International Baccalaureate

Extended essay cover

Candidates must complete this page and then give this cover and their final version of the extended essay to their supervisor.			
Candidate session number			
Candidate name			
School number			
School name			
Examination session (May or November)	May	Year	2013

Diploma Programme subject in which this extended essay is registered: Physics

(For an extended essay in the area of languages, state the language and whether it is group 1 or group 2.)

Title of the extended essay: An investigation into the Gaussian and the Laplacian method for the determination of a near-Earth asteroid's orbit using three observations of its position

Candidate's declaration

This declaration must be signed by the candidate; otherwise a grade may not be issued.

The extended essay I am submitting is my own work (apart from guidance allowed by the International Baccalaureate).

I have acknowledged each use of the words, graphics or ideas of another person, whether written, oral or visual.

I am aware that the word limit for all extended essays is 4000 words and that examiners are not required to read beyond this limit.

This is the final version of my extended essay.

Candidate's signature:

Date:

Supervisor's report and declaration

The supervisor must complete this report, sign the declaration and then give the final version of the extended essay, with this cover attached, to the Diploma Programme coordinator.

Name of supervisor (CAPITAL letters)

Please comment, as appropriate, on the candidate's performance, the context in which the candidate undertook the research for the extended essay, any difficulties encountered and how these were overcome (see page 13 of the extended essay guide). The concluding interview (viva voce) may provide useful information. These comments can help the examiner award a level for criterion K (holistic judgment). Do not comment on any adverse personal circumstances that may have affected the candidate. If the amount of time spent with the candidate was zero, you must explain this, in particular how it was then possible to authenticate the essay as the candidate's own work. You may attach an additional sheet if there is insufficient space here.

was initially interested in performing an EE in chemistry, as he is greatly interested in electrochemical cells and re-chargeable batteries. He spent a great deal of time building cells, but was unable to generate a research question he was satisfied with.

During the summer between the first and second years of the course, he had the opportunity to attend a summer science program (SSP) in the U.S. During this time, he became very interested in astronomy and came up with an idea for his research question. This ~~allowd~~ allowed him the opportunity to utilize the telescopes available to him at the SSP and to apply some newly-learned

(continued on back flap)

This declaration must be signed by the supervisor; otherwise a grade may not be issued.

I have read the final version of the extended essay that will be submitted to the examiner.

To the best of my knowledge, the extended essay is the authentic work of the candidate.

I spent 2.5 hours with the candidate discussing the progress of the extended essay.

Supervisor's signature: Date:

programming skills.

A letter outlining the support received at the SSP is included with his essay. Most of the time that I spent with (including the viva voce) involved asking him to explain to me how he went about constructing his ideas, so that I could valida the authenticity of this essay.

Assessment form (for examiner use only)

Criteria	maximum	Examiner 2	maximum	Examiner 3
	Achievement level			
A research question	2	1	2	
B introduction	2	2	2	
C investigation	4	3	4	
D knowledge and understanding	4	3	4	
E reasoned argument	4	4	4	
F analysis and evaluation	4	4	4	
G use of subject language	4	3	4	
H conclusion	2	2	2	
I formal presentation	4	4	4	
J abstract	2	2	2	
K holistic judgment	4	3	4	
Total out of 36		31		

An investigation into the Gaussian and the Laplacian method for the determination of a near-Earth asteroid's orbit using three observations of its position

May 2013

Supervisor:

Word Count: 3296

Abstract

This paper explores a branch of astronomy called orbit determination to answer the following research question: "Between the Gaussian and the Laplacian method of orbit determination, which method calculates the orbit of a near-Earth asteroid more accurately?"

Initially, I compare and contrast the derivations of the Gaussian and the Laplacian method, identifying their strengths and weaknesses. The hypothesis, drawn from the derivations, claims that the Gaussian method is more accurate than the Laplacian method when the time intervals between the three observations are unequal. To investigate whether this hypothesis also holds true for the asteroid (5626) 1991FE, I collect observational data of the asteroid's position by taking digital images of the asteroid, processing the images, centroiding, and conducting Least Squares Plate Reduction (LSPR). Then, I analyze the data with the computer softwares that I create in the VPython programming language, using the three observations of the asteroid's position as the input.

The data analysis leads to the conclusion that the Gaussian method is more accurate than the Laplacian method at calculating the orbital elements of the asteroid 1991FE when the time intervals between the observations are unequal, thus confirming the hypothesis. In fact, the Gaussian method proves to be more accurate than the Laplacian method even when the time intervals between the observations are equal. The shortcomings of the Laplacian method are thought to be the repercussions of concentrating in the middle observation, truncating the Taylor series expansion of $\hat{\rho}$, and approximating the Earth-Sun mass ratio.

Both methods of orbit determination have several limitations, such as the light travel time correction, the stellar aberration, and the assumption of a two-body problem. Further investigation could examine the long-term dynamical behaviors of the asteroid based on the calculated orbital elements.

Word Count: 288

Table of Contents

1. Introduction

Orbit determination refers to the estimation and the calculation of orbits of celestial objects. The purpose of this paper is to investigate into the Gaussian and the Laplacian method of near-Earth asteroid (NEA) orbit determination to identify which method calculates the orbit of a near-Earth asteroid more accurately. I formally derive the two methods from the fundamental laws of physics and create computer softwares in the VPython programming language to model the two methods. Ultimately, I compare and contrast the two methods using the observational data collected for the asteroid (5626) 1991FE.

It is crucial to investigate which method of orbit determination works better in distinctive situations because an accurate estimation of orbital elements helps us predict the potential paths the NEAs might take. We must keep in mind that there is always the possibility that an asteroid might hit the earth and cause a large number of deaths. Therefore, I choose target NEA, which is potentially hazardous and poorly measured to date, and once this research is done, I intend to submit the observations of 1991FE's position to the Minor Planet Center (MPC) in order to make a tangible contribution to the scientific community.

In section 2, I explain how $\vec{r}$ and $\dot{\vec{r}}$, which are essential in calculating the orbital elements, cannot be determined directly from right ascension, declination, and time, the three data obtained through observations. Therefore, in section 4 and 5, I derive the Gaussian and the Laplacian method from Newton's law of universal gravitation and Kepler's laws of planetary motion. In section 7, I describe how $\vec{r}$ and $\dot{\vec{r}}$, obtained through the Gaussian and the Laplacian method, are transformed from equatorial coordinates to ecliptic coordinates. In section 8, I introduce the six orbital elements and demonstrate how they can be obtained via $\vec{r}$ and $\dot{\vec{r}}$.

2. From Observations to $\vec{r}$, $\dot{\vec{r}}$

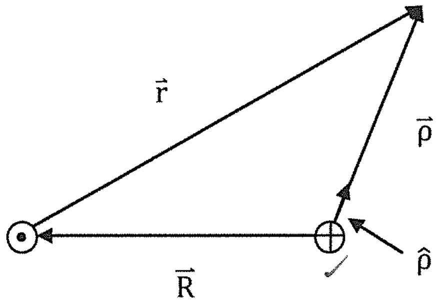

Figure 1. Earth-Sun-Asteroid Vectors

Figure 1 demonstrates the three vectors that connect the Earth, the Sun, and the asteroid.

Let:

- $\vec{R}$ be the vector from the Earth to the Sun (this can obtained via JPL Horizons),
- $\vec{r}$ be the vector from the Sun to the asteroid, and
- $\vec{\rho}$ be the vector from the Earth to the asteroid.

The relationship between these vectors are given by

$$\vec{r} = \vec{\rho} - \vec{R} = \rho\hat{\rho} - \vec{R} \tag{1}$$

$$\dot{\vec{r}} = \dot{\vec{\rho}} - \dot{\vec{R}} = \dot{\rho}\hat{\rho} + \rho\dot{\hat{\rho}} - \dot{\vec{R}} \tag{2}$$

$$\ddot{\vec{r}} = \ddot{\rho}\hat{\rho} + 2\dot{\rho}\dot{\hat{\rho}} + \rho\ddot{\hat{\rho}} - \ddot{\vec{R}} \tag{3}$$

To find the six orbital elements, I need the position and the velocity vector from the Sun to the asteroid ($\vec{r}$ and $\dot{\vec{r}}$). However, the only data collected from the observations are the right ascension (α), the declination (δ), and the time (t). Looking at Figure 1, notice that the unit vector from the Earth to the asteroid ($\hat{\rho}$) is the only quantity that I can calculate directly from my observations:

$$\hat{\rho} = (\cos\alpha\cos\delta)\hat{\imath} + (\sin\alpha\cos\delta)\hat{\jmath} + (\sin\delta)\hat{k} \tag{4}$$

To determine $\vec{\rho}$, $\vec{r}$, and their derivatives, I have to resort to the Gaussian or the Laplacian method of orbit determination.

3. Short Comment on Units

Let me introduce a new time unit, τ, which will make calculations easier later on.

Kepler's third law, as generalized by Newton, is given by

$$P^2 = \frac{4\pi^2}{GM} a^3 \tag{5}$$

where P is the orbital period of the object and a is the semi-major axis of the orbit.

