum = randnum * 499999.e0 + 1.e0 !for each of the
        Sat_Dist_Drawn(i) = Sat_Dist(NINT(randnum),i) !satellites to
    END DO !generate err_samps
                !posible x,y,z coords
```

    DO i \(=1,27\)
        \(\operatorname{pos}(1, i)=\operatorname{ABS}\left(S a t \_D i s t \_D r a w n(i) * \cos (t h e t a(i)) * \tan (x i(i))\right)\) ! Determine length of \(x\) vector for each satellite
        IF (xi(i) . lt. 0.e0) THEN
            \(\operatorname{pos}(1, \mathrm{i})=-1 . e 0 * \operatorname{pos}(1, \mathrm{i}) \quad\) ! Determine if x is positive or negative
        END IF !
        \(\operatorname{pos}(2, i)=\operatorname{ABS}\left(S a t \_D i s t \_D r a w n(i) * \sin (e t a(i))\right) \quad\) Determine length of y vector for each satellite
        IF (eta(i) .lt. 0.e0) THEN
            \(\operatorname{pos}(2, i)=-1 . e 0 * \operatorname{pos}(2, i) \quad\) ! Determine if \(y\) is positive or negative
        END IF !
        \(\operatorname{pos}(3, \mathrm{i})=\) Sat_Dist_Drawn(i) \(* \cos (\) theta(i)) - m31_dist \(\quad\) Determine length and sign of \(z\) vector
    END DO
    CALL MaxSigFind
    alpha_set \(=-(90 . e 0-12.5 \mathrm{e} 0) *(\mathrm{pi} / 180 . \mathrm{e} 0)\) ! Rotate to bring back out of M31's inclination !
    gamma_set \(=+(90 . e 0-39.8 \mathrm{e} 0) *(\mathrm{pi} / 180 . \mathrm{e} 0)\) !angle and PA (i.e. to view from above the M31 pole) !
                                    ! Change
    CALL Rotate
                        !
    best_fit_vect \(=\) MATMUL(z_rot, best_fit_vect) ! Convert vectors back to how they would appear !
    best_fit_vect \(=\) MATMUL(x_rot, best_fit_vect) !in M31 reference frame !M31
    gamma_set \(=90 . \mathrm{e} 0 *(\mathrm{pi} / 180 . \mathrm{e} 0) \quad\) ! \(\quad\) !oordinate
    CALL Rotate ! Additional rotation in M31 galactic longitude !system
    best_fit_vect \(=\) MATMUL(z_rot, best_fit_vect) !
    CALL Theta_Phi (best_fit_vect(1), best_fit_vect(2), best_fit_vect (3))
    err_sig(it) = max_plane_sig
    WRITE (14, '(7F16.5)') REAL(it), err_sig(it), theta_coord, phi_coord, best_fit-vect(1), best_fit_vect(2), best_fit_vect(3)
    END DO



// TRIM(ADJUSTL(folder)) // '/err_samp_PPD_' // TRIM(ADJUSTL(subsize)) // , sats.jpg,
all system (command)

```
!!/\ Determine best fit plane and significance for "err_samp" samples of
!!| possible satellite positions generated from sampled distances and plot
!!|Determine best fit plane and significance for "nit"
!!\/random realizations of satellite positions and plot
min_signif = 1000.e0
DO it = 1, nit
    CALL RandomPoints
    CALL MaxSigFind
    IF (max_plane_sig . lt. min_signif) THEN
        min_signif = max_plane_sig
    END IF
    alpha_set = - (90.e0 - 12.5e0)* (pi/180.e0) !Rotate to bring back out of M31's inclination !
    gamma_set = + (90.e0 - 39.8e0)* (pi/180.e0) !angle and PA (i.e. to view from above the M31 pole) !
                                    !Change
    CALL Rotate
    \, !to
    best_fit_vect = MATMUL(z_rot, best_fit_vect) ! Convert vectors back to how they would appear !
    best_fit_vect = MATMUL(x_rot, best_fit_vect) !in M31 reference frame !M31
    gamma_set = 90.e0 *(pi/180.e0) ! !coordinate
```



```
    CALL Rotate !Additional rotation in M31 galactic longitude !system
    best_fit_vect = MATMUL(z_rot, best_fit_vect) !
    CALL Theta_Phi(best_fit_vect(1), best_fit_vect(2), best_fit_vect(3))
    sig(it) = max_plane_sig
    WRITE (13, '(7F16.5)') REAL(it), sig(it), theta_coord, phi_coord, best_fit_vect(1), best_fit_vect(2), best_fit_vect(3)
END DO
string2 = TRIM(ADJUSTL(folder)) // '/sig_PPD_` // TRIM(ADJUSTL(subsize)) // , _sats.ps/CPS
CALL HistoPlot(nit,101, sig,'LOG10(Minimum_RMS)','Probability,, TRIM(ADJUSTL(string2)), .true.)
WRITE (command,*) 'convert_-rotate_90_' // TRIM(ADJUSTL(folder)) // '/sig_PPD_' // TRIM(ADJUSTL(subsize)) // ' - sats.ps_, &
                            // TRIM(ADJUSTL(folder)) // '/sig_PPD_' // TRIM(ADJUSTL(subsize)) // , sats.jpg,
call system(command)
!!/\ Determine best fit plane and significance for "nit"
!!|random realizations of satellite positions and plot
END SUBROUTINE Significance
SUBROUTINE HistoPlot(nval, data_hist_bins, data, xlabel, ylabel, device, normalize)
IMPLICIT NONE
!******* Created 24 Feb 2012********
!
!INTEGER nval = number of data points in histogram
```

```
5 6 2
```

!INTEGER data_hist_bins = number of bins in histogram

```
!INTEGER data_hist_bins = number of bins in histogram
REAL data(nval) = The array containing the data
REAL data(nval) = The array containing the data
!CHARACTER xlabel = Label of x-axis of histogram
!CHARACTER xlabel = Label of x-axis of histogram
!CHARACTER ylabel = Label of y-axis of histogram
!CHARACTER ylabel = Label of y-axis of histogram
!CHARACTER device = The plotting device ('?' if unsure)
!CHARACTER device = The plotting device ('?' if unsure)
!LOGICAL normalize = ,true. if histogram is to be
!LOGICAL normalize = ,true. if histogram is to be
!normalized, else set to .false
!normalized, else set to .false
!
!
!Uses PGPLOT
!Uses PGPLOT
!
!
INTEGER :: data_hist_bins, nval, it_num
INTEGER :: data_hist_bins, nval, it_num
REAL :: bw, data(nval), data_hist(data_hist_bins, 2), data_min, data_max
REAL :: bw, data(nval), data_hist(data_hist_bins, 2), data_min, data_max
CHARACTER(LEN=*) :: xlabel, ylabel, device
CHARACTER(LEN=*) :: xlabel, ylabel, device
LOGICAL :: normalize
LOGICAL :: normalize
data_hist = 0.e0
data_hist = 0.e0
data_min = MINVAL(data) ; data_max = MAXVAL(data)
data_min = MINVAL(data) ; data_max = MAXVAL(data)
bw = (data_max - data_min)/(REAL(data_hist_bins) - 1.e0)
bw = (data_max - data_min)/(REAL(data_hist_bins) - 1.e0)
DO it_num = 1, data_hist_bins
DO it_num = 1, data_hist_bins
    data_hist(it_num,1) = data_min + REAL(it_num - 1) * bw
    data_hist(it_num,1) = data_min + REAL(it_num - 1) * bw
END DO
END DO
DO it_num = 1, nval
DO it_num = 1, nval
    data_hist(NINT((data(it_num) - data_min)/bw) + 1,2)=&
    data_hist(NINT((data(it_num) - data_min)/bw) + 1,2)=&
    data_hist(NINT((data(it_num) - data_min)/bw) + 1,2) + 1.e0
    data_hist(NINT((data(it_num) - data_min)/bw) + 1,2) + 1.e0
END DO
END DO
IF (normalize) THEN
IF (normalize) THEN
data_hist(:,2) = data_hist(:,2) / SUM(data_hist(:,2))
data_hist(:,2) = data_hist(:,2) / SUM(data_hist(:,2))
END IF
END IF
CALL pgbegin(0,TRIM(ADJUSTL(device)),1,1)
CALL pgbegin(0,TRIM(ADJUSTL(device)),1,1)
CALL pgenv(MINVAL(data, mask = data . ne. 0.), &
CALL pgenv(MINVAL(data, mask = data . ne. 0.), &
    MAXVAL(data, mask = data .ne. 0.), &
    MAXVAL(data, mask = data .ne. 0.), &
    0., 1.1*MAXVAL(data_hist (:, 2)), 0, 0)
    0., 1.1*MAXVAL(data_hist (:, 2)), 0, 0)
CALL pgbin (data_hist_bins, data_hist(:, 1), data_hist(:, 2), .true.)
CALL pgbin (data_hist_bins, data_hist(:, 1), data_hist(:, 2), .true.)
CALL pglab(TRIM(ADJUSTL(xlabel)), TRIM(ADJUSTL(ylabel)), ',')
CALL pglab(TRIM(ADJUSTL(xlabel)), TRIM(ADJUSTL(ylabel)), ',')
CALL pgend
CALL pgend
END SUBROUTINE HistoPlot
END SUBROUTINE HistoPlot
!logical function in_poly(x,y,np,xp,yp) omitted - see MF_TRGB.f95 in preceding appendix
!logical function in_poly(x,y,np,xp,yp) omitted - see MF_TRGB.f95 in preceding appendix
!real function fimag(x0, xs,xe,y0,ys,ye) omitted - see MF_TRGB.f95 in preceding appendix
!real function fimag(x0, xs,xe,y0,ys,ye) omitted - see MF_TRGB.f95 in preceding appendix
SUBROUTINE SampledDist !Read in samples from the distance distributions of
SUBROUTINE SampledDist !Read in samples from the distance distributions of
USE Global !M31 and its satellites
USE Global !M31 and its satellites
IMPLICIT NONE
IMPLICIT NONE
DOUBLE PRECISION :: sla_DSEP
```

DOUBLE PRECISION :: sla_DSEP

```
```

OPEN (unit = 11, file = ,./AndromedaIe/other_plots2/Sampled_MWy_Distances.dat,, status = 'old')
OPEN (unit = 12, file = './AndromedaIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 13, file =, ./AndromedaIIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 14, file = './AndromedaVe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 15, file ='./AndromedaIXe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN(unit = 16, file =,./AndromedaXe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 17, file = './AndromedaXIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN(unit = 18, file =,./AndromedaXIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 19, file = './AndromedaXIIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 20, file =,./AndromedaXIVe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 21, file =, ./AndromedaXVe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 22, file = './AndromedaXVIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 23, file = './AndromedaXVIIe/other_plots2/Sampled_MWy_Distances.dat', status ='old')
OPEN (unit = 24, file = './AndromedaXVIIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 25, file =,./AndromedaXIXe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN(unit = 26, file = './AndromedaXXe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 27, file =, ./AndromedaXXIe/other_plots2/Sampled_MWy_Distances.dat',, status = 'old')
OPEN (unit = 28, file =, //AndromedaXXIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 29, file = './AndromedaXXIIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 30, file ='./AndromedaXXIVe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 31, file = './AndromedaXXVe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 32, file = './AndromedaXXVIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 33, file =,./AndromedaXXVIIe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 34, file = './AndromedaXXXe/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 35, file =,./NGC147e_outer/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 36, file = './NGC185e_outer/other_plots2/Sampled_MWy_Distances.dat', status = 'old')
OPEN (unit = 37, file ='./M33e/other_plots2/Sampled_MWy_Distances.dat', status ='old')
OPEN (unit = 38, file = './M31e/other_plots/M31_Distance_PPD.dat', status = 'old')

```
\(\mathrm{i}=0\)
DO WHILE (.TRUE.)
\(\mathrm{i}=\mathrm{i}+1\)
IF (i . gt . 500000) THEN
    exit
END IF
READ (11, *, IOSTAT \(=\) ios) Sat_Dist (i, 1)
IF (ios == -1) THEN
    \(\mathrm{i}=\mathrm{i}-1\)
    exit
ELSE IF (ios .gt. 0) THEN
    WRITE (*,*) i
    \(\mathrm{i}=\mathrm{i}-1\)
    cycle
END IF
END DO
! Files 12 through 36 read in as shown for
! file 11 above and 37 below
```

i = 0
DO WHILE (.TRUE.)
i = i + 1
IF (i .gt. 500000) THEN
exit
END IF
READ (37, *, IOSTAT = ios) Sat_Dist(i, 27)
IF (ios == -1) THEN
i = i - 1
exit
ELSE IF (ios .gt. 0) THEN
WRITE (*,*) i
i=i-1
cycle
END IF
END DO
i = 0
DO while (.TRUE.)
i = i + 1
IF (i .gt. 3000000) THEN
exit
END IF
READ (38, *, IOSTAT = ios) M31_Dist_PPD(i)
IF (ios == -1) THEN
i = i - 1
exit
ELSE IF (ios .gt. 0) THEN
WRITE (*,*) i
i=i-1
cycle
END IF
END DO
!| Tangent Plane projection angles (xi, eta)
!\/for each satellite
Sat_Pos(1,:) = (/ 0.577417966865471, -3.2314283795568426 /)
Sat_Pos(2,:) = (/ 7.122225162668977, -7.592252808099334 /)
Sat_Pos(3,:) = (/ - 1.457960734448612, -4.765087242682244 /)
Sat_Pos(4,:) = (/ 4.67571438294161, 6.595921326738737 /)
Sat_Pos(5,:) = (/ 1.8486898643911536, 1.9594865642747519 /)
Sat_Pos(6,:) = (/ 4.243395063076322, 3.7004040941268976 /)
Sat_Pos(7,:) = (/ 0.7517384673615299, -7.5056115753940515 /)
Sat_Pos (8,:) = (/ 0.979245230608749, -6.921767335891222 /)
Sat_Pos(9,:) = (/ 1.9303864363068866, -8.301405961956007 /)

```
```

Sat_Pos(10,:) = (/ 1.9616750556905713, -11.722713906650105 /)
Sat_Pos(11,:) = (/ 6.234635220222043, -2.8843033494675128 /)
Sat_Pos(12,:) = (/ 3.585309428238602, -8.89395559219651 /)
Sat_Pos(13,:) = (/ -1.0085393602174035, 3.0650628218938585 /)
Sat_Pos(14,:) = (/ -7.181147930491751, 4.280747404302224 /)
Sat_Pos(15,:) = (/ -4.784510400133266, -6.109702992340868 /)
Sat_Pos(16,:) = (/ -7.269591542188213, -5.840766998597464 /)
Sat_Pos(17,:) = (/ -8.888840848373246, 1.8332101560489324 /)
Sat_Pos(18,:) = (/ 10.246238044162409, -12.92831734742824 /)
Sat_Pos(19,:) = (/ 9.147466892705275, - 1.961574445626079 /)
Sat_Pos (20,:) = (/ 6.211342279324403, 5.4699308688721202 /)
Sat_Pos(21,:) = (/ -2.1573196501753253, 5.648080455771416 /)
Sat_Pos(22,:) = (/ -3.203360921263025, 6.772438490131299 /)
Sat_Pos(23,:) = (/ -0.9036995038298262, 4.117443311518852 /)
Sat_Pos (24,:) = (/ - 1.0080494766702268, 8.448293881357133 /)
Sat_Pos(25,:) = (/ - 1.5924793331852394, 7.303792725460755 /)
Sat_Pos(26,:) = (/ -0.6317798640615756, 7.108624424318093 /)
Sat_Pos(27,:) = (/ 11.273546933713943, -10.076848401834724 /)
!/\
! 11
DO i = 1, nsats
xi(i) = Sat_Pos(i,1)
eta(i) = Sat_Pos(i,2)
xi(i)=xi(i)*(pi/180.e0) !Convert angles from
eta(i)= eta(i) *(pi/180.e0) !degrees to radians
END DO
DO i = 1, nsats
xi_dble = xi(i); eta_dble = eta(i)
CALL sla_DTP2S(xi_dble, eta_dble, 0.d0, 0.d0, RA, DEC) !Convert tangent plane
IF (xi_dble .lt. 0.d0) then !projection angles into
RA = RA - (2.e0 * pi) !their true angles using
END IF !sla_DTP2S
xi(i) = RA
eta(i) = DEC
END DO
DO i = 1, nsats !Find the true angle
xi_dble = xi(i) !theta - the angle on
eta_dble = eta(i) !the sky between M31
theta(i) = sla_DSEP(0.d0, 0.d0, xi_dble, eta_dble) !and the object
END DO !(uses sla_DSEP)
END SUBROUTINE SampledDist
SUBROUTINE BorderGet ! Read in the }134\mathrm{ points in xi and eta defining the PAndAS
USE Global !Survey Border. These points are used to reject satellites
IMPLICIT NONE !that fall out of bounds (see RandomPoints Subroutine)
OPEN (unit = 40, file = '../SurveyArea/Border_Coords_XiEta.dat', status = 'old')
DO i = 1, 134
READ (40, *, IOSTAT = ios) SAP_xi(i), SAP_eta(i)
END DO
END SUBROUTINE BorderGet

```

\section*{Program: Alternative Plane Fitting Code Segments}

Creation Date: First versions Feb 2012
Relevant Section: Ch. 5; Paper III §3.1, §3.2
Notes: Presented here are four separate code segments, each one performing the calculation of the goodness of fit of a tested plane. The first uses the root-mean-square (RMS) of the perpendicular distances of the satellites from the plane. This is the one used in the 'MaxSigFind' subroutine presented in PlaneS igRMS.f95 (p. 264). The other code segments are alternatives to this RMS code segment. The second code segment calculates the goodness of fit of a given plane by summing the absolute values of the perpendicular distances of each satellite from the plane. The third uses a maximum likelihood approach and replaces the zero-thickness plane with a Gaussian distribution of some (to be determined) thickness. The fourth and final code segment serves a different purpose to the previous three in that it finds the plane of maximum asymmetry. It seeks the plane which can divide the sample most unequally. Note that some other minor modifications to the code of PlaneSigRMS.f95 would be required for correct operation. These segments are intended to illustrate precisely how the various forms of plane fitting utilized in Paper III are implemented.

```

!As used in the included version of the 'MaxSigFind, subroutine in 'PlaneSigRMS.f95
!Where an alternate plane fitting statistic is used, this segment of code should be
!replaced with one of the versions below for each of the four times it appears in the
!'MaxSigFind, subroutine. Note that other minor code variations are necessary
!but these are included to show the way the actual plane fitting statistic is
!calculated in each case
plane_sig = 0.d0
ms = 0.d0
DO k = 1, nsats !RMS Calculation
planeDist = norm(1)*\operatorname{pos}(1,k) + norm(2)*pos(2,k) + norm(3)*\operatorname{pos}(3,k)
rms = rms + (planeDist)**2
END DO
rms = SQRT(rms/nsats)
plane_sig = LOG10(rms)
IF (plane_sig .lt. max_plane_sig) THEN !Most significant plane has lowest rms
max_plane_sig = plane_sig !Store approx, low resolution values
best_fit_vect = norm !of best fit pole and significance
pole_alpha = alpha_set !Store best fit pole for
pole_beta = beta_set !high resolution search
END IF
!-------For Fitting using Sum of Absolute Values of Satellite Distances from Plane_--------
plane_sig = 0.d0
ab_val = 0.d0
DO k = 1, nsats !Absolute Value of distance sum Calculation

