The Pennsylvania State University
The Graduate School
College of Engineering

LONG-TERM ORBIT PROPAGATION USING SYMPLECTIC INTEGRATION ALGORITHMS

A Thesis in
Aerospace Engineering
by
Koundinya Kuppa

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2016

The thesis of Koundinya Kuppa was reviewed and approved* by the following:

David B. Spencer
Professor of Aerospace Engineering
Thesis Advisor

Lyle N. Long
Distinguished Professor of Aerospace Engineering, Computational Science, and Mathematics
Director of Computational Science Graduate Minor

George A. Lesieutre
Professor of Aerospace Engineering
Head of Department of Aerospace Engineering

*Signatures are on file in the Graduate School.

Abstract

Understanding the evolution of satellite orbits in the long-term is of great importance in astrodynamics. In order to achieve this, accurate propagation of the orbital dynamics of the satellite is required. This thesis attempts to implement and evaluate a class of numerical integration methods known as symplectic algorithms. This class of algorithms is highly regarded in scientific applications, especially in long-term studies. The objective of this thesis is to demonstrate the superior accuracy and efficient speed of several algorithms of this class and obtain long-term state of satellites under the several influencing forces. Within each application, several cases with different values for parameters such as the time step and duration are executed. In addition, long-term orbital evolution of a satellite in various orbital regimes is conducted. The results indicate that the symplectic algorithms are more accurate for orbit propagation at various time increments tested. In addition, the symplectic algorithms are more computationally efficient in all but a few cases.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\ddot{\mathbf{r}}$	Acceleration vector of a Satellite relative to the Earth (km/s^2)
μ	Gravitational parameter of the Earth (km^3/s^2)
$\mathbf{r}$,r	Vector/Distance from the Earth to the Satellite (km)
G	Universal gravitational constant ($km^3/kg\ s^2$)
m_1	Mass of the Planet (kg)
m_2	Mass of the Satellite (kg)
θ	True anomaly (rad)
ε	Specific orbital energy of the Satellite (km^2/s^2)
$\mathbf{v}$,v	Orbital Velocity vector/magnitude of the Satellite (km/s)
a	Semi-major axis of the Satellite orbit (km)
e	Eccentricity of the Satellite orbit
p	Semi-latus rectum of the Satellite orbit (km)
h	Specific angular momentum (km^2/s)
$\mathbf{a}_{GR}$	Acceleration vector due to the gravitational force of the Earth (km/s^2)
$\mathbf{a}_{3^{rd}}$	Total acceleration vector from third-body gravitational forces (km/s^2)
$\mathbf{a}_{geo}$	Acceleration vector from geopotential forces (km/s^2)
$\mathbf{a}_{SRP}$	Acceleration vector from Solar Radiation Pressure (km/s^2)

$\mathbf{a}_D$	Perturbing acceleration vector due to atmospheric drag (km/s^2)
μ_j	The gravitational parameter of a third-body (km^3/s^2)
$\mathbf{d}_j$,d_j	Vector/Distance from a Satellite to a third body (km/s)
$\boldsymbol{\rho}_j$,ρ_j	Vector/Distance from the Earth to a third body (km/s^2)

Acknowledgments

I would like to sincerely thank my thesis advisor, Dr. David Spencer. All throughout my Master's degree pursuit at Penn State, he has been a constant source of encouragement and guidance. He always pointed me in the right direction and gave me advice and taught me a lot. I really enjoyed talking with him during our weekly meetings and I always learned something from them. He was also instrumental in getting me job opportunities and I cannot thank him enough for what he has done for my professional career as an aerospace engineer. I especially would like to thank him for his constant critical reviews of my thesis drafts in my final semester and for encouraging me to stay on schedule. I will always appreciate the time he took to talk to me, even outside out weekly scheduled meetings. I have learned so much about astrodynamics and other topics in aerospace engineering by taking his courses and having conversations with him. Thank you, Dr. Spencer.

I would also like to thank Dr. Long and Dr. Lesieutre for taking the time to read and critique my thesis.

I would like to also thank my friends. In the two years I have been at this university, I have grown so much as an individual. I truly believe that this was shaped by the friendships that I have made here. I always appreciated our conversations and truly enjoyed the time we shared together. You guys have always

had advice and motivation when I needed it and I really appreciate that. You have made my time as a student at Penn State truly memorable.

Finally, I would like to thank my parents who have believed in me since childhood to be destined to do great things. Without their moral and financial support, I don't believe I could have achieved anything in my life. I owe it all to their hard work over the years in helping me achieve my goals. Whenever I would feel the pressures of schoolwork or research weigh me down, I always was able to turn to them to lift me up. In retrospect, I can recall countless times where they went above and beyond to secure me a good and a successful life. With that realization, I sincerely hope that my academic and professional achievements justify these efforts. Thank you Mom and Dad.

Chapter 1
Introduction and Background

In many fields of science and engineering, analytical solutions to dynamical systems are generally not possible without making one or more assumptions to simplify the model. This is true in astrodynamics with the only exception of the two-body problem. While it is useful in many ways, it assumes only one force affecting the motion of an object: the gravitational force of the central body. While this may not be an erroneous assumption in the short-term, it is not valid for long-term orbital dynamics studies. Therefore, we must account for other perturbing forces such as the gravity of other celestial bodies, geopotential, atmospheric drag, solar radiation pressure among others. However, the governing equations describing the motion of a satellite subject to the aforementioned forces can only be solved through numerical integration, as no analytic solution can be determined. With any integration algorithm, there are numerical errors that are arise due to the finite difference update scheme used. The work presented in this thesis aims to apply several symplectic numerical integration algorithms that have been widely regarded for their long-term accuracy to determine the orbital evolution of a satellite in various regimes. The term "symplectic" is derived form the latin root "symp-" which is equivalent to the word complex. Evaluating the accuracy and effectiveness of the chosen algorithms is a primary objective of this work. Another objective is

to study the long-term orbital behavior of a satellite in various orbits in the vicinity the Earth.

Previously, long-term orbital study in references [1] and [2] explored averaged equations for the perturbing accelerations. In reference [4], Kaufman used a semi-analytic theory in which equations are developed for Earth and lunar orbiters subject to third-body perturbations. While these averaged formulations are sufficient valid in the short-term, numerical integration of the dynamics is warranted for long-term studies. Previous implementations of symplectic algorithms include reference [3], where the leapfrog algorithm was used to simulate n-body problems. A comprehensive test of various algorithms was performed for an unperturbed two-body problem while focusing on the speed and accuracy of these algorithms in reference [5]. The work presented in this thesis implements previously explored algorithms such as Verlet and some Runge-Kutta methods but also includes two other symplectic algorithms that have yet to be applied to orbit propagation studies: Forest-Ruth and PEFRL. This study also attempts to directly integrate the equations of motion with various perturbations (e.g. third-body and geopotential) to evaluate the long-term orbital behavior of satellites. In addition, the efficient implementation of the numerical integration algorithms is of importance in this thesis. The orbits are propagated for one, five, and ten year time spans and each case was performed thrice with 10, 100, and 1000 second time increments.

In addition to long-term orbit modeling, the work presented in this thesis is also important to other areas in astrodynamics such as space tracking and predicting the orbits of space debris. In space tracking, accurate determination of the state of an orbiting object is limited by the quality of the tracking data as well as errors in numerical integration of the dynamics. The algorithms used in this thesis can alleviate the latter of these issues. In the analysis of space-debris dynamics, these

algorithms can also be useful in prediction the long-term orbits of a debris cloud resulting from a collision.

The structure of this thesis is as follows. Chapter 2 outlines the dynamics used in this thesis. Chapter 3 introduces the several numerical integration algorithms of interest. Chapter 4 is a summary, observations and discussion of the results, and finally Chapter 5 ends with a concluding discussion of the work presented in the thesis and suggesting ideas for future work in this area.

Chapter 2
Dynamics

2.1 Simple Two-Body Problem

Consider two objects in motion around their common center of mass (barycenter) affected by the mutual gravitational force. This is known as the two-body problem. Usually, the absolute motion of both bodies is not of interest in the field of astrodynamics but rather the motion of one object relative to the other. Equation (2.1) describes that motion, where $\mathbf{r}$ is the relative position vector of the satellite to the primary body, and $\ddot{\mathbf{r}}$ is the acceleration,

$$\ddot{\mathbf{r}} + \frac{\mu}{r^3}\mathbf{r} = 0 \tag{2.1}$$

where

$$\mu = G(m_1 + m_2) \tag{2.2}$$

which is known as the Gravitational Parameter. When $m_2 << m_1$,

$$\mu \approx Gm_1 \tag{2.3}$$

In astrodynamics, we rarely consider the mass of the satellite (m_2) to be significant (compared to the mass of the central body (m_1)) and hence equation (2.3) is more appropriate for the purposes of modeling satellite motion around

a celestial object and will be used to formulate the gravitational parameter. As described earlier, the two-body problem has an analytical solution known as the orbit equation which is given in equation (2.4). This equation relates θ (which is the angle between the periapse point and P_1) to the distance from the occupied focus F to the point P_1 (r). The graphical representation of this is given in Figure 2.1 in which the central body is located at F.

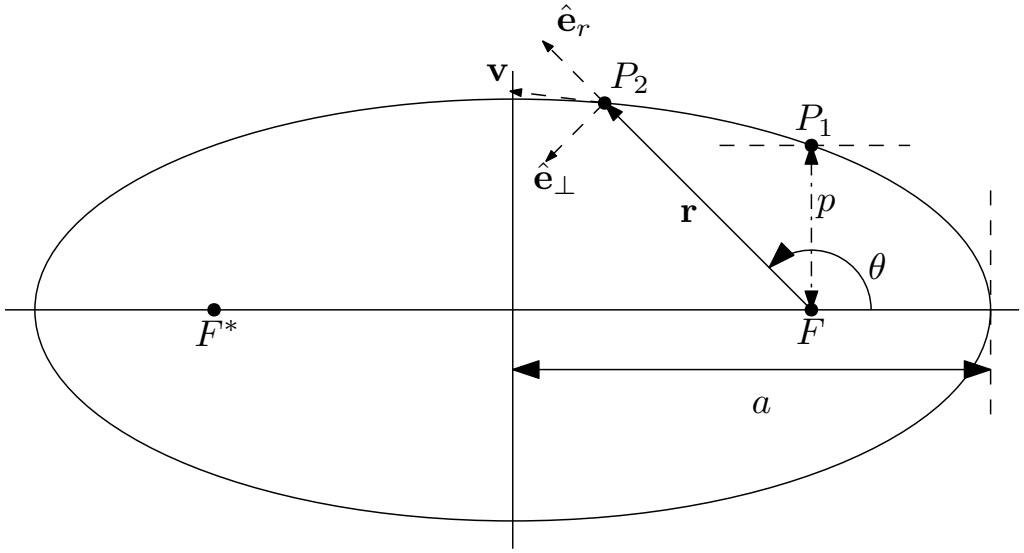

Figure 2.1. Orbital Geometry

$$r = \frac{a(1-e^2)}{1+e\cos\theta} \tag{2.4}$$

Since gravity is a conservative force, the energy of a satellite orbit does not change over time. This implies that integration of equation (2.1) should conserve the total energy of the system. With numerical integration of the differential equation (described in equation (2.1)), this energy (specific energy) can be used as a measure of the numerical accuracy for the orbit. The vis-viva equation defines the specific orbital energy in terms of the position (r) and velocity (v) at a given

time (or equivalently at a given θ in its orbit),

$$\varepsilon = \frac{v^2}{2} - \frac{\mu}{r} = -\frac{\mu}{2a} \tag{2.5}$$

Therefore, the accuracy of a numerical integration algorithm can be determined by monitoring deviations of ε from its initial (reference) value.

While the energy of an orbit is an invariant, there is no physical intuition of this quantity. Therefore, it could be considered as an indirect measure of the integration error. A geometric quantity would be better suited to present a physical measure of the accuracy. The first of these quantities in the semi-major axis of an orbit which is shown in Figure 2.1 and it can defined in terms of ε in equation (2.6) by rearranging terms in equation (2.5).

$$a = -\frac{\mu}{2\varepsilon} \tag{2.6}$$

The eccentricity, which is a measure of the deviation of an orbit from a circle, is also an invariant (for a given orbit) which implies it can be useful in assessing the accuracy of a orbit propagation scheme. The eccentricity of an orbit can be expressed in terms of p and a (shown in Figure 2.1),

$$e = \sqrt{1 - \frac{p}{a}} \tag{2.7}$$

where $p = \frac{h^2}{\mu}$ is the semi-latus rectum of the orbit. The specific angular momentum, h, of the orbit can be expressed in terms of the state vector,

$$h = |\mathbf{r} \times \mathbf{v}| \tag{2.8}$$

Both the semi-major axis and the eccentricity are more geometrically intuitive

ways to measure the accuracy of an integration algorithm. Despite this, all three measures (energy, semi-major axis, and eccentricity), are used here to display the accuracy of the various algorithms used. An example of the use of the semi-major axis as a measure of error is given in reference [6]. The use of orbital energy as a measure of accuracyis seen in reference [7]. The following section of this chapter outlines the formulation some of the other perturbing forces that affect the orbital motion of a satellite.

2.2 Perturbed Two-Body Problem

Although the model present in previous section can be very useful for problems astrodynamics, any long-term orbital study must account for other sources of gravitational and non-gravitational forces that can influence the motion of a satellite. These can include third-body effects (gravitational influences from the sun, moon, and other planets), geopotential perturbations, atmospheric drag, solar radiation pressure among others. A general formulation for modeling orbital perturbations is shown in reference [10] as

$$\ddot{\mathbf{r}} = \mathbf{a}_{GR} + +\mathbf{a}_{3^{rd}} + \mathbf{a}_{geo} + \mathbf{a}_{SRP} + \mathbf{a}_{D} \tag{2.9}$$

where $\mathbf{a}_{GR}$ was previously defined as the gravity of the primary body:

$$\mathbf{a}_{GR} = -\frac{\mu}{r^3}\mathbf{r}$$

Some of the remaining terms of equation (2.9) are discussed in the subsequent sections.

2.2.1 Third-Body Effects

A formulation for third body effects ($\mathbf{a}_{3rd}$) in the following derivation through the computation of a disturbing acceleration (p. 387-389 of [8]). This can be useful in modeling the perturbing forces of additional bodies aside from the primary. For a satellite orbiting the Earth perturbations from the Sun, moon, and other planets can be modeled using the formulation. A physical depiction of this formulation is given in Figure 2.2 where P_1 is the primary body (Earth) around which P_2 (a satellite) orbits and P_j is a third body (Sun, Moon etc.) that influences the motion of P_2. The unit vectors $\hat{I}, \hat{J}, \hat{K}$ denote the orientation of the Cartesian coordinate system.

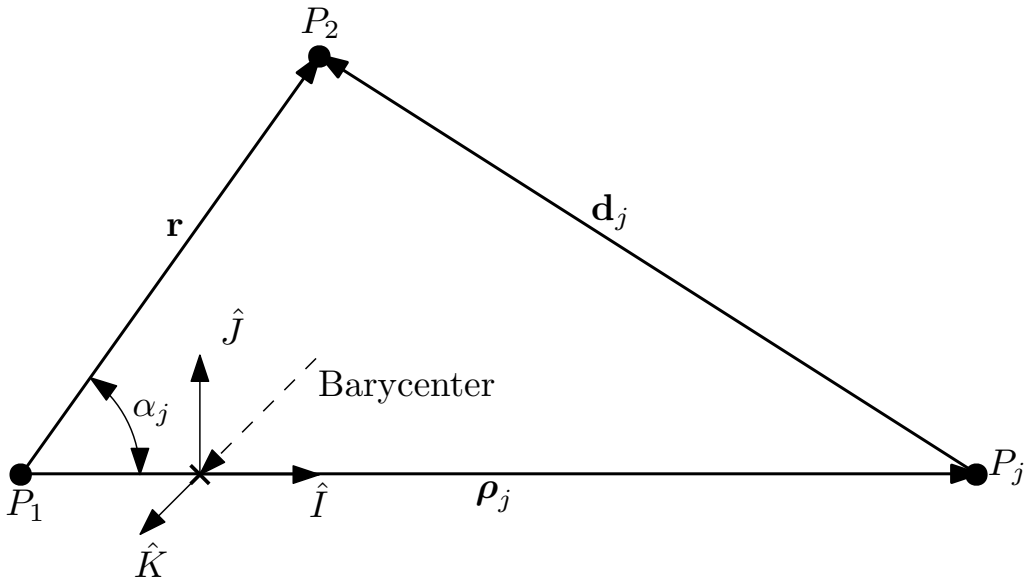

Figure 2.2. Three-Body Problem Geometry

$$\mathbf{a}_{3rd} = -\sum_{j=3}^{N} \mu_j \left(\frac{1}{d_j^3}\mathbf{d}_j + \frac{1}{\rho_j^3}\boldsymbol{\rho_j} \right) \tag{2.10}$$

where

$$\boldsymbol{\rho}_j = \mathbf{r_j} - \mathbf{r} \text{ and } \mathbf{d_j} = \mathbf{r} - \boldsymbol{\rho}_j \tag{2.11}$$

and drop the indices for convenience

$$\begin{aligned}\frac{1}{d^3}\mathbf{d} + \frac{1}{\rho^3}\boldsymbol{\rho} &= \frac{1}{d^3}(\mathbf{r} - \boldsymbol{\rho}) + \frac{1}{\rho^3}\boldsymbol{\rho} \\ &= \frac{1}{d^3}\mathbf{r} - \frac{1}{d^3}\boldsymbol{\rho} + \frac{1}{\rho^3}\boldsymbol{\rho} \\ &= \frac{1}{d^3}\left[\mathbf{r} + \left(\frac{d^3}{\rho^3} - 1\right)\boldsymbol{\rho}\right] \end{aligned} \tag{2.12}$$

The formulation in either equation (2.10) or (2.12) requires the computation of $\mathbf{d}_j$ and $\boldsymbol{\rho}_j$. It would be convenient to have the perturbing acceleration only as a function of $\boldsymbol{\rho}_j$ and ρ_j. Equations (2.13a)-(2.13f) shows the derivation of that formulation of the perturbing acceleration.

$$\left(\frac{d^3}{\rho^3} - 1\right) = \sqrt{\frac{\mathbf{d}\cdot\mathbf{d}}{\boldsymbol{\rho}\cdot\boldsymbol{\rho}}}\left(\frac{\mathbf{d}\cdot\mathbf{d}}{\boldsymbol{\rho}\cdot\boldsymbol{\rho}}\right) - 1 \tag{2.13a}$$

$$= \left(\frac{\mathbf{d}\cdot\mathbf{d}}{\boldsymbol{\rho}\cdot\boldsymbol{\rho}}\right)^{\frac{3}{2}} - 1 \tag{2.13b}$$

Then, using definitions in (2.11),

$$= \left(\frac{(\mathbf{r} - \boldsymbol{\rho})\cdot(\mathbf{r} - \boldsymbol{\rho})}{\boldsymbol{\rho}\cdot\boldsymbol{\rho}}\right)^{\frac{3}{2}} - 1 \tag{2.13c}$$

$$= \left(\frac{\mathbf{r}\cdot\mathbf{r} - 2\boldsymbol{\rho}\cdot\mathbf{r} + \boldsymbol{\rho}\cdot\boldsymbol{\rho}}{\boldsymbol{\rho}\cdot\boldsymbol{\rho}}\right)^{\frac{3}{2}} - 1 \tag{2.13d}$$

$$= \left(\frac{\mathbf{r}\cdot(\mathbf{r} - 2\boldsymbol{\rho}) + \boldsymbol{\rho}\cdot\boldsymbol{\rho}}{\boldsymbol{\rho}\cdot\boldsymbol{\rho}}\right)^{\frac{3}{2}} - 1 \tag{2.13e}$$

$$= \left(\frac{\mathbf{r}\cdot(\mathbf{r} - 2\boldsymbol{\rho})}{\boldsymbol{\rho}\cdot\boldsymbol{\rho}} + 1\right)^{\frac{3}{2}} - 1 \tag{2.13f}$$

Therefore, with the following definition for q,

$$q = \frac{\mathbf{r} \cdot (\mathbf{r} - 2\boldsymbol{\rho})}{\boldsymbol{\rho} \cdot \boldsymbol{\rho}} \tag{2.14}$$

the term in equation (2.13f) can be expressed

$$\left(\frac{d^3}{\rho^3} - 1\right) = (1+q)^{\frac{3}{2}} - 1 = f(q) \tag{2.15}$$

While $f(q)$ can be expanded in a series, the following closed form solution also exists

$$f(q) = q\frac{3 + 3q + q^2}{(1+q)^{\frac{3}{2}} + 1} \tag{2.16}$$

Hence, d_j is no longer requried to calculate the acceleration. To conclude this derivation, a summary of the perturbing acceleration of a third-body in terms of q, $\mathbf{r}$, and $\boldsymbol{\rho_j}$ is given below.

$$\mathbf{a}_{3^{rd}} = \sum_{j=3}^{N} \frac{\mu_j}{d_j^3} \left[\mathbf{r} + f(q_j)\boldsymbol{\rho}_j\right] \tag{2.17}$$

with

$$\begin{aligned} q_j &= \frac{\mathbf{r} \cdot (\mathbf{r} - 2\boldsymbol{\rho}_j)}{\boldsymbol{\rho}_j \cdot \boldsymbol{\rho}_j} = \frac{\mathbf{r} \cdot \mathbf{r} - 2\boldsymbol{\rho}_j}{\boldsymbol{\rho}_j \cdot \boldsymbol{\rho}_j} \\ &= \frac{r^2 - 2r\rho_j \cos\alpha_j}{\rho_j^2} \\ &= \frac{r}{\rho_j}\left(\frac{r}{\rho_j} - 2\cos\alpha_j\right) \end{aligned} \tag{2.18}$$

and α_j is the angle between the vectors $\mathbf{r}$ and $\boldsymbol{\rho}_j$ as shown in Figure 2.2. Thus, the perturbing acceleration from a third-body can be calculated only requiring the knowledge of the position of the third-body (the sun or moon for example) relative to the Earth. The relative positions of these third bodies relative to the Earth are given in Chapter 5 of reference [9]. There, approximations to the ECI (Earth Centered Inertial) reference frame positions of the Sun and Moon are derived and

an algorithm is outlined to calculate them.

2.2.2 Geopotential Perturbation

In developing the equations for a satellite in orbit around the Earth in the previous section of this chapter, an assumption was made that the Earth is a perfectly homogeneous sphere. This assumption greatly simplifies the gravitational potential of the Earth and is not an accurate representation. A derivation for the perturbing acceleration due to a non-spherical gravity field is given in pp. 508-521 of [9]. The resulting geopotential equation is

$$U = \frac{\mu}{r}\left\{1 - \sum_{l=2}^{\infty} J_l \left(\frac{R_\oplus}{r}\right)^l P_l[\sin(\phi_{sat})] + \sum_{l=2}^{\infty}\sum_{m=1}^{l}\left(\frac{R_\oplus}{r}\right)^l P_{l,m}[\sin(\phi_{sat})]\left[C_{l,m}\cos(m\lambda_{sat}) + S_{l,m}\sin(m\lambda_{sat})\right]\right\} \tag{2.19}$$

where

$$P_l[\gamma] = \frac{1}{2^l l!}\frac{d^l(\gamma^2-1)^l}{d\gamma^l} \tag{2.20}$$

$$P_{l,m}[\gamma] = \frac{1}{2^l l!}(1-\gamma^2)^{m/2}\frac{d^{l+m}(\gamma^2-1)^l}{d\gamma^{l+m}} = (1-\gamma^2)^{m/2}\frac{d^m P_l[\gamma]}{d\gamma^m} \tag{2.21}$$

$$J_l = -C_{l,0} \text{ (zonal harmonics)}$$

and $\gamma = \sin(\phi_{sat})$

(2.22)

P_l and $P_{l,m}$ are Legendre polynomials and associated Legendre functions respectively, $R_\oplus$ is the mean radius of the planet, and $C_{l,m}$ and $S_{l,m}$ are coefficients for sectorial and tesseral harmonics of the planet. These values for Earth are given in Appendix D of [9]. The zonal harmonics are only a function of the latitude (ϕ_{sat}) while the sectorial harmonics are only dependent on longitude (λ_{sat}). The tesseral harmonics

describe the pertubation from a particular region of the planet. A graphical description of these three harmonics for the Earth are given in Figure 2.1 on p. 9 of reference [10] as well as Figures 8.5 and 8.6 on p. 519-520 of reference [9]. These figures are very insightful and allow for an intuitive understanding of the gravity model. The perturbing acceleration as a result of the geopotential can be calculated using the gradient of the potential, U, as (shown in reference [12]),

$$\mathbf{a}_{geo} = \nabla U = \frac{\partial U}{\partial \mathbf{r}} = \frac{\partial U}{\partial x}\hat{\mathbf{i}} + \frac{\partial U}{\partial y}\hat{\mathbf{j}} + \frac{\partial U}{\partial z}\hat{\mathbf{k}} \tag{2.23}$$

where $\hat{\mathbf{i}}, \hat{\mathbf{j}}, \hat{\mathbf{k}}$ are the Cartesian unit vectors.

With equation (2.23), the perturbing geopotential acceleration can be calculated. However, the geopotential is given in terms of the geocentric latitude and longitude which are in ECF (Earth Centered Fixed) reference frame while the orbit propagation is performed in ECI frame. Therefore, a coordinate transformation of the acceleration in equation (2.23) into ECI needs to be done. This can be achieved by the following transformation matrix [13].

$$\mathbf{a}_{geo}^{ECI} = T_{xyz}^{XYZ} T_{r\phi\lambda}^{xyz} \mathbf{a}_{geo} \tag{2.24}$$

where

$$T_{r\phi\lambda}^{xyz} = \begin{bmatrix} \cos\phi\cos\lambda & -\sin\phi\cos\lambda & -\sin\lambda \\ \cos\phi\sin\lambda & -\sin\phi\sin\lambda & \cos\lambda \\ \sin\phi & \cos\phi & 0 \end{bmatrix}$$

and

$$T_{xyz}^{XYZ} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

While equation (2.19) can be used to model the geopotential field of the Earth to any degree, with increasing order of l and m, the terms affect on the spacecraft decreases. With this in mind, a 5×5 gravity field is sufficiently complex model to accurately model the geopotential perturbation on a satellite.

This chapter outlined the dynamics used in the orbit propagation performed in this study and to their usefulness. The next chapter outlines the integration algorithms that are of interest in this thesis and explains their properties in detail.

Chapter 3
An Overview of the Algorithms

The numerical integration methods presented here are categorized into a group of algorithms known as symplectic algorithms. They are used to integrate equations of motion for Hamiltonian systems. The dynamics presented in Chapter 2 are an example of a Hamiltonian system. They form a subset of a class of geometric numerical integration algorithms. All algorithms of this class preserve the geometric properties in the differential equations. An interesting and important property of these algorithms is that they conserve the Hamiltonian quantity of the dyanmical system. This conservation property makes them very useful for long-term propagation of chaotic systems as this can be helpful in acheiving accuracy. They widely used in molecular dynamics, quantum physics and celestial mechanics. An example of the effectiveness of a symplectic integrator can be seen in Section I.2.3 of reference [11] where the motion of outer planets in the solar-system is simulated through the symplectic Euler and the Störmer-Verlet with the implicit and explicit Euler method. The results (in Figure 2.4 on p. 14 in reference [11]) show that through the explicit and implicit Euler methods, Jupiter and Saturn are ejected out of the Solar System while the symplectic algorithms (even with a larger time step) maintain the periodic orbits of the planets. Also, these results demonstrate the stability of the first two (symplectic) algorithms compared than the implicit

and explicit Euler. These results show the effectiveness of symplectic algorithms to dynamical system in astrodynamics where similar equations are used to propagate satellite orbits. The algorithms outlined in this chapter are symplectic methods of increasing accuracy which are described in detail with emphasis on their speed and efficiency including the basis for choosing them.

3.1 Verlet

The Verlet algorithm is credited to Verlet who, in the 1960s, used it in molecular dynamics for simulating the motion of particles [14]. Other early uses of this algorithm include the works of Cowell and Crommelin in determining the Halley's Comet in 1909 and Størmer in 1907 for determining trajectories of electric particles within magnetic field. One of the formulations, among many, of the Verlet Algorithm is called Störmer-Verlet Method and it is described in equation (3.1).

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \ddot{\mathbf{r}}_n \delta t^2 + \mathcal{O}(\delta t^4) \tag{3.1}$$

for $n = 1, 2, \ldots$ and where δt is the time step.

The basic Störmer-Verlet Method does not involve the explicit calculation of the velocity. A common formula used to calculate the velocity in the is given in equation (3.2) where a simple averaging formula is used,

$$\mathbf{v}_n = \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{\delta t} + \mathcal{O}(\delta t^2) \tag{3.2}$$

Few observations can be drawn from the above equations. First, the position error is of order 4 but the Verlet algorithm overall is of order 2 since the velocity is of order 2. As the update scheme in equation (3.1) is not dependent on velocity,

equation (3.2) can be ignored. However, the velocity is necessary in calculating many conservation quantities such as kinetic energy and momentum. Secondly, the algorithm requires the position at the $(n-1)^{th}$ time step in order to calculate the position and velocity at the n^{th} time step. Algorithms of this kind are not considered to be "self-starting" implying that the first value for position has to be calculated some other way using the initial conditions. This is the challenge for performing a numerical integration routine. A solution to this problem is to initialize the first position vector ($\mathbf{r}_1$) using a Taylor series approximation of any order greater than 2 (since that the the order of the Verlet position update scheme). This might result in an overall increase in the error however over a long period of time, this initial error in the update scheme can be ignored. The following is the initialization formula used in this thesis,

$$\mathbf{r}_1 = \mathbf{r}_0 + \mathbf{v}_0 \delta t \tag{3.3}$$

where $\mathbf{r}_0$ and $\mathbf{v}_0$ are the initial conditions.

With this initialization, the Verlet algorithm can be used. However, there are a couple of problems with the Verlet algorithm. The acceleration ($\ddot{\mathbf{r}}$) can only be dependent on the position. This presents a problem when used in astrodynamics, since some perturbing accelerations can be a function of both position and velocity. However, there are a lot of applications in which this algorithm can be used for orbit propagation.

