

A comparison of
two approaches for estimating
local stellar kinematics:
The Projection Method versus
Maximum Likelihood

Toktam Aghajani

Lund Observatory
Lund University



2011-EXA51

Degree project of 60 higher education credits (for a degree of Master)
March 2011

Lund Observatory
Box 43
SE-221 00 Lund
Sweden

Abstract

We test a new approach to calculate the mean velocity and velocity dispersion for a sample of nearby stars when their radial velocities are not available. The most commonly used method (here called the projection method, PM) was introduced in a paper by Dehnen & Binney (1998). That method is here compared, theoretically and numerically, with an application of the Maximum Likelihood (ML) method. The two methods are tested on synthetically generated samples as well as on real samples from the Hipparcos Catalogue. In general it turns out that ML is not significantly more accurate than PM, except that ML allows to take into account observational errors and therefore gives more correct dispersions when the uncertainty in the proper motions is significant. Applying PM and ML to samples from the Geneva–Copenhagen survey (Nordström et al. 2004), we find that both methods give very similar results as when the published three-dimensional velocities are used.

Acknowledgements

I owe my deepest gratitude to my supervisor, Lennart Lindegren, whose encouragement, guidance and support made this project possible.

I would also like to extend my gratitude to the wonderful people at Lund Observatory, who made this journey a pleasant one.

Lastly, I would like to thank my dear mother and my loving partner whose love and support allowed me to follow my dreams.

Contents

1	Introduction	1
2	Previous work	7
2.1	Definitions	7
2.2	The Projection Method (PM)	7
2.2.1	Description of the method	7
2.2.2	Deficiencies of the Projection Method	9
2.2.3	An example when the Projection Method fails	9
2.3	Other methods	10
3	A Maximum Likelihood (ML) approach	12
3.1	What is Maximum Likelihood?	12
3.2	Model assumptions	12
3.3	Exact expression for the likelihood	13
3.4	Eliminating the parallaxes	14
3.5	Numerical solution method	15
3.6	A test example	16
4	A comparison of PM and ML results	17
4.1	Generation of data	17
4.1.1	Generating positions	17
4.1.2	Generating the velocity vector	17
4.1.3	Generating the true parallax and true proper motion	18
4.1.4	Generating the observed parallax and proper motion	18
4.1.5	Simulated data sets	18
4.2	Results using PM	19
4.3	Results using ML	19
4.4	Comparison	19
5	Application to real data	24
6	Conclusions	24
A	MATLAB codes	28

1 Introduction

A paper by Dehnen & Binney (1998) explains a simple method to derive local stellar kinematics (mean velocity and velocity dispersion) for a group of stars, when the tangential velocities are known, but not the radial velocities. Since from the Hipparcos Catalogue we can derive the tangential velocity from the parallax and proper motion but not the full 3-dimensional space velocity, this method is clearly very useful. The method has become popular and standard for this purpose and we will refer to it as the projection method (PM). But as derived by Dehnen & Binney, this method is not founded on any particular estimation principle such as the least-squares method, maximum likelihood or Bayesian estimation. We will see later that by using this method we sometimes get an unphysical dispersion matrix. In this project the projection method is compared, theoretically and numerically, with an application of the Maximum Likelihood method (ML). Both methods are tested on synthetically generated samples as well as on real samples from the Hipparcos Catalogue.

The motions of stars relative to the Sun can give us useful information about the origin and history of the stars and also about the structure and evolution of the Galaxy. Throughout this report we use heliocentric galactic coordinates (x, y, z) as defined in Fig. 1. The velocity components are denoted (u, v, w) , (U, V, W) or (v_x, v_y, v_z) .

In order to see why the knowledge about local stellar kinematics is important, we mention six areas of application:

1. *Heating mechanisms in the galactic disk:* The kinematics of stars in the disk in the solar neighbourhood show several interesting systematics, such as the increase of velocity dispersion with age (disk heating; Fig. 2), and that old stars lag behind the young stars in the galactic rotation (asymmetric drift). Disk heating is mainly attributed to the diffusion of stars by massive compact objects like molecular clouds (Hänninen & Flynn 2002) and dynamical interaction with the bar or transient spiral arms (Sellwood 2011). The age-velocity dispersion relation is often given as:

$$\sigma \propto \tau^\alpha, \quad (1.1)$$

where σ is the velocity dispersion (along one axis, or the total dispersion) and τ is the age. α is the heating exponent, which may be different for different heating mechanisms. Different mechanisms may also give different exponents along the three axes. Investigation of this relation therefore can give insight into the importance of different heating mechanisms (Hänninen & Flynn 2002).

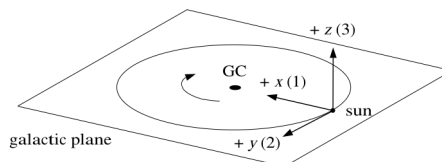


Figure 1: Heliocentric galactic coordinates: x is pointing towards the Galactic center, y in the direction of rotation of the disk and z is normal to the plane of the disk (Lindegren 2010).

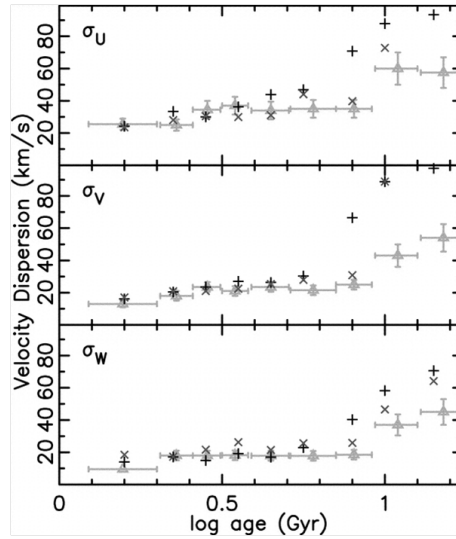


Figure 2: Increase of velocity dispersion with age (Brook et al. 2004). The triangles with error bars show the velocity dispersions in U , V and W versus age for the solar neighbourhood according to Quillen & Garnett (2001). The plus signs and crosses are the results of disk galaxy simulations including merger events at look-back times of ~ 9 Gyr.

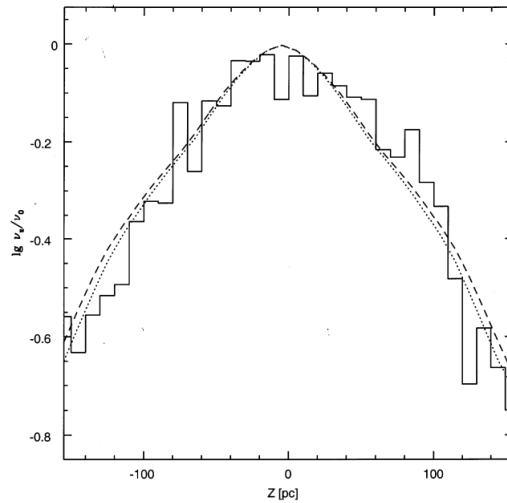


Figure 3: The vertical distribution of A stars in the solar neighbourhood (Holmberg & Flynn 2000). The histogram shows the observed distribution and the curves are the predictions of models with mass density $\rho(0) = 0.103 M_{\odot} \text{pc}^{-3}$ (dotted line) and $\rho(0) = 0.093 M_{\odot} \text{pc}^{-3}$ (dashed line).

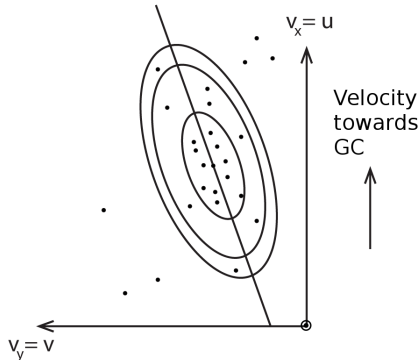


Figure 4: If we plot the velocities of stars relative to the sun in the solar neighbourhood, we can derive the velocity ellipsoid (here shown projected in the (u, v) plane). In this plot the origin represents the velocity of the Sun. To test if the velocity ellipsoid is not aligned with the principal axes of the coordinate system (vertex deviation), we consider correlations such as between u and v as shown in the diagram. It turns out that the average $\langle u(v - \langle v \rangle) \rangle$ is non-zero for many stellar types, which could be explained by a non-axisymmetric potential.

2. *Mapping the local mass density:* By differentiating the vertical Jeans equation with respect to the height above the galactic plane, z , and using Poisson's equation, we get the following expression:

$$4\pi G\rho(z) = -\sigma_z^2 \frac{\partial^2 \ln n(z)}{\partial z^2} - 2(A^2 - B^2), \quad (1.2)$$

(Binney & Tremaine 2008), where σ_z is the velocity dispersion of stars in the direction perpendicular to the galactic plane, $n(z)$ is the number density of stars at z , G is the gravitational constant, $\rho(z)$ is the mass density and A and B are Oort's constants. This allows to calculate the local mass density $\rho(0)$ from the local kinematics (σ_z, A, B) and the distribution of stars (Fig. 3). Using this method it has been shown that, although there is a large amount of dark matter in the halo, the disk is dominated by the visible matter (Holmberg & Flynn 2000; Creze et al. 1989).

3. *The shape of the Milky Way potential:* The velocity distribution of stars in the solar neighbourhood is approximately an ellipsoid with its major axis (vertex) pointing towards the galactic center. The angle between the major axis and the direction towards the galactic center is called the vertex deviation (see Fig. 4). If the Galaxy was axisymmetric, and in a steady state with stars that are distributed randomly in their orbits, we would expect no vertex deviation thanks to the symmetry of the system. So if the vertex deviation differs from zero, it shows that at least one of these conditions does not hold. Vertex deviations of 10–30 degrees (the largest values for the bluest stars) are typically measured (e.g., Bienaymé 1999). Explanations for this fact were given as early as in the 1930's by B. Lindblad and J. Oort (Strömgren 1987). Theorists relate the vertex deviation to the presence of spiral arms.

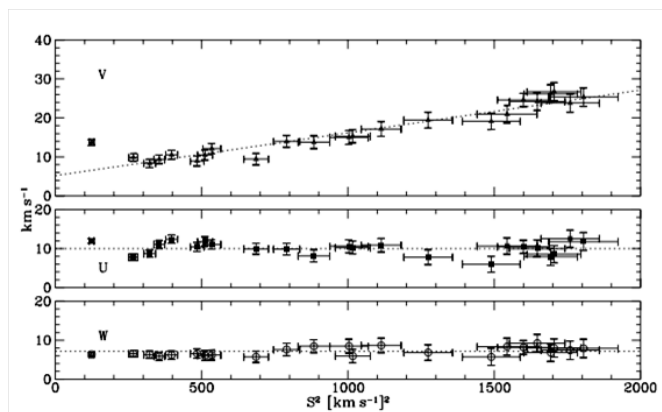


Figure 5: The dependence of the mean U , V and W velocities on the total variance $S^2 = \sigma_U^2 + \sigma_V^2 + \sigma_W^2$ (Dehnen & Binney 1998). For V the dotted line is the fitted linear relation, and for U and W the lines are the mean values for stars bluer than $B - V = 0$. The velocity of the Local Standard of Rest (LSR) relative to the sun is obtained by extrapolating to the case of zero velocity dispersion.

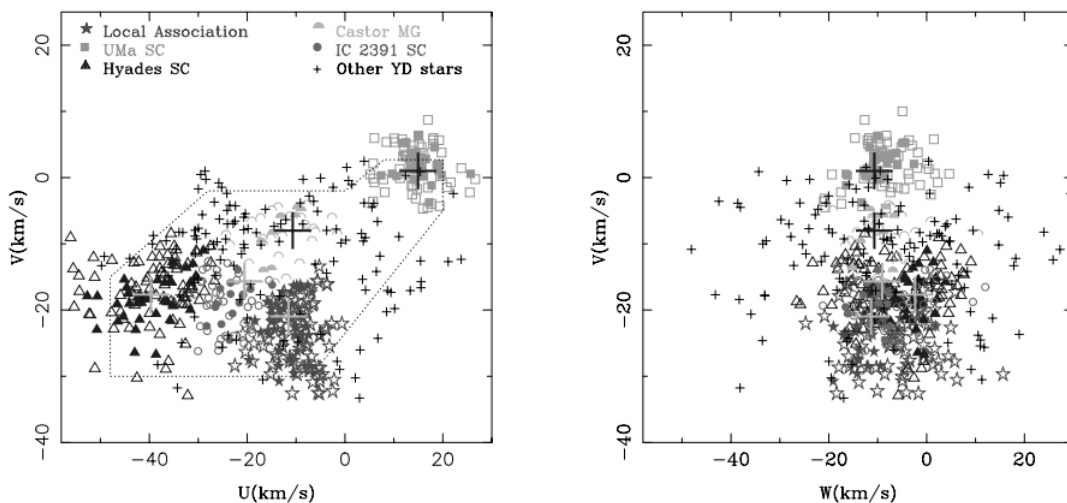


Figure 6: Stellar streams and groups in the solar neighbourhood (Montes et al. 2001). The two diagrams show the (U, V) and (W, V) planes for the velocities of stars that are believed to belong to different stellar kinematic groups, identified by different symbols.

4. *Determination of the Local Standard of Rest (LSR) and the motion of the sun:* We can infer the solar motion relative to the stars from measuring either radial velocities or proper motions of stars. The LSR is defined as the velocity vector for a star, in the solar neighborhood, that has a circular orbit around the galactic center. In practice it is defined by the mean motion of a population with zero random velocities. (Note that this concept assumes an axisymmetric potential.)

We can infer the sun's velocity relative to such a population by extrapolating the observations to the case of zero velocity dispersion (Fig. 5). If we also know the (circular) motion of the LSR, we have complete knowledge of the Sun's galactocentric motion, and its orbit can then be derived, given the galactic potential.

