IIT Kanpur Hyperion 2021

Lightening up the Dark Matter

Solution by **Team Glueballs**

Members

Rishabh Mehta[1], Vanshaj Kerni[2], Piyush Marmat[3]

Affiliations:

[1] *Tata Institute of Fundamental Research*
[2] [3] *Indian Institute of Technology Roorkee*

[1] *rmehta@ph.iitr.ac.in*, [2] *vkerni@ph.iitr.ac.in*, [3] *pmarmat@ph.iitr.ac.in*

Contents

Chapter 1

Introduction

The present article is the solution prepared by team Glueballs towards the Hyperion case study competition organized by the Astronomy Club, IIT Kanpur. We have designed a research thesis solution for the problem statement asked in the case study. Here we have described how we attempted to help the case orator *Hyperion* in its journey to shed light on dark matter and its astronomical signatures.

We start with establishing a theoretical framework based on the Classical Newtonian Gravitation, in which the Milky-Way galaxy is modelled as disk shaped structure with all the mass contribution due to the visible matter located at the center (Galactic Center). We then assume the dark matter to be distributed throughout the whole galaxy in a spherically symmetrical distribution and try to develop the framework for calculating the DM density using this model with a thorough discussion on the assumptions and validity of the solutions. The next chapter deals with the solution's experimental, observational and computational aspects with analysis of the provided data sets and appropriate error estimation. We have tried to model and subsequently fit the radial velocities and acceleration of stars due to Dark Matter and normal visible matter added with some astrometrical details. Lastly, in this chapter, we finally estimate the local DM density from the experimental analysis done previously.

Chapter 2

Theoretical Framework

2.1 Gauss's Law

The gravitational field flux through a surface S enclosing volume V with mass distribution as ρ is proportional to the the contained mass inside the volume.

$$\nabla \cdot g = -4\pi G\rho \tag{2.1}$$

2.1.1 Derivation

Let a test mass be at the position vector r while the mass element of a distribution is at r' with distribution given by $\rho(r')$. Then the gravitational field is given as follows using newton's law of gravitation:

$$g = \int G\frac{\rho(r')(\mathbf{r - r'})}{|r - r'|^3}d\tau' \tag{2.2}$$

Taking divergence with respect to r gives

$$\nabla \cdot g = G\int \nabla_r \cdot \frac{\rho(r')(\mathbf{r - r'})}{|r - r'|^3}d\tau \tag{2.3}$$

$$\nabla \cdot g = G\int \rho(r')\nabla_r \cdot \frac{\mathbf{r - r'}}{|r - r'|^3}d\tau' \tag{2.4}$$

Using the standard divergence of $r/|r|^3$ in terms of delta, we have

$$\nabla \cdot g = -4\pi G\int \rho(r')\delta(r - r')d\tau' \tag{2.5}$$

From the property of delta under integral gives the relation

$$\nabla \cdot g = -4\pi G\rho(r) \tag{2.6}$$

This marks our solution of the first question.

2.2 Local Dark Matter Density

2.2.1 Derivation of the formula

The local dark matter contribution to radial acceleration a_{dm} on a star in a universe of just DM is given as

$$\begin{aligned} a_{dm} &= -G \int_R^r \frac{\rho(r')r'^2}{r^2} \, 4\pi dr' \\ \frac{\partial a_{dm}}{\partial r} &= \frac{-4\pi G}{r^2}[\rho(r)r] + \frac{8\pi G}{r^3} \int_R^r \rho(r')r'^2 dr', \end{aligned} \tag{2.7}$$

where R is initial radial position of DM halo.

Replacing the r in the expression with position coordinate for sun r_s and writing neighbourhood dark matter density as ρ_{dm}, we have

$$\frac{\partial a_{dm}}{\partial r} = -4\pi G\rho_{dm} + \frac{2G}{r_s^3} \int_R^{r_s} 4\pi\rho(r')r'^2 dr' \tag{2.8}$$

Assuming that dark matter starts from GC, thereby replacing R=0, we write the expression as

$$\frac{\partial a_{dm}}{\partial r} = -4\pi G\rho_{dm} + \frac{2G}{r_s^3} M_{dm} \tag{2.9}$$

Further assuming that $\Delta r \approx 0$ implying $r_s \approx r_0$; thereby;

$$a_{dm} \approx a_r \implies \frac{\partial a_r}{\partial r} = -4\pi G\rho_0 + \frac{2G}{r_o^3} M_{dm} \tag{2.10}$$

Previous approximation gives $v_s \approx v_0$ for the centripetal acceleration, thereby

$$\frac{GM_{dm}}{r_{dm}^3} = \frac{v_0^2}{r_0^2}, \tag{2.11}$$

Using:

$$\begin{aligned} A &= \frac{1}{2}\left(\frac{V_0}{R_0} - \left.\frac{dv}{dr}\right|_{R_0}\right) \\ B &= -\frac{1}{2}\left(\frac{V_0}{R_0} + \left.\frac{dv}{dr}\right|_{R_0}\right) \end{aligned} \tag{2.12}$$

Where A and B are Oort constants [3], we have:

$$\frac{\partial a_r}{\partial r} = -4\pi G\rho_{dm} + 2(A-B)^2 \tag{2.13}$$

This gives the final expression for the local dark matter density near the Sun as:

$$\rho_{dm} = \frac{1}{4\pi G}\left[2(A-B)^2 - \frac{\partial a_r}{\partial r}\right] \tag{2.14}$$

2.2.2 Discussion on assumptions and the A-B term

In deriving local DM distribution, we neglected the separation between the orbits of sun and star to estimate the DM contribution to sun's radial acceleration in terms of the observed DM acceleration on star. Assumption of circular orbits is implicitly assumed in Oort constants and thereby in the final expression. In this theoretical model, we have only considered the effect of Dark Matter on the acceleration of both the objects; contribution of normal matter towards acceleration via DM interaction with normal matter is not considered. Assumption of spherical symmetry is assumed in deriving the density distribution relation and any vertical (perpendicular to the galactic plane) motion is also neglected.