3.2 Velocity Verlet/Leapfrog

The Velocity Verlet algorithm is a slight modification of the Verlet algorithm. Unlike Verlet, the Velocity Verlet does not require that the acceleration be only dependent on position, which makes it more useful in applications in astrodynamics. The Leapfrog algorithm is mathematically identical to the Velocity Verlet but with staggered position and velocity updates. The update scheme* is given as:

Leapfrog:

$$\mathbf{r}_n = \mathbf{r}_{n-1} + \mathbf{v}_{n-1/2}\delta t \tag{3.4a}$$

$$\mathbf{v}_{n+1/2} = \mathbf{v}_{n-1/2} + \ddot{\mathbf{r}}_n\delta t \tag{3.4b}$$

Velocity-Verlet:

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_n\delta t + \ddot{\mathbf{r}}_n\delta t^2/2 \tag{3.4c}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + (\ddot{\mathbf{r}}_n + \ddot{\mathbf{r}}_{n+1})\delta t/2 \tag{3.4d}$$

It is apparent from observing equation (3.4a) that in order to use the Leapfrog update method, $\mathbf{v}_{1/2}$ must be computed in order to initialize the algorithm. While it can be calculated using similar method described for initializing Verlet, the alternate formulation of the Velocity Verlet outlined by equations (3.4c)-(3.4d). The latter formulation is also advantageous for calculations of conserved quantities of dynamical systems such as energy and angular momentum since the position and velocity are defined at the n^{th} time step instead of being one-half step away from one another. Another interesting property of these two algorithms, besides their symplecticity is time reversiblility. This guarantees that one can integrate

*"The Leapfrog Integrator" Leapfrog Integrator.Available: http://einstein.drexel.edu/courses/Comp_Phys/Integrators/leapfrog/, Accessed: March 28, 2016

a system n steps forward in time, then integrate n steps backward and achieve the initial conditions exactly (within floating point precision). This property of this algorithm is among the reasons for its use in numerical integration of various dynamical systems and is therefore chosen here due to this prolific use.

3.3 Forest-Ruth

While the previous algorithms are of order 2, higher-order algorithms do exist. Some software applications such as MATLAB come with higher-order Runge-Kutta methods. However, the following were chosen since since they are higher-order symplectic algorithms. The intention for choosing these algorithms is to investigate the trade-off of between the accuracy obtained due to the higher-order update method and the increase in computational time due to the number of function evaluations for various orbtial dynamics. The first of the algorithms is known as the Forest-Ruth algorithm (reference [15]) which is outlined in the recursive relationships shown in equations (3.5a)-(3.5f).

Forest-Ruth:

$$\mathbf{r}_1 = \mathbf{r}_{n-1} + \theta \mathbf{v}_{n-1} \delta t/2 \tag{3.5a}$$

$$\mathbf{v}_1 = \mathbf{v}_{n-1} + \theta \ddot{\mathbf{r}}_1 \delta t \tag{3.5b}$$

$$\mathbf{r}_2 = \mathbf{r}_1 + (1-\theta)\mathbf{v}_1 \delta t/2 \tag{3.5c}$$

$$\mathbf{v}_2 = \mathbf{v}_1 + (1-2\theta)\ddot{\mathbf{r}}_2 \delta t \tag{3.5d}$$

$$\mathbf{r}_n = \mathbf{r}_2 + \theta \mathbf{v}_2 \delta t/2 \tag{3.5e}$$

$$\mathbf{v}_n = \mathbf{v}_2 + \theta \ddot{\mathbf{r}}_3 \delta t \tag{3.5f}$$

where

$$\theta = \frac{1}{2 - \sqrt[3]{2}} \tag{3.5g}$$

As the update scheme suggests, the Forest-Ruth algorithm requires three function evaluations for each update. While this results in an increase in the required computational time, there is an improvement in accuracy. Whether the higher accuracy justifies the decrease in computational efficiency is something that is discussed in Chapter 4 and is the main reason for implementing this algorithm in this thesis. Another observation regarding the Forest-Ruth algorithm is that the value of θ (defined in equation (3.5g)) is greater than 1 (≈ 1.351207192) which implies that in equations (3.5c) and (3.5d), the algorithm performs a reverse time step before computing the next intermediate time step. (Note that this reverse time step updates the positions that are before $\mathbf{r}_1$)[†]. This is one of the ways that the algorithm achieves the improvements in accuracy. To conclude to observations regarding this algorithm, it should be noted that the Forest-Ruth Algorithm is of order 4 and is considered to be symplectic as well as time reversible. Hence, it is very applicable to dynamical systems in astrodynamics.

3.4 PEFRL

Omelyan *et al.* [16] explored methods to improve the accuracy of the Forest-Ruth algorithm while maintaining the time reversable update scheme and achieving similar computational efficiency. One of the optimized algorithms containted in that paper is known as the Position Extented Forest-Ruth Like (PEFRL) algorithm. Due to the optimization, it is an improvement over Forest-Ruth in terms of accuracy. Also,

[†]Peter Young, https://physics.ucsd.edu/students/courses/winter2016/physics141/Assignments/leapfrog.pdf, Accessed: March 28, 2016

since it is still a 4^{th} order integrator, the computational expense is not immediately noticable. However, it does require an additional function evaluation over Forest-Ruth which, as the results in Chapter 4 demonstrate, still costs additional time. However, this algorithm is still of interest in this work due to the intriguing trade-offs between the improved accuracy and computational cost. The recursive update scheme for PEFRL algorithm is given as follows in equations (3.6a) - (3.6i):

PEFRL:

$$\mathbf{r}_1 = \mathbf{r}_{n-1} + \mathbf{v}_{n-1}\xi\delta t \tag{3.6a}$$

$$\mathbf{v}_1 = \mathbf{v}_{n-1} + \ddot{\mathbf{r}}_1(1-2\lambda)\delta t/2 \tag{3.6b}$$

$$\mathbf{r}_2 = \mathbf{r}_1 + \mathbf{v}_1\chi\delta t \tag{3.6c}$$

$$\mathbf{v}_2 = \mathbf{v}_1 + \ddot{\mathbf{r}}_2\lambda\delta t \tag{3.6d}$$

$$\mathbf{r}_3 = \mathbf{r}_2 + \mathbf{v}_2(1-2(\chi+\xi))\delta t \tag{3.6e}$$

$$\mathbf{v}_3 = \mathbf{v}_2 + \ddot{\mathbf{r}}_3\lambda\delta t \tag{3.6f}$$

$$\mathbf{r}_4 = \mathbf{r}_3 + \xi\delta t \tag{3.6g}$$

$$\mathbf{v}_n = \mathbf{v}_3 + \ddot{\mathbf{r}}_4(1-2\lambda)\delta t/2 \tag{3.6h}$$

$$\mathbf{r}_n = \mathbf{r}_4 + \mathbf{v}_n\xi\delta t \tag{3.6i}$$

where

$$\xi = 0.1786178958448091$$

$$\lambda = -0.2123418310626054$$

$$\chi = -0.06626458266981849$$

3.5 Other Algorithms and Considerations

3.5.1 Other Algorithms

These symplectic algorithms need to be validated against other widely used algorithms such as Runge-Kutta methods. In this thesis, the software program MATLAB was used which includes various integration algorithms based on Runge-Kutta methods. Among these, ODE45 is a popular choice which is based on the Dormand-Prince method[‡]. This method [17] delivers fourth-order accuracy with a fifth-order error control. This algorithm was chosen since it is of the same order of accuracy as the Forest-Ruth and PEFRL algorithms. In addition, the interesting question of whether a lower order symplectic algorithm such as Verlet or Leapfrog can be more accurate than the higher order Runge-Kutta.

These algorithms were chosen for various reasons. The Verlet and Velocity Verlet have been previously explored for orbit propagation but Forest-Ruth and PEFRL have not. Therefore, all of these algorithms are to be tested to compare them to each other and analyze their performance and applicability to problems in orbit propagation. Also, given their varying order of accuracy, it is an interesting test to compare their accuracy to Runge-Kutta methods to see whether even the second order symplectic algorithms such as Verlet and Velocity Verlet perform better due to their energy preserving properties.

[‡]"Solve nonstiff differential equations" Available: http://www.mathworks.com/help/matlab/ref/ode45.html, Accessed: March 28, 2016

3.5.2 Other Considerations

In the process of completing the objectives of this thesis, some difficulties were encountered concerning the limits of the computer programming including choice of programming environment. The limitations of MATLAB in the way it utilized memory on a computer was a limiting factor in evaluating the performance of the algorithms. Eventually, several solutions were applied to this problem. One possible solution was to implement the algorithms in C++ instead of MATLAB. Its lower-level programming environment granted more computational power over MATLAB. However, due to the more convenient graphing utilities available in MATLAB and for the included Runge-Kutta suite of integration algorithms, it was still the choice in this work.

This chapter gave a detailed overview of the algorithms used in this thesis and discussed their advantages, shortcomings, accuracy and efficiency. The next chapter, Chapter 4, gives the results obtained from the orbit propagation tests performed in this thesis and discusses the performance of these algorithms.

Chapter 4
Summary and Analysis of the Results

In this chapter, the results obtained from performing various orbit propagation tests are outlined along with explanations and discussion of these results. The ultimate objective of this chapter is to perform accurate long-term orbit propagation using these symplectic algorithms. Before these results are shown, the numerical accuracy, computational efficiency of these algorithms must be explored. In order to accomplish this, the first section of this chapter outlines the results obtained by propagating the dynamics of the two-body problem described in Chapter 2. The analytical solution to the two-body problem (known as the orbit equation) is propagated to measure the accuracy of the algorithms. Along with these results, performance characteristics of these algorithms are given. The next section presents results from orbit propagation using various perturbations as were described in Chapter 2. The objective of this section is to analyze and study, using these algorithms, the orbit evolution of a satellite in various orbits around the Earth. As with the first section, the computational efficiency of these algorithms is also presented.

4.1 Evaluation of the Algorithms

The first test shown in the following series of figures lists the orbit propagation of a GEO orbit using the Velocity Verlet, Forest-Ruth, PEFRL and ODE45 algorithms for a time-span of 1 year in 1000 second increments. A GEO orbit is circular orbit of radius 42157 km and velocity 3.0749 km/s and the gravitational parameter of the Earth is 398600 km^3/s^2. The orbits using the aforementioned algorithms are shown in Figure 4.1.

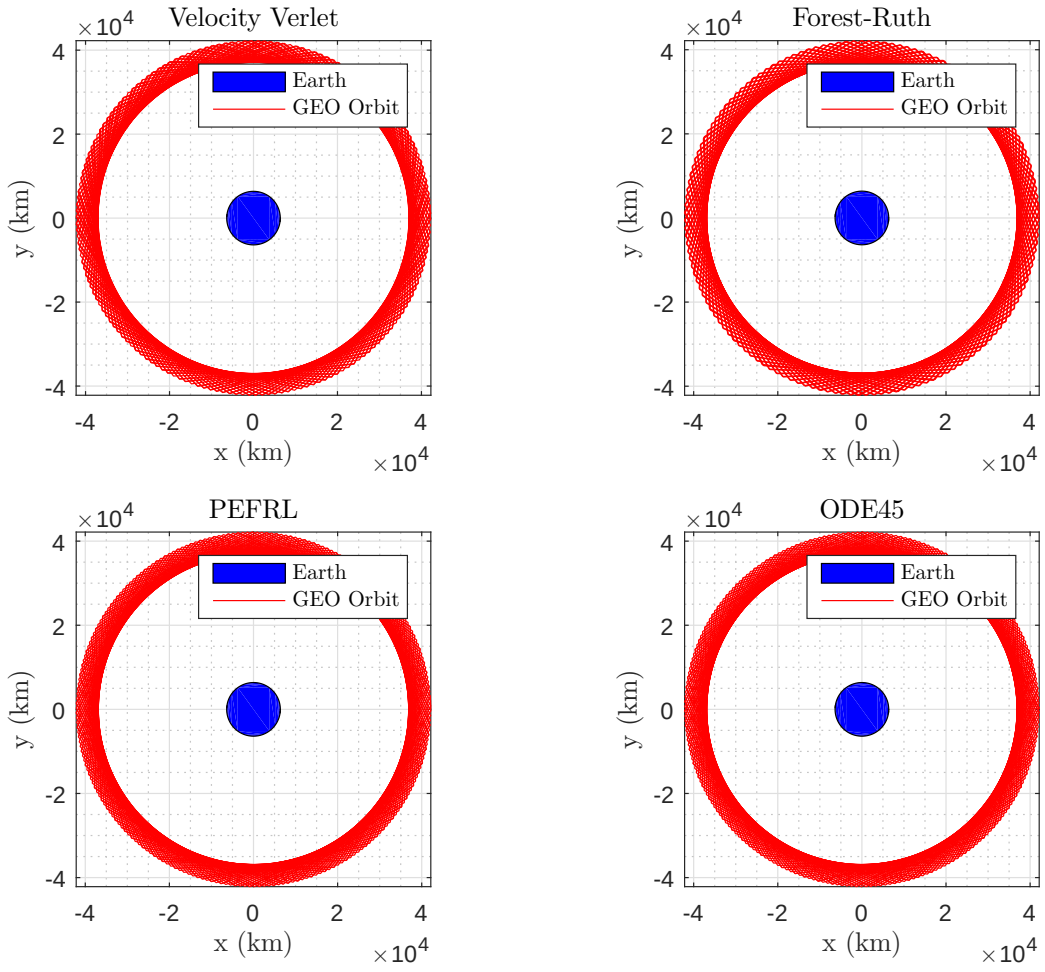

Figure 4.1. GEO Orbit ($\delta t = 1000s$ and time span of 1 year)

(Note: x, y, z are the three directions of a right handed Cartesian coordinate system.)

The graphs in Figure 4.1 shows large deviations in the position. However, this can be attributed to the fact that MATLAB connects two plot points by a straight line which means that a two points on the circumference of a circle would be connected by the chord between them and not an arc. This can be seen with greater detail by observing the plot in Figure 4.2 where a GEO orbit is plotted for two days (which is the period of the orbit) using the ODE45 algorithm. (NOTE: the time step used here is $10000s$). The diamond points are the results from the integration. As the figure shows, the there is some deviation between nearby points. Ideally, these two points would overlap since this is an integration of the simple two-body problem. However, as previously described, MATLAB connects the points with straight lines which when coupled with the integration errors, results in the large deviations. Therefore, the two points that should overlap do not and it results in the bands that are visible, especially for larger time steps of $1000s$ and $100s$.

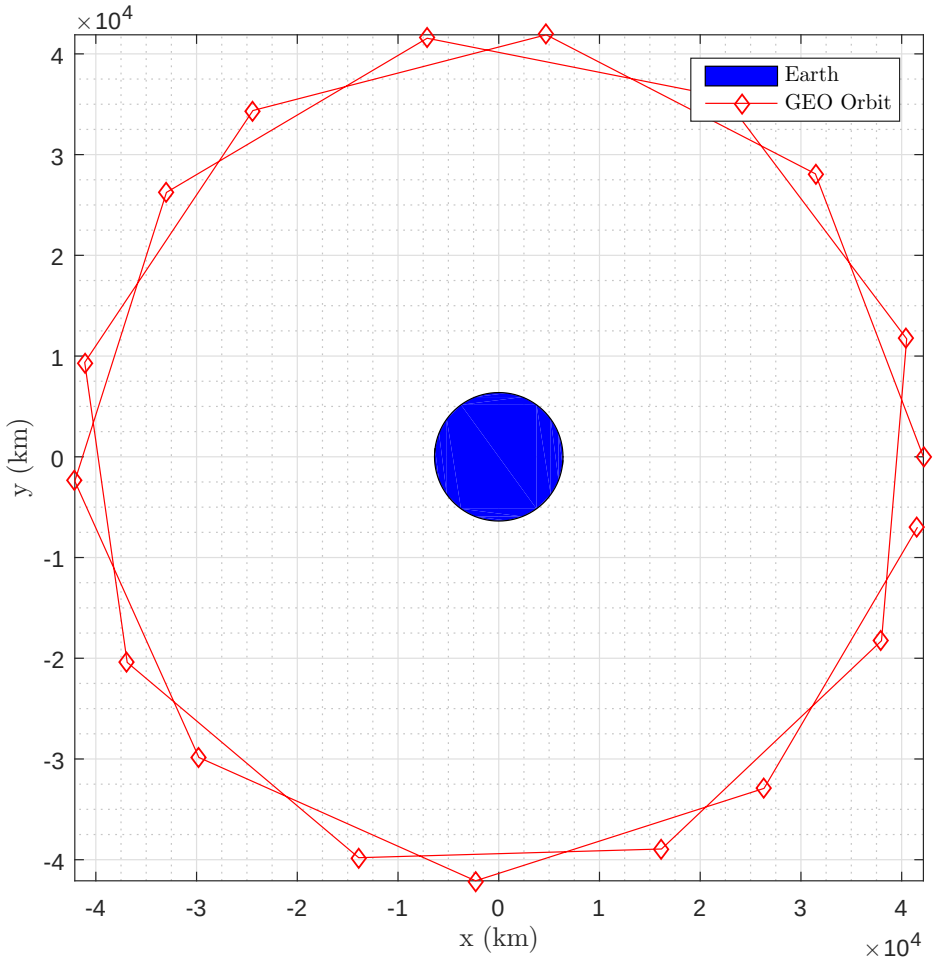

Figure 4.2. GEO Orbit ($\delta t = 10000s$) for two days

To measure the accuracy, the orbital energy (using equation (2.5)) is used. Using the reference radius and velocity for a Geostationary Orbit, the reference orbital energy is calculated as $-4.7275\ km^2/s^2$. Since the orbital energy is an invariant of the two-body problem, accuracy can be measured by measuring deviations from this reference value. The plots in Figure 4.3 display this error for the algorithms. Since the time span of the orbit propagation is only a year, the second-order Velocity Verlet and the fourth-order Forest-Ruth and PEFRL algorithms do not differ significantly. The latter two are only one order of magnitude more accurate

at preserving the energy for this time span. The other contributing factor is the time step. The formula for the percent error is given in equation 4.1.

$$\%E = \frac{|E - E_0|}{E_0} \times 100 \tag{4.1}$$

In the subsequent figures that show the percent orbital energy and percent semi-major axis error, the red lines represent the mean values of the respective quantities.

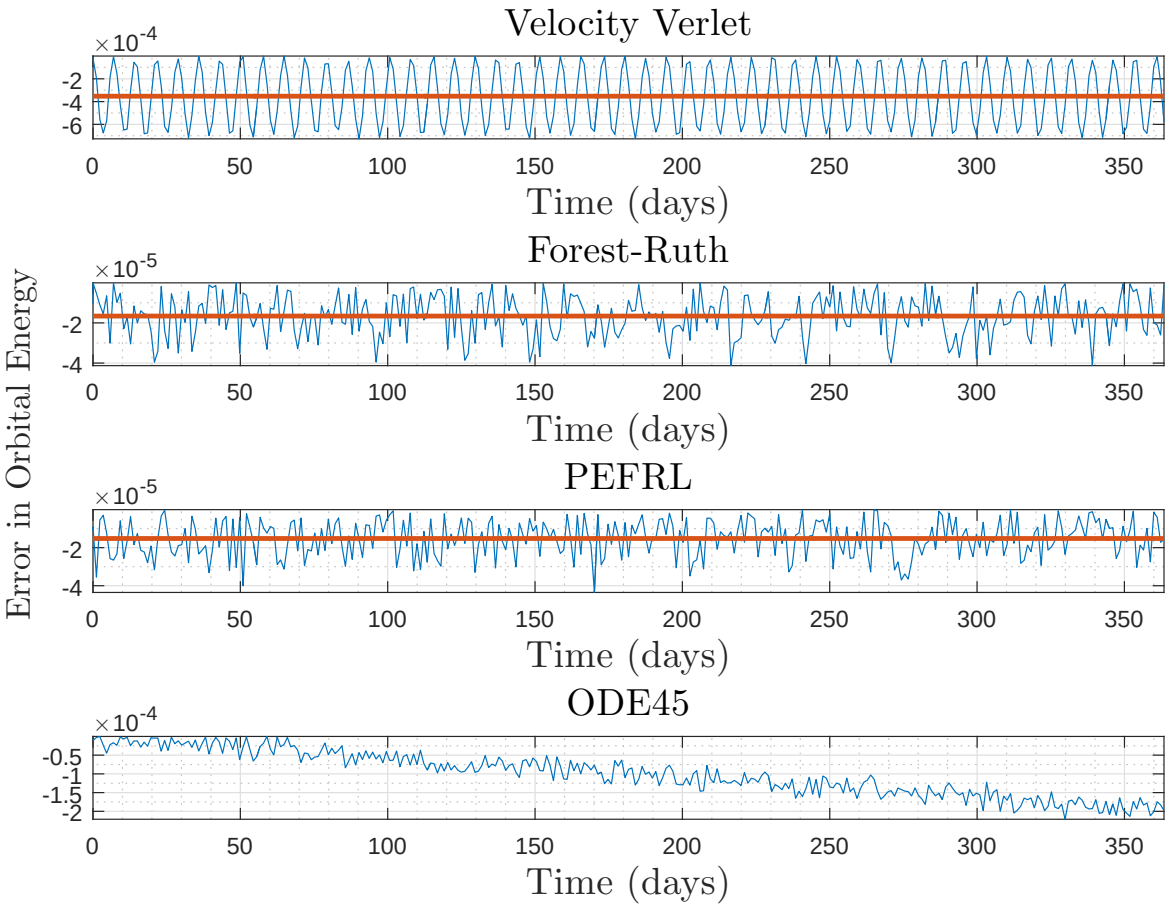

Figure 4.3. Energy Error for a GEO Orbit ($\delta t = 1000s$ and time span of 1 year)

In Figure 4.3, the Velocity Verlet and ODE45 have the same order of accuracy (10^{-4}) while the Forest-Ruth and PEFRL algorithms maintain the orbital energy to a higher order. The similarity between the Velocity Verlet and ODE45 ends there as closer observation shows that there is an increase in the orbital energy error for the ODE45 over time while the other algorithms, including Velocity Verlet, do not exhibit the same behavior. The difference between Forest-Ruth and PEFRL algorithms seems to be minimal which might be due to the short time span or the large time step. In order to observe the effect of changing the time step (δt) on the behavior of the algorithms, the same orbit was propagated with time steps of 100 and 10 seconds. The following graphs in (Figure 4.4-Figure 4.10) show the resulting orbits and the percent error in the two invariant quantities.

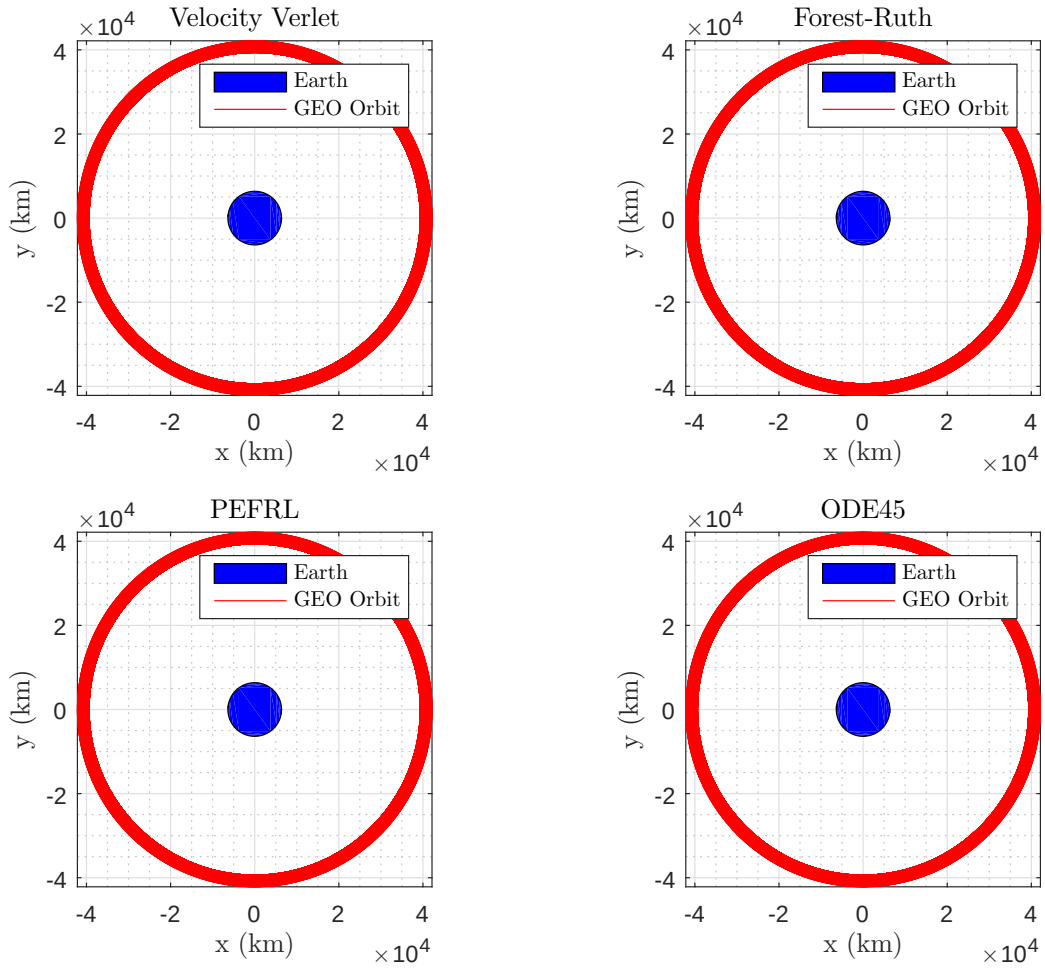

Figure 4.4. GEO Orbit ($\delta t = 100s$ and time span of 1 year)

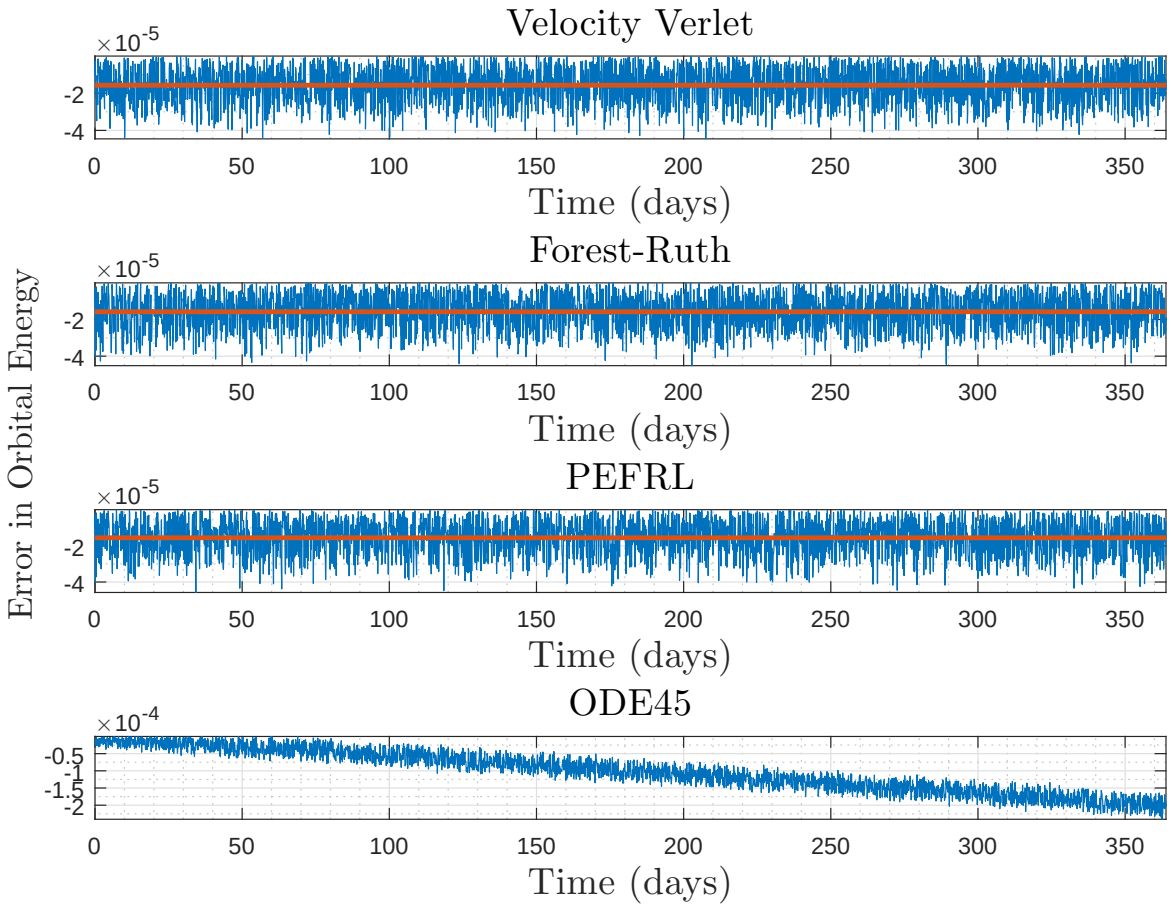

Figure 4.5. Energy Error for a GEO Orbit ($\delta t = 100s$ and time span of 1 year)

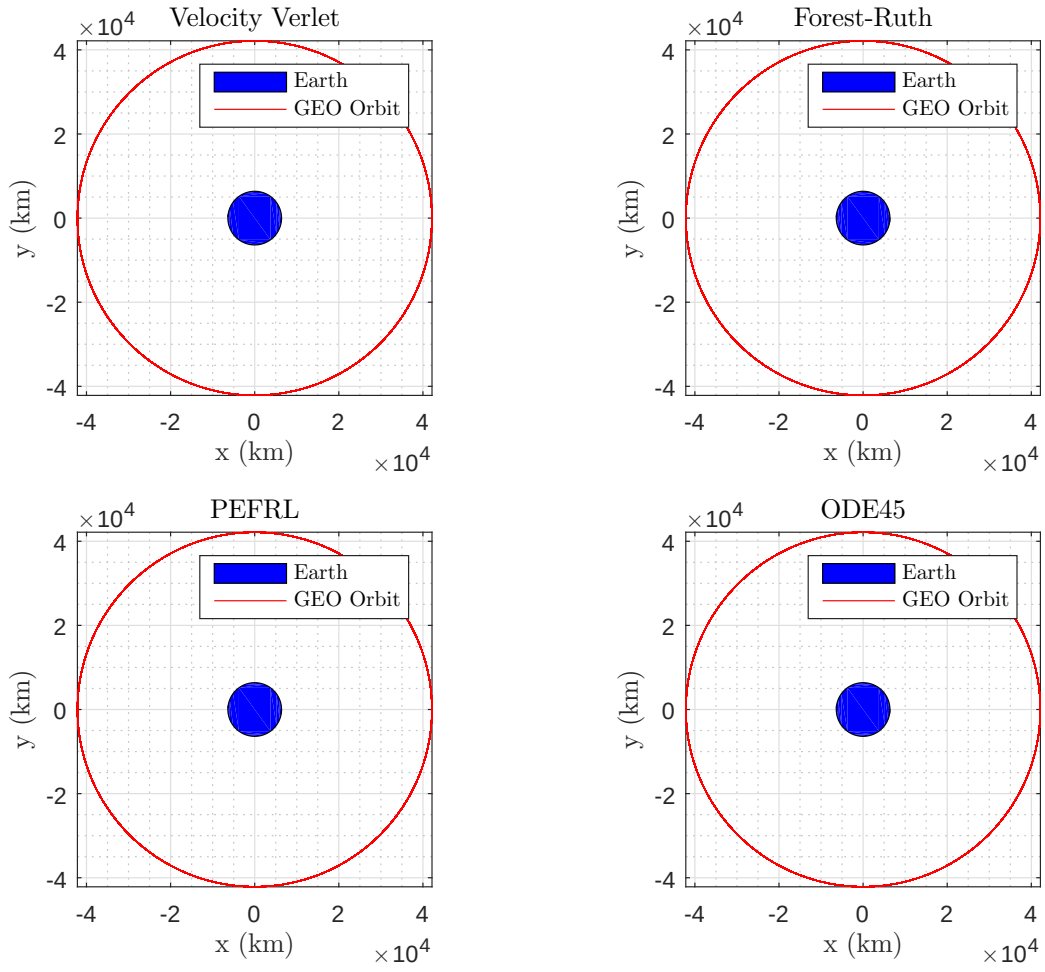

Figure 4.6. GEO Orbit ($\delta t = 10s$ and time span of 1 year)

The immediate and obvious accuracy gain by changing the time step is evident. However, more insight of the accuracy can be gained by the observing the following figure (Figure 4.7) where the percent error in the orbital energy is plotted over the time span of the orbit propagation as well as the graphs in Figure 4.22-Figure 4.10) where the percent error in the semi-major axis of the orbit is shown. The lattter is another measure of the accuracy, as described in Chapter 2.