```
```

    planeDist = abs(norm(1)*\operatorname{pos}(1,k) + norm(2)*\operatorname{pos}(2,k) + norm(3)*pos(3,k))
    ab_val = ab_val + planeDist
    END DO
    plane_sig = LOG10(ab_val)
    IF (plane_sig .lt. max_plane_sig) THEN !Most significant plane has lowest AbVal
        max_plane_sig = plane_sig !Store approx, low resolution values
        best_fit_vect = norm !of best fit pole and significance
        pole_alpha = alpha_set !Store best fit pole for
        pole_beta = beta_set !high resolution search
    END IF
    !------------------------------------------
!Replace max_plane_sig = 9999999.e0 with max_plane_sig = -9999999.e0 as initial value
DO s = 1, 30
sigma = REAL}(\textrm{s})*5.d
plane_sig = 0.d0
DO k = 1, nsats
planeDist = abs(norm(1)*\operatorname{pos(1,k) + norm(2)*pos(2,k) + norm(3)*pos(3,k))}
like = exp(-(planeDist **2.d0)/(2.d0 * sigma ** 2.d0))/ (sigma * SQRT(2.d0 * pi))
IF (LOG10(like) .le. -9999.d0) THEN
plane_sig = plane_sig - 9999.d0
ELSE
plane_sig = plane_sig + LOG10(like)
END IF
END DO
IF (plane_sig .gt. max_plane_sig) THEN !Significance Calculation
max_plane_sig = plane_sig !Store approx, low resolution values
best_fit_vect = norm !of best fit pole, significance
best_fit_sigma = sigma !and Gaussian one sigma
pole_alpha = alpha_set !Store best fit pole for
pole_beta = beta_set !high resolution search
END IF
END DO
!Replace max_plane_sig = 9999999.e0 with max_plane_asymm = 0.e0 as initial value
pos.side = 0.e0
DO k = 1, nsats !Count satellites on
planeDist = norm(1)*pos(1,k) + norm(2)*pos(2,k) + norm(3)*pos(3,k) !one side of plane.
IF (planeDist .ge. 0.d0) THEN !Satellites on the
pos_side = pos_side + 1.e0 !other side of the
END IF !plane is known
END DO !automatically from
neg_side = nsats - pos_side !total number of sats
IF (pos_side .gt. neg_side) THEN
plane_asymm = pos_side !Calculate asymmetry, defined as
ELSE !the number of satellites on the side
plane_asymm = neg_side !with the most satellites
END IF
IF (plane_asymm .gt . max_plane_asymm) THEN
max_plane_asymm = plane_asymm !If a higher asymmetry plane is
max_asymm_vect = norm !found, note normal vector of that
ma_pos_side = pos_side !plane as well as the satellite
ma_neg_side = neg_side !counts on each side.
pole_alpha = alpha_set !Store highest asymmetry pole for
pole_beta = beta_set !high resolution search
END IF

```

Program: Subroutines for Processing Satellite Subsets
Creation Date: 18 July 2012 (first version 26 Apr 2012) Many modifications.
Relevant Section: Ch. 5; Paper III §3.3
Notes: The three subroutines presented here essentially modify PlaneS igRMS.f95 (p. 264) so that it can process every possible combination of a given number of satellites rather than just the full sample. The 'Combinations' subroutine steps through every possible combination of the specified size (sizes of 3 through 7 satellites are shown) and for each one calls the 'Significance’ subroutine which samples positions for each satellite in the combination and then calls 'MaxSigFind' to perform the plane fitting. Note that each possible combination of satellites is sampled 'err_samps' (currently set to 100 as used in \(\S 3.3\) of Paper III) times so as to account for the uncertainties in the satellite distances. Also, this code compresses storage file size by indexing each possible pole position and then recording the number of instances of that pole as well as the number of times each satellite contributes to that pole (information which is used by pole_vicinity_counts_satid_w.f95-p. 294).
```

!Code Segments for testing *all* combinations of a particular number (nsatsub)
!of satellites possible from the total sample (total sample is 25 satellites here
!as NGC147/NGC185/AndXXX are treated as a single point).
!See 'PlaneSigRMS.f95, for all subroutines called that are not incuded
SUBROUTINE Combinations !Finds the best fit plane to every possible combination of 'nsatsub,
USE Global !satellites. The pole of each combination's best fit plane is converted
IMPLICIT NONE !to M31-centric lat. and long. and stored for plotting as a pole plot
!map on an aitoff-hammer projection
subsetcounts = 0
RMSmin = 9999999.e0
IF (nsatsub .eq. 3) THEN
DO s1 = 1, nsats -2
DO s2 = s1+1, nsats -1
DO s3 = s2+1, nsats
satholder(1) = s1
satholder(2) = s2
satholder(3) = s3
subsetcounts = subsetcounts + 1
CALL Significance
write(13) Actual_sig, theta_coord, phi_coord, Actual_bfv(1), Actual_bfv(2), Actual_bfv(3),\&
s1, s2, s3
END DO
END DO
END DO
ELSE IF (nsatsub .eq. 4) THEN

```
```

DO s1 = 1, nsats - 3
DO s2 = s1+1, nsats -2
DO s3 = s2+1, nsats -1
DO s4 = s3+1, nsats
satholder(1) = s1
satholder(2) = s2
atholder(3) = s3
satholder(4) = s4
subsetcounts = subsetcounts + 1
CALL Significance
write(13) Actual_sig, theta_coord, phi_coord, Actual_bfv(1), Actual_bfv(2), Actual_bfv(3), \&
s1, s2, s3, s4
END DO
END DO
END DO
END DO
ELSE IF (nsatsub .eq. 5) THEN
DO s1 = 1, nsats -4
DO s2 = s1+1, nsats -3
DO s3 = s2+1, nsats -2
DO s4 = s3+1, nsats -1
DO s5 = s4+1, nsats
satholder(1) = s1
satholder(2) = s2
satholder(3) = s3
satholder(4) = s4
satholder(5) = s5
subsetcounts = subsetcounts + 1
CALL Significance
write(13) Actual_sig, theta_coord, phi_coord, Actual_bfv(1), Actual_bfv(2), Actual_bfv(3),\&
s1, s2, s3, s4, s5
END DO
END DO
END DO
END DO
END DO
ELSE IF (nsatsub .eq. 6) THEN
DO s1 = 1, nsats -5
DO s2 = s1+1, nsats -4
DO s3 = s2+1, nsats - 3
DO s4 = s3+1, nsats -2
DO s5 = s4+1, nsats -1
DO s6 = s5+1, nsats
satholder(1) = sl
satholder(2) = s2
satholder(3) = s3
satholder(4) = s4
atholder(5) = s5
satholder(6) = s6
subsetcounts = subsetcounts + 1
CALL Significance
write(13) Actual_sig, theta_coord, phi_coord, Actual_bfv(1), Actual_bfv(2), Actual_bfv(3), \&

```
```

95
END DO
END DO
END DO
END DO
END DO
END DO
ELSE IF (nsatsub .eq. 7) THEN
DO s1 = 1, nsats -6
DO s2 = s1+1, nsats -5
DO s3 = s2+1, nsats -4
DO s4 = s3+1, nsats -3
DO s5 = s4+1, nsats -2
DO s6 = s5+1, nsats -1
DO s7 = s6+1, nsats
satholder(1) = s1
satholder(2) = s2
satholder(3) = s3
satholder(4) = s4
satholder(5) = s5
satholder(6) = s6
satholder(7) = s7
subsetcounts = subsetcounts + 1
CALL Significance
write(13) Actual_sig, theta_coord, phi_coord, Actual_bfv(1), Actual_bfv(2), Actual_bfv(3), \&
s1, s2, s3, s4, s5, s6, s7
END DO
END DO
END DO
END DO
END DO
END DO
END DO
__________
END IF
write(11,*) "Results:"
write(11,*) "Total_number_of_combinations_of", nsatsub, "satellites:", subsetcounts
write(11,*) "Best_satellite_combination:"
DO i = 1, nsatsub
write(11, *) best_sat_combo(i)
END DO

```

```

write(11,*) best_sat_bfv(1), best_sat_bfv(2), best_sat_bfv(3)
CALL Theta_Phi(best_sat_bfv(1), best_sat_bfv(2), best_sat_bfv(3))

```

```

write(11,*) "Theta\iota=", theta_coord, ";\&Phi\iota=", phi_coord
write(11,*) "LOG10(RMS)_of_best_fit_plane_for_this__combination:", RMSmin
mode_counts = 0.e0 !| Every pole position possible is given an index and the number of times
DO i = 1, 31
!|a pole is recorded at that position is recorded. This greatly reduces
DO j = 1, 120 !|file storage size. The number of times a particular satellite contributes
DO k = 1, 15 !\/to a pole at each possible position is also recorded.
DO 1 = 1, 30
IF (poles_per_pos(i,j,k,1,6) .ne. 0.e0) THEN

```
```

    WRITE (17, '(31F11.5)') poles_per_pos(i,j,k,l,1), poles_per_pos(i,j,k,l,2), poles_per_pos(i,j,k,l,3), &
                poles_per_pos(i,j,k,l,4), poles_per_pos(i,j,k,l,5), poles_per_pos(i,j,k,l,6),&
                    poles_per_pos(i,j,k,l,7), poles_per_pos(i,j,k,l,8), poles_per_pos(i,j,k,l,9),&
                    poles_per_pos(i,j,k,l,10), poles_per_pos(i,j,k,l,11), poles_per_pos(i,j,k,l,12), &
                    poles_per_pos(i,j,k,l,13), poles_per_pos(i,j,k,l,14), poles_per_pos(i,j,k,l,15), &
                    poles_per_pos(i,j,k,l,16), poles_per_pos(i,j,k,l,17), poles_per_pos(i,j,k,l,18), &
                            poles_per_pos(i,j,k,l,19), poles_per_pos(i,j,k,l,20), poles_per_pos(i,j,k,l,21), &
                    poles_per_pos(i,j,k,l,22), poles_per_pos(i,j,k,1,23), poles_per_pos(i,j,k,1,24), &
                    poles_per_pos(i,j,k,1,25), poles_per_pos(i,j,k,l,26), poles_per_pos(i,j,k,l,27), &
                    poles_per_pos(i,j,k,l,28), poles_per_pos(i,j,k,l,29), poles_per_pos(i,j,k,l,30),&
                        poles_per_pos(i,j,k,l,31)
    END IF
    IF (poles_per_pos(i,j,k,l,6) .gt. mode_counts) THEN
    mode_counts = poles_per_pos(i,j,k,l,6)
    pos_mpc(1) = i ; pos_mpc (2) = j ; pos_mpc(3) = k ; pos_mpc (4) = l
    END IF
END DO
END DO
END DO
END DO
x_mode = poles_per_pos(pos_mpc(1),pos_mpc(2),pos_mpc(3),pos_mpc(4),1) !Most freq. normal vector x
y_mode = poles_per_pos(pos_mpc(1),pos_mpc(2),pos_mpc(3),pos_mpc(4),2) !Most freq. normal vector y
z_mode = poles_per_pos(pos_mpc(1),pos_mpc(2),pos_mpc(3),pos_mpc(4),3) !Most freq. normal vector z
theta_mode = poles_per_pos(pos_mpc(1),pos_mpc(2),pos_mpc(3),pos_mpc(4),4) !Most freq. pole theta
phi_mode = poles_per_pos(pos_mpc(1),pos_mpc(2),pos_mpc(3),pos_mpc(4),5) !Most freq. pole phi
write(11,*) "Normal\_vector „of \&most\_frequently encountered_plane:"
write(11,*) x_mode, y_mode, z_mode
write(11,*) "Theta_and_phi_of_most_frequently_encountered_pole:"
write(11,*) "Theta\_=", theta_mode, ";\lrcornerPhi\iota=", phi_mode
write(11,*) "Number_of_instances_of_this_pole:", mode_counts
END SUBROUTINE Combinations
SUBROUTINE Significance !Finds RMS and pole of best fit plane to a given satellite combination
USE Global !Does this for 'err_samps, possible versions of the combination
IMPLICIT NONE !using distances drawn from the respective satellite distance PPDs.
WRITE(16,*) "Combinations_tested_so_far:", subsetcounts ! Progress update
DO samp_it = 1, err_samps
CALL random_number(randnum) !Read one possible M31
randnum = randnum * 2999999.e0 + 1.e0 !distance to generate
m31_1K_dist = M31_Dist_PPD (NINT(randnum)) !one possible set of
!x,y,z coords
DO i = 1, nsatsub !Read in one possible
IF (satholder(i) .1t. 24) THEN ! distance for each of the
CALL random_number(randnum) !dwarf sph sats except
randnum = randnum * 499999.e0 + 1.e0 !AND XXX to generate one
sat_1K_dist(satholder(i)) = Sat_Dist(NINT(randnum), satholder(i))!set of possible distances
ELSE IF (satholder(i) .eq. 24) THEN
DO j = 25, 26
!Get possible distances
CALL random_number(randnum) !for NGC147 and NGC185
randnum = randnum * 499999.e0 + 1.e0 !to combine into one point
sat_1K_dist(j) = Sat_Dist(NINT(randnum),j) !to represent the NGC147,
END DO !NGC185, AND XXX group

```
```

    ELSE IF (satholder(i) .eq. 25) THEN
        CALL random_number(randnum)
        randnum = randnum * 499999.e0 + 1.e0
    sat_1K_dist(27) = Sat_Dist(NINT(randnum),27)
        !Get a possible
        !distance for
        !M33
    END IF
    END DO
DO i = 1, nsatsub !Convert distances to 3D positions for:
IF (satholder(i) .lt. 24) THEN !A: All the dwarf spheroidal satellites except Andromeda XXX
pos(1, satholder(i)) = ABS(sat_1K_dist(satholder(i)) * cos(theta(satholder(i))) * tan(xi(satholder(i))))!Determine length of x
IF (xi(satholder(i)).lt. 0.e0) THEN !vector for each satellite
pos(1,satholder(i)) = -1.e0 * pos(1,satholder(i)) !Determine if x is positive or negative
END IF
pos(2,satholder(i)) = ABS(sat_1K_dist(satholder(i)) * sin(eta(satholder(i)))) !Determine length of y vector for each satellite
IF (eta(satholder(i)) .lt. 0.e0) THEN
pos(2,satholder(i)) = -1.e0 * pos(2,satholder(i)) !Determine if y is positive or negative
END IF
pos(3,satholder(i)) = sat_1K_dist(satholder(i)) * cos(theta(satholder(i))) - m31_1K_dist!Determine length and sign of z vector
ELSE IF (satholder(i) .eq. 24) THEN !B: The NGC147/NGC185/AND XXX subgroup
DO j = 25, 26
pos(1,j) = ABS(sat_1K_dist(j) * cos(theta(j)) * tan(xi(j)))!Determine length of x vector for each
IF (xi(j) .lt. 0.e0) THEN !satellite vector for each satellite
pos(1,j)=-1.e0* pos(1,j) !Determine if x is positive or negative
END IF
pos(2,j) = ABS(sat_1K_dist(j) * sin(eta(j))) !Determine length of y vector for each satellite
IF (eta (j) . lt. 0.e0) THEN
pos(2,j)=-1.e0 * pos(2,j) !Determine if y is positive or negative
END IF !
pos(3,j) = sat_1K_dist(j) * cos(theta(j)) - m31_1K_dist !Determine length and sign of z vector
END DO
pos(:,24) = pos(:,25) + ((100.e0) **(0.2 e0*0.2e0)) * pos(:,26)
pos(:,24)}=\operatorname{pos}(:,24)/(1.e0 + (100.e0)**(0.2 e0*0.2 e0)
ELSE IF (satholder(i) .eq. 25) THEN !C: M33
pos(1,25) = ABS(sat_1K_dist(27) * cos(theta(27)) * tan(xi(27)))!Determine length of x vector for each
IF (xi(27) .lt. 0.e0) THEN !satellite vector for each satellite
pos(1,25) = -1.e0 * pos(1,25) !Determine if x is positive or negative
END IF
pos(2,25) = ABS(sat_1K_dist(27) * sin(eta(27))) !Determine length of y vector for each satellite
IF (eta(27) .1t. 0.e0) THEN !
pos(2,25) = -1.e0 * pos(2,25) !Determine if y is positive or negative
END IF
!
pos(3,25) = sat_1K_dist(27)* cos(theta(27)) - m31_1K_dist !Determine length and sign of z vector
END IF
END DO
CALL MaxSigFind

```
```

Actual_sig = max_plane_sig
Actual_bfv = best_fit_vect
alpha_set = - (90.e0 - 12.5e0) * (pi/180.e0) !Rotate to bring back out of M31's inclination !
gamma_set = + (90.e0 - 39.8e0) * (pi/180.e0) !angle and PA (i.e. to view from above the M31 pole) !
CALL Rotate !
Actual_bfv = MATMUL(z_rot, Actual_bfv) ! Convert vectors back to how they would appear
Actual_bfv = MATMUL(x_rot, Actual_bfv) !in M31 reference frame !M31
gamma_set = 90.e0 * (pi/180.e0) ! !coordinate
CALL Rotate !Additional rotation in M31 galactic longitude !system
Actual_bfv = MATMUL(z_rot, Actual_bfv) !
CALL Theta_Phi(Actual_bfv(1), Actual_bfv(2), Actual_bfv(3))
!|Every pole position possible is given an index and the number of times
!|a pole is recorded at that position is recorded. This greatly reduces
!|file storage size. The number of times a particular satellite contributes
!\/to a pole at each possible position is also recorded.
poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),1) = Actual_bfv(1) !
poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),2) = Actual_bfv(2) !Update
poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),3) = Actual_bfv(3) !counts
poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),4) = theta_coord !at a
poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),5) = phi_coord ! particular

```

```

poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),6) +(1.e0/(10.e0 ** Actual_sig))!
DO i = 1, nsatsub
poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),6+satholder(i)) = \&
poles_per_pos(best_pol_loc(1), best_pol_loc(2), best_pol_loc(3), best_pol_loc(4),6+satholder(i)) + (1.e0/(10.e0 ** Actual_sig))
END DO
IF (Actual_sig . lt. RMSmin) THEN
RMSmin = Actual_sig !
DO i = 1, nsatsub !Store best possible satellite
best_sat_combo(i) = satholder(i) !combination encountered so far
END DO
best_sat_bfv = Actual_bfy
!(i.e. the combination within lowest
RMS of its best fit plane)
END IF
END DO
END SUBROUTINE Significance
SUBROUTINE MaxSigFind ! Finds best fit plane for a satellite distribution by testing goodness of fit of each
USE Global
IMPLICIT NONE
tested plane. The poles of the tested planes are all approximately equi-distant, taking
!into account the surface area of a shere as a function of latitude.
!A low resolution run finds the approximate location of the best fit plane's pole and then
!poles around this point are searched at higher resolution.
par_like = 0.e0
max_plane_sig = 9999999.e0
!| Low resolution
!\/ plane tests
DO i = 1, 30