Oort constants appears in DM distribution relation as $A - B$. This difference term gives the velocity of revolution Ω and therefore an estimation of time for the local neighbourhood of Sun to revolve around the galactic center.

The validity of 2.14 is up to spherically symmetric distribution of DM with assuming negligible contribution of DM-NM (dark matter normal matter) interaction via weak effects on observed acceleration.

This gives the solution for the second problem.

2.3 Numerical analysis of the results

Here we try to derive an expression for the relative uncertainty in the DM density (denoted as a'_r) obtained in eq. 2.14.

$$\delta\rho_{dm} = \frac{\partial\rho_{dm}}{\partial a'_r}\delta a'_r + \frac{\partial\rho_{dm}}{\partial(A-B)}\delta(A-B) \tag{2.15}$$

from eq 2.14, we obtain,

$$\frac{\delta\rho_{dm}}{\rho_{dm}} = \frac{-1}{4\pi G\rho_{dm}}\frac{\delta a'_r}{\rho_{dm}} + \frac{A-B}{\pi G\rho_{dm}}\delta(A-B) \tag{2.16}$$

Chapter 3

Experimental Proceedings

3.1 Modes of Observation

The observations that Hyperion has to make are a set of stars that are bright over a certain threshold (determined by the sensitivity of the available telescope and we expect that Hyperion chooses some space telescope that operates in a broad frequency range to collect more information with lesser noise) and since our objective is to determine radial velocities of these stars over a long period of time (spanning over years), we need high precision for calculating the Doppler shifts in the spectra of these stars (technically called spectro-photometry).

These required precision can be achieved via a laser device called **frequency comb** which is synthesized such that its spectrum has multiple peaks at frequency lines with uniform spacing (This looks like a usual hair comb thus its name). This devices can help Hyperion to achieve the extra precision it requires.

In order to correctly extract the radial velocity and acceleration information from the observed spectra, it is also necessary for Hyperion to be able to separate the apparent radial velocity of any star due to motion of the stars in the background. This issue can be resolved via carefully choosing the star in our observation set specifically those which are not traversing the sky (motion along the celestial sphere) not to fast.

3.2 The Radial Velocity Analysis

We are provided a data-set containing the observed radial velocity (in cm/s) **??** of a star w.r.t. the Galactic Centre (GC) over the course of 10 years. We require to find the velocity contributions due to the MW(Milky Way) gravitational potential (which constitutes contributions from both GC and DM).

The net velocity given in the data also contains influences from exoplanets which are sinusoidal in nature. These give rise to annual periodic patterns. Also, there are multiple sources of noise in data eg noise in instrumentation, stellar background etc. Thus, the net velocity can be written as:

$$v_{net} = v_{MW} + v_{planet} + v_{noise} \tag{3.1}$$

Since we do not know the exact nature of the noise, we neglect it for the time being.

3.2.1 Contribution due to exoplanets

To model the contribution from the exoplanets, we consider only yearly data, as shown in fig. 3.1.

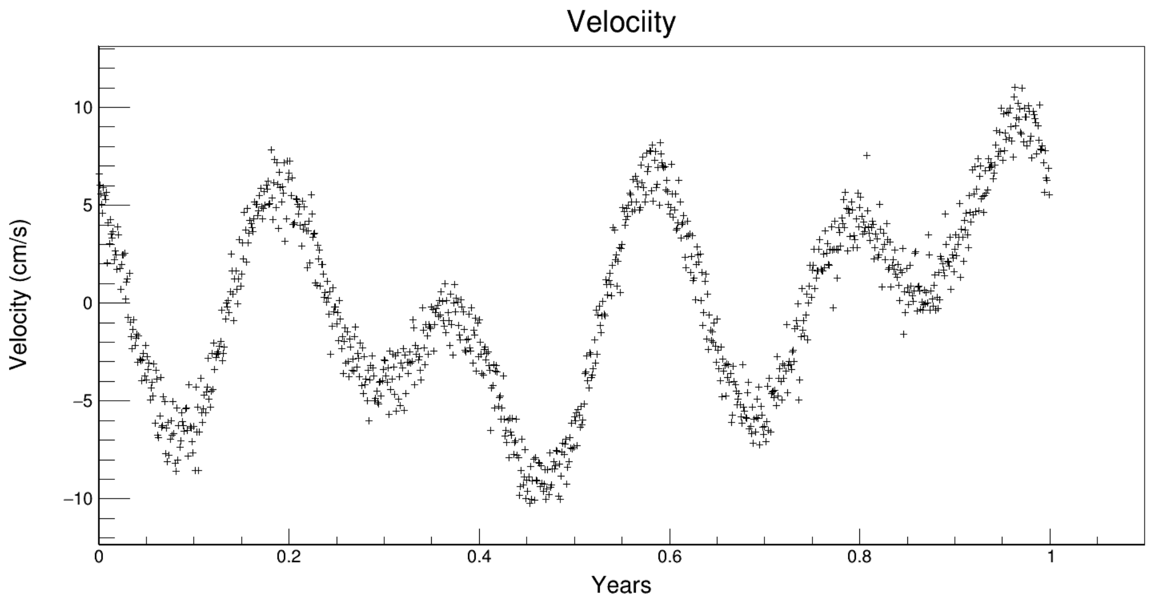

Figure 3.1: Velocity data for one year.