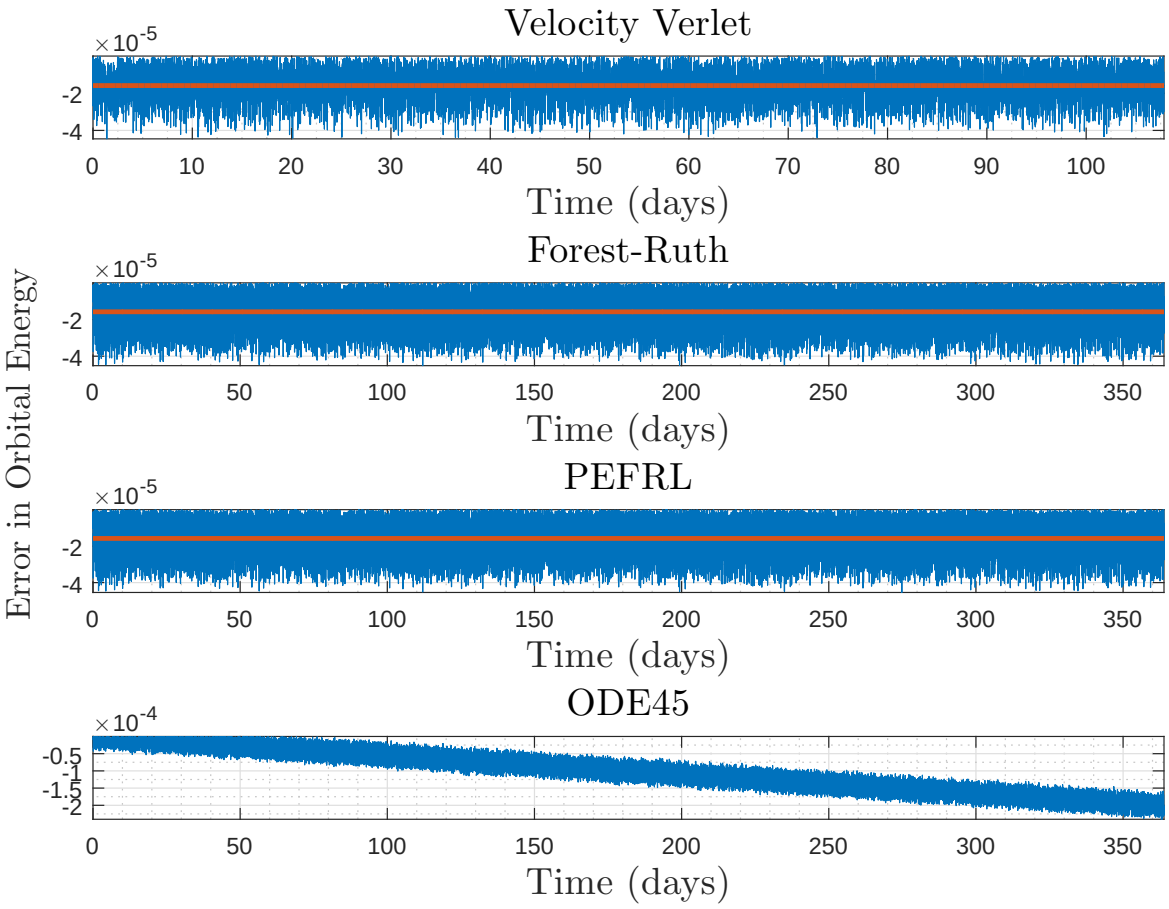

Figure 4.7. Energy Error for a GEO Orbit ($\delta t = 10s$ and time span of 1 year)

The following series of figures shows the percent error in the semi-major axis for 1000, 100 and 10 second time steps.

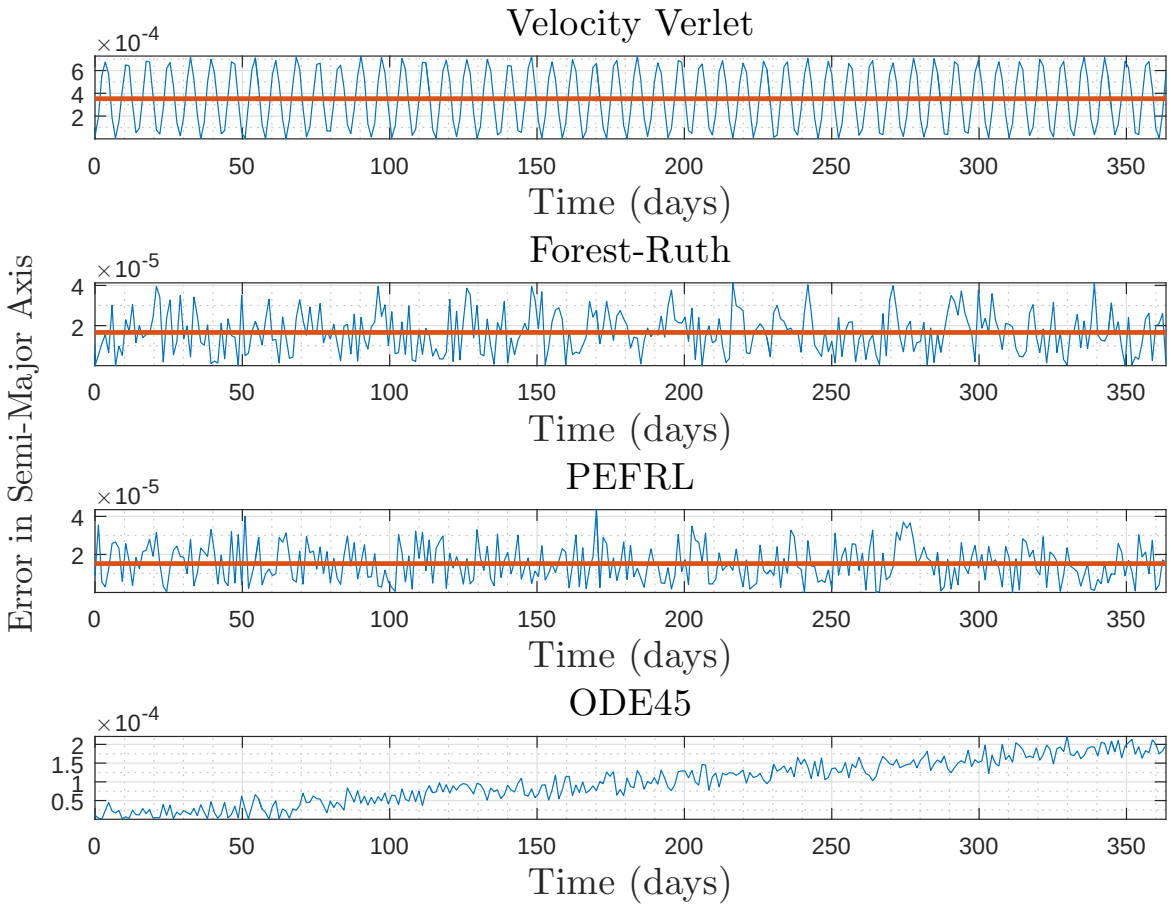

Figure 4.8. Error in Semi-major Axis for a GEO Orbit ($\delta t = 1000s$ and time span of 1 year)

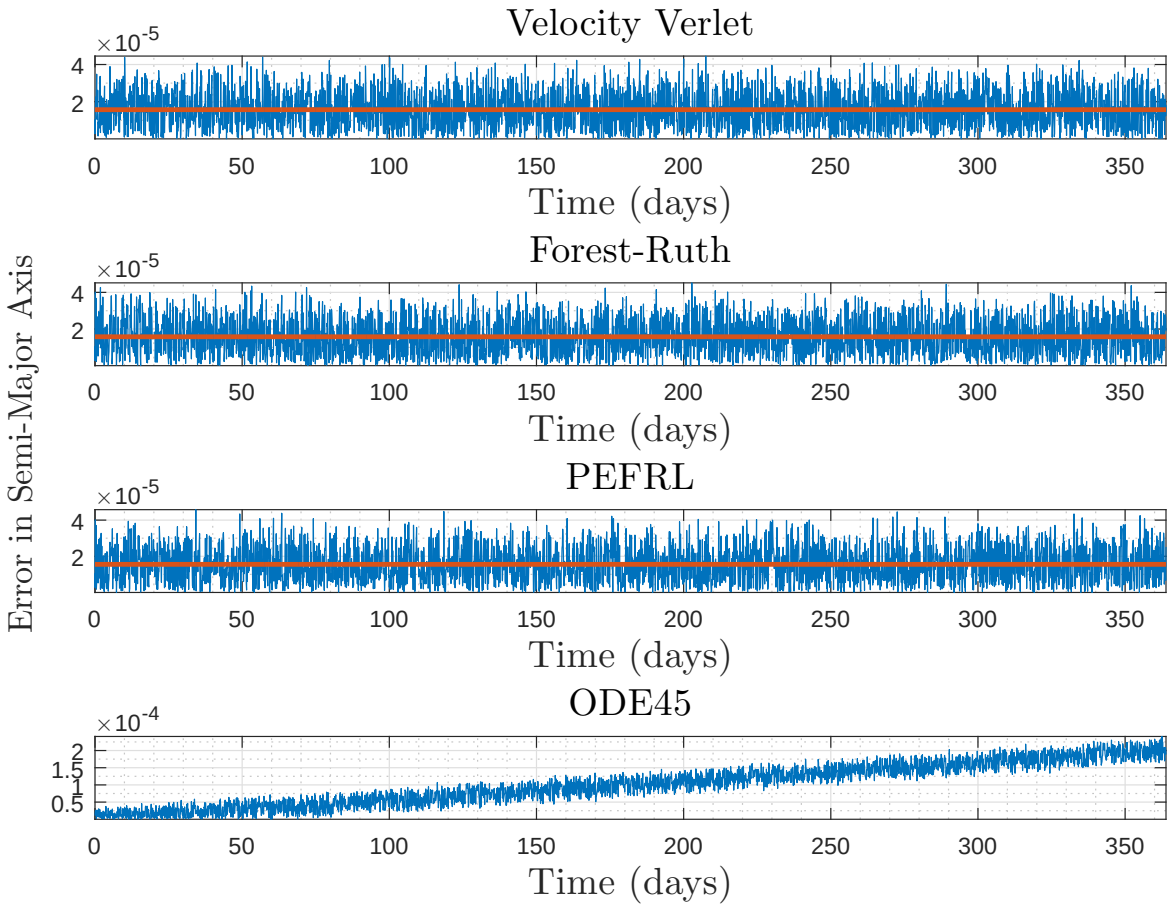

Figure 4.9. Error in Semi-major Axis for a GEO Orbit ($\delta t = 100s$ and time span of 1 year)

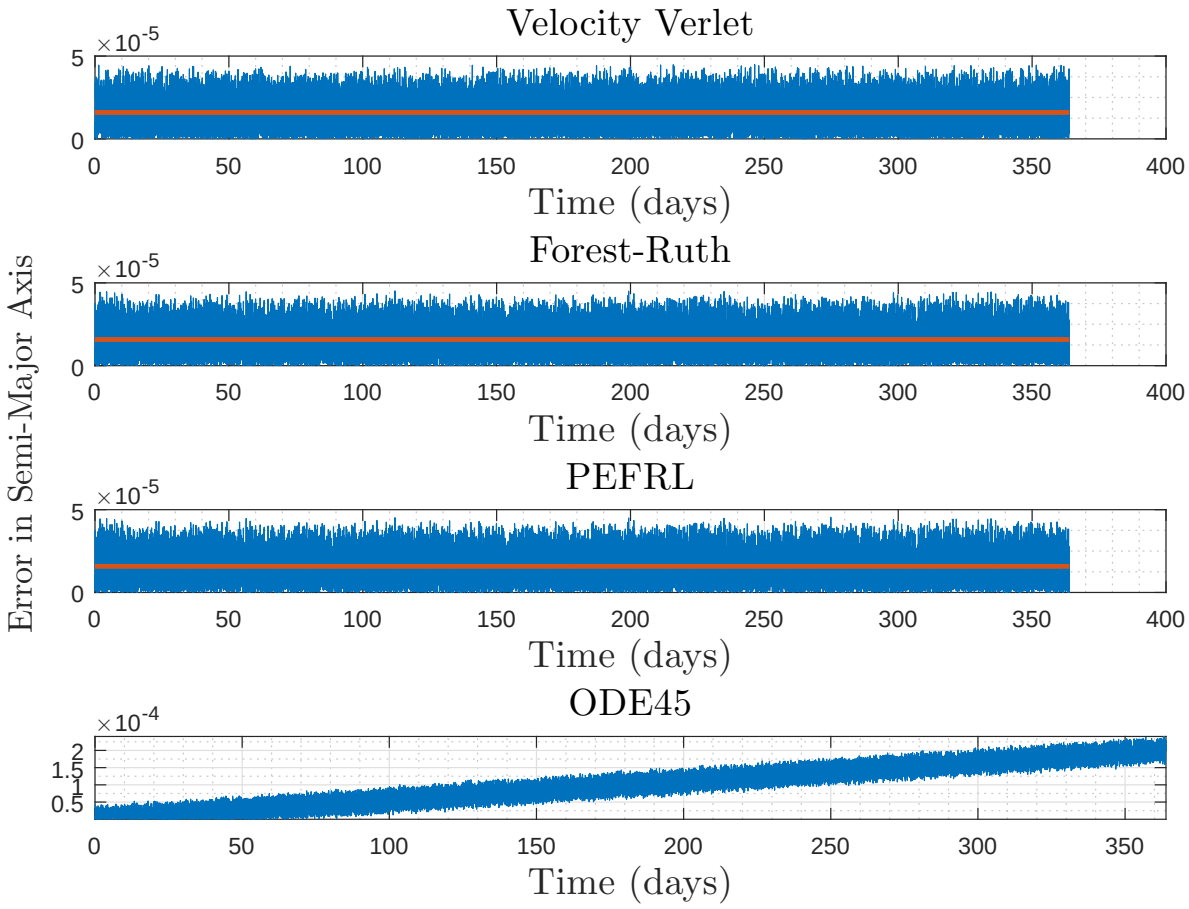

Figure 4.10. Error in Semi-major Axis for a GEO Orbit ($\delta t = 10s$ and time span of 1 year)

While the first three symplectic algorithms seem to oscillate around a nominal value, the ODE45 algorithm shows a steady rise in the energy error over time for even the smallest time step. One can conclude from the energy error figures shown above that the symplectic algorithms are more accurate than ODE45. However, the graph in figure Figure 4.9 indicates that with a time step of $1000s$, the ODE45 algorithm is more accurate than the Velocity Verlet with the semi-major axis of the orbit. Decreasing the time step to even $100s$ seems to shift the advantage in favor of the symplectic algorithms. The ODE45 algorithm seems to be imperceptible

to changes in the time step as the order of accuracy in the semi-major axis is settled around 10^{-4} at all three time steps while the symplectic algorithms get more accurate when the time step is reduces by a factor of 10.

The next step in evaluating the algorithms is to test them for longer time spans. In Figure 4.11-Figure 4.19, the orbits, the energy error, and semi-major axis error are shown for a time span of 5 years and three different time steps of $1000s$,$100s$, and $10s$.

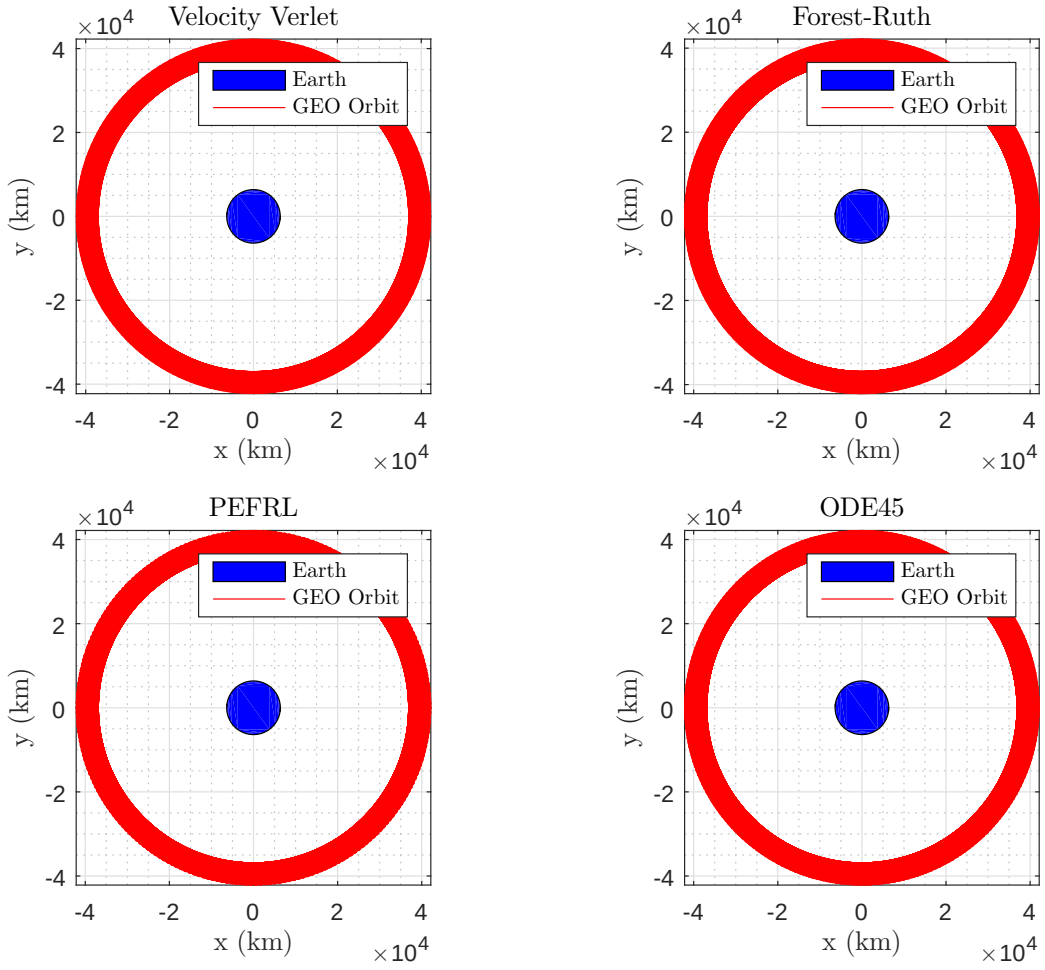

Figure 4.11. GEO Orbit ($\delta t = 1000s$ and time span of 5 years)

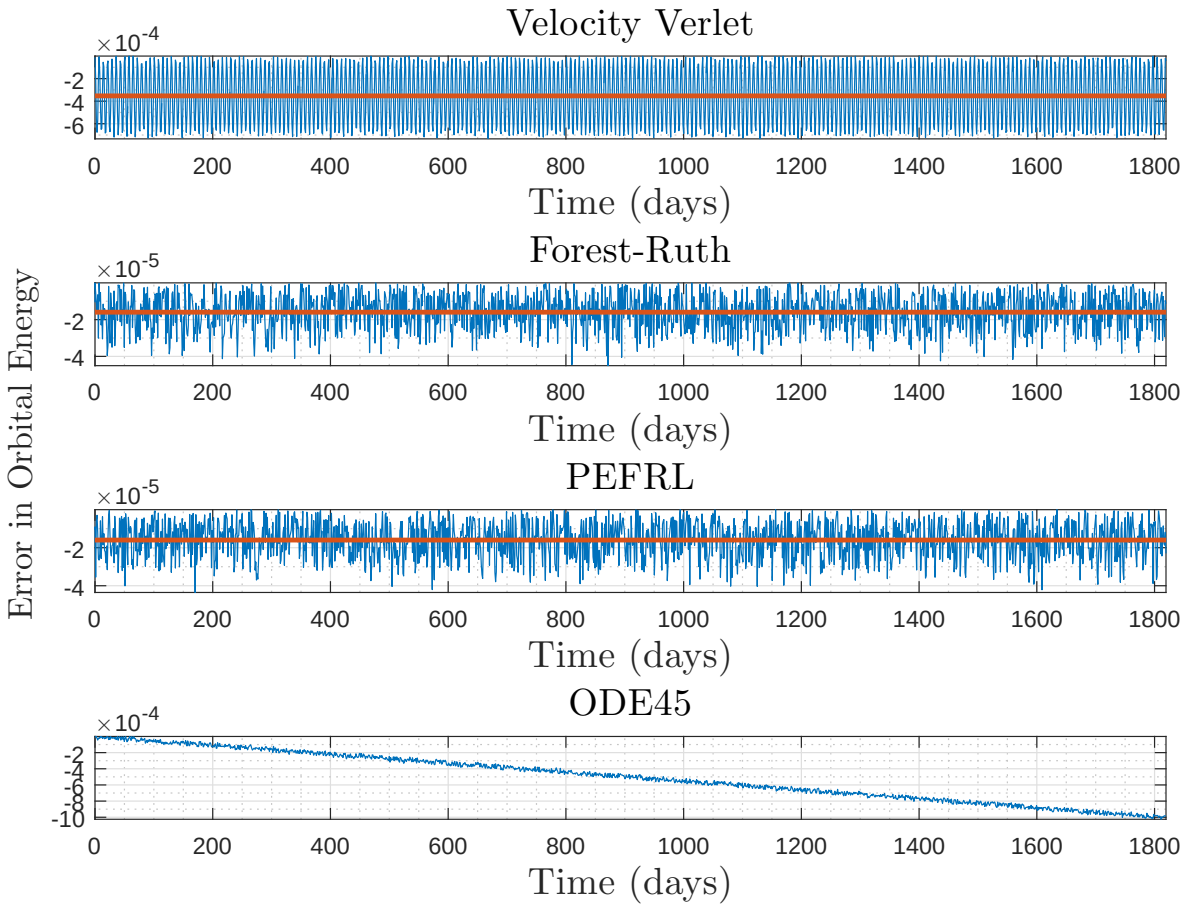

Figure 4.12. Energy Error for a GEO Orbit ($\delta t = 1000s$ and time span of 5 years)

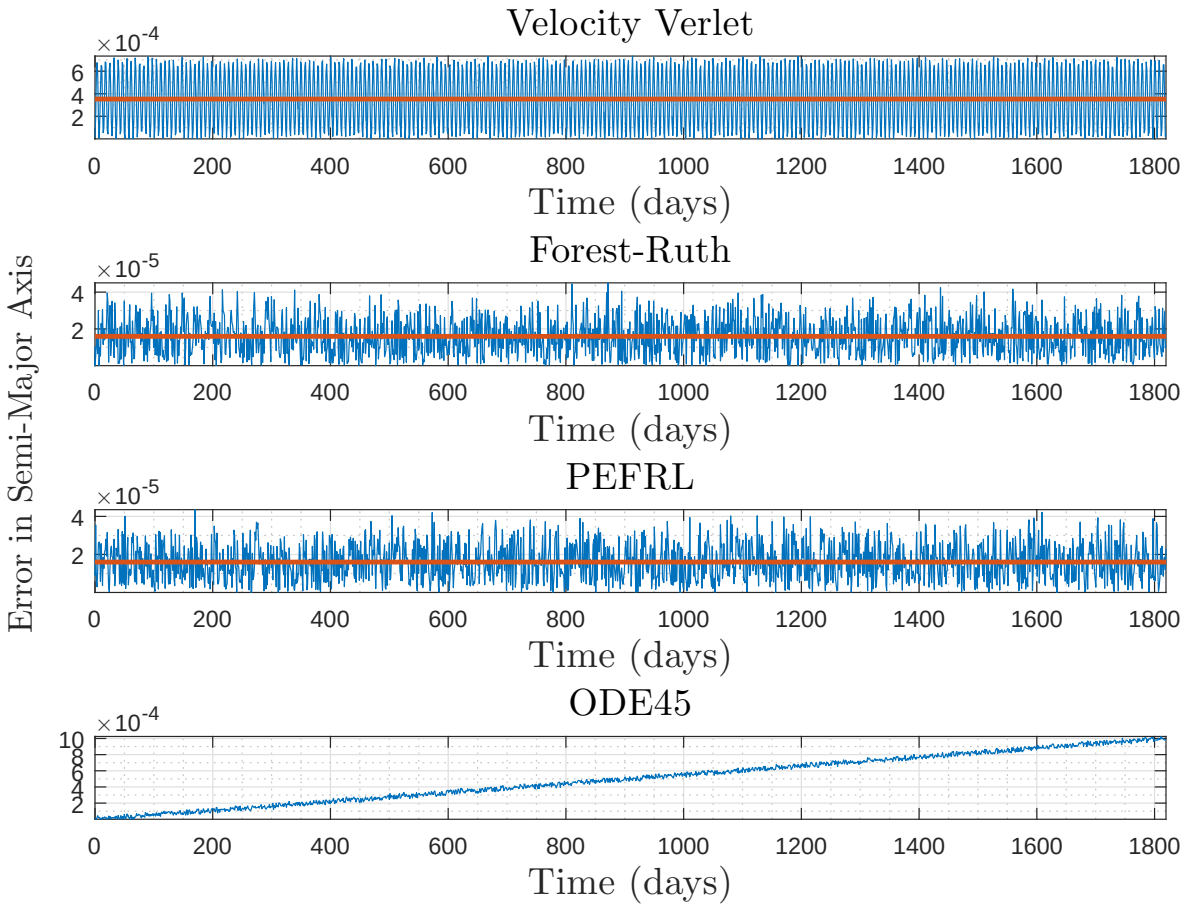

Figure 4.13. Error in Semi-major Axis for a GEO Orbit ($\delta t = 1000s$ and time span of 5 year)

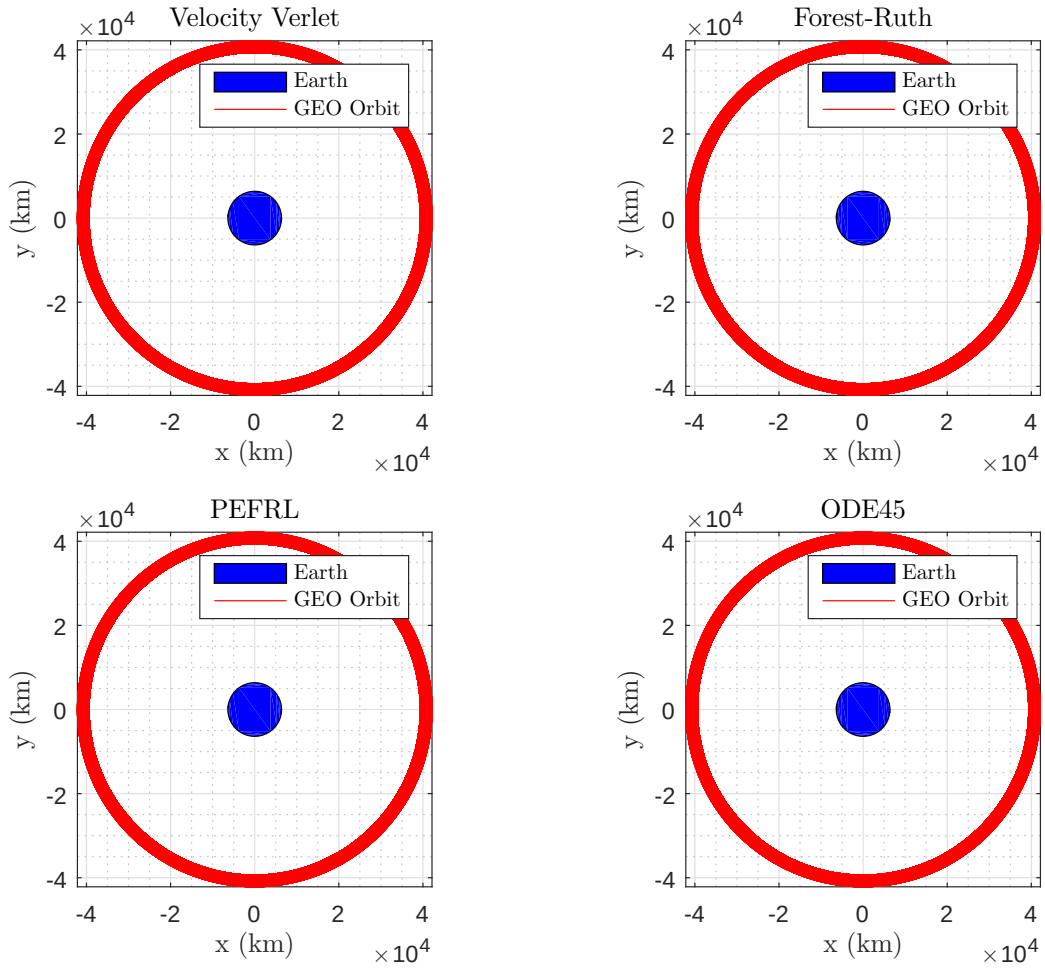

Figure 4.14. GEO Orbit ($\delta t = 100s$ and time span of 5 years)

