Project S Final Report

Xin Chen, Yun Yeong Choi
Department of Materials Science and Engineering
University of California, Berkeley
{chenxin0210, yychoi94}@berkeley.edu

Ke Ma
Department of Civil and Environment Engineering
University of California, Berkeley
kema@berkeley.edu

Fengzhe Shi
Department of Chemical Engineering
University of California, Berkeley
schweini@berkeley.edu

1 Abstract

With the purpose of developing a prediction model of stellar bodies, we performed a learning process with several models. Start from the most basic linear regression model, and to ridge regression, LASSO, we found that the mean square error of certain features in the test set remains high (larger than103) and conclude that these models are not appropriate to predict the motion of stellar bodies on the celestial sphere in time series. Moreover, moon phase prediction by using Neural Network also shows poor validation accuracy. To overcome this problem, we choose the important Fourier features by ordering them to minimize the mean square error of the training set. By using this method, we were able to predict Sun's daily, hourly altitude successfully (0.005, 8.692 MSE)with only 20 features. [1]

2 Introduction

Historically, many ancient astronomers were trying to predict the movement of the moon, planets, and stars for the purpose of astrology or agriculture. These kinds of efforts last for thousands of years and lead to the development of physics, from Newton's law of motion to Einstein's general relativity.

In this project, we will focus to capture their efforts and inspiration by using several regression techniques. we start by collecting the virtual observation data of ancient China by using the Stellarium API which made by another group from the early project. With the data, we will use several regression methods to test how much can predict the movement of celestial bodies based on the observation data. Considering that not all planets are easily observed on the Earth, our focus is on the prediction of Sun, Moon, Mars azimuthal angle and altitude on the celestial sphere and also phase change of the Moon.

After that, we will try to mimic the human learning process by means of machine learning techniques. What the old astronomers did was looking into data long and deeply, and find out some equations and features that explain their data better. Instead of those painful and time-consuming process, what we are going to do in this project is to add possible features and fit them with data, choosing the most appropriate features which make models better using machine learning techniques. Start from the ancient one, we will develop our own model and predict the movement of stellar bodies based on these machine-learned models. Unlike previous regression methods, this method will only use limited numbers but the most important features considering that humans only can deal with few variables to predict. Also, our final goal is to find the features that made great improvement, and explain what is the physics behind these features.

[1] Source code available on "https://github.com/mkmark/CS-289A-Proj-SF-XYKF"

3 Method

3.1 About the Ancient Model

Ancient Chinese has long discovered that the planets are not moving the same way as the stars, but could not propose a precise model to predict their movements. It turns out under the ancient Chinese Astronomy model, which is basically a geocentric model with Horizontal Coordinate System, the planets move in an extremely complex way (see Appendix - Physical Model).

Under such a system, the only input we are allowed to get is the azimuth and altitude of a stellar body at some specific time. Considering the actual observing restrictions, there are also limitations on the star's altitude as a star with an altitude lower than 0 (or close to the horizon) can not be observed. Furthermore, weather conditions may also restrict observations.

To honor the famous Chinese astronomer, ZHNAG Heng, who invented the first hydraulic armillary sphere in 117 AD, we times all our 'observation' to start at Jan 1st 118 AD, in the capital city Luoyang at that time.

3.2 Data Sources

From the early project S, we collected data from the Stellarium and it is consists of 9 celestial bodies with 8 features; time, right ascension, declination, azimuthal angle, altitude, Distance, Distance from the sun, constellations. In principle, the motion of a celestial body in the solar system is determined by 13 variables; velocity of body, position of body, velocity of earth, position of the earth with respect to sun and time. With the given mass of earth, body, and sun, one can predict the future position of the celestial body quite exactly based on classical mechanics.
However, there are limitations on project S early teams that they have to use data not directly available such that the data collection, especially with large quantity, can become a potential problem. In our specific case, where data of hourly spaced data over a period of 100 years are required, no project S early team project can fulfill the requirement in a timely reasonable manner. Thus other data sources are employed, including the python package 'solar system'. Note that these data are equivalent to those extracted by project S early teams, just in a more time-efficient way.
We get the moon phase data from 'Pylunar'('https://pylunar.readthedocs.io/en/latest/usage.html')