Rearranging, I have:

$$P = \frac{2\pi a^{\frac{3}{2}}}{\sqrt{GM_S}} \tag{6}$$

Since the mass of the asteroid is negligible compared to the mass of the Sun, it is not included in equation (6).

Let $k = \sqrt{GM_S}$, then

$$P = \frac{2\pi a^{\frac{3}{2}}}{k} \tag{7}$$

In 1809, Gauss calculated k = 0.01720209895, when a is in Astronomical Units (AU) and P is in Gaussian days (τ). For P to be in Gaussian days, the time unit must be converted from t to τ:

$$\tau = kt \tag{8}$$

The reasoning is as follows:

The equation of orbital motion gives

$$\ddot{\vec{r}} = -\frac{GM_S\vec{r}}{r^3} \tag{9}$$

Substitute k and note that

$$\frac{d^2\vec{r}}{dt^2} = -k^2\frac{\vec{r}}{r^3} \tag{10}$$

Rewrtie this as

$$\frac{1}{k^2}\frac{d^2\vec{r}}{dt^2} = -\frac{\vec{r}}{r^3} \tag{11}$$

and given $d\tau = kdt$ (from equation (8)) and $d\tau^2 = k^2dt^2$,

$$\frac{d^2\vec{r}}{d\tau^2} = -\frac{\vec{r}}{r^3} \tag{12}$$

Recognize that equation (12) corresponds to equation (9) and that $GM_S = 1$ when the time unit is in Guassian days (τ) instead of Julian days (t).

4. Gaussian Method of Orbit Determination

The Gaussian method has its basis in celestial geometry. In figure 2, the three vectors $\overrightarrow{r_{-1}}$, $\overrightarrow{r_0}$, and $\overrightarrow{r_{+1}}$ define 3 sectors, B_1, B_2, and B_3. B_2 is the combined area of B_1 and B_3.

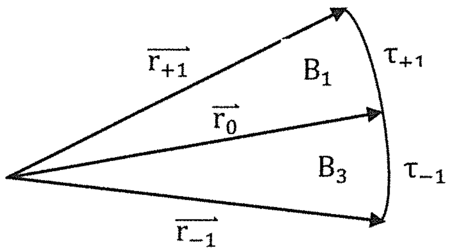

Figure 2. Gaussian Geometry

4.1 Initial approximations

Since the asteroid is part of a Keplerian orbit, it lies on a single plane. Therefore, the vectors are linealy dependent and the middle vector can be described as follows:

$$\overrightarrow{r_0} = a_1\overrightarrow{r_{-1}} + a_3\overrightarrow{r_{+1}} \tag{13}$$

where a_1, a_3 is initally estimated using sectors:

$$a_1 = \frac{B_1}{B_2} = \frac{\tau_{+1}}{\tau_0} \tag{14}$$

$$a_3 = \frac{B_3}{B_2} = -\frac{\tau_{-1}}{\tau_0} \tag{15}$$

Substitute equation (13) into equation (1) to get:

$$\overrightarrow{\rho_0} = \overrightarrow{R_0} + a_1\overrightarrow{r_{-1}} + a_3\overrightarrow{r_{+1}} \tag{16}$$

$$= \overrightarrow{R_0} + a_1(\overrightarrow{\rho_{-1}} - \overrightarrow{R_{-1}}) + a_3(\overrightarrow{\rho_{+1}} - \overrightarrow{R_{+1}}) \tag{17}$$

Perceive that everything is now written in terms of known quantities except for the vectors $\overrightarrow{\rho_{-1}}$, $\overrightarrow{\rho_0}$, and $\overrightarrow{\rho_{+1}}$. These are parametrized as following:

$$\vec{\rho} = \begin{pmatrix} \hat{\imath}\,\rho \\ \hat{\jmath}\,\rho \\ \hat{k}\,\rho \end{pmatrix} \tag{18}$$

Note that $\hat{\imath}$, $\hat{\jmath}$, and $\hat{k}$ are components of $\hat{\rho}$ (refer to equation(1)). Thus, equation (17) is three equations with the unknown scalars ρ_{-1}, ρ_0, and ρ_{+1}, which can be solved using Cramer's rule. Another method is to use vector products on equation (17) to get:

$$\rho_{-1} = \frac{a_1(\overrightarrow{R_{-1}} \times \widehat{\rho_0}) \cdot \widehat{\rho_{+1}} - (\overrightarrow{R_0} \times \widehat{\rho_0}) \cdot \widehat{\rho_{+1}} + a_3(\overrightarrow{R_{+1}} \times \widehat{\rho_0}) \cdot \widehat{\rho_{+1}}}{a_1(\widehat{\rho_{-1}} \times \widehat{\rho_0}) \cdot \widehat{\rho_{+1}}} \tag{19}$$

$$\rho_0 = \frac{a_1(\widehat{\rho_{-1}} \times \overrightarrow{R_{-1}}) \cdot \widehat{\rho_{+1}} - (\widehat{\rho_{-1}} \times \overrightarrow{R_0}) \cdot \widehat{\rho_{+1}} + a_3(\widehat{\rho_{-1}} \times \overrightarrow{R_{+1}}) \cdot \widehat{\rho_{+1}}}{-(\widehat{\rho_{-1}} \times \widehat{\rho_0}) \cdot \widehat{\rho_{+1}}} \tag{20}$$

$$\rho_{+1} = \frac{a_1(\widehat{\rho_0} \times \overrightarrow{R_{-1}}) \cdot \widehat{\rho_{-1}} - (\widehat{\rho_0} \times \overrightarrow{R_0}) \cdot \widehat{\rho_{-1}} + a_3(\widehat{\rho_0} \times \overrightarrow{R_{+1}}) \cdot \widehat{\rho_{-1}}}{a_3(\widehat{\rho_0} \times \widehat{\rho_{+1}}) \cdot \widehat{\rho_{-1}}} \tag{21}$$

Substituting ρ_{-1}, ρ_0, and ρ_{+1} in equation (1) yields $\overrightarrow{r_{-1}}$, $\overrightarrow{r_0}$, and $\overrightarrow{r_{+1}}$. I can also approximate $\dot{\vec{r}}$ and obtain the six orbital elements, but these elements will not be accurate, which is why I move on to explore the f and g series.

4.2 The f and g series

The f and g series are paramount for the Gaussian method because they give better estimations of a_1 and a_3. Remember from section 3 that $GM_S = 1$ when the time unit is in Gaussian days (τ).

Expand $\vec{r}(\tau)$ in a Taylor series about the middle observation.

$$\vec{r}(\tau) = \overrightarrow{r_0} + \dot{\overrightarrow{r_0}}\tau + \ddot{\overrightarrow{r_0}}\frac{\tau^2}{2} + (\frac{d^3\overrightarrow{r_0}}{d\tau^3})\frac{\tau^3}{6} + \cdots \tag{22}$$

Newton's Law of Universal Gravitation states:

$$\ddot{\overrightarrow{r_0}} = -\frac{GM_S\overrightarrow{r_0}}{{r_0}^3} \tag{23}$$

Differentiate this with respect to time:

$$\frac{d^3\overrightarrow{r_0}}{dt^3} = -GM_S\left[\frac{r_0^3\dot{\overrightarrow{r_0}} - 3r_0^2\overrightarrow{r_0}\dot{r_0}}{r_0^6}\right] \tag{24}$$

Since $\overrightarrow{r_0} \cdot \dot{\overrightarrow{r_0}} = r_0\dot{r}_0$,

$$\frac{d^3\overrightarrow{r_0}}{dt^3} = -GM_S\left[\frac{r_0^2\dot{\overrightarrow{r_0}} - 3\overrightarrow{r_0}(\overrightarrow{r_0}\cdot\dot{\overrightarrow{r_0}})}{r_0^5}\right] \tag{25}$$

Substitute equations (23) and (25) into (22), collect like terms, and eliminate terms higher than the third order to get:

$$\vec{r}(\tau) = [1 - \frac{\tau^2}{2r_0^3} + \frac{\tau^3(\vec{r_0} \cdot \dot{\vec{r_0}})}{2r_0^5}]\,\vec{r_0} + [\tau - \frac{\tau^3}{6r_0^3}]\,\dot{\vec{r_0}} \tag{26}$$

or

$$\vec{r}(\tau) = f(\tau)\vec{r_0} + g(\tau)\dot{\vec{r_0}} \tag{27}$$

where

$$f(\tau) = 1 - \frac{\tau^2}{2r_0^3} + \frac{\tau^3(\vec{r_0} \cdot \dot{\vec{r_0}})}{2r_0^5} \tag{28}$$

$$g(\tau) = \tau - \frac{\tau^3}{6r_0^3} \tag{29}$$

Now we plug the f and g series into equation (27) for the 1st and 3rd observations:

$$\overrightarrow{r_{-1}} = f_{-1}\vec{r_0} + g_{-1}\dot{\vec{r_0}} \tag{30}$$

$$\overrightarrow{r_{+1}} = f_{+1}\vec{r_0} + g_{+1}\dot{\vec{r_0}} \tag{31}$$

Multiply equation (30) by g_{+1} and equation (31) by g_{-1}, subtract one equation from another, and rearrange to get:

$$\vec{r_0} = \frac{g_{+1}}{f_{-1}g_{+1} - f_{+1}g_{-1}}\overrightarrow{r_{-1}} - \frac{g_{-1}}{f_{-1}g_{+1} - f_{+1}g_{-1}}\overrightarrow{r_{+1}} \tag{32}$$

Notice that this is equation (13). Therefore,

$$a_1 = \frac{g_{+1}}{f_{-1}g_{+1} - f_{+1}g_{-1}} \tag{33}$$

$$a_3 = -\frac{g_{-1}}{f_{-1}g_{+1} - f_{+1}g_{-1}} \tag{34}$$

Next, to find $\dot{\vec{r_0}}$, we rearrange equation (30) and (31) to yield:

$$\dot{\vec{r_0}} = \frac{\overrightarrow{r_{-1}} - \overrightarrow{f_{-1}r_0}}{g_{-1}} \tag{35}$$

$$\dot{\vec{r_0}} = \frac{\overrightarrow{r_{+1}} - \overrightarrow{f_{+1}r_0}}{g_{+1}} \tag{36}$$

Average the two $\dot{\vec{r_0}}$ vectors to obtain a better approximation.