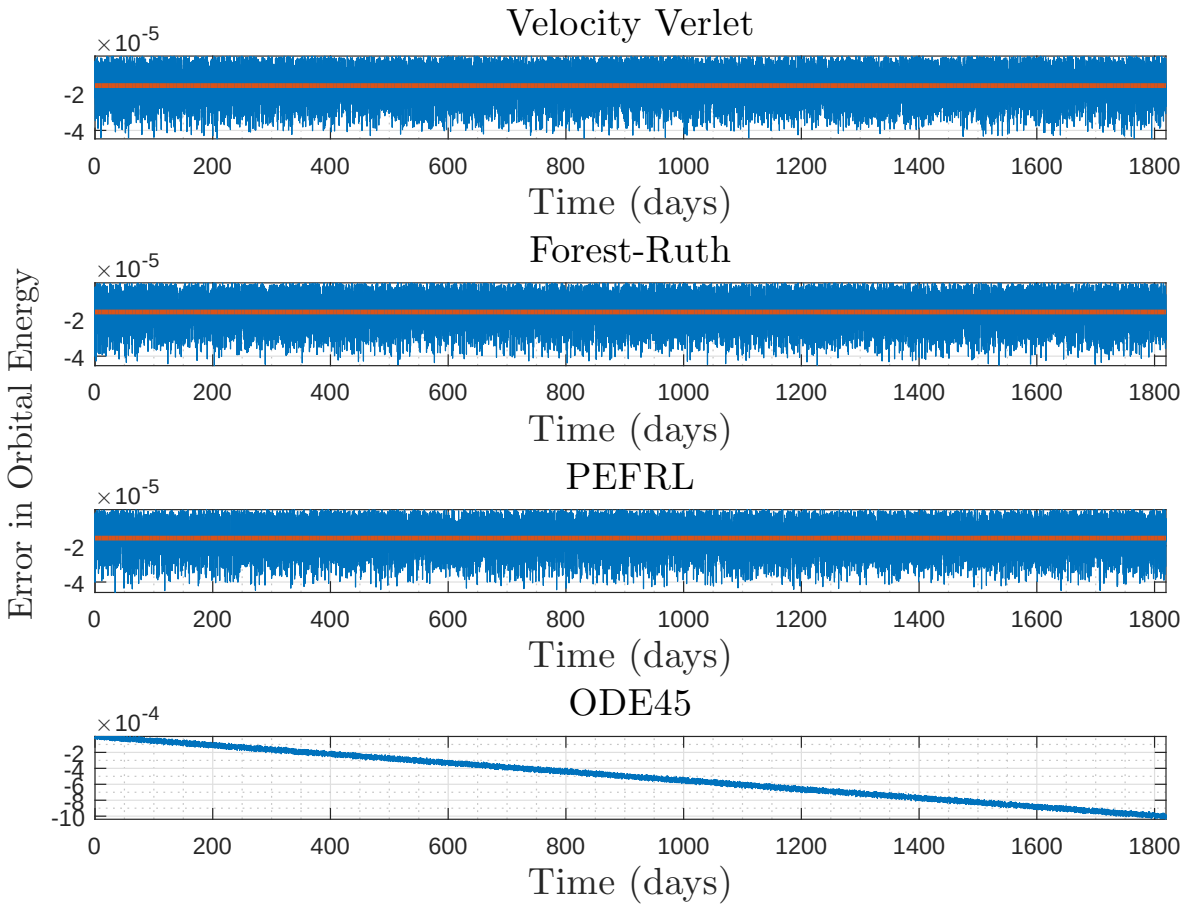

Figure 4.15. Energy Error for a GEO Orbit ($\delta t = 100s$ and time span of 5 years)

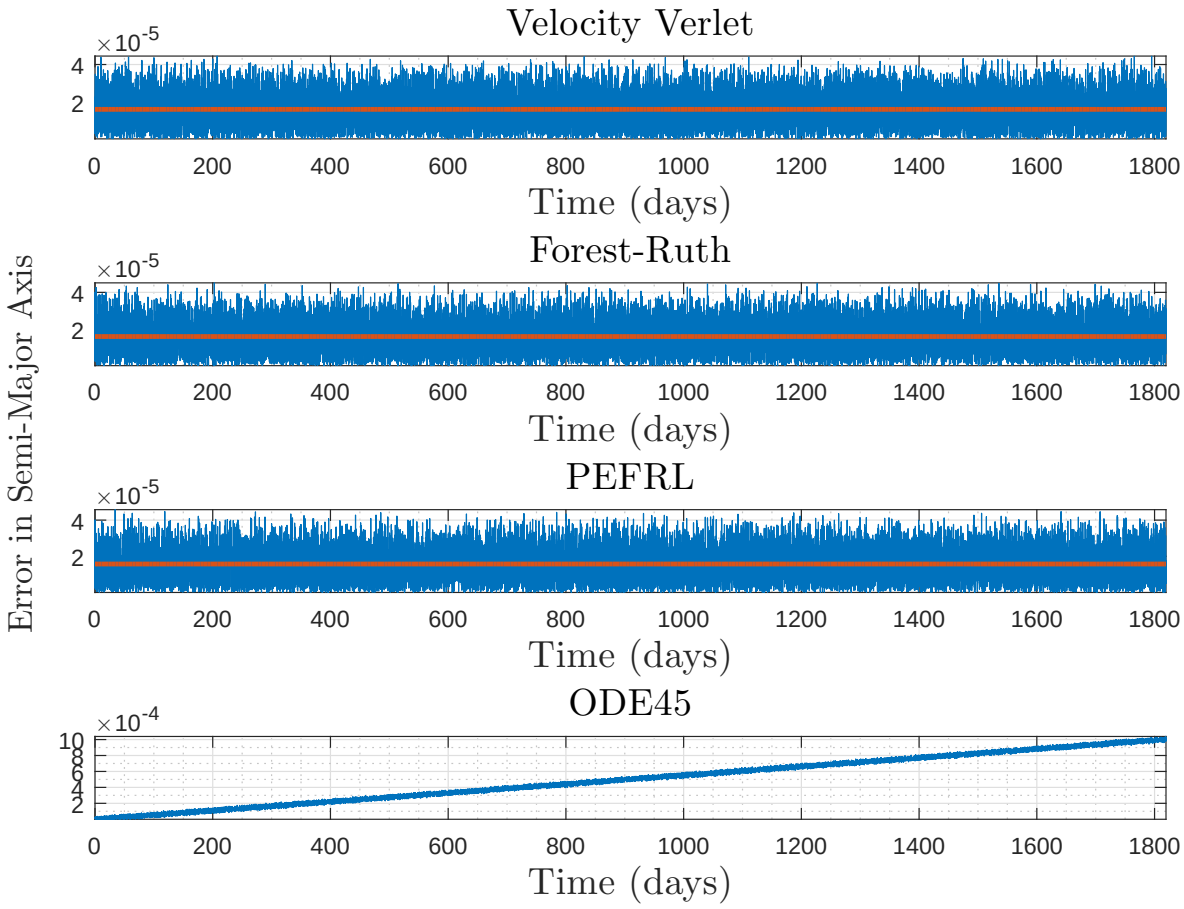

Figure 4.16. Error in Semi-major Axis for a GEO Orbit ($\delta t = 100s$ and time span of 5 year)

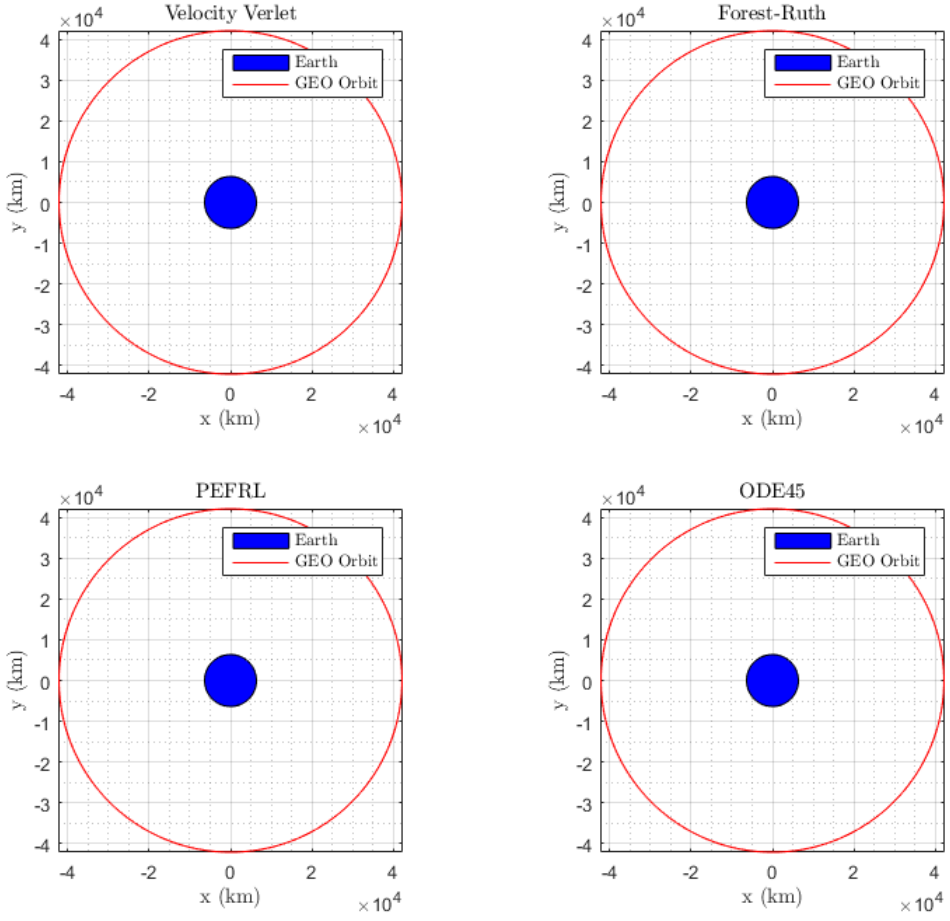

Figure 4.17. GEO Orbit ($\delta t = 10s$ and time span of 5 years)

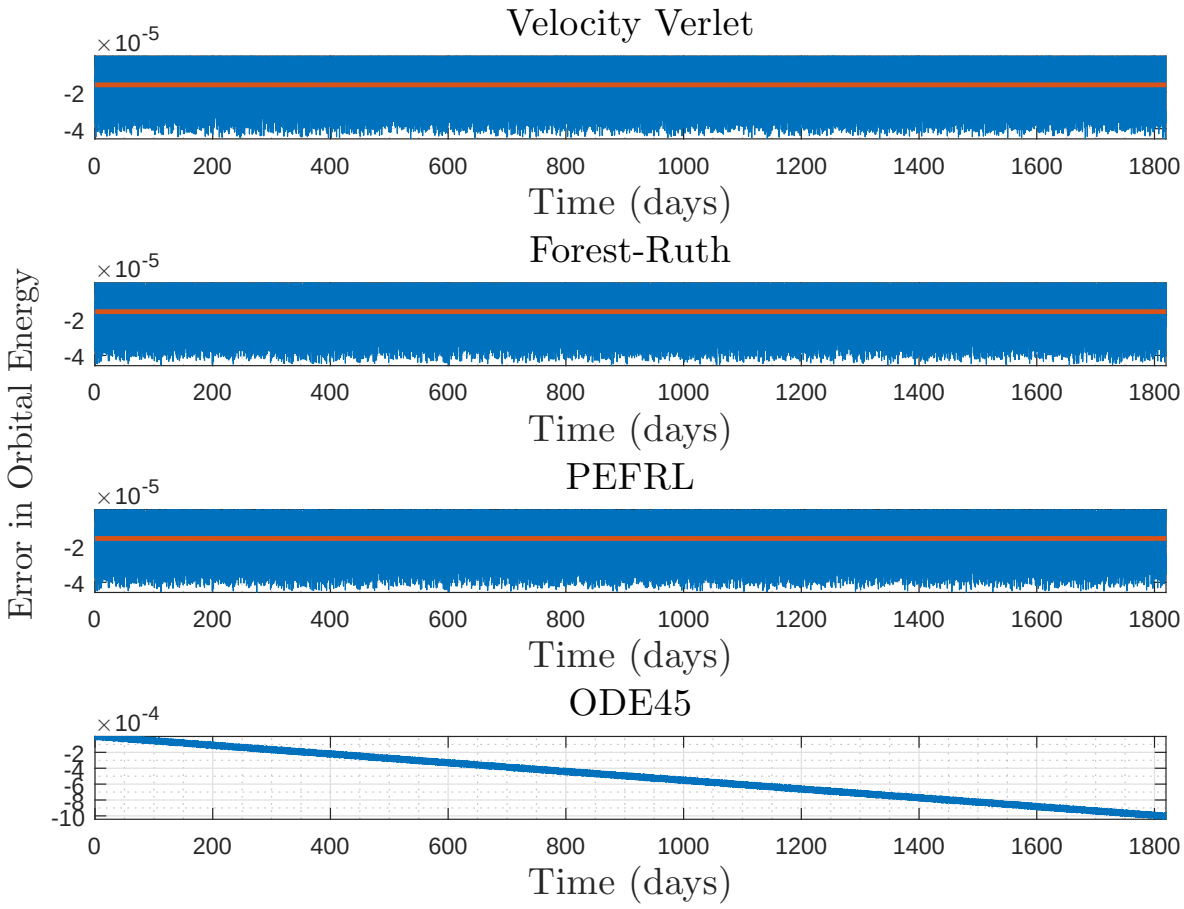

Figure 4.18. Energy Error for a GEO Orbit ($\delta t = 10s$ and time span of 5 years)

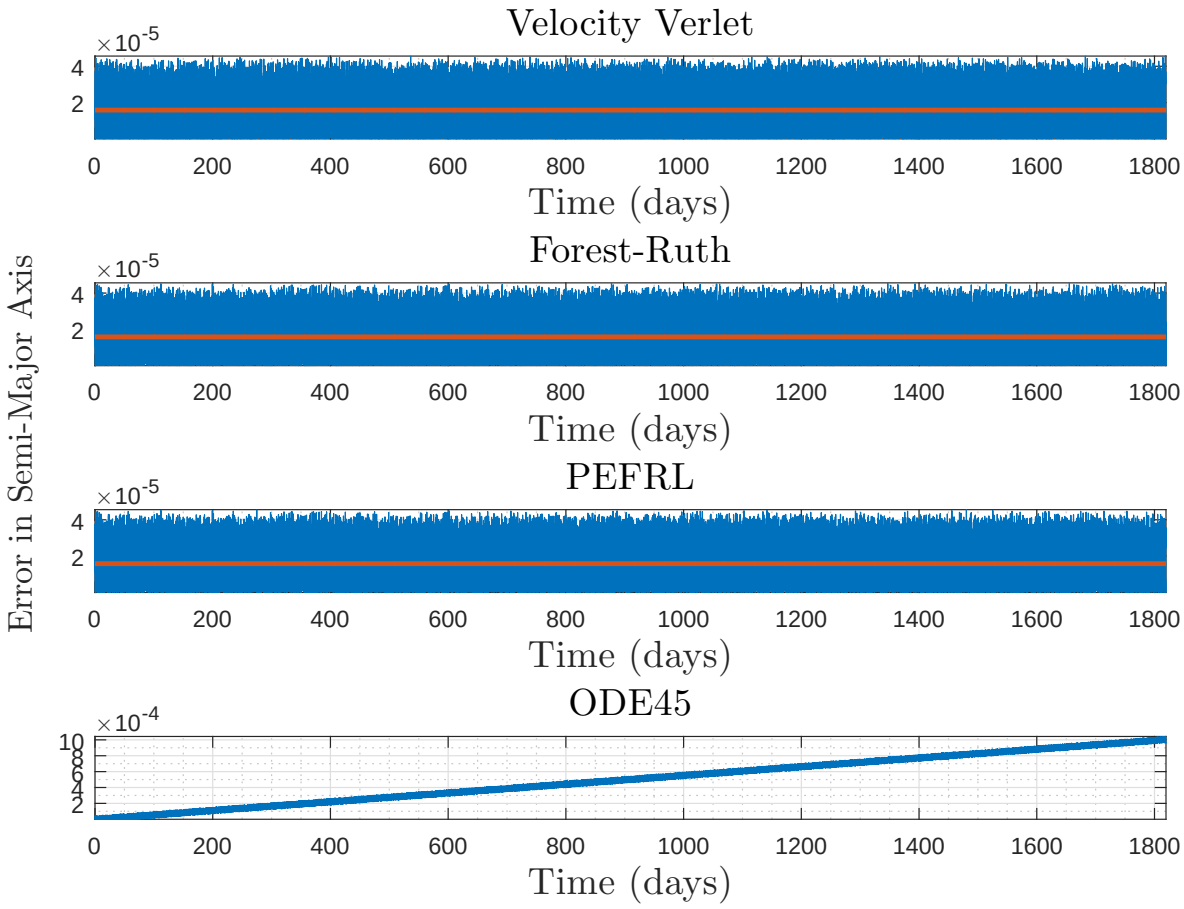

Figure 4.19. Error in Semi-major Axis for a GEO Orbit ($\delta t = 10s$ and time span of 5 year)

For this time span, the symplectic algorithms perform much better than the Runge-Kutta Method of ODE45. The energy error, especially at a time step of 10 seconds as shown in Figure 4.18, is an order of magnitude better for the symplectic algorithms. For the most part, the figures showing the percent error in the semi-major axis (Figure 4.13, Figure 4.16, and Figure 4.19) show similar results. The only exception can be seen in Figure 4.13 where error for Velocity Verlet is higher than that of ODE45. This advantage was also seen for the same time step in the Figure 4.8 and is not present in the for the figures corresponding to the $100s$ and $10s$

time steps for both time spans. The next series of graphs in Figure 4.20-Figure 4.28 show the plots for the orbit, percent error in the orbital energy, and percent error in for a time span of 10 years) at three time steps ($10s$,$100s$, and $1000s$).

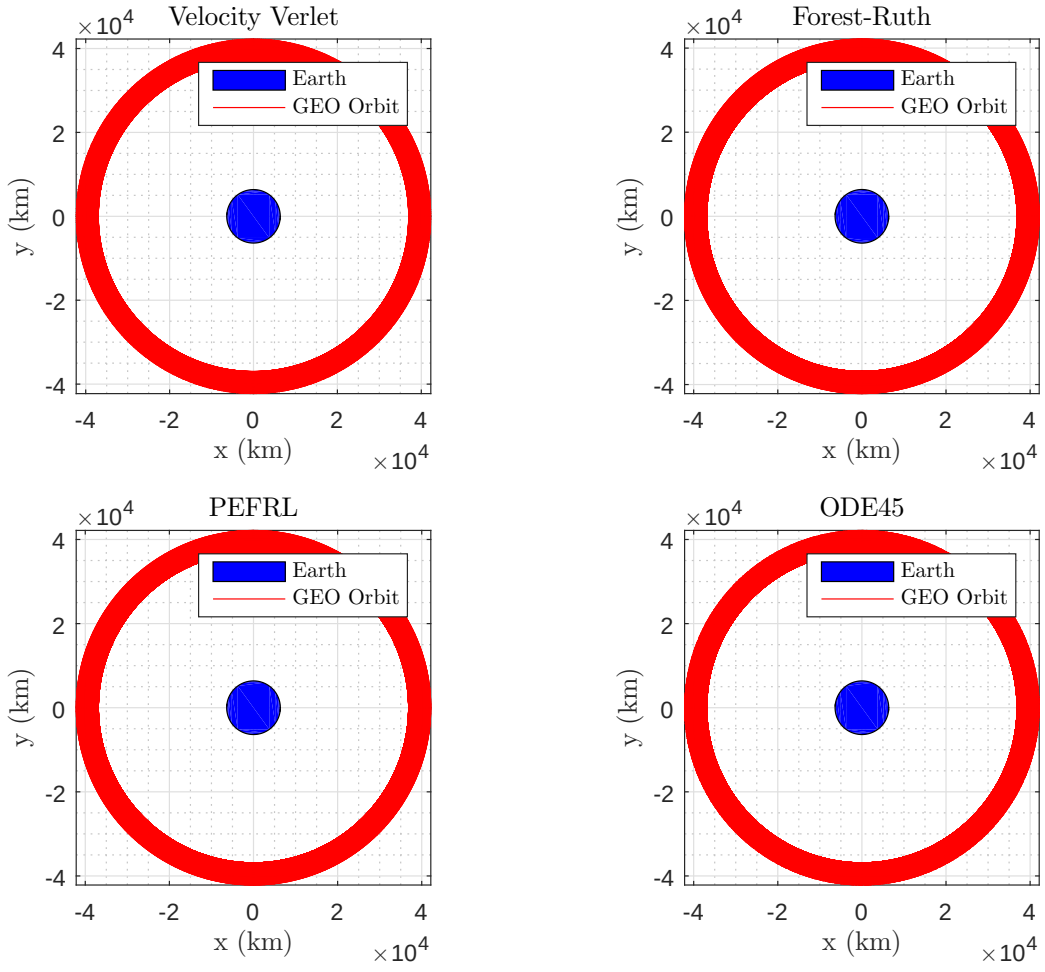

Figure 4.20. GEO Orbit ($\delta t = 1000s$ and time span of 10 years)

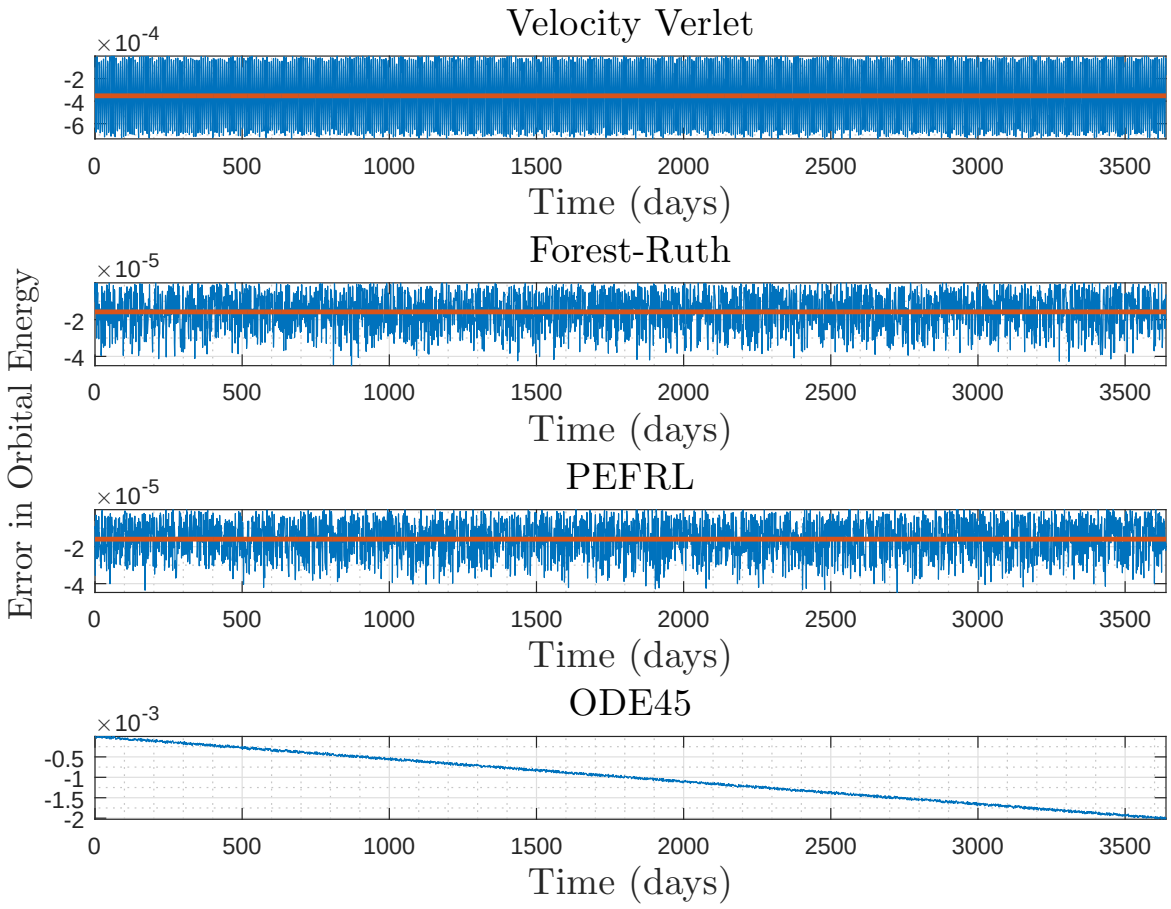

Figure 4.21. Energy Error for a GEO Orbit ($\delta t = 1000s$ and time span of 10 years)

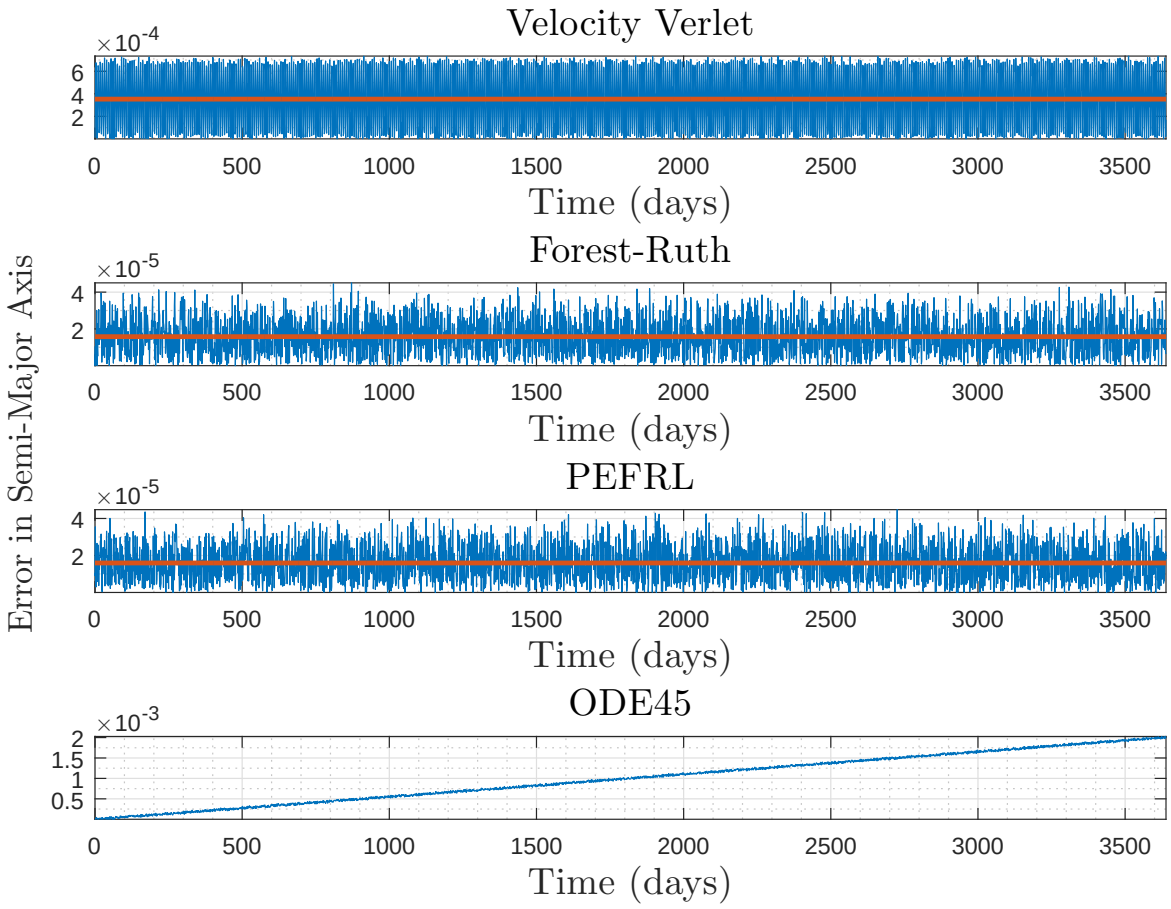

Figure 4.22. Error in Semi-major Axis for a GEO Orbit ($\delta t = 1000s$ and time span of 10 year)

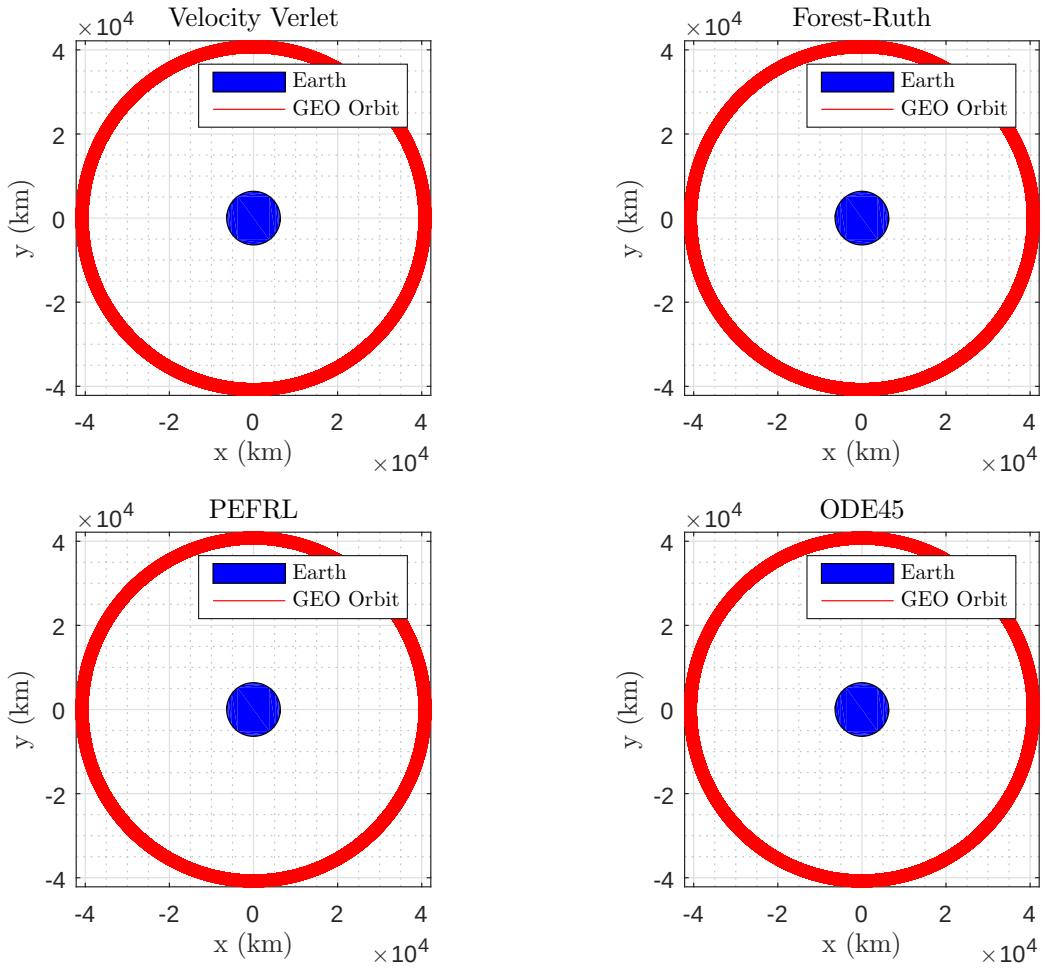

Figure 4.23. GEO Orbit ($\delta t = 100s$ and time span of 10 years)