To model this data, we take contributions from 3 exoplanets (inspired from the hint). Each of these contributions is sinusoidal[1], with their own coefficients. Thus, we can have:

$$v_{planet1} = a_1 sin(b_1 t + c_1) \tag{3.2}$$
$$v_{planet2} = a_2 sin(b_2 t + c_2) \tag{3.3}$$
$$v_{planet3} = a_3 sin(b_3 t + c_3) \tag{3.4}$$

Thereby, giving us:

$$\begin{aligned} v_{planet} &= v_{planet1} + v_{planet2} + v_{planet3} \\ &= a_1 sin(b_1 t + c_1) + a_2 sin(b_2 t + c_2) + a_3 sin(b_3 t + c_3) \end{aligned} \tag{3.5}$$

Thereby, we have a net function with 9 floated parameters to be fitted into the data for 1 year (fig. 3.1).

We perform the fit using ROOT^{TM}[2] Data Analysis Framework. We do not rely on the fit probability to ensure the goodness of the fit due to a high noise contamination. We rather use manual inspection to do so. The fitted result is shown in fig. 3.2.

[1]stars with their planetary system revolve around the barycenter and this motion w.r.t. to the GC appears periodic; modelled by simple harmonics

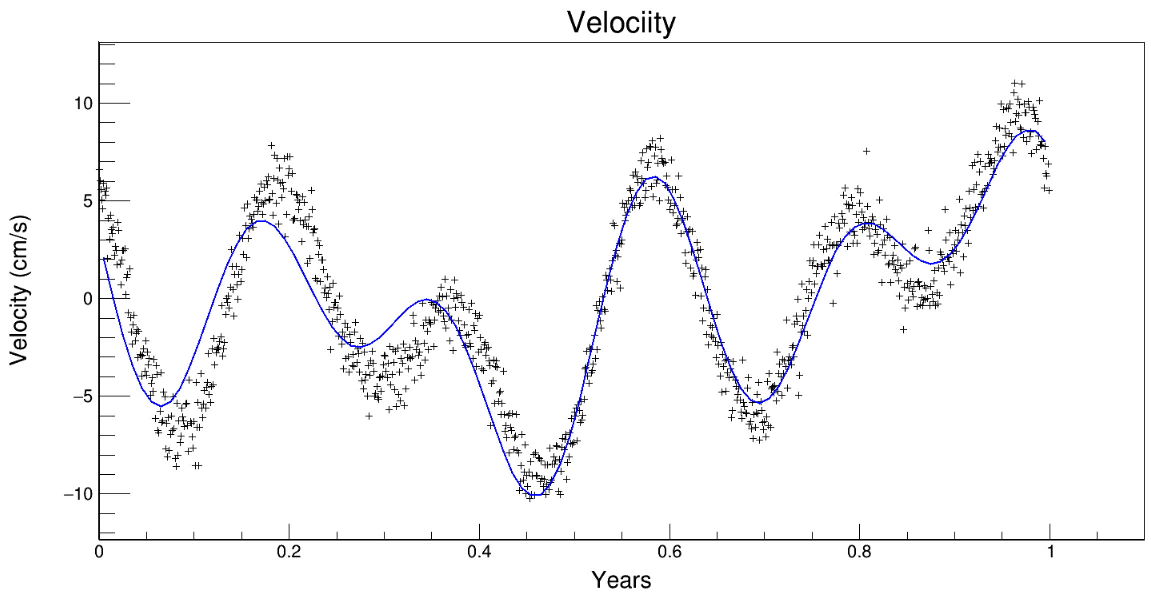

Figure 3.2: Velocity data for one year fitted with sinusoidal function.

The values of parameters obtained by the fit are shown in table 3.1.

Table 3.1: Fit parameters for sinusoidal fit.

Parameter	Value
a_1	3.30
b_1	6.19
c_1	2.07
a_2	4.35
b_2	29.93
c_2	-3.14
a_3	3.82
b_3	18.84
c_3	3.06

Note: The entire code used for this analysis can be found in Appendix A.

3.2.2 Contribution from MW (GC + DM)

Now that we have the exact fit function for the exoplanet contribution, we will float only the contribution from MW gravitational potential in the cumulative data fit for 10 years (Fig. 3.3).