5. Laplacian Method of Orbit Determination

Unlike the Gaussian method, which is based on the celestial geometry of sectors and arcs, the Laplacian method is focused on the mathematical manipulations of Newton and Kepler's laws.

5.1 Determination of $\hat{\rho}$, $\dot{\hat{\rho}}$, and $\ddot{\hat{\rho}}$ for Laplacian method

In the Laplacian method, $\vec{r}$ and $\dot{\vec{r}}$ are determined via $\hat{\rho}$, $\dot{\hat{\rho}}$, and $\ddot{\hat{\rho}}$.

To obtain $\dot{\hat{\rho}}$ and $\ddot{\hat{\rho}}$, $\hat{\rho}$ is expanded in a Taylor series aboout the middle observation:

$$(\hat{\rho}_{-1} - \hat{\rho}_0) = \dot{\hat{\rho}}(t_{-1} - t_0) + \ddot{\hat{\rho}}\frac{(t_{-1} - t_0)^2}{2} + \cdots \tag{37}$$

$$(\hat{\rho}_{+1} - \hat{\rho}_0) = \dot{\hat{\rho}}(t_{+1} - t_0) + \ddot{\hat{\rho}}\frac{(t_{+1} - t_0)^2}{2} + \cdots \tag{38}$$

Drop the terms higher than dt^2 and solve simultaneously for $\dot{\hat{\rho}}$ and $\ddot{\hat{\rho}}$ to obtain:

$$\dot{\hat{\rho}} = \frac{(t_{+1} - t_0)^2(\hat{\rho}_{-1} - \hat{\rho}_0) - (t_{-1} - t_0)^2(\hat{\rho}_{+1} - \hat{\rho}_0)}{(t_{+1} - t_0)(t_{-1} - t_0)(t_{+1} - t_{-1})} \tag{39}$$

$$\ddot{\hat{\rho}} = -2[\frac{(t_{+1} - t_0)(\hat{\rho}_{-1} - \hat{\rho}_0) - (t_{-1} - t_0)(\hat{\rho}_{+1} - \hat{\rho}_0)}{(t_{+1} - t_0)(t_{-1} - t_0)(t_{+1} - t_{-1})}] \tag{40}$$

Using the Gaussian units from section 3,

$$\dot{\hat{\rho}} = \frac{\tau_{+1}^2(\hat{\rho}_{-1} - \hat{\rho}_0) - \tau_{-1}^2(\hat{\rho}_{+1} - \hat{\rho}_0)}{\tau_{+1}\tau_{-1}\tau_0} \tag{41}$$

$$\ddot{\hat{\rho}} = -2[\frac{\tau_{+1}(\hat{\rho}_{-1} - \hat{\rho}_0) - \tau_{-1}(\hat{\rho}_{+1} - \hat{\rho}_0)}{\tau_{+1}\tau_{-1}\tau_0}] \tag{42}$$

5.2 Finding r and ρ

Use equation (1) to re-write the Netwon's Law of Universal Gravitation:

$$\ddot{\vec{r}} = -\frac{GM_S\vec{r}}{r^3} = -\frac{GM_S(\vec{\rho} - \vec{R})}{r^3} \tag{43}$$

$$\ddot{\vec{R}} = -\frac{G(M_S + M_E)\vec{R}}{R^3} \tag{44}$$

where M_S is the mass of the Sun and M_E is the mass of the Earth.

The mass of the asteroid is not included in equation (43) because it is negligible compared to the mass of the Sun.

Susbtitute equations (43) and (44) into equation (3) to get

$$-\frac{GM_S(\vec{\rho} - \vec{R})}{r^3} = \ddot{\rho}\hat{\rho} + 2\dot{\rho}\dot{\hat{\rho}} + \rho\ddot{\hat{\rho}} + \frac{G(M_S + M_E)\vec{R}}{R^3} \tag{45}$$

And isolate $\vec{R}$ and ρ on opposite sides of the equation to yield

$$G\left(\frac{M_S}{r^3} - \frac{M_S + M_E}{R^3}\right)\vec{R} = \ddot{\rho}\hat{\rho} + 2\dot{\rho}\dot{\hat{\rho}} + \rho\ddot{\hat{\rho}} + \frac{GM_S}{r^3}\vec{\rho} \ . \tag{46}$$

This equation can be simplified by taking the dot product of both sides with $(\hat{\rho}\times\dot{\hat{\rho}})$ and $(\hat{\rho}\times\ddot{\hat{\rho}})$.

$$[\hat{\rho}\times\dot{\hat{\rho}}\cdot\vec{R}]G\left(\frac{M_S}{r^3} - \frac{M_S + M_E}{R^3}\right) = \rho[\hat{\rho}\times\dot{\hat{\rho}}\cdot\ddot{\hat{\rho}}] \tag{47}$$

$$[\hat{\rho}\times\ddot{\hat{\rho}}]\cdot\vec{R}]G\left(\frac{M_S}{r^3} - \frac{M_S + M_E}{R^3}\right) = 2\dot{\rho}[\hat{\rho}\times\ddot{\hat{\rho}}\cdot\dot{\hat{\rho}}]$$

$$= -2\dot{\rho}[\hat{\rho}\times\dot{\hat{\rho}}\cdot\ddot{\hat{\rho}}] \tag{48}$$

Rearrange equation (47) to obtain

$$\rho = G\left(\frac{M_S}{r^3} - \frac{M_S + M_E}{R^3}\right)\left[\frac{\hat{\rho}\times\dot{\hat{\rho}}\cdot\vec{R}}{\hat{\rho}\times\dot{\hat{\rho}}\cdot\ddot{\hat{\rho}}}\right] \tag{49}$$

$$= \left(\frac{1}{r^3} - \frac{1 + \frac{1}{328900.5}}{R^3}\right)\left[\frac{\hat{\rho}\times\dot{\hat{\rho}}\cdot\vec{R}}{\hat{\rho}\times\dot{\hat{\rho}}\cdot\ddot{\hat{\rho}}}\right] \tag{50}$$

Dot product of the equation (1) with itself gives:

$$r^2 = \rho^2 + R^2 - 2\vec{R}\cdot\vec{\rho} \ . \tag{51}$$

Rearranging, we get

$$r = \sqrt{\rho^2 + R^2 - 2\vec{R}\cdot\vec{\rho}} \ . \tag{52}$$

To solve equations (50) and (52), iterate as follows:

1. Make a reasonable initial guess for r, for example r = 2. AU
2. Calculate ρ from this r using equation (50).
3. Calculate a new r from this ρ using equation (52).
4. Repeat steps 2 and 3 until convergence.

After convergence, obtain $\dot{\rho}$ using equation (48)

$$\dot{\rho} = -\frac{G}{2}\left(\frac{M_S}{r^3} - \frac{M_S + M_E}{R^3}\right)\left[\frac{\hat{\rho}\times\ddot{\hat{\rho}}]\cdot\vec{R}}{\hat{\rho}\times\dot{\hat{\rho}}\cdot\ddot{\hat{\rho}}}\right] \tag{53}$$

$$= -\frac{1}{2}\left(\frac{1}{r^3} - \frac{1 + \frac{1}{328900.5}}{R^3}\right)\left[\frac{\hat{\rho}\times\ddot{\hat{\rho}}]\cdot\vec{R}}{\hat{\rho}\times\dot{\hat{\rho}}\cdot\ddot{\hat{\rho}}}\right] \tag{54}$$

Finally from equation (1) and (2)

$$\vec{r} = \vec{\rho} - \vec{R} = \rho\hat{\rho} - \vec{R} \ , \tag{55}$$

$$\dot{\vec{r}} = \dot{\vec{\rho}} - \dot{\vec{R}} = \dot{\rho}\hat{\rho} + \rho\dot{\hat{\rho}} - \dot{\vec{R}} \ , \tag{56}$$

6. Brief Evaluation of the Gaussian and the Laplacian Method

The use of equations (30), (31), and (32) in the Gaussian method makes sure that the observation intervals do not have to be equal because the equations incorporate the different time intervals by using the f and g series for 1st and 3rd observations. Another advantage of the Gaussian method is that the mass ratio of Earth to Sun is not required, as it is in the Laplacian method (refer to equation (54)).