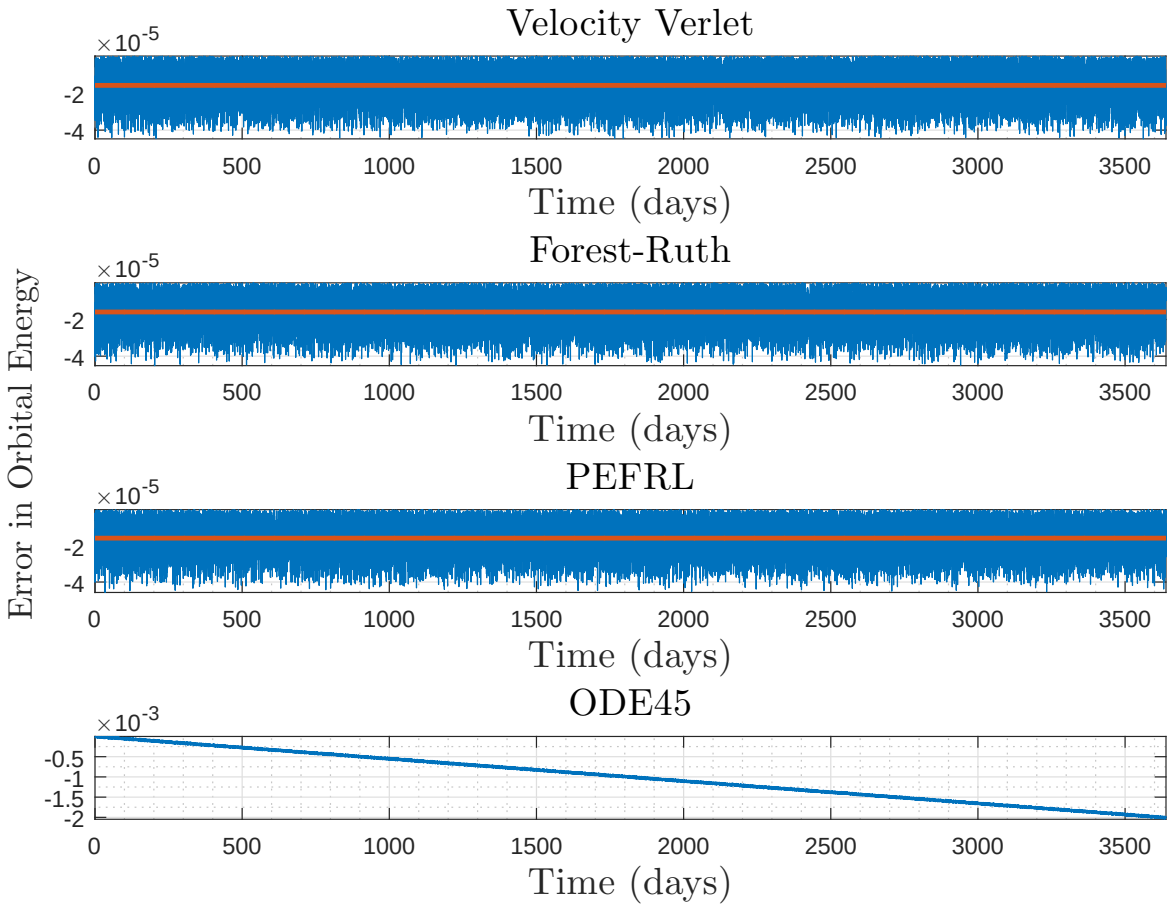

Figure 4.24. Energy Error for a GEO Orbit ($\delta t = 100s$ and time span of 10 years)

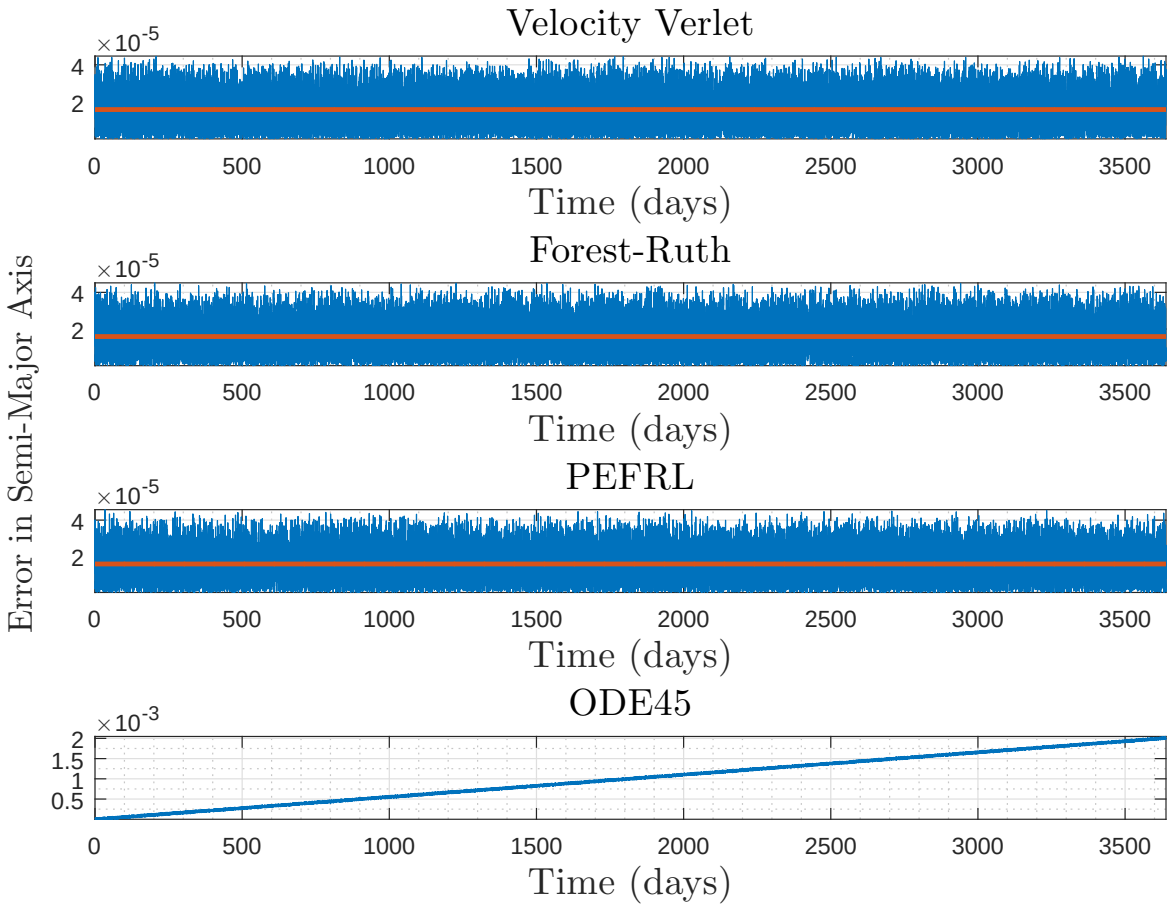

Figure 4.25. Error in Semi-major Axis for a GEO Orbit ($\delta t = 100s$ and time span of 10 year)

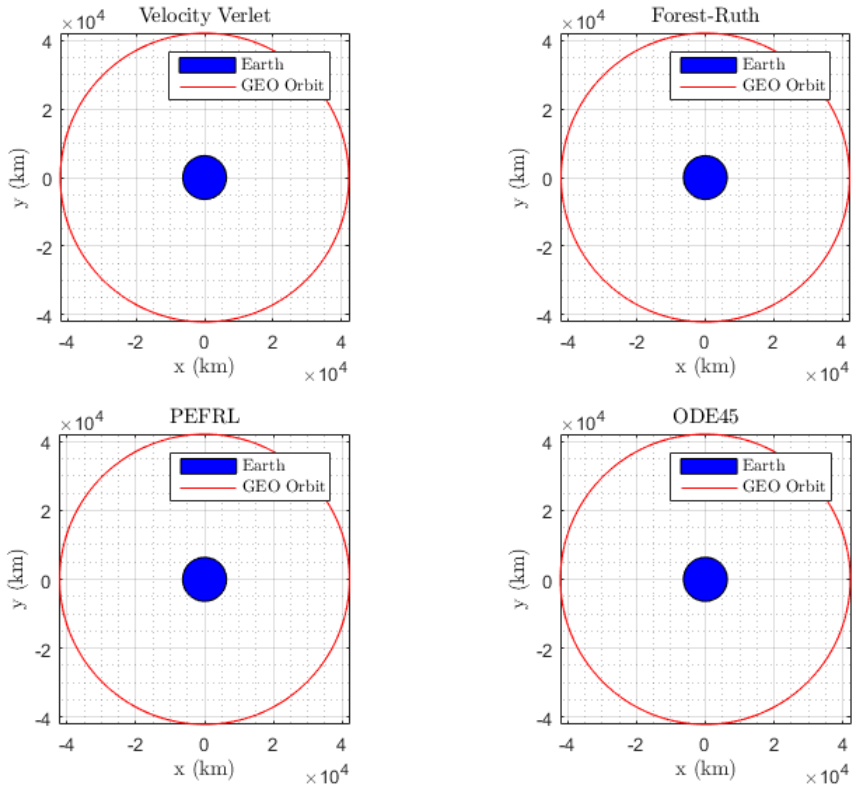

Figure 4.26. GEO Orbit ($\delta t = 10s$ and time span of 10 years)

x

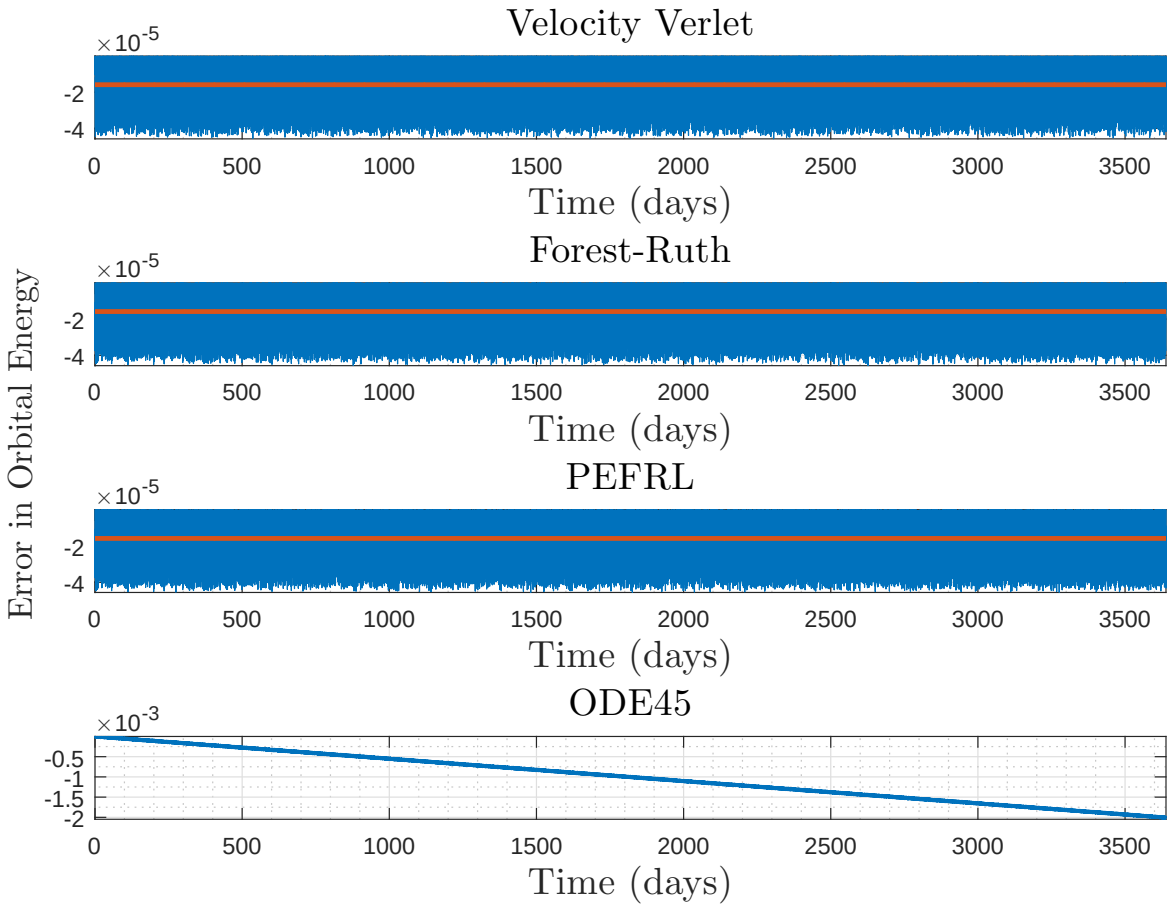

Figure 4.27. Energy Error for a GEO Orbit ($\delta t = 10s$ and time span of 10 years)

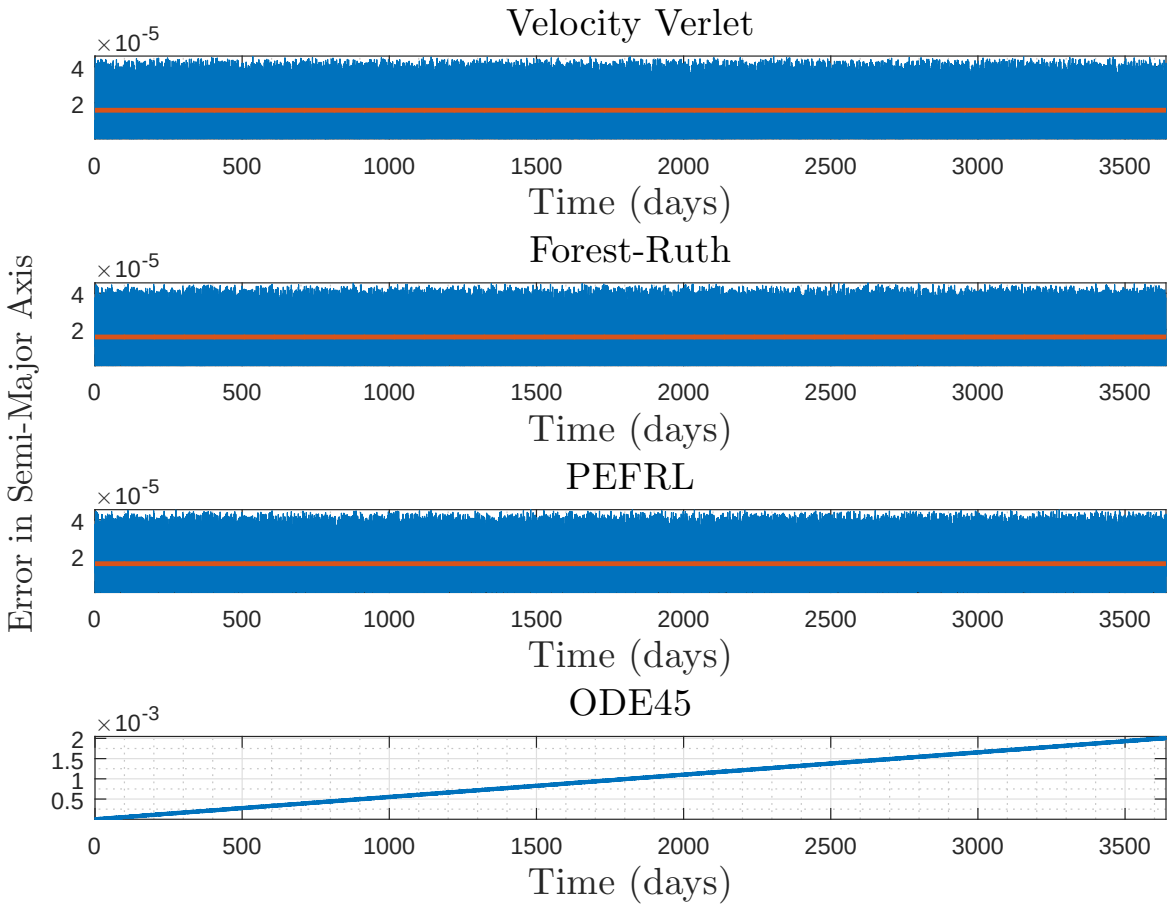

Figure 4.28. Error in Semi-major Axis for a GEO Orbit ($\delta t = 10s$ and time span of 10 year)

The difference between in the semi-major axis accuracy of the ODE45 algorithm compared to that Velocity Verlet that was persistent in the previous series of figures (at a time step of $1000s$) is not present for this time span. The energy error and the semi-major axis errors for the ODE45 have increased an order of magnitude to $10^{-3}\%$ while the symplectic algorithms exhibit the same accuracy as they did for the previous time spans.

At all three time spans and for the three time steps, one persistent problem with ODE45 algorithm is present: the secular error growth. This is evidenced by

the linear growth in the orbital energy error and semi-major axis error. This is evidence to the fact that the Runge-Kutta methods. while accurate in the short term, will result in secular energy growth while the symplectic algorithms maintain accuracy over time because of their energy preserving properties that are depicted in the graphs shown above. This shows the viability of these algorithms for orbit propagation.

The specifications of the system used is shown in below in Table 4.1.

Computer System Specifications	
Model	ASUS-Q550LF
CPU	Intel® Core™ i7-4500U @ 1.80 GHz (Turbo Boost to 2.8GHz)
RAM	8.0 GB
System Type	x86-64 Processor and 64-bit OS
Operating System	Microsoft Windows 10 Home
Matlab Version	R2015b (Academic License)

Table 4.1. Computer Specifications

The next part of this chapter will be a discussion of the efficiency of the algorithms. While the symplectic algorithms have been demonstrated to be more accurate, their efficiency is yet to be explained. The following tables (Table 4.2-Table 4.4) shows the execution time the various time spans at various time steps for all of the algorithms of study in this thesis. The following three tables show the computational time for each of the algorithms at the three times steps and time spans. (Note: To generate the data shown below, each case was performed five times and the execution times were recorded using MATLAB's timing feature. The

data shown are the average of those five tests). Tables 4.2 - 4.4 clearly demonstrate the advantages of the symplectic algorithms over the Runge-Kutta method in terms of computational efficiency. This can be attributed to the error control features that are fund in ODE45 which can increase the number of function evaluations if the error is detected to be above accepted tolerance levels.

Algorithm	**Time Step** (*seconds*)	**Execution Time** (*seconds*)
Velocity Verlet	1000	0.465
Forest-Ruth	1000	0.242
PEFRL	1000	0.287
ODE45	1000	3.123
Velocity Verlet	100	4.871
Forest-Ruth	100	2.456
PEFRL	100	2.879
ODE45	100	6.271
Velocity Verlet	10	49.701
Forest-Ruth	10	26.127
PEFRL	10	31.635
ODE45	10	39.396

Table 4.2. Execution Time of the Algorithms for Propagation of 1 year

Algorithm	**Time Step** (*seconds*)	**Execution Time** (*seconds*)
Velocity Verlet	1000	2.268
Forest-Ruth	1000	1.332
PEFRL	1000	1.348
ODE45	1000	15.252
Velocity Verlet	100	22.319
Forest-Ruth	100	13.623
PEFRL	100	13.617
ODE45	100	30.357
Velocity Verlet	10	277.154
Forest-Ruth	10	155.782
PEFRL	10	169.865
ODE45	10	211.407

Table 4.3. Execution Time of the Algorithms for Propagation of 5 years

Algorithm	**Time Step** (*seconds*)	**Execution Time** (*seconds*)
Velocity Verlet	1000	5.864
Forest-Ruth	1000	3.120
PEFRL	1000	3.277
ODE45	1000	37.619
Velocity Verlet	100	51.253
Forest-Ruth	100	29.881
PEFRL	100	32.124
ODE45	100	73.566
Velocity Verlet	10	480.379
Forest-Ruth	10	246.646
PEFRL	10	319.581
ODE45	10	448.127

Table 4.4. Execution Time of the Algorithms for Propagation of 10 years

The following graph in Figure 4.29 show the data represented in the Tables 4.2 - 4.4 above on a semi-log plot.

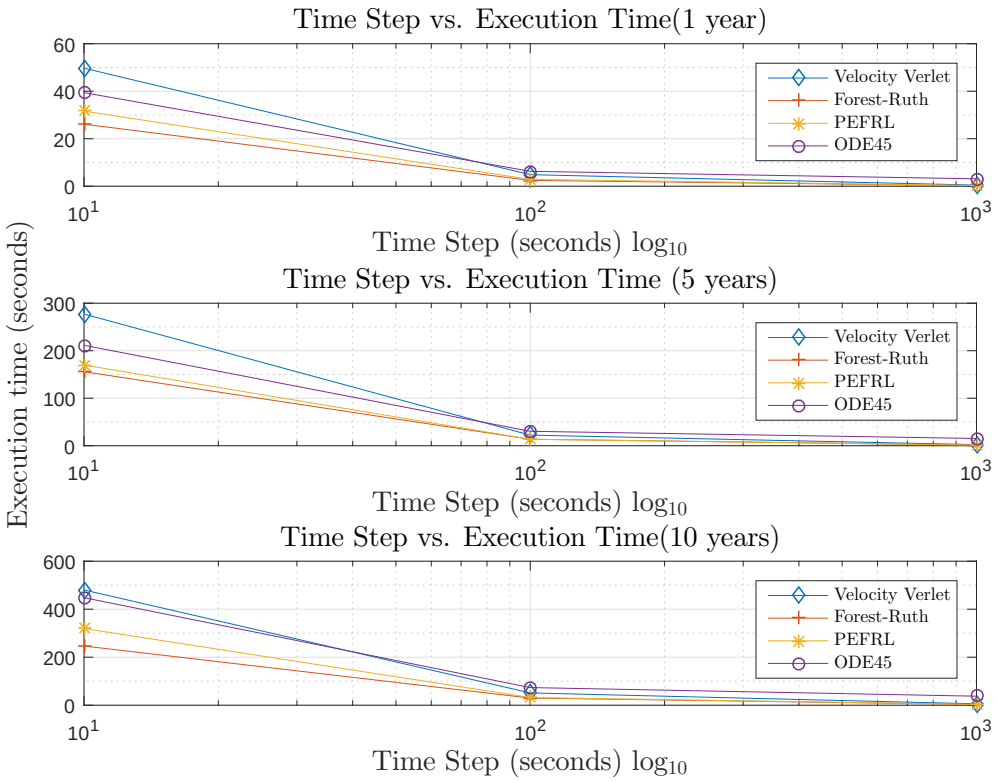

Figure 4.29. Semi-log Plot of the Execution Times

The graphs in Figure 4.29 ndicates that the all of the algorithms shown above behave similarly to change in the step size. However, as the previous tables indicate, the symplectic algorithms are more efficient at all three time steps. Although, there seems to be a diminishing returns on decreasing the step size as the scaling of the execution times is not close to being linear.

The next section of this chapter outlines the results performed for the perturbed problem which was described in Chapter 2.

4.2 Orbit Propagation Results

In the following series of graphs, the orbit propagation results using the perturbed two-body dynamics outlined in Chapter 2. The third-body perturbations from the Sun and Moon as well as geopotential perturbations (a field of 5×5) are used.

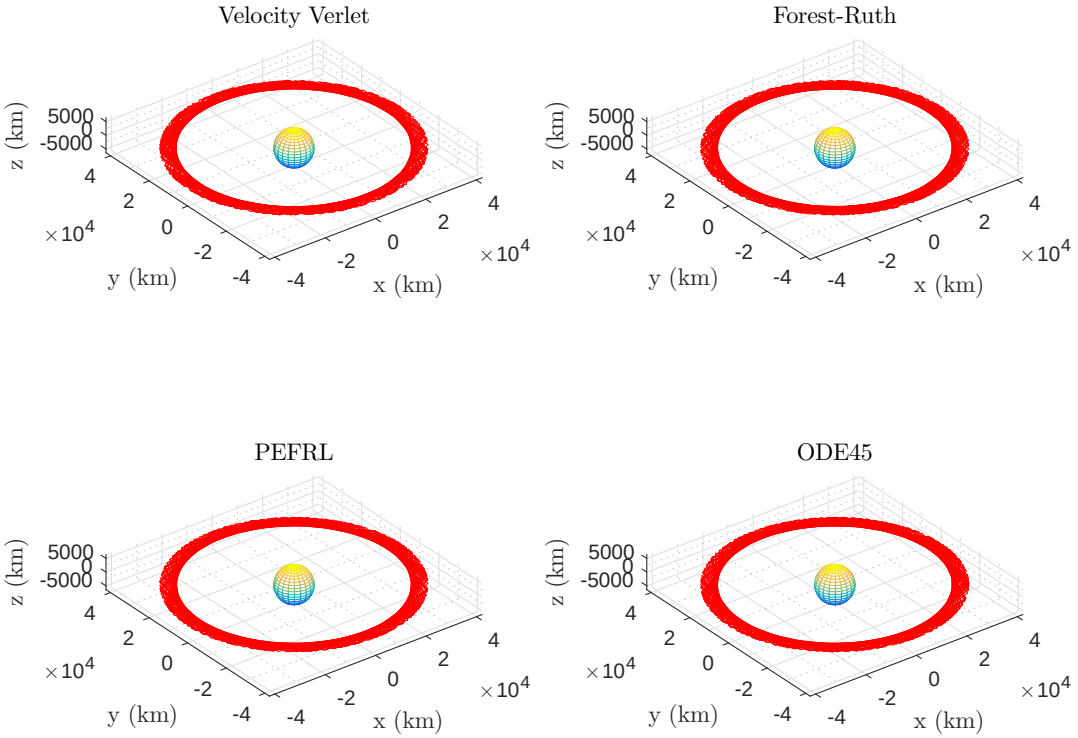

Figure 4.30. Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 1 year)

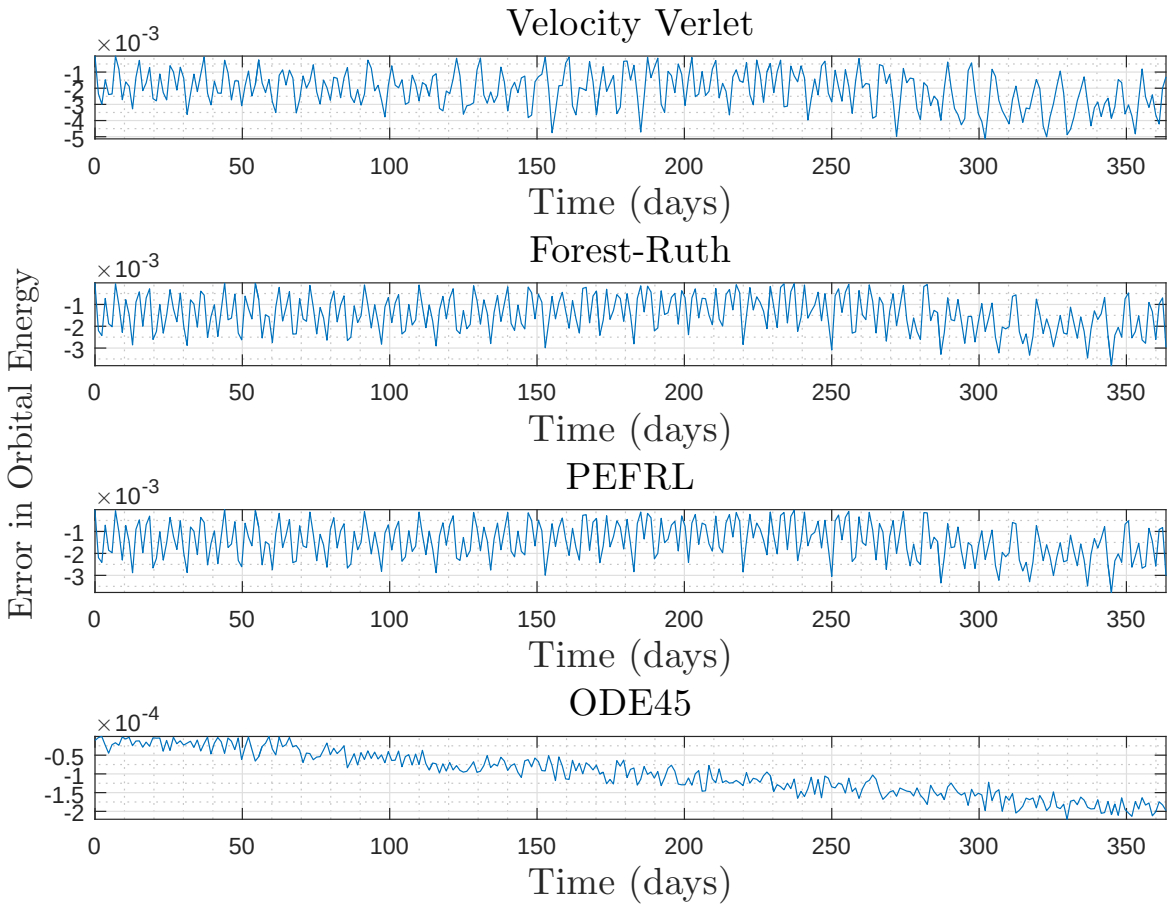

Figure 4.31. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 1 year)

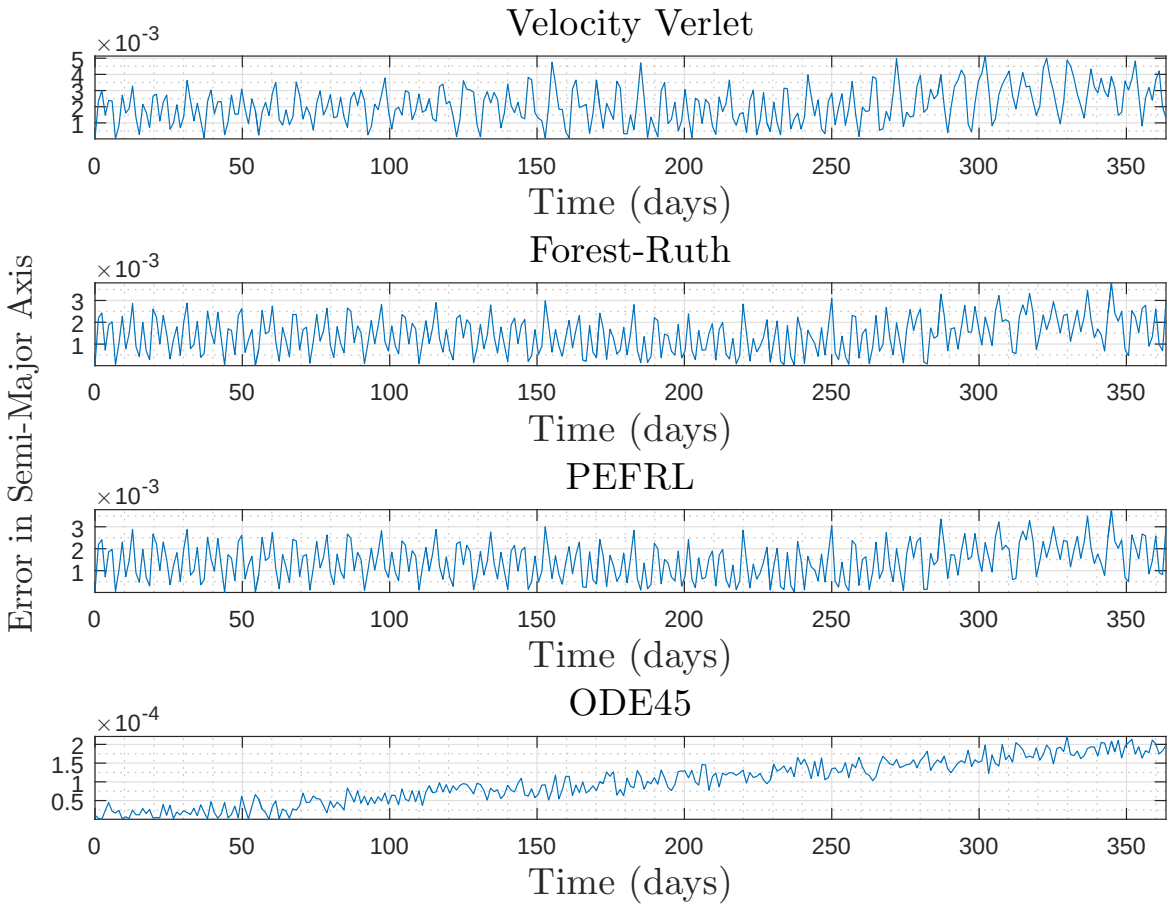

Figure 4.32. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 1 year)

The graphs in Figures 4.30 - 4.32 demonstrate the difference between the symplectic and the Runge-Kutta method. While the symplectic algorithms deviate (energy and semi-major) around 10^{-3}%, ODE45 deviates around 10^{-4}%. The symplectic algorithms are more accurate to the deviations that are induced due to the included perturbations. The next three graphs in Figures 4.33 - 4.35 display the orbit, error in the orbital energy, and semi-major axis respectively for a time span of 1 year and time step of 100 seconds.

