Orbit Mechanics
Exercise 2: Numericial Integration of Satellite Orbits

Hsin-Feng Ho
03770686

January 12, 2024

Introduction

In the first exercise we have seeen that the orbit of a satellite can be calculated by an analytical solution of the two-body problem. The satellite orbit can also be determined by numerical integration of the equations of motion.

The Keplerian elements for the Sentinel-3 satellite is given:

Satellite	a[km]	e	i[deg]	Ω[deg]	ω[deg]	T_0[s]
Sentinel-3	7192	0.004	98.3	257.7	144.2	00:00

Undisturbed Orbit

Using the given Keplerian elements and analytical solution of two body problem we can calculate the orbit of the Sentinel-3 satellite.

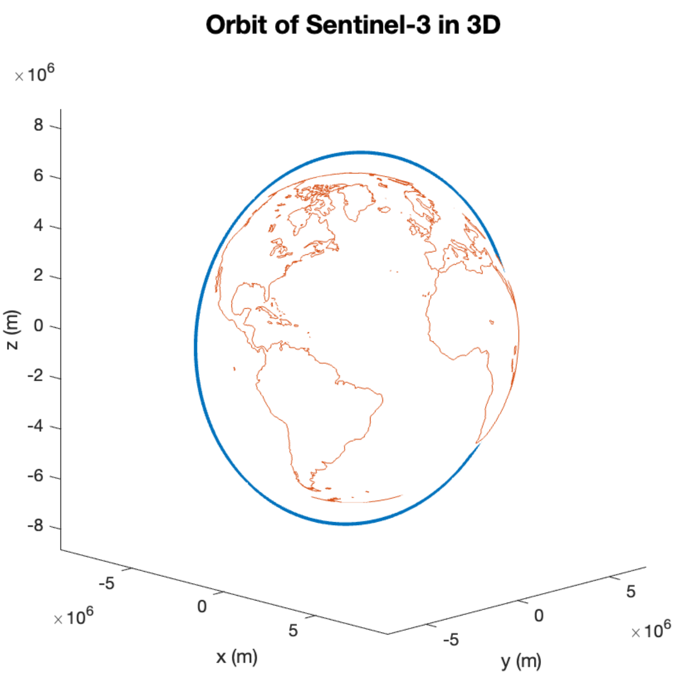


Figure 1: undisturbed orbit of Sentinel-3 satellite for 3 revolutions

Now we have to calculate the satellite orbit using numerical integration. The 2nd order differential equation for the two-body problem is:

$$\ddot{\mathbf{r}}_i = -\frac{GM}{||\mathbf{r}_i||^3}\mathbf{r}_i$$

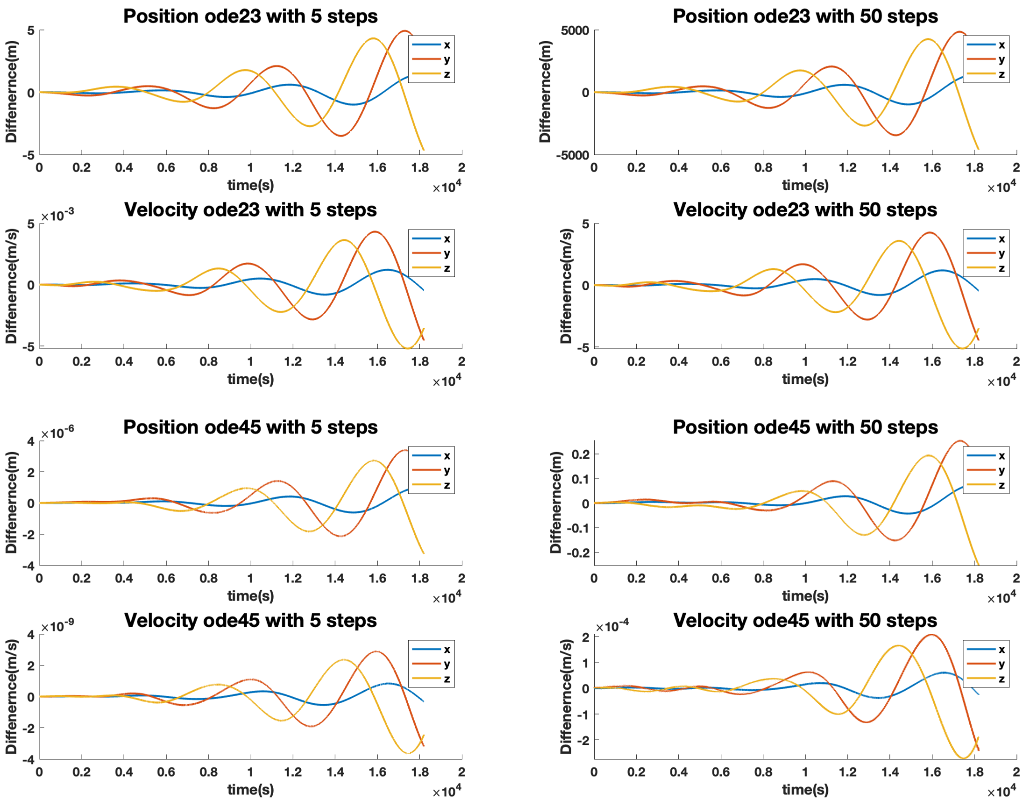
where G is the gravitational constant, M is the mass of the Earth and $\mathbf{r}_i$ is the position vector of the satellite. The equations of motion can be written as a system of first order differential equations:

$$\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} \dot{r_x} \\ \dot{r_y} \\ \dot{r_z} \end{pmatrix}$$

$$\begin{pmatrix} \dot{v_x} \\ \dot{v_y} \\ \dot{v_z} \end{pmatrix} = -\frac{GM}{R^3} \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix}$$

where $R = \sqrt{r_x^2 + r_y^2 + r_z^2}$.

Matlab offers several numerical integration functions. Now we want to compare the function **ode23** and **ode45** with step size of 5 and 50. The results are shown in the following figures:



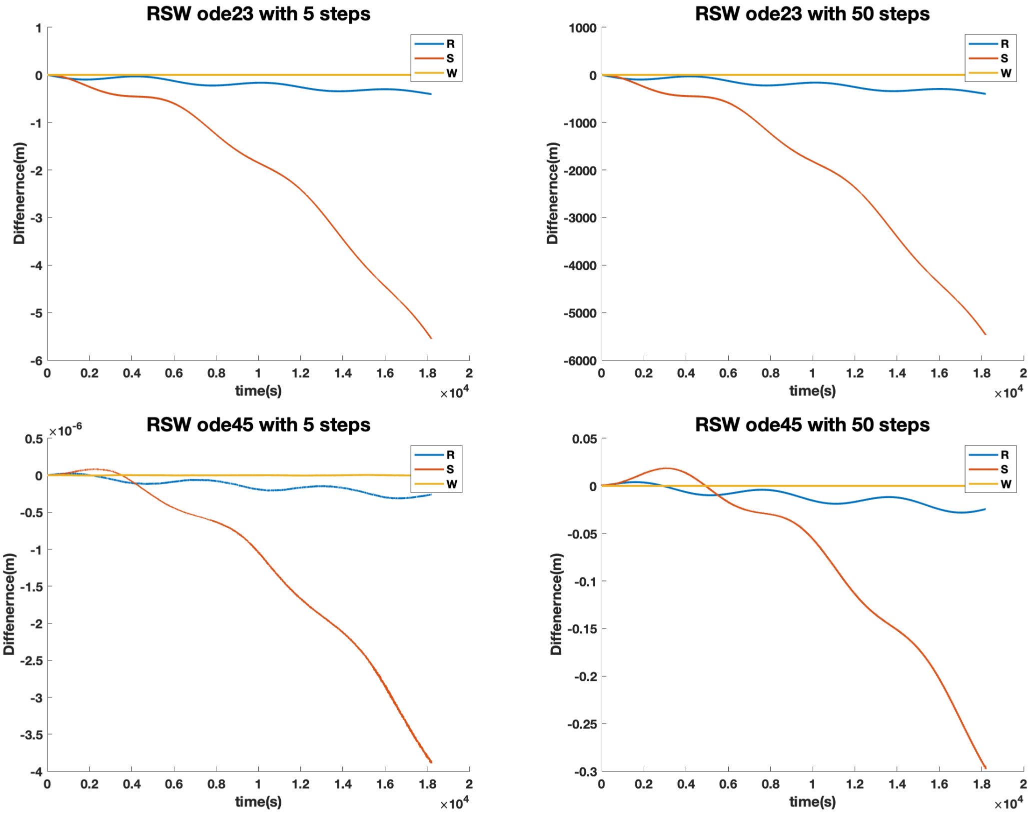
By observing the figures we can see that **ode45** is more accurate than **ode23**. The smaller the step size is, the more accurate the result is. The result of **ode45** with step size of 5 can even reach a precision of 10^{-6}m in position. In comparison, the result of **ode23** with step size of 5 is only accurate to meter level in position. With step size of 50, the result of **ode23** has a difference of 5km position and 5m/s velocity compared to the analytical solution, which we should avoid in practice.