```
```

    beta_set = REAL(i*3) * (pi/180.e0)
    DO j = 1, NINT(120.e0 * cos(beta_set)) !The higher the latitude, the smaller the
                    !number of points
    alpha_set = (REAL(j)/NINT(120.e0 * cos(beta_set))) * 360.e0 * (pi/180.e0)
    norm = (/ 0.e0, 0.e0, 1.e0 /)
    CALL Rotate
    norm = MATMUL(y_rot, norm)
    norm = MATMUL(x_rot, norm)
    plane_sig = 0.d0
    rms = 0.d0
    DO k = 1, nsatsub !RMS calculation
        planeDist = norm(1)*pos(1,satholder(k)) + norm(2)*pos(2,satholder(k)) + norm(3)*pos(3,satholder(k))
        rms = rms + (planeDist)**2
    END DO
    rms = SQRT(rms/nsatsub)
    plane_sig = LOG10(rms)
    IF (plane_sig .lt. max_plane_sig) THEN !Most significant plane has lowest rms
        max_plane_sig = plane_sig !Store approx, low resolution values
        best_fit_vect = norm !of best fit pole and significance
        pole_alpha = alpha_set !Store best fit pole for
        pole_beta = beta_set !high resolution search
        best_pol_loc(1) = i !Used for cumulative
        best_pol_loc(2) = j !pole count
    END IF
    END DO
    END DO
norm = (/ -1.e0, 0.e0, 0.e0 /) !Test at the actual pole (not included in above loop)
plane_sig = 0.do
rms = 0.d0
DO k = 1, nsatsub !RMS calculation
planeDist = norm(1)*pos(1, satholder(k)) + norm(2)*pos(2, satholder(k)) + norm(3)*pos(3, satholder(k))
rms = rms + (planeDist )**2
END DO
rms = SQRT(rms/nsatsub)
plane_sig = LOG10(rms)
!| High resolution search
!\/ around best fit pole
IF (plane_sig .lt. max_plane_sig) THEN !Condition not met unless the RMS at the actual pole
!was better than anywhere else in the low res search
best_pol_loc(1) = 31 !Used for cumulative
best_pol_loc(2) = 1 !pole count
max_plane_sig = plane_sig
best_fit_vect = norm
DO i = 1, 15
beta_set = (88.5e0 + (REAL(i)/10.e0)) * (pi/180.e0)
DO j = 1, NINT(1200.e0 * cos(beta_set)) !The higher the latitude, the smaller the
number of points
alpha_set =(\operatorname{REAL}(j)/NINT(1200.e0* cos(beta_set))) * 360.e0*(pi/180.e0)
norm = (/ 0.e0, 0.e0, 1.e0 /)

```

CALL Rotate
norm \(=\) MATMUL ( y_rot, norm \()\)
norm \(=\) MATMUL( \(\mathrm{x} \_\)_rot, norm \()\)
plane_sig \(=0 . d 0\)
\(\mathrm{rms}=0 . \mathrm{d} 0\)
DO \(\mathrm{k}=1\), nsatsub ! RMS calculation
planeDist \(=\operatorname{norm}(1) * \operatorname{pos}(1\), satholder \((k))+\operatorname{norm}(2) * \operatorname{pos}(2, \operatorname{satholder}(k))+\operatorname{norm}(3) * \operatorname{pos}(3, \operatorname{satholder}(k))\)
rms \(=r m s+(\) planeDist \() * * 2\)
END DO
rms \(=\operatorname{SQRT}(\mathrm{rms} /\) nsatsub \()\)
plane_sig \(=\) LOG10(rms)
IF (plane_sig .lt. max_plane_sig) THEN ! Most significant plane has lowest rms
max_plane_sig \(=\) plane_sig ! Store final, high resolution values
best_fit_vect \(=\) norm \(\quad\) of best fit pole and significance
best_pol_loc (3) \(=\mathrm{i}\) ! Used for cumulative
best_pol_loc (4) \(=\mathrm{j} \quad\) ! pole count
END IF
END DO
END DO

ELSE

DO i \(=1,11\)
DO \(\mathrm{j}=1,11\)
beta_set \(=\) pole_beta \(+2 . \mathrm{e} 0 * \operatorname{REAL}(\mathrm{j}-6) *(0.15 \mathrm{e} 0) *(\mathrm{pi} / 180 . \mathrm{e} 0)\)
alpha_set \(=\) pole_alpha \(+2 . e 0 * \operatorname{REAL}(i-6) *(0.15 e 0) *(p i / 180 . e 0) *\left(1 . e 0 / \cos \left(b e t a \_s e t\right)\right)\)
norm \(=(/ 0 . e 0,0 . e 0,1 . e 0 /)\)

CALL Rotate
norm \(=\) MATMUL ( y _rot, norm \()\)
norm \(=\) MATMUL( \(\mathrm{x} \_\)rot, norm \()\)
plane_sig \(=0 . d 0\)
\(\mathrm{rms}=0 . \mathrm{d} 0\)
DO \(\mathrm{k}=1\), nsatsub !RMS calculation
planeDist \(=\operatorname{norm}(1) * \operatorname{pos}(1, \operatorname{satholder}(\mathrm{k}))+\operatorname{norm}(2) * \operatorname{pos}(2, \operatorname{satholder}(\mathrm{k}))+\operatorname{norm}(3) * \operatorname{pos}(3, \operatorname{satholder}(\mathrm{k}))\) rms \(=r m s+(\) planeDist \() * * 2\)
END DO
\(\mathrm{rms}=\operatorname{SQRT}(\mathrm{rms} / \mathrm{nsatsub})\)
plane_sig \(=\operatorname{LOG10(rms)}\)
IF (plane_sig .lt. max_plane_sig) THEN ! Most significant plane has lowest rms
max_plane_sig \(=\) plane_sig ! Store final, high resolution values
best_fit_vect \(=\) norm \(\quad\) ! of best fit pole and significance
best_pol_loc (3) \(=\mathrm{i}\) ! Used for cumulative
best_pol_loc(4) \(=\mathrm{j} \quad\) ! pole count
END IF
END DO
END DO

END IF

END SUBROUTINE MaxSigFind

\section*{Program: PlaneSigSubSets_RandReal4_noGroup.f95}

Creation Date: 3 Oct 2012 (first version 26 Apr 2012) Many modifications.
Relevant Section: Ch. 5; Paper III §3.1, §3.2, §3.4
Notes: This program is designed specifically for finding the most planar combination of large subsets of satellites. It can only be used where we do not require a measurement for every possible subset (i.e. a pole distribution map). In this program, the 'MaxSigFind' subroutine is completely different to the version seen in PlaneSigRMS.f95 (p. 264). It throws down 10,000 random planes and finds the closest 'nsatsub' ( 15 in this case) satellites to the tested plane out of the full sample (nsats \(=27\) ) and records the associated RMS. That combination which is fit with the lowest RMS is then taken to approximate the most planar sub set. Note that the 'RandomPoints' subroutine presented here is also substantially different to that in PlaneS igRMS.f95 as it represents each satellite by a distance distribution containing 1,000 possible positions along the line of sight from Earth.
```

MODULE Global ! Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, j, k, l, s, mm, ios, idum = -9999, it, nit
INTEGER :: ndata_max, nsats, nsatsub, subsetcounts
PARAMETER (ndata_max = 10000000)
PARAMETER (nsats = 27)
PARAMETER (nit = 10000)
PARAMETER (nsatsub = 15)
REAL*8 :: pi
PARAMETER (pi = ACOS(-1.e0))
REAL :: randnum, sig(nit), norm(3), best_fit_vect(3), best_fit_sigma
REAL :: pos(3, ndata_max), temp_pos(3)
REAL :: a(ndata_max), b(ndata_max), c(ndata_max), d(ndata_max)
REAL :: a_hist (201,2), b_hist(201,2), c_hist (201,2), d_hist(2001,2)
REAL :: logL, LikeA, LikeB, r, p(4), p_temp(4), min_sigma, max_sigma
REAL*8 :: sigma, planeDist, like, plane_sig, rms, min_rms, rms_average(nit)
REAL :: max_plane_sig, RMSmin, Actual_sig, Actual_bfv(3), best_sat_bfv(3), Actual_bfs
INTEGER :: dummy, sat_pick (27)
REAL :: Sat_Dist(500000,nsats), Sat_Pos(nsats,2), xi(nsats), eta(nsats), theta(nsats), M31_Dist_PPD(3000000)
REAL :: Sat_Dist_change(nsats), new_Earth_Dist, Sat_Dist_store
REAL :: art_xi(nsats), art_eta(nsats), art_theta(nsats)
REAL*8 :: RA, DEC, xi_dble, eta_dble
REAL :: xi_test, eta_test, theta_test, SAP_xi(134), SAP_eta(134), spotR
REAL :: Best_Sat_Dist(nsats)
REAL :: m31_dist
REAL :: alpha_set, beta_set, gamma_set, pole_alpha, pole_beta
REAL :: x_rot (3,3), y_rot (3,3), z_rot (3,3)
REAL :: par_like(180,6)
REAL :: theta_coord, phi_coord
REAL :: poles_per_pos(31,120,15,30,6) = 0.e0
INTEGER :: best_pol_loc(4)
INTEGER :: s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12, s13
INTEGER :: s 14, s 15, s16, s17, s18, s19, s20, s21, s22, s23, s24, s25, s26
INTEGER :: satholder(nsats), best_sat_combo(nsats), pos_mpc(4)

```
```

REAL :: x_mode, y_mode, z_mode, theta_mode, phi_mode, mode_counts
CHARACTER :: argv*30, folder * 100, string*200, string2*200, command *200, subsize *3, itnum *5
LOGICAL :: new_sats
REAL :: closest_sats(nsatsub)
INTEGER :: closest_sats_id(nsatsub), best_sats(nsatsub), u
END MODULE Global
PROGRAM PlaneSignificance !Master program
USE Global
IMPLICIT NONE
WRITE (subsize,*) nsatsub
WRITE (folder,*) 'Plane_Stats_' // TRIM(ADJUSTL(subsize)) // ' _sats_RandReal_weighted'!Create
WRITE (string,*) ,./' // TRIM(ADJUSTL(folder)) !primary
!output
WRITE (command,*) 'mkdir_` // TRIM(ADJUSTL(folder)) !directory CALL system(command) CALL random_seed !Insure random seed for random numbers CALL SampledDist !Get sampled CALL FixedDist !satellite distances CALL BorderGet !Get PAndAS survey boundary points string2 = TRIM(ADJUSTL(folder)) // ,/sat_pos.dat, !Positions of satellites in OPEN(12, file=TRIM(ADJUSTL(string2)), status = 'unknown') !Random Realizations string2 = TRIM(ADJUSTL(folder)) // '/RMS_' // TRIM(ADJUSTL(subsize)) // ' -sats.dat' !Best Plane RMS OPEN(13, file=TRIM(ADJUSTL(string2)), status = 'unknown')}\quad\mathrm{ !output file rms_average = 0.e0 ! !Principal loop     CALL RandomPoints !random realization     rms_average(it) = rms_average(it)/REAL(1000) !and finds average RMS     WRITE (13, '(2F16.5)') REAL(it), rms_average(it) ! for best fit plane of     CALL Flush(13) !Empty buffer !most planar satellite END DO !combination in each string2 = TRIM(ADJUSTL(folder)) // '/sig_PPD.ps/CPS' CALL HistoPlot(nit,101, REAL(rms_average), 'RMS_(kpc)','Probability', TRIM(ADJUSTL(string2)), .true.) WRITE (command,*) 'convert,-rotate_90`, // TRIM(ADJUSTL(folder)) // \&
'/sig_PPD.ps_, // TRIM(ADJUSTL(folder)) // \&
,/sig_PPD.jpg,
call system(command)
CLOSE(11) ; CLOSE(12) ; CLOSE(13) ; CLOSE(15) ; CLOSE(17)
END PROGRAM PlaneSignificance
SUBROUTINE RandomPoints !Generates a random realization containing nsats satellites

```
```

USE Global ! Each satellite is represented by 1,000 samples of possible
IMPLICIT NONE !positions along the line of sight from Earth.
!This routine is different to that of the same name in
!'PlaneSigRMS.f95, which includes only 1 possible position
!for each artificial satellite
LOGICAL :: in_poly
DOUBLE PRECISION :: sla_DSEP
CALL random_number (randnum)
randnum = randnum * 2999999.e0 + 1.e0
m31_dist = M31_Dist_PPD(NINT(randnum))
DO i = 1, nsats
2 CALL random_number(randnum)
sat_pick(i) = 1 + NINT(randnum*REAL(nsats - 1)) !Draw a random satellite
CALL random_number(randnum)
randnum = randnum * 499999.e0 + 1.e0
Sat_Dist_store = Sat_Dist(NINT(randnum),sat_pick(i))
pos(1,i) = ABS(Sat_Dist_store * cos(theta(sat_pick(i))) * tan(xi(sat_pick(i)))) !Determine length of x vector for each satellite
IF (xi(sat_pick(i)) . lt. 0.e0) THEN
pos(1,i)=-1.e0 * pos(1,i) !Determine if x is positive or negative
END IF
pos(2,i) = ABS(Sat_Dist_store * sin(eta(sat_pick(i)))) !Determine length of y vector for each satellite
IF (eta(sat_pick(i)) .lt. 0.e0) THEN
pos(2,i) = -1.e0 * pos(2,i) !Determine if y is positive or negative
END IF
pos(3,i) = Sat_Dist_store * cos(theta(sat_pick(i))) - m31_dist !Determine length and sign of z vector
pos(3,i)= SQRT((pos(1,i)**2.e0) + (pos(2,i)**2.e0) + (pos(3,i)**2.e0)) !Rotate position vector to point
pos(1,i)=0.e0 ; pos(2,i) = 0.e0 !along z-axis
CALL random_number(randnum)
alpha_set = randnum * 360.e0 * (pi/180.e0) !Pick random longitude
CALL random_number(randnum) !Pick random latitude between 0 and 90 weighted
beta_set = ASIN(randnum) !by area of a sphere as a function of latitude
CALL random_number(randnum) !
IF (randnum .lt. 0.5e0) THEN !Re-assign latitude as
beta_set = beta_set !
ELSE !-1* latitude in
beta_set = -beta_set
END IF !50% 0f cases
CALL Rotate
pos(:,i) = MATMUL(y_rot,pos(:,i)) !Rotate to the chosen
pos(:,i) = MATMUL(x_rot,pos(:,i)) !random angle
xi_test = ATAN(abs(pos(1,i))/(m31_dist + pos(3,i)))) !Convert
IF (pos(1,i) . 1t. 0.e0) THEN !new random
xi_test = -xi_test !position
END IF !vector
eta_test = ATAN(abs(pos(2,i))/SQRT(pos(1,i) **2 + (m31_dist + pos(3,i))**2))! into
IF (pos(2,i) . lt. 0.e0) THEN !non t.p
eta_test = -eta_test !eta and

```
```

    END IF
    art_xi(i)= xi_test !
    art_eta(i) = eta_test !Store position on sky of
    xi_dble = art_xi(i) !randomly oriented satellite
    eta_dble = art_eta(i) !for 1000 samples.
    art_theta(i) = sla_DSEP(0.d0, 0.d0, xi_dble, eta_dble) !
    new_Earth_Dist = abs(pos(2,i))/sin(abs(eta_test)) !Calculate new distance of sat from Earth after rotation
    Sat_Dist_change(i) = new_Earth_Dist - Sat_Dist_store !Calculate the difference between the new and old
                ! Earth distances for each satellite
    RA = xi_test
    DEC = eta_test !Use sla_DS2TP
                            !to convert true
    CALL sla_DS2TP (RA, DEC, 0.d0, 0.d0, xi_dble, eta_dble, j) !eta amd xi to
            !their tangent
    xi_test = xi_dble * (180.e0/pi) !plane projections
    eta_test = eta_dble * (180.e0/pi) !
    IF (in_poly(xi_test,eta_test,134,SAP_xi,SAP_eta)) THEN !Re-generate
                                    !the new
    ELSE
        goto 2
                                !randomized
                                    !satellite
                                    ! position if
                                    !the current
    spotR = ((xi_test*cos(51.9d0*pi/180.d0) + eta_test*sin(51.9d0*pi/180.d0))**2 / 6.25d0) + & !choice doesn't
    ((xi_test*sin(51.9d0*pi/180.d0) - eta_test*\operatorname{cos(51.9d0*pi/180.d0))**2 / 1.d0) !fall within}
                                    !}\mathrm{ the PAndAS
    IF (spotR .le. 1.e0) THEN !footprint
        goto 2 las viewed
    END IF !from Earth
    END DO
new_sats = .true
CALL MaxSigFind
new_sats = .true.
DO j = 1, 999
CALL random_number(randnum)
randnum = randnum * 2999999.e0 + 1.e0
m31_dist = M31_Dist_PPD(NINT(randnum))
DO i = 1, nsats
CALL random_number (randnum)
randnum = randnum * 499999.e0 + 1.e0
Sat_Dist_store = Sat_Dist(NINT(randnum), sat_pick(i)) + Sat_Dist_change(i) ! Adjust drawn Earth distance for new position
pos(1,i) = ABS(Sat_Dist_store * cos(art_theta(i)) * tan(art_xi(i))) !Determine length of x vector for each satellite
IF (art_xi(i) .lt. 0.e0) THEN
pos(1,i)=-1.e0* pos(1,i) !Determine if x is positive or negative
END IF
pos(2,i) = ABS(Sat_Dist_store * sin(art_eta(i))) !Determine length of y vector for each satellite
IF (art_eta(i).lt. 0.e0) THEN
pos(2,i)=-1.e0* pos(2,i) !Determine if y is positive or negative
END IF
pos(3,i) = Sat_Dist_store * cos(art_theta(i)) - m31_dist !Determine length and sign of z vector
END DO