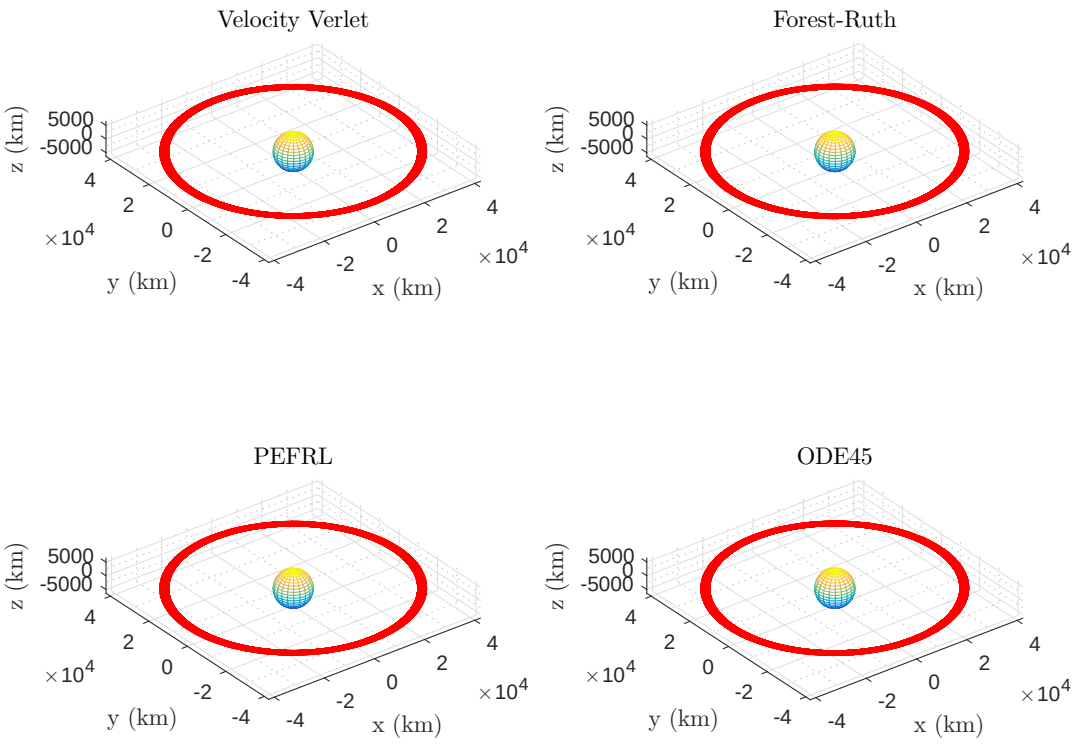

Figure 4.33. Perturbed GEO Orbit ($\delta t = 100s$ and time span of 1 year)

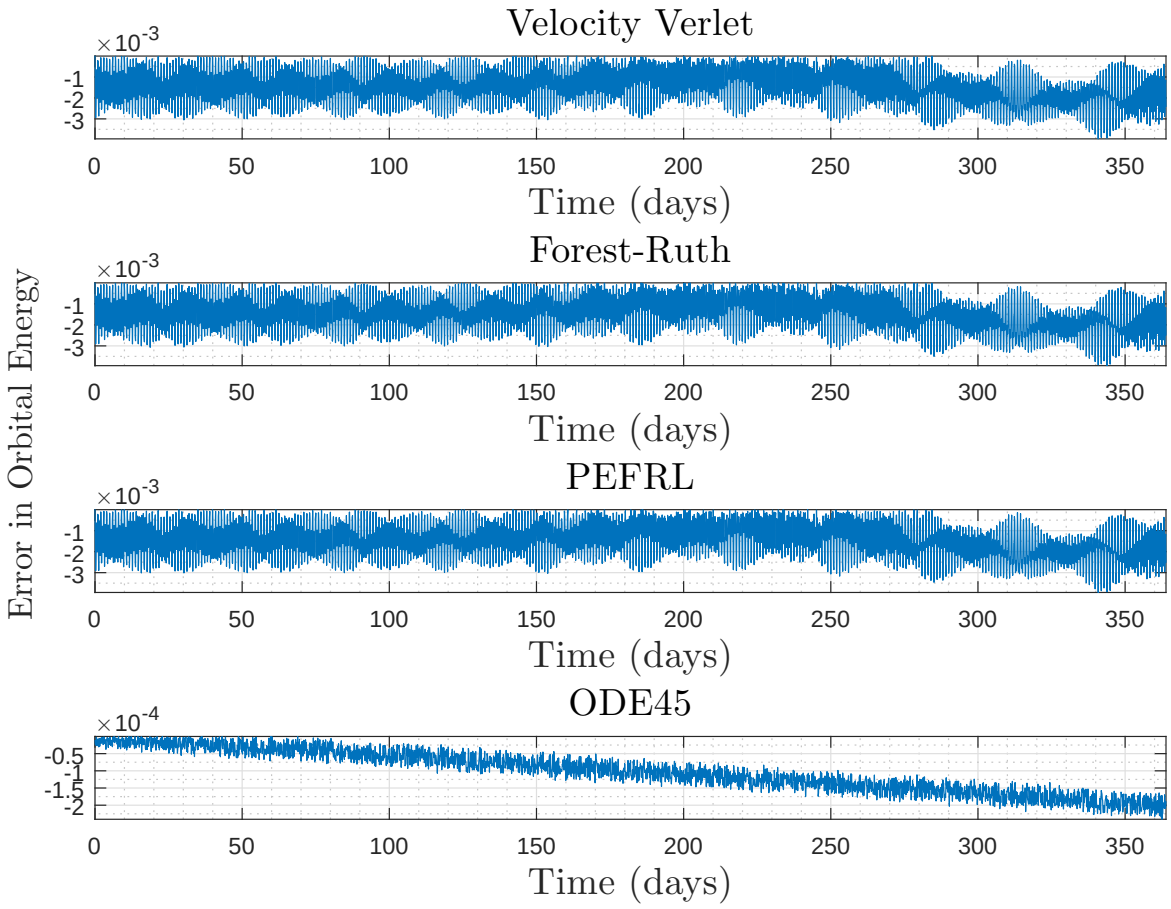

Figure 4.34. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 100s$ and time span of 1 year)

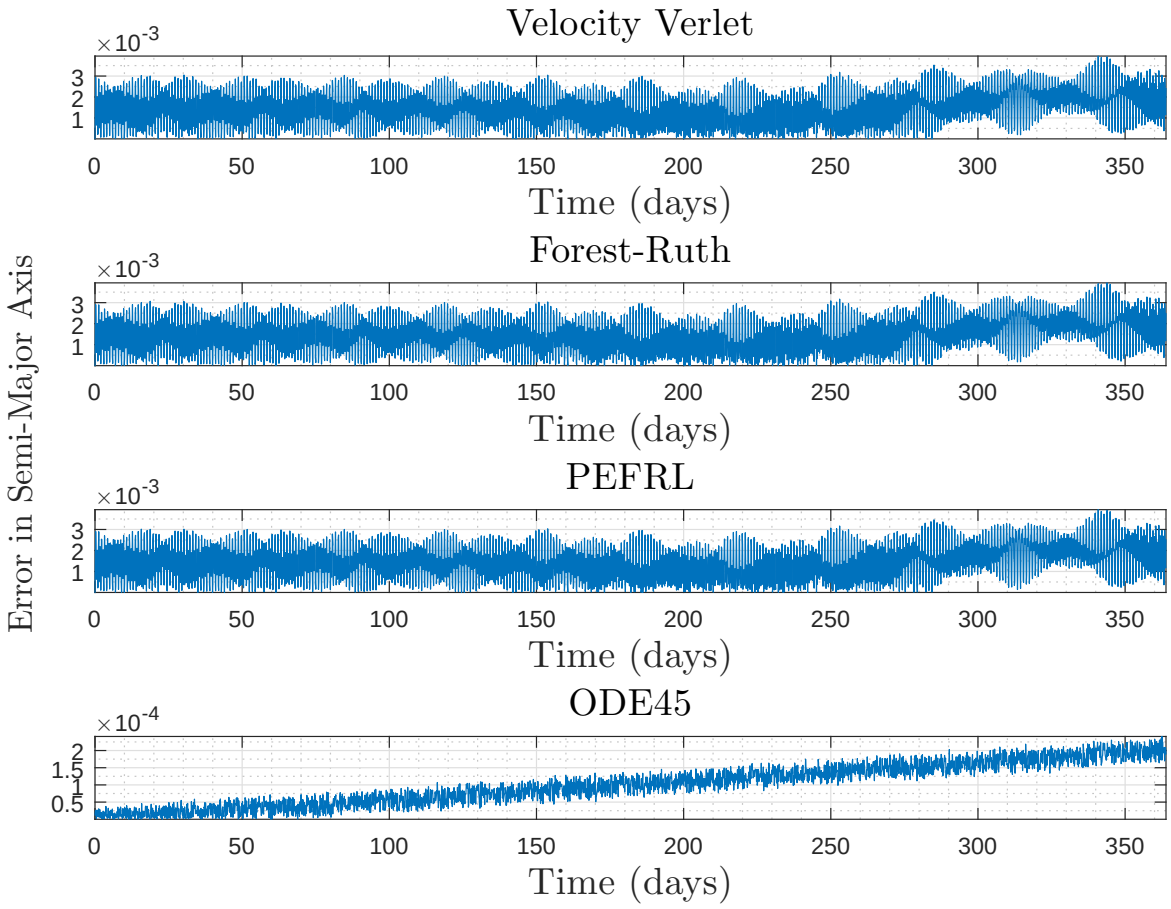

Figure 4.35. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 100s$ and time span of 1 year)

In the graphs in Figures 4.36 - 4.38 in the following figures, the time span is 1 year and time step is 10 seconds.

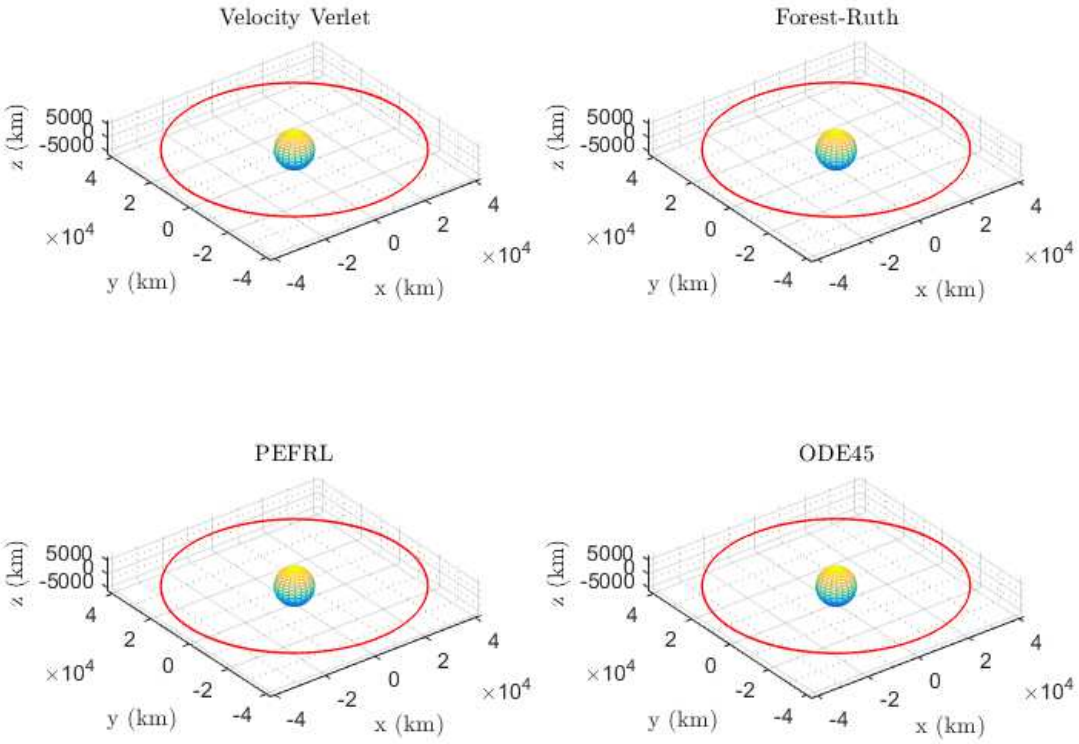

Figure 4.36. Perturbed GEO Orbit ($\delta t = 10s$ and time span of 1 year)

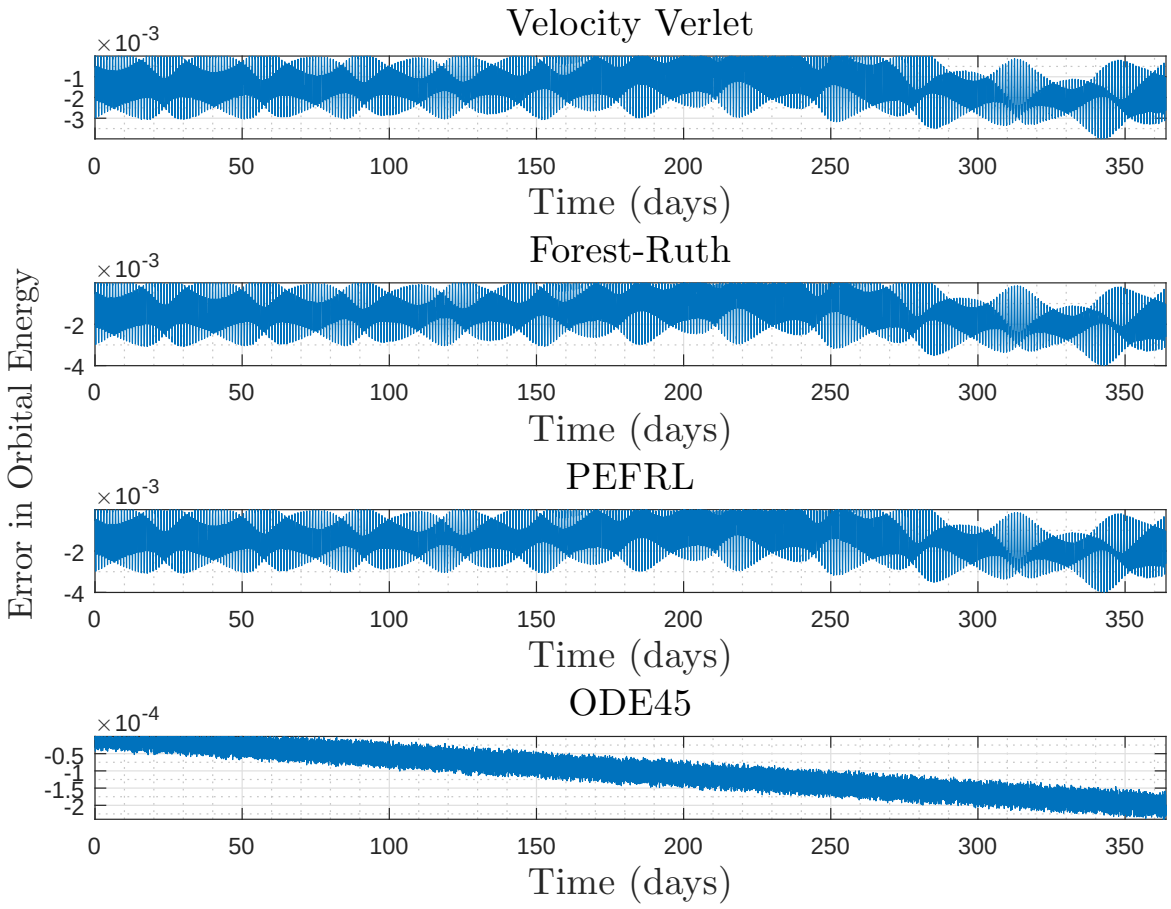

Figure 4.37. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 10s$ and time span of 1 year)

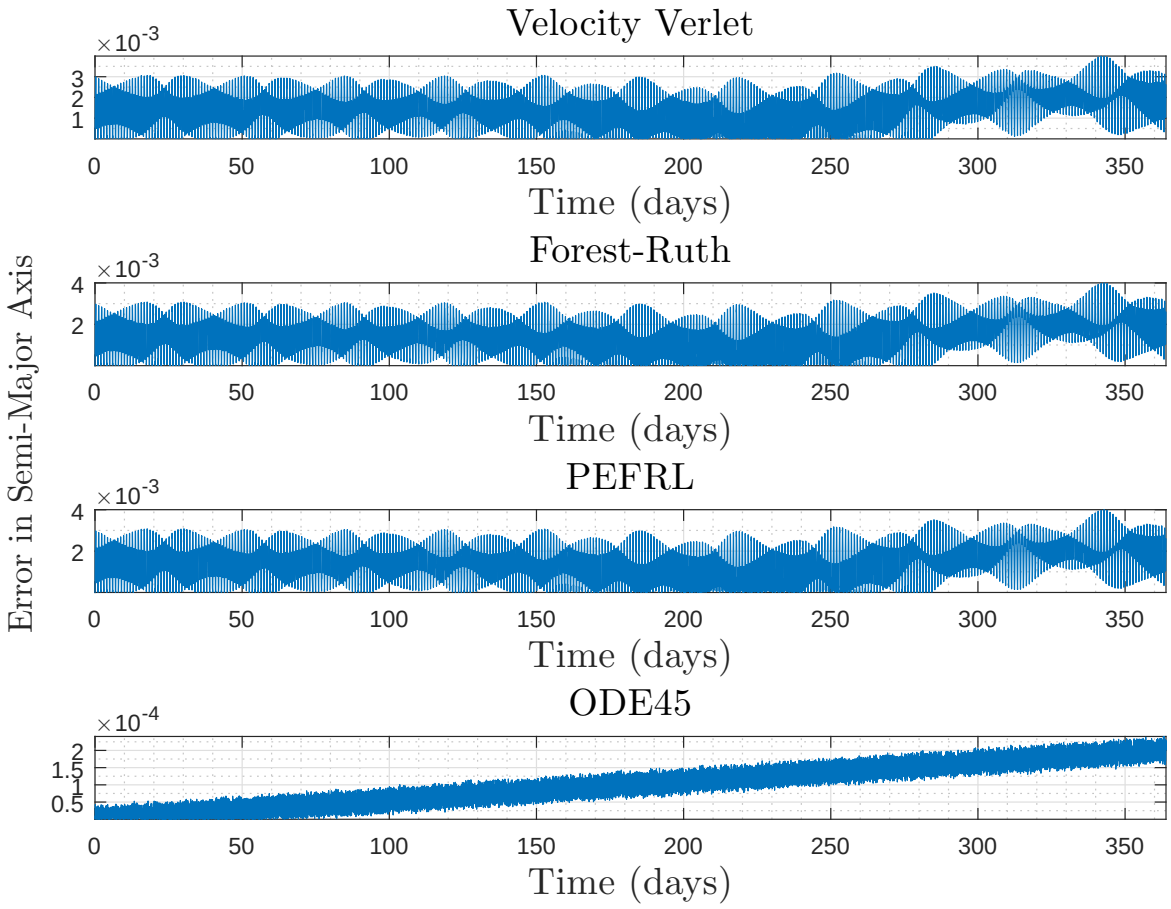

Figure 4.38. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 10s$ and time span of 1 year)

With a smaller time step, more oscillatory modes in the energy and semi-major axis deviations are visible. These features are more apparent for the symplectic algorithms than the ODE45 algorithm. Another source of evidence that the symplectic algorithms perform more accurately than the Runge-Kutta methods.

With the perturbations from non-spherical potential and third body forces, the invariant measures are expected to change. However, over a span of a year, the change is expected to minimal. While the three symplectic and ODE45 algorithms show some change, there is a difference in the growth of the energy and semi-major

axis between them. As presented in the previous section of this chapter, the next series of figures present the orbits, orbital energy, and semi-major axis for a time span of 5 and 10 years (all three time increments are shown).

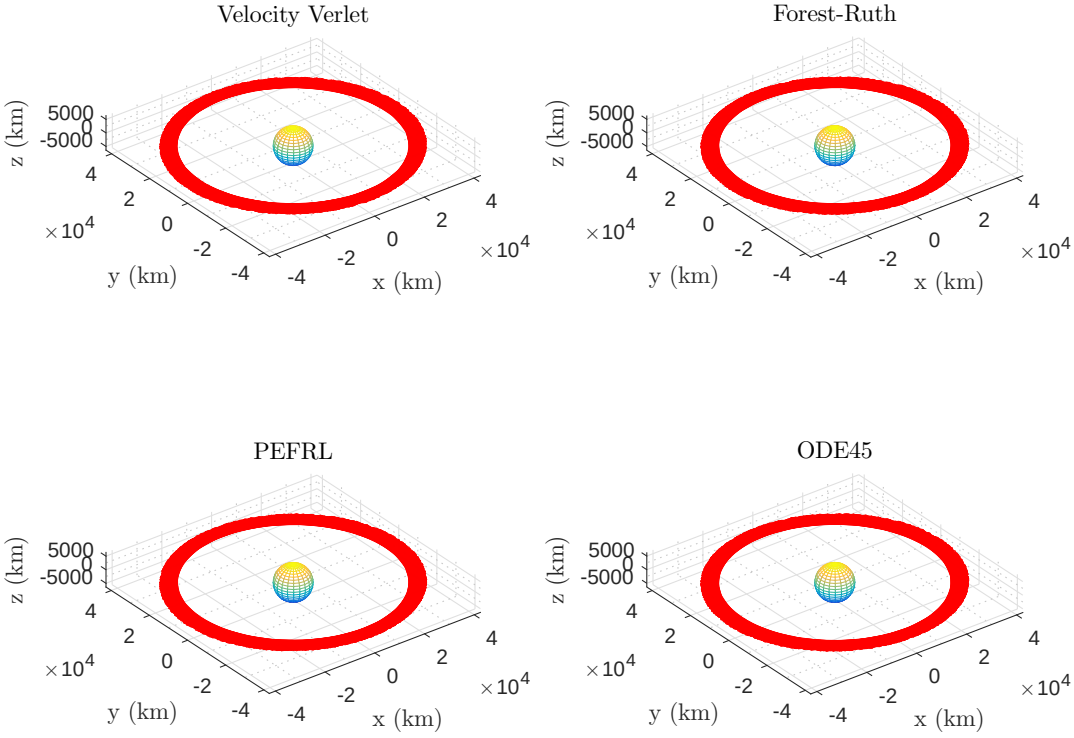

Figure 4.39. Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 5 year)

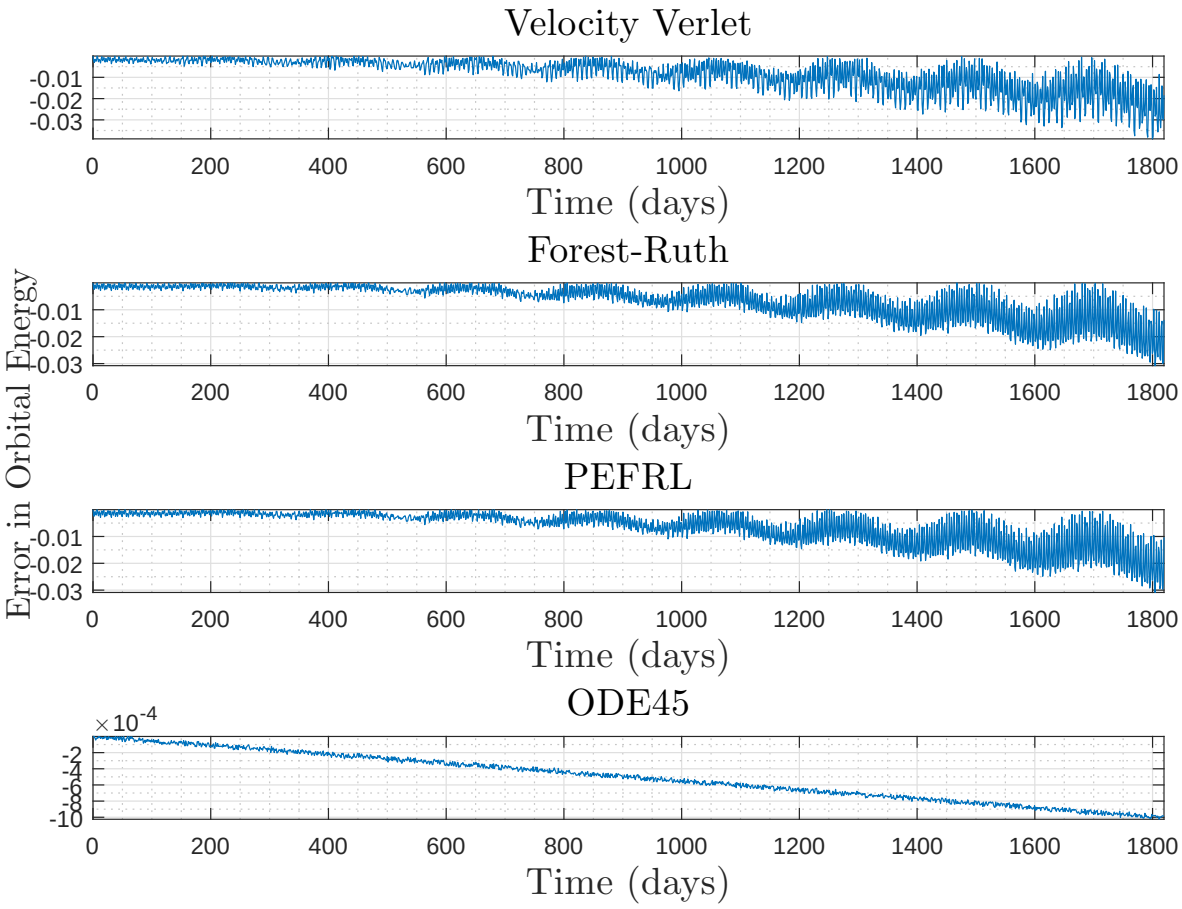

Figure 4.40. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 5 year)

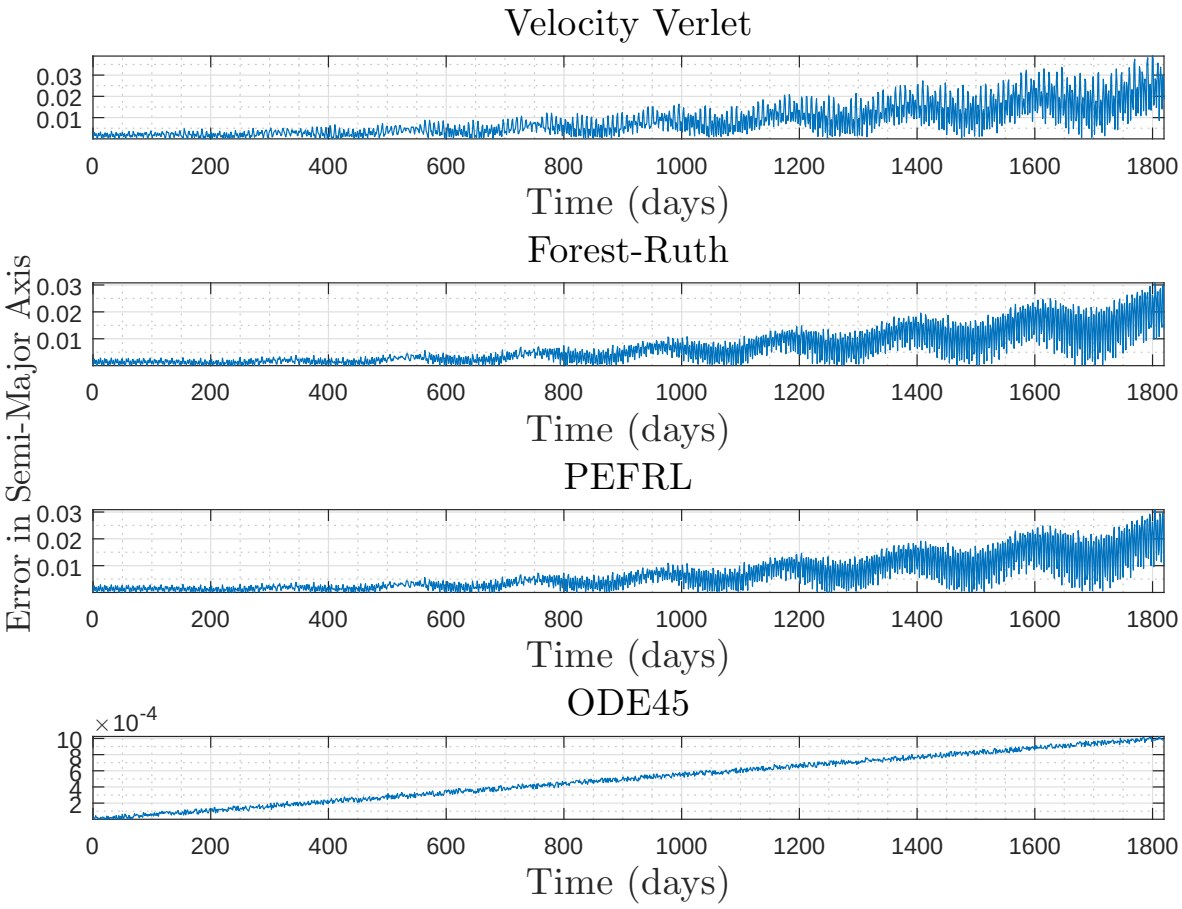

Figure 4.41. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 5 year)