3.3 Predict Mars and Sun position based on Linear, Ridge regression, LASSO with Fourier features

From here, our approach is slightly different from physics. We will try to predict the position of the body in the celestial sphere based on past data using different regression techniques. As a start, we will try the basic regression techniques - Linear Regression, Ridge Regression, LASSO.
There are a few principles we are going to keep during regression; since we are trying to mimic past astronomers whose object is predicted future data based on past data, we will going to collect $40 \sim 50$ year's location of a celestial body in the celestial sphere and going to predict next $5 \sim 10$ years location of them. Our data is collected from certain locations in China (Luoyang), from A.C. 118 to 168. This would be a harsh condition for training since our training point and testing point are totally separated and time series problem. Thus, we are going to use a huge number of additional features, which are the Fourier featured version of time (Considering that celestial motions are periodic function, Fourier features might be helpful). However a lot of features are not feasible works for the past astronomers, so we will try to predict the motion of a stellar body with a small number of features in part 4.3.

3.4 Predict Stellar Body Positions with Tuned Linear Regression based on Fourier Features

Theoretically, due to the nature of planet orbits, given a long enough period, any movement, despite the reference system, can be found repetitive, which makes regression on Fourier Features highly feasible. The problem with such an approach is that the period is a difficult parameter to be estimated, which makes simultaneously minimizing target function with high numbers of nonlinear features almost impossible.

An alternative method is proposed by tuning the Fourier Feature parameters according to training MSE one by one, and nesting the process so that a higher-order parameter can be derived from the

error of the previous prediction, such that a high order fitting is possible. The detailed method can be found in the jupyter notebook. Such a method allows us to precisely predict some of the stellar body movements.

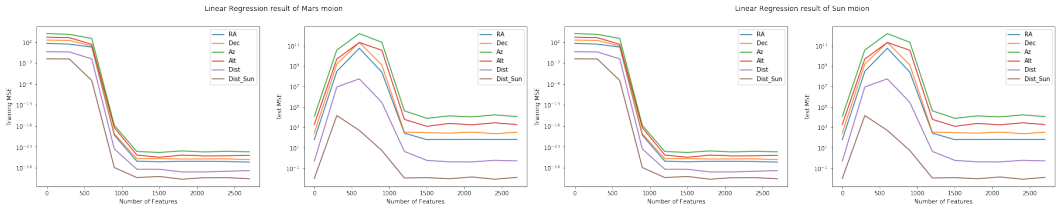
3.5 Predict Moon position and phase based on Logistic regression, Neural Network

The method of predicting the moon position is the same as the previous part.However, moon phase is different from the position or movement of the celestial body. Basically, moon phase prediction is a classification problem, although it can be treated as regression to by using certain regression techniques such as Lasso. Therefore, we will train the moon phase data in two ways in this project. The first one is Lasso regression and the second one is Neural Network for classification.

4 Results

4.1 Prediction on positions of Mars and Sun

4.1.1 Linear Regression on Mars and Sun with Fourier Features



(a) Linear Regression results of Mars position features (b) Linear Regression results of Sun position features

