Comparison of numerical methods for propagating the orbit of Apophis

gjg1g18

2019

Introduction

Differential equations are the base mathematical tool for modeling the motion of objects in space. To propagate an object's orbit from a known starting position, numerous perturbations create a complex right-hand side which is, in general, impossible to analytically solve. Therefore, numerical integration is at the heart of all propagation solutions today. The aim of this Master's project is to study and compare the efficiency of Multistep Collocation Methods (MCMs) to state-of-the art methods, in the orbit propagation of the near-Earth asteroid (99942) Apophis, and its close encounters with Earth. These results may prove useful for propagating other asteroids and planetary bodies more efficiently.

Apophis

Near-Earth Asteroids (NEAs) present a risk to human life [16] with a continuous threat of possible impact. (99942) Apophis, originally designated 2004 MN4 [14], is perhaps the most famous NEA, with possible deflection missions to it taken very seriously by leading academics [10]. Apophis is most well known for, shortly after its discovery, briefly having a 2.7% probability of colliding with Earth on April 13, 2029 [9] [1]. Further observations corrected its ephemeris to one with a negligible chance of impact, but a close approach of about 38000km from Earth's center at this time [3]. This entry into Earth's gravitational field drastically increases the uncertainty region of its ephemeris post-2029, resulting in several tiny possibilities of collisions throughout the 21st century [18].

Numerical methods

The motion of Apophis, or of any object in space, can be modelled by the following differential equation:

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}),$$
$$\mathbf{y}(t_0) = \mathbf{y_0},$$

where $\mathbf{y}$ and $\mathbf{f}$ are vectors of equal size, and $\mathbf{f}$ is a vector function of $\mathbf{y}$ and t. In our case, $\mathbf{y}$ represents the state vector of Apophis, which is a 6-element vector composed of its position and velocity vectors in a given reference frame. Therefore, its time derivative $\mathbf{f}$ is composed of its velocity vector followed by its acceleration vector. Imposing some known initial conditions $\mathbf{y_0}$ guarantees a unique solution for $\mathbf{y}(t)$.

As $\mathbf{f}$ is a function of both $\mathbf{y}$ and t, we typically have a non-linear differential equation, and one which is very challenging to solve analytically - especially in our astrophysical context [19]. To solve this equation, we will be investigating numerical methods which use successive forward time steps to approximate values of $\mathbf{y}(t)$, starting from t_0. My Master's project is focused on the comparison of these methods with each other, in terms of accuracy versus computational cost, or efficiency.

Efficiency is not the only important measure of a method's performance - the idea of *stability* measures how well a method converges to the correct solution. Taking the test equation $y' = ky$, the *stability region* of a method is the set of $k \in \mathbb{C}$ such that the test equation converges using the method as $t \to \infty$. A widely used classification is that of A-stability: a method is considered A-stable if the stability region includes all $z \in \mathbb{C} : Re(z) < 0$ ([13], p.v42). Stability is particularly important in the study of *stiff* equations - equations with rapid variation in their solution during certain time intervals, with respect to the rest of the time space. In the case of Apophis

For discussion of numerical integrator methods, we simplify y as a scalar function, without loss of generality. Runge-Kutta methods are an extremely popular family of numerical methods, which rely on stacking successive estimates of y on each other between t_n and t_{n+1}, in order to obtain a high-order solution. The general formulation of Runge-Kutta methods is [4]:

$$g_i = f(t_n + c_i h, y(t_n) + h \sum_{j=1}^{s} a_{ij} g_j)), \text{ for } i = 1..s,$$
$$y(t_{n+1}) = y(t_n) + h \sum_{i=1}^{s} b_i g_i,$$

with the condition that

$$c_i = \sum_{j=0}^{s} a_{ij} \text{ for } i = 1..s.$$

If $a_{ij} = 0$ for all $j > i$, the method is explicit and can be computed directly - otherwise it is implicit and root finding techniques, primarily Newton's method

[4], must be used for the g_i. Implicit methods can offer high stability, with A-stable methods for any order ([2], p.433), but the root-finding methods involved incur further computational cost.