```
```

    CALL MaxSigFind
    END DO
END SUBROUTINE RandomPoints

```
!'Rotate, Subroutine - See 'PlaneSigRMS.f95,
SUBROUTINE MaxSigFind ! Finds the best fit plane through the \(*\) most planar* combination
USE Global ! of 'nsatsub, satellites. This is achieved by "throwing in" 10,000
IMPLICIT NONE ! random planes and determining the 'nsatsub, closest satellites to
    !each plane and the associated RMS
min_rms \(=\) 9999.e0
DO i \(=1,10000\)
    IF (new_sats) THEN
        closest_sats \(=999 . e 0\) ! If finding the best fit combo each time
        closest_sats_id \(=0 \quad\) !reset these parameters
    ELSE
        closest_sats_id \(=\) best_sats
    END IF
    norm \(=(/ 0 . \mathrm{e} 0,0 . \mathrm{e} 0,1 . \mathrm{e} 0 /)\)
    CALL random_number (randnum)
    alpha_set \(=\) randnum \(* 360 . e 0 *(\) pi/180.e0) !Pick random longitude
    CALL random_number (randnum) ! Pick random latitude between 0 and 90 weighted
    beta_set \(=\operatorname{ASIN}(\) randnum \() \quad\) !by area of a sphere as a function of latitude
    CALL random_number (randnum) !
    IF (randnum . lt. 0.5 e 0 ) THEN ! Re-assign latitude as
        beta_set \(=\) beta_set !
    ELSE !-1* latitude in
        beta_set \(=-b e t a \_s e t\)
            !
    END IF \(\quad!50 \%\) of cases
    CALL Rotate
    norm (:) \(=\) MATMUL(y_rot, norm (:)) !Rotate to the chosen
    norm (:) \(=\) MATMUL(x_rot, \(\operatorname{norm}(:)\) ) !random angle
    IF (new_sats) THEN
    DO \(\mathrm{k}=1\), nsats
        planeDist \(=\operatorname{abs}(\operatorname{norm}(1) * \operatorname{pos}(1, \mathrm{k})+\operatorname{norm}(2) * \operatorname{pos}(2, \mathrm{k})+\operatorname{norm}(3) * \operatorname{pos}(3, \mathrm{k}))\) !
        IF (planeDist . lt. MAXVAL(closest-sats)) THEN ! Find the closest , nsatsub,
            \(u=\operatorname{MAXLOC}\left(c l o s e s t \_s a t s, ~ D I M=1\right) \quad\) !satellites to the currently
            closest_sats \((u)=\) planeDist !tested plane
            closest_sats_id(u) \(=k\)
                        !
        END IF
    END DO
    END IF
    \(\mathrm{rms}=0 . \mathrm{d} 0 \quad\) !
    DO \(k=1\), nsatsub ! Measure the RMS for the plane based on the closest 'nsatsub' satellites

        rms \(=\mathrm{rms}+(\) planeDist \() * * 2\) !
    END DO
    \(\mathrm{rms}=\operatorname{SQRT}(\mathrm{rms} / \mathrm{nsatsub}) \quad\) !
    IF (rms . lt. min_rms) THEN
```

    min_rms = rms !If the RMS is the
    best_sats = closest_sats_id !lowst encountered so far
    END IF !then store it
    END DO
rms_average(it) = rms_average(it) + min_rms !min_rms is now a good approximation to the lowest possible for the tested sample
END SUBROUTINE MaxSigFind

```

```

!'Theta_Phi' Subroutine - See 'PlaneSigRMS.f95,

```

```

!'HistoPlot, Subroutine - See 'PlaneSigRMS.f95,

```

```

!logical function in_poly(x,y,np,xp,yp) omitted - see MF_TRGB.f95 in preceding appendix
!real function fimag(x0,xs,xe,y0,ys,ye) omitted - see MF_TRGB.f95 in preceding appendix

```
\(\qquad\)
```

!'SampledDist, Subroutine - See 'PlaneSigRMS.f95,

```

```

!'FixedDist' Subroutine - See 'PlaneSigRMS.f95'
!The function of this subroutine is to read in the best fit satellite positions
!(as opposed to the positions generated from sampled satellite distances). This
!subroutine is not included specifically in 'PlaneSigRMS.f95' but it's functions
lare performed at the beginning of the 'Significance, subroutine and at the end
!of the 'SampledDist, subroutine

```

```

!'BorderGet' Subroutine - See 'PlaneSigRMS.f95'

```

Program: pole_vicinity_counts_satid_w.f95
Creation Date: 24 June 2012
Relevant Section: Ch. 5; Paper III §3.3
Notes: This is an analysis program for handling pole distribution maps from the real data, as produced using Subroutines for Processing Satellite Subsets (p. 280). A similar program was written to process the individual pole distribution maps from the many random realizations of satellites, which are then averaged. The code in this program performs two main tasks. The first is to generate a density profile for all poles falling within \(15^{\circ}\) of the most frequent pole location (e.g. Fig. 12 in Paper III). The second is to produce a histogram showing the extent to which each satellite has contributed to the most frequent pole (e.g. Fig. 13 in Paper III).
```

MODULE Global ! Defines all variables
IMPLICIT NONE
INTEGER :: i, j, k, ios
REAL :: counts, angle, err_samps, rad_bins(15,2) = 0.e0, sat_counts(25,2)
REAL :: max_counts, pole_theta_mode, pole_phi_mode
PARAMETER(err_samps = 100.e0)
REAL*8 :: dummy, best_theta, best_phi, pole_theta, pole_phi, pi
PARAMETER(pi = acos(-1.d0))
REAL :: ncombos = 53130.e0
REAL :: cum_pole_count
REAL :: sat (25)
LOGICAL :: cumulative
PARAMETER(cumulative = .true.)
END MODULE Global

```
```

PROGRAM pole_vicinity_counts !Counts number of poles within 'x' degrees of the best-fit pole
USE Global !where }x\mathrm{ is an integer such that 1 .ge. x .ge. 15
IMPLICIT NONE !Counts are divided by the number of samples of each combination
DOUBLE PRECISION :: sla_DSEP
best_theta = 38.37154d0 ; best_phi = -78.7439d0 !lat and long of most freq pole
!best_theta = 9.9d0 ; best_phi = -87.9d0 !lat and long of blob
!|Open file with
!\/pole positions
OPEN(unit = 11, file='Sat_Combo_Planes/Plane_Stats_5_sats_err_weighted/poles_per_pos_5_sats.dat', status = 'old')
DO i = 1, 15 !degree
rad_bins(i,1) = REAL(i) !values

```
```

END DO !of bin
best_theta = best_theta * (pi/180.d0) !Convert to
best_phi = best_phi * (pi/180.d0) !radians
cum_pole_count = 0.e0
i = 0
DO WHILE (.TRUE.)
i = i + 1
!|Read in pole positions and satellite
!\/contributions at that position
READ (11, *, IOSTAT = ios) dummy, dummy, dummy, pole_theta, pole_phi, counts, \&
sat(1), sat(2), sat(3), sat(4), sat(5), sat(6), sat(7), sat(8), sat(9), sat(10), \&
sat(11), sat(12), sat(13), sat(14), sat(15), sat(16), sat(17), sat(18), sat(19), sat(20), \&
sat(21), sat(22), sat (23), sat (24), sat (25)
cum_pole_count = cum_pole_count + counts
IF (counts .gt. max_counts) THEN
max_counts = counts
pole_theta_mode = pole_theta
pole_phi_mode = pole_phi
END IF
pole_theta = pole_theta * (pi/180.d0) !Convert to
pole-phi = pole_phi * (pi/180.d0) !radians
!|Measure angular distance between current
!\/pole and best-fit pole (uses SLALIB)
angle = sla_DSEP(best_phi, best_theta, pole_phi, pole_theta)
angle = angle * (180.d0/pi) !Convert back to degrees
!|Find angular distance bin to put
! \/pole into (if it is within 15 degrees)
IF (angle .le. 1.e0) THEN
rad_bins(1,2) = rad_bins(1,2) + counts
DO k = 1, 25 ! Count number of
sat_counts(k,2) = sat_counts(k,2) + sat(k) !contributions to this
END DO !pole from each satellite
ELSE IF (angle .gt. 1.e0 .and. angle . le. 2.e0) THEN
rad_bins(2,2) = rad_bins(2,2) + counts
DO k = 1, 25 ! Count number of
sat_counts(k,2) = sat_counts(k,2) + sat(k) !contributions to this
END DO !pole from each satellite
ELSE IF (angle .gt. 2.e0 .and. angle . le. 3.e0) THEN
rad_bins(3,2) = rad_bins(3,2) + counts
DO k = 1, 25 !Count number of
sat_counts(k,2) = sat_counts(k,2) + sat(k) !contributions to this
END DO !pole from each satellite
ELSE IF (angle .gt. 3.e0 .and. angle . le. 4.e0) THEN
rad_bins(4,2) = rad_bins (4,2) + counts
ELSE IF (angle .gt. 4.e0 .and. angle .le. 5.e0) THEN
rad_bins(5,2) = rad_bins(5,2) + counts
ELSE IF (angle .gt. 5.e0 . and. angle . le. 6.e0) THEN
rad_bins(6,2) = rad_bins(6,2) + counts

```
```

ELSE IF (angle.gt. 6.e0 .and. angle.le. 7.e0) THEN
rad_bins(7,2) = rad_bins(7,2) + count
ELSE IF (angle.gt. 7.e0 .and. angle .le. 8.e0) THEN
rad_bins(8,2) = rad_bins(8,2) + counts
ELSE IF (angle.gt. 8.e0 .and. angle.le. 9.e0) THEN
rad_bins(9,2) = rad_bins(9,2) + counts
ELSE IF (angle .gt. 9.e0 .and. angle . le. 10.e0) THEN
rad_bins(10,2) = rad_bins(10,2) + counts
ELSE IF (angle.gt. 10.e0 .and. angle .le. 11.e0) THEN
rad_bins(11,2) = rad_bins(11,2) + counts
ELSE IF (angle.gt. 11.e0 .and. angle . le. 12.e0) THEN
rad_bins(12,2) = rad_bins(12,2) + counts
ELSE IF (angle .gt. 12.e0 .and. angle . le. 13.e0) THEN
rad_bins(13,2) = rad_bins(13,2) + counts
ELSE IF (angle.gt. 13.e0 .and. angle .le. 14.e0) THEN
rad_bins(14,2) = rad_bins(14,2) + counts
ELSE IF (angle.gt. 14.e0 .and. angle.le. 15.e0) THEN
rad_bins(15,2) = rad_bins(15,2) + counts
END IF
!/\Find angular distance bin to put
!|pole into (if it is within 15 degrees)
!| Chack for
!\/end of file
IF (ios == -1) THEN
i = i - 1
exit
ELSE IF (ios .gt. 0) THEN
WRITE (*,*) i
i=i-1
cycle
END IF
END DO
WRITE (*,*) "Most_frequent_pole_at_theta_=", pole_theta_mode, "phi_=", pole_phi_mode, "with", max_counts, "counts."
ncombos = cum_pole_count/ err_samps
rad_bins(:,2) = rad_bins(:,2)/ err_samps !Divide by number of samples
sat_counts(:,2) = sat_counts(:,2)/err_samps
DO i = 1, 25
sat_counts(i,1) = REAL(i)
END DO
IF (cumulative) THEN
DO i = 2, 15
rad_bins(i,2) = rad_bins(i,2) + rad_bins(i-1,2) ! Convert to cumulative counts
END DO
END IF
DO i = 1, 15
!Print number
WRITE (*, '(3F16.5)') rad_bins(i,1), rad_bins(i,2), \& !of poles
(rad_bins(i,2)/ncombos) * 100.e0 !between x-1 and
END DO !x (< 15) degrees
IF (cumulative) THEN
WRITE (*,*) "Total_poles_within_15_degrees_of_best_fit_pole:", rad_bins(15,2),\&

```
```

"(",(rad_bins(15,2)/ncombos) * 100.e0, "%_)"
ELSE
WRITE (*,*) "Total\_poles\lrcornerwithin\_15^degrees\_of\lrcornerbest-fit\_pole:", SUM(rad_bins(:,2)), \&
"(",(SUM(rad_bins(:,2))/ncombos) * 100.e0, "%%)"!
END IF
WRITE (*,*) ""
WRITE (*,*) "Contributions_from_each_satellite_to_a_pole_within_3_degrees_of_most_frequent_pole:_"
!Print number of contributions
DO i = 1, 25 !from each satellite to a
WRITE (*, '(2F16.5)') sat_counts(i,1), sat_counts(i,2)!pole within 3 degrees of the
END DO !best fit pole
!|Make histogram of average pole density in 15 nested
!\/one degree wide annuli around the most frequent pole
CALL pgbegin(0,'pole_sat_prof_err-w.ps/CPS',1,1)
!CALL pgbegin(0,'pole_sat_prof_err_w_blob.ps/CPS',1,1)
CALL pgenv(0., 15.,0., 1.1*MAXVAL(rad_bins(:,2))/ncombos, 0, 0)
CALL pgbin(15, rad_bins(:,1) - 0.5, rad_bins(:,2)/ncombos, .true.)
IF (cumulative) THEN
CALL pglab('Degrees', 'Cumulative_Probability',',')
ELSE
CALL pglab('Degrees', 'Probability',',')
END IF
CALL pgend
!/\Make histogram of average pole density in 15 nested
!|lone degree wide annuli around the most frequent pole

```
\(\qquad\)
```

CLOSE(11)
!|Make histogram of contributions of each satellite to a pole
!\/with in 3 degrees of the location of most frequent pole
CALL pgbegin(0,' pole_sat_cont_err-w.ps/CPS',1,1)
!CALL pgbegin(0,'pole_sat_cont_err_w_blob.ps/CPS', 1, 1)
CALL pgsvp(0.1,0.9,0.1,0.9)
CALL pgswin(0.5,25.5,0.,1.1*MAXVAL(sat_counts(:,2)))
CALL pgbox('BCST', 0.0,0, 'BCNST', 0.0,0)
CALL pgbin(25, sat_counts(:,1), sat_counts(:,2), .true.)
CALL pglab(,','counts',',)
CALL PGPTXT (1.15, -1., 90.0, 1.0, 'I')
CALL PGPTXT (2.15, -1., 90.0, 1.0, ' II')
CALL PGPTXT (3.15, -1., 90.0, 1.0, 'III')
CALL PGPTXT (4.15, -1., 90.0, 1.0, 'V')
CALL PGPTXT (5.15, -1., 90.0, 1.0, 'IX')
CALL PGPTXT (6.15, -1., 90.0, 1.0, 'X')
CALL PGPTXT (7.15, -1., 90.0, 1.0, 'XI')
CALL PGPTXT (8.15, -1., 90.0, 1.0, 'XII')
CALL PGPTXT (9.15, -1., 90.0, 1.0, 'XIII')

```
```

CALL PGPTXT (10.15, -1., 90.0, 1.0, 'XIV')
CALL PGPTXT (11.15, -1., 90.0, 1.0, 'XV')
CALL PGPTXT (12.15, -1., 90.0, 1.0, 'XVI')
CALL PGPTXT (13.15, -1., 90.0, 1.0, 'XVII')
CALL PGPTXT (14.15, -1., 90.0, 1.0, 'XVIII')
CALL PGPTXT (15.15, -1., 90.0, 1.0, 'XIX')
CALL PGPTXT (16.15, -1., 90.0, 1.0, 'XX')
CALL PGPTXT (17.15, -1., 90.0, 1.0, 'XXI')
CALL PGPTXT (18.15, -1., 90.0, 1.0, 'XXII')
CALL PGPTXT (19.15, -1., 90.0, 1.0, 'XXIII')
CALL PGPTXT (20.15, -1., 90.0, 1.0, 'XXIV')
CALL PGPTXT (21.15, -1., 90.0, 1.0, 'XXV')
CALL PGPTXT (22.15, -1., 90.0, 1.0, 'XXVI')
CALL PGPTXT (23.15, -1., 90.0, 1.0, 'XXVII')
CALL PGPTXT (24.15, -1., 90.0, 1.0, 'group')
CALL PGPTXT (25.15, -1., 90.0, 1.0, 'M33')
CALL pgend
!/\Make histogram of contributions of each satellite to a pole
!| with in 3 degrees of the location of most frequent pole
END PROGRAM pole_vicinity_counts

```

Program: aitoff_hammer.f95
Creation Date: Cir. May 2012 Many versions
Relevant Section: Ch. 5 (Paper III aitoff-hammer plots)
Notes: This program illustrates the way in which the aitoff-hammer plots were produced. I wrote several versions but that presented here is the one used for the standard plots which show the plane member satellites, the great circle on the sky representing the plane, and the pole and anti-pole of the plane (see Fig. 15 of Paper III for example). The aitoff-hammer grid is produced by first making a rectangular grid of a large number of points along the desired lines of latitude and longitude and then transforming the \(x\) and \(y\) of the points via a Hammer projection. The points are then linked up to produce the final grid. All positions in Paper III (satellites and plane poles) are actually calculated first in the Cartesian coordinate system of Fig. 4.1, rotated into the M31-centric reference frame, converted to Spherical coordinates and then finally transformed into their equivalent Hammer projection locations for plotting.
```

MODULE Global ! Defines all variables
IMPLICIT NONE
INTEGER :: i, j, s, idum = -9999, nit, seam_loc, color(29), nsats
PARAMETER( n it = 1000000)
PARAMETER(nsats = 27)
REAL :: pi
PARAMETER(pi = acos(-1.e0))
REAL :: sat_xyz (29,3)
REAL :: theta_coord, phi_coord
REAL :: theta(29), phi(29), theta_t(28), phi_t(28)
REAL :: theta_ah, phi_ah
REAL :: alpha_set, beta_set, gamma_set
REAL :: x_rot (3,3), y_rot (3,3), z_rot (3,3)
REAL :: pole(2), pole2(2), gc (361,2), gc_t (361,2)
REAL :: cart_hold(361,3), pole_cart(3), gc_cart (361,3)
REAL :: seam_val
REAL :: Best_Sat_Dist(27), Sat_Pos(27,2), xi(27), eta(27), m31_dist
REAL*8 :: RA, DEC, xi_dble, eta_dble
END MODULE Global

```
PROGRAM aitoff_hammer_proj ! Master program
USE Global
IMPLICIT NONE
CALL sat_xyz_data
CALL aitoff_hammer
END PROGRAM aitoff_hammer_proj
```

SUBROUTINE aitoff_hammer !Produce the plot
USE Global
IMPLICIT NONE
INTEGER :: it_i, it_j, it_k
REAL :: lat (181,2,13), lon(181,2,13), lat_t (181,2,13), lon_t(181,2,13)
!Note lat and lon are lines of constant latitude and longitude
!respectively, each made up of 181 dots
!lat (:,1,:) is the longitude of the dot
!lat (:,2,:) is the latitude of the dot
!So lon(81,1,3) is the longitude of the 81st dot of the 3rd parallel
! It has a value of +120 (i.e. the fixed longitude of this parallel)
!lon(81,2,3) has a value of -10 i.e. the latitude of the dot along
!this line of longitude
!
!lat_t and lon_t store the lat and lon values before their conversion
!to the aitoff hammer projection
lon(:,1,1) = 180.e0
lon(:,1,2) = 150.e0
lon(:,1,3) = 120.e0
lon(:,1,4) = 90.e0
lon(:,1,5) = 60.e0
lon(:,1,6) = 30.e0
lon(:, 1,7) = 0.e0
lon(:,1,8) = -30.e0
lon(:,1,9) = -60.e0
lon(:,1,10) = -90.e0
lon(:,1,11) = -120.e0
lon(:,1,12) = - 150.e0
lon(:,1,13) = -180.e0
lat(:,2,1) = 90.e0
lat(:,2,2) = 75.e0
lat(:,2,3) = 60.e0
lat(:,2,4) = 45.e0
lat (:,2,5) = 30.e0
lat(:,2,6) = 15.e0
lat(:,2,7) = 0.e0
lat(:,2,8) = -15.e0
lat(:,2,9) = -30.e0
lat(:,2,10) = -45.e0
lat}(:,2,11)=-60.e
lat(:,2,12) = -75.e0
lat(:,2,13) = -90.e0
DO it_i = 1, 181
lon(it_i , 2,:) = REAL(it_i - 91)
lat(it_i, 1,:) = REAL((2 * it_i) - 182)
END DO
lon = lon * (pi/180.e0) ; lat = lat * (pi/180.e0) !Convert to radians
lon_t = lon ; lat_t = lat
DO it_i = 1, 13
!The conversion to an aitoff-
DO it_j = 1, 181
!hammer projection