Figure 1: Train and Test MSE of Mars and Sun postion

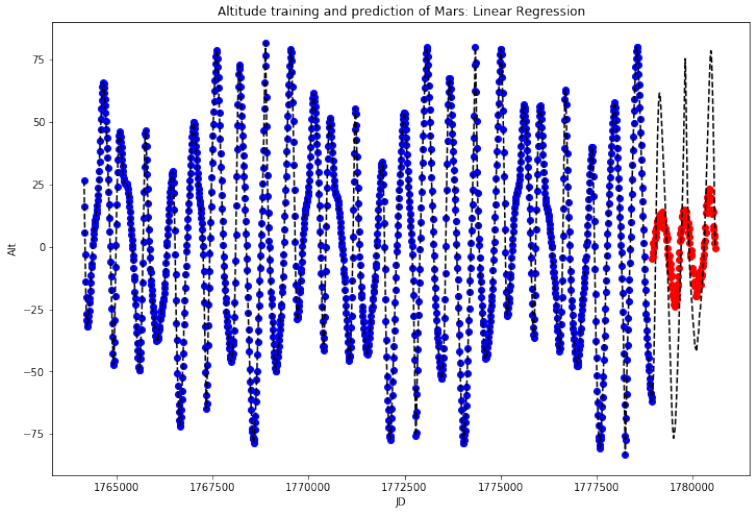
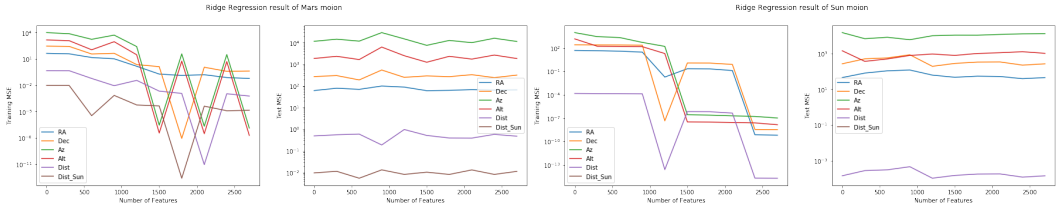


Figure 2: Trained and tested altitude of Mars by Linear regression

Figure 1 a and b shows training and test Mean Square Error (MSE) of Mars and Sun based on the Linear regression when using Fourier featurized time. As one can see, training MSE is very low since we used enough number of features to match with training set. However, test MSE is not much improved by adding more features. This happens because position of the Mars and Sun is not perfectly periodic, so if we trying to fit training set perfectly, in the test set it deviates more from the true data. It can be understood as one example of bias-variance trade off. One can check what happened during training by looking at figure 2. Training data is perfectly fitted and periodicity of test data is well

predicted, but absolute value of predicted data decrease since there are too many coefficients offset each other at the outside of training set. Therefore, we checked that linear regression is not a good way to predict future position of celestial body and need to compress or constrain coefficient vector using Ridge or LASSO.

4.1.2 Ridge Regression on Mars and Sun with Fourier Features



(a) Linear Regression results of Mars position features (b) Linear Regression results of Sun position features

Figure 3: Train and Test MSE of Mars and Sun postion

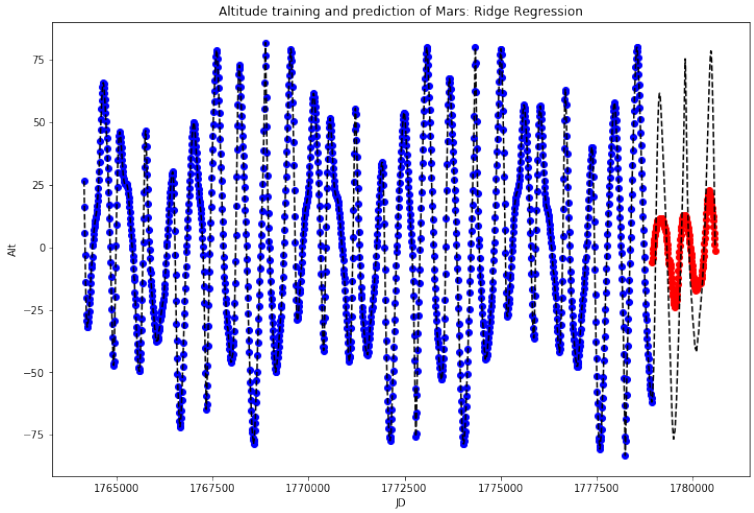
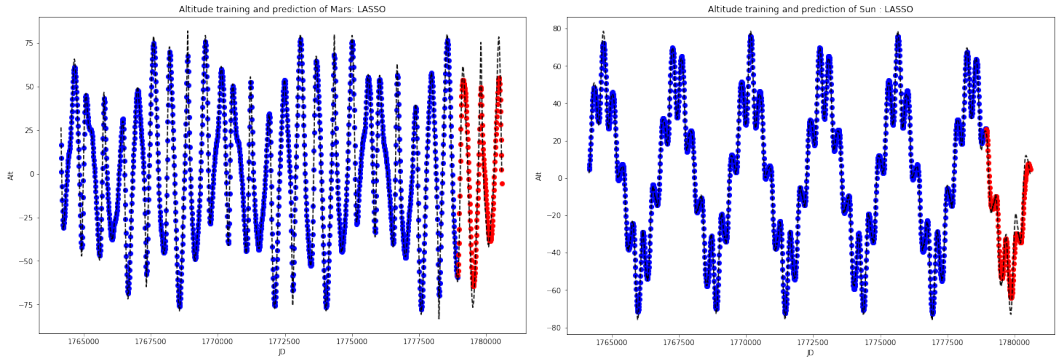