It is interesting to study the sun's motion around the galactic center and perpendicular to the plane. For example, as we pass through the densest part of the plane, gravitational forces from the surrounding giant gas and dust clouds, or stars that accidentally come close to the solar system, may dislodge comets from their paths in the Oort cloud. The comets plunge into the inner part of the solar system, some of them colliding with the Earth. We pass through the galactic plane every 35 to 40 million years, increasing the chances of a comet collision severalfold (Wickramasinghe & Napier 2008). This might be related to mass extinctions, such as that of the dinosaurs 65 million years ago.

5. *Stellar streams and groups:* From stellar kinematics we also can determine the stellar streams and groups (Fig. 6). A stellar group usually refers to a group of stars which moves at the same velocity in the solar neighborhood although they may have different origins. They might be part of a dissolved cluster, or they can be part of a tidal stream after a merger process (Navarro et al. 2004). However, it is also suggested that some (or all) of the stellar kinematic groups in the solar neighbourhood can be created by dynamical resonances of the galactic bar (Dehnen 2000; Fux 2001) or spiral arms (De Simone et al. 2004). Analysis of the stellar groups can thus be used to constrain non-axisymmetric models of the Galaxy (Antoja et al. 2009).
6. *Kinematic populations:* Three different populations of stars in the solar neighbourhood have been identified: thin disk stars with a small velocity dispersion, thick disk stars with a relatively large velocity dispersion and halo stars with a very large velocity dispersion (Fig. 7). The populations also have different mean velocities in the direction of galactic rotation (asymmetric drift). In each population the stellar velocities are smoothly distributed apart from streams and groups which form separate clumps in the space velocity. Knowing the kinematics of these different populations allows us to compute the probability that a star belongs to a certain population (e.g., Bensby et al. 2003), see Fig. 8.

The six areas mentioned above show that it is important to know the stellar kinematics (velocity and velocity dispersion). This thesis is about how to calculate them.

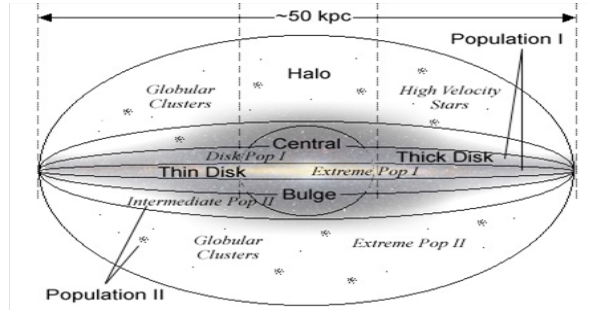


Figure 7: Schematic edge-on view of the Milky Way Galaxy, illustrating the different populations. (Figure from the lecture notes of Prof. Dale E. Gary, New Jersey's Science & Technology University.)

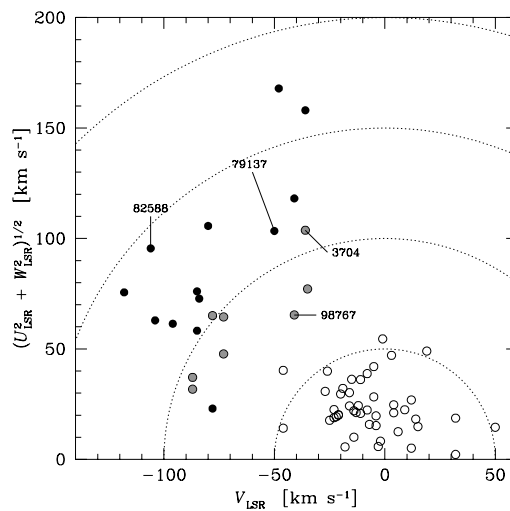


Figure 8: Toomre diagram for a local stellar sample (Bensby et al. 2003). U_{LSR} , V_{LSR} , W_{LSR} are velocities relative to the Local Standard of Rest. Dotted lines indicate constant peculiar space velocities, $v = (U_{LSR}^2 + V_{LSR}^2 + W_{LSR}^2)^{1/2}$, in steps of 50 km s^{-1} . Stars are shown with their Hipparcos numbers. Thin disk stars are marked by empty circles and thick disk stars by filled (black and gray) circles.

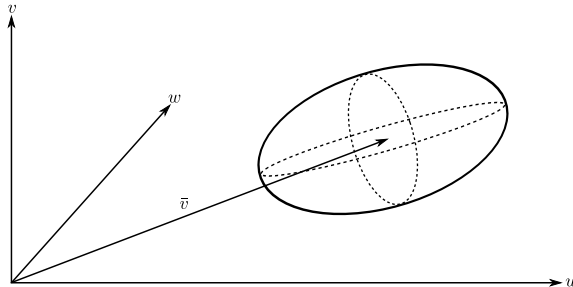


Figure 9: Illustration of the kinematic parameter $\bar{\mathbf{v}}$ (mean velocity) and \mathbf{D} (velocity dispersion). The components of \mathbf{D} define the axes of the velocity ellipsoid (their lengths and orientations).

2 Previous work

2.1 Definitions

For a homogeneous population of stars the phase space density $f(\mathbf{r}, \mathbf{v}, t)$ describes the density of stars as a function of position (\mathbf{r}), velocity (\mathbf{v}) and time t . By the *local kinematics* we mean the distribution function here ($\mathbf{r} = \mathbf{0}$) and now ($t = 0$), that is $f(\mathbf{v}) \equiv f(\mathbf{0}, \mathbf{v}, 0)$. It is usually assumed that $f(\mathbf{v})$ is a smooth function, and the most common assumption is the Schwarzschild approximation, that $f(\mathbf{v})$ is a three-dimensional Gaussian distribution (velocity ellipsoid), or a combination of a few Gaussian distributions. The velocity ellipsoid is completely described by the mean velocity \mathbf{v} and dispersion tensor \mathbf{D} (Fig. 9). These quantities are defined as:

$$\bar{\mathbf{v}} = \langle \mathbf{v} \rangle \quad (2.1)$$

$$\mathbf{D} = \langle (\mathbf{v} - \bar{\mathbf{v}}) \otimes (\mathbf{v} - \bar{\mathbf{v}}) \rangle, \quad (2.2)$$

where \mathbf{v} is the heliocentric velocity of a star and the angular brackets $\langle \cdot \rangle$ denote averages or expected values.¹ Given \mathbf{v}_i , $i = 1, 2, \dots, n$ from a sample of stars we can estimate $\bar{\mathbf{v}}$ and \mathbf{D} as:

$$\bar{\mathbf{v}} = \frac{1}{n} \sum_{i=1}^n \mathbf{v}_i, \quad (2.3)$$

$$\mathbf{D} = \frac{1}{n} \sum_{i=1}^n (\mathbf{v}_i - \bar{\mathbf{v}})(\mathbf{v}_i - \bar{\mathbf{v}})^T \quad (2.4)$$

2.2 The Projection Method (PM)

2.2.1 Description of the method

From the Hipparcos Catalogue we can get the parallaxes and proper motions of the stars near the sun. But the Hipparcos astrometry mission was not complemented by a program to

¹ $\mathbf{a} \otimes \mathbf{b}$ is the outer product of vectors \mathbf{a} and \mathbf{b} , which gives a tensor of rank 2. In the x, y, z coordinates this is the matrix:

$$\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \cdot \begin{pmatrix} b_x & b_y & b_z \end{pmatrix} = \begin{pmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{pmatrix}.$$

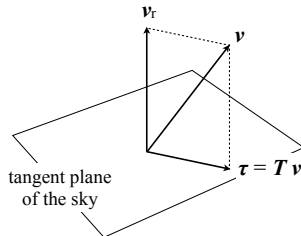


Figure 10: Decomposing the velocity vector \mathbf{v} in the radial (\mathbf{v}_r) and tangential ($\boldsymbol{\tau}$) components. \mathbf{T} is the tangential projection matrix.

measure the radial velocities of the same stars, so we do not know the space velocities of all the stars but only their tangential velocities. In order to calculate the full three-dimensional kinematics we need to use statistics for a sample of stars. Dehnen & Binney (1998) gives the following method to get the mean velocity $\bar{\mathbf{v}}$ and velocity dispersion \mathbf{D} from tangential velocities.

For each star (i), we do not know the full three-dimensional vector \mathbf{v}_i , but only its projection $\boldsymbol{\tau}_i$ on the tangent plane in the sky (Fig. 10). The two vectors are related by:

$$\boldsymbol{\tau}_i = \mathbf{T}_i \mathbf{v}_i \quad , \quad (2.5)$$

where \mathbf{T}_i is the tangential projection matrix given by

$$\mathbf{T}_i = \mathbf{I} - \mathbf{u}_i \otimes \mathbf{u}_i, \quad (2.6)$$

where \mathbf{I} is the identity matrix and \mathbf{u}_i is the unit vector normal to the tangential plane (i.e., unit vector towards the star). \mathbf{T}_i is a singular matrix for any i , so it is not possible to solve \mathbf{v}_i from Equation (2.5). Knowing the tangential velocities for a number of stars scattered on the sky is however enough to compute $\langle \mathbf{v} \rangle$ as follows: \mathbf{T}_i depends only on the position of the star, and if we assume that the velocity distribution is constant within the volume under study, it is statistically independent of \mathbf{v}_i . So taking the average over the stars in the sample gives:

$$\langle \boldsymbol{\tau}_i \rangle = \langle \mathbf{T}_i \mathbf{v}_i \rangle = \langle \mathbf{T}_i \rangle \langle \mathbf{v}_i \rangle. \quad (2.7)$$

We could separate $\langle \mathbf{T}_i \mathbf{v}_i \rangle$ because of the the statistical independence of \mathbf{T}_i and \mathbf{v}_i . The mean velocity can now be obtained as:

$$\bar{\mathbf{v}} = \langle \mathbf{v}_i \rangle = [\langle \mathbf{T}_i \rangle]^{-1} \langle \boldsymbol{\tau}_i \rangle, \quad (2.8)$$

since $\langle \mathbf{T}_i \rangle$ turns out to be non-singular if stars are reasonably scattered on the sky.

We can also obtain a symmetric 3×3 matrix \mathbf{D} for the velocity dispersion, if we consider the tangential component of each star's peculiar velocity

$$\Delta \boldsymbol{\tau}_i = \boldsymbol{\tau}_i - \mathbf{T}_i \bar{\mathbf{v}}, \quad (2.9)$$

which can be computed from the Hipparcos data when we know $\bar{\mathbf{v}}$. By looking at the scatter of $\Delta \boldsymbol{\tau}$ in different directions we can have the full 3×3 matrix for the velocity dispersion. The detailed calculation of \mathbf{D} from $\Delta \boldsymbol{\tau}$ is described in Dehnen & Binney (1998).

Table 1: Positions and velocities for a sample of five stars used to demonstrate that the projection method may give an unphysical dispersion matrix. (l, b) are the galactic longitudes and latitudes, and v_x, v_y, v_z the velocity components along the galactic axes.

i	l	b	v_x	v_y	v_z
–	[rad]	[rad]	[km s ⁻¹]	[km s ⁻¹]	[km s ⁻¹]
1	-0.3409	-0.0609	-43.456	14.209	13.515
2	-2.0634	0.5882	29.433	10.741	3.335
3	-1.3504	-0.0443	2.379	29.042	17.864
4	0.2081	0.3700	-2.289	4.561	2.985
5	-0.6366	0.5388	22.934	33.692	3.087

2.2.2 Deficiencies of the Projection Method

As mentioned before the projection method has been widely used for this purpose. In addition to the original work by Dehnen & Binney (1998), the same method (or variants of it) has been used, e.g., by Mignard (2000), Brosche et al. (2001), van Leeuwen (2007), and Aumer & Binney (2009). Despite of the wide usage of the method, it has deficiencies which could make the method less reliable. Firstly the uncertainty in the data is not considered, meaning that even if we take the case when the true velocity dispersion is zero, because of the observational errors, the velocity dispersion calculated with this method will not be zero. So we probably overestimate the values for the velocity dispersion.

Secondly, when using the projection method it is possible to get a dispersion matrix \mathbf{D} which is not positive-definite.² This is unphysical, because it would mean that the squared velocity dispersion is negative in some direction. The velocity along unit vector \mathbf{u} is $v_u = \mathbf{u}^T \mathbf{v}$. Then:

$$v_u^2 = \mathbf{u}^T \mathbf{v} \mathbf{v}^T \mathbf{u} \Rightarrow \langle v_u^2 \rangle = \mathbf{u}^T \langle \mathbf{v} \mathbf{v}^T \rangle \mathbf{u} = \mathbf{u}^T \mathbf{D} \mathbf{u}, \quad (2.10)$$

which is positive for all directions \mathbf{u} if and only if \mathbf{D} is positive-definite.

2.2.3 An example when the Projection Method fails

We now give a simple example where the projection method gives a dispersion matrix which is not positive-definite. The fewer number of stars we pick for our sample the more likely this will happen. Since there are 9 parameters in $\langle \mathbf{v} \rangle$ and \mathbf{D} and each star gives two components of the tangential velocity, the smallest sample for which we can compute the parameters is $n = 5$ stars. If we can find an example with five stars that gives a negative \mathbf{D} we have shown that the method is, in principle, mathematically invalid. In Table 1 the positions and true velocities of a sample of five stars are shown. The dispersion matrix estimated from full the 3-dimensional vectors is:

$$\mathbf{D}_{3D} = \frac{1}{n} \sum_{i=1}^n (\mathbf{v}_i - \bar{\mathbf{v}})(\mathbf{v}_i - \bar{\mathbf{v}})^T = \begin{pmatrix} 655.010 & 72.796 & -91.224 \\ 72.796 & 122.961 & 22.364 \\ -91.224 & 22.364 & 39.728 \end{pmatrix}. \quad (2.11)$$

By its construction, this matrix is positive definite.

²Mathematically, that \mathbf{D} is positive-definite means that $\mathbf{u}^T \mathbf{D} \mathbf{u} > 0$ for all $\mathbf{u} \neq \mathbf{0}$. For example, the identity matrix \mathbf{I} is positive-definite, since $\mathbf{u}^T \mathbf{I} \mathbf{u} = \mathbf{u}^T \mathbf{u} = \|\mathbf{u}\|^2 > 0$ if $\mathbf{u} \neq \mathbf{0}$.