A collocation method is a subset of Implicit Runge-Kutta (IRK) methods, and is one which can be visualised more intuitively. Given s time points of a function, the unique polynomial of $\mathcal{O}(s)$ whose derivative matches, or 'co-locates', with f at these points, is integrated instead. By definition, the error term is then certain to be at least $\mathcal{O}(s+1)$ [6].

Another family of integrators are multistep methods, which take already-computed y values at multiple previous time points to move forward to the next. These methods can also be explicit or implicit - implicit multistep methods have been found [7] to be more stable and accurate than the explicit type for a wide variety of inputs. The general form of a multistep method with r total points is shown ([8], p. 107):

$$y_{n+1} = \sum_{i=1}^{r} \alpha_i y_{(n+1)-i} + h \sum_{i=0}^{r} \beta_i f(t_{(n+1)-i}, y_{(n+1)-i}).$$

Runge-Kutta and multistep methods are both well-studied and state-of-the-art in many different fields of use. Multistep Collocation Methods (MCMs) are a less well-studied family of methods, which combine the ideas of both Runge-Kutta/collocation methods and multistep methods. The basic idea is that we find a collocation polynomial which collocates with k past solution points, and whose derivative matches f at s intermediate solution points; i.e.

$$\begin{aligned} u(t_j) &= y_j & j &= n-k+1,..,n \\ u'(t_n + c_i h) &= f(t_n + c_i h, u(t_n + c_i h)) & i &= 1,..,s. \end{aligned}$$

which gives the numerical solution $y_{n+1} = u_{t_{n+1}}$. However, as we don't know $u(t_n + c_i h)$, we find estimates of u for each i in the same way as in the classical Runge-Kutta method. The full method is described as such:

$$\begin{aligned} g_i &= \sum_{j=1}^{k} a_{ij} y_{n-k+j} + h \sum_{j=1}^{s} b_{ij} f(t_n + c_j h, g_j), & i &= 1,2,..,s, \\ y &= g_s, \end{aligned}$$

where a_{ij} and b_{ij} are the weighting coefficients. Similarly to classical Runge-Kutta methods, MCMs are implicit when a_{ij} and b_{ij} are not both strictly lower triangular. Here,

Varying s and r makes a big difference in stability and accuracy properties [4], and so comparing the properties of different $r-s$ combinations is a great starting point in the analysis of MCMs.

The points where f is evaluated in a Runge-Kutta method, i.e. the c_i points, have a dramatic effect on the properties of the method. A Runge-Kutta method with constant spacings - i.e. $h_i = h_j \ \forall i,j = 1..s$ - can have up to order s. The

introduction of specific sampling locations can have a large effect on the order. We are particularly interested in the Radau-IIA class which can have order up to $2s-1$ for the same problem, and exhibits A-stability [15]. Here, the sampling locations are solutions to the equation

$$\frac{d^{s-1}}{dx^{s-1}}(x^{s-1}(1-x)^s) = 0.$$

Project methodology

To test the accuracy of my right-hand side, I followed the initial conditions and right-hand side of Giorgini(2008) [12] as closely as possible. For the positions and velocities of planets, I used NASA JPL's DE405, formerly their general purpose recommended ephemeris [11], which was used in [12]. Masses of planets were obtained from JPL's HORIZONS system.

The relevant initial conditions, given in osculating Keplerian elements with respect to the Sun, in the ecliptic J2000 frame, are given below:

Osculating element	Value	Units
Eccentricity	0.1910573105795565	(none)
Perihelion radius	0.7460599319224038	AU
Longitude of ascending node	204.45996801109067	Degrees
Argument of perihelion	126.39643948747843	Degrees
Inclination	3.33132242244163	Degrees
Mean anomaly	61.41677858002747	Degrees
Epoch	2453979.5	Julian Days

To convert between Keplerian elements and Cartesian coordinates, I used the `spiceypy` library, a wrapper for NASA JPL's SPICE library. To convert between the ecliptic and equatorial J2000 frames, I used the J2000 ecliptic obliquity constant used in the Astronomical Almanac, 2007 - the year the paper was written. Finally, I used the DE405 position of the sun to convert from heliocentric (centred at the Sun) to solar system barycentric (centred at the centre of mass of the Solar sysytem) coordinates.