Figure 4: Trained and tested Altitude of Mars by Ridge regression

Next, we ran Ridge regression instead of linear regression. Since ridge rigression has effect of compressing size of coefficients, we expect slightly better results than linear regression. As one can see from the results, training and test MSE of ridge is not much improved by adding features. It's value is similar with starting point. This is somewhat natural behavior since we only use Fourier-like features and constraints total size of coefficients, there are only limiting choices of features to minimize MSE. The prediction results become smoother than linear regression version but still bad to use.

4.1.3 LASSO on Mars and Sun with Fourier features

Finally we test LASSO. We expected this would be better than ridge regression since we use lots of number of features, and LASSO is good for removing irrelevant (or having small coefficient) features by setting appropriate hyper-parameter alpha. We used grid search function from scikit-learn package to determine alpha there. As one can see from the results (Figure 5), not only the training set is properly trained, but also test set shows better behavior than linear regression or ridge regression. Therefore we can conclude that LASSO is proper way to predict the motion of stellar body with Fourier features. Based on the idea that removing small coefficient features might help for the better prediction, we will try to use most relevant features in the next section.



(a) LASSO results of Mars position features (b) LASSO results of Sun position features

Figure 5: Trained and Predicted altitude of Mars and Sun by LASSO

4.2 Predict Stellar Body Positions with Tuned Linear Regression based on Fourier Features

By fitting the error of previous prediction with new tuned Fourier features such method can be used to predict extremely complex function over a long period.

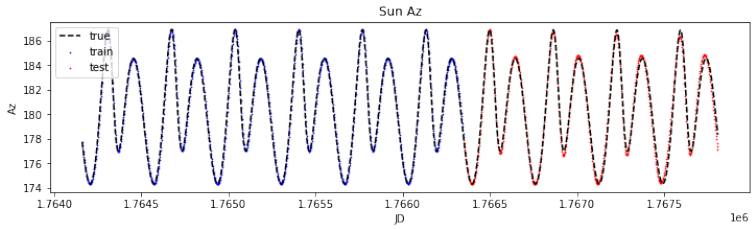


Figure 6: Sun Azimuth at Noon Every 48h For 10 Years with Test MSE = 0.416

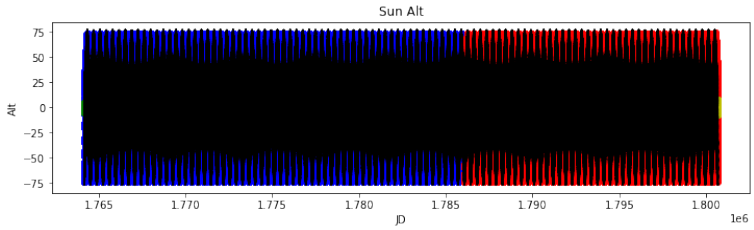


Figure 7: Sun Altitude Every Hour For 100 Years with Test MSE = 8.69

Note that the prediction tends to deviate from the true value after a long period of time. Taking the advantage of Fourier features of this method, such deviation tends to stay in range and preserve the right trends even after decades of prediction.