Using the projection method we get the following dispersion matrix:

$$\mathbf{D}_{\text{PM}} = \begin{pmatrix} 192.224 & 228.333 & -56.623 \\ 228.333 & 144.605 & 58.493 \\ -56.623 & 58.493 & -36.904 \end{pmatrix}. \quad (2.12)$$

This matrix is clearly not positive definite since it has a negative element on its diagonal. In Sect. 3.6 we will try the Maximum Likelihood method on the same sample.

2.3 Other methods

There have been some other attempts to calculate the three-dimensional kinematics by using Hipparcos data (tangential velocity from parallax and proper motion, etc). For example Hogg et al. (2005) developed a general algorithm which works with missing data (in this case the radial velocity). It also calculates the covariance of the measurement uncertainties on the tangential velocity components. They applied their algorithm to a kinematically unbiased sample of 11 865 stars from Hipparcos Catalogue all chosen to have parallaxes measured at signal to noise ratio $\pi/\sigma_\pi > 10$. They modeled each color-selected bin of stars as a mixture of two three-dimensional Gaussian ellipsoids. One Gaussian is for the halo which is fixed at the known mean velocity -220 km s^{-1} in the \mathbf{y} direction and an isotropic dispersion of 100 km s^{-1} . The other Gaussian is an ellipsoid describing the disk population with a mean velocity \mathbf{v}_{disk} (three parameters) and dispersion tensor \mathbf{V}_{disk} (six parameters). The fraction of halo stars is α_{halo} . They also used artificial data sets with the same error properties as Hipparcos to show that the analysis is unbiased. They consider the tangential velocity as the low dimension observations \mathbf{w}_i that are noisy projections of higher dimensions true velocities \mathbf{v}_i drawn independently and identically distributed from a probability density function $p(\mathbf{v})$. A model for $p(\mathbf{v})$ is fitted using projected velocities \mathbf{w}_i , their covariances \mathbf{S}_i , and the projection matrices \mathbf{R}_i . In order to optimize the $p(\mathbf{v})$ (with 10 parameters \mathbf{v}_{disk} , \mathbf{V}_{disk} , α_{halo} for each sample), they see higher dimensional quantities as hidden variables and use an algorithm called EM (expectation-maximization; Dempster et al. 1977) to iteratively maximize the likelihood function. The algorithm starts with the starting parameters given by the user and the iterations generate a sequence of the wanted parameters that monotonically increase the likelihood of a fixed data set under the model. If the derivative of \mathbf{w}_i with respect to the Hipparcos entries \mathbf{c}_i is represented by a matrix \mathbf{Q}_i :

$$d\mathbf{w}_i = \mathbf{Q}_i d\mathbf{c}_i, \quad (2.13)$$

then the measurement uncertainty covariance \mathbf{S}_i for \mathbf{w}_i is given as:

$$\mathbf{S}_i = \mathbf{Q}_i \mathbf{C}_i \mathbf{Q}_i^T, \quad (2.14)$$

where \mathbf{C}_i is the single-star covariance matrix from the Hipparcos Catalogue. This is accurate in small parallax errors and the whole procedure ignores star-to-star covariance since it is not reported in the Hipparcos Catalogue. For the velocity of the sun relative to LSR they find:

$$\mathbf{v}_\odot = \begin{pmatrix} 10.1 \pm 0.5 \\ 4.0 \pm 0.8 \\ 6.7 \pm 0.2 \end{pmatrix} \text{ km s}^{-1} \quad (2.15)$$

where Dehnen & Binney (1998) obtained:

$$\mathbf{v}_\odot = \begin{pmatrix} 10.00 \pm 0.36 \\ 5.25 \pm 0.62 \\ 7.17 \pm 0.38 \end{pmatrix} \text{ km s}^{-1} \quad (2.16)$$

The authors show that their method is insensitive to the assumption of Gaussian velocity distribution. However they claim that the incorrect assumption of Gaussianity enters differently in Dehnen & Binney (1998), which may account for the differences in result.

Our project differs from Hogg et al. (2005) in several respects. For example, we treat the error in parallax differently and also the methods to optimize the likelihood function are different. Hogg et al. (2005) also consider their stars to be from two different populations where we consider only one.

Another paper by Branham (2004) makes a quadric surface fit to the velocity components by means of total least square (TLS). It reduces the data by calculating an ellipsoid which represent a global minimum of the reduction criterion. The reduction technique allows for error in parallax and implements a more rigorous reduction model than statistical moments. The method suffers from severe drawbacks however, and it is not recommended for the general use. Instead Branham (2004) is using a more complicated technique called semidefinite programming, which does not allow for the error in parallax but works to calculate the velocity ellipsoid. The description of the method is however rather obscure and difficult to follow, and it appears that the method has not been used in the literature.

3 A Maximum Likelihood (ML) approach

3.1 What is Maximum Likelihood?

Maximum Likelihood (ML) is a statistical method to fit a probabilistic model to data and estimate the model parameters $\boldsymbol{\theta}$. The likelihood function is defined as follows: Given data \boldsymbol{x} and a probabilistic model $f(\boldsymbol{x}|\boldsymbol{\theta})$ (that is, \boldsymbol{x} has the probability density function $f(\boldsymbol{x}|\boldsymbol{\theta})$ depending on the parameters $\boldsymbol{\theta}$) the likelihood function is $L(\boldsymbol{\theta}|\boldsymbol{x}) = f(\boldsymbol{x}|\boldsymbol{\theta})$. The log-likelihood is $l(\boldsymbol{\theta}|\boldsymbol{x}) = \ln[L(\boldsymbol{\theta}|\boldsymbol{x})]$. The maximum likelihood estimate (MLE) $\hat{\boldsymbol{\theta}}$ is the point in parameter space where the likelihood function (or the log-likelihood) has its global maximum: $L(\hat{\boldsymbol{\theta}}|\boldsymbol{x}) \geq L(\boldsymbol{\theta}|\boldsymbol{x}) \Leftrightarrow l(\hat{\boldsymbol{\theta}}|\boldsymbol{x}) \geq l(\boldsymbol{\theta}|\boldsymbol{x})$ for given data \boldsymbol{x} and any $\boldsymbol{\theta}$. Since $\hat{\boldsymbol{\theta}}$ is a function of the data, the MLE is a statistic (Fig. 11). Normally, it can be obtained by solving the likelihood equations:

$$\frac{\partial l(\boldsymbol{\theta}|\boldsymbol{x})}{\partial \theta_j} = 0, \quad j = 1 \dots m \quad (3.1)$$

where $m = \dim \boldsymbol{\theta}$ is the number of model parameters.

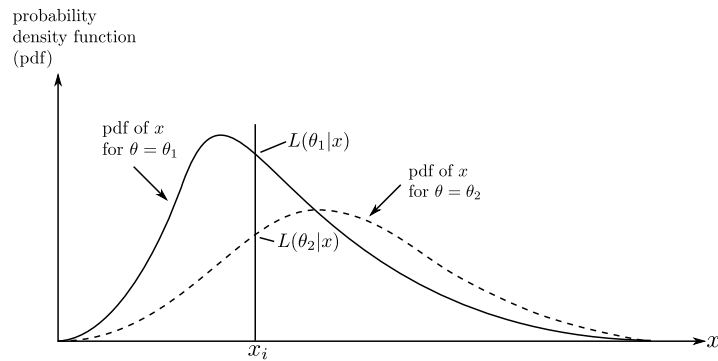


Figure 11: This diagram shows the relation between the probability density function (pdf) of data x , and the likelihood function of a parameter θ . Given the observed data x_i , the model with parameter $\theta = \theta_1$ is more likely than the same model with parameter $\theta = \theta_2$.

3.2 Model assumptions

We assume that the distribution of velocity in all directions and also the distribution of observational errors are Gaussian. That is, the velocity of a star follows a 3-dimensional normal distribution, $\boldsymbol{v} \sim \mathbf{N}(\bar{\boldsymbol{v}}, \boldsymbol{D})$. Given the position and true parallax p of the star we can calculate its true proper motion in longitude and latitude, (μ_l, μ_b) . Adding errors to these we get the observed proper motions $\tilde{\mu}_l \sim N(\mu_l, \sigma_\mu)$, $\tilde{\mu}_b \sim N(\mu_b, \sigma_\mu)$ and the observed parallax $\tilde{p} \sim N(p, \sigma_p)$. The observational uncertainties σ_μ and σ_p are assumed to be known. The ML formulation requires that the probability density function of the observed data $(\tilde{m}u_l, \tilde{\mu}_b, p)$ is written as a function of the model parameters.

3.3 Exact expression for the likelihood

For a problem involving n stars, the parameters of the model are:

- $\bar{\mathbf{v}}$ = the mean velocity of the stellar population (a 3-vector, or 3×1 matrix);
- \mathbf{D} = the dispersion tensor of the stellar population (a symmetric 3×3 matrix, so it contains 6 non-redundant elements);
- \mathbf{p} = the true parallaxes of the stars (an n -vector, or $n \times 1$ matrix).

It is necessary to introduce the true parallaxes p_i as formal model parameters, although the strategy is that they will be eliminated on a star-by-star basis, leaving us with a problem with only nine model parameters, viz. the (non-redundant) components of $\bar{\mathbf{v}}$ and \mathbf{D} . We denote by the vector $\boldsymbol{\theta}$ the complete set of model parameters (i.e., the unknowns to be estimated). For n stars the total number of model parameters is $n + 9$. The observables are, for each star $i = 1 \dots n$, the observed components of proper motion, $\tilde{\mu}_{li}$ and $\tilde{\mu}_{bi}$, and the observed parallax \tilde{p}_i . The total number of observables is $3n$. These have observational uncertainties that are given by the 3×3 covariance matrix \mathbf{C}_i . Sometimes it is useful to denote by the vector \mathbf{x} the complete set of observables (or data).

Given the observations, the likelihood function $L(\boldsymbol{\theta}) \equiv L(\bar{\mathbf{v}}, \mathbf{D}, \mathbf{p})$ numerically equals the probability density function (pdf) $f_{\mathbf{x}}(\mathbf{x}|\boldsymbol{\theta})$ of the observables (\mathbf{x}), given the model parameters ($\boldsymbol{\theta}$). The objective is to find the Maximum Likelihood (ML) estimate of $\boldsymbol{\theta}$, denoted $\hat{\boldsymbol{\theta}}$, i.e., the (hopefully unique) set of parameter values that maximizes $L(\boldsymbol{\theta})$ or, equivalently, the log-likelihood $\ell(\boldsymbol{\theta}) = \ln L(\boldsymbol{\theta})$.

We have the total log-likelihood function

$$\ell(\bar{\mathbf{v}}, \mathbf{D}, \mathbf{p}) = \sum_i [\ln f_{\tilde{\boldsymbol{\mu}},i}(\tilde{\boldsymbol{\mu}}_i|p_i) + \ln g_i(\tilde{p}_i - p_i)] , \quad (3.2)$$

where g_i is the centered normal pdf with standard deviation $\sigma_{p,i} = [\mathbf{C}_i]_{33}^{1/2}$, i.e., the uncertainty of the parallax p_i . It is clear that the parameter p_i only affects the i th term in the sum above. Therefore, when maximizing with respect to p_i we only need to consider that one term. For simplicity we drop, for the moment, subscript i so that the term to consider (for one star) can be written

$$\ell(\bar{\mathbf{v}}, \mathbf{D}, p) = \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p) + \ln g(\tilde{p} - p) , \quad (3.3)$$

where g is the normal pdf with standard deviation σ_p . Note that $f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p)$ of course depends on $\bar{\mathbf{v}}$ and \mathbf{D} as well, although it is only the dependence on p that matters here. In order to make the Maximum Likelihood estimation we need an explicit expression for the pdf $f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p)$ of the observed proper motion $\tilde{\boldsymbol{\mu}}$ when the true parallax p is known. It is assumed to be a two-dimensional normal distribution with expected value

$$\text{E}[\tilde{\boldsymbol{\mu}}] = \mathbf{M}\bar{\mathbf{v}} . \quad (3.4)$$

Here, \mathbf{M} is the 2×3 matrix

$$\mathbf{M} = \frac{p}{K} \begin{bmatrix} -\sin l & \cos l & 0 \\ -\cos l \sin b & -\sin l \cos b & \cos b \end{bmatrix} \quad (3.5)$$

projecting any vector (in this case $\bar{\mathbf{v}}$) to its components in the directions of increasing galactic coordinates l and b , and the factor p/K converts the velocity to proper motion. $K = 4.7405$

is the numerical constant relating mas yr^{-1} to $\text{km s}^{-1} \text{ kpc}^{-1}$. The covariance of $\tilde{\boldsymbol{\mu}}$ is assumed to be the sum of the covariance due to the velocity dispersion and the covariance due to the observational errors, i.e.:

$$\text{Cov}[\tilde{\boldsymbol{\mu}}] \equiv \mathbf{C}_{\tilde{\boldsymbol{\mu}}} = \mathbf{M} \mathbf{D} \mathbf{M}^T + \begin{bmatrix} \sigma_{\mu l}^2 & \rho \sigma_{\mu l} \sigma_{\mu b} \\ \rho \sigma_{\mu l} \sigma_{\mu b} & \sigma_{\mu b}^2 \end{bmatrix}; \quad (3.6)$$

where $\sigma_{\mu l}$ and $\sigma_{\mu b}$ are the observational uncertainties in μ_l and μ_b , and ρ their correlation coefficient. The pdf is then

$$f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p) = (2\pi)^{-1} |\mathbf{C}_{\tilde{\boldsymbol{\mu}}}|^{-1/2} \exp \left[-\frac{1}{2} (\tilde{\boldsymbol{\mu}} - \mathbf{M}\bar{\mathbf{v}})' \mathbf{C}_{\tilde{\boldsymbol{\mu}}}^{-1} (\tilde{\boldsymbol{\mu}} - \mathbf{M}\bar{\mathbf{v}}) \right]. \quad (3.7)$$

