

# **A Review and a Quantitative Comparison among the Exact Solution and the Numerical Integration Methods with Fixed Time Step Commonly Used to Solve the Two-Body Problem**

Tamer Mekky Ahmed Habib

Researcher at the Egyptian National Authority for Remote Sensing and Space Science,  
Cairo, Egypt.

[tamermekky@hotmail.com](mailto:tamermekky@hotmail.com)

## **ABSTRACT**

The motion of a spacecraft around the earth is affected with many forces. The major force affecting this motion is the gravity force resulting from a spherical central body (the earth). This motion is commonly known in the literature as the two-body problem. The main drive for this research is to select the best numerical integration algorithm of the two-body problem on the basis of quantitative measures. The error of each integration algorithm is measured with respect to the exact solution of the two-body problem and the test results were embarrassing. Also, the average execution time is compared for all of these algorithms.

**KEYWORDS:** Two-Body, Numerical Integration Algorithm, Circular Orbit.

## **1. Introduction:-**

The motion of a spacecraft orbiting the earth is affected with many forces. These forces are: the gravity force resulting from a spherical central body (the earth), the aerodynamic drag and lift forces, the effect of non-spherical earth model (earth's oblateness), the gravity forces resulting from the sun and other planets, the solar radiation pressure forces, tidal forces (solid earth and ocean tides) [1]. The motion of a spacecraft in its orbit around the earth is primarily due to the spherical central body force. All other forces represent only perturbations (i.e., small fractions) with respect to this force, so our main concern is mainly devoted to the spherical central body force. The problem of finding the spacecraft motion due to only the force of a spherical central body is commonly known as the two-body problem. Several numerical integration algorithms of the two-body problem are found in the literature. Fourth order Runge-Kutta method, and Runge-Kutta-Gill are discussed in [2]. Ref. [3] utilizes Runge-Kutta five to integrate the two-body equation of motion. Ref. [4] presents the 8<sup>th</sup> order Runge-Kutta. Ref. [5] solves the problem using ODE45 solver provided by Matlab.

Most of remote-sensing spacecraft missions are chosen to be sun-synchronous. And a vast number of these missions are characterized by nearly circular orbits. In Egypt, we are concerned mainly with these types of missions such as the former Egyptian remote-sensing satellite Egypt-sat 1. Therefore, our interest in this research at hand is mainly devoted to circular orbits.

The main role of the research at hand is to answer simply the problem of finding a suitable numerical integration algorithm to be used for orbit propagation and estimation such as that found in [8]. This is done through a review of the commonly used numerical integration algorithms with fixed time step of the two-body problem. The decision to use a certain algorithm or another is mainly taken based on quantitative measures. These measures are the root mean square (RMS) of the position error, in addition to the average execution time.

## **2. The two body problem and the exact solution:-**

The two-body equation of motion is given by [1]

$\begin{bmatrix} \overset{\circ}{X}_I \\ \overset{\circ}{Y}_I \\ \overset{\circ}{Z}_I \\ \overset{\circ\circ}{X}_I \\ \overset{\circ\circ}{Y}_I \\ \overset{\circ\circ}{Z}_I \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\mu_E/\ R_I\ ^3 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\mu_E/\ R_I\ ^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\mu_E/\ R_I\ ^3 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_I \\ Y_I \\ Z_I \\ \overset{\circ}{X}_I \\ \overset{\circ}{Y}_I \\ \overset{\circ}{Z}_I \end{bmatrix} \quad (1)$	
---	--

Where

$\mu_E$  : Is the earth's gravitational parameter ( $\mu_E = 3.986 \times 10^{14} \text{ m}^3/\text{s}^2$ ).

$\left[ X_I \ Y_I \ Z_I \ \overset{\circ}{X}_I \ \overset{\circ}{Y}_I \ \overset{\circ}{Z}_I \right]^T$  : defined as the orbital state vector comprised from inertial position and velocity components respectively.

$R_I$  : is the spacecraft inertial position vector.