Meanwhile, the Laplacian method concentrates on the middle observation and reduces the significance of the other two observations, thus rendering it susceptible to failure when the middle observation is poor or when the time intervals are unequal. In addition, since the equations (39) and (40) truncate with the second derivative, the next few terms have to be small, or else the method fails. Hence, if the observational intervals are unequal, the third term would be large, and thus the method is likely to fail.

7. Rotation from Equatorial to Ecliptic Coordinates

It is common for orbital elements in the solar system to be given in ecliptic coordinates, in which the Sun is at the origin and the x, y plane corresponds with the ecliptic plane. However, the asteroid position is given in equatorial coordinates (RA and Dec), and therefore $\vec{r}$ and $\dot{\vec{r}}$ must be rotated into the ecliptic coordinate before calculating the orbital elements.

The following rotation matrix is used to rotate $\vec{r}$ and $\dot{\vec{r}}$ from equatorial to ecliptic coordinates:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varepsilon & \sin\varepsilon \\ 0 & -\sin\varepsilon & \cos\varepsilon \end{pmatrix}$$

The ecliptic tilt $\varepsilon \fallingdotseq 23.4376600557$ is the inclination of the ecliptic relative to the celestial equator.

8. Description of Orbital Elements

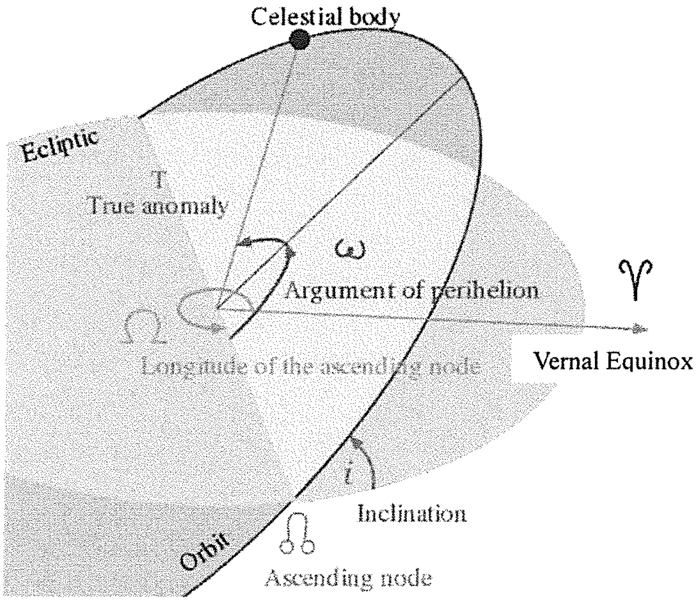

Figure 3. Orbital Elements [1]

NEAs follow an elliptical orbit about the Sun that can be characterized by six orbital elements:

Semi-major axis (a)

NEAs follow an elliptical orbit, satisfying the equation $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$. The semi-major axis is half the distance between the perihelion and the aphelion, thus defining the size of the orbit.

$$a = \frac{1}{[\frac{2}{r} - \dot{\vec{r}} \cdot \dot{\vec{r}}]}$$

Eccentricity (e)

The eccentricity defines the shape of the orbit, and it can be expressed as $e = e = \sqrt{1 - \frac{b^2}{a^2}}$.

For NEAs, $0 \leq e \leq 1$ because $a > b > 0$.

$$\vec{h} = \vec{r} \times \dot{\vec{r}}$$

$$e = \sqrt{1 - \frac{h^2}{a}}$$

Inclination (i)

The inclination is the angle between the orbit plane and the ecliptic plane, in which the Earth orbits. Hence, the inclination defines the orientation of the orbit with respect to the Earth's equator.

$$i = \arctan\left(\frac{\sqrt{h_x^2 + h_y^2}}{h_z}\right)$$

Longitude of the Ascending Node (Ω)

The longitude of the ascending node is the angle between the Vernal Equinox and the ascending node. It defines the location of the ascending and the descending orbit with respect to the Earth's equatorial plane.

$$\Omega = \arcsin\left(\frac{h_x}{h\sin i}\right) \qquad \Omega = \arccos\left(-\frac{h_y}{h\sin i}\right)$$

Argument of Perihelion (ω)

The argument of perihelion is the angle measured from the ascending node to the object's perihelion point.

$$\vec{r} = x\hat{\imath} + y\hat{\jmath} + z\hat{k}$$

True anomaly (v)

$$U = \arccos\left(\frac{x\cos\Omega + y\sin\Omega}{r}\right) \qquad v = \arccos\left(\frac{a(1-e^2)-r}{er}\right)$$

$$U = \arcsin\left(\frac{z}{r\sin i}\right) \qquad v = \arcsin\left(\frac{a(1-e^2)}{eh}\frac{\vec{r}\cdot\dot{\vec{r}}}{r}\right)$$

$$\omega = U - v$$

Mean anomaly (M)

The mean anomaly is the angle measured from the perihelion point to the object's position, assuming unifrom motion. It determines the object's current position along its orbit.

Eccentric anomaly (E)

$$E = \arccos\left(\frac{a-r}{ae}\right)$$

There is no ambiguity in the quadrant of E, since $0° \leq E \leq 180°$ when $0° \leq v \leq 180°$.

$$M = E - e\sin E$$

9. Observational Data

Over a period of a month in summer 2012, I took digital images of the asteroid (5626) 1991 FE using three telescopes: Hut Observatory, 14″ Meade, and 16″ Prompt 2. More information on each telescope is provided in Appendix 1.

Before each observation, I generated the ephemerides using JPL Horizons and created the star chart using TheSkyX, in order to approximate the position of the asteroid, to determine its transit time, and to decide with which stars to sync the telescope. Most of the time, I took three series of seven images of two minute exposures. This was to prevent the asteroid from streaking, while assuring its visibility.

Once the raw images were taken from the CCD, I created the calibrated images by removing gain and bias. This involved reducing, aligning, and combining the images using CCDSoft. After finding the asteroid by blinking the images, I did astrometry using CCDSoft and TheSkyX to determine the right ascension and declination of the asteroid. I also had to determine the middle time of observation by looking at the FITS header info of the images.

Following are the data obtained through observations:

Table 1. Time, Right Ascension, and Declination of (5626) 1991 FE on Five Observations

Observation	Time (± 0.01s)	Right Ascension (± 0.01s)	Declination (± 0.01″)
1	19 June 2012 06h 08m 29.877s UT	18h 19m 38.98s	-16° 59′ 50.81″
2	10 July 2012 09h 12m 39s UT	17h 53m 0.69s	-17° 8′ 15.64″
3	14 July 2012 07h 37m 54s UT	17h 48m 23.58s	-17° 11′ 41.33″
4	18 July 2012 04h 03m 55s UT	17h 44m 09.12s	-17° 15′ 42.75″
5	22 July 2012 03h 36m 25s UT	17h 40m 08.02s	-17° 20′ 36.55″

10. VPython codes

After collecting the observational data, I wrote my own computer software in the VPython programming language for the Gaussian and the Laplacian method of orbit determination. The input files include the time, the right ascension (RA), the declination (Dec), and the Earth-Sun vector ($\vec{R}$) for three observations. Appendix 2 and Appendix 3 are the VPython codes that calculate $\vec{r}$ and $\dot{\vec{r}}$ through the Gaussian and the Laplacian method, respectively, and use those vectors to determine the six orbital elements.

Before using the two VPython codes to analyze the data, I first test the validity of the algorithms through test files generated with JPL Horizons. For the test input file and result, refer to Appendix 4. Following is the analysis of the test file, comparing the orbital elements generated by my programs and those published on JPL Horizons:

Table 2. Evaluating the Gaussian and the Laplacian VPython Codes through Test File

Orbital Elements	Percent Error* / % ($\pm 1\times10^{-8}$)	
	Gaussian Method**	Laplacian Method**
Semi-major axis (a)	0.70270627	1.9027919
Eccentricity (e)	1.5457482	1.3527382
Inclination (i)	0.21054808	0.13252035
Longitude of the ascending node (Ω)	0.12230530	0.22581141
Argument of the perihelion (w)	0.15455159	0.91301850
Mean anomaly (M)	0.54082963	0.53577797

$$* \ \% \text{ error} = \frac{(\text{calculated value} - \text{known value})}{\text{known value}} * 100\%$$

** Confident to 8 significant figures because of RA and Dec

Keeping in mind that the data published on JPL Horizons for (5626) 1991FE are not very accurate due to the lack of data and taking into account the small percentage errors that are all less than 2%, I can safely conclude that my programs are working and viable as tools for analysis of the real data collected.