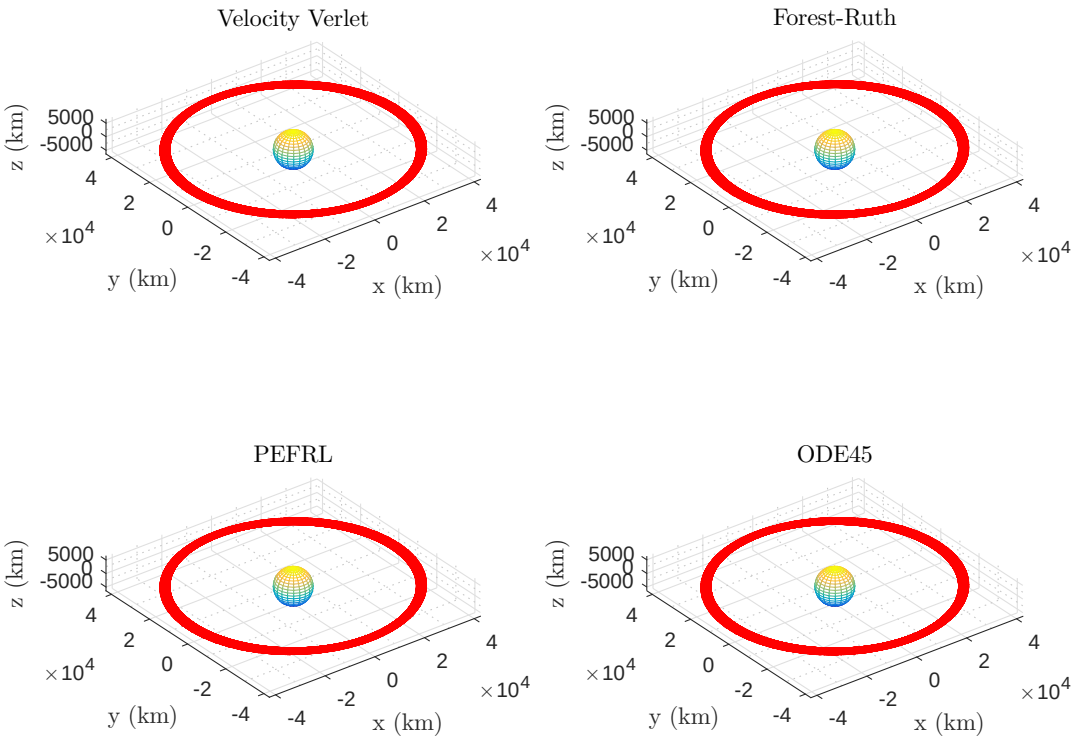

Figure 4.42. Perturbed GEO Orbit ($\delta t = 100s$ and time span of 5 year)

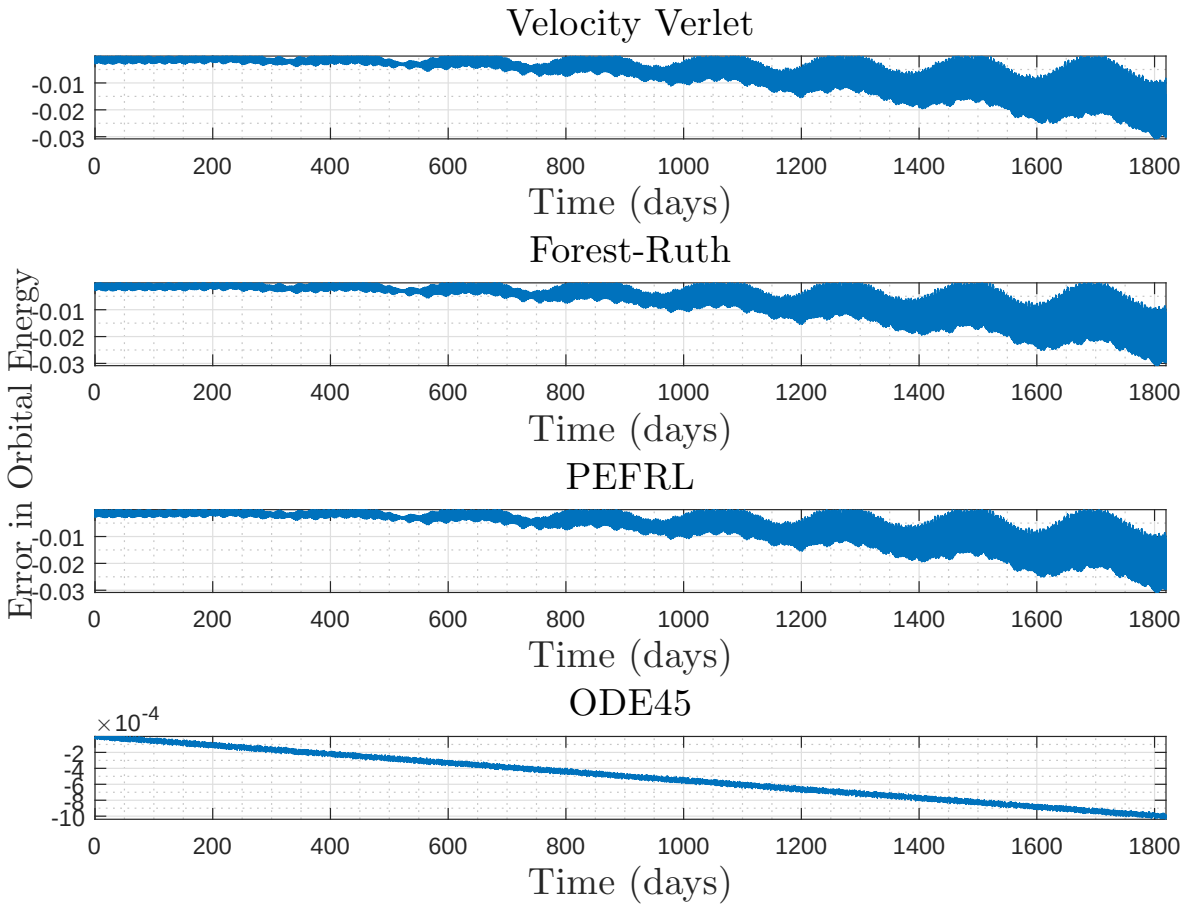

Figure 4.43. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 100s$ and time span of 5 year)

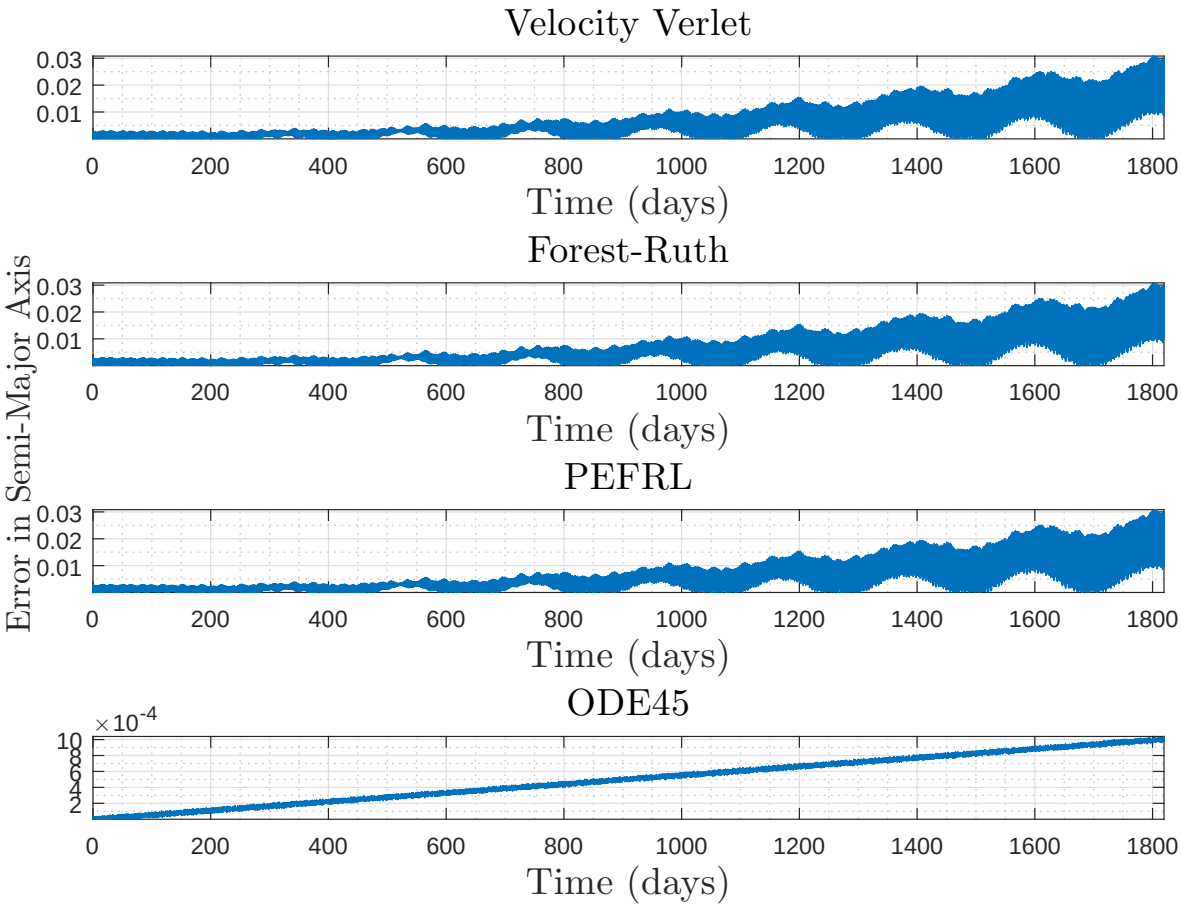

Figure 4.44. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 100s$ and time span of 5 year)

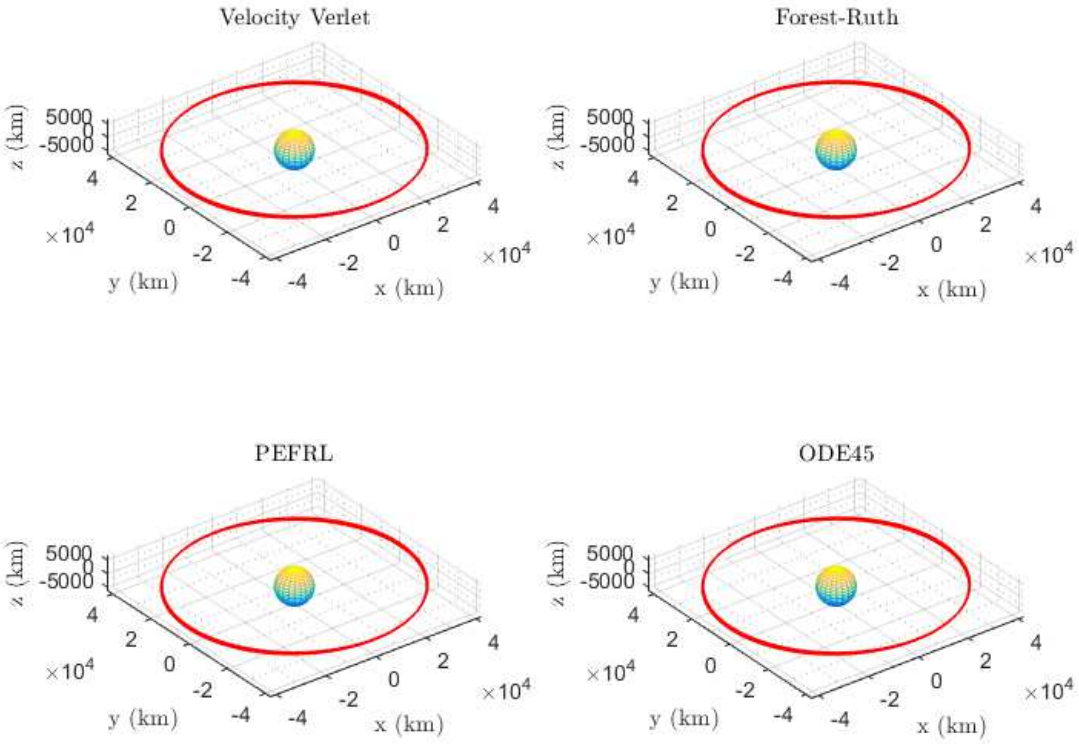

Figure 4.45. Perturbed GEO Orbit ($\delta t = 10s$ and time span of 5 year)

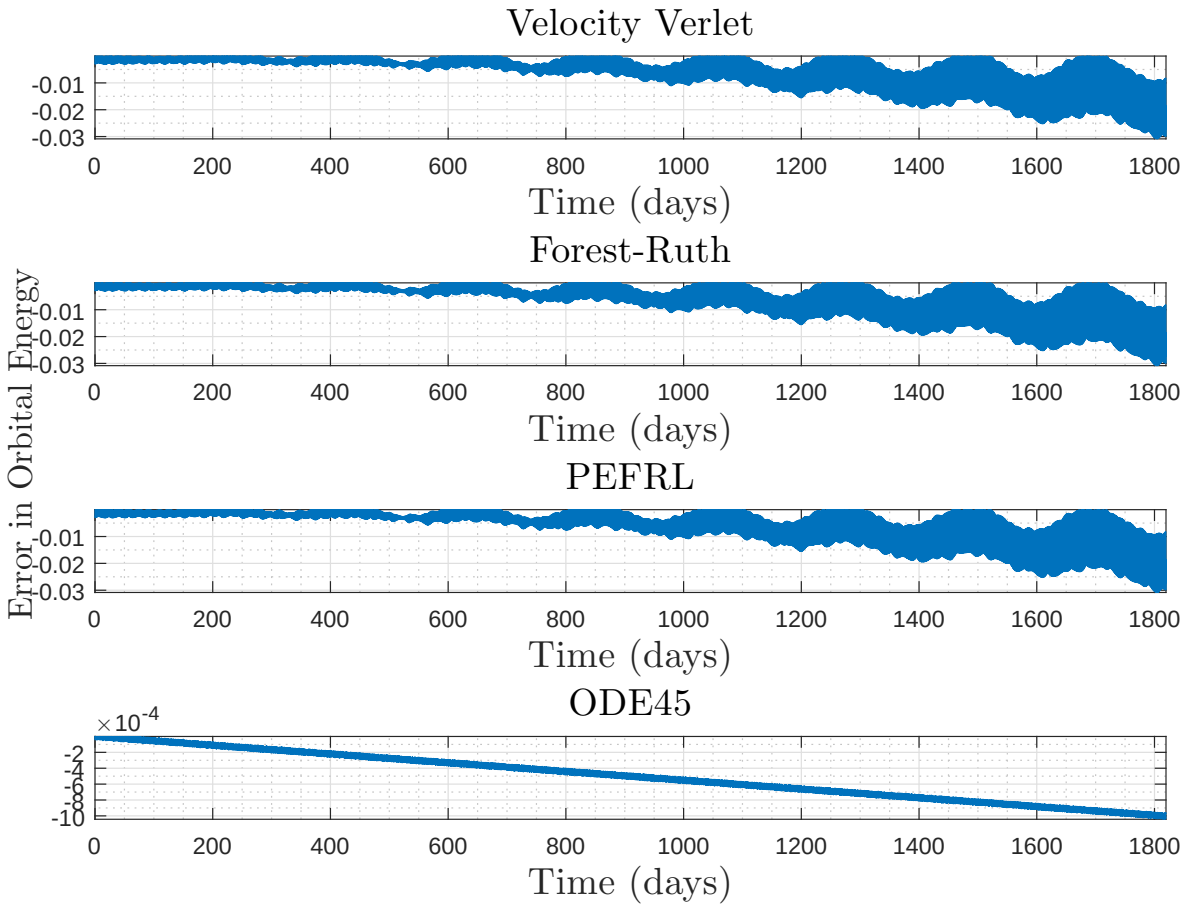

Figure 4.46. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 10s$ and time span of 5 year)

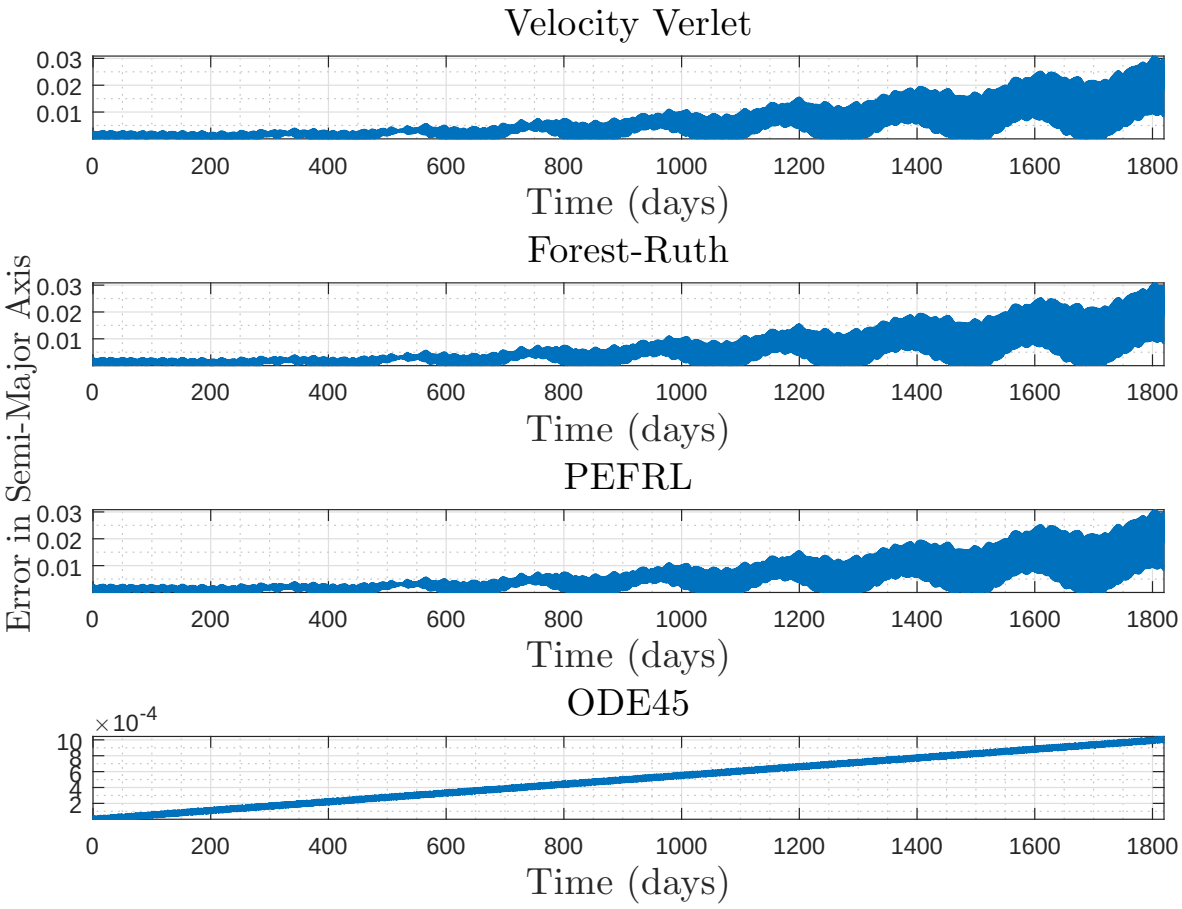

Figure 4.47. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 10s$ and time span of 5 year)

Just like the last series of figures for a time span of 1 year, the results for this time span (5 years) seems to show that the step size of integration has relatively little affect on the performance of the algorithms. Also, the oscillatory changes in the invariant quantities are much more apparent for smaller time steps. In addition, the three symplectic algorithms seem to evolve in the same manner which indicates that the higher order accuracy of Forest-Ruth and PEFRL algorithms are not significant. Perhaps, over a 10 year span, some differences can arise.

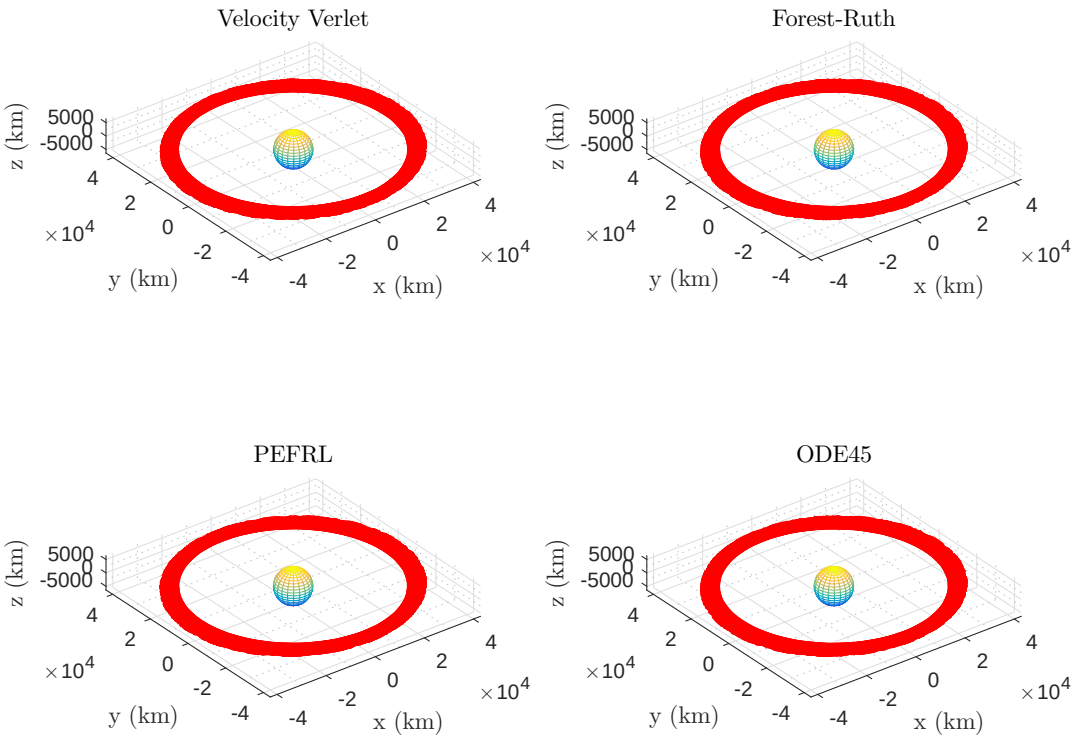

Figure 4.48. Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 10 year)

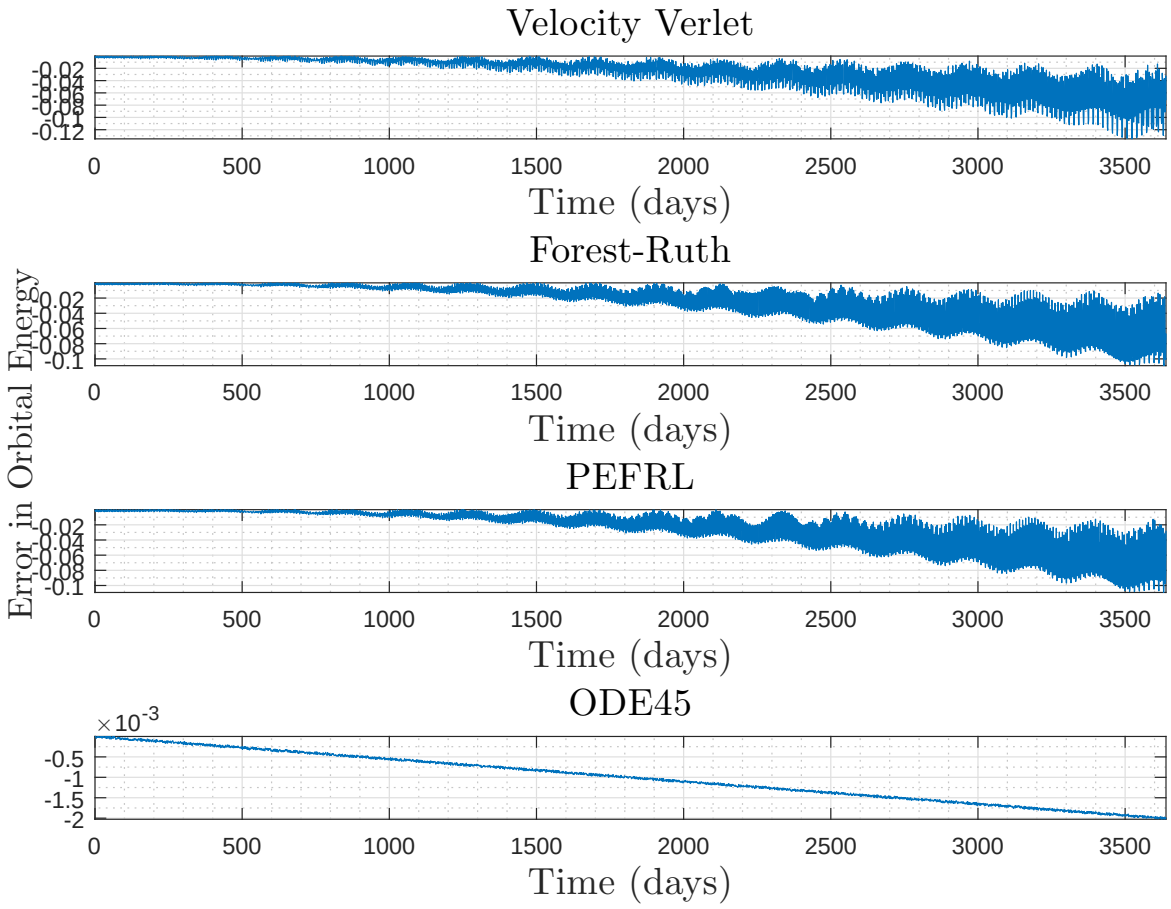

Figure 4.49. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 10 year)

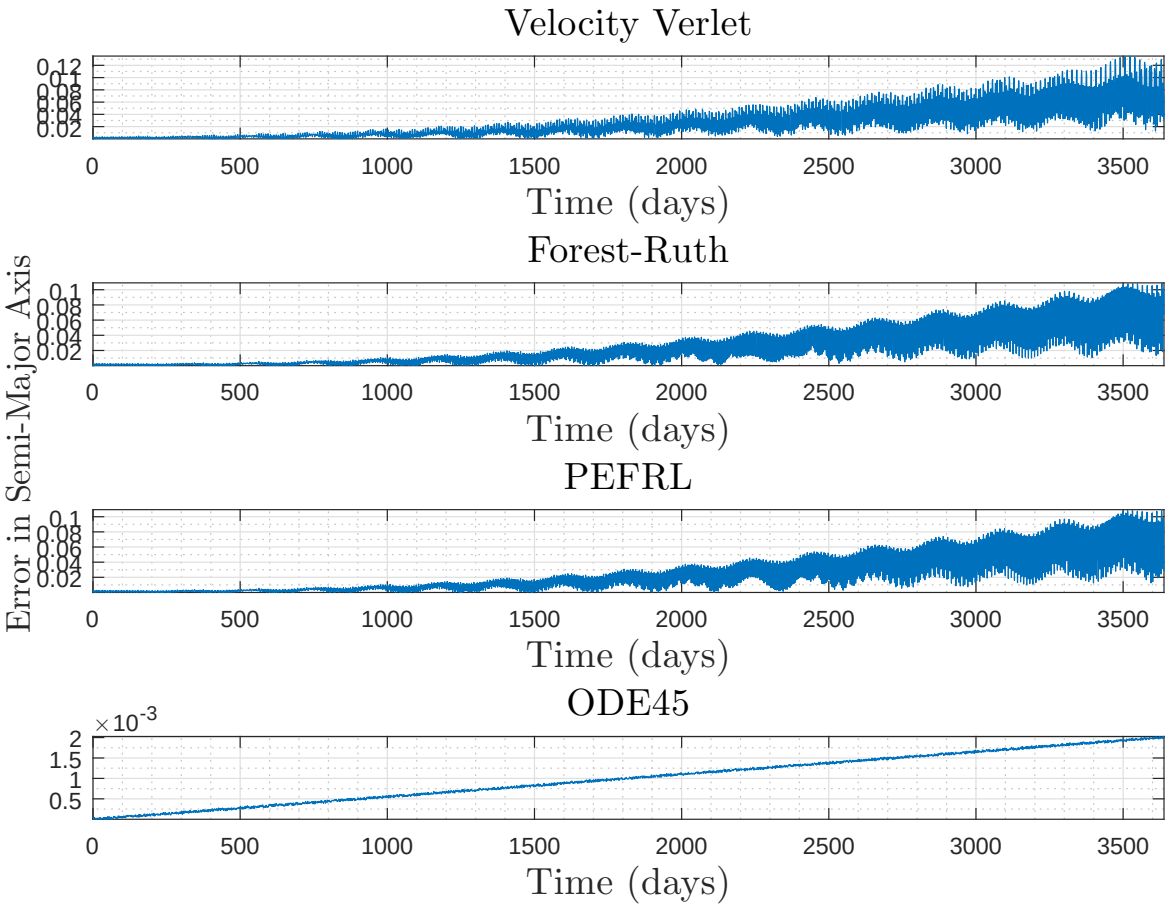

Figure 4.50. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 1000s$ and time span of 10 year)