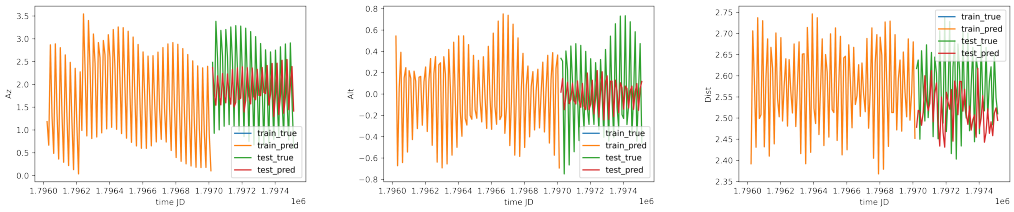
4.3 Prediction on Positions of the Moon and Moon Phases

Based on the previous results, we treated the data of the moon with the same featurization method. We tried linear regression(LR), ridge regression(RR) and Lasso to predict the positions of the moon. We also predicted the moon phase using Lasso and neural network(NN).

4.3.1 Prediction on Positions of the Moon

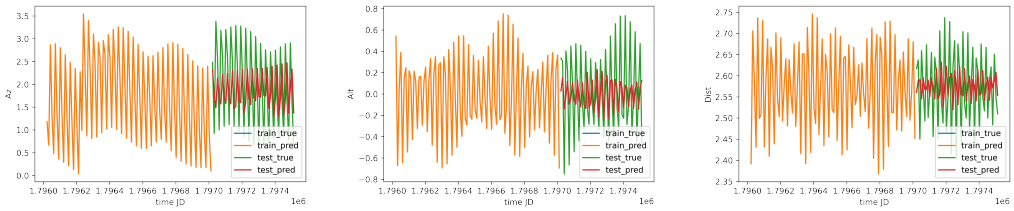
The following figures shows the training and prediction results of different machine learning methods. We use the order of '3000' when we featurized the input time based on the previous research. For

the ridge regression and Lasso, we conducted hypertuning to select the best hyperparameters using 'GridSearchCV' method from 'Sklearn'.



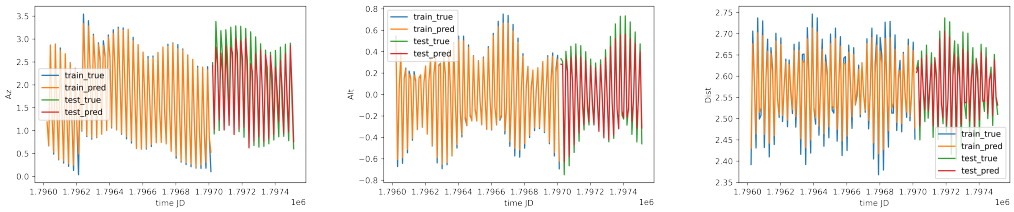
(a) Azimuth of Moon (b) Altitude of Moon (c) Distance of Moon

Figure 8: Prediction on Moon with Linear Regression. The test mse is 0.90, 0.13, 0.01 for Az, Alt, Dist respectively



(a) Azimuth of Moon (b) Altitude of Moon (c) Distance of Moon

Figure 9: Prediction on Moon with Ridge Regression. The test mse is 0.90, 0.13, 0.01 for Az, Alt, Dist respectively



(a) Azimuth of Moon (b) Altitude of Moon (c) Distance of Moon

Figure 10: Prediction on Moon with Lasso. The test mse is 0.21, 0.02, 0.002 for Az, Alt, Dist respectively

It is obvious that Lasso is good at making prediction on the test set, just as we expected.

4.3.2 Prediction on the Moon phase

The illumination fraction of the moon is strongly correlated with the moon phase. The following fig11 illustrates well on how the moon phases are classified from the illumination fraction.