11. Data Analysis

In section 11, I will analyze the suitability of the Gaussian and the Laplacian method with respect to the distribution of the observations. In other words, I will examine whether the Gaussian or the Laplacian method is affected when the observation intervals are unequal. Different combinations of the five observational data will be used as inputs for the programs that I have created. Once the orbital elements have been obtained using the software, they will be compared with those published by JPL Horizons. The comparison of the percentage errors of the six orbital elements will indicate which method is more accurate.

The combinations that will be analyzed are:

Unequal intervals

- Observations 1, 2, and 5
- Observations 1, 2, and 4
- Observations 1, 2, and 3
- Observations 1, 3, and 5
- Observations 1, 3, and 4

Equal intervals

- Observations 2, 3, and 4
- Observations 3, 4, and 5

Sample orbital elements have been obtained using observations 1, 2, and 5. Table 3 presents the orbital elements obtained via the Gaussian and the Laplacian method, and Table 4 presents their respective percent errors compared to the values published on JPL Horizons.

Table 3. Orbital Elements of Observations 1, 2, and 5

Orbital Element	JPL Horizons	Gaussian Method*	Laplacian Method*
Semi-major axis (a)	2.195255848576980AU	2.1732156	3.4190337
Eccentricity (e)	0.4543064682717765	0.45132687	0.63575571
Inclination (i)	3.854139495825758°	3.8568536	4.2372584
Longitude of the ascending node (Ω)	173.2889321893739°	173.79201	161.73502
Argument of the perihelion (w)	231.4186146804029°	232.56428	222.19316
Mean anomaly (M)	282.2478103012744°	279.50150	327.46415

* Confident to 8 significant figures because of RA and Dec

Table 4. Percent Errors of Orbital Elements of Observations 1, 2, and 5

Orbital Elements	Percent Error % ($\pm 1\times10^{-8}$)*	
	Gaussian Method*	Laplacian Method*
Semi-major axis (a)	1.0039931	55.746480
Eccentricity (e)	0.65585598	39.939833
Inclination (i)	0.070419298	9.9404523
Longitude of the ascending node (Ω)	0.29031333	6.6674271
Argument of the perihelion (w)	0.49506221	3.9864787
Mean anomaly (M)	0.97301391	16.020085

* Confident to 8 significant figures because of RA and Dec

From Table 4, notice that the Guassian method is significantly more accurate than the Laplacian method for the observations 1, 2, and 5. For example, the percent error of the semi-major axis calculated through the Gaussian method is approximately 1%, while that calculated through the Laplacian method is approximately 56%. A key aspect to notice is the unequal time interval between the three observations. The time interval between observation 1 and 2 is 21 days and the time interval between observation 2 and 5 is 12 days, so the overall difference between the time intervals is 9 days (21 – 12 = 9). The 9 day difference in the time intervals seems to be the reason behind the failure of the Laplacian method. This is also corroborated by the mathematics and physics behind the derivation of the Laplacian method. As explored in section 6, the concentration in the middle observation and the truncation of the Taylor series expanision leave the Laplacian method susceptible to failure when the time intervals are unequal. To verify this phenomena and check the validity of my conjecture, I have analyzed the five combinations of observations with unequal intervals and I have attempted to find patterns that are applicable to a general situation. The results for these combinations are reported in Appendix 5.

The main trend noticed from the analysis of data is that the Laplacian method, unlike the Gaussian method, is greatly influenced by the distribution of the observations. The

Laplacian method becomes increasingly more inaccurate when the difference between the time intervals increases. This discovery is summarized in the following table:

Table 5. Correlation between time intervals and accuracy of Laplacian method

Observations	1st Interval / day (± 0.5)*	2nd Interval / day (± 0.5)*	Difference between Time Intervals / day (± 1)**	Percent Error of Semi-major Axis / % (± 1)
1, 2, 5	21	12	9	56
1, 2, 4	21	8	13	93
1, 2, 3	21	4	17	160
1, 3, 5	25	8	17	224
1, 3, 4	25	4	21	580

*There is uncertainty because observations were not taken at the same time each day.

** Addition of the uncertainties of the two intervals

Meanwhile, when the time intervals are equal, as is the case with observations 3, 4, and 5 and observations 2, 3, and 4, the Laplcian method appears much more stable and useful. In fact, the Laplacian method is more accurate than the Gaussian method when it comes to observations 3, 4, and 5, although the Gausian method is once again significantly better than the Laplacian method for observations 2, 3, and 4. I was not able to find an appropriate explanation for such phenomena through this research paper. More data of equal intervals are necessary.

12. Conclusion and Evaluation

This research paper investigates into the Gaussian and the Laplacian method of orbit determination for the near-Earth asteroid, (5626) 1991 FE, using three observations of its position. Since only three observations are used as the input, the orbital elements calculated through the computer programs are only approximate. However, depending on the method and the observations, surprisingly good results can be obtained.

The Gaussian method of orbit determination is generally better than the Laplacian method. Referring to the percent errors of the orbital elements calculated via computer softwares, we realize that those of the Gaussian method is all less than 3%, except for the anomalous observations 3, 4, and 5. Meanwhile, the Laplacian method is significantly worse than the Gaussian method, especially when the time intervals are unequal. However, the Laplacian method did prove to be more effective when the time intervals are equal.

Overall, the quality of the observations and of the VPython codes of the Gaussian and the Laplacian method suffices to serve the purpose of analyzing the data and contrasting the two methods. Nonetheless, there is still plenty of room for improvement. For example, the inclusion of a fourth observation in the input could improve the orbital elements.

One of the major limitations concerns the travel time of light. The observed positions of the asteroid were not the positions that they occupied at the instant they were observed. When travel time of light is taken into account, the asteroid actually occupies the observed positions at times $t_{-1} - \rho_{-1}/c$, $t_0 - \rho_0/c$ and $t_{+1} - \rho_{+1}/c$. This correction makes a negligible difference in the value of ρ but the orbital elements will change more significantly.

Another major limitation concerns the stellar aberration. Stars appear to be shifted a little ahead of their true position due to the "motion of Earth in its orbit around the sun" and the "finite speed of light" [2]. That aberration is the same for stars and far away asteroids that do not move very fast relative to the Earth. However, that is not true for near-Earth asteroids.

For example, an asteroid in opposition that moves in the same direction and speed as the Earth will appear to be trailing the stars by up to 20". Therefore, I must find the tangential part of the relative velocity of Earth and the asteroid and compare my stellar aberration of a star in that position and my asteroid to find the correction in RA and Dec. Then, I must apply these new star RAs and Decs to the Least Squares Plate Reduction (LSPR) program and obtain new RAs and Decs for all three asteroid positions and reinsert them into my orbit determination program to get better orbital elements.

Last but not least, the two methods make the assumption of a two-body problem, but the orbit of a near-Earth asteroid is evidently an N-body problem. Near-Earth asteroids, such as the (5626) 1991FE for which I have collected digital images, follow an elliptical orbit about the Sun that can be characterized by the six orbital elements: semi-major axis (a), eccentricity (e), inclination (i), longitude of the ascending node (Ω), argument of perihelion (ω), and mean anomaly (M). However, this two-body approximation works if there are only two bodies in the system, the Sun and the asteroid. Fortunately, the two-body approximation works decently well for short periods of time because the Sun is 100 times more massive than anything else in the solar system. The two-body approximation fails over long timescales or if there are close approaches with a planet. As further investigation, I could use a numerical integration program called Swift to examine the long-term dynamical behaviours of the asteroid over the next 50 million years.

Appendix 1. Telescope Vital Stats

14" Meade

aperture = 356mm

focal length = 3556mm

SBIG STL-1301E CCD camera

- 1280x1024 pixel array
- pixel size = 16 microns
- plate scale = 0.928"/pix
- field of view = 20' x 16'

16" Prompt 2

aperture = 407mm

focal length = 4536mm

ALTA U47+ camera

- 3073x2048 pixel array
- pixel size = 9 microns
- plate scale = 0.41"/pix
- field of view = 21' x 14'

HUT Observatory

Apogee CCD model Alta U47, back-illuminated 1024x1024

13-micron pixels

16-inch f/8 reflector

All exposures binned 2x2

CCD temp = -30 degrees

filter = Cousins R

Some high haze.