```
```

    lon(it_j,1,it_i) = 2.e0 * SQRT(2.e0) * cos(lon_t(it_j, 2,it_i)) * sin(lon_t(it_j,1,it_i)/2.e0) / &
    SQRT(1.e0 + cos(lon_t(it_j,2,it_i)) * cos(lon_t(it_j,1,it_i)/2.e0))
    lon(it_j,2,it_i) = SQRT(2.e0) * sin(lon_t(it_j, 2,it_i)) / \&
SQRT(1.e0 + cos(lon_t(it_j, 2,it_i)) * cos(lon_t(it_j,1,it_i)/2.e0))
lat(it_j,1,it_i) = 2.e0 * SQRT(2.e0) * cos(lat_t(it_j, 2,it_i)) * sin(lat_t(it_j,1,it_i)/2.e0)/\&
SQRT(1.e0 + cos(lat_t(it_j,2,it_i)) * cos(lat_t(it_j,1,it_i)/2.e0))
lat(it_j, 2,it_i) = SQRT(2.e0) * sin(1at_t(it-j,2,it_i)) / \&
SQRT(1.e0 + cos(lat_t(it_j, 2,it_i)) * cos(lat_t(it_j,1,it_i)/2.e0))
END DO
END DO
|| Plotting
! \/ Code
CALL pgbegin(0,'sat_combo_15_sats.ps/CPS',1,1)
CALL pgenv(-1.0* pi, 1.0 * pi, -0.5* pi, 0.5* pi, 1, -2)
DO it_i = 1, 13
DO it_j = 2, 181
CALL pgline(2, (/lon(it_j - 1, 1, it_i), lon(it_j, 1, it_i)/), (/lon(it_j - 1, 2, it_i), lon(it_j, 2, it_i)/))
CALL pgline(2, (/lat(it_j - 1, 1, it_i), lat(it_j, 1, it_i)/), (/lat(it_j - 1, 2, it_i), lat(it_j, 2, it_i)/))
END DO
END DO
!This segment takes the normal vector of the best fit plane, finds the corresponding
!pole, anti-pole and great-circle in lat., long. and then converts all to an aitoff
!-hammer projection. These are then plotted on the aitoff-hammer sphere
pole_cart = (/ 0.15819097, 0.76853156, 0.61994755 /) !x,y,z of normal vector to BFP
color(1) = 2 ; color(8) = 2 ; color(15) = 1 ; color(22) = 2
color(2) = 1 ; color(9)=2 ; color(16)=1; color(23)=2
color(3) = 2 ; color(10) = 2 ; color(17) = 1 ; color(24) = 2 ! 8
color(4) = 1; color(11)=1 ; color(18)=1; color(25)=2 !8
color(5) = 2 ; color(12) = 2 ; color(19) = 1 ; color(26) = 2 !8
color(6) = 1 ; color(13) = 2 ; color(20) = 1 ; color(27) = 1
color(7) = 2 ; color(14)=1 ; color(21) = 2 ; color(28)=1
color(29)=2 !Colour for the NGC147/ NGC185/ AND XXX group midpoint icon
CALL Theta_Phi(pole_cart(1), pole_cart(2), pole_cart(3)) !
pole(1) = theta_coord !lat, long
pole(2) = phi_coord
pole2(1)= -1.e0 * pole(1) !
pole2(2) = pole(2) + 180.e0 !Find latitude and
IF (pole2(2).gt. 180.e0) THEN !longitude of anti-pole
pole2(2) = pole2(2) - 360.e0 !
END IF
DO i = 1, 361 !Find x,y,z of vectors perpendicular
gc_cart(i,1) = cos(REAL(i-181)* (pi/180.e0)) !to pole_cart - i.e. cartesian
gc_cart(i,2)= sin(REAL(i-181)* (pi/180.e0)) !coordinates of best fit plane !!!!!!!!!!!!!!
gc_cart(i,3) = -1.e0 * (pole_cart(1) * gc_cart(i,1) + pole_cart(2) * gc_cart(i,2))/ \& !
pole_cart(3)
gc_cart(i,: ) = gc_cart(i,:)/ SQRT(gc_cart(i,1)**2.e0 + gc_cart(i,2) ** 2.e0 + \& !
gc_cart(i,3)**2.e0)

```
```

END DO
seam_val = pi
DO i = 1, 361
CALL Theta_Phi(gc_cart(i,1), gc_cart(i,2), gc_cart(i,3)) ! Convert best
!fit plane x,y,z
gc(i,1)= theta_coord * (pi/180.e0) !to their lat.
gc(i,2) = phi_coord * (pi/180.e0) !and long. values
IF (gc(i,2) .le. seam_val) THEN !Find seam
seam_val = gc(i,2) !where -180
seam_loc = i !longitude
END IF !meets +180
END DO
gc_t = gc !Set gc_t to pre-order-fix gc
IF (gc(1,2) .lt.ggc(361,2)) THEN !
gc(1,:) = gc_t (361,:) !fix
gc(361,:) = gc_t (1,:) !order
END IF !
gc_t = gc !Set gc_t to new gc
DO i = 1, 361
gc(i,2)=2.e0*SQRT(2.e0)*\operatorname{cos(gc_t(i,1)) * sin(gc_t(i,2)/2.e0) !Transform great -}
gc(i,2) = gc(i,2) / SQRT(1.e0 + cos(gc_t(i,1)) * cos(gc_t(i,2)/2.e0)) !circle into
gc(i,1)=\operatorname{SQRT}(2.e0)*\operatorname{sin}(gc_t(i,1))}\quad\mathrm{ !aitoff-hammer
gc(i,1) = gc(i,1) / SQRT(1.e0 + cos(gc_t(i,1)) * cos(gc_t(i,2)/2.e0)) !projection
END DO
CALL pgsci(4)
CALL pgslw (3)
CALL pgsch(2.0)
CALL aitoff_convert(pole(1), pole(2))
CALL pgpt(1, pole(2), pole(1), 845) !Plot pole
CALL aitoff_convert(pole2(1), pole2(2))
CALL pgpt(1, pole2(2), pole2(1), 846) !anti-pole and
CALL pgsch(1.0) !
CALL pgslw(1) !great circle
DO i = 2, 361
CALL pgline(2, (/gc(i-1,2), gc(i,2)/), (/gc(i-1,1), gc(i,1)/))
END DO
!|Plot individual satellites
!\/and labels
s = 1
CALL pgsci(color(s))
CALL pgpt(1, phi(s), theta(s), 843)
CALL pgptxt(phi(s) - 0.1, theta(s)+0.0, 0., 0., 'I')
!:
!: S = 2, ..., 26
!:
s = 27
CALL pgsci(color(s))
CALL pgpt(1, phi(s), theta(s), 768)

```
```

CALL pgptxt(phi(s) - 0.30, theta(s) +0.05, 0., 0., 'M33')
s = 28
CALL pgsci(color(s))
CALL pgpt(1, phi(s), theta(s), 2284)
CALL pgptxt(phi(s) - 0.3, theta(s)+0.05, 0., 0., 'MWy')
!s = 29 !Ngc147/ NGC185/ And XXX Group midpoint icon
!CALL pgsci(color(s))
!CALL pgpt(1, phi(s), theta(s), 0904)
!/\ Plot individual satellites
!|and labels
CALL pgsci(1)
CALL pgpt(1, 0., 0., 2293)
CALL pgptxt(-0.3, 0.05, 0., 0., 'M31')
CALL pgsch(1.0)
!| Plots labels for lines of
!\/constant lat. and long
CALL pgptxt(lat (1,1,1) - 0.05, lat(1,2,1)+0.05, 0., 0., '90')
CALL pgptxt(lat ( 1, 1,2) - 0.06, lat ( 1, 2, 2) + 0.05, 0., 0., '75,)
CALL pgptxt(lat ( 1, 1,3) - 0.08, lat ( 1,2,3) +0.05, 0., 0., '60')
CALL pgptxt(lat (1, 1,4) - 0. 14, lat (1,2,4) +0.03, 0., 0., '45')
CALL pgptxt(lat (1, 1,5) - 0.18, lat (1,2,5) +0.0, 0., 0., '30')
CALL pgptxt(lat (1,1,6) - 0.20, lat( (1,2,6) -0.02, 0., 0., '15')
CALL pgptxt(lat (1, 1,7) -0.15, lat (1, 2, 7) -0.02, 0., 0., '0')
CALL pgptxt(lat (1, 1,8) - 0.33, lat ( 1, 2,8) -0.05, 0., 0., , - 15')
CALL pgptxt(lat (1, 1,9) - 0.31, lat (1,2,9) -0.07, 0., 0., , -30')
CALL pgptxt(lat (1,1,10) - 0.24, lat (1,2,10) -0.13, 0., 0., ,-45')
CALL pgptxt(lat (1,1,11) - 0.20, lat (1,2,11)-0.13, 0., 0., '-60')
CALL pgptxt(lat (1, 1,12) - 0.21, lat ( 1,2,12) -0.14, 0., 0., , -75')
CALL pgptxt(lat (1,1,13) - 0.20, lat (1,2,13) -0.12, 0., 0., '-90')
CALL pgsch(0.5)
CALL pgptxt(lon(91,1,11) - 0.20, lon(91,2,11)-0.12, 0., 0., ' - 120')
CALL pgptxt(lon(91,1,9) - 0.15, lon (91,2,9)-0.12, 0., 0., , -60')
CALL pgptxt(lon(91,1,5) - 0.12, lon (91,2,5) -0.12, 0., 0., '60')
CALL pgptxt(lon(91,1,3) - 0.16, lon(91,2,3)-0.12, 0., 0., '120')
CALL pgptxt(lon(91,1,1) - 0.16, lon(91,2,1)-0.12, 0., 0., , 180')
CALL pgsch(1.0)
CALL pgend
END SUBROUTINE aitoff_hammer
SUBROUTINE sat_xyz_data !Get the data
USE Global
IMPLICIT NONE
DOUBLE PRECISION :: sla_DSEP
m31_dist = 779.e0 !M31
! Best_Sat_Dist(1:27) as per 'PlaneSigRMS.f95' - see 'Significance' subroutine

```
!Sat_Pos (1:27,:) as per 'PlaneSigRMS.f95' - see 'SampledDist' subroutine
DO \(\mathrm{i}=1\), nsats
    \(x i(i)=S a t \_P o s(i, 1)\)
    eta(i) \(=\) Sat_Pos(i,2)
    \(x i(i)=x i(i) *(p i / 180 . e 0) \quad\) Convert angles from
    eta \((\mathrm{i})=\) eta \((\mathrm{i}) *(\mathrm{pi} / 180 . \mathrm{e} 0) \quad\) !degrees to radians
END DO
DO i \(=1\), nsats
    xi_dble \(=x i(i) ;\) eta_dble \(=\) eta(i)
    CALL sla_DTP2S (xi_dble, eta_dble, 0.d0, 0.d0, RA, DEC) ! Convert tangent plane
    IF (xi_dble.lt. 0.d0) then !projection angles into
        \(\mathrm{RA}=\mathrm{RA}-(2 . \mathrm{e} 0 * \mathrm{pi}) \quad\) ! their true angles using
    END IF !sla_DTP2S
    \(x i(i)=R A\)
    eta \((\) i \()=\) DEC !
END DO
DO \(\mathrm{i}=1\), nsats
    xi_dble \(=x i(i) \quad\) !Find the true angle
    eta_dble \(=\) eta(i) \(\quad\) !theta_t - the angle on
    theta_t (i) = sla_DSEP (0.d0, 0.d0, xi_dble, eta_dble)!the sky between M31 and
END DO ! the object (uses sla_DSEP)
DO \(\mathrm{i}=1\), nsats
    sat_xyz(i, 1) \(=\operatorname{ABS}\left(B e s t \_S a t \_D i s t(i) * \cos \left(t h e t a \_(i)\right) * \tan (x i(i))\right)\) ! Determine length of \(x\) vector for each satellite
    IF (xi(i) . lt. 0.e0) THEN
            sat_xyz(i, 1) \(=-1 . e 0 *\) sat_xyz(i,1) \(\quad\) ! Determine \(i f x\) is positive or negative
    END IF !
        sat_xyz(i,2) \(\operatorname{ABS}(\) Best_Sat_Dist (i) \(* \sin (e t a(i))) \quad\) ! Determine length of y vector for each satellite
    IF (eta (i) . 1t. 0.e0) THEN
            sat_xyz(i,2) \(=-1 . e 0 *\) sat_xyz(i,2) ! Determine if y is positive or negative
    END IF !
    sat_xyz(i,3)= Best_Sat_Dist(i) * cos(theta_t(i)) - m31_dist Determine length and sign of \(z\) vector
END DO
sat_xyz(28,:) \(=(/ 0 . e 0,0 . e 0,-779 . e 0 /) \quad\) !MWy
sat_xyz(29,:) \(=\) sat_xyz(25,:) + ((100.e0) ** (0.2e0*0.2e0)) * sat-xyz(26,:) ! NGC147/ NGC185/ AND XXX
sat_xyz \((29,:)=\) sat_xyz \((29,:) /(1 . e 0+(100 . e 0) * *(0.2 \mathrm{e} 0 * 0.2 \mathrm{e} 0)) \quad\) !group mid point
DO i \(=1,29\)
WRITE (*,*) i, sat_xyz(i,1), sat-xyz(i,2), sat-xyz(i, 3)
alpha_set \(=-(90 . \mathrm{e} 0-12.5 \mathrm{e} 0) *(\mathrm{pi} / 180 . \mathrm{e} 0)\) ! Rotate to bring back out of M31's inclination !
gamma_set \(=+(90 . \mathrm{e} 0-39.8 \mathrm{e} 0) *(\mathrm{pi} / 180 . \mathrm{e} 0)\) !angle and PA (i.e. to view from above the M31 pole) !
                                    ! Change
CALL Rotate
!
! to
sat_xyz(i,: ) = MATMUL(z_rot, sat-xyz(i,: )) Convert vectors back to how they would appear !
sat_xyz(i,:) = MATMUL(x_rot, sat_xyz(i,:)) !in M31 reference frame !M31
gamma_set \(=90 . \mathrm{e} 0 *(\) pi/180.e 0\() \quad\) ! coordinate
CALL Rotate ! Additional rotation in M31 galactic longitude !system
sat_xyz(i,: ) = MATMUL(z_rot, sat_xyz(i,:))!

341
342

344
345
346
347
348
349
350
351
352
353
354

356
```

WRITE (*,*) i, sat_xyz(i,1), sat_xyz(i, 2), sat_xyz(i, 3)
CALL Theta_Phi(sat_xyz(i, 1), sat_xyz(i, 2), sat_xyz(i, 3))
CALL aitoff_convert(theta_coord, phi_coord)
theta(i) = theta_coord ; phi(i) = phi_coord
END DO
END SUBROUTINE sat_xyz_data

```
SUBROUTINE aitoff_convert (theta_ah, phi_ah) ! Convert to aitoff-
IMPLICIT NONE ! hammer projection
REAL :: theta_ah, phi_ah, pre_theta_ah, pre_phi_ah
REAL :: pi
\(\operatorname{PARAMETER}(\mathrm{pi}=\operatorname{acos}(-1 . \mathrm{e} 0))\)
pre_theta_ah \(=\) theta_ah \(*(\) pi/180.e0 \()\)
pre_phi_ah \(=\) phi_ah \(*(\) pi/180.e 0\()\)
phi_ah \(=2 . e 0 * \operatorname{SQRT}(2 . e 0) * \cos (\) pre_theta_ah \() * \sin (\) pre_phi_ah/2.e0)
phi_ah \(=\) phi_ah \(/ \operatorname{SQRT}(1 . e 0+\cos (\) pre_theta_ah \() * \cos (\) pre_phi_ah/2.e0))
theta_ah \(=\operatorname{SQRT}(2 . e 0) * \sin (\) pre_theta_ah)
theta_ah \(=\) theta_ah \(/ \operatorname{SQRT}(1 . e 0+\cos (\) pre_theta_ah \() * \cos (\) pre_phi_ah/2.e0) \()\)
END SUBROUTINE aitoff_convert
!'Rotate, Subroutine - See 'PlaneSigRMS.f95,
', Theta_Phi, Subroutine - See 'PlaneSigRMS.f95,

\section*{Program: RR_Histograms.f95}

\section*{Creation Date: 7 Oct 2012}

Relevant Section: Ch. 5; Paper III Figs. 5 (RH column), 6, 9 (RH column), 16 (b)
Notes: This program illustrates the way in which the histograms of the goodness of fit statistic for the random realizations were generated. Throughout the entire thesis I have generated many histograms and toward the end I decided to make a stand alone subroutine 'HistoPlot' (see PlaneS igRMS.f95-p. 264) that automated the process. I later modified that subroutine to add the credibility interval color-coding used for the papers. It is this subroutine which is shown here: 'HistoPlotAdv.' The 'DataCall' subroutine is designed to handle the many different outputs from the various plane fitting programs. It is set up to plot the histogram of the average RMS (or other plane fitting statistic) values from the random satellite realizations and also to take the average of the histogram produced from the real data.
```

MODULE Global !Defines all variables used by BayesianTRGB
IMPLICIT NONE
INTEGER :: i, idum = -9999, ios, ndata, ndata2, counts
PARAMETER (ndata = 10000)
PARAMETER (ndata2 = 200000)
REAL :: signif(ndata), signif2, dummy, scale_factor, average_sig
CHARACTER :: folder*300, string*300, string2*300, command*300
LOGICAL :: Best15, RMS, AbVal, Asy, AsyFP, ML, Sigma
PARAMETER (Best15 = .true.)
PARAMETER (RMS = .false.)
PARAMETER (AbVal = .false.)
PARAMETER (Asy = .false.)
PARAMETER (AsyFP = .false.)
PARAMETER (ML = .false.)
PARAMETER (Sigma = .false.)
END MODULE Global

```
```

PROGRAM DataCall !Reads in data to be
USE Global ! plotted and passes
IMPLICIT NONE ! to HistoPlotAdv
IF (Best15) THEN !For Plane of best 15 satellites
OPEN(unit = 11, file='./Sat_Combo_planes/Plane_Stats_15_sats_RandReal_weighted/RMS_15_sats.dat', status = 'old')
OPEN(unit = 12, file='./Sat_Combo_planes/RMS_Plane_Stats_Best_15_sats/real_sig_wth_err.dat', status = 'old')
END IF
IF (RMS) THEN !For RMS distribution
OPEN(unit = 11, file='./Sat_Comp_Set_Stats/Plane_Stats_27_sats_RandReal_weighted_RMS/RMS_27_sats.dat', status = 'old')
OPEN(unit = 12, file='./Sat_Comp_Set_Stats/RMS_Plane_Stats_27_sats/real_sig_wth_err.dat', status = 'old')
END IF
IF (AbVal) THEN !For 'sum of Absolute Values' distribution
OPEN(unit = 11, file='./Sat_Comp_Set_Stats/Plane_Stats_27_sats_RandReal_weighted_AV/AV_27_sats.dat', status = 'old')