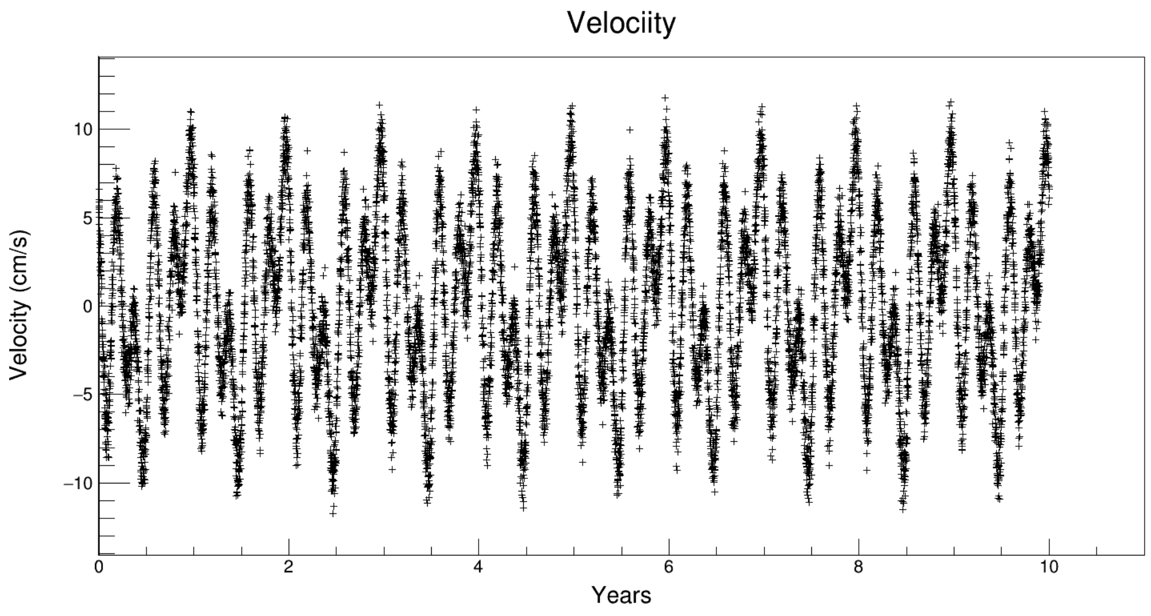

Figure 3.3: Velocity data for 10 years.

The MW gravitational potential contribution should come out as linear (as there is a constant a_r contribution from dark matter and negligible from GC.). Thus, we have:

$$v_{MW} = a_r t \tag{3.6}$$

. Finally, we have:

$$v_{net} = a_r t + f_{planet}(t) \tag{3.7}$$

where $f_{planet}(t)$ is the fit function already determined in previous section. The final fit is done using the ROOT$^{\text{TM}}$[2] Data Analysis Framework is shown in fig 3.4.

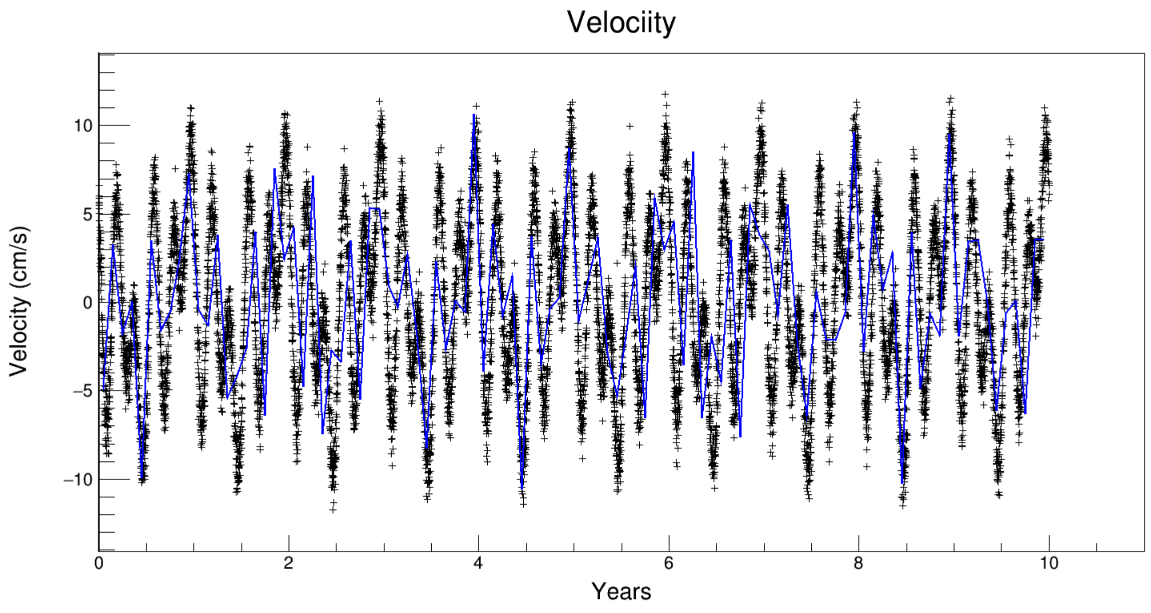

Figure 3.4: Velocity data fo Data Analysis Frameworkr 10 years fitted.

The value of a_r so obtained from the fit is:

$$a_r = 2.6e-2 \pm 7.9e-3 \ \ cms^{-1}year^{-1} \tag{3.8}$$

We acknowledge that our fit is not perfect, and does not take into account the noise in the system. However, it still gives us a non zero value of a_r, enough to show signatures of DM contribution over all the noise.

Note: The entire code used for this analysis can be found in Appendix B.

3.3 The Acceleration Analysis

We are given a data of 1200 stars all within a distance of $3kpc$ from our sun. We need to find the value of acceleration that represents the acceleration of most of the stars in this region. We do this by assigning a mean value to the acceleration distribution given in the data**??**.