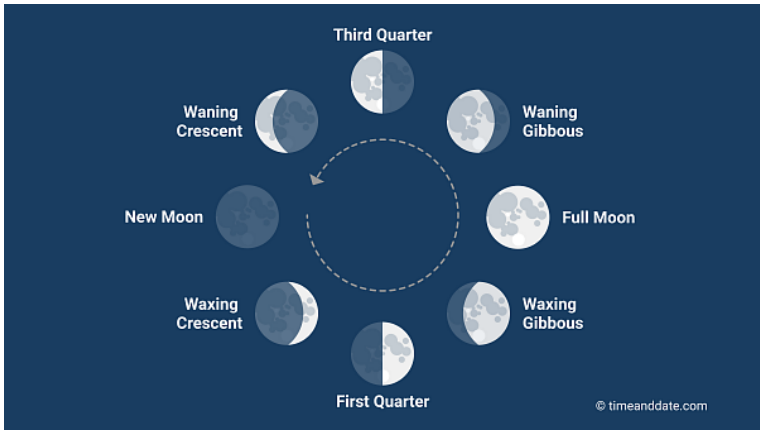
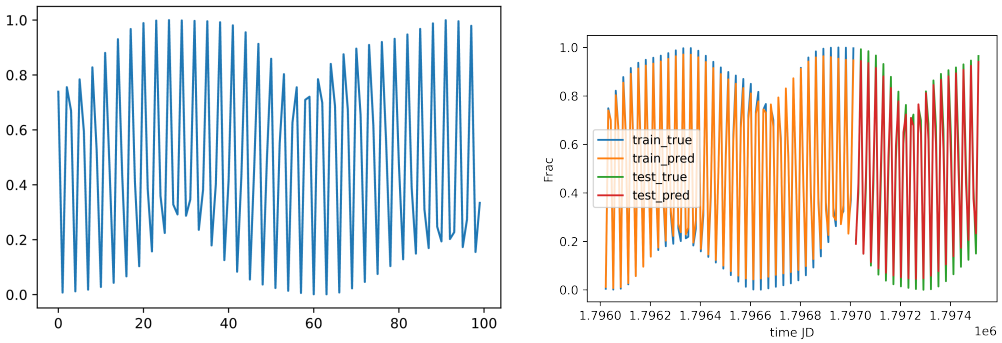


Figure 11: Illumination fraction for different moon phases[3]

The illumination fraction is a periodic function of time, as shown in the following fig12a.



(a) Illumination fraction period

(b) Prediction on illumination fraction with Lasso.

Figure 12: Illumination fraction of moon. The test mse for subfig(b) is 0.012

This periodicity gives us the intuition that the illumination fraction can be fitted using the same method of predicting the positions.

We use featurize the time with order of 3000 as what we did in the previous part and predict the illumination fraction with Lasso. Fig12b shows the results.

After we predict the illumination fraction, we can compute the phase name of the moon as shown in fig13 from our Jupyter Notebook.

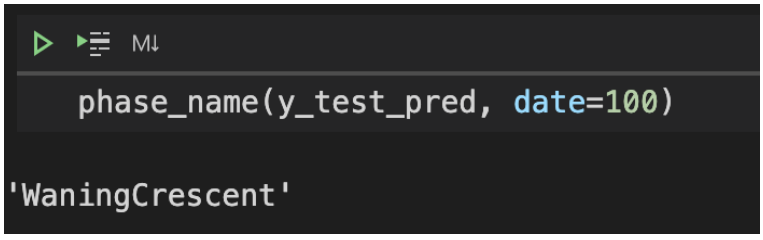


Figure 13: Prediction on the moon phase name

Considering that there are 8 different moon phases, we can treat this task as a classification problem. Therefore, we also built a neural network to classify the moon phase. The result is shown in fig14.

However, the validation accuracy is pretty low which indicates that our neural network model is not suitable for this task. Further work needs to be done to improve the neural network.