Exposures 2-minutes

Appendix 2. Gaussian Method VPython Codes

Code: GassuianOrbitDetermination.py

```
# Gaussian Orbit Determination
# Written August 4, 2012

from math import *
from numpy import *
from visual import *

debug = True
iteration = False

# Import data file
if debug == True:
    filename = "JPL2.txt"
else:
    filename = raw_input("Enter name of data file:")

##################################################
##################################################

# Define functions

# Convert (RA) hours, minutes, and seconds to radians
def hms2Rad(hms):
    if hms[0]>=0: #positive RA
        return radians((hms[0] + hms[1]/60. + hms[2]/3600.)*360./24.)
    else: #negative RA
        return radians((hms[0] + hms[1]/60. + hms[2]/3600.)*360./24.)

# Convert (Dec) degrees, arcminutes, and arcseconds to radians
def dms2Rad(dms):
    if dms[0]>=0: #positive Dec
        return radians(dms[0] + dms[1]/60. + dms[2]/3600.)
    else: #negative Dec
        return radians(dms[0] - dms[1]/60. - dms[2]/3600.)

# Convert UT to JD
def UT2JD(dmyhms): #day,month,year,hour,minute,second
    decimalTime = float(dmyhms[3]+(dmyhms[4]/60.)+(dmyhms[5]/3600.))
    Jnaught = 367*dmyhms[2] - int(7*(dmyhms[2]+int((dmyhms[1]+9)/12))/4) +
int(275*dmyhms[1]/9) + dmyhms[0] + 1721013.5
    return Jnaught+(decimalTime/24.)

# Check ambiguity by comparing values obtained through sine and cosine
def ambiguity_check(A,B,C,D):
    if round(A*1e6)==round(C*1e6):
        return A
    elif round(B*1e6)==round(C*1e6):
        return B
    elif round(A*1e6)==round(D*1e6):
        return A
    elif round(B*1e6)==round(D*1e6):
        return B

# Calculate rho unit vector
def rho_hat(ra,dec):
    rho_hat = vector(cos(ra)*cos(dec),sin(ra)*cos(dec),sin(dec))
    return rho_hat
```

```
# f series
def fseries(tau,r,rdot):
    f = 1 - tau**2./(2*mag(r)**3)
    f += (tau**3.)*dot(r,rdot)/(2* mag(r)**5.)
    f += (tau**4./24.)*((3./mag(r)**3.)*(dot(rdot,rdot)/mag(r)**2. -
1./mag(r)**3.) - (15./mag(r)**2.)*dot(rdot,rdot)**2. + (1./mag(r)**6.))
    return f

# g series
def gseries(tau,r,rdot):
    g = tau - (tau**3.)/(6* mag(r)**3.)
    g += tau**4.*dot(rdot,rdot)/(4.*mag(r)**5.)
    return g

####################################################
####################################################

# Constants
mu = 1
k = 0.01720209895 #Boltzmann constant
c = 173.1446 #speed of light (AU/day)
epsilon = radians(23.4376600557) #ecliptic tilt

####################################################
####################################################

# Extract data from file
data = loadtxt(filename,delimiter=',')

t1 = UT2JD(data[0])
ra1 = hms2Rad(data[1][0:3])
dec1 = dms2Rad(data[1][3:])
R1 = vector(data[2][0:3])
R_dot1 = vector(data[2][3:])

t2 = UT2JD(data[3])
ra2 = hms2Rad(data[4][0:3])
dec2 = dms2Rad(data[4][3:])
R2 = vector(data[5][0:3])
R_dot2 = vector(data[5][3:])

t3 = UT2JD(data[6])
ra3 = hms2Rad(data[7][0:3])
dec3 = dms2Rad(data[7][3:])
R3 = vector(data[8][0:3])
R_dot3 = vector(data[8][3:])

# Define Earth-Sun vectors

R = array([R2,R3,R1])
R_dot = array([R_dot2,R_dot3,R_dot1])

####################################################
####################################################

# rho unit vector
rho_hat1 = rho_hat(ra1,dec1)
rho_hat2 = rho_hat(ra2,dec2)
rho_hat3 = rho_hat(ra3,dec3)
rho_hat = array([rho_hat2,rho_hat3,rho_hat1])
```

```
# tau
tau = k * array([t3-t1,t3-t2,t1-t2])

# Estimate a1 and a3 using sectors
a1 = tau[1]/tau[0]
a3 = -tau[-1]/tau[0]

# rho magnitude
rho_mag1 = ( a1*dot(cross(R[-1],rho_hat[0]),rho_hat[1]) -
dot(cross(R[0],rho_hat[0]),rho_hat[1]) +
a3*dot(cross(R[1],rho_hat[0]),rho_hat[1]) ) / ( a1*dot(cross(rho_hat[-
1],rho_hat[0]),rho_hat[1]) )
rho_mag2 = ( a1*dot(cross(rho_hat[-1],R[-1]),rho_hat[1]) -
dot(cross(rho_hat[-1],R[0]),rho_hat[1]) + a3*dot(cross(rho_hat[-
1],R[1]),rho_hat[1]) ) / ( -dot(cross(rho_hat[-1],rho_hat[0]),rho_hat[1]) )
rho_mag3 = ( a1*dot(cross(rho_hat[0],R[-1]),rho_hat[-1]) -
dot(cross(rho_hat[0],R[0]),rho_hat[-1]) +
a3*dot(cross(rho_hat[0],R[1]),rho_hat[-1]) ) /
( a3*dot(cross(rho_hat[0],rho_hat[1]),rho_hat[-1]) )
rho_mag = array([[rho_mag2],[rho_mag3],[rho_mag1]])

# rho vector
rho = rho_hat * rho_mag

# r vector
r = rho - R

# initial velocity estimate
r_dot = (r[1]-r[-1])/(tau[1]-tau[-1])

print "##############"
print "Initial Values"
print "##############"
print "rho unit vector: ",rho_hat
print "rho magnitude: ", rho_mag
print "rho vector: ", rho
print "r vector: ", r
print "r dot vector: ", r_dot,'\n'

for i in range (0,1000):

    # Calculate f&g series
    f = fseries(tau,r[0],r_dot)
    g = gseries(tau,r[0],r_dot)

    # New a1, a3
    a1_new = g[1] / ( f[-1]*g[1] - f[1]*g[-1] )
    a3_new = -g[-1] / ( f[-1]*g[1] - f[1]*g[-1] )

    # rho magnitude
    rho_mag1 = ((a1_new*dot(cross(R[-1],rho_hat[0]),rho_hat[1]))-
dot(cross(R[0],rho_hat[0]),rho_hat[1])+a3_new*dot(cross(R[1],rho_hat[0]),rh
o_hat[1]))/ (a1_new*dot(cross(rho_hat[-1],rho_hat[0]),rho_hat[1]))
    rho_mag2 = ((a1_new*dot(cross(rho_hat[-1],R[-1]),rho_hat[1]))-
dot(cross(rho_hat[-1],R[0]),rho_hat[1])+a3_new*dot(cross(rho_hat[-
1],R[1]),rho_hat[1])) / (-dot(cross(rho_hat[-1],rho_hat[0]),rho_hat[1]))
    rho_mag3 = ((a1_new*dot(cross(rho_hat[0],R[-1]),rho_hat[-1]))-
dot(cross(rho_hat[0],R[0]),rho_hat[-
1])+a3_new*dot(cross(rho_hat[0],R[1]),rho_hat[-1])) /
(a3_new*dot(cross(rho_hat[0],rho_hat[1]),rho_hat[-1]))
```

```
    rho_mag = array([[rho_mag2],[rho_mag3],[rho_mag1]])

    # rho vector
    rho = rho_hat * rho_mag

    # r vector
    r = rho - R

    # r vector dot
    r_dot1 = (r[1] - f[1]*r[0]) / g[1]
    r_dot2 = (r[-1] - f[-1]*r[0]) / g[-1]
    r_dot = (r_dot1 + r_dot2) / 2

    # Loop end condition
    if abs(a1-a1_new)<1e-12 and abs(a3-a3_new)<1e-12:
       print "loop broken at iteration ", i, '\n'
       break

    a1 = a1_new
    a3 = a3_new

    if iteration:
       print "###################################"
       print "         ITERATION",i
       print "###################################"
       print "f: ", f
       print "g: ", g, '\n'
       print "a1, a3: ",a1_new,a3_new,'\n'
       print "r vector: ",r
       print "r dot vector: ", r_dot,'\n'

print "############"
print "Final Values"
print "############"
print "f: ", f
print "g: ", g, '\n'
print "equatorial r vector: ", r
print "equatorial r dot vector: ", r_dot,'\n'

##################################################
##################################################

r = vector(r[0])
# Rotate r vector to ecliptic coordinates
r = vector(r.x, r.y*cos(epsilon)+r.z*sin(epsilon), -
r.y*sin(epsilon)+r.z*cos(epsilon))
print "ecliptic r vector: ", r

r_dot = vector(r_dot)
# Rotate r vector dot to ecliptic coordinates
r_dot = vector(r_dot.x, r_dot.y*cos(epsilon)+r_dot.z*sin(epsilon), -
r_dot.y*sin(epsilon)+r_dot.z*cos(epsilon))
print "ecliptic r dot vector: ", r_dot, '\n'

##################################################
##################################################

# Calculate Orbital Elements

# Semi-major axis (a)
a = 1/(2/mag(r) - dot(r_dot,r_dot))
```

```
print "Semi-major axis (a): ", a

# Eccentricity (e)
h = cross(r, r_dot) #angular momentum per unit mass
e = sqrt(1 - mag(h)**2/(mu*a))
print "Eccentricity (e): ", e

# Inclination (i)
i = atan(sqrt((h.x**2 + h.y**2))/h.z)
print "Inclination (i): ", degrees(i)

# Longitude of the ascending node (O)
O1 = asin(h.x/(mag(h)*sin(i)))
O2 = pi - O1
O3 = acos(-h.y/(mag(h)*sin(i)))
O4 = 2*pi - O3
O = ambiguity_check(O1%(2*pi),O2%(2*pi),O3%(2*pi),O4%(2*pi))
print "Longitude of the ascending node (O)", degrees(O)

# True anomaly (v)
v1 = acos((a-a*(e**2)-mag(r))/(e*mag(r)))
v2 = 2*pi - v1
v3 = asin((a*(1-e**2)*(dot(r,r_dot))) / (e*mag(h)*mag(r)) )
v4 = pi - v3
v = ambiguity_check(v1%(2*pi),v2%(2*pi),v3%(2*pi),v4%(2*pi))

# Argument of perihelion (w)
U1 = acos((r.x*cos(O)+r.y*sin(O))/mag(r))
U2 = 2*pi - U1
U3 = asin(r.z/(mag(r)*sin(i)))
U4 = pi - U3
U = ambiguity_check(U1%(2*pi),U2%(2*pi),U3%(2*pi),U4%(2*pi))

w = (U - v)%(2*pi)
print "Argument of Perihelion (w): ", degrees(w)

# Eccentric anomaly (E)
E = acos((a-mag(r))/(a*e))
if v>=pi:
    E = 2*pi - E

# Mean anomaly (M)
M = E - e*sin(E)
print "Mean anomaly (M): ", degrees(M)
```