By directly looking at the data and by applying some basic mathematical operations for mean (μ) and standard deviation (σ) as:

$$\mu = \sum a_i$$
$$\sigma = \sqrt{\frac{\sum (a_i - \mu)^2}{N}}$$

we get the following values:

$$\mu = 1.517 \quad cms^{-1}year^{-1} \tag{3.9}$$
$$\sigma = 0.612 \quad cms^{-1}year^{-1} \tag{3.10}$$

We now aim to fit a Gaussian PDF over this data to get a more reliable value of μ and σ. For this we construct a histogram with appropriate binning.

3.3.1 Optimal bin calculation

We aim to make a histogram from the provided data that represents the distribution of the given data. For this, we need to decide the optimal binning. We use the **Freedman–Diaconis rule** for this. According to this,

$$h = 2\frac{IQR(x)}{\sqrt[3]{n}} \tag{3.11}$$

, where IQR is the interquartile range of the data and is estimated to be around 1.348σ for a normal distribution. Using Eqn. 3.11 and the preliminary calculations of our σ (Eqn. 3.10), we have our bin width (h) as:

$$h = 0.155 \ cms^{-1}year^{-1} \tag{3.12}$$

Thus, our bin number is:

$$\begin{aligned} n_{bins} &= \frac{Range}{h} \\ &\approx \frac{3.5}{0.155} \\ &\approx 23 \end{aligned} \tag{3.13}$$

3.3.2 Histogram and Fit

We use the ROOT^{TM} [2] data analysis framework to create and fit histograms. The range of the histogram is taken to be [0-3.5] with appropriate units and the bin number is 23 [3.13]. The resulting histogram is shown in figure 3.5.

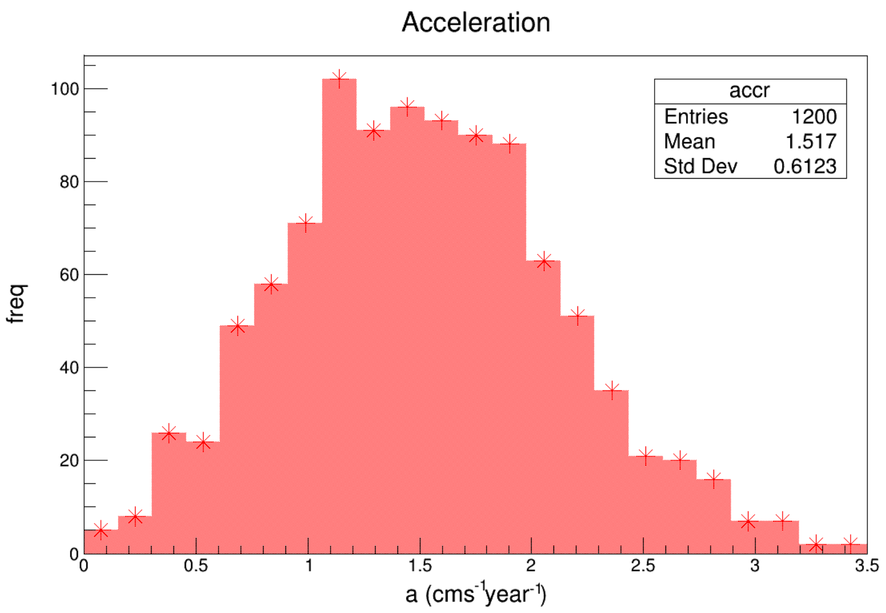

Figure 3.5: Histogram depicting the distribution of acceleration for 1200 stars.

As can be seen, the distribution is fairly Gaussian. We now fit a Gaussian function of the form:

$$f(x) = A\exp\left(-\frac{1}{2}\left(\frac{(x-\mu)}{\sigma}\right)^2\right) \tag{3.14}$$

We use the ROOT^{TM}[2] framework to fit in the histogram, as shown in fig 3.6.
The exact values of χ^2/NDF and Fit probability are:

$$\chi^2/NDF = 0.912$$
$$\text{Fit Probability} = 57.03\%$$

We see that even though the fit probability is not very high, this can be attributed to the lack of data. Overall, by looking at the fit and its χ^2/NDF values, we can say that we have a good fit.

The values of parameters obtained from the Gaussian fit are:

$$A = 99.487 \pm 3.994 \tag{3.15}$$
$$\mu = 1.505 \pm 2.021e-2 \quad cms^{-1}year^{-1} \tag{3.16}$$
$$\sigma = 0.619 \pm 1.634e-2 \quad cms^{-1}year^{-1} \tag{3.17}$$

On comparing these values with Eqns. 3.9 and 3.10, we see a good agreement. Thus we can again say we have a good fit on our distribution.

Note: The entire code used for this analysis can be found in Appendix C.