3.4 Eliminating the parallaxes

As we see in above equations, we have too many parameters which makes it very difficult to maximize the likelihood function in Eq. (3.2). So we want to eliminate \mathbf{p} by finding the maximum of Eq. (3.3) with respect to \mathbf{p} for each star. Because of the very complicated expression for $f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p)$ it is not likely that this can be done exactly, by analytical means, but we can find an approximate solution valid in the limit of small relative parallax error, $\sigma_p/p \ll 1$. To obtain this, we take the partial derivative of (3.3) with respect to p and set it equal to 0. Using that $g(x) = (2\pi\sigma_p^2)^{-1/2} \exp(-x^2/2\sigma_p^2)$ we find

$$0 = \frac{\partial \ell}{\partial p} = \frac{\partial \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p)}{\partial p} + \frac{1}{\sigma_p^2} (\tilde{p} - p), \quad (3.8)$$

from which

$$p = \tilde{p} + \sigma_p^2 \frac{\partial \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p)}{\partial p}. \quad (3.9)$$

So far no approximation has been made. However, the derivative on the right hand side is to be evaluated for the *true* parallax, which means that we have not really solved the problem. However, if $\sigma_p \ll p$ we can approximate the derivative by its value at \tilde{p} , rather than p . This is safe to do under the assumption of small relative parallax error, since it means that \tilde{p} is always positive. (It would not have been possible to evaluate $f_{\tilde{\boldsymbol{\mu}}}$ for zero or negative p .) Thus we have the *approximation*

$$p \simeq \tilde{p} + \sigma_p^2 \left(\frac{\partial \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p)}{\partial p} \right)_{p=\tilde{p}}, \quad (3.10)$$

which is obviously correct to first order in σ_p^2 . Let us introduce the function

$$F(p) \equiv \frac{\partial \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p)}{\partial p} \quad (3.11)$$

so that the result can be written

$$p \simeq \tilde{p} + \sigma_p^2 F(\tilde{p}). \quad (3.12)$$

Since our objective was to eliminate p from the expression for the log-likelihood function, let us now introduce the approximate result (3.12) into the log-likelihood term (3.3). Using the first-order Taylor expansion

$$\ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p) \simeq \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|\tilde{p}) + (p - \tilde{p}) F(\tilde{p}) \quad (3.13)$$

the total likelihood function becomes:

$$\ell(\bar{\mathbf{v}}, \mathbf{D}) \simeq \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|p) - \frac{(\tilde{p} - p)^2}{2\sigma_p^2} - \frac{1}{2} \ln(2\pi\sigma_p^2) \quad (3.14)$$

Replacing from Eq. (3.12) we get:

$$\ell(\bar{\mathbf{v}}, \mathbf{D}) \simeq \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|\tilde{p}) + \sigma_p^2 F(\tilde{p})^2 - \frac{\sigma_p^4 F(\tilde{p})^2}{2\sigma_p^2}. \quad (3.15)$$

Here we have neglected the additive constant $-\frac{1}{2} \ln(2\pi\sigma_p^2)$, since it does not depend on the model parameters and therefore is irrelevant for the ML problem. So the likelihood equation for one star finally becomes:

$$\ell(\bar{\mathbf{v}}, \mathbf{D}) \simeq \ln f_{\tilde{\boldsymbol{\mu}}}(\tilde{\boldsymbol{\mu}}|\tilde{p}) + \frac{1}{2}\sigma_p^2 F(\tilde{p})^2. \quad (3.16)$$

Recalling that this was just the log-likelihood term for one star, the total log-likelihood function, after maximization with respect to \mathbf{p} , is therefore:

$$\ell(\bar{\mathbf{v}}, \mathbf{D}) \simeq \sum_i \left[\ln f_{\tilde{\boldsymbol{\mu}},i}(\tilde{\boldsymbol{\mu}}_i|\bar{\mathbf{v}}, \mathbf{D}, \tilde{p}_i) + \frac{1}{2}\sigma_{p,i}^2 F_i(\tilde{p}_i)^2 \right]. \quad (3.17)$$

3.5 Numerical solution method

Since the derivatives of Eq. (3.17) becomes very complicated and almost impossible to solve, we had to use numerical methods instead. So a MATLAB program was written to make the ML estimate of our model parameters, $\bar{\mathbf{v}}$ and \mathbf{D} . In order to maximize Eq. (3.17) we use the simplex method called `fminsearch` in MATLAB. This is a numerical minimization algorithm in which we do not need to have the derivatives. The simplex algorithm is well explained in Numerical Recipes (Press et al. 2007).

As soon as we have enough stars in the sample the `fminsearch` algorithm works fine but we have two situations in which the algorithm does not work properly. The first case happens when we have very small parallaxes or very large tangential velocities for some stars. The second case is when we have a small sample of stars. In both situations it can happen that the estimated dispersion becomes zero in some direction. That is because sometimes the measurement errors are enough to explain the dispersion and then `fminsearch` will try to make the contribution from \mathbf{D} in that direction equal to zero. The result is a semi-definite dispersion matrix, which `fminsearch` cannot handle.³

In order to compute a solution even in these cases, we introduced a regularization parameter $\alpha > 0$. The regularized log-likelihood is:

$$\ell(\bar{\mathbf{v}}, \mathbf{D}) \simeq \sum_i \left[\ln f_{\tilde{\boldsymbol{\mu}},i}(\tilde{\boldsymbol{\mu}}_i|\bar{\mathbf{v}}, \mathbf{D}, \tilde{p}_i) + \frac{1}{2}\sigma_{p,i}^2 F_i(\tilde{p}_i)^2 \right] - \alpha \ln \frac{S_{\max}}{S_{\min}}, \quad (3.18)$$

where S_{\max} and S_{\min} are the extreme singular values of the singular value decomposition⁴ of \mathbf{D} . S_{\max}/S_{\min} is the square of the ratio of the longest and shortest axes of the velocity

³It is not clear if it is the mathematical problem itself that is degenerate, or if it is only the numerical optimization that fails. It would have been interesting to try a different optimization algorithm, if time had permitted.

⁴Singular value decomposition (SVD) is a factorization of a real or complex matrix, with many applications in data analysis. The SVD of \mathbf{D} is $\mathbf{D} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$, where \mathbf{U} is an orthogonal matrix and $\boldsymbol{\Sigma}$ is a diagonal matrix of singular values. S_{\max} and S_{\min} are the maximum and minimum diagonal elements of $\boldsymbol{\Sigma}$. For a dispersion matrix, the singular values equal the variances along the principal axes of the dispersion ellipsoid.

ellipsoid (Fig. 12). If α is large `fminsearch` will try to make $S_{\max} = S_{\min}$, and we get an isotropic velocity dispersion in the limit of large α . In order to use as little regularization as necessary, we try Eq. (3.18) with $\alpha = 0$ and, if diverging, increase α by steps of 0.5 until `fminsearch` converges. The result is a dispersion matrix that is always positive-definite, but when $\alpha > 0$ it is more isotropic than it should be (Table 2).

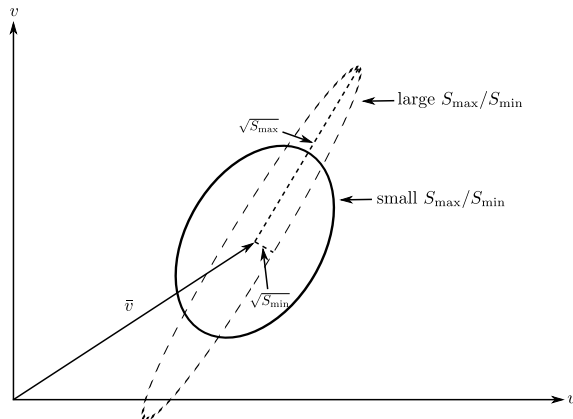


Figure 12: Regularization tends to make estimated velocity ellipsoid more isotropic by reducing the ratio S_{\max}/S_{\min} , which is the square of the ratio of the longest to the shortest axis.

Table 2: An example with 100 stars illustrating how the estimated velocity dispersion depends on the value of α . In this case $\alpha = 0$ would actually be used (no regularization). The solutions for $\alpha > 0$ were only made to show that the velocity dispersion becomes more isotropic when regularization is used.

parameters:	\bar{u}	\bar{v}	\bar{w}	σ_u	σ_v	σ_w
true values	10.000	15.000	7.000	22.000	14.000	10.000
$\alpha = 0.0$	10.090	14.775	7.083	21.712	13.636	9.783
$\alpha = 0.5$	10.033	14.808	7.090	21.610	13.638	9.902
$\alpha = 1.0$	10.032	14.806	7.091	21.429	13.619	10.041
$\alpha = 1.5$	10.031	14.805	7.093	21.251	13.604	10.182
$\alpha = 2.0$	10.029	14.804	7.094	21.076	13.591	10.324
$\alpha = 10.0$	10.015	14.824	7.143	18.587	13.845	12.819

Also to calculate the derivative in Eq. (3.11) we need to do it numerically in MATLAB. The following approximation is used:

$$\frac{\partial \ln f_{\tilde{\mu}}(\tilde{\mu}|p)}{\partial p} \simeq \frac{f_{\tilde{\mu}}(\tilde{\mu}|p + \sigma_p) - f_{\tilde{\mu}}(\tilde{\mu}|p - \sigma_p)}{2\sigma_p} \quad (3.19)$$

This is done in the MATLAB function `dpdf`.

3.6 A test example

We apply the ML method to the same test example as in Sect. 2.2.3. With the projection method we got the unphysical dispersion matrix in Eq. (2.12). With the ML method we get

a positive-definite matrix, but unfortunately we need regularization to prevent divergence. In this example we needed to set $\alpha = 2$ to get the numerical algorithm to converge. The resulting dispersion matrix is:

$$\mathbf{D}_{\text{ML}} = \begin{pmatrix} 83.061 & 31.615 & 0.017 \\ 31.615 & 80.574 & 0.016 \\ 0.017 & 0.016 & 50.178 \end{pmatrix} \quad (3.20)$$

Comparing with \mathbf{D}_{PM} from Sect. 2.2.3, it is not obvious that the ML result is better in this case, even if the matrix is positive-definite.

4 A comparison of PM and ML results

Here we test both methods, using much larger synthetic samples with known mean velocity and dispersion matrix.

4.1 Generation of data

The generation of synthetic data follows the probabilistic model outlined in Sect. 3.2. Observational uncertainties were set to $\sigma_\mu = 1, 3, 10$ and 30 mas yr^{-1} for the proper motions, and to $\sigma_p = 1 \text{ mas}$ for the parallaxes.

4.1.1 Generating positions

We started by simulating the random positions for 2000 stars with a uniform density within a radius of $R = 100 \text{ pc}$ from the sun. First we generated random coordinates x, y, z with a uniform density in $[-R, R]$ for each axis. Then we calculated the distance $r = \sqrt{x^2 + y^2 + z^2}$ from the sun and rejected the point if $r > R$. This was repeated until we had 2000 accepted points. From the rectangular coordinates the galactic longitude l and galactic latitude b are obtained as follows:

$$l = \text{atan2}(y, x) \quad (4.1)$$

$$b = \text{atan2}(z, \sqrt{x^2 + y^2}) \quad (4.2)$$

Then for any position in sky (l, b) we define three orthogonal unit vectors,

$$\mathbf{u} = \begin{bmatrix} \cos b \cos l \\ \cos b \sin l \\ \sin b \end{bmatrix} \quad \mathbf{l} = \begin{bmatrix} -\sin l \\ \cos l \\ 0 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} -\sin b \cos l \\ -\sin b \sin l \\ \cos b \end{bmatrix} \quad (4.3)$$

(Fig. 13), where \mathbf{u} is the direction towards the star, and \mathbf{l} and \mathbf{b} are unit vectors in the tangential plane in the sky, $+\mathbf{l}$ in the direction of increasing longitude, and $+\mathbf{b}$ in the direction of increasing latitude. $[\mathbf{u}, \mathbf{l}, \mathbf{b}]$ is called the normal galactic triad at the point (l, b) .

4.1.2 Generating the velocity vector

We randomized the velocity of each star using a normal distribution with the mean parameter $\bar{\mathbf{v}} = (10, 15, 7)^T \text{ km s}^{-1}$ and dispersion matrix $\mathbf{D} = \text{diag}(22^2, 14^2, 10^2) \text{ km}^2 \text{ s}^{-2}$.

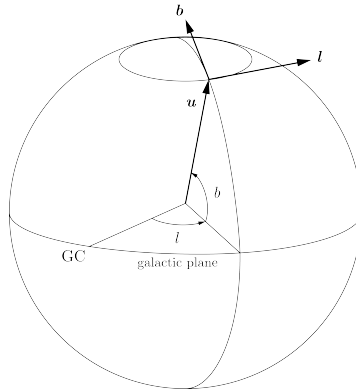


Figure 13: Definition of the normal galactic triad $[\mathbf{u}, \mathbf{l}, \mathbf{b}]$ for the arbitrary point (l, b) on the sky (Lindegren 2010). GC is the direction towards the galactic center, l is the galactic longitude and b the galactic latitude.

4.1.3 Generating the true parallax and true proper motion

The true parallax is calculated as:

$$p = \frac{1000 \text{ mas}}{r} \quad (4.4)$$

and the true proper motion in longitude and latitude as:

$$\mu_l = \frac{p}{K} \mathbf{l} \cdot \mathbf{v} \quad \mu_b = \frac{p}{K} \mathbf{b} \cdot \mathbf{v} \quad (4.5)$$

where $K = 4.7405 \text{ km s}^{-1} \text{ kpc}^{-1} (\text{mas yr}^{-1})^{-1}$.

4.1.4 Generating the observed parallax and proper motion

In order to get the observed proper motions and parallax we need to add the errors in proper motion and parallax to the true values. Errors were drawn from normal distributions with zero mean value and standard deviation equal to the observational uncertainties σ_μ and σ_p .