```
```

OPEN(unit = 12, file='./Sat_Comp_Set_Stats/AbVal_Plane_Stats/real_sig_wth_err.dat', status = 'old')
END IF
IF (Asy) THEN !For Asymmetry distribution
OPEN(unit = 11, file='./Sat_Comp_Set_Stats/Plane_Stats_27_sats_RandReal_weighted_Asy/Asy_27_sats.dat', status = 'old')
OPEN(unit = 12, file='./Sat_Comp_Set_Stats/Asymm_Stats/real_asy_wth_err.dat', status = 'old')
END IF
IF (AsyFP) THEN !For Distribution of Asymmetry about M31 tangent plane
OPEN(unit = 11, file='./Sat_Comp_Set_Stats/Plane_Stats_27_sats_RandReal_weighted_AsyFP/AsyFP_27_sats.dat', status = ,old')
OPEN(unit = 12, file='./Sat_Comp_Set_Stats/Asymm_Stats_FixedPlane/real_asy_wth_err.dat', status = 'old')
END IF
IF (ML .or. Sigma) THEN !For Maximum Likelihood (and sigma) distributions
OPEN(unit = 11, file='./Sat_Comp_Set_Stats/Plane_Stats_27_sats_RandReal_weighted_ML/ML_27_sats.dat', status = 'old')
OPEN(unit = 12, file='./Sat_Comp_Set_Stats/Plane_Stats/real_sig_wth_err.dat', status = 'old')
END IF
!|Read in plane fitting statistic (e.g. RMS) from
!\/Random Realizations for generation of histogram
i = 0
DO WHILE (.TRUE.)
i = i + 1
IF (i .gt. ndata) THEN
i = i-1
exit
END IF
IF (Sigma) THEN
READ (11, *, IOSTAT = ios) dummy, dummy, signif(i)
ELSE
READ (11, *, IOSTAT = ios) dummy, signif(i)
END IF
IF (ios == -1) THEN
i = i - 1
exit
ELSE IF (ios .gt. 0) THEN
WRITE (*,*) i
i=i - 1
cycle
END IF
END DO
!/\Read in plane fitting statistic (e.g. RMS) from
!|Random Realizations for generation of histogram
!|Read in plane fitting statistic (e.g. RMS) from Realizations
!\/of possible positions of the real satellites to find average
i = 0
DO WHILE (.TRUE.)
i = i + 1

```
```

IF (i .gt. ndata2) THEN
i = i-1
exit
END IF
IF (Sigma) THEN
READ (12, *, IOSTAT = ios) dummy, dummy, signif2
ELSE
READ (12, *, IOSTAT = ios) dummy, signif2
END IF
IF (Asy .or. AsyFP .or. ML .or. Sigma) THEN
ELSE
signif2 = 10.e0 ** signif2
END IF
average_sig = average_sig + signif2
IF (ios == -1) THEN
i = i - 1
exit
ELSE IF (ios .gt. 0) THEN
WRITE (*,*) i
i=i-1
cycle
END IF
END DO
!/\Read in plane fitting statistic (e.g. RMS) from Realizations
!|of possible positions of the real satellites to find average
average_sig = average_sig/ REAL(ndata2)
CALL pgbegin(0,'RR_Histogram.ps/CPS',1,1)
IF (Best15) THEN
CALL pgenv(5.,35.,0.,0.15, 0, 0) !For Best 15 satellites
END IF
IF (RMS) THEN
CALL pgenv(30.,90.,0.,0.07, 0, 0) !For RMS distribution
END IF
IF (AbVal) THEN
CALL pgenv(700.,1900.,0.,0.003, 0, 0) ! For AbVal distribution
END IF
IF (Asy) THEN
CALL pgenv(13.,28.,0.,0.4, 0, 0) !For Asymmetry distribution
END IF
IF (AsyFP) THEN
CALL pgenv(13.,28.,0.,0.8, 0, 0) !For Asymmetry Fixed Plane distribution
END IF
IF (ML) THEN
CALL pgenv(-69., - 59.,0.,0.4, 0, 0) !For ML distribution
END IF
IF (Sigma) THEN
CALL pgenv(30.,90.,0.,0.07, 0, 0) !For Sigma distribution
END IF
IF (Sigma) THEN
CALL HistoPlotAdv(ndata, 21, signif, ',, ,', '?', .true.) !Make Histogram

```
```

ELSE
CALL HistoPlotAdv(ndata, 51, signif, ',, ',, '?', .true.) !Make Histogram
END IF
!|Plot dot-dash line at location of histogram average from realizations
!\/of possible positions of the real satellites
CALL pgsls(3)
CALL pgsci(8
CALL pgslw (5)
IF (Best 15) THEN
CALL pgline (2, (/ average_sig, average_sig /), (/ 0.0, 0.15 /)) !For Best 15 satellites
END IF
IF (RMS) THEN
CALL pgline (2, (/ average_sig, average_sig /), (/ 0.0, 0.07 /)) !For RMS distribution
END IF
IF (AbVal) THEN
CALL pgline (2, (/ average_sig, average_sig /), (/ 0.0, 0.003 /)) !For AbVal distribution
END IF
IF (Asy) THEN
CALL pgline (2, (/ average_sig, average_sig /), (/ 0.0, 0.4 /)) !For Asymmetry distribution
END IF
IF (AsyFP) THEN
CALL pgline (2, (/ average_sig, average_sig /), (/ 0.0, 0.8 /)) !For Asymmetry distribution
END IF
IF (ML) THEN
CALL pgline (2, (/ average_sig, average_sig /), (/ 0.0, 0.4 /)) !For ML distribution
END IF
IF (Sigma) THEN
CALL pgline (2, (/ average_sig, average_sig /), (/ 0.0, 0.07 /)) !For Sigma distribution
END IF
CALL pgslw(1)
CALL pgsci(1)
CALL pgsls(1)
!/\ Plot dot-dash line at location of histogram average from realizations
!| of possible positions of the real satellites
IF (RMS .or. Best15) THEN
CALL pglab('Minimum_RMS_(kpc)', 'Probability', ',') !For RMS distribution
END IF
IF (AbVal) THEN
CALL pglab('Minimum_Absolute_Distance_Sum_(kpc)', 'Probability', ',') !For AbVal distribution
END IF
IF (Asy . or . AsyFP) THEN
CALL pglab('Maximum_Hemisphere_Satellite_Count', 'Probability', ',') !For Asymmetry distribution
END IF
IF (ML) THEN
CALL pglab('LOG10(Maximum_Likelihood)', 'Probability,, ',') !For ML distribution
END IF
IF (Sigma) THEN
CALL pglab('Plane_Sigma_(kpc)', 'Probability', ',') !For Sigma distribution
END IF
CALL pgend
WRITE (*,*) "Average_of_observed_plane:", average_sig
IF (Best15) THEN
counts = 0
DO i = 1, ndata
IF (signif(i) .le. average_sig) THEN
counts = counts + 1

```
```

    END IF
    END DO
    WRITE (*,*) "An\_RMS\_of", average_sig, "was\_equalled\_or\_exceeded", counts, "out\_of", ndata, "times."
    END IF
IF (Asy .or. AsyFP) THEN
counts = 0
DO i = 1, ndata
IF (signif(i) .ge. average_sig) THEN
counts = counts + 1
END IF
END DO
WRITE (*,*) "An_Asymmetry_of", average_sig, "was_equalled_or_exceeded", counts, "out_of", ndata, "times."
END IF
END PROGRAM
SUBROUTINE HistoPlotAdv(nval, data_hist_bins, data, xlabel, ylabel, device, normalize)
USE Global
IMPLICIT NONE
!
!**************Created 16 Jun 2012**************
!
!INTEGER nval = number of data points in histogram
!INTEGER data_hist_bins = number of bins in histogram
!REAL data(nval) = The array containing the data
!CHARACTER xlabel = Label of x-axis of histogram
!CHARACTER ylabel = Label of y-axis of histogram
!CHARACTER device = The plotting device ('?' if unsure)
!LOGICAL normalize = .true. if histogram is to be
!normalized, else set to .false.
!
!Uses PGPLOT
!
INTEGER :: data_hist-bins, nval, it_num, BFL
REAL :: bw, data(nval), data_hist(data_hist_bins,2), data_min, data_max
REAL :: BFV, psig, msig, max_bin_height, data_counts, pcounts, mcounts
REAL :: xpts(2), ypts(2), p90, m90, p99, m99
CHARACTER(LEN=*) :: xlabel, ylabel, device
LOGICAL :: normalize
!| Builds the specified
!\/ histogram
data_hist = 0.e0
data_min = MINVAL(data) ; data_max = MAXVAL(data)
bw = (data_max - data_min)/(REAL(data_hist_bins) - 1.e0)
DO it_num = 1, data_hist_bins
data_hist(it_num,1) = data_min + REAL(it_num -1) * bw
END DO
DO it_num = 1, nval
data_hist(NINT((data(it_num) - data_min)/bw) + 1,2) = \&
data_hist(NINT((data(it_num) - data_min)/bw) + 1,2) + 1.e0

```
```

END DO
IF (normalize) THEN
data_hist(:,2) = data_hist(:,2) / (bw * SUM(data_hist(:,2)))
END IF
!/\ Builds the specified
!| histogram
max_bin_height = 0.d0
DO it_num = 1, data_hist_bins
IF (data_hist(it_num,2) .gt. max_bin_height) THEN !
max_bin_height = data_hist(it_num,2) ! Find best fit TRGB value
BFV = data_hist(it_num,1)
BFL}= it_num
END IF
END DO
WRITE (*,*) "Best^fit_value_is_at:", BFV
data_counts = 0.d0 ; pcounts = 0.d0
DO it_num = BFL, data_hist_bins
pcounts = pcounts + data_hist(it_num,2)
END DO
DO it_num = BFL, data_hist_bins
data_counts = data_counts + data_hist(it_num,2)!Finds positive one sigma
IF (data_counts .ge. 0.682*pcounts) THEN !error in distance
psig = data_hist(it_num,1) - BFV
exit
END IF
END DO
WRITE (*,*) "Plusu1\_sigma:^", psig
data_counts = 0.d0 ; mcounts = 0.d0
DO it_num = BFL, 1, -1
mcounts = mcounts + data_hist(it_num,2)
END DO
DO it_num = BFL, 1, -1
data_counts = data_counts + data_hist(it_num,2)!Finds negative one sigma
IF (data_counts .ge. 0.682*mcounts) THEN !error in distance
msig = BFV - data_hist(it_num,1)
exit
END IF
END DO
WRITE (*,*) "Minus_1_sigma:", msig
data_counts = 0.d0 ; pcounts = 0.d0
DO it_num = BFL, data_hist_bins
pcounts = pcounts + data_hist(it_num,2)
END DO
DO it_num = BFL, data_hist_bins
data_counts = data_counts + data_hist(it_num,2)!Finds positive 90% credibility
IF (data_counts .ge. 0.9*pcounts) THEN !error in distance
p90 = data_hist(it_num,1) - BFV
exit
END IF
END DO
WRITE (*,*) "Plus_90%:_", p90
data_counts = 0.d0 ; mcounts = 0.d
DO it_num = BFL, 1, -1
mcounts = mcounts + data_hist(it_num,2)

```
```

END DO
DO it_num = BFL, 1, -1 !
data_counts = data_counts + data_hist(it_num,2)! Finds negative 90% credibility
IF (data_counts .ge. 0.9*mcounts) THEN !error in distance
m90 = BFV - data_hist(it_num,1)
exit
END IF
END DO
WRITE (*,*) "Minus_90%:", m90
data_counts = 0.d0 ; pcounts = 0.d0
DO it_num = BFL, data_hist_bins
pcounts = pcounts + data_hist(it_num , 2)
END DO
DO it_num = BFL, data_hist_bins
data_counts = data_counts + data_hist(it_num,2)!Finds positive 99% credibility
IF (data_counts .ge. 0.99*pcounts) THEN !error in distance
p99 = data_hist(it_num,1) - BFV
exit
END IF
END DO
WRITE (*,*) "Plus 999%:\iota", p99
data_counts = 0.d0 ; mcounts = 0.d0 !
DO it_num = BFL, 1, -1
mcounts = mcounts + data hist(it_num ,2) !
END DO
DO it_num = BFL, 1, -1
data_counts = data_counts + data_hist(it_num,2)!Finds negative 99% credibility
IF (data_counts .ge. 0.99*mcounts) THEN !error in distance
m99 = BFV - data_hist(it_num,1)
exit
END IF
END DO
WRITE (*,*) "Minus_99%:", m99
!|Plot histogram with coloured
!\/credibility intervals
DO it_num = 1, data_hist_bins -1
IF (data_hist(it_num,1) .ge. BFV - msig . and. data_hist(it_num,1).lt. BFV + psig) THEN
CALL pgsci(2)
CALL pgbin (2, data_hist(it_num,1), data_hist(it_num,2),.false.)!
IF (data_hist(it_num,1) .eq. BFV - msig) THEN
xpts = data_hist(it_num,1)
ypts(1)=0.e0 ; ypts(2)= data_hist(it_num,2) !One Sigma
CALL pgline (2, xpts, ypts)
END IF !Credibility
IF (data_hist(it_num+1,1).eq. BFV + psig) THEN
xpts = data_hist(it_num+1,1) !Interval
ypts(1) = 0.e0 ; ypts(2) = data_hist(it_num,2)
CALL pgline (2, xpts, ypts)
END IF
ELSE IF (data_hist(it_num,1) .ge. BFV - m90 . and. data_hist(it_num,1) .lt. BFV + p90) THEN
CALL pgsci(3)
CALL pgbin (2, data_hist(it_num,1), data_hist(it_num, 2),.false.) !
IF (data_hist(it_num,1) .eq. BFV - m90) THEN !
xpts = data_hist(it_num,1)
ypts(1) = 0.e0 ; ypts(2) = data_hist(it_num ,2) !90 percent
CALL pgline (2, xpts, ypts)
END IF !Credibility
IF (data_hist(it_num+1,1).eq. BFV + p90) THEN

```
```

            xpts = data_hist(it_num+1,1) !Interval
            ypts(1) = 0.e0 ; ypts(2) = data_hist(it_num ,2)
            CALL pgline (2, xpts, ypts)
            END IF
    ELSE IF (data_hist(it_num,1) .ge. BFV - m99 .and. data_hist(it_num, 1)
            CALL pgsci(4)
            CALL pgbin (2, data_hist(it_num,1), data_hist(it_num,2),.false.)
            IF (data_hist(it_num,1) .eq. BFV - m99) THEN
            xpts = data_hist(it_num,1)
            ypts(1) = 0.e0 ; ypts(2) = data_hist(it_num,2)
            CALL pgline (2, xpts, ypts)
            END IF
            IF (data_hist(it_num+1,1).eq. BFV + p99) THEN
            xpts = data_hist(it_num +1,1)
            ypts(1) = 0.e0 ; ypts(2) = data_hist(it_num,2)
            CALL pgline (2, xpts, ypts)
            END IF
    ELSE
            CALL pgsci(1)
            CALL pgbin (2, data_hist(it_num,1), data_hist(it_num,2),.false.) !
            IF (it_num .eq. 1) THEN
            xpts = data_hist(it_num,1)
            ypts(1)=0.e0 ; ypts(2)= data_hist(it_num,2)
            CALL pgline (2, xpts, ypts)
    END IF
    IF (it_num .eq. data_hist_bins -1) THEN !outside of 99%
            xpts = data_hist(it_num+1,1) + bw
            ypts(1) = 0.e0 ; ypts(2) = data_hist(it_num +1,2)
            CALL pgline (2, xpts, ypts)
            END IF
    END IF
    END DO
!/\Plot histogram with coloured
!|confidence intervals
END SUBROUTINE HistoPlotAdv

```

\section*{List of Abbreviations}

The following list is neither exhaustive nor exclusive, but may be helpful.
\(2 M A S S \ldots\). The Two-Micron All-Sky Survey
\(\lambda C D M\) or \(C D M\) [Lambda] Cold Dark Matter (cosmological model)
\(A G B \ldots \ldots\). Asymptotic Giant Branch
\(C F H T \ldots \ldots\) The Canada-France-Hawaii Telescope
\(C M D \ldots \ldots\). Colour-Magnitude Diagram
\(L F \ldots \ldots\). Luminosity Function
\(P A n d A S \ldots\). The Pan-Andromeda Archaeological Survey
\(R G B \ldots \ldots\). Red Giant Branch
\(S D S S \ldots .\). The Sloan Digital Sky Survey
\(T R G B \ldots .\). Tip of the Red Giant Branch