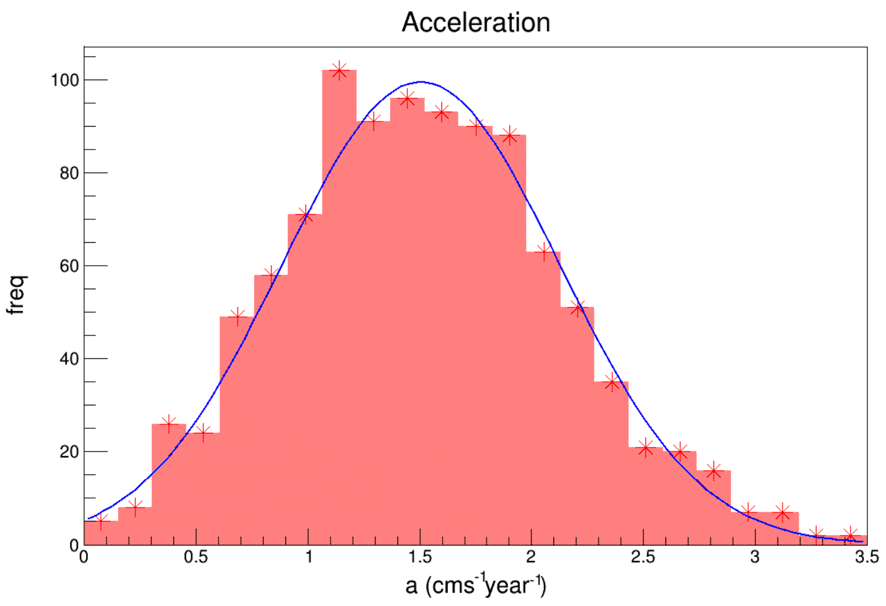

Figure 3.6: Histogram at fig. 3.5 fitted with a Gaussian

3.3.3 Acceleration gradient

Now, our next goal is to calculate $\frac{\partial a}{\partial r}$. To do this, we note that the accelerations here represent change wrt our local acceleration (sun). As we have already assigned a mean value of $a(r)$ to all the stars within a certain range ($\Delta r = r_{avg} = 3kpc$), we can say that:

$$\begin{aligned}\Delta a &= a_{avg}(r) = \mu \\ &= 1.505 \quad cms^{-1}year^{-1} \\ &= 4.771 * 10^{-8} \quad cms^{-2}\end{aligned} \tag{3.18}$$

Thus, the value of acceleration gradient can be found as:

$$\begin{aligned}\frac{\partial a}{\partial r} &= \frac{\Delta a}{\Delta r} \\ &= \frac{4.771 * 10^{-8} \quad cms^{-2}}{3kpc} \\ &= 5.153 \times 10^{-30} s^{-2}\end{aligned} \tag{3.19}$$

3.4 Dark Matter Estimates

We have the values of Oort's constants as [1]:

$$\begin{aligned}A &= 15.3 \pm 0.4 \quad kms^{-1}kpc^{-1} \\ B &= 11.9 \pm 0.4 \quad kms^{-1}kpc^{-1}\end{aligned}$$

Thus, we get the values of $A - B$ as:

$$\begin{aligned}A - B &= 27.2 \pm 0.8 \quad kms^{-1}kpc^{-1} \\ &= (8.8 \pm 0.3)e - 16s^{-1}\end{aligned} \tag{3.20}$$

We do not yet know the direction of $\frac{\partial a_r}{\partial r}$. It is easy to verify, that if this direction is away from the galaxy, we get a negative value of local dark matter density. Thus, we take the direction towards the Galactic center, making the value of $\frac{\partial a_r}{\partial r}$ negative.

Thus, we have, using Eqn's 2.14, 3.20 and 3.19:

$$\begin{aligned}\rho_{DM} &= \frac{1}{4\pi G}(2(A-B)^2 - \frac{\partial a_r}{\partial r}) \\ &= 7.99 \times 10^{-21} kgm^{-3}\end{aligned} \tag{3.21}$$

The errors in this value are:

$$\begin{aligned}\Delta\rho_{DM} &= \frac{1}{4\pi G}[2 * 5.28 + 6.93] \\ &= 0.20 \times 10^{-21} kgm^{-3}\end{aligned} \tag{3.22}$$

These might not be the only sources of uncertainty, for example additional uncertainties might arise from the assumption that the sun star distance is small, or the calculation of G.

Bibliography

[1] J. Bovy. Galactic rotation in Gaia DR1. *Monthly Notices of the Royal Astronomical Society: Letters*, 468(1):L63–L67, 02 2017. ISSN 1745-3925. doi: 10.1093/mnrasl/slx027. URL https://doi.org/10.1093/mnrasl/slx027.

[2] R. Brun and F. Rademakers. Root — an object oriented data analysis framework. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 389(1):81–86, 1997. ISSN 0168-9002. doi: https://doi.org/10.1016/S0168-9002(97)00048-X. URL https://www.sciencedirect.com/science/article/pii/S016890029700048X. New Computing Techniques in Physics Research V.

[3] F. J. Kerr and D. Lynden-Bell. Review of galactic constants. *Monthly Notices of the Royal Astronomical Society*, 221(4):1023–1038, 08 1986. ISSN 0035-8711. doi: 10.1093/mnras/221.4.1023. URL https://doi.org/10.1093/mnras/221.4.1023.