Ref. [6], presents another form of the two-body problem in terms of classical orbital elements. This form is expressed as

$\begin{bmatrix} \overset{\circ}{a} \\ \overset{\circ}{e} \\ \overset{\circ}{i} \\ \overset{\circ}{\Omega} \\ \overset{\circ}{\omega} \\ \overset{\circ}{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{\sqrt{\mu_E} (1 + e \cos \theta)^2}{[a(1 - e^2)]^{1.5}} \end{bmatrix} \quad (2)$	
--	--

Where  $a$  is the semi-major axis,  $e$  is the eccentricity,  $i$  is the orbit inclination,  $\Omega$  is the right ascension of the ascending node,  $\omega$  is the argument of perigee, and  $\theta$  is the true anomaly. For simplicity let  $e = 0$ . The last equation of equation (2) then becomes

$\overset{\circ}{\theta} = \sqrt{\frac{\mu_E}{a^3}}$	
--	--

Integrating equation (3) gives,

$\theta = \theta_0 + \sqrt{\frac{\mu_E}{a^3}}(t - t_0)$	(4)
---	-----

The solution of equation (2) then becomes

$\begin{bmatrix} a \\ e \\ i \\ \Omega \\ \omega \\ \theta \end{bmatrix} = \begin{bmatrix} a_0 \\ e_0 \\ i_0 \\ \Omega_0 \\ \omega_0 \\ \theta_0 + \sqrt{\frac{\mu_E}{a^3}}(t - t_0) \end{bmatrix}$	(5)
---	-----

### 3. Integration algorithms:-

All of the integration algorithms described herein are used to solve a nonlinear first order differential equation described by

$\frac{dx}{dt} = f(t, x)$	(6)
---------------------------	-----

With initial conditions given by

$x(t_0) = x_0$	(7)
----------------	-----

### 3.1 Fourth order Runge-Kutta

The standard fourth order Runge-Kutta method is given as [2]

With

$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$	(8)
$k_1 = hf(t_n, x_n)$	(9)
$k_2 = hf\left(t_n + \frac{h}{2}, x_n + \frac{k_1}{2}\right)$	(10)
$k_3 = hf\left(t_n + \frac{h}{2}, x_n + \frac{k_2}{2}\right)$	(11)
$k_4 = hf(t_n + h, x_n + k_3)$	(12)

Where

$h$  : Is the step-size.

$n$  : Is the increment number.

### 3.2 Runge-Kutta-Gill

This method is developed basically for high speed computers to control round off error growth. The formula is [2]

$x_{n+1} = x_n + \frac{k_1}{6} + \frac{1}{3}\left(1 - \sqrt{\frac{1}{2}}\right)k_2 + \frac{1}{3}\left(1 + \sqrt{\frac{1}{2}}\right)k_3 + \frac{k_4}{6}$	(13)
$k_1 = hf(t_n, x_n)$	(14)
$k_2 = hf\left(t_n + \frac{h}{2}, x_n + \frac{k_1}{2}\right)$	(15)
$k_3 = hf\left[t_n + \frac{h}{2}, x_n + k_1\left(\sqrt{\frac{1}{2}} - \frac{1}{2}\right)\right] + k_2\left(1 - \sqrt{\frac{1}{2}}\right)$	(16)

$k_4 = hf \left[ t_n + h, x_n - \frac{k_2}{\sqrt{2}} + k_3 \left( 1 + \sqrt{\frac{1}{2}} \right) \right]$	(17)
---	------

### 3.3 Runge-Kutta 5

The basic procedure of Runge-Kutta 5 is given in [3] as

$x_{n+1} = x_n + \frac{1}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6)$	(18)
$k_1 = hf(t_n, x_n)$	(19)
$k_2 = hf\left(t_n + \frac{h}{4}, x_n + \frac{1}{4}k_1\right)$	(20)
$k_3 = hf\left(t_n + \frac{h}{4}, x_n + \frac{1}{8}k_1 + \frac{1}{8}k_2\right)$	(21)
$k_4 = hf\left(t_n + \frac{h}{2}, x_n - \frac{1}{2}k_2 + k_3\right)$	(22)
$k_5 = hf\left(t_n + \frac{3}{4}h, x_n + \frac{3}{16}k_1 + \frac{9}{16}k_4\right)$	(23)
$k_6 = hf\left(t_n + h, x_n - \frac{3}{7}k_1 + \frac{2}{7}k_2 + \frac{12}{7}k_3 - \frac{12}{7}k_4 + \frac{8}{7}k_5\right)$	(24)