\section*{References}

\author{
C. Alcock, R. A. Allsman, D. R. Alves, T. S. Axelrod, A. C. Becker, D. P. Bennett, K. H. Cook, N. Dalal, A. J. Drake, K. C. Freeman, M. Geha, K. Griest, M. J. Lehner, S. L. Marshall, D. Minniti, C. A. Nelson, B. A. Peterson, P. Popowski, M. R. Pratt, P. J. Quinn, C. W. Stubbs, W. Sutherland, A. B. Tomaney, T. Vandehei, and D. Welch. The MACHO Project: Microlensing Results from 5.7 Years of Large Magellanic Cloud Observations. Astrophysical Journal, 542:281-307, October 2000. 17
}
G. W. Angus, A. Diaferio, and P. Kroupa. Using dwarf satellite proper motions to determine their origin. Monthly Notices of the Royal Astronomical Society, 416:1401-1409, September 2011.
M. Bayes and M. Price. An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. Royal Society of London Philosophical Transactions Series I, 53:370-418, 1763. 47
K. Bekki and K. C. Freeman. Formation of \(\omega\) Centauri from an ancient nucleated dwarf galaxy in the young Galactic disc. Monthly Notices of the Royal Astronomical Society, 346:L11-L15, December 2003. 7
M. Bellazzini. The Tip of the Red Giant Branch. Memorie della Societa Astronomica Italiana, 79:440, 2008. 29
M. Bellazzini, F. R. Ferraro, and E. Pancino. A Step toward the Calibration of the Red Giant Branch Tip as a Standard Candle. Astrophysical Journal, 556:635-640, August 2001. 29
R. A. Benjamin, E. Churchwell, B. L. Babler, R. Indebetouw, M. R. Meade, B. A. Whitney, C. Watson, M. G. Wolfire, M. J. Wolff, R. Ignace, T. M. Bania, S. Bracker, D. P. Clemens, L. Chomiuk, M. Cohen, J. M. Dickey, J. M. Jackson, H. A. Kobulnicky, E. P. Mercer, J. S. Mathis, S. R. Stolovy, and B. Uzpen. First GLIMPSE Results on the Stellar Structure of the Galaxy. Astrophysical Journal Letters, 630:L149-L152, September 2005. 3
J. Bland-Hawthorn and K. C. Freeman. Galactic history: formation Memorie della Societa Astronomica Italiana evolution. Memorie della Societa Astronomica Italiana, 77:1095-+, 2006. 13
A. Z. Bonanos, K. Z. Stanek, R. P. Kudritzki, L. M. Macri, D. D. Sasselov, J. Kaluzny, P. B. Stetson, D. Bersier, F. Bresolin, T. Matheson, B. J. Mochejska, N. Przybilla, A. H. Szentgyorgyi, J. Tonry, and G. Torres. The First DIRECT Distance Determination to a Detached Eclipsing Binary in M33. Astrophysical Journal, 652:313-322, November 2006.
A. Brunthaler, M. J. Reid, H. Falcke, L. J. Greenhill, and C. Henkel. The Geometric Distance and Proper Motion of the Triangulum Galaxy (M33). Science, 307:1440-1443, March 2005.
R. G. Carlberg, H. B. Richer, A. W. McConnachie, M. Irwin, R. A. Ibata, A. L. Dotter, S. Chapman, M. Fardal, A. M. N. Ferguson, G. F. Lewis, J. F. Navarro, T. H. Puzia, and D. Valls-Gabaud. Density Variations in the NW Star Stream of M31. Astrophysical Journal, 731:124, April 2011. 26
B. Chaboyer, P. Demarque, P. J. Kernan, and L. M. Krauss. The Age of Globular Clusters in Light of Hipparcos: Resolving the Age Problem? Astrophysical Journal, 494:96-+, February 1998. 6
B. Chaboyer, P. Demarque, and A. Sarajedini. Globular Cluster Ages and the Formation of the Galactic Halo. Astrophysical Journal, 459:558-+, March 1996. 6
R. Cockcroft, W. E. Harris, A. M. N. Ferguson, A. Huxor, R. Ibata, M. J. Irwin, A. W. McConnachie, K. A. Woodley, S. C. Chapman, G. F. Lewis, and T. H. Puzia. The M33 Globular Cluster System with PAndAS Data: the Last Outer Halo Cluster? Astrophysical Journal, 730:112, April 2011. 25
A. R. Conn, R. A. Ibata, G. F. Lewis, Q. A. Parker, D. B. Zucker, N. F. Martin, A. W. McConnachie, M. J. Irwin, N. Tanvir, M. A. Fardal, A. M. N. Ferguson, S. C. Chapman, and D. Valls-Gabaud. A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. II. Distances to the Satellites of M31. Astrophysical Journal, 758:11, October 2012. vi, 68
A. R. Conn, G. F. Lewis, R. A. Ibata, Q. A. Parker, D. B. Zucker, A. W. McConnachie, N. F. Martin, M. J. Irwin, N. Tanvir, M. A. Fardal, and A. M. N. Ferguson. A Bayesian Approach to Locating the Red Giant Branch Tip Magnitude. I. Astrophysical Journal, 740:69, October 2011. vi, 52
A. R. Conn, G. F. Lewis, R. A. Ibata, Q. A. Parker, D. B. Zucker, A. W. McConnachie, N. F. Martin, D. Valls-Gabaud, N. Tanvir, M. J. Irwin, A. M. N. Ferguson, and S. C. Chapman. The Three-dimensional Structure of the M31 Satellite System; Strong Evidence for an Inhomogeneous Distribution of Satellites. Astrophysical Journal, 766:120, April 2013. vi, 94
G. S. Da Costa, T. E. Armandroff, N. Caldwell, and P. Seitzer. The Dwarf Spheroidal Companions to M31: WFPC2 Observations of Andromeda I. Astronomical Journal, 112:2576, December 1996.
G. S. Da Costa, T. E. Armandroff, N. Caldwell, and P. Seitzer. The Dwarf Spheroidal Companions to M31: WFPC2 Observations of Andromeda II. Astronomical Journal, 119: 705-726, February 2000.
J. H. J. de Bruijne. Science performance of Gaia, ESA's space-astrometry mission. Astrophysics and Space Science, 341:31-41, September 2012. 14
G. M. De Silva, K. C. Freeman, J. Bland-Hawthorn, M. Asplund, and M. S. Bessell. Chemically Tagging the HR 1614 Moving Group. Astronomical Journal, 133:694-704, February 2007. 5
G. de Vaucouleurs. Photoelectric photometry of the Andromeda nebula in the UBV system. Astrophysical Journal, 128:465, November 1958.
G. de Vaucouleurs, A. de Vaucouleurs, H. G. Corwin, R. J. Buta, G. Paturel, and P. Fouqué. Third reference catalogue of bright galaxies. Third Reference Catalogue of Bright Galaxies. Springer-Verlag, 1991.
J. Diemand, M. Kuhlen, and P. Madau. Dark Matter Substructure and Gamma-Ray Annihilation in the Milky Way Halo. Astrophysical Journal, 657:262-270, March 2007. 18, 20
P.-A. Duc, J.-C. Cuillandre, P. Serra, L. Michel-Dansac, E. Ferriere, K. Alatalo, L. Blitz, M. Bois, F. Bournaud, M. Bureau, M. Cappellari, R. L. Davies, T. A. Davis, P. T. de Zeeuw, E. Emsellem, S. Khochfar, D. Krajnović, H. Kuntschner, P.-Y. Lablanche, R. M. McDermid, R. Morganti, T. Naab, T. Oosterloo, M. Sarzi, N. Scott, A.-M. Weijmans, and L. M. Young. The ATLAS \({ }^{3 D}\) project - IX. The merger origin of a fast- and a slow-rotating early-type galaxy revealed with deep optical imaging: first results. Monthly Notices of the Royal Astronomical Society, 417:863-881, October 2011.
O. J. Eggen and A. R. Sandage. Stellar groups, IV. The Groombridge 1830 group of high velocity stars and its relation to the globular clusters. Monthly Notices of the Royal Astronomical Society, 119:255-+, 1959. 4

ESA. The Hipparcos and Tycho Catalogues (ESA 1997). VizieR Online Data Catalog, 1239: 0-+, February 1997. 10
A. M. N. Ferguson, J. S. Gallagher, and R. F. G. Wyse. On the Nature of Andromeda IV. Astronomical Journal, 120:821-832, August 2000.
A. M. N. Ferguson, M. J. Irwin, R. A. Ibata, G. F. Lewis, and N. R. Tanvir. Evidence for Substructure in the Halo of M31. In American Astronomical Society Meeting Abstracts
\#200, volume 34 of Bulletin of the American Astronomical Society, page 716, May 2002. 24
S. Fouquet, F. Hammer, Y. Yang, M. Puech, and H. Flores. A possible impact near the Milky Way of a former major merger in the Local Group. In S. Boissier, P. de Laverny, N. Nardetto, R. Samadi, D. Valls-Gabaud, and H. Wozniak, editors, SF2A-2012: Proceedings of the Annual meeting of the French Society of Astronomy and Astrophysics, pages 333-336, December 2012.
K. Freeman and J. Bland-Hawthorn. The New Galaxy: Signatures of Its Formation. Annual Reviews in Astronomy and Astrophysics, 40:487-537, 2002. 2, 3, 4, 5, 6, 7
F. Fusi Pecci, M. Bellazzini, C. Cacciari, and F. R. Ferraro. The Young Globular Clusters of the Milky Way and the Local Group Galaxies: Playing with Great Circles. Astronomical Journal, 110:1664, October 1995.
M. Geha, R. P. van der Marel, P. Guhathakurta, K. M. Gilbert, J. Kalirai, and E. N. Kirby. Local Group Dwarf Elliptical Galaxies. II. Stellar Kinematics to Large Radii in NGC 147 and NGC 185. Astrophysical Journal, 711:361-373, March 2010.
Y. M. Georgelin and Y. P. Georgelin. The spiral structure of our Galaxy determined from H II regions. Astronomy and Astrophysics, 49:57-79, May 1976. 2
G. Gilmore and N. Reid. New light on faint stars. III - Galactic structure towards the South Pole and the Galactic thick disc. Monthly Notices of the Royal Astronomical Society, 202: 1025-1047, March 1983. 2
T. M. Girard, D. I. Dinescu, W. F. van Altena, I. Platais, D. G. Monet, and C. E. López. The Southern Proper Motion Program. III. A Near-Complete Catalog to \(\mathrm{V}=17.5\). Astronomical Journal, 127:3060-3071, May 2004. 12
N. Gouda. Series of JASMINE missions. In EAS Publications Series, volume 45 of EAS Publications Series, pages 393-396, February 2011. 12
N. Gouda, T. Yano, Y. Yamada, Y. Kobayashi, T. Tsujimoto, and The Jasmine Working Group. Japan Astrometry Satellite Mission for Infrared Exploration (JASMINE). In C. Turon, K. S. O'Flaherty, and M. A. C. Perryman, editors, The Three-Dimensional Universe with Gaia, volume 576 of ESA Special Publication, pages 77-+, January 2005. 12
D. O. Gough. Lessons Learned From Solar Oscillations. In T. von Hippel, C. Simpson, and N. Manset, editors, Astrophysical Ages and Times Scales, volume 245 of Astronomical Society of the Pacific Conference Series, pages 31-+, 2001. 5
P. C. Gregory. Bayesian Logical Data Analysis for the Physical Sciences: A Comparative Approach with 'Mathematica' Support. Cambridge University Press, 2005. 43
S. Gwyn. The MegaCam ugriz filter set. 2010. URL http://www2 . cadc-ccda.hia-iha. nrc-cnrc.gc.ca/megapipe/docs/filters.html. 24, 26
F. Hammer, Y. B. Yang, J. L. Wang, M. Puech, H. Flores, and S. Fouquet. Does M31 Result from an Ancient Major Merger? Astrophysical Journal, 725:542-555, December 2010.
F. D. A. Hartwick. The Structure of the Outer Halo of the Galaxy and its Relationship to Nearby Large-Scale Structure. Astronomical Journal, 119:2248-2253, May 2000.
W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. Biometrika, 57(1):97-109, 1970. 43
P. W. Hodge. Distribution of luminosity and color in the galaxy NGC 185. Astronomical Journal, 68:691, November 1963.
E. Høg, C. Fabricius, V. V. Makarov, S. Urban, T. Corbin, G. Wycoff, U. Bastian, P. Schwekendiek, and A. Wicenec. The Tycho-2 catalogue of the 2.5 million brightest stars. Astronomy and Astrophysics, 355:L27-L30, March 2000. 10
E. Holmberg. A study of physical groups of galaxies. Arkiv for Astronomi, 5:305-343, 1969.
F. Hoyle and M. Schwarzschild. On the Evolution of Type II Stars. Astrophysical Journal Supplement Series, 2:1, June 1955. 28
E. P. Hubble. Cepheids in Spiral Nebulae. Popular Astronomy, 33:252, 1925. 27
R. Ibata, S. Chapman, A. M. N. Ferguson, M. Irwin, G. Lewis, and A. McConnachie. Taking measure of the Andromeda halo: a kinematic analysis of the giant stream surrounding M31. Monthly Notices of the Royal Astronomical Society, 351:117-124, June 2004. 23
R. Ibata, M. Irwin, G. Lewis, A. M. N. Ferguson, and N. Tanvir. A giant stream of metal-rich stars in the halo of the galaxy M31. Nature, 412:49-52, July 2001a. 24
R. Ibata, G. F. Lewis, M. Irwin, E. Totten, and T. Quinn. Great Circle Tidal Streams: Evidence for a Nearly Spherical Massive Dark Halo around the Milky Way. Astrophysical Journal, 551:294-311, April 2001b. 3, 21, 22, 23
R. Ibata, N. F. Martin, M. Irwin, S. Chapman, A. M. N. Ferguson, G. F. Lewis, and A. W. McConnachie. The Haunted Halos of Andromeda and Triangulum: A Panorama of Galaxy Formation in Action. Astrophysical Journal, 671:1591-1623, December 2007. 24
R. A. Ibata, G. F. Lewis, A. R. Conn, M. J. Irwin, A. W. McConnachie, S. C. Chapman, M. L. Collins, M. Fardal, A. M. N. Ferguson, N. G. Ibata, A. D. Mackey, N. F. Martin, J. Navarro, R. M. Rich, D. Valls-Gabaud, and L. M. Widrow. A vast, thin plane of corotating dwarf galaxies orbiting the Andromeda galaxy. Nature, 493:62-65, January 2013. 94, 114, 119
R. A. Ibata, G. F. Lewis, M. J. Irwin, and L. Cambrésy. Substructure of the outer Galactic halo from the 2-Micron All-Sky Survey. Monthly Notices of the Royal Astronomical Society, 332:921-927, June 2002a. 9, 21
R. A. Ibata, G. F. Lewis, M. J. Irwin, and T. Quinn. Uncovering cold dark matter halo substructure with tidal streams. Monthly Notices of the Royal Astronomical Society, 332: 915-920, June 2002b. 20
I. Iben, Jr. and A. Renzini. Asymptotic giant branch evolution and beyond. Annual Reviews in Astronomy and Astrophysics, 21:271-342, 1983. xvii, 28
M. J. Irwin. Andromeda XXX/ Cassiopeia II discovery paper - In preparation. 2012. 26
M. J. Irwin, A. M. N. Ferguson, A. P. Huxor, N. R. Tanvir, R. A. Ibata, and G. F. Lewis. Andromeda XVII: A New Low-Luminosity Satellite of M31. Astrophysical Journal Letters, 676:L17-L20, March 2008.
K. V. Johnston, D. N. Spergel, and C. Haydn. How Lumpy Is the Milky Way's Dark Matter Halo? Astrophysical Journal, 570:656-664, May 2002. 20, 21
G. Jungman, M. Kamionkowski, and K. Griest. Supersymmetric dark matter. Physics Reports, 267:195-373, March 1996. 17
M. Jurić et al. The Milky Way Tomography with SDSS. I. Stellar Number Density Distribution. Astrophysical Journal, 673:864-914, February 2008. 2
S. C. Keller, B. P. Schmidt, M. S. Bessell, P. G. Conroy, P. Francis, A. Granlund, E. Kowald, A. P. Oates, T. Martin-Jones, T. Preston, P. Tisserand, A. Vaccarella, and M. F. Waterson. The SkyMapper Telescope and The Southern Sky Survey. Publications of the Astronomical Society of Australia, 24:1-12, May 2007. 9
A. Klypin, H. Zhao, and R. S. Somerville. \(\Lambda\) CDM-based Models for the Milky Way and M31. I. Dynamical Models. Astrophysical Journal, 573:597-613, July 2002. 23
A. Knebe, S. P. D. Gill, B. K. Gibson, G. F. Lewis, R. A. Ibata, and M. A. Dopita. Anisotropy in the Distribution of Satellite Galaxy Orbits. Astrophysical Journal, 603:7-11, March 2004.
Y. Kobayashi, Y. Shimura, Y. Niwa, T. Yano, N. Gouda, and Y. Yamada. Nano-JASMINE: cosmic radiation degradation of CCD performance and centroid detection. In Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, volume 8442 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, September 2012. 12
A. Koch and E. K. Grebel. The Anisotropic Distribution of M31 Satellite Galaxies: A Polar Great Plane of Early-type Companions. Astronomical Journal, 131:1405-1415, March 2006. xix
A. V. Kravtsov, O. Y. Gnedin, and A. A. Klypin. The Tumultuous Lives of Galactic Dwarfs and the Missing Satellites Problem. Astrophysical Journal, 609:482-497, July 2004. 18
P. Kroupa. The Dark Matter Crisis: Falsification of the Current Standard Model of Cosmology. Publications of the Astronomical Society of Australia, 29:395-433, June 2012.
P. Kroupa, B. Famaey, K. S. de Boer, J. Dabringhausen, M. S. Pawlowski, C. M. Boily, H. Jerjen, D. Forbes, G. Hensler, and M. Metz. Local-Group tests of dark-matter concordance cosmology . Towards a new paradigm for structure formation. Astronomy and Astrophysics, 523:A32, November 2010.
P. Kroupa, C. Theis, and C. M. Boily. The great disk of Milky-Way satellites and cosmological sub-structures. Astronomy and Astrophysics, 431:517-521, February 2005.
P.S. Laplace. Théorie analytique des probabilités. Courcier, 1812. URL http://books. google.com.au/books?id=fjAVAAAAQAAJ. 47
M. G. Lee, W. L. Freedman, and B. F. Madore. The Tip of the Red Giant Branch as a Distance Indicator for Resolved Galaxies. Astrophysical Journal, 417:553, November 1993. xvii, 28, 32, 34
B. Letarte, S. C. Chapman, M. Collins, R. A. Ibata, M. J. Irwin, A. M. N. Ferguson, G. F. Lewis, N. Martin, A. McConnachie, and N. Tanvir. A Keck/DEIMOS spectroscopic survey of the faint M31 satellites AndXV and AndXVI. Monthly Notices of the Royal Astronomical Society, 400:1472-1478, December 2009.
N. I. Libeskind, C. S. Frenk, S. Cole, J. C. Helly, A. Jenkins, J. F. Navarro, and C. Power. The distribution of satellite galaxies: the great pancake. Monthly Notices of the Royal Astronomical Society, 363:146-152, October 2005.
M. R. Lovell, V. R. Eke, C. S. Frenk, and A. Jenkins. The link between galactic satellite orbits and subhalo accretion. Monthly Notices of the Royal Astronomical Society, 413: 3013-3021, June 2011.
H.-G. Ludwig, E. Caffau, M. Steffen, P. Bonifacio, and L. Sbordone. Accuracy of spectroscopy-based radioactive dating of stars. Astronomy and Astrophysics, 509:A84, January 2010. 5
D. Lynden-Bell. Dwarf galaxies and globular clusters in high velocity hydrogen streams. Monthly Notices of the Royal Astronomical Society, 174:695-710, March 1976.
D. Lynden-Bell. The Fornax-Leo-Sculptor system. The Observatory, 102:202-208, October 1982. xix
D. Lynden-Bell and R. M. Lynden-Bell. Ghostly streams from the formation of the Galaxy's halo. Monthly Notices of the Royal Astronomical Society, 275:429-442, July 1995.
A. D. Mackey, A. P. Huxor, A. M. N. Ferguson, M. J. Irwin, N. R. Tanvir, A. W. McConnachie, R. A. Ibata, S. C. Chapman, and G. F. Lewis. Evidence for an Accretion Origin for the Outer Halo Globular Cluster System of M31. Astrophysical Journal Letters, 717:L11-L16, July 2010. 25
P. Madau and J. Silk. Population III and the near-infrared background excess. Monthly Notices of the Royal Astronomical Society, 359:L37-L41, May 2005. 19
B. F. Madore, V. Mager, and W. L. Freedman. Sharpening the Tip of the Red Giant Branch. Astrophysical Journal, 690:389-393, January 2009.
S. R. Majewski, R. L. Beaton, R. J. Patterson, J. S. Kalirai, M. C. Geha, R. R. Muñoz, M. S. Seigar, P. Guhathakurta, K. M. Gilbert, R. M. Rich, J. S. Bullock, and D. B. Reitzel. Discovery of Andromeda XIV: A Dwarf Spheroidal Dynamical Rogue in the Local Group? Astrophysical Journal Letters, 670:L9-L12, November 2007.
D. Makarov, L. Makarova, L. Rizzi, R. B. Tully, A. E. Dolphin, S. Sakai, and E. J. Shaya. Tip of the Red Giant Branch Distances. I. Optimization of a Maximum Likelihood Algorithm. Astronomical Journal, 132:2729-2742, December 2006.
N. F. Martin, R. A. Ibata, M. J. Irwin, S. Chapman, G. F. Lewis, A. M. N. Ferguson, N. Tanvir, and A. W. McConnachie. Discovery and analysis of three faint dwarf galaxies and a
globular cluster in the outer halo of the Andromeda galaxy. Monthly Notices of the Royal Astronomical Society, 371:1983-1991, October 2006.
N. F. Martin, A. W. McConnachie, M. Irwin, L. M. Widrow, A. M. N. Ferguson, R. A. Ibata, J. Dubinski, A. Babul, S. Chapman, M. Fardal, G. F. Lewis, J. Navarro, and R. M. Rich. PAndAS' CUBS: Discovery of Two New Dwarf Galaxies in the Surroundings of the Andromeda and Triangulum Galaxies. Astrophysical Journal, 705:758-765, November 2009. 26
A. W. McConnachie. Satellites and Substructure in the Local Group. PhD Thesis; Institute of Astronomy, University of Cambridge, 2005.
A. W. McConnachie. The Pan-Andromeda Archaeological Survey: Galaxy Formation In The Near-Field. In American Astronomical Society Meeting Abstracts \#213, volume 41 of Bulletin of the American Astronomical Society, page \#307.05, January 2009.
A. W. McConnachie. PAndAS: The Pan-Andromeda Archaeological Survey. In StSci 2010 May Symposium, Stellar Populations in the Cosmological Context, May 2010. 25
A. W. McConnachie. The Observed Properties of Dwarf Galaxies in and around the Local Group. Astronomical Journal, 144:4, July 2012. 3, 18
A. W. McConnachie, A. Huxor, N. F. Martin, M. J. Irwin, S. C. Chapman, G. Fahlman, A. M. N. Ferguson, R. A. Ibata, G. F. Lewis, H. Richer, and N. R. Tanvir. A Trio of New Local Group Galaxies with Extreme Properties. Astrophysical Journal, 688:1009-1020, December 2008. 26
A. W. McConnachie and M. J. Irwin. The satellite distribution of M31. Monthly Notices of the Royal Astronomical Society, 365:902-914, January 2006.
A. W. McConnachie, M. J. Irwin, A. M. N. Ferguson, R. A. Ibata, G. F. Lewis, and N. Tanvir. Determining the location of the tip of the red giant branch in old stellar populations: M33, Andromeda I and II. Monthly Notices of the Royal Astronomical Society, 350:243-252, May 2004.
A. W. McConnachie, M. J. Irwin, A. M. N. Ferguson, R. A. Ibata, G. F. Lewis, and N. Tanvir. Distances and metallicities for 17 Local Group galaxies. Monthly Notices of the Royal Astronomical Society, 356:979-997, January 2005.
A. W. McConnachie, M. J. Irwin, R. A. Ibata, J. Dubinski, L. M. Widrow, N. F. Martin, P. Côté, A. L. Dotter, J. F. Navarro, A. M. N. Ferguson, T. H. Puzia, G. F. Lewis, A. Babul, P. Barmby, O. Bienaymé, S. C. Chapman, R. Cockcroft, M. L. M. Collins, M. A. Fardal, W. E. Harris, A. Huxor, A. D. Mackey, J. Peñarrubia, R. M. Rich, H. B. Richer, A. Siebert, N. Tanvir, D. Valls-Gabaud, and K. A. Venn. The remnants of galaxy formation from a panoramic survey of the region around M31. Nature, 461:66-69, September 2009. xviii, 24, 25
C. F. McKee and J. C. Tan. Massive star formation in 100,000 years from turbulent and pressurized molecular clouds. Nature, 416:59-61, March 2002. 5
A. McWilliam and R. M. Rich. The first detailed abundance analysis of Galactic bulge K giants in Baade's window. Astrophysical Journal Supplement Series, 91:749-791, April 1994. 2
B. Méndez, M. Davis, J. Moustakas, J. Newman, B. F. Madore, and W. L. Freedman. Deviations from the Local Hubble Flow. I. The Tip of the Red Giant Branch as a Distance Indicator. Astronomical Journal, 124:213-233, July 2002. xviii, 28, 33
R. B. Metcalf and P. Madau. Compound Gravitational Lensing as a Probe of Dark Matter Substructure within Galaxy Halos. Astrophysical Journal, 563:9-20, December 2001. 19
N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics, 21: 1087-1092, June 1953. 43
M. Metz, P. Kroupa, and H. Jerjen. The spatial distribution of the Milky Way and Andromeda satellite galaxies. Monthly Notices of the Royal Astronomical Society, 374:1125-1145, January 2007.
M. Metz, P. Kroupa, and H. Jerjen. Discs of satellites: the new dwarf spheroidals. Monthly Notices of the Royal Astronomical Society, 394:2223-2228, April 2009a.
M. Metz, P. Kroupa, and N. I. Libeskind. The Orbital Poles of Milky Way Satellite Galaxies: A Rotationally Supported Disk of Satellites. Astrophysical Journal, 680:287-294, June 2008.
M. Metz, P. Kroupa, C. Theis, G. Hensler, and H. Jerjen. Did the Milky Way Dwarf Satellites Enter The Halo as a Group? Astrophysical Journal, 697:269-274, May 2009b.
S. P. Meyn and R. L. Tweedie. Markov chains and stochastic stability. Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1993. ISBN 3-540-19832-6. URL http://probability.ca/MT/. 43
D. Monet, B. Canzian, H. Harris, N. Reid, A. Rhodes, and S. Sell. The PMM USNO-A1.0 Catalogue (Monet 1997). VizieR Online Data Catalog, 1243:0-+, July 1998. 11
D. G. Monet. The 526,280,881 Objects In The USNO-A2.0 Catalog. In Bulletin of the American Astronomical Society, volume 30 of Bulletin of the American Astronomical Society, pages 1427-+, December 1998. 11
B. Moore, J. Diemand, P. Madau, M. Zemp, and J. Stadel. Globular clusters, satellite galaxies and stellar haloes from early dark matter peaks. Monthly Notices of the Royal Astronomical Society, 368:563-570, May 2006. 18
J. Mould and J. Kristian. The dwarf spheroidal galaxy Andromeda I. Astrophysical Journal, 354:438-445, May 1990.
J. F. Navarro, M. G. Abadi, and M. Steinmetz. Tidal Torques and the Orientation of Nearby Disk Galaxies. Astrophysical Journal Letters, 613:L41-L44, September 2004.
B. Nordstrom, M. Mayor, J. Andersen, J. Holmberg, F. Pont, B. R. Jorgensen, E. H. Olsen, S. Udry, and N. Mowlavi. Geneva-Copenhagen Survey of Solar neighbourhood (Nordstrom+, 2004). VizieR Online Data Catalog, 5117:0-+, May 2004. 13
C. Palma, S. R. Majewski, and K. V. Johnston. On the Distribution of Orbital Poles of Milky Way Satellites. Astrophysical Journal, 564:736-761, January 2002.
M. S. Pawlowski, P. Kroupa, G. Angus, K. S. de Boer, B. Famaey, and G. Hensler. Filamentary accretion cannot explain the orbital poles of the Milky Way satellites. Monthly Notices of the Royal Astronomical Society, 424:80-92, July 2012a.
M. S. Pawlowski, P. Kroupa, and K. S. de Boer. Making counter-orbiting tidal debris. The origin of the Milky Way disc of satellites? Astronomy and Astrophysics, 532:A118, August 2011.
M. S. Pawlowski, J. Pflamm-Altenburg, and P. Kroupa. The VPOS: a vast polar structure of satellite galaxies, globular clusters and streams around the Milky Way. Monthly Notices of the Royal Astronomical Society, 423:1109-1126, June 2012b. xix
T. Pearson. PGPLOT: Device-independent Graphics Package for Simple Scientific Graphs. Astrophysics Source Code Library, page 3002, March 2011. Note: The PGPLOT Subroutine library was commenced in 1983. A list of the subroutines can be found at http://www.astro.caltech.edu/ tjp/pgplot/subroutines.html. vii
S. F. Portegies Zwart. The Lost Siblings of the Sun. Astrophysical Journal Letters, 696: L13-L16, May 2009. 5
W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. Numerical recipes in FORTRAN. The art of scientific computing. 1992. viii
P. J. Quinn and J. Goodman. Sinking satellites of spiral systems. Astrophysical Journal, 309: 472-495, October 1986.
B. Reipurth. The Early Sun: Evolution and Dynamic Environment. In A. N. Krot, E. R. D. Scott, and B. Reipurth, editors, Chondrites and the Protoplanetary Disk, volume 341 of Astronomical Society of the Pacific Conference Series, pages 54-+, December 2005. 5
J. C. Richardson, M. J. Irwin, A. W. McConnachie, N. F. Martin, A. L. Dotter, A. M. N. Ferguson, R. A. Ibata, S. C. Chapman, G. F. Lewis, N. R. Tanvir, and R. M. Rich. PAndAS'