Appendix A

Code Snippet for exoplanet velocity contribution fit.

```
#include <fstream>
#include <iostream>
#include <TCanvas.h>
#include "TLegend.h"
#include "TGraph.h"
#include <cmath>
#include "TMath.h"
#include <string>
#include <TStyle.h>
#include "TAxis.h"
#include <cstring>

using namespace std;

TCanvas* createCanvas(TString name, TString title, Int_t w, Int_t h) {

    TCanvas *c = new TCanvas(name, title, w, h);
    c->SetFillColor(10);
    c->SetBorderMode(0);
    c->SetBorderSize(2);
    c->SetLeftMargin(0.1322082);
    c->SetRightMargin(0.05063291);
    c->SetTopMargin(0.04862579);
    c->SetBottomMargin(0.1501057);
    c->SetFrameFillColor(0);
    c->SetFrameBorderMode(0);
    c->SetFrameBorderMode(0);

    return c;
}

TGraph* createGraph(TString title, TString X_Title, TString Y_Title) {

    // define and style graphs
    TGraph *graph = new TGraph();
    graph->SetTitle(title + ';' + X_Title + ';' + Y_Title);
    graph->SetMarkerColor(1);
    graph->SetMarkerStyle(2);
    graph->GetXaxis()->CenterTitle(true);
    graph->GetXaxis()->SetLabelFont(42);
    graph->GetXaxis()->SetLabelSize(0.03);
    graph->GetXaxis()->SetTitleSize(0.04);
    graph->GetXaxis()->SetTitleFont(42);
    graph->GetYaxis()->CenterTitle(true);
    graph->GetYaxis()->SetLabelFont(42);
    graph->GetYaxis()->SetLabelSize(0.03);
    graph->GetYaxis()->SetTitleSize(0.04);
    graph->GetYaxis()->SetTitleFont(42);

    return graph;
}

Double_t Periodic(Double_t *x, Double_t *par){

    Double_t func1 = par[0]*sin(par[1]*x[0] + par[2]);
    Double_t func2 = par[3]*sin(par[4]*x[0] + par[5]);
    Double_t func3 = par[6]*sin(par[7]*x[0] + par[8]);
    return  func1 + func2 + func3;
}

void vel() {

    string file = "vel_data.csv", line, word;
    vector<string> row;

    ifstream infile(file, ios::in);
    TGraph *vel = createGraph("Velociity", "Years", "Velocity (cm/s)");
    getline(infile, line);

    while (getline(infile, line)) {
        row.clear();
        stringstream s(line);

        while (getline(s, word, ',')) {
            row.push_back(word);
        }
        if (stod(row[0]) <= 1.) {
            vel->AddPoint(stod(row[0]), stod(row[1]));
        }
    }

    TF1 *f = new TF1("f",Periodic,0.,1.,9);
    f->SetParameter(0, 7.);
    f->SetParLimits(0, 1.,10.);
    f->SetParameter(1,1.5*M_PI);
    f->SetParLimits(1, 0, 5*M_PI);
    f->SetParameter(2,M_PI/2);
    f->SetParLimits(2, -M_PI, M_PI);
    f->SetParameter(3, 4.);
    f->SetParLimits(3,0.,8.);
    f->SetParameter(4,8*M_PI);
    f->SetParLimits(4, 7*M_PI, 13*M_PI);
    f->SetParameter(5,-M_PI);
    f->SetParLimits(5, -M_PI, M_PI);
    f->SetParameter(6, 8.);
    f->SetParLimits(6,0.,10.);
    f->SetParameter(7,2*M_PI);
    f->SetParLimits(7, 0, 6*M_PI);
    f->SetParameter(8,0);
    f->SetParLimits(8, -2*M_PI, M_PI);
    f->SetLineStyle(1);
    f->SetLineWidth(2);
    f->SetLineColor(4);

    vel->Fit("f","R");
    TCanvas *c = createCanvas("c","velocity",1600,800);
    vel->Draw("AP");}
```

Appendix B

Code Snippet for final velocity fit.

```
#include <fstream>
#include <iostream>
#include <TCanvas.h>
#include "TLegend.h"
#include "TGraph.h"
#include <cmath>
#include "TMath.h"
#include <string>
#include <TStyle.h>
#include "TAxis.h"
#include <cstring>

using namespace std;

TCanvas* createCanvas(TString name, TString title, Int_t w, Int_t h) {

    TCanvas *c = new TCanvas(name, title, w, h);
    c->SetFillColor(10);
    c->SetBorderMode(0);
    c->SetBorderSize(2);
    c->SetLeftMargin(0.1322082);
    c->SetRightMargin(0.05063291);
    c->SetTopMargin(0.04862579);
    c->SetBottomMargin(0.1501057);
    c->SetFrameFillColor(0);
    c->SetFrameBorderMode(0);
    c->SetFrameBorderMode(0);
    return c;
}

TGraph* createGraph(TString title, TString X_Title, TString Y_Title) {

    // define and style graphs
    TGraph *graph = new TGraph();
    graph->SetTitle(title + ';' + X_Title + ';' + Y_Title);
    graph->SetMarkerColor(1);
    graph->SetMarkerStyle(2);
    graph->GetXaxis()->CenterTitle(true);
    graph->GetXaxis()->SetLabelFont(42);
    graph->GetXaxis()->SetLabelSize(0.03);
    graph->GetXaxis()->SetTitleSize(0.04);
    graph->GetXaxis()->SetTitleFont(42);
    graph->GetYaxis()->CenterTitle(true);
    graph->GetYaxis()->SetLabelFont(42);
    graph->GetYaxis()->SetLabelSize(0.03);
    graph->GetYaxis()->SetTitleSize(0.04);
    graph->GetYaxis()->SetTitleFont(42);

    return graph;
}

Double_t Func(Double_t *x, Double_t *par){
    Double_t a1 = 3.30,
        a2 = 4.35,
        a3 = 3.82,
        b1 = 6.19,
        b2 = 29.93,
        b3 = 18.84,
        c1 = 2.07,
        c2 = -3.14,
        c3 = 3.06;
    Double_t func1 = a1*sin(b1*x[0] + c1);
    Double_t func2 = a2*sin(b2*x[0] + c2);
    Double_t func3 = a3*sin(b3*x[0] + c3);
    Double_t MW = par[0]*x[0];
    return  func1 + func2 + func3 + MW;
}

void a() {

    string file = "vel_data.csv", line, word;
    vector<string> row;

    ifstream infile(file, ios::in);
    TGraph *vel = createGraph("Velociity", "Years", "Velocity (cm/s)");
    getline(infile, line);

    while (getline(infile, line)) {
        row.clear();
        stringstream s(line);

        while (getline(s, word, ',')) {
            row.push_back(word);
        }
        if (stod(row[0]) <= 10.) {
            vel->AddPoint(stod(row[0]), stod(row[1]));
        }
    }

    TF1 *f = new TF1("f",Func,0.,10.,1);
    f->SetParameter(0, 0.5);
    f->SetParLimits(0, -3.5,3.5);
    f->SetLineStyle(1);
    f->SetLineWidth(2);
    f->SetLineColor(4);

    vel->Fit("f","R");
    TCanvas *c = createCanvas("c","velocity",1600,800);
    vel->Draw("AP");
}
```