### 3.4 Runge-Kutta 8

The commonly used 8<sup>th</sup> order Runge-Kutta formula is given by [4]

$x_{n+1} = x_n + \frac{1}{840}(41k_1 + 27k_4 + 272k_5 + 27k_6 + 216k_7 + 216k_9 + 41k_{10})$	(25)
$k_1 = hf(t_n, x_n)$	(26)
$k_2 = hf\left(t_n + \frac{4}{27}h, x_n + \frac{4}{27}k_1\right)$	(27)
$k_3 = hf\left(t_n + \frac{2}{9}h, x_n + \frac{1}{18}k_1 + \frac{1}{6}k_2\right)$	(28)
$k_4 = hf\left(t_n + \frac{1}{3}h, x_n + \frac{1}{12}k_1 + \frac{1}{4}k_3\right)$	(29)

$k_5 = hf \left( t_n + \frac{1}{2}h, x_n + \frac{1}{8}k_1 + \frac{3}{8}k_4 \right)$	(30)
---	------

$k_6 = hf \left( t_n + \frac{2}{3}h, x_n + \frac{1}{54} [13k_1 - 27k_3 + 42k_4 + 8k_5] \right)$	(31)
---	------

$k_7 = hf \left( t_n + \frac{1}{6}h, x_n + \frac{1}{4320} [389k_1 - 54k_3 + 966k_4 - 824k_5 + 243k_6] \right)$	(32)
--	------

$k_8 = hf \left( t_n + h, x_n + \frac{1}{20} [-231k_1 + 81k_3 - 1164k_4 + 656k_5 - 122k_6 + 800k_7] \right)$	(33)
--	------

$k_9 = hf \left( t_n + \frac{5}{6}h, x_n + \frac{1}{288} [-127k_1 + 18k_3 - 678k_4 + 456k_5 - 9k_6 + 576k_7 + 4k_8] \right)$	(34)
--	------

$k_{10} = hf \left( t_n + h, x_n + \frac{1}{820} [1481k_1 - 81k_3 + 7104k_4 - 3376k_5 + 72k_6 - 5040k_7 - 60k_8 + 720k_9] \right)$	(35)
--	------

### 3.5 Adams-Moulton:-

The fourth order method retaining third differences is described here [2]. The predictor term is given by

$x_{n+1}^p = x_n + \frac{h}{24} [55f(t_n, x_n) - 59f(t_{n-1}, x_{n-1}) + 37f(t_{n-2}, x_{n-2}) - 9f(t_{n-3}, x_{n-3})]$	(36)
---	------

The corrector term is given by

$x_{n+1}^c = x_n + \frac{h}{24} [9f(t_{n+1}, x_{n+1}^p) + 19f(t_n, x_n) - 5f(t_{n-1}, x_{n-1}) + f(t_{n-2}, x_{n-2})]$	(37)
--	------

## 4. Performance measures:

In order to test the integrators, performance measures must be defined. In the research at hand, two performance measures are defined. The first performance measure is the RMS position error. First, we define the position error at each integration time step as

$\Delta R_i =  R_i^{exact} - R_i^{computed} _i$	(38)
---	------

The RMS position error is calculated as [7]

$\Delta R_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta R_i)^2}$	(39)
---	------

The second performance measure used is the average execution time of the integration algorithm.

