

# AST1501 Notes

## 1. Coordinate Transformation

### a. Stream to Equatorial Coordinates

$$\begin{bmatrix} \cos(\phi_1) \cos(\phi_2) \\ \sin(\phi_1) \cos(\phi_2) \\ \sin(\phi_2) \end{bmatrix} = \begin{bmatrix} -0.4776303088 & -0.1738432154 & 0.8611897727 \\ 0.510844589 & -0.8524449229 & 0.111245042 \\ 0.7147776536 & 0.493068392 & 0.4959603976 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha) \cos(\delta) \\ \sin(\alpha) \cos(\delta) \\ \sin(\delta) \end{bmatrix} \quad (1)$$

So if we assume the left hand side matrix is A and the others are B and C, respectively, we can get C matrix by:

$$\begin{aligned} A &= BC \\ B^{-1}A &= (B^{-1}B)C \\ B^{-1}A &= C, \end{aligned} \quad (2)$$

therefore, we get:

$$x = y + z \quad (3)$$

### b. Cylindrical to Stream coordinates

Coverion from cylindrical coordinates  $(R, z, \phi)$  to cartesian coordinates  $(x, y, z)$ :

$$\begin{aligned} x &= R \cos(\phi) \\ y &= R \sin(\phi) \\ z &= z \end{aligned} \tag{4}$$

Conversion from cartesian coordinates  $(x, y, z)$  to equatorial coordinates  $(\delta, \alpha, d)$ :

$$\begin{aligned} d &= \sqrt{x^2 + y^2 + z^2} \\ \delta &= \arcsin\left(\frac{z}{d}\right) \\ \alpha &= \arctan\left(\frac{y}{x}\right) \end{aligned} \tag{5}$$

Finally, conversion from equatorial  $(\delta, \alpha, d)$  to stream coordinates  $(\phi_1, \phi_2)$  can be done using equation 1.

## 2. Likelihood Procedure

$$\begin{aligned} L(\text{model}|\text{orbit}) = P(V_c, q_\phi|\text{Data}) &= \prod_i P(\text{data}_i|\text{model}) = \\ \prod_i (P(\phi_{2,i}|\phi_{1,i}), P(D_i|\phi_{1,i}), P(Vrad_i|\phi_{1,i}), P(\mu_i|\phi_{1,i})) &\quad (6) \end{aligned}$$

$$\ln(L) = \sum_i P(\text{data}_i|\text{model}). \tag{7}$$

We can get the likelihood by computing this probability. However, in Koposov et al. 2010, they marginalized over the oarameters rather than finding the likelihood. Marginalization can be

done in the following way:

$$P(V_c, q_\phi | \text{data}(D)) = \int_7^{10} P(V_c, q_\phi | D, R_0) P(R_0) dR_0, \quad (8)$$

where  $D$  represents the data and  $P(R_0)$  is given by:

$$P(R_0) = M(R_0 | 8.4, 0.42) = \frac{1}{2\pi} e^{-\frac{(R_0 - 8.4)^2}{2 \times 0.42^2}}. \quad (9)$$

The value of our guess for  $R_0$  being 8.4 with the error of 0.42 comes from Kopolov et al. 2010. We should then do the same calculation with values of 8.2 for  $R_0$  with the error of 0.1 .

I made my likelihood function faster by eliminating the for loop existed in the previous version. It now takes arrays of data and model values and computes the likelihood for each set of parameter (such as  $\phi_2, V_{\text{rad}}$  and  $\mu$  using `numpy.tile()` function. The `numpy.tile()` generates an array that has the array you pass to it repeated  $n$  times. So in order to have the calculations correct, I made an array as shown in equation (??):

$$\begin{bmatrix} \text{data1} & \text{data2} & \cdots & \text{dataN} \\ \text{data1} & \text{data2} & \cdots & \text{dataN} \\ \text{data1} & \text{data2} & \cdots & \text{dataN} \\ \vdots & \vdots & \ddots & \vdots \\ \text{data1} & \text{data2} & \cdots & \text{dataN} \end{bmatrix}, \quad (10)$$

where the number of rows is equal to the length of model and the number of the columns is equal to the length of data.

Likewise, we can write down a similar matrix for the model values as in equation (??):

$$\begin{bmatrix} \text{model 1} & \text{model 1} & \cdots & \text{model 1} \\ \text{model 2} & \text{model 2} & \cdots & \text{model 2} \\ \text{model 3} & \text{model 3} & \cdots & \text{model 3} \\ \vdots & \vdots & \ddots & \vdots \\ \text{model N} & \text{model N} & \cdots & \text{model N} \end{bmatrix}, \quad (11)$$

where the number of rows is equal to the length of model and the number of columns is equal to the length of data. Then, in order for the matrices to have the same shape so that we will be able to do mathematical operations on them, we have to take the transpose of the model matrix. Finally, to compute the integral of each column in the final value of the tiled matrix, I used `simps` integration function including `axis=0` as an argument so that it takes the integral of the columns and returns the values of the integra of each column as an array to avoid using for loops. Then we will be able to the likelihood calculations as normal.

### 3. General Notes

- The actions of stars in the cluster are not conserved (because the self-gravity of the cluster is important), but that the actions of stream members freeze once they are stripped.
- The angle difference between stars in a stream and the progenitor increases linearly with time.
- 
- 
- 
-