Appendix 3. Laplacian Method VPython Codes

Code: LaplacianOrbitDetermination.py

```
# Laplacian Orbit Determination
# Written August 5, 2012

from math import *
from numpy import *
from visual import *

debug = True
iteration = False

# Import data file
if debug == True:
    filename = "JPL2.txt"
else:
    filename = raw_input("Enter name of data file:")

###################################################
###################################################

# Define functions

# Convert (RA) hours, minutes, and seconds to radians
def hms2Rad(hms):
    if hms[0]>=0: #positive RA
        return radians((hms[0] + hms[1]/60. + hms[2]/3600.)*360./24.)
    else: #negative RA
        return radians((hms[0] + hms[1]/60. + hms[2]/3600.)*360./24.)

# Convert (Dec) degrees, arcminutes, and arcseconds to radians
def dms2Rad(dms):
    if dms[0]>=0: #positive Dec
        return radians(dms[0] + dms[1]/60. + dms[2]/3600.)
    else: #negative Dec
        return radians(dms[0] - dms[1]/60. - dms[2]/3600.)

# Convert UT to JD
def UT2JD(dmyhms): #day,month,year,hour,minute,second
    decimalTime = float(dmyhms[3]+(dmyhms[4]/60.)+(dmyhms[5]/3600.))
    Jnaught = 367*dmyhms[2] - int(7*(dmyhms[2]+int((dmyhms[1]+9)/12))/4)
    Jnaught += int(275*dmyhms[1]/9) + dmyhms[0] + 1721013.5
    return Jnaught + (decimalTime/24.)

# Check ambiguity by comparing values obtained through sine and cosine
def ambiguity_check(A,B,C,D):
    if round(A*1e6)==round(C*1e6):
        return A
    elif round(B*1e6)==round(C*1e6):
        return B
    elif round(A*1e6)==round(D*1e6):
        return A
    elif round(B*1e6)==round(D*1e6):
        return B

# Calculate rho unit vector
def rho_hat(ra,dec):
    rho_hat = vector(cos(ra)*cos(dec),sin(ra)*cos(dec),sin(dec))
    return rho_hat
```

```
#################################################
#################################################

# Constants
mu = 1
k = 0.01720209895 #Boltzmann constant
c = 173.1446 #speed of light (AU/day)
eps = radians(23.4376600557) #ecliptic tilt (epsilon)

#################################################
#################################################

# Extract data from file
data = loadtxt(filename,delimiter=',')

t1 = UT2JD(data[0])
ra1 = hms2Rad(data[1][0:3])
dec1 = dms2Rad(data[1][3:])
R1 = vector(data[2][0:3])
Rdot1 = vector(data[2][3:])

t2 = UT2JD(data[3])
ra2 = hms2Rad(data[4][0:3])
dec2 = dms2Rad(data[4][3:])
R2 = vector(data[5][0:3])
Rdot2 = vector(data[5][3:])

t3 = UT2JD(data[6])
ra3 = hms2Rad(data[7][0:3])
dec3 = dms2Rad(data[7][3:])
R3 = vector(data[8][0:3])
Rdot3 = vector(data[8][3:])

# Define Earth-Sun vectors

R = R2 #R[-1] and R[1] are not used in the Laplacian method
Rdot = Rdot2 #Rdot[-1] and Rdot[1] are not used in the Laplacian method

#################################################
#################################################

# rho unit vector
rho_hat1 = rho_hat(ra1,dec1)
rho_hat2 = rho_hat(ra2,dec2)
rho_hat3 = rho_hat(ra3,dec3)
rho_hat = array([rho_hat2,rho_hat3,rho_hat1])

# tau
tau = k * array([t3-t1,t3-t2,t1-t2])

# denominator for rho hat dot and rho hat double dot
denom = tau[-1]*tau[1]*tau[0]

# rho hat dot
rho_hat_dot = (tau[1]**2.*(rho_hat[-1]-rho_hat[0]) - tau[-
1]**2.*(rho_hat[1]-rho_hat[0]))/denom

# rho hat dot dot
rho_hat_dot_dot = -2.*(tau[1]*(rho_hat[-1]-rho_hat[0]) - tau[-
1]*(rho_hat[1]-rho_hat[0]))/denom
```

```
# initial guess: r magnitude
r_mag = 2.5

print "###############"
print "Initial Values"
print "###############"
print "rho unit vector: ", rho_hat[0]
print "rho hat dot vector: ", rho_hat_dot
print "rho hat dot dot vector: ", rho_hat_dot_dot

for i in range (0,1000):

    # rho magnitude
    A = dot(cross(rho_hat[0],rho_hat_dot),R)/dot(cross(rho_hat[0],
    rho_hat_dot),rho_hat_dot_dot)
    B = (1. + 1/328900.5) / (mag(R)**3.) * A
    rho_mag = ( A / r_mag**3. ) - B

    # rho vector
    rho = rho_mag * rho_hat[0]

    # r magnitude
    r_mag_new = sqrt( rho_mag**2. + mag(R)**2. - 2.*dot(R,rho) )

    # Loop end condition
    if abs(r_mag - r_mag_new)<1e-12:
       print "loop broken at iteration ", i, '\n'
       break

    r_mag = r_mag_new

    if iteration:
       print "##################################"
       print "       ITERATION",i
       print "##################################"
       print "rho magnitude: ", rho_mag
       print "r magnitude: ", r_mag,'\n'

# rho dot magnitude
A = dot(cross(rho_hat[0],rho_hat_dot_dot),R)/dot(cross(rho_hat[0],
rho_hat_dot),rho_hat_dot_dot)
B = (1. + 1./328900.5) / (mag(R)**3.) * A
rho_mag_dot = (-1./2.)*( A/(r_mag**3.) - B )

# r vector
r = rho - R

# r vector dot
rdot = rho_mag_dot*rho_hat[0] + rho_mag*rho_hat_dot - Rdot/k

print "############"
print "Final Values"
print "############"
print "equatorial r vector: ", r
print "equatorial r dot vector: ", rdot,'\n'

##################################################
##################################################

r = vector(r)
# Rotate r vector to ecliptic coordinates
```

```
r = vector(r.x, r.y*cos(eps)+r.z*sin(eps), -r.y*sin(eps)+r.z*cos(eps))
print "ecliptic r vector: ", r

rdot = vector(rdot)
# Rotate r vector dot to ecliptic coordinates
rdot = vector(rdot.x, rdot.y*cos(eps)+rdot.z*sin(eps), -
rdot.y*sin(eps)+rdot.z*cos(eps))
print "ecliptic r dot vector: ", rdot, '\n'

####################################################
####################################################

# Calculate Orbital Elements

# Semi-major axis (a)
a = 1./(2./mag(r) - dot(rdot,rdot))
print "Semi-major axis (a): ", a

# Eccentricity (e)
h = cross(r, rdot) #angular momentum per unit mass
e = sqrt(1. - mag(h)**2./(mu*a))
print "Eccentricity (e): ", e

# Inclination (i)
i = atan(sqrt((h.x**2. + h.y**2.))/h.z)
print "Inclination (i): ", degrees(i)

# Longitude of the ascending node (O)
O1 = asin(h.x/(mag(h)*sin(i)))
O2 = pi - O1
O3 = acos(-h.y/(mag(h)*sin(i)))
O4 = 2.*pi - O3
O = ambiguity_check(O1%(2*pi),O2%(2*pi),O3%(2*pi),O4%(2*pi))
print "Longitude of the ascending node (O)", degrees(O)

# True anomaly (v)
v1 = acos((a-a*(e**2)-mag(r))/(e*mag(r)))
v2 = 2.*pi - v1
v3 = asin((a*(1-e**2)*(dot(r,rdot))) / (e*mag(h)*mag(r)) )
v4 = pi - v3
v = ambiguity_check(v1%(2*pi),v2%(2*pi),v3%(2*pi),v4%(2*pi))

# Argument of perihelion (w)
U1 = acos((r.x*cos(O)+r.y*sin(O))/mag(r))
U2 = 2*pi - U1
U3 = asin(r.z/(mag(r)*sin(i)))
U4 = pi - U3
U = ambiguity_check(U1%(2*pi),U2%(2*pi),U3%(2*pi),U4%(2*pi))

w = (U - v)%(2*pi)
print "Argument of Perihelion (w): ", degrees(w)

# Eccentric anomaly (E)
E = acos((a-mag(r))/(a*e))
if v>=pi:
    E = 2*pi - E

# Mean anomaly (M)
M = E - e*sin(E)
print "Mean anomaly (M): ", degrees(M)
```