## 5. Test-case parameters and results:-

In order to test the given numerical integrators a test-case spacecraft is utilized. The test scenario lasts for one week. It starts at 1-7-2007 12:00:00 UTC and ends at 8-7-2007 12:00:00 UTC. The integration time step ( $\Delta T$ ) is first selected as 5 seconds. The initial conditions

$\left[ X_I \quad Y_I \quad Z_I \quad \overset{\circ}{X}_I \quad \overset{\circ}{Y}_I \quad \overset{\circ}{Z}_I \right]$  are given by

$10^3 [1113.475306 \quad -6977.855318 \quad 0 \quad -1.050671 \quad -0.167658 \quad 7.434913]$ . Fig. 1. shows the log of the position error for the previously presented numerical integration algorithms for integration time step of 5 seconds. As seen in this figure, the initial position error is zero for all of the integration algorithms and starts to degrade afterwards. The worst position error is achieved by the Runge-Kutta-Gill algorithm. Afterwards, there comes the Runge-Kutta 5. Runge-Kutta 4, Runge-Kutta 8, and Adams-Moulton which have nearly similar behaviors for the position error. Table 1. shows performance measures for the different utilized numerical integration algorithms for integration time step of 5 seconds. The maximum average execution time is achieved by the Adams-Moulton algorithm. Then Runge-Kutta 8, achieves the second maximum average execution time. Runge-Kutta 4, Runge-Kutta 5, and Runge-Kutta-Gill have nearly the same average execution time. Fig. 2 shows the log of the position error for the previously presented numerical integration algorithms for integration time step of 30 seconds. Table 2. shows performance measures for the different utilized numerical integration algorithms for integration time step of 30 seconds. The Runge-Kutta 8 algorithm achieves a very good RMS position error. And then there comes the Runge-Kutta 4 algorithm. Runge-Kutta-Gill, Runge-Kutta 5, and Adams-Moulton have high RMS position error. The maximum average execution time is achieved by the Adams-Moulton algorithm. And then comes Runge-Kutta 8 algorithm. Runge-Kutta 4, Runge-Kutta-Gill, and Runge-Kutta 5, have similar average execution time. The RMS error of Runge-Kutta 8 could reaches 16.7 m if the integration time step is increased up to 135 sec.