The well-known Newtonian formula for the gravitational influence of n bodies on an object of interest is:

$$\ddot{\mathbf{r}} = G\sum_{i=1}^{n}\frac{m_i(\mathbf{r}_i - \mathbf{r})}{r_i^3}.$$

Here,

- m_i represents the mass of body i.
- $\mathbf{r}_i$ represents the distance between body i and the Solar System barycentre.
- $\mathbf{r}$ represents the distance between the object of interest and Solar System barycentre.

- $r_i = |\mathbf{r}_i - \mathbf{r}|$.
- G represents the Newtonian gravitational constant.

The full relativistic n-body gravity model can be written as a power series in $\frac{1}{c^2}$, where c is the speed of light in a vacuum. The zeroth-order terms (i.e. independent of c) are just equal to the Newtonian formula, and the Einstein-Infeld-Hoffmann, or EIH, equations, give the zeroth and first order terms of this series [20] - i.e. up to $\mathcal{O}(\frac{1}{c^2})$. The EIH equations are given below:

$$\begin{aligned}\ddot{\mathbf{r}} = G\sum_{i=1}^{n}\frac{m_i(\mathbf{r}_i-\mathbf{r})}{r_i^3}\Big(1 - \frac{2(\beta+\gamma)}{c^2}G\sum_{j=1}^{n}\frac{m_j}{r_j} - \frac{2\beta-1}{c^2}G\sum_{j=1,j\neq i}^{n}\frac{m_j}{r_{ij}} + \frac{\gamma|\dot{\mathbf{r}}|^2}{c^2} \\ + \frac{(1+\gamma)|\dot{\mathbf{r}}_i|^2}{c^2} - \frac{2(1+\gamma)}{c^2}\dot{\mathbf{r}}\cdot\dot{\mathbf{r}}_i - \frac{3}{2c^2}[\frac{(\mathbf{r}-\mathbf{r}_i)\cdot\dot{\mathbf{r}}_i}{r_i}]^2 + \frac{1}{2c^2}(\mathbf{r}_i-\mathbf{r})\cdot\ddot{\mathbf{r}}_i\Big) \\ + G\sum_{i=1}^{n}\frac{m_i}{c^2 r_i}(\frac{3+4\gamma}{2}\ddot{\mathbf{r}}_i + \frac{\{[\mathbf{r}-\mathbf{r}_i]\cdot[(2+2\gamma)\dot{\mathbf{r}} - (1+2\gamma)\dot{\mathbf{r}}_i]\}(\dot{\mathbf{r}}-\dot{\mathbf{r}}_i)}{r_i^2})\end{aligned}$$

.

In this equation, some extra terms are required:

- $\mathbf{r}_{ij}$ represents the displacement of body i from body j.
- β and γ are called the *Post-Newtonian Parameters*, which are both equal to one for these equations [17].

A less complete, but less time-consuming, version of this equation can be found in [5]. The principle here is that we only include terms proportional to the mass of the Sun - and assume the other planetary masses are sufficiently relatively small to be ignored. Taking m_0 as the mass of the Sun, this leaves us with a rather nicer looking equation for $\ddot{\mathbf{r}}$:

$$\ddot{\mathbf{r}} = G\sum_{i=1}^{n}\frac{m_i(\mathbf{r}_i-\mathbf{r})}{r_i^3} + \frac{Gm_0}{c^2 r_i^3}\Big((4\frac{Gm_0}{r_i} - \dot{\mathbf{r}_i}^2)\mathbf{r}_i + (\mathbf{r}_i\cdot\dot{\mathbf{r}_i})\dot{\mathbf{r}_i}\Big)$$

The Standard Dynamical Model (SDM) is a commonly used approximation for the acceleration of a body in space, which uses these equations with the planets and Sun in the Solar System. The SDM used in conjunction with a JPL ephemeris can build a simple ephemeris for a body, given accurate initial data and a stable integrator.