Appendix 4. Test file: input and output

Input (input.txt)

05, 07, 2012, 12, 00, 00

17, 59, 50.65, -17, 04, 33.3

-2.405579733688322E-01, 9.063044720766212E-01, 3.929017895577459E-01

-1.642978546997832E-02, -3.672259697586189E-03, -1.591588140080336E-03

15, 07, 2012, 12, 00, 00

17, 47, 49.47, -17, 13, 01.4

-4.007751183445531E-01, 8.570377658029277E-01, 3.715410404186910E-01

-1.553709953940518E-02, -6.162876952047848E-03, -2.672347974923526E-03

25, 07, 2012, 12, 00, 00

17, 37, 51.11, -17, 25, 42.7

-5.497531195215302E-01, 7.835851943193904E-01, 3.396968143598786E-01

-1.418392341277206E-02, -8.490691283011583E-03, -3.680289867519964E-03

Output

Table. Orbital Elements of Test File

Orbital Element	JPL Horizons	Computed with the Gaussian*	Computed with the Laplacian*
Semi-major axis (a)	2.195246692884144AU	2.1798206AU	2.2370177AU
Eccentricity (e)	0.4543080457422227	0.44728559	0.44816245
Inclination (i)	3.854140588204837°	3.8460258°	3.8592481°
Longitude of the ascending node (Ω)	173.2888663178230°	173.50081°	173.68017°
Argument of the perihelion (w)	231.4192149530281°	231.77688°	229.30631°
Mean anomaly (M)	283.7976363246500°	282.26277°	285.31816°

* Confident to 8 significant figures because of RA and Dec

Appendix 5. Data Comparison

Table 1. Percent Errors of Orbital Elements of Observations 2, 3, and 4

Orbital Elements	Percent Error % ($\pm 1\times10^{-8}$)	
	Gaussian Method	Laplacian Method
Semi-major axis (a)	15.104342	11.878890
Eccentricity (e)	9.2525567	4.7716949
Inclination (i)	4.6377953	0.16903366
Longitude of the ascending node (Ω)	0.74617701	1.3882061
Argument of the perihelion (w)	7.8926755	4.3814252
Mean anomaly (M)	9.9148955	6.3750869

Table 2. Percent Errors of Orbital Elements of Observations 3, 4, and 5

Orbital Elements	Percent Error % ($\pm 1\times10^{-8}$)	
	Gaussian Method	Laplacian Method
Semi-major axis (a)	1.5782657	12.836234
Eccentricity (e)	1.3560714	4.5317739
Inclination (i)	1.7534327	0.82281717
Longitude of the ascending node (Ω)	0.44012611	1.1487718
Argument of the perihelion (w)	2.2306000	5.4966738
Mean anomaly (M)	1.9831094	7.0585969

Table 3. Percent Errors of Orbital Elements of Observations 1, 2, and 4

Orbital Elements	Percent Error % ($\pm 1\times10^{-8}$)	
	Gaussian Method	Laplacian Method
Semi-major axis (a)	1.3472246	93.050714
Eccentricity (e)	0.74685286	58.698204
Inclination (i)	0.18046011	13.802467
Longitude of the ascending node (Ω)	0.407100627	8.6586370
Argument of the perihelion (w)	0.71648700	3.1823321
Mean anomaly (M)	1.3624637	19.750335

Table 4. Percent Errors of Orbital Elements of Observations 1, 2, and 3

Orbital Elements	Percent Error % ($\pm 1\times10^{-8}$)	
	Gaussian Method	Laplacian Method
Semi-major axis (a)	2.1509338	160.23162
Eccentricity (e)	0.84846868	76.844105
Inclination (i)	1.0273976	18.050050
Longitude of the ascending node (Ω)	0.68867615	10.471505
Argument of the perihelion (w)	1.7453097	2.214267
Mean anomaly (M)	2.5735405	22.843811

Table 5. Percent Errors of Orbital Elements of Observations 1, 3, and 5

Orbital Elements	Percent Error % ($\pm 1\times10^{-8}$)	
	Gaussian Method	Laplacian Method
Semi-major axis (a)	0.019336164	224.17408
Eccentricity (e)	0.77040452	82.799103
Inclination (i)	0.48102343	15.127051
Longitude of the ascending node (Ω)	0.16731322	9.1737591
Argument of the perihelion (w)	0.39115698	5.7056734
Mean anomaly (M)	0.14892134	23.812610

Table 6. Percent Errors of Orbital Elements of Observations 1, 3, and 4

Orbital Elements	Percent Error % ($\pm 1\times10^{-8}$)	
	Gaussian Method	Laplacian Method
Semi-major axis (a)	0.11402970	579.6710
Eccentricity (e)	1.6652274	103.17216
Inclination (i)	0.81099004	19.695824
Longitude of the ascending node (Ω)	0.20602177	11.001659
Argument of the perihelion (w)	0.64915715	4.5140394
Mean anomaly (M)	0.22868244	26.011463

Works Cited

1. "Diagram illustrating and explaining various terms in relation to Orbits of Celestial bodies."*Wikipedia.* 10 October 2007. <http://upload.wikimedia.org/wikipedia/commons/e/eb/Orbit1.svg>
2. Bradley, James. *An account of the new discovered motion of the fixed stars.* Philosophical Transactions of the Royal Society of London, 1729.

Bibliography

Bate, Roger R., Donald D. Mueller, and Jerry E. White. *Fundamentals of Astrodynamics.* New York: Dover Publications, 1971. Print.

Danby, J. M. A. *Fundamentals of Celestial Mechanics.* New York: Macmillan, 1962. Print.

Gronchi, Giovanni F. *Classical and Modern Orbit Determination for Asteroids.* Thesis. Department of Mathematics, University of Pisa, 2004.

Gronchi, Giovanni F. *Multiple Solutions in Preliminary Orbit Determination from Three Observations.* Thesis. Department of Mathematics, University of Pisa.

PURDUE
UNIVERSITY
CALUMET
Department of Chemistry and Physics

Adam Rengstorf
2200 169th St, Gyte 268
Hammond, IN 46383
T 219-989-2624
F 219-989-2130
adamwr@purduecal.edu

October 26, 2012

To whom it may concern,

I am writing this cover letter on behalf of is currently working on his IB Diploma Extended Essay, *An Investigation into the Gaussian and the Laplacian Methods for the Determination of a Near-Earth Asteroid's Orbit Using Three Observations of its Position*. I had the opportunity to meet and work with during an intensive astronomy summer program this past summer.

I am an Associate Professor of Physics and Astronomy in the department of Chemistry and Physics at Purdue University Calumet (PUC) in Hammond, IN. I have been teaching physics and astronomy courses at PUC since 2005 and have been tenured since 2010. My undergraduate background is in physics (B.S., 1996, Binghamton University) and I have master's and doctoral degrees in astronomy (M.A., 1999, Ph.D., 2003, Indiana University). More germanely, during Summer 2012, I served as the Associate Academic Director for the Summer Science Program (SSP). I and one other teaching faculty were responsible for preparing and delivering nearly 120 hours' worth of university-level lectures on calculus, physics, astronomy, and computer programming over a 5-1/2 week program. Atop the lecture material, the students worked in teams to make nightly observations of a near-Earth asteroid. The goal of the program was to successfully image an asteroid on multiple nights and to write a computer code that utilized the orbital mechanics taught during the day to fully describe the asteroid's orbit about the Sun.

Between myself and one other teaching faculty at SSP, we oversaw all aspects of this project: lecturing, overseeing the Teaching Assistants (TAs) who were on hand at the telescope every night, providing assistance and guidance during the data reduction and image analysis stage, and helping, again along with our TAs, to trouble-shoot and debug students' computer codes. This was a very intensive program and, while we expected a good deal of self-motivation and even self-teaching at times, we were on hand on a daily, close to continual, basis for support and guidance.

I have read a draft version of ' essay and find it to be an accurate representation of the work he did during our time together. If I can be of any further assistance, please do not hesitate to get in touch.

Sincerely yours,

Adam Rengstorf

Associate Professor of Physics & Astronomy, Purdue University Calumet

Associate Academic Director, Summer Science Program