Appendix C

Code Snippet for Acceleration analysis

```
#include <fstream>
#include <iostream>
#include <TCanvas.h>
#include "TLegend.h"
#include "TGraph.h"
#include <cmath>
#include "TMath.h"
#include <string>
#include <cstring>
#include <TStyle.h>
#include "TAxis.h"
#include "TH1.h"

using namespace std;

TH1F* createHist(
    TString name,
    TString title,
    TString X_Title,
    TString Y_Title,
    Int_t NBINS,
    Float_t lim1,
    Float_t lim2,
    Int_t m_color,
    Int_t m_style,
    Int_t m_size
) {
    // define and style histogram
    TH1F* hist = new TH1F(name, title, NBINS, lim1, lim2);
    hist->SetMarkerColor(m_color);
    hist->SetMarkerStyle(m_style);
    hist->SetMarkerSize(m_size);
    hist->SetFillColor(m_color);
    hist->SetFillStyle(3001);
    hist->GetXaxis()->SetTitle(X_Title);
    hist->GetXaxis()->CenterTitle(true);
    hist->GetXaxis()->SetLabelFont(42);
    hist->GetXaxis()->SetLabelSize(0.03);
    hist->GetXaxis()->SetTitleSize(0.04);
    hist->GetXaxis()->SetTitleFont(42);
    hist->GetYaxis()->SetTitle(Y_Title);
    hist->GetYaxis()->CenterTitle(true);
    hist->GetYaxis()->SetLabelFont(42);
    hist->GetYaxis()->SetLabelSize(0.03);
    hist->GetYaxis()->SetTitleSize(0.04);
    hist->GetYaxis()->SetTitleFont(42);
    hist->SetStats(0);

    return hist;
}

TCanvas* createCanvas(TString name, TString title, Int_t w, Int_t h) {

    TCanvas *c = new TCanvas(name, title, w, h);
    c->SetFillColor(10);
    c->SetBorderMode(0);
    c->SetBorderSize(2);
    c->SetLeftMargin(0.1322082);
    c->SetRightMargin(0.05063291);
    c->SetTopMargin(0.04862579);
    c->SetBottomMargin(0.1501057);
    c->SetFrameFillColor(0);
    c->SetFrameBorderMode(0);
    c->SetFrameBorderMode(0);

    return c;
}

Double_t Gaussian (Double_t *x, Double_t *par) {
    return par[0]*exp(-0.5*(x[0]-par[1])*(x[0]-par[1])/pow(par[2], 2));
}
void accr() {

    string file = "acc_data.csv", line, word;
    vector<string> row;

    ifstream infile(file, ios::in);
    TH1F *accr = createHist(
        "accr",
        "Acceleration",
        "a (cms^{-1}year^{-1})",
        "freq",
        23, 0., 3.5,
        2, 3, 3
    );

    while (getline(infile, line)) {
        row.clear();
        stringstream s(line);

        while (getline(s, word, ',')) {
            row.push_back(word);
        }

        accr->Fill(stof(row[1]));

    }

    TF1 *f = new TF1("f",Gaussian,0.,3.5,3);
    f->SetParameter(0,100.);
    f->SetParLimits(0,50.,150);
    f->SetParameter(1,1.5);
    f->SetParLimits(1, 1., 2.);
    f->SetParameter(2,0.6);
    f->SetParLimits(2, 0.2, 1.);
    f->SetLineStyle(1);
    f->SetLineWidth(2);
    f->SetLineColor(4);

    accr->Fit("f","R");
    double A = f->GetParameter(0);
    double mu = f->GetParameter(1);
    double sigma = f->GetParameter(2);

    cout << f->GetNDF() << endl;
    cout << f->GetChisquare() << endl;
    cout << f->GetProb() << endl;

    TCanvas *c = createCanvas("c","accr",1200,800);
    accr->Draw("BP");}
```