Progeny: Extending the M31 Dwarf Galaxy Cabal. Astrophysical Journal, 732:76, May 2011. 26
T. Riehm, E. Zackrisson, O. Möller, E. Mörtsell, and K. Wiik. Prospects for CDM subhalo detection using high angular resolution observations. Journal of Physics Conference Series, 131(1):012045-+, October 2008. 19
A. G. Riess, J. Fliri, and D. Valls-Gabaud. Cepheid Period-Luminosity Relations in the Nearinfrared and the Distance to M31 from the Hubble Space Telescope Wide Field Camera 3. Astrophysical Journal, 745:156, February 2012.
L. Rizzi, R. B. Tully, D. Makarov, L. Makarova, A. E. Dolphin, S. Sakai, and E. J. Shaya. Tip of the Red Giant Branch Distances. II. Zero-Point Calibration. Astrophysical Journal, 661:815-829, June 2007.
C. M. Rockosi, M. Odenkirchen, E. K. Grebel, W. Dehnen, K. M. Cudworth, J. E. Gunn, D. G. York, J. Brinkmann, G. S. Hennessy, and Ž. Ivezić. A Matched-Filter Analysis of the Tidal Tails of the Globular Cluster Palomar 5. Astronomical Journal, 124:349-363, July 2002. xviii
S. Ruphy, A. C. Robin, N. Epchtein, E. Copet, E. Bertin, P. Fouque, and F. Guglielmo. New determination of the disc scale length and the radial cutoff in the anticenter with DENIS data. Astronomy and Astrophysics, 313:L21-L24, September 1996. 2
V. Sahni. Dark Matter and Dark Energy. In E. Papantonopoulos, editor, Lecture Notes in Physics, Berlin Springer Verlag, volume 653 of Lecture Notes in Physics, Berlin Springer Verlag, pages 141-+, 2004. 15, 16, 17, 18
S. Sakai, B. F. Madore, and W. L. Freedman. Tip of the Red Giant Branch Distances to Galaxies. III. The Dwarf Galaxy Sextans A. Astrophysical Journal, 461:713, April 1996. 32, 34
M. Salaris and S. Cassisi. The 'tip' of the red giant branch as a distance indicator: results from evolutionary models. Monthly Notices of the Royal Astronomical Society, 289:406414, August 1997. 27
M. Salaris, S. Cassisi, and A. Weiss. Red Giant Branch Stars: The Theoretical Framework. Publications of the Astronomical Society of the Pacific, 114:375-402, April 2002. 28
D. J. Schlegel, D. P. Finkbeiner, and M. Davis. Maps of Dust Infrared Emission for Use in Estimation of Reddening and Cosmic Microwave Background Radiation Foregrounds. Astrophysical Journal, 500:525, June 1998.

SDSS-III Collaboration, :, C. P. Ahn, R. Alexandroff, C. Allende Prieto, S. F. Anderson, T. Anderton, B. H. Andrews, É. A. S. Bailey, R. Barnes, and et al. The Ninth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the SDSS-III Baryon Oscillation Spectroscopic Survey. ArXiv e-prints, July 2012. 8
A. Siebert, M. E. K. Williams, A. Siviero, W. Reid, C. Boeche, M. Steinmetz, J. Fulbright, U. Munari, T. Zwitter, F. G. Watson, R. F. G. Wyse, R. S. de Jong, H. Enke, B. Anguiano, D. Burton, C. J. P. Cass, K. Fiegert, M. Hartley, A. Ritter, K. S. Russel, M. Stupar, O. Bienaymé, K. C. Freeman, G. Gilmore, E. K. Grebel, A. Helmi, J. F. Navarro, J. Binney, J. Bland-Hawthorn, R. Campbell, B. Famaey, O. Gerhard, B. K. Gibson, G. Matijevič, Q. A. Parker, G. M. Seabroke, S. Sharma, M. C. Smith, and E. Wylie-de Boer. The RAdial Velocity Experiment (RAVE): Third Data Release. Astronomical Journal, 141:187, June 2011. 14
M. F. Skrutskie et al. The Two Micron All Sky Survey (2MASS). Astronomical Journal, 131:1163-1183, February 2006. 9
V. Springel, J. Wang, M. Vogelsberger, A. Ludlow, A. Jenkins, A. Helmi, J. F. Navarro, C. S. Frenk, and S. D. M. White. The Aquarius Project: the subhaloes of galactic haloes. Monthly Notices of the Royal Astronomical Society, 391:1685-1711, December 2008.
K. Z. Stanek and P. M. Garnavich. Distance to M31 with the Hubble Space Telescope and HIPPARCOS Red Clump Stars. Astrophysical Journal Letters, 503:L131, August 1998.
E. Starkenburg, A. Helmi, G. De Lucia, Y.-S. Li, J. F. Navarro, A. S. Font, C. S. Frenk, V. Springel, C. A. Vera-Ciro, and S. D. M. White. The satellites of the Milky Way -
insights from semi-analytic modelling in a \(\Lambda\) CDM cosmology. Monthly Notices of the Royal Astronomical Society, 429:725-743, February 2013.
M. Steinmetz et al. The Radial Velocity Experiment (RAVE): First Data Release. Astronomical Journal, 132:1645-1668, October 2006. 14
G. A. Tammann, A. Sandage, and B. Reindl. The expansion field: the value of \(\mathrm{H}_{0}\). The Astronomy and Astrophysics Review, 15:289-331, July 2008.
M. B. Taylor. TOPCAT Memorie della Societa Astronomica Italiana STIL: Starlink Table/VOTable Processing Software. In P. Shopbell, M. Britton, and R. Ebert, editors, Astronomical Data Analysis Software and Systems XIV, volume 347 of Astronomical Society of the Pacific Conference Series, page 29, December 2005. 91
E. J. Totten and M. J. Irwin. The APM survey for cool carbon stars in the Galactic halo. I. Monthly Notices of the Royal Astronomical Society, 294:1-+, February 1998. 21
S. E. Urban, T. E. Corbin, and G. L. Wycoff. The ACT Reference Catalog. Astronomical Journal, 115:2161-2166, May 1998. 11
J. P. Vallée. The Spiral Arms and Interarm Separation of the Milky Way: An Updated Statistical Study. Astronomical Journal, 130:569-575, August 2005. 3
S. van den Bergh. Search for Faint Companions to M31. Astrophysical Journal Letters, 171: L31, January 1972.
C. A. Vera-Ciro, L. V. Sales, A. Helmi, C. S. Frenk, J. F. Navarro, V. Springel, M. Vogelsberger, and S. D. M. White. The shape of dark matter haloes in the Aquarius simulations: evolution and memory. Monthly Notices of the Royal Astronomical Society, 416:13771391, September 2011.
P. T. Wallace. The SLALIB Library. In D. R. Crabtree, R. J. Hanisch, and J. Barnes, editors, Astronomical Data Analysis Software and Systems III, volume 61 of Astronomical Society of the Pacific Conference Series, page 481, 1994. ix
X. X. Xue, H. W. Rix, G. Zhao, P. Re Fiorentin, T. Naab, M. Steinmetz, F. C. van den Bosch, T. C. Beers, Y. S. Lee, E. F. Bell, C. Rockosi, B. Yanny, H. Newberg, R. Wilhelm, X. Kang, M. C. Smith, and D. P. Schneider. The Milky Way's Circular Velocity Curve to 60 kpc and an Estimate of the Dark Matter Halo Mass from the Kinematics of \({ }^{\sim} 2400\) SDSS Blue Horizontal-Branch Stars. Astrophysical Journal, 684:1143-1158, September 2008. 3
S.-C. Yang and A. Sarajedini. HST/WFPC2 imaging of the dwarf satellites And XI and And XIII: horizontal branch morphology and RR Lyraes. Monthly Notices of the Royal Astronomical Society, 419:1362-1375, January 2012.
B. Yanny et al. SEGUE: A Spectroscopic Survey of 240,000 Stars with \(g=14-20\). Astronomical Journal, 137:4377-4399, May 2009. 13
D. G. York et al. The Sloan Digital Sky Survey: Technical Summary. Astronomical Journal, 120:1579-1587, September 2000. 8
N. Zacharias, S. E. Urban, M. I. Zacharias, G. L. Wycoff, D. M. Hall, D. G. Monet, and T. J. Rafferty. The Second US Naval Observatory CCD Astrograph Catalog (UCAC2). Astronomical Journal, 127:3043-3059, May 2004. 11
A. R. Zentner, A. V. Kravtsov, O. Y. Gnedin, and A. A. Klypin. The Anisotropic Distribution of Galactic Satellites. Astrophysical Journal, 629:219-232, August 2005. xix
M. Zoccali and G. Piotto. Comparison between observed and theoretical Red Giant Branch luminosity functions of galactic globular clusters. Astronomy and Astrophysics, 358:943955, June 2000. 28
D. B. Zucker, A. Y. Kniazev, D. Martínez-Delgado, E. F. Bell, H.-W. Rix, E. K. Grebel, J. A. Holtzman, R. A. M. Walterbos, C. M. Rockosi, D. G. York, J. C. Barentine, H. Brewington, J. Brinkmann, M. Harvanek, S. J. Kleinman, J. Krzesinski, D. Long, E. H. Neilsen, Jr., A. Nitta, and S. A. Snedden. Andromeda X, a New Dwarf Spheroidal Satellite of M31: Photometry. Astrophysical Journal Letters, 659:L21-L24, April 2007.
F. Zwicky. Die Rotverschiebung von extragalaktischen Nebeln. Helvetica Physica Acta, 6: 110-127, 1933. 15```


[^0]:    ${ }^{1}$ The atomic cooling mass $M_{H}$ is the critical mass above which gas can cool efficiently allowing for condensation and subsequent fragmentation via excitation of the Lyman $\alpha$ transition of hydrogen. Assuming a virial temperature above $10^{4} K, M_{H} \approx 10^{8}[(1+z) / 10]^{-3 / 2} M_{\odot}$ which gives $M>0.067 \times 10^{9} M_{\odot}$ at $\mathrm{z}=12$ in order for a luminous component to develop (ref: Madau and Silk (2005))

[^1]:    ${ }^{1}$ Specifically the width between the two inflection points of the Gaussian

[^2]:    ${ }^{2}$ Note that in Paper I, the background fields used were long stripes running along the Galactic Latitude of the object, as can be seen in Paper I, Fig. 3. It was later determined however that smaller fields provided a better approximation to the localized background contained in the object field. Hence, the background fields used for Paper II were rectangles of approximately $2^{\circ} \times 1^{\circ}$ oriented as before with the longer axis parallel to lines of equal Galactic Latitide.

[^3]:    ${ }^{1}$ Due to the advantages of using a single data set for all measurements, only those satellites contained within the PAndAS survey were actually included in the paper. An inner cutoff ellipse around the M31 disk was also necessary due to its obscuring effects (see Fig. 10 (c)).

[^4]:    13 Defined so as to point north in Equatorial coordinates.