4.1.5 Simulated data sets

The simulated data set depends on many different parameters so it is not possible to investigate the PM and ML methods for all possible combination of the parameters. We decided to keep most of the parameters fixed ($\bar{\mathbf{v}}$, \mathbf{D} , R and σ_p), and only study how the methods perform as a function of the number of stars in the sample (n) and the uncertainty of the proper motions (σ_μ). We used $n = 30, 100, 300$ and 1000 , and $\sigma_\mu = 1, 3, 10$ and 30 mas yr^{-1} , which gives 16 different combinations. For each combination we generated 100 samples by using different seeds for the random generator. This means that the 100 samples are different both in the generated true positions and velocities, and in the observational errors. Then we applied the PM and ML methods to all 1600 samples and calculated mean values and RMS variations of the estimated parameters. The results are shown in Tables 3–4 and further analyzed below.

4.2 Results using PM

Using the projection method to calculate velocity and velocity dispersion for the simulated data gives the results in Table 3 and Figs. 15 and 16 (left columns). From the table and diagrams we note two things:

1. The larger the uncertainty in proper motion is, the larger the dispersions become (they are more biased). The effect is negligible when $\sigma_\mu = 1$ or 3 mas yr^{-1} , but for $\sigma_\mu = 10$ or 30 mas yr^{-1} the increase in the dispersions is very significant. This can be understood since an error of 10 mas yr^{-1} corresponds to a linear velocity of about 5 km s^{-1} at a distance of 100 pc (the maximum distance in the sample).
2. For the smallest samples ($n = 30$) the resulting dispersion matrix is unphysical in about 1% of the cases (see column 3 in Table 3). For the larger samples it never happened.

Otherwise, the PM method gives good results in nearly all cases.

4.3 Results using ML

Using the Maximum Likelihood method to calculate velocity and velocity dispersion for the simulated data gives the results in Table 4 and Figs. 15 and 16 (right columns). From the table and diagrams we note:

1. The ML method is able to give unbiased dispersions even for the largest σ_μ . This shows that the method corrects for the uncertainty in the proper motion data, which was not the case for the PM method.
2. For the smallest samples ($n = 30$) and smallest σ_μ , the ML method sometimes required regularization. In most of the cases when regularization was applied, it was sufficient to use $\alpha = 0.5$, which did not change the axis ratio very much. Only in very few cases was $\alpha > 3$ needed, which gave a nearly isotropic dispersion.

Except for the cases when regularization was needed, the ML method gives good results.

4.4 Comparison

Comparing the PM and ML results, they are in general quite similar. The ML method is not significantly more accurate than the PM method, except in the cases when σ_μ is large. The ML method is able to take into account the observational errors and therefore gives more correct dispersions (smaller bias) in those cases. However, ML sometimes has divergence problem.

As already mentioned, the minimum number of stars that we can calculate the mean velocity and velocity dispersion for is $n = 5$ (since we have 9 parameters and 2 proper motion components per star, we must have $2n > 9$). As we can see in Fig 14, the more stars we have (more data), the more accurate values we get for our parameters. This is true for both methods.

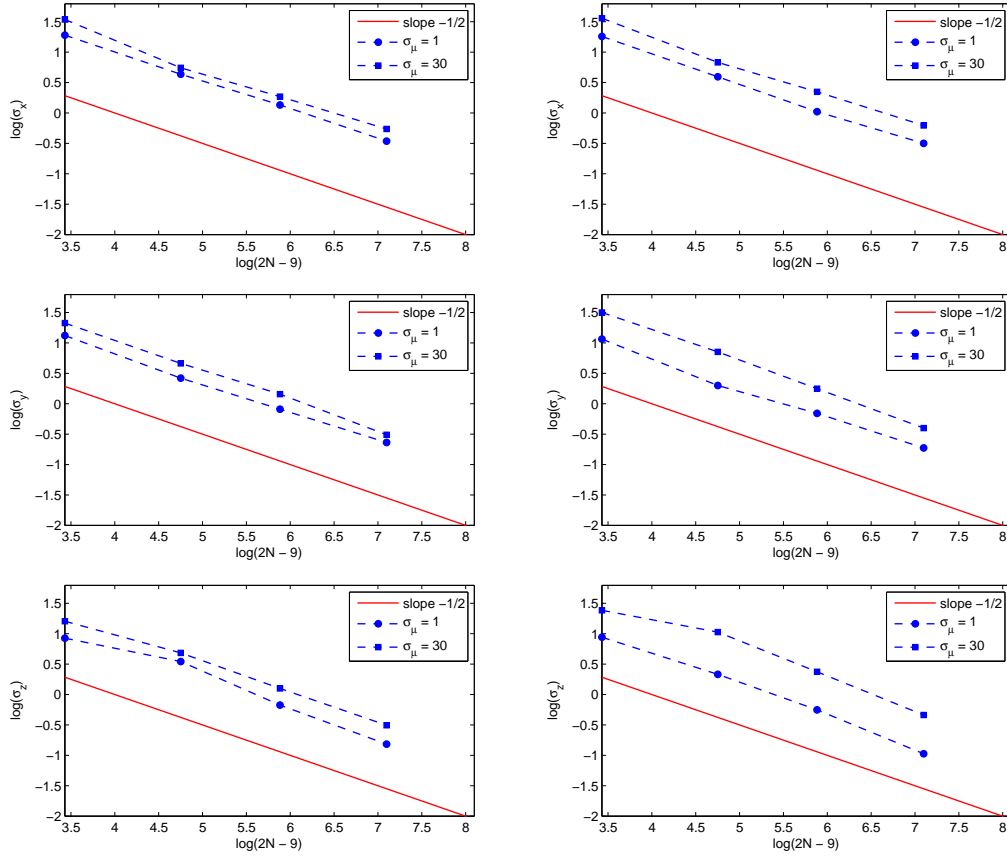


Figure 14: Results illustrating how the uncertainty in dispersion depends on the number of stars (n) in the sample. The left diagrams are for the PM method, the right diagrams for the ML method. The diagrams show the uncertainty in σ_u , σ_v and σ_w versus $\log(2n-9)$ for $\sigma_\mu = 1$ and 30 mas yr^{-1} . The larger the sample is, the more accurate values we get for our parameters in both methods. The solid line has the theoretically expected slope, $-1/2$.

Table 3: Results for synthetic samples using the projection method (PM). n is the number of stars, σ_μ the uncertainty in proper motion and P is the percentage of cases where PM fails. The following columns give the mean velocities \bar{u} , \bar{v} , \bar{w} and the velocity dispersions σ_u , σ_v , σ_w (in km s^{-1}) and the correlation coefficients $\rho_{ij} = D_{ij}(D_{ii}D_{jj})^{-1/2}$ (where $i, j = u, v, w$) from the estimated dispersion matrix. Numbers after \pm show the RMS scatter between different results, indicating the precision of the estimated values.

n	σ_μ	P	\bar{u}	\bar{v}	\bar{w}	σ_u	σ_v	σ_w	ρ_{uu}	ρ_{uv}	ρ_{uw}	ρ_{vw}
True values												
30	1	1%	9.985 \pm 4.527	15.063 \pm 3.333	6.641 \pm 3.117	20.692 \pm 3.593	13.443 \pm 3.079	9.929 \pm 2.525	-0.066 \pm 0.339	-0.042 \pm 0.378	-0.042 \pm 0.378	0.019 \pm 0.379
30	3	1%	9.976 \pm 0.378	15.066 \pm 0.378	6.631 \pm 3.118	20.695 \pm 3.633	13.476 \pm 3.102	9.971 \pm 2.586	-0.068 \pm 0.338	-0.045 \pm 0.381	-0.045 \pm 0.381	0.017 \pm 0.378
30	10	1%	9.944 \pm 4.587	15.077 \pm 3.410	6.599 \pm 3.177	20.883 \pm 3.817	13.897 \pm 3.209	10.517 \pm 2.781	-0.071 \pm 0.330	-0.049 \pm 0.363	-0.049 \pm 0.363	0.013 \pm 0.374
30	30	0%	9.870 \pm 5.025	15.126 \pm 3.935	6.489 \pm 3.749	22.898 \pm 4.656	17.243 \pm 3.775	14.537 \pm 3.336	-0.068 \pm 0.318	-0.041 \pm 0.315	-0.041 \pm 0.315	0.012 \pm 0.364
100	1	0%	10.092 \pm 2.720	14.982 \pm 1.713	7.115 \pm 1.600	22.097 \pm 1.896	14.109 \pm 1.521	9.944 \pm 1.723	0.016 \pm 0.154	0.005 \pm 0.155	0.005 \pm 0.155	0.037 \pm 0.183
100	3	0%	10.091 \pm 2.726	14.978 \pm 1.717	7.106 \pm 1.601	22.117 \pm 1.884	14.144 \pm 1.519	9.989 \pm 1.755	0.014 \pm 0.155	0.004 \pm 0.155	0.004 \pm 0.155	0.036 \pm 0.178
100	10	0%	10.088 \pm 2.768	14.966 \pm 1.774	7.073 \pm 1.639	22.374 \pm 1.865	14.551 \pm 1.560	10.563 \pm 1.819	0.005 \pm 0.155	0.001 \pm 0.153	0.001 \pm 0.153	0.034 \pm 0.163
100	30	0%	10.077 \pm 3.061	14.929 \pm 2.226	6.979 \pm 2.005	24.592 \pm 2.104	17.782 \pm 1.946	14.776 \pm 1.986	-0.012 \pm 0.160	0.002 \pm 0.147	0.002 \pm 0.147	0.026 \pm 0.158
300	1	0%	10.067 \pm 1.469	15.188 \pm 0.962	6.940 \pm 0.899	22.199 \pm 1.139	14.123 \pm 0.915	10.145 \pm 0.840	0.007 \pm 0.082	0.002 \pm 0.095	0.002 \pm 0.095	0.005 \pm 0.115
300	3	0%	10.076 \pm 1.473	15.190 \pm 0.973	6.938 \pm 0.906	22.225 \pm 1.135	14.163 \pm 0.917	10.201 \pm 0.858	0.006 \pm 0.081	0.002 \pm 0.095	0.002 \pm 0.095	0.006 \pm 0.114
300	10	0%	10.110 \pm 1.497	15.199 \pm 1.034	6.932 \pm 0.952	22.506 \pm 1.140	14.600 \pm 0.946	10.801 \pm 0.916	0.003 \pm 0.077	0.003 \pm 0.095	0.003 \pm 0.095	0.009 \pm 0.111
300	30	0%	10.206 \pm 1.634	15.222 \pm 1.354	6.914 \pm 1.225	24.821 \pm 1.307	17.966 \pm 1.172	15.056 \pm 1.104	-0.004 \pm 0.071	0.007 \pm 0.097	0.007 \pm 0.097	0.016 \pm 0.106
1000	1	0%	9.977 \pm 0.729	15.045 \pm 0.557	6.927 \pm 0.466	22.225 \pm 0.629	14.207 \pm 0.529	10.149 \pm 0.441	0.007 \pm 0.044	0.002 \pm 0.053	0.002 \pm 0.053	0.009 \pm 0.055
1000	3	0%	9.977 \pm 0.729	15.048 \pm 0.563	6.927 \pm 0.468	22.251 \pm 0.635	14.241 \pm 0.524	10.205 \pm 0.445	0.007 \pm 0.043	0.002 \pm 0.052	0.002 \pm 0.052	0.009 \pm 0.055
1000	10	0%	9.974 \pm 0.737	15.059 \pm 0.592	6.910 \pm 0.488	22.538 \pm 0.662	14.659 \pm 0.518	10.807 \pm 0.467	0.006 \pm 0.042	0.001 \pm 0.050	0.001 \pm 0.050	0.006 \pm 0.054
1000	30	0%	9.966 \pm 0.816	15.088 \pm 0.743	6.861 \pm 0.634	24.884 \pm 0.768	17.985 \pm 0.601	15.071 \pm 0.602	0.003 \pm 0.039	0.001 \pm 0.050	0.001 \pm 0.050	0.002 \pm 0.052

Table 4: Results for synthetic samples using the Maximum Likelihood method (ML). n is the number of stars, σ_μ the uncertainty in proper motion and P the percentage of cases that needed regularization. The following columns are as in Table 3.