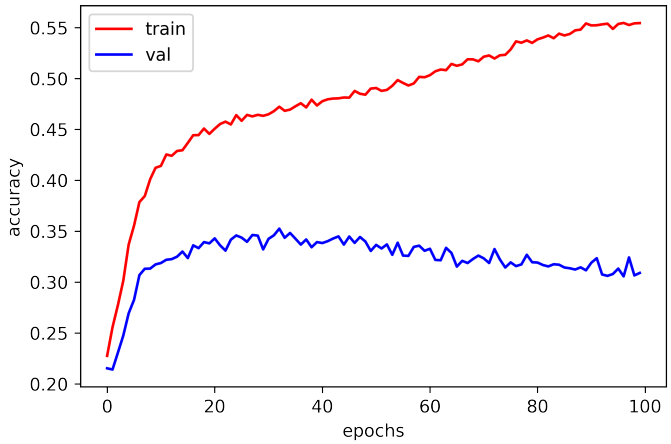


Figure 14: Training and testing process of prediction on the moon phase with NN

In summary, based on above results, we suggest to use Lasso for beginners. Neural network could be improved better if the users are more experienced.

5 Conclusion

In conclusion, we found that predicting time-series data is a difficult problem to solve with regression models. Even though we used Fourier featured time, the data was not perfectly periodic, thus a large number of features show adversary effects on the test set. LASSO works best among regression models but also shows a high mean square error over 100 for certain features. The next trial, repeatedly finds the best Fourier features minimizing error on the training set significantly reduces the previous problems. Using this method, we achieved relevantly good MSE with only 20 Fourier features. Considering that we did not consider elliptical orbit, it would be one of the most effective ways to predict stellar body motion using regression models.

Predicting periodic functions turns out to require a lot of data, or at least data covering several times more than the longest period to learn from. Prediction without enough data tends to incur unexpected divergence. However, since there are almost always potential long period features hiding somewhere, long-period prediction tends to deviate from the true value, though the trend may be preserved with the right training method.

References

[1] Fang, X. (1974). Jin shu. Beijing: Zhonghua shu ju.

[2] Bruce G. Thompson. (2005) Using retrograde motion to understand and determine orbital parameters. *American Journal of Physics 73, 1023*

[3] https://www.timeanddate.com/astronomy/moon/phases.html

Appendix - Physical Model

In this section we will introduce a simplified physical model of planet motions so that we can understand the problem more properly.

Solar-centered ecliptic coordinate system

Solar-centered ecliptic coordinate system is centered at the sun, using spring equinox as $x+$ or polar axis, ecliptic as xy plane. Earth loacation

as (polar)

$$\mathbf{x}_e = (r_e, \theta_e, 0), \theta_e = \frac{2\pi}{T_{ey}} t + \theta_{e0}$$

or (dirichlet)

$$\mathbf{x}_e = \begin{bmatrix} r_e \cos(\frac{2\pi}{T_{ey}} t + \theta_{e0}) \\ r_e \sin(\frac{2\pi}{T_{ey}} t + \theta_{e0}) \\ 0 \end{bmatrix}$$

other planets can have a similar definition.

Earth-centered ecliptic coordinate system

Solar-centered ecliptic coordinate system is centered at the earth, using spring equinox as $x+$ or polar axis, ecliptic as xy plane.

Equatorial coordinate system

Equatorial coordinate system is centered at the earth, using spring equinox as $x+$ or polar axis, equator as xy plane. Planets location in such system is defined as right ascension α and declination δ,

as (polar)

$$\mathbf{X}_p = (R_p, \frac{\pi}{2} - \delta, \alpha)$$

as we normally do with spherical coordinate system (r, θ, ϕ)

or (dirichlet)

$$\mathbf{X}_p = \begin{bmatrix} R_p \cos\delta \cos\alpha \\ R_p \cos\delta \sin\alpha \\ R_p \sin\delta \end{bmatrix}$$

Horizontal coordinate system

Horizontal coordinate system is centered at the observer, using local north as $x+$ or polar axis, local vertical up direction as $z+$. Planets location in such system is defined as azimuth A and altitude a,

as (polar)

$$\hat{\mathbf{X}_p} = (R_p, \frac{\pi}{2} - a, A)$$

as we normally do with spherical coordinate system (r, θ, ϕ)

or (dirichlet)

$$\hat{\mathbf{X}_p} = \begin{bmatrix} R_p \cos a \cos A \\ R_p \cos a \sin A \\ R_p \sin a \end{bmatrix}$$