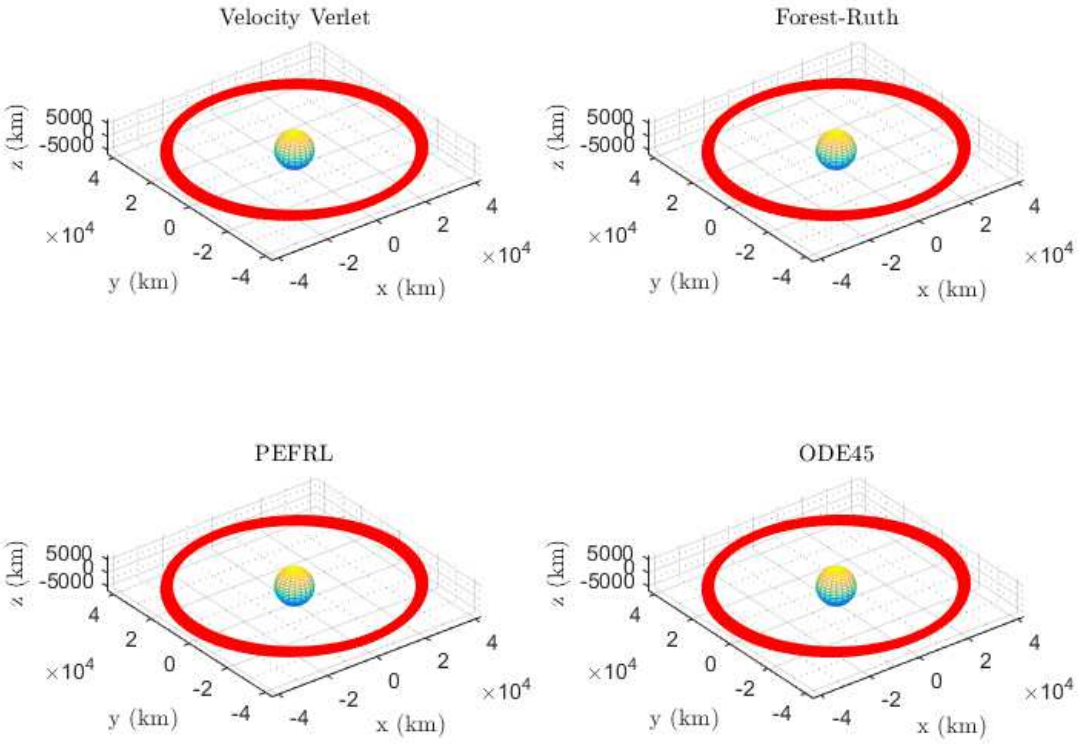

Figure 4.51. Perturbed GEO Orbit ($\delta t = 100s$ and time span of 10 year)

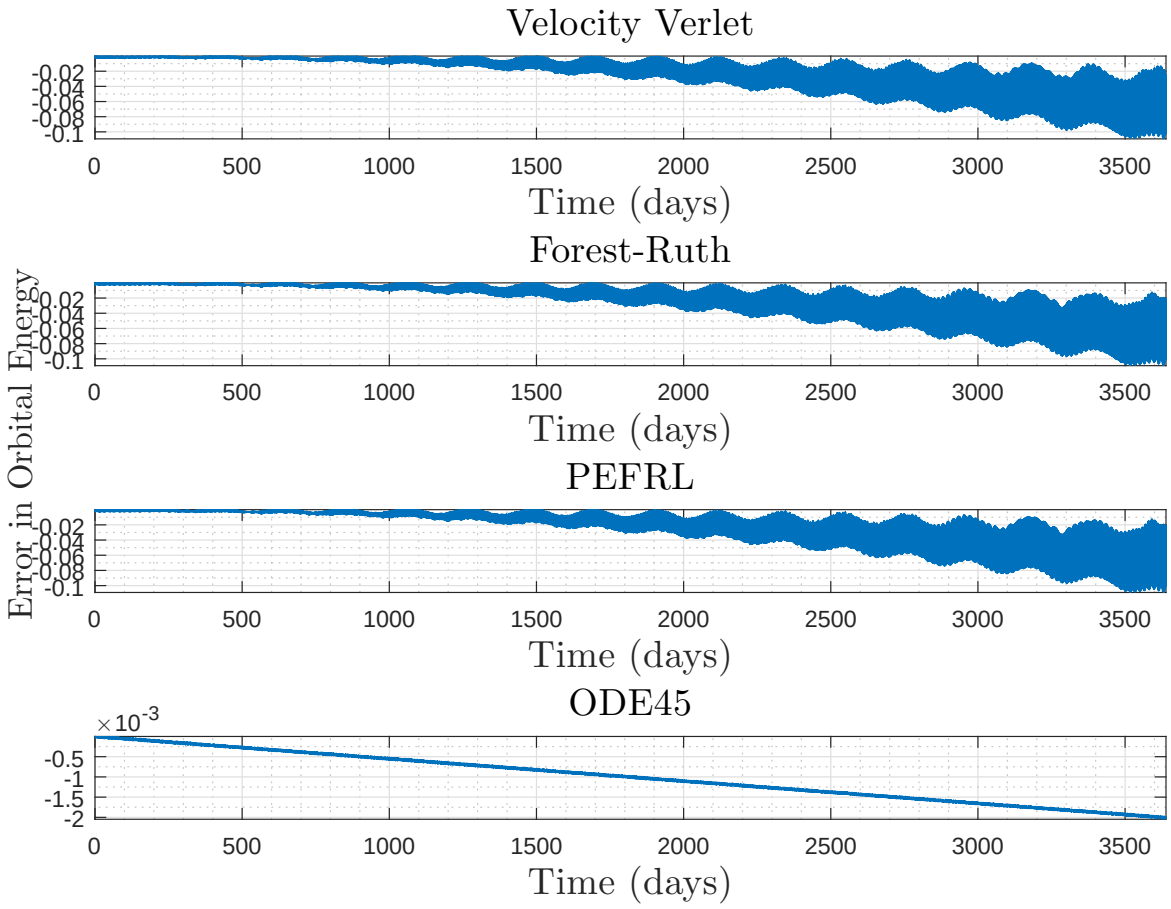

Figure 4.52. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 100s$ and time span of 10 year)

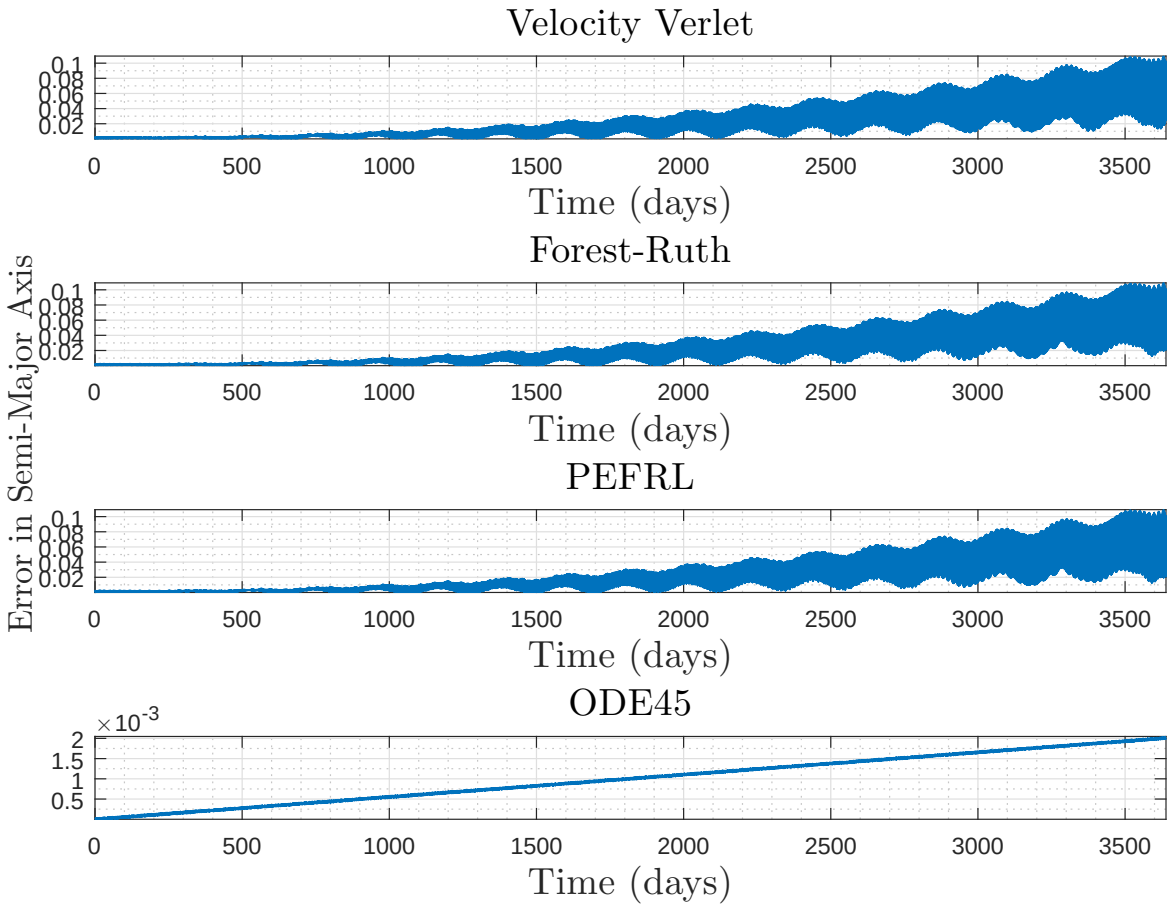

Figure 4.53. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 100s$ and time span of 10 year)

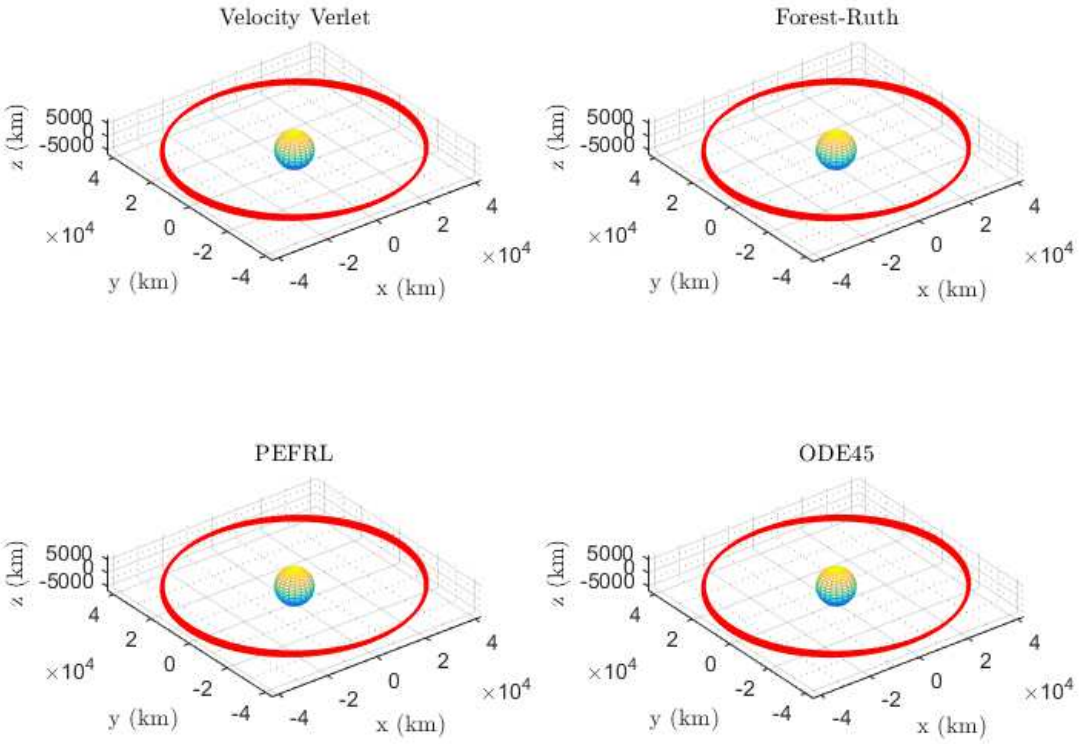

Figure 4.54. Perturbed GEO Orbit ($\delta t = 10s$ and time span of 10 year)

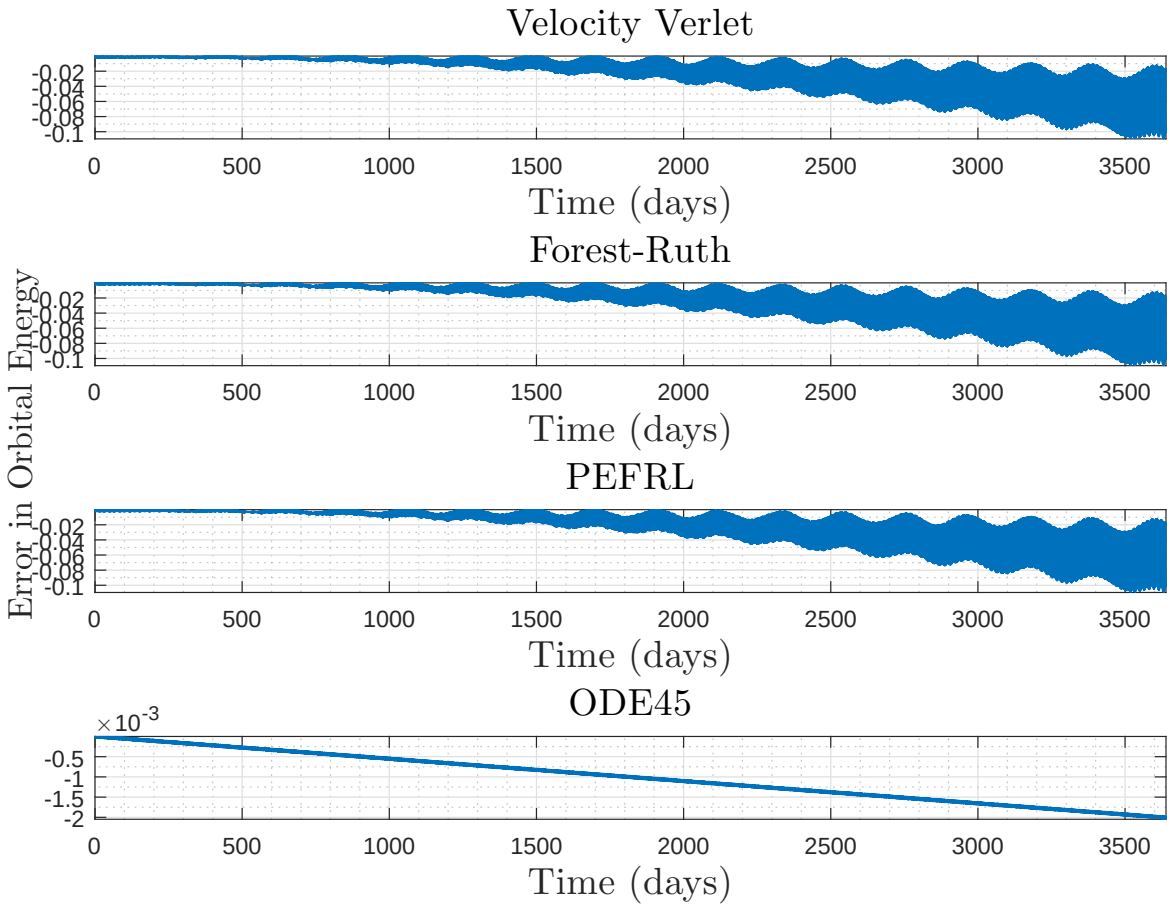

Figure 4.55. Energy Deviation(%) for a Perturbed GEO Orbit ($\delta t = 10s$ and time span of 10 year)

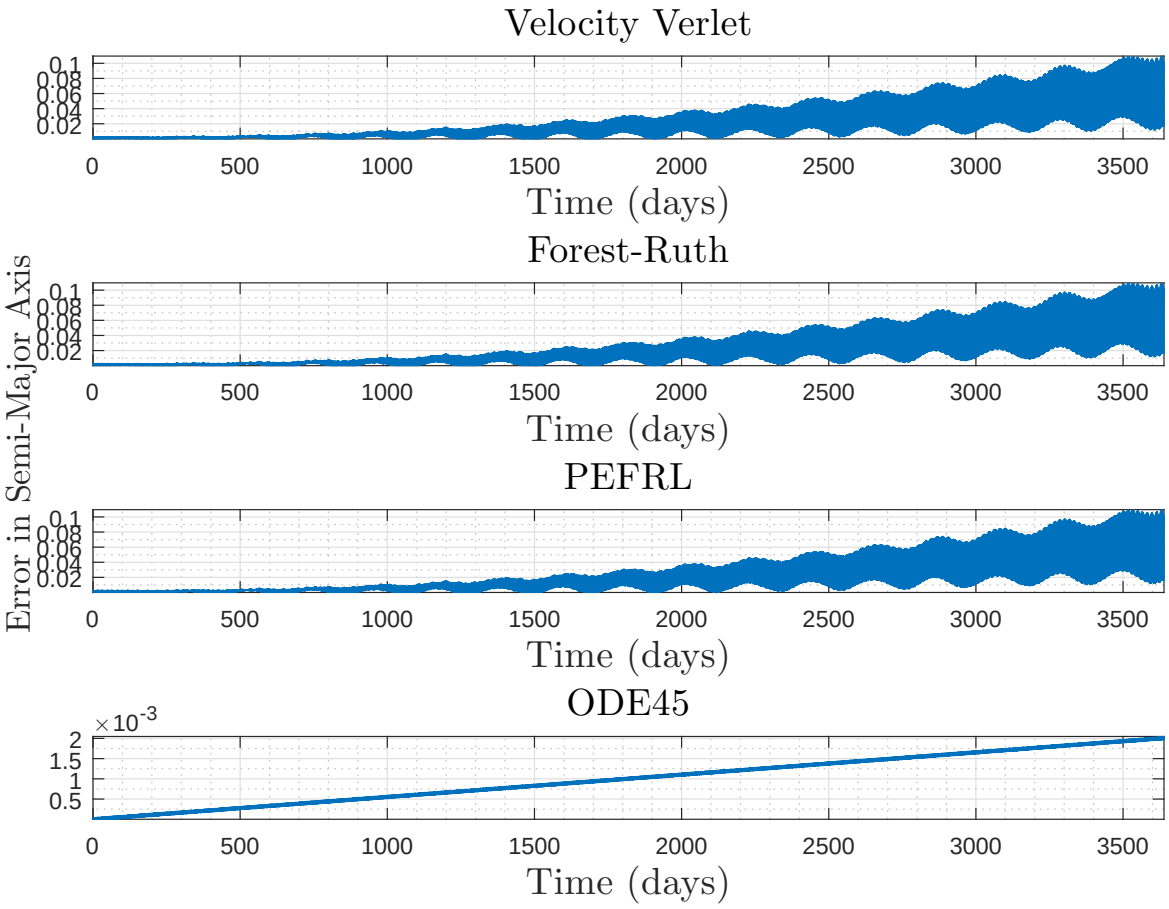

Figure 4.56. Semi-Major Axis Deviation(%) for a Perturbed GEO Orbit ($\delta t = 10s$ and time span of 10 year)

The previous figures that show the similar differences between exist the three symplectic algorithms and ODE45 in regards to the energy and semi-major axis deviations. The step size seems to have little effect on the overall evolution of the quantities for all three time spans, even for the symplectic algorithms. Moreover, the differences between the three symplectic algorithms seems to be minimal, even for a time span of 10 years. The efficiency of the algorithms is given in the following tables and subsequent graph in Figure 4.57.

Algorithm	**Time Step** (*seconds*)	**Execution Time** (*seconds*)
Velocity Verlet	1000	10.757
Forest-Ruth	1000	19.974
PEFRL	1000	23.162
ODE45	1000	56.078
Velocity Verlet	100	89.732
Forest-Ruth	100	141.820
PEFRL	100	169.364
ODE45	100	42.465
Velocity Verlet	10	823.866
Forest-Ruth	10	1306.034
PEFRL	10	1628.470
ODE45	10	91.601

Table 4.5. Execution Time of the Algorithms for Propagation of Perturbed Orbit for 1 year

Algorithm	**Time Step** (*seconds*)	**Execution Time** (*seconds*)
Velocity Verlet	1000	48.033
Forest-Ruth	1000	77.876
PEFRL	1000	108.597
ODE45	1000	242.055
Velocity Verlet	100	575.411
Forest-Ruth	100	811.240
PEFRL	100	1118.901
ODE45	100	247.033
Velocity Verlet	10	4595.529
Forest-Ruth	10	7166.326
PEFRL	10	18486.329
ODE45	10	556.030

Table 4.6. Execution Time of the Algorithms for Propagation of Perturbed Orbit for 5 years

Algorithm	**Time Step** (*seconds*)	**Execution Time** (*seconds*)
Velocity Verlet	1000	95.333
Forest-Ruth	1000	174.951
PEFRL	1000	224.363
ODE45	1000	472.424
Velocity Verlet	100	571.227
Forest-Ruth	100	948.047
PEFRL	100	1223.966
ODE45	100	387.156
Velocity Verlet	10	6928.301508
Forest-Ruth	10	8683.73481
PEFRL	10	42155.55148
ODE45	10	902.869694

Table 4.7. Execution Time of the Algorithms for Propagation of Perturbed Orbit for 10 years

The following graph in Figure 4.57 shows the data in the Tables 4.5 - 4.7 on a semi-log plot.

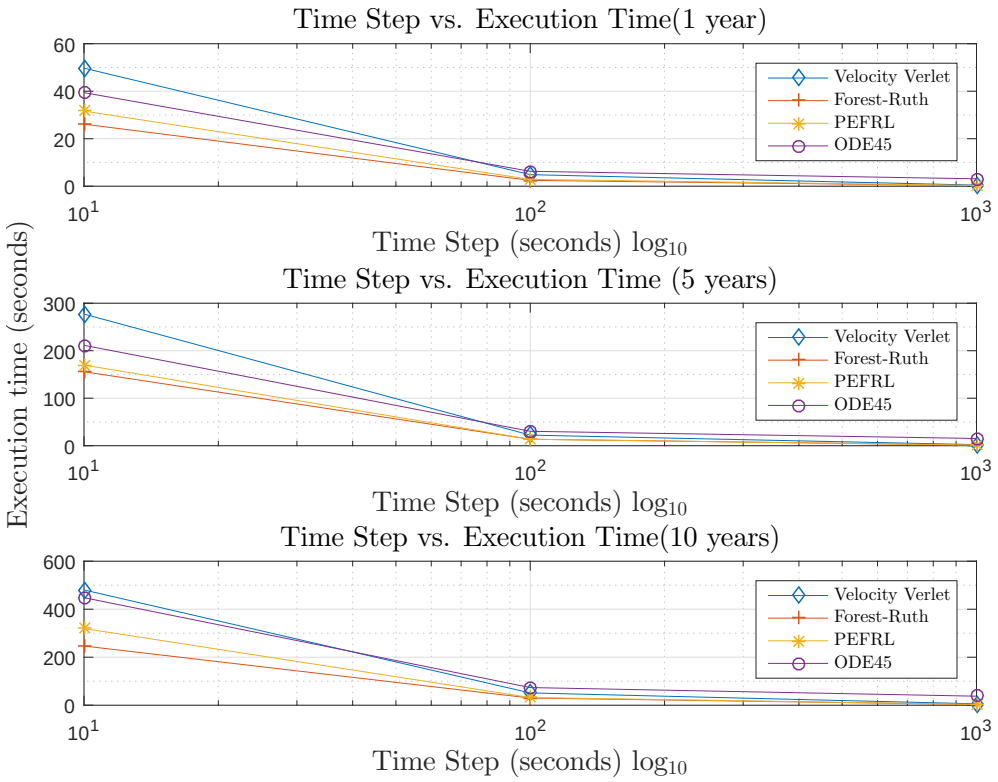

Figure 4.57. Semi-log Plot of the Execution Times

Tables 4.5 - 4.7 as well as Figure 4.57 indicate that with a more complex force model, the symplectic algorithms do not always maintain an advantage in efficiency over the Runge-Kutta algorithm. There seems to be no difference in execution time between 1000 second and 100 second time steps for the ODE45 algorithm. The accuracy of the algorithms is more difficult to evaluate since in this case, unlike the two-body problem, there is no invariant energy measure. However, as the previous section of this chapter indicates, the symplectic algorithms maintain their accuracy advantage even with a time step of 1000 seconds and are more efficient. This indicates that even while ODE45 algorithm is more efficient with a smaller time step, this might not be necessary since the symplectic algorithms allow for a larger time step without a significant loss in accuracy. Therefore, one can use a larger time

step to propagate the orbits using the symplectic algorithms and thereby achieve significant accuracy over ODE45. The advantages in computational efficiency for ODE45 can be attributed to its variable step size and error control implementations which can be useful in speeding up the integration process. However, as the results in this chapter demonstrate, there is a distinct accuracy advantage for the symplectic algorithms.

In this chapter, the results obtained through simple and perturbed two-body orbit propagation using the various algorithms were outlined. In the next chapter, a few conclusions regarding the results, algorithms among others will be made. In addition, suggestions of future work relating to this thesis will also be mentioned.

Chapter 5
Conclusions and Future Work

5.1 Conclusions

The accuracy of the symplectic algorithms seem to saturate (settle around a value). This holds true for all three symplectic algorithms even though they are of differing order. This is evidenced by observing the percent error graphs (both orbital energy and semi-major axis) seem to settle around a value of $10^{-5}\%$. While this may seem to be a limitation, further analysis reveals that since there is no secular growth in the error (unlike ODE45), the symplectic algorithms still maintain accuracy in the long term.

Comparing the execution times of the algorithms for the two-body problems reveals that the symplectic methods are superior at this than ODE45. However, this is reversed in the case of the perturbed problem where ODE45 is shown to be more efficient for certain time spans and time increments. The perturbed problem requires the evaluation of many functions for the equations of motion. As ODE45 is a pre-installed code in MATLAB, it could have been optimized such that that these function become the bottleneck of the code while the symplectic algorithms have other bottlenecks such as writing the data to a file. In addition, the error control features of the ODE45 algorithm could be contributing to ODE45's efficiency

advantage in this case. Since the symplectic algorithms are used in conservative system (energy is constant), the addition of perturbations should seem to violate this rule. However, the perturbations used here are mainly from gravitational sources and hence are energy conserving forces.

5.2 Future Work

This section will provide some thoughts on related topics that could be of interest in future work in the area of orbit propagation. In the process of completing the work, implementations of these algorithms in C++ was being considered. A few test cases of the two-body problem showed significant performance improvements over the algorithms written in MATLAB. In the end, MATLAB remained the software that was used due to the pre-installed suite of Runge-Kutta methods and the graphical capabilities that allow for the figures of the data to be generated with relative ease. However, it is recommended for future work to be done in C++ or other lower-level programming languages due to the aforementioned performance improvements as well as stability (MATLAB often crashed while attempting integration for longer time spans due to memory overflow).

The applications for these algorithms has been briefly mentioned in the first chapter of this thesis. One of the ancillary applications for this thesis were concentrated in the area of conjunction analysis. This, along with other area of space debris studies such as debris cloud evolution are interesting applications for these symplectic algorithms where the long-term accuracy of these symplectic integration algorithms can be very advantageous.

The algorithms presented here do not encompass the range of symplectic integration methods found in literature. One interesting paper [18], explores the

idea of variable time step symplectic integration. This could vastly improve the efficiency of these algorithms. Likewise, comparisons with higer order Runge-Kutta methods such as RK78 can be another interesting opportunity to assess the accuracy of symplectic algorithms. In addition, more comprehensive perturbations especially non-conservative forces such as drag and solar radiation pressure would be another interesting addition to the dynamics that would test the algorithms.

References

[1] P.J Cefola, A. Long and G. Holloway, Jr, "The long-term prediction of artificial satellite orbits," 12th Aerospace Sciences Meeting, Aerospace Sciences Meetings, 1974.

[2] J.C. Van der Ha, "Long-term evolution of near-geostationary orbits," Journal of Guidance, Control, and Dynamics. vol. 9, no. 3. pp. 363-370, May 1986.

[3] T. Quinn, N. Katz, J. Stadel and G. Lake, "Time stepping N-body simulations," October 1997. Preprint, astro-ph/9710043

[4] B. Kaufman and R. Basenbrock, "Semianalytic theory of long-term behavior of earth and lunar orbiters," Journal of Spacecraft and Rockets, vol. 10, no. 6, June 1973, pp. 377-383.

[5] K. Fox, "Numerical integration of the equations of motion of celestial mechanics," Celestial Mechanics and Dynamical Astronomy, vol. 33, no. 2, pp. 127-142, June 1984.

[6] Van Der Ha, J. C., "Long-term evolution of near-geostationary orbits," Journal of Guidance, Control, and Dynamics, vol. 9, pp. 363-370, June, 1986.

[7] Mikkola, S., "Efficient symplectic integration of satellite orbits," Celestial Mechanics and Dynamical Astronomy , vol. 74, pp. 275-285, July, 1999.

[8] R.H. Battin, *An introduction to the mathematics and methods of astrodynamics*, New York, N.Y: AIAA, 1987.

[9] D.A. Vallado, *Fundamentals of astrodynamics and applications*, 2nd ed. El Segundo, CA: Microcosm Press, 2001.

[10] C.C. Chao, *Applied orbit perturbation and maintenance*. El Segundo, CA: Aerospace Press, 2005.

[11] E. Hairer, C. Lubich and G. Wanner, *Geometric numerical integration*, 2nd ed, Berlin: Springer, 2006.

[12] S. Herrick, *Astrodynamics*, London: Van Nostrand Reinhold Co., 1971.

[13] B. D. Tapley, B. E. Schutz, and G. H. Born, *Statistical orbit determination*, Amsterdam: Elsevier Academic Press, 2004

[14] L. Verlet, "'Computer Experiments' on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules," Phys. Rev. vol. 159, no. 1, pp. 98-103, July 1967.

[15] E. Forest and R.D. Ruth, "Fourth-order symplectic integration." Physica D: Nonlinear Phenomena vol. 43, no. 1, pp. 105-117. May 1990.

[16] I.P. Omelyan *et al.* "Optimized Forest-Ruth and Suzuki-like algorithms for integration of motion in many-body systems." Computer Physics Communications, vol. 146, no. 2, pg. 188-202, July 2002.

[17] Dormand, J., and Prince, P., "A family of embedded Runge-Kutta formulae," Journal of Computational and Applied Mathematics, vol. 6, pp. 19-26, 1980.

[18] Hairer, E., "Variable time step integration with symplectic methods," Applied Numerical Mathematics, vol. 25, pp. 219-227, 1997.