n	σ_μ	P	\bar{u}	\bar{v}	\bar{w}	σ_u	σ_v	σ_w	ρ_{uu}	ρ_{uv}	ρ_{uw}	ρ_{vw}
True values												
30	1	13%	9.805 \pm 4.567	14.662 \pm 3.548	6.560 \pm 2.968	20.526 \pm 3.517	13.285 \pm 2.899	9.918 \pm 2.572	-0.032 \pm 0.299	-0.032 \pm 0.369	-0.032 \pm 0.369	-0.015 \pm 0.375
30	3	13%	9.783 \pm 4.581	14.664 \pm 3.563	6.532 \pm 2.970	20.538 \pm 3.612	13.304 \pm 2.984	9.898 \pm 2.652	-0.035 \pm 0.305	-0.037 \pm 0.380	-0.037 \pm 0.380	-0.015 \pm 0.381
30	10	19%	9.767 \pm 4.679	14.759 \pm 3.705	6.459 \pm 3.010	20.543 \pm 3.990	13.371 \pm 3.440	9.873 \pm 2.920	-0.04 \pm 0.440	-0.032 \pm 0.440	-0.032 \pm 0.440	-0.045 \pm 0.460
30	30	68%	9.921 \pm 5.161	15.018 \pm 4.067	6.408 \pm 3.430	19.578 \pm 4.770	13.131 \pm 4.480	9.817 \pm 4.000	-0.029 \pm 0.560	-0.054 \pm 0.560	-0.054 \pm 0.560	0.013 \pm 0.590
100	1	0%	10.033 \pm 2.698	14.809 \pm 1.776	7.088 \pm 1.499	21.794 \pm 1.816	13.659 \pm 1.353	9.764 \pm 1.397	0.009 \pm 0.136	-0.007 \pm 0.148	-0.007 \pm 0.148	-0.011 \pm 0.172
100	3	0%	10.041 \pm 2.703	14.811 \pm 1.774	7.083 \pm 1.502	21.797 \pm 1.807	13.655 \pm 1.367	9.760 \pm 1.412	0.007 \pm 0.136	-0.006 \pm 0.148	-0.006 \pm 0.148	-0.011 \pm 0.170
100	10	0%	10.073 \pm 2.745	14.837 \pm 1.803	7.063 \pm 1.535	21.815 \pm 1.829	13.659 \pm 1.493	9.730 \pm 1.547	0.003 \pm 0.142	-0.004 \pm 0.159	-0.004 \pm 0.159	-0.004 \pm 0.175
100	30	0%	10.119 \pm 3.055	14.915 \pm 2.189	6.932 \pm 1.819	21.788 \pm 2.301	13.555 \pm 2.343	9.270 \pm 2.796	-0.002 \pm 0.211	0.005 \pm 0.293	0.005 \pm 0.293	0.010 \pm 0.371
300	1	0%	10.026 \pm 1.451	15.029 \pm 0.976	6.897 \pm 0.790	21.795 \pm 1.023	13.733 \pm 0.856	9.953 \pm 0.777	-0.003 \pm 0.078	-0.003 \pm 0.094	-0.003 \pm 0.094	-0.012 \pm 0.107
300	3	0%	10.038 \pm 1.455	15.035 \pm 0.983	6.898 \pm 0.800	21.799 \pm 1.023	13.742 \pm 0.859	9.955 \pm 0.788	-0.004 \pm 0.078	-0.003 \pm 0.095	-0.003 \pm 0.095	-0.010 \pm 0.107
300	10	0%	10.085 \pm 1.478	15.068 \pm 1.029	6.908 \pm 0.853	21.825 \pm 1.056	13.794 \pm 0.910	9.976 \pm 0.870	-0.005 \pm 0.078	-0.001 \pm 0.101	-0.001 \pm 0.101	-0.004 \pm 0.113
300	30	0%	10.190 \pm 1.617	15.149 \pm 1.272	6.903 \pm 1.100	21.854 \pm 1.413	13.905 \pm 1.279	9.924 \pm 1.455	-0.004 \pm 0.099	0.009 \pm 0.151	0.009 \pm 0.151	0.011 \pm 0.169
1000	1	0%	9.916 \pm 0.720	14.906 \pm 0.560	6.903 \pm 0.440	21.808 \pm 0.607	13.806 \pm 0.484	9.945 \pm 0.377	-0.004 \pm 0.038	-0.003 \pm 0.049	-0.003 \pm 0.049	-0.005 \pm 0.050
1000	3	0%	9.917 \pm 0.720	14.909 \pm 0.568	6.898 \pm 0.448	21.812 \pm 0.614	13.809 \pm 0.482	9.950 \pm 0.382	-0.004 \pm 0.038	-0.003 \pm 0.049	-0.003 \pm 0.049	-0.006 \pm 0.050
1000	10	0%	9.923 \pm 0.730	14.931 \pm 0.601	6.889 \pm 0.468	21.843 \pm 0.646	13.841 \pm 0.494	9.980 \pm 0.417	-0.003 \pm 0.039	-0.002 \pm 0.051	-0.002 \pm 0.051	-0.005 \pm 0.054
1000	30	0%	9.929 \pm 0.830	14.981 \pm 0.743	6.854 \pm 0.612	21.913 \pm 0.813	13.872 \pm 0.670	9.969 \pm 0.715	0.000 \pm 0.052	-0.001 \pm 0.075	-0.001 \pm 0.075	-0.011 \pm 0.091

4 A COMPARISON OF PM AND ML RESULTS

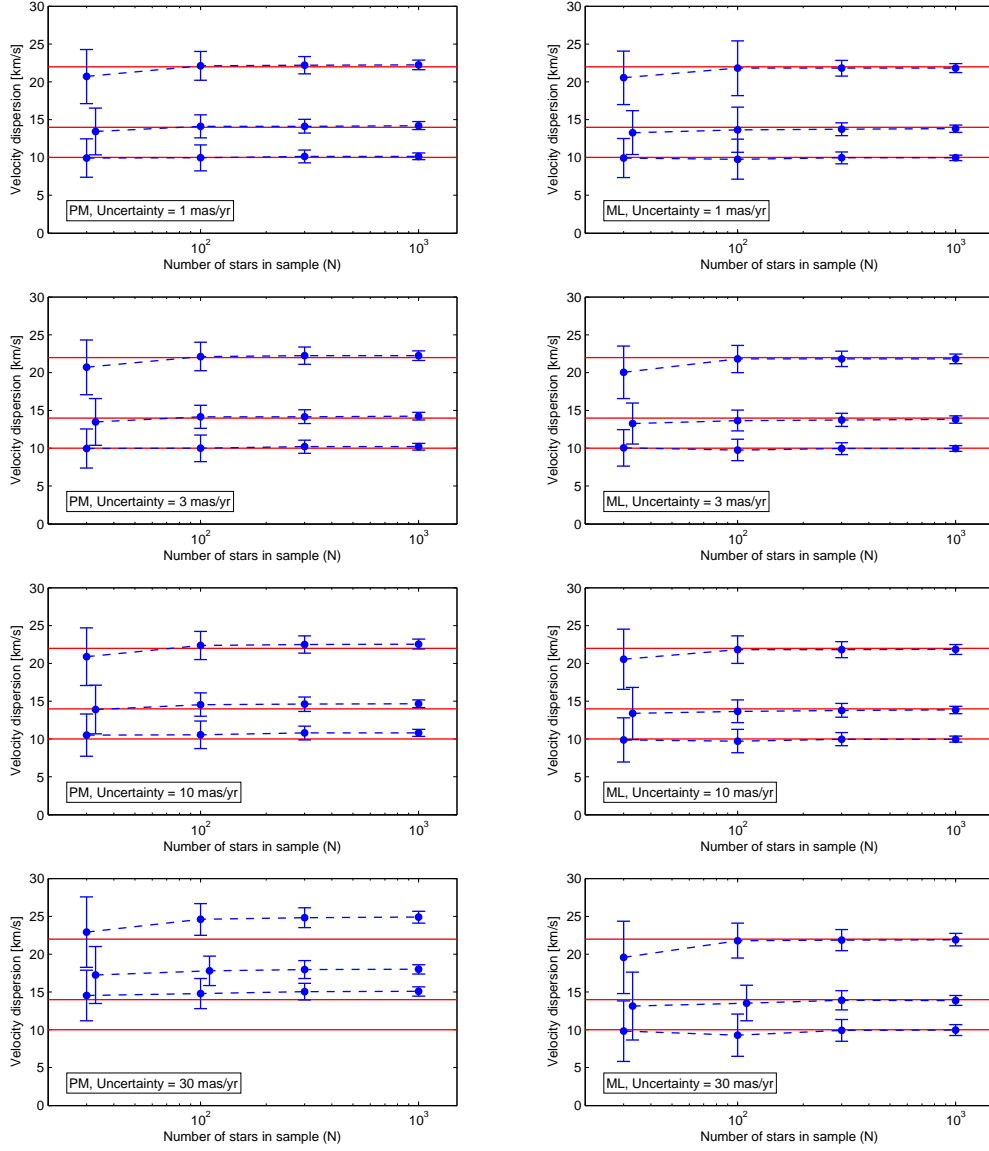


Figure 15: Plots illustrating velocity dispersion versus the number of stars, for different values of σ_μ . Diagrams in the left column are for the projection method (PM), the right column is for the Maximum Likelihood method (ML).

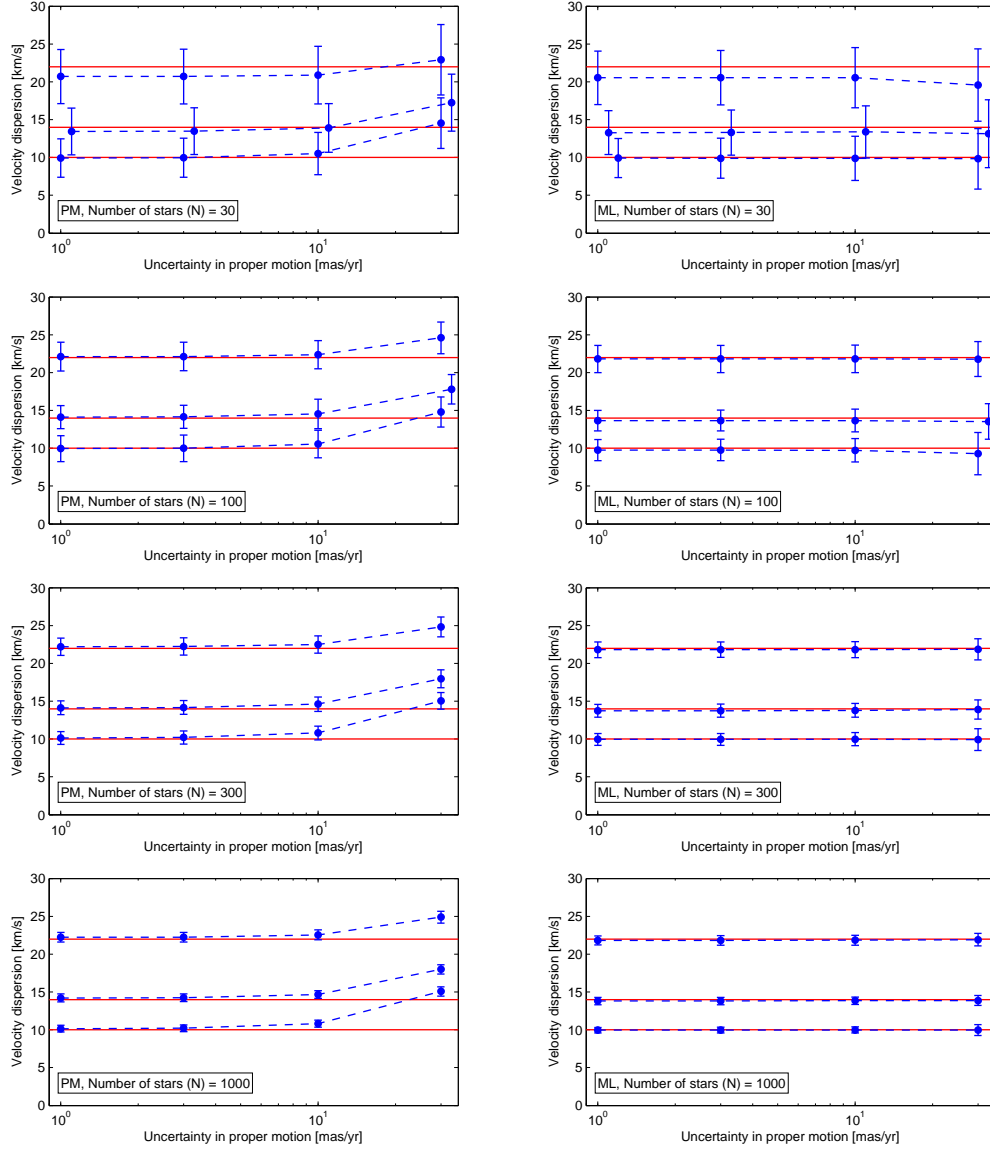


Figure 16: Plots illustrating velocity dispersion versus uncertainty in proper motion, for different number of stars. Diagrams in the left column are for the projection method (PM), the right column is for the Maximum Likelihood method (ML).

5 Application to real data

In the previous sections we tested the PM and ML methods on simulated data. This is very useful because we know what the true values are, so we can investigate bias, and we can simulate many samples to get reliable statistics. In this section we apply the methods to real stellar samples.

The Geneva–Copenhagen survey of nearby F and G dwarfs (Nordström et al. 2004) is the largest and most complete study of a magnitude-limited, kinematically unbiased sample of stars in the solar neighbourhood. It contains the the space velocities, Strömgren photometry, metallicities, rotation velocities and ages for more than 16 000 stars. The catalogue gives complete (u, v, w) space velocities based on Hipparcos data and radial velocities. We have used data from this survey to test the PM and ML methods.

Because the Geneva–Copenhagen survey gives complete space velocities, it is possible to estimate the mean velocity and velocity dispersion directly from Eqs. (2.3)–(2.4). We use these as the ‘true’ values for comparison with the PM and ML results.

Some stars could not be used because they did not have all the data necessary for this study. We also eliminated stars with a large relative parallax error ($p/\sigma_p < 10$) or a large peculiar tangential velocity ($|\Delta\tau| > 200 \text{ km s}^{-1}$). The remaining stars were divided into 10 bins of approximately equal size, sorted by their Strömgren color index $b - y$. We then estimated $\bar{\mathbf{v}}$ and \mathbf{D} for the 10 samples, using three different methods:

GC: using the full space velocities (u, v, w) as given in the survey catalogue,

PM: using the PM method based on the tangential velocities only,

ML: using the ML method based on the tangential velocities only.

The results are shown in Table 5 and Fig. 17.

As we can see, the PM and ML methods give almost the same results for all samples. Also, the results of both methods agree very well with the ‘true’ values (GC). The bottom line in Table 5 shows in how many of the 10 samples the PM or ML method gave the better result, i.e., a value closer to the GC value. Counting all 9 parameters and 10 samples, the PM was better in 39 out of 90 cases (43%), and ML was better in 51 cases (57%). Thus ML is perhaps marginally better than PM, but the difference is hardly significant.

6 Conclusions

We have tried a more rigorous Maximum Likelihood (ML) approach to calculate mean velocity and velocity dispersion for a group of nearby stars. In contrast to the projection method (PM) by Dehnen & Binney (1998) the new method takes into account uncertainties in the data, and therefore gives more accurate results in some cases where the observational errors are important.

When applied to very small samples ($n = 30$ or smaller) the PM sometimes gives unphysical results, which is avoided with the ML method. On the other hand, ML may require regularization for such small samples, which tends to give an almost isotropic dispersion matrix.

Another problem with ML is that it is difficult to apply this method. It is very complicated analytically, more complicated to program and also takes much more computer time to get the solution compared to PM.

Table 5: Results of the PM and ML applied to samples from the Geneva–Copenhagen survey (Nordström et al. 2004). The stars are divided into color bins according to the limits in $b - y$ in the first column. n is the number of stars in the bin. The third column shows the estimation method used (GC means using the full space velocities).