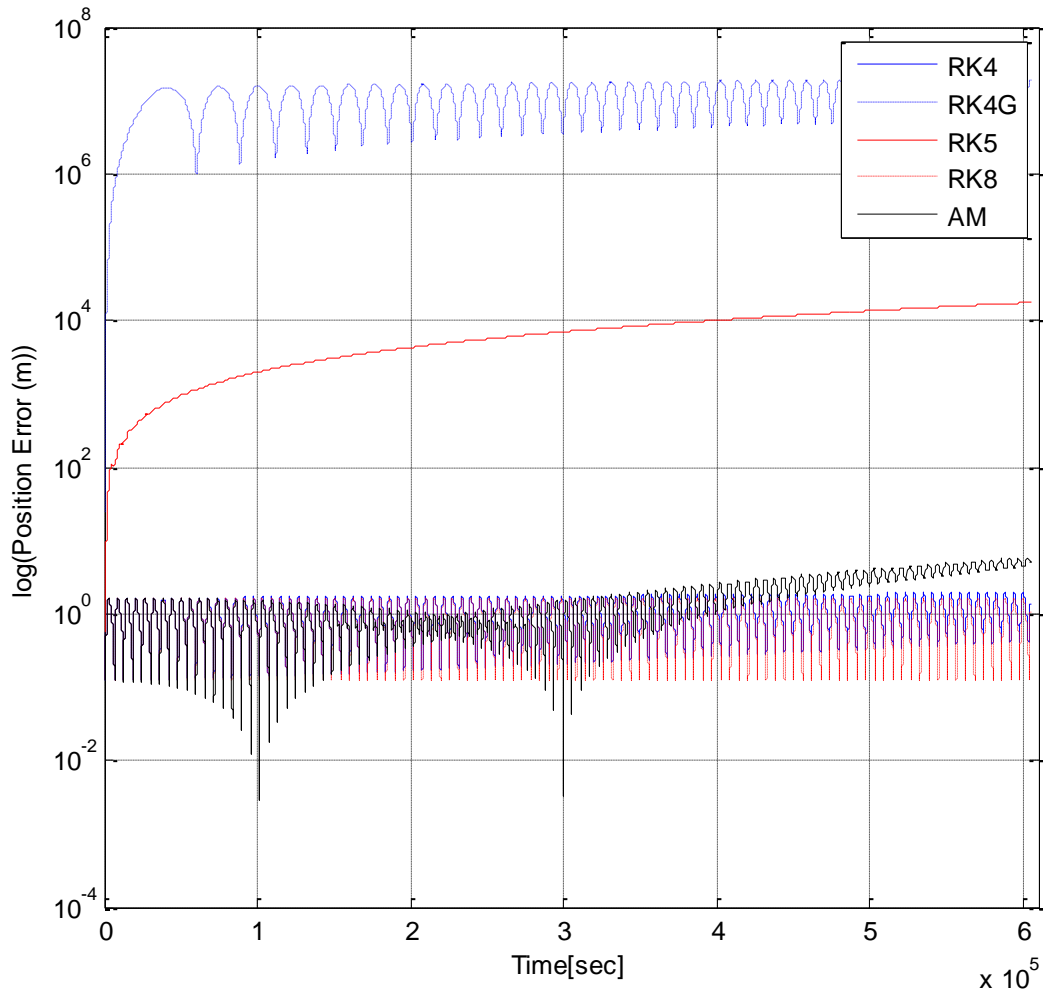


Fig. 1. Position error log for different numerical integration algorithms ( $\Delta T = 5$  sec.).

Performance Measure		
Numerical Integration Algorithm	$\Delta R_{RMS}$ (m)	Average execution time (sec)
Runge-Kutta 4	1.186	0.0018
Runge-Kutta-Gill	$1.2583 \times 10^7$	0.0016
Runge-Kutta 5	$9.1713 \times 10^3$	0.0018
Runge-Kutta 8	1.0868	0.002
Adams-Moulton	2.2453	0.003

Table 1. Performance measures for numerical integration algorithms ( $\Delta T = 5$  sec.).