Decomposition of the Error in RSW system

The error of the numerical integration can be decomposed into radial, along-track and cross-track directions in the RSW system. The radial direction is the direction from the satellite to the center of the Earth. The along-track direction is the direction of the velocity vector of the satellite. The cross-track direction is the direction perpendicular to the radial and along-track directions. The error in the RSW system can be calculated by:

$$\mathbf{e}_R = \frac{\mathbf{r}}{|\mathbf{r}|} \qquad \mathbf{e}_W = \frac{\mathbf{r}\times\mathbf{v}}{|\mathbf{r}\times\mathbf{v}|} \qquad \mathbf{e}_S = \mathbf{e}_W \times \mathbf{e}_R$$
$$\Delta r_R = \mathbf{e}_R \cdot \Delta\mathbf{r} \qquad \Delta r_W = \mathbf{e}_W \cdot \Delta\mathbf{r} \qquad \Delta r_S = \mathbf{e}_S \cdot \Delta\mathbf{r}$$

The results are shown in the following figures:



We can see that the error is mostly in the radial direction. The error in the along-track and cross-track directions are relatively small. Compare the two different step sizes, we can see that the error in the radial direction is smaller with smaller step size.

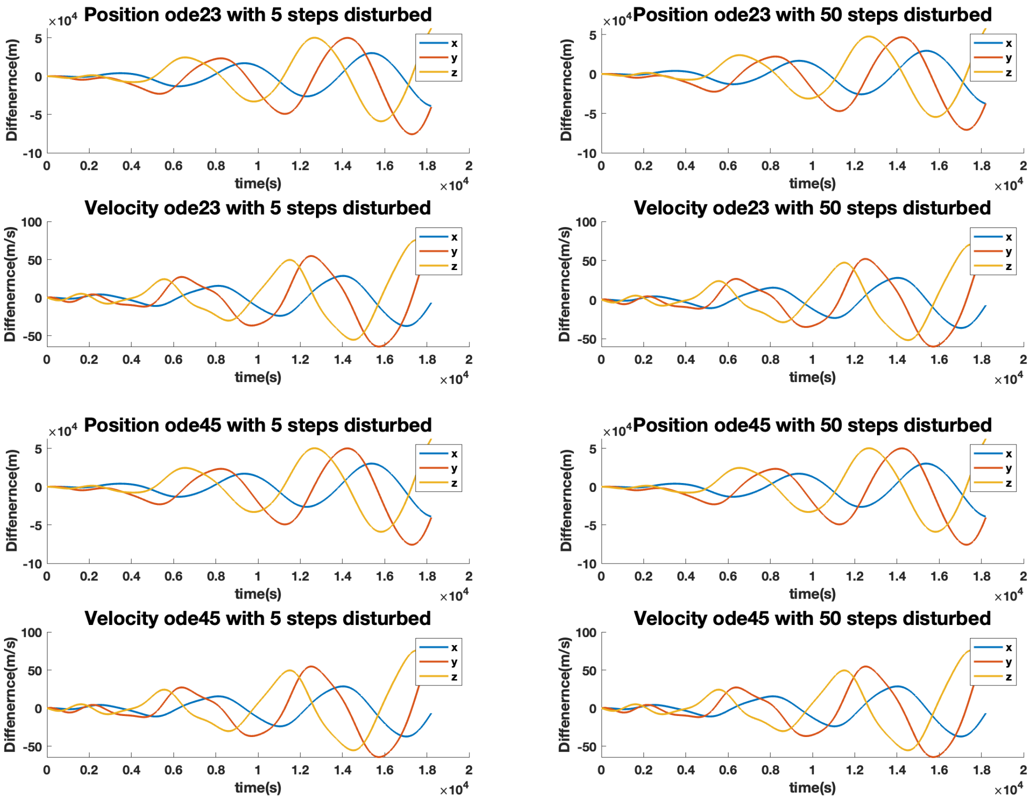
Disturbed Orbit

Now we want to add the perturbation of the Earth's oblateness to the equations of motion. The perturbation of the Earth's oblateness is given by:

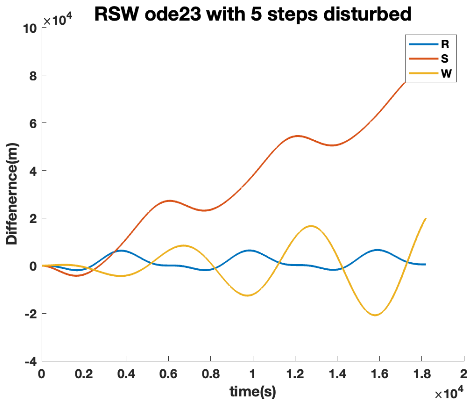
$$\ddot{\mathbf{r}}_i = -\frac{GM}{r^3}\mathbf{r}_i + \frac{3}{2}\frac{J_2 a_e^2}{r^2}\begin{pmatrix} x\left(5\left(\frac{z}{r}\right)^2 - 1\right) \\ y\left(5\left(\frac{z}{r}\right)^2 - 1\right) \\ z\left(5\left(\frac{z}{r}\right)^2 - 3\right)\end{pmatrix}$$

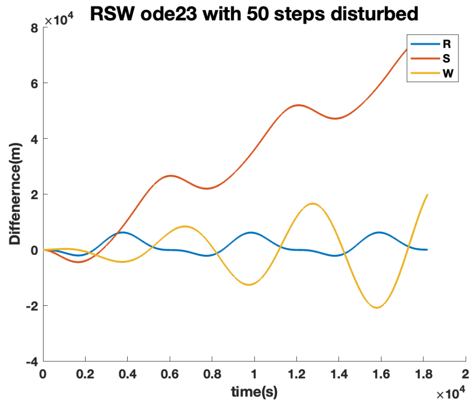
where J_2 is the second zonal harmonic coefficient, a_e is the equatorial radius of the Earth and $r = \sqrt{x^2 + y^2 + z^2}$.

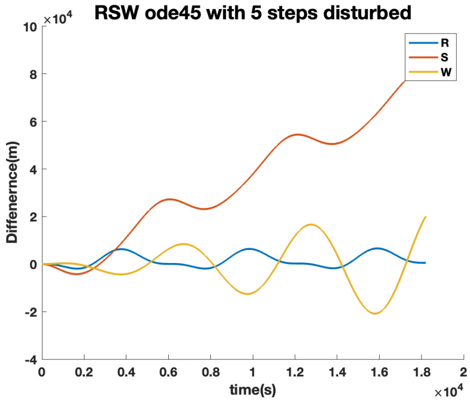
The results of the numerical integration with the perturbation of the Earth's oblateness are shown in the following figures:

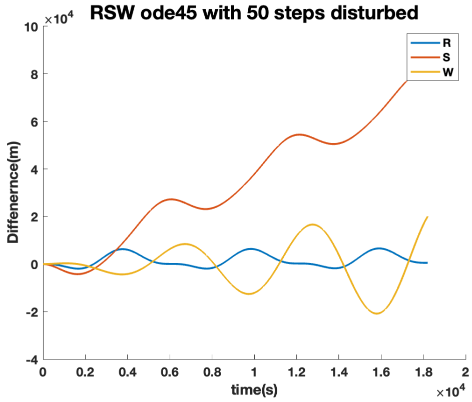


In RSW frame









Own Implementation of the Numerical Integration

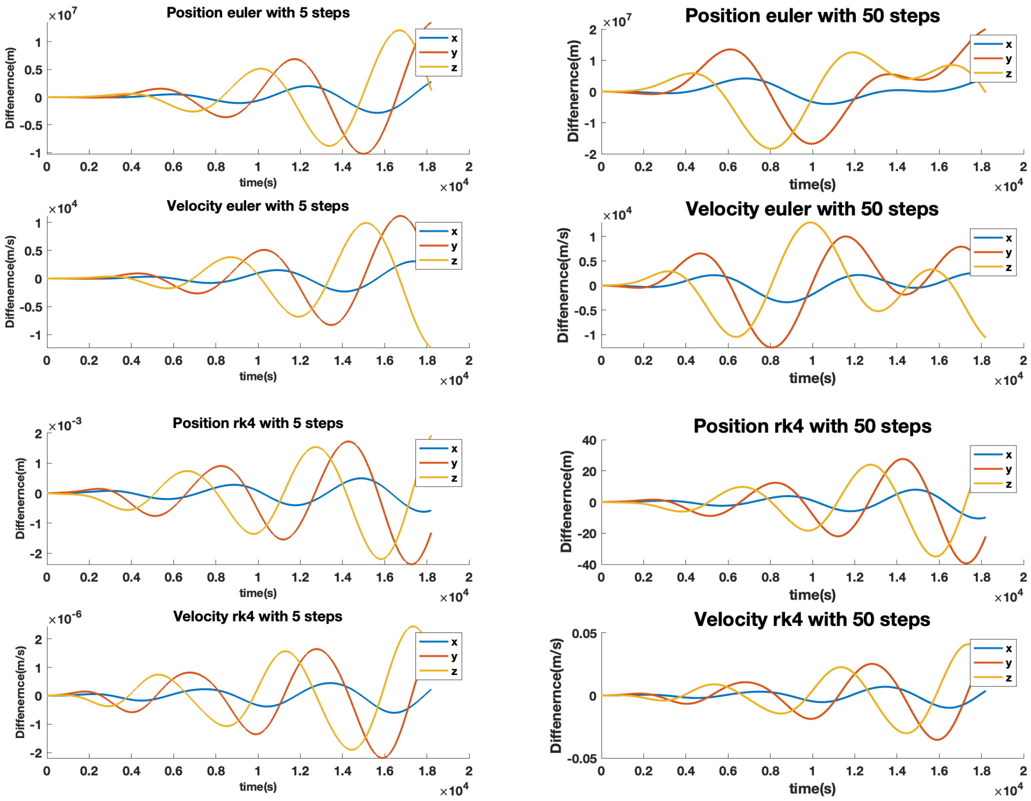
Now we want to implement our own numerical integration function. A simple case is to use Euler method to solve the differential equation. The Euler method is given by:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h\mathbf{f}(t_n, \mathbf{y}_n)$$

A more precise method is the Runge-Kutta method. The Runge-Kutta 4th order method is given by:

$$\begin{aligned}
\mathbf{y}_{n+1} &= \mathbf{y}_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= \mathbf{f}(t_n, \mathbf{y}_n) \\
k_2 &= \mathbf{f}(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}k_1) \\
k_3 &= \mathbf{f}(t_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}k_2) \\
k_4 &= \mathbf{f}(t_n + h, \mathbf{y}_n + hk_3)
\end{aligned}$$

The results of the numerical integration with the Euler method and Runge-Kutta method are shown in the following figures:


Position euler with 5 steps
Velocity euler with 5 steps
Position euler with 50 steps
Velocity euler with 50 steps
Position rk4 with 5 steps
Velocity rk4 with 5 steps
Position rk4 with 50 steps
Velocity rk4 with 50 steps
Diffenernce(m)
Diffenernce(m/s)
time(s)
x
y
z