$b - y$	n	Method	\bar{u}	\bar{v}	\bar{w}	σ_u	σ_v	σ_w	ρ_{uv}	ρ_{uw}	ρ_{vw}
0.205 – 0.272	683	GC	-11.858	-11.939	-6.751	23.718	14.064	10.523	0.260	-0.033	-0.035
		PM	-11.789	-12.360	-6.308	22.522	13.487	10.407	0.389	0.017	-0.097
		ML	-11.580	-12.012	-6.496	22.654	13.236	9.928	0.361	-0.010	-0.102
0.272 – 0.296	719	GC	-12.120	-12.890	-7.132	25.371	16.334	12.275	0.247	0.034	-0.012
		PM	-12.206	-12.841	-7.031	24.380	17.255	11.508	0.339	0.103	0.065
		ML	-12.215	-12.647	-7.036	24.472	16.226	11.187	0.305	0.100	-0.055
0.296 – 0.313	704	GC	-9.241	-13.124	-8.304	27.728	16.831	13.878	0.271	-0.027	-0.019
		PM	-8.988	-13.525	-7.018	27.934	17.066	13.964	0.253	-0.094	-0.061
		ML	-9.492	-13.194	-7.011	28.418	16.575	12.532	0.227	-0.083	0.000
0.313 – 0.331	702	GC	-9.107	-15.494	-7.379	30.623	17.827	13.999	0.104	-0.004	0.045
		PM	-9.463	-16.506	-6.664	29.986	18.532	14.284	0.111	0.024	-0.020
		ML	-9.523	-16.249	-7.227	30.039	17.928	13.789	0.096	0.039	-0.008
0.331 – 0.348	768	GC	-10.135	-15.746	-8.559	32.489	19.300	17.090	0.092	-0.151	0.009
		PM	-10.648	-16.478	-9.569	30.600	17.754	15.595	0.071	-0.060	0.224
		ML	-10.396	-15.981	-9.381	30.101	16.841	15.117	0.064	-0.061	0.145
0.348 – 0.370	777	GC	-7.414	-20.488	-7.073	36.753	27.695	18.590	0.142	-0.068	-0.032
		PM	-7.979	-19.887	-6.765	38.220	26.288	15.787	0.023	-0.125	0.242
		ML	-7.014	-19.370	-6.856	32.981	24.694	15.770	-0.031	0.020	0.216
0.370 – 0.391	820	GC	-9.677	-23.517	-7.246	40.572	28.745	21.096	0.140	-0.018	-0.082
		PM	-8.312	-24.032	-7.579	39.549	31.045	24.188	-0.052	-0.107	-0.087
		ML	-8.151	-23.242	-7.626	38.827	25.979	22.304	-0.006	-0.064	-0.041
0.391 – 0.416	835	GC	-11.387	-26.934	-6.958	42.065	27.055	22.001	0.144	0.002	-0.075
		PM	-12.929	-27.096	-7.493	40.925	24.298	22.842	0.251	0.050	-0.093
		ML	-12.449	-26.904	-6.991	40.013	23.276	23.160	0.213	0.040	-0.070
0.416 – 0.454	865	GC	-13.105	-26.997	-7.188	37.805	26.820	20.640	0.156	-0.079	0.041
		PM	-13.067	-27.322	-7.065	39.524	27.460	19.579	0.085	-0.123	0.063
		ML	-13.088	-27.283	-7.024	38.634	26.581	19.987	0.087	-0.084	0.000
0.454 – 0.981	923	GC	-14.276	-25.999	-8.550	41.626	31.182	21.655	0.165	-0.011	0.086
		PM	-13.320	-27.132	-8.524	42.997	31.057	21.292	0.103	-0.073	0.068
		ML	-13.741	-26.293	-8.823	42.781	29.997	19.592	0.143	-0.104	-0.022
		PM better	4	2	4	5	7	6	4	2	5
		ML better	6	8	6	5	3	4	6	8	5

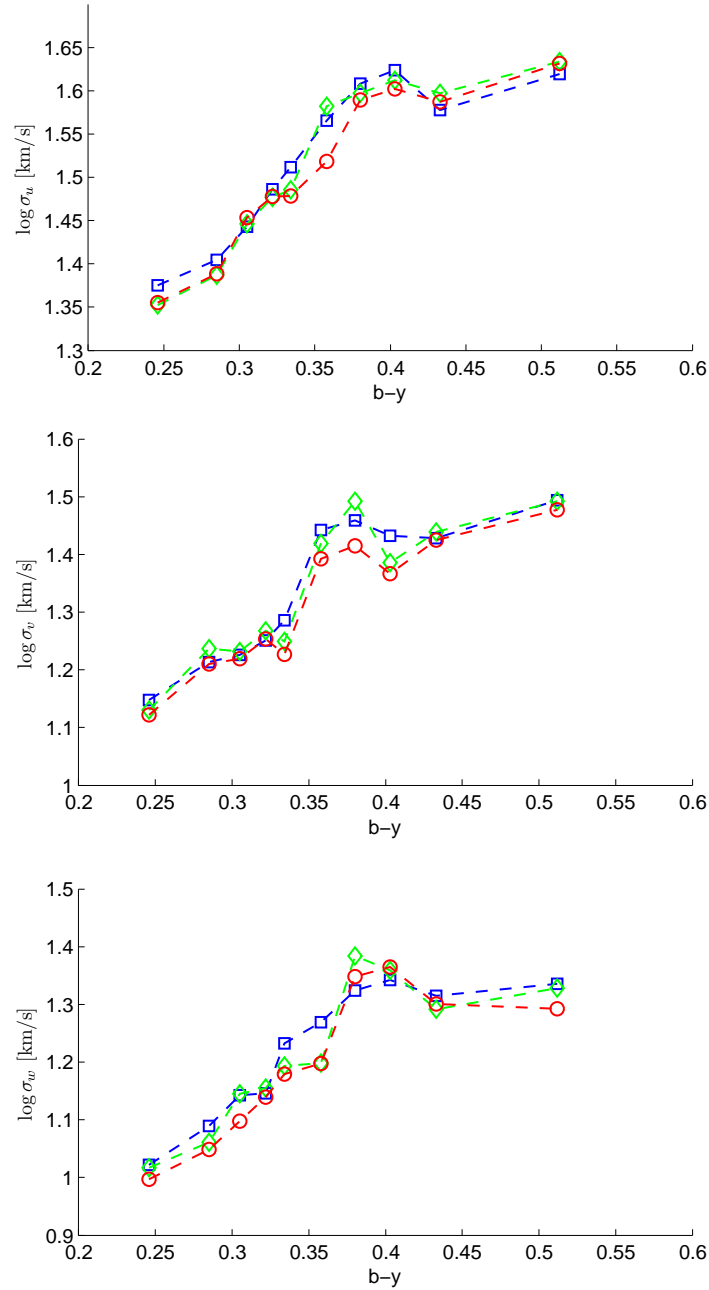


Figure 17: Velocity dispersion versus color index $b-y$ for stars in the Geneva-Copenhagen survey. Dispersions calculated directly from GC data are shown as blue squares, PM results as green rhombi, and ML results as red circles.

References

- Antoja, T., Valenzuela, O., Pichardo, B., et al. 2009, *ApJ*, 700, L78
- Aumer, M. & Binney, J. J. 2009, *MNRAS*, 397, 1286
- Bensby, T., Feltzing, S., & Lundström, I. 2003, *A&A*, 410, 527
- Bienaymé, O. 1999, *A&A*, 341, 86
- Binney, J. & Tremaine, S. 2008, *Galactic Dynamics: Second Edition*, ed. Binney, J. & Tremaine, S. (Princeton University Press)
- Branham, Jr., R. L. 2004, *A&A*, 421, 977
- Brook, C. B., Kawata, D., Gibson, B. K., & Freeman, K. C. 2004, *The Astrophysical Journal*, 612, 894
- Brosche, P., Schwan, H., & Schwarz, O. 2001, *Astronomische Nachrichten*, 322, 15
- Creze, M., Robin, A. C., & Bienayme, O. 1989, *A&A*, 211, 1
- De Simone, R., Wu, X., & Tremaine, S. 2004, *MNRAS*, 350, 627
- Dehnen, W. 2000, *AJ*, 119, 800
- Dehnen, W. & Binney, J. J. 1998, *MNRAS*, 298, 387
- Dempster, A. P., Laird, N. M., & Rubin, D. B. 1977, *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, pp. 1
- Fux, R. 2001, *A&A*, 373, 511
- Hänninen, J. & Flynn, C. 2002, *MNRAS*, 337, 731
- Hogg, D. W., Blanton, M. R., Roweis, S. T., & Johnston, K. V. 2005, *ApJ*, 629, 268
- Holmberg, J. & Flynn, C. 2000, *MNRAS*, 313, 209
- Lindgren, L. 2010, *Dynamical astronomy: Lecture notes for ASTM13 (Lund Observatory)*
- Mignard, F. 2000, *A&A*, 354, 522
- Montes, D., López-Santiago, J., Gálvez, M. C., et al. 2001, *MNRAS*, 328, 45
- Navarro, J. F., Helmi, A., & Freeman, K. C. 2004, *ApJ*, 601, L43
- Nordström, B., Mayor, M., Andersen, J., et al. 2004, *A&A*, 418, 989
- Press, W., Teukolsky, S., Vetterling, W., & Flannery, B. 2007, *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. (Cambridge University Press)
- Quillen, A. C. & Garnett, D. R. 2001, in *Astronomical Society of the Pacific Conference Series*, Vol. 230, *Galaxy Disks and Disk Galaxies*, ed. J. G. Funes & E. M. Corsini, 87–88
- Sellwood, J. A. 2011, *MNRAS*, 410, 1637
- Strömgren, B. 1987, in *NATO ASIC Proc. 207: The Galaxy*, ed. G. Gilmore & B. Carswell, 229–246
- van Leeuwen, F. 2007, *A&A*, 474, 653
- Wickramasinghe, J. T. & Napier, W. M. 2008, *MNRAS*, 387, 153

Appendix

A MATLAB codes

This Appendix lists the MATLAB codes used in this thesis. Figures 18 and 19 show how they are related.

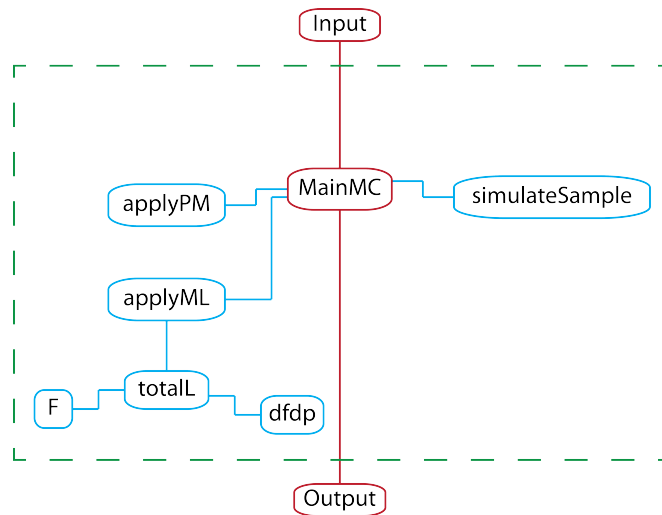


Figure 18: Diagram illustrating the relation between codes to calculate the parameters using the simulated data.

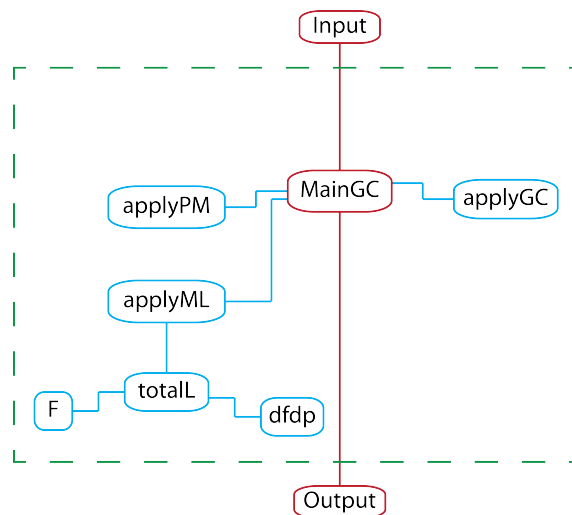


Figure 19: Diagram illustrating the relation between codes to calculate parameters using real data (Geneva-Copenhagen survey).