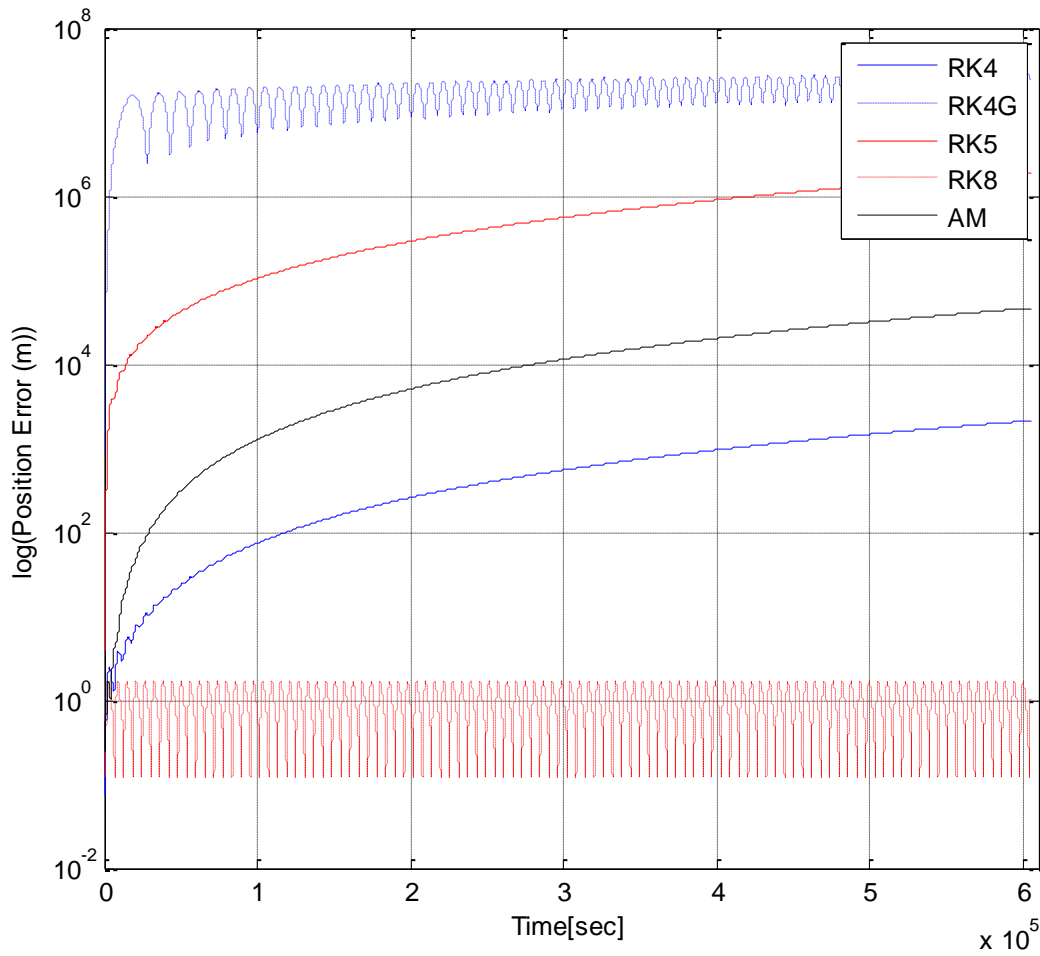


Fig. 2. Position error log for different numerical integration algorithms ( $\Delta T = 30$  sec.).

Performance Measure		
Numerical Integration Algorithm	$\Delta R_{RMS}$ (m)	Average execution time (sec)
Runge-Kutta 4	958.0656	$5.0802 \times 10^{-4}$
Runge-Kutta-Gill	$1.8654 \times 10^7$	$5.1085 \times 10^{-4}$
Runge-Kutta 5	$8.7817 \times 10^5$	$6.2769 \times 10^{-4}$
Runge-Kutta 8	1.0870	$8.8122 \times 10^{-4}$
Adams-Moulton	$2.2702 \times 10^4$	$6.0661 \times 10^{-4}$

Table 2. Performance measures for numerical integration algorithms ( $\Delta T = 30$  sec.).

## 6. Conclusion:-

Runge-Kutta 8 algorithm is characterized by high accuracy and high computational load compared to the previously discussed numerical integration algorithms. The excessive computational load needed by the Runge-Kutta 8 algorithm could be greatly compensated by using larger integration time step while retaining the same high accuracy. The accuracy of Runge-Kutta 8 with a time step 30 seconds is better than the accuracy of Runge-Kutta 4 with a time step 5 seconds. In addition, the total execution time of Runge-Kutta 8 is greatly less than that of Runge-Kutta 4 due to its ability to use larger time step. Thus, Runge-Kutta 8 achieves a superior performance. Runge-Kutta 4 achieves the second best performance after the Runge-Kutta 8. Runge-Kutta-Gill, Runge-Kutta 5, and Adams-Moulton achieved lower performance measures compared to Runge-Kutta 8, and Runge-Kutta 4 algorithms.

## 7. References:-

1. Vallado, D., "Fundamentals of Astrodynamics and applications", Microcosm Press and Kluwer Academic Publishers, second printing, 2004.
2. Bate, R., et al., "Fundamentals of Astrodynamics", Dover Publication, INC., first printing, 1971.
3. Boulet, D., "Method of Orbit Determination for the Micro Computer", Willmann-Bell, INC., first English edition, 1991.
4. Guochang X., "Orbits", Springer-Verlag, 2008.
5. Foster, B., "Orbit Determination for a Microsatellite Rendezvous with a Non-Cooperative Target", MSc Thesis, Air Force Institute of Technology, 2003.
6. Bar-Itzhack, I., "Satellite Autonomous Navigation Based on Magnetic Field Measurements", Journal of Guidance, Control, and Dynamics, Vol.18, NO. 4, pp. 843-850, 1995.
7. Berry, M., "A Variable-Step Double-Integration Multi-Step Integrator", PhD Thesis, Virginia Polytechnic Institute and State University, 2004.
8. T.M. Habib, "Concurrent Spacecraft Attitude and Orbit Estimation with Attitude Control Based on Magnetometer, Gyroscope, and GPS Measurements through Extended Kalman Filter", Proceeding of the 15-th International Conference on Aerospace science and Aviation Technology, 2013