Coordinate transformation

It is easy to transform between the solar-centered ecliptic coordinate system and the earth-centered ecliptic coordinate system. A planet with coordinate $\mathbf{x}_p$ in solar-centered ecliptic coordinate system is at $\mathbf{x}_p - \mathbf{x}_e$ in earth-centered ecliptic coordinate system.

$$\mathbf{x}_p - \mathbf{x}_e = \begin{bmatrix} r_p \cos(\frac{2\pi}{T_{py}}t + \theta_{p0}) - r_e \cos(\frac{2\pi}{T_{ey}}t + \theta_{e0}) \\ r_p \sin(\frac{2\pi}{T_{py}}t + \theta_{p0}) - r_e \sin(\frac{2\pi}{T_{ey}}t + \theta_{e0}) \\ 0 \end{bmatrix}$$

transformation from the earth-centered ecliptic to equatorial coordinate system

$$\begin{bmatrix} x_{\text{equatorial}} \\ y_{\text{equatorial}} \\ z_{\text{equatorial}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varepsilon & -\sin\varepsilon \\ 0 & \sin\varepsilon & \cos\varepsilon \end{bmatrix} \begin{bmatrix} x_{\text{ecliptic}} \\ y_{\text{ecliptic}} \\ z_{\text{ecliptic}} \end{bmatrix}$$

where ecliptic obliquity

$$\varepsilon = 23^\circ 26' 20.512''$$

so we have

$$\begin{bmatrix} R_p \cos\delta\cos\alpha \\ R_p \cos\delta\sin\alpha \\ R_p \sin\delta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varepsilon & -\sin\varepsilon \\ 0 & \sin\varepsilon & \cos\varepsilon \end{bmatrix} \begin{bmatrix} r_p \cos(\frac{2\pi}{T_{py}}t + \theta_{p0}) - r_e \cos(\frac{2\pi}{T_{ey}}t + \theta_{e0}) \\ r_p \sin(\frac{2\pi}{T_{py}}t + \theta_{p0}) - r_e \sin(\frac{2\pi}{T_{ey}}t + \theta_{e0}) \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} r_p \cos(\frac{2\pi}{T_{py}}t + \theta_{p0}) - r_e \cos(\frac{2\pi}{T_{ey}}t + \theta_{e0}) \\ \cos\varepsilon(r_p \sin(\frac{2\pi}{T_{py}}t + \theta_{p0}) - r_e \sin(\frac{2\pi}{T_{ey}}t + \theta_{e0})) \\ \sin\varepsilon(r_p \sin(\frac{2\pi}{T_{py}}t + \theta_{p0}) - r_e \sin(\frac{2\pi}{T_{ey}}t + \theta_{e0})) \end{bmatrix}$$

transformation from equatorial to horizontal coordinate system

$$\cos A \cdot \cos a = -\cos\phi \cdot \sin\delta + \sin\phi \cdot \cos\delta \cdot \cos H$$
$$\sin A \cdot \cos a = \cos\delta \cdot \sin H$$
$$\sin a = \sin\phi \cdot \sin\delta + \cos\phi \cdot \cos\delta \cdot \cos H$$

or

$$\begin{bmatrix} \cos A \cdot \cos a \\ \sin A \cdot \cos a \\ \sin a \end{bmatrix} = \begin{bmatrix} \sin\phi & 0 & -\cos\phi \\ 0 & 1 & 0 \\ \cos\phi & 0 & \sin\phi \end{bmatrix} \begin{bmatrix} \cos\delta\cos H \\ \cos\delta\sin H \\ \sin\delta \end{bmatrix}$$

where hour angle

$$H(t, \alpha) = GST(t) + \lambda - \alpha$$

One of the final goal of this project is to predict A and a with t, given longitude λ and latitude ϕ under specific model, which we are to try explaing.

Note on the physical model

- Planets are actually in ellipse orbits instead of circle ones and z is not 0 to be precise, here we made some simplification just to show the complexity of the problem

- A slow motion of Earth's axis, precession, causes a slow, continuous turning of the coordinate system westward about the poles of the ecliptic, completing one circuit in about 26,000 years.