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% mainMC - main program to make Monte Carlo simulations
% for testing PM and ML methods.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all
clc

global alpha
global loglik neglmin it

neglmin = 1e300;
it = 0;

% max distance [pc]
rmax = 100;

% true parameters [km/s] and correlation coefficients
vTrue = [ 10; 15; 7 ];
sTrue = [ 22; 14; 10 ];
rTrue(1) = 0;
rTrue(2) = 0;
rTrue(3) = 0;

% observational errors [mas/yr]
sigmaPar = 1;
sigmaMu = 1;

% number of stars per random sample
Nstars = 1000;

% number of Monte Carlo experiments
Nexp = 1;

% set seedOffset to Nexp to generate next set of Nexp experiments
seedOffset = 0;

% regularization parameter - normally, use alpha = 0
% if matrix becomes singular, try alpha = 1 (or possibly a higher value)
alpha = 0;

sigmas = [ sigmaPar, sigmaMu, sigmaMu ];
rhos = [ 0, 0, 0 ];

for Iexp = 1:Nexp
```

```

seed = seedOffset + Iexp;

% generate sample

fprintf('\nNstars, Nexp, Iexp = %d %d %d\n', Nstars, Nexp, Iexp);
[ sample ] = simulateSample(Nstars, rmax, vTrue, sTrue, sigmas, seed);
dlmwrite('sample.txt', sample, 'precision', 10, 'newline', 'pc');
sample = dlmread('sample.txt');
fprintf(...
'true values: %7.3f %7.3f %7.3f %7.3f %7.3f %7.3f %6.3f %6.3f %6.3f\n',...
vTrue(1:3), sTrue(1:3), rhos);

%%
% apply projection method
[ Vpm, Dpm ] = applyPM(sample);
dpm = zeros(6,1);
dpm(1) = sqrt(Dpm(1,1));
dpm(2) = sqrt(Dpm(2,2));
dpm(3) = sqrt(Dpm(3,3));
dpm(4) = Dpm(1,2)/(dpm(1)*dpm(2));
dpm(5) = Dpm(1,3)/(dpm(1)*dpm(3));
dpm(6) = Dpm(2,3)/(dpm(2)*dpm(3));
fprintf(...
'PM: %7.3f %7.3f %7.3f %7.3f %7.3f %7.3f %6.3f %6.3f %6.3f\n',...
Vpm(1:3), dpm(1:6));

%%
% apply Maximum Likelihood method, using velocities from PM as starting
% values
alpha = 0;
exitflag = 0;
while (exitflag ~= 1)
    V0 = Vpm;
    D0 = zeros(3,3);
    for i = 1 : 3
        D0(i,i) = max(Dpm(i,i),10);
    end
    [ Vml, Dml, exitflag ] = applyML(sample, V0, D0);
    if (exitflag ~= 1)
        alpha = alpha + 0.5;
        fprintf('trying alpha = %f\n',alpha);
    end
end
end
dml = zeros(6,1);
dml(1) = sqrt(Dml(1,1));
dml(2) = sqrt(Dml(2,2));
dml(3) = sqrt(Dml(3,3));
dml(4) = Dml(1,2)/(dml(1)*dml(2));

```

```
dml(5) = Dml(1,3)/(dml(1)*dml(3));
dml(6) = Dml(2,3)/(dml(2)*dml(3));
fprintf(...
'ML: %7.3f %7.3f %7.3f %7.3f %7.3f %7.3f %6.3f %6.3f %6.3f %6.2f\n',...
    Vml(1:3), dml(1:6), alpha);

% write line of data to results table
table = [ Nstars, rmax, vTrue(1:3)', sTrue(1:3)', sigmaPar, sigmaMu, ...
    seed, Vpm(1:3)', dpm(1:6)', Vml(1:3)', dml(1:6)' ];
dlmwrite('results.txt', table, 'precision', 10, 'newline', 'pc',...
    '-append');
%%
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% simulateSample - function called by mainMC that simulates a sample of n
% stars with given parameters:
% n = number of stars to simulate
% rmax = max distance from sun [pc]
% vTrue = true mean velocity [km/s]
% sTrue = true velocity dispersions in x, y, z [km/s]
% sigmas(1:3) = observational errors in (par, mul, mub) [mas, mas/yr]
% seed = seed for random number generator
% Returns an array sample() with one line per star and 9 columns,
% similar content as the first 9 columns generated by hipgal
% (hip, lDeg, bDeg, p, mul, mub, sPar, sMul, sMub).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ sample ] = simulateSample(n, rmax, vTrue, sTrue, sigmas, seed)

% Set constants
K = 4.7405;
sigmaPar = sigmas(1);
sigmaMul = sigmas(2);
sigmaMub = sigmas(3);

% Assign array for output
sample = zeros(n,9);

% initialize random generator
RandStream.setDefaultStream(RandStream('mt19937ar', 'seed', seed));

% simulate n stars
for i = 1 : n

    % generate (x,y,z) random within radius of rmax
    r = 2*rmax;
    while (r > rmax)
        x = rmax * (2*rand() - 1);
        y = rmax * (2*rand() - 1);
        z = rmax * (2*rand() - 1);
        r = sqrt(x^2 + y^2 + z^2);
    end

    % parallax
    p = 1000/r;

    % galactic longitude and latitude
    l = atan2(y, x);
    b = atan2(z, sqrt(x^2+y^2));

```



```
% true space velocity
V = [ normrnd(vTrue(1),sTrue(1)) ; normrnd(vTrue(2),sTrue(2)) ; ...
      normrnd(vTrue(3),sTrue(3)) ];

% unit vectors towards star and along l and b
U = [ cos(b)*cos(l) ; cos(b)*sin(l) ; sin(b) ];
L = [ -sin(l) ; cos(l) ; 0 ];
B = [ -sin(b)*cos(l) ; -sin(b)*sin(l) ; cos(b) ];

% true proper motions
mulTrue = L' * V * p / K;
mubTrue = B' * V * p / K;

% Adding observational errors
pObs = normrnd(p, sigmaPar);
mulObs = normrnd(mulTrue, sigmaMul);
mubObs = normrnd(mubTrue, sigmaMub);

sample(i,1) = i;
sample(i,2) = l * 180 / pi;
sample(i,3) = b * 180 / pi;
sample(i,4) = pObs;
sample(i,5) = mulObs;
sample(i,6) = mubObs;
sample(i,7) = sigmaPar;
sample(i,8) = sigmaMul;
sample(i,9) = sigmaMub;

end

end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% applyPM - function called by mainMC/mainGC that applies the Projection
% Method on the simulated/real data (MC/GC).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [V, D, vPec] = applyPM(sample)

% conversion factor
K = 4.7405;

% create variables for the objects in sample
[ n ncol ] = size(sample);
l = sample(:,2) * pi / 180;
b = sample(:,3) * pi / 180;
pobs = sample(:,4);
muelobs = sample(:,5);
muebobs = sample(:,6);

%Calculating U, L and B for all stars, as well as their tangential
%velocities and their T-matrices
Tmean = zeros(3);
taumean = zeros(3,1);
for i=1:n
    U = [cos(b(i)) * cos(l(i)); cos(b(i)) * sin(l(i)); sin(b(i))];
    L = [-sin(l(i)); cos(l(i)); 0];
    B = [-sin(b(i)) * cos(l(i)); -sin(b(i)) * sin(l(i)); cos(b(i))];
    T = eye(3,3) - U * U';
    tau = K / pobs(i) * (L*muelobs(i) + B*muebobs(i));
    Tmean = Tmean + T;
    taumean = taumean + tau;
end

% calculating mean values
Tmean = Tmean / n;
taumean = taumean / n;
V = Tmean \ taumean;

% Calculates A
map = [ 1,1; 1,2; 1,3; 2,2; 2,3; 3,3 ];
Bm = zeros(3);
A = zeros(6);
vPec = zeros(n,1);
for i = 1:n
    U = [cos(b(i)) * cos(l(i)); cos(b(i)) * sin(l(i)); sin(b(i))];
    L = [-sin(l(i)); cos(l(i)); 0];
    B = [-sin(b(i)) * cos(l(i)); -sin(b(i)) * sin(l(i)); cos(b(i))];

```

```
T = eye(3,3) - U * U';
tau = K / pobs(i) * (L*muelobs(i) + B*muebobs(i));
Deltatau = tau - T * V;
Bm = Bm + Deltatau * Deltatau';

vPec(i) = sqrt(Deltatau' * Deltatau);

for k = 1:6
    A(k,1) = A(k,1) + T(map(k,1),1) * T(map(k,2),1);
    A(k,2) = A(k,2) + T(map(k,1),1) * T(map(k,2),2) + T(map(k,1),2)...
        * T(map(k,2),1);
    A(k,3) = A(k,3) + T(map(k,1),1) * T(map(k,2),3) + T(map(k,1),3)...
        * T(map(k,2),1);
    A(k,4) = A(k,4) + T(map(k,1),2) * T(map(k,2),2);
    A(k,5) = A(k,5) + T(map(k,1),2) * T(map(k,2),3) + T(map(k,1),3)...
        * T(map(k,2),2);
    A(k,6) = A(k,6) + T(map(k,1),3) * T(map(k,2),3);
end
end

A = A / n;
Bm = Bm / n;

bm = [Bm(1,1); Bm(1,2); Bm(1,3); Bm(2,2); Bm(2,3); Bm(3,3)];
d = A \ bm;

D = zeros(3,3);
for k = 1 : 6
    D(map(k,1),map(k,2)) = d(k);
end
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% applyML - function called by mainMC/mainGC that applies the Maximum
% Likelihood Method (estimates V and D, using V0, D0 as starting values)
% on the simulated/real data (MC/GC).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ V, D, exitflag ] = applyML(sample, V0, D0)

global k n l b p muel mueb sigmamuel sigmamueb sigmap rho

% conversion factor
k = 4.7405;

% create variables for the objects in sample
[ n ncol ] = size(sample);

l = sample(:,2) * pi / 180;
b = sample(:,3) * pi / 180;
p = sample(:,4);
muel = sample(:,5);
mueb = sample(:,6);
sigmap = sample(:,7);
sigmamuel = sample(:,8);
sigmamueb = sample(:,9);
rho = zeros(n,1);

Roff = 1;
X0 = [ V0(1), V0(2), V0(3), 0.5*log(D0(1,1)), 0.5*log(D0(2,2)),...
      0.5*log(D0(3,3)), Roff, Roff, Roff ];

options = optimset('MaxFunEvals', 5000, 'MaxIter', 3000, ...
                  'TolFun',1e-6, 'TolX',1e-3, 'FunValCheck','on');

try
    [X, fmin, exitflag, output] = fminsearch(@totalL, X0, options);
catch exception
    exitflag = -1;
    V = V0;
    D = D0;
    return
end

V = [ X(1); X(2); X(3) ];
R = [exp(X(4)) X(7) X(8) ; 0 exp(X(5)) X(9) ; 0 0 exp(X(6))];
D = R' * R;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% totalL - function called by applyML that sum all the
% log-likelihood.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function negloglik = totalL(X)

global k n l b p muel mueb sigmamuel sigmamueb sigmap rho
global alpha

vmean = [X(1); X(2); X(3)];
R = [exp(X(4)) X(7) X(8) ; 0 exp(X(5)) X(9) ; 0 0 exp(X(6))];
D = R' * R;

loglik = 0;

for i = 1 : n
    loglik = loglik + F(p(i),vmean,D,i) + ...
        0.5 * ( sigmap(i) * dfdp(p(i),sigmap(i),vmean,D,i) )^2;
end

% regularization, using alpha parameter (only done if alpha > 0):
%if (alpha > 0)
    s = svd(D);
    cond = s(1)/s(3);
    loglik = loglik - alpha * log(cond);
%end

if (cond < 1000)
    negloglik = -loglik;
else
    negloglik = NaN;
end

end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% F - function called by totall that calculates the log-likelihood.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function loglik = F(ptry, vmean, D, i)

global k n l b p muel mueb sigmamuel sigmamueb sigmap rho

s1 = sin(l(i));
c1 = cos(l(i));
sb = sin(b(i));
cb = cos(b(i));

M = (ptry/k) * [ -s1 , c1 , 0 ; -c1*sb , -s1*sb , cb ];

mueobs = [ muel(i) ; mueb(i) ];

deltamue = mueobs - M * vmean;

sml = sigmamuel(i);
smb = sigmamueb(i);
r = rho(i);

cmue = M * D * M' + [ sml^2 , r*sml*smb ; r*sml*smb , smb^2 ];

loglik = -0.5 * ( log(det(cmue)) + deltamue' * (cmue\deltamue) );

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% dfdp - function called by totall that calculates the derivative of
% log-likelihood calculated in function F.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function derivative = dfdp(ptry,dp,vmean,D,i)

derivative = (F(ptry + dp,vmean,D,i) - F(ptry - dp,vmean,D,i)) / (2 * dp);

```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% mainGC - main program to analyze Geneva-Copenhagen samples

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
clc

global alpha

alpha = 0;

file = 'GC/bin3.txt';
sample = dlmread(file);

par = sample(:,4);
sigma_par = sample(:,7);

% find the good stars with par > 10*sigma_par:

good = find(par > 10*sigma_par);

good_sample = sample(good,:);

sample = good_sample;

[ n col ] = size(sample);

% apply projection method:

[Vpm, Dpm, vPec] = applyPM(sample);

good = find(vPec < 200);
good_sample = sample(good,:);

[ n col ] = size(good_sample);

% apply projection method:

[Vpm, Dpm, vPec] = applyPM(good_sample);

fprintf('file = %s, n = %d\n', file, n);

dpm = zeros(6,1);
dpm(1) = sqrt(Dpm(1,1));
```

```
dpm(2) = sqrt(Dpm(2,2));
dpm(3) = sqrt(Dpm(3,3));
dpm(4) = Dpm(1,2)/(dpm(1)*dpm(2));
dpm(5) = Dpm(1,3)/(dpm(1)*dpm(3));
dpm(6) = Dpm(2,3)/(dpm(2)*dpm(3));
fprintf('PM: %7.3f %7.3f %7.3f %7.3f %7.3f %7.3f %6.3f %6.3f %6.3f\n', ...
        Vpm(1:3), dpm(1:6));

% apply Maximum Likelihood method, using PM as starting values:

V0 = Vpm;
D0 = zeros(3,3);
for i = 1 : 3
    D0(i,i) = max(Dpm(i,i),10);
end
[Vml, Dml] = applyML(good_sample, V0, D0);
dml = zeros(6,1);
dml(1) = sqrt(Dml(1,1));
dml(2) = sqrt(Dml(2,2));
dml(3) = sqrt(Dml(3,3));
dml(4) = Dml(1,2)/(dml(1)*dml(2));
dml(5) = Dml(1,3)/(dml(1)*dml(3));
dml(6) = Dml(2,3)/(dml(2)*dml(3));
fprintf('ML: %7.3f %7.3f %7.3f %7.3f %7.3f %7.3f %6.3f %6.3f %6.3f\n', ...
        Vml(1:3), dml(1:6));
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% applyGC - function called by mainGC that calculates the mean velocity and
% dispersion for each real sample bin.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [ V, sTrue ] = applyGC(sample)

l = length(sample);

SUM = sum(sample);
meanU = SUM(11)/l;
meanV = SUM(12)/l;
meanW = SUM(13)/l;

V = [meanU, meanV, meanW];

D = zeros(3,3);
for i = 1:l

A = [sample(i,11); sample(i,12); sample(i,13)] - [meanU; meanV; meanW];

D = D + A * A';
end

D = D/l;

sTrue(1) = sqrt(D(1,1));
sTrue(2) = sqrt(D(2,2));
sTrue(3) = sqrt(D(3,3));
sTrue(4) = D(1,2) / sqrt(D(1,1)*D(2,2));
sTrue(5) = D(1,3) / sqrt(D(1,1)*D(3,3));
sTrue(6) = D(2,3) / sqrt(D(2,2)*D(3,3));
```