For the body mass values, I used the ones given in the NASA HORIZONS system. For the positions and velocities of planets, I used NASA JPL's DE405 ephemeris, which was used in [12] and was NASA's recommended ephemeris from 2003 to 2013. For the accelerations of planets, I used a finite difference approximation based on their velocities.

When propagating near-Earth asteroids, the usual summary statistic of interest is the distance of closest approach, i.e. the minimum distance between

Earth and the asteroid. One problem with obtaining this is it requires the *time* of closest approach to be known very accurately - an error of one minute in the time can lead to an error of 500 km in the closest approach. This means that a completely constant time step is not suitable for this equation, as the extremely fine time steps required during approach are unfeasible to use during the entire propagation.

My program used progressively finer time steps to circumvent this problem, as follows:

- Using a constant time step analysis, the solution is known to be between JD 2453979.5 and 2462245.5 (April 8 2029 00:00 and April 18 2029 00:00). The main propagation is made from the initial conditions in 2006 to April 8, with a constant time step h.
- The solution is integrated between April 8 and April 18 using a time step $\frac{h}{100}$. At each point the distance from Earth's DE405 position is computed. The state vector y_1 with minimum distance, at time t_1, is returned. The true time of closest approach is now certain to be in the region $(t_1 - \frac{h}{100}, t_1 + \frac{h}{100})$.
- The solution is integrated from $t_1 - \frac{h}{100}$ to $t_1 + \frac{h}{100}$, with a step size of $\frac{h}{10000}$. Using the same method, we find the approximate time of closest approach t_2 to an accuracy of $\frac{h}{10000}$.
- Finally, the solution is integrated from $t_2 - \frac{h}{10000}$ to $t_2 + \frac{h}{10000}$ with step size $\frac{h}{1000000}$. This gives a solution y_3 at time t_3 to an accuracy of $\frac{h}{1000000}$.
- The relative velocity of Apophis, with respect to the Earth, is computed using y_3 and DE405. Multiplying this by $h/2000000$ gives an upper bound on the error remaining in y_3.

For example, with a step size of 1 day = 86400 seconds, the closest approach is found to an accuracy of 0.0864 seconds. The maximum error caused by this discretisation is about 320 metres. This implementation relies on

The most intuitive and direct metric for measuring integrator efficiency is time - comparing the time taken for each method to achieve a certain accuracy, for identical problems and y-ranges. This would involve adjusting the time step h - or scaling the time step h_n for methods with variable time step. The main insufficiency of this method is that different implementations of the same method can differ in time - a rough coding of a 'fast' method can take more time than a highly optimised implementation of a 'slower' method. Also, adjusting the time steps of different methods so that they arrive at a similar accuracy may be extremely time consuming, requiring a large number of runs. Nevertheless, I will measure time taken in my code, as it will be useful to compare how well implemented each integrator is.

A more consistent metric of efficiency between implementations is one involving some sort of measure of 'steps' taken. Bartram [4] uses the following formula for 'total evaluation cost':

$$TEC = F_{evals} + J_{evals} + C_{scale},$$

where F_{evals} is the number of right-hand side evaluations, J_{evals} is the number of Jacobian evaluations, and C_{scale} is a scaling factor dependent on the problem, which for Apophis will be determined upon code familiarisation.

Properties of MCMs have been investigated in recent literature [4] [21] as a proposed alternative to the current state-of-the-art. It has been concluded in [4] that MCMs are most efficient with either a low k or a low s, and are less efficient with both high k and high s. The novel contribution of my project is to directly compare various MCMs with older methods using the same specific benchmarking problem, leading to much greater clarity on the differences between the methods.

References

[1] *2004 MN4 Impact Risk*. URL: `https://web.archive.org/web/20050314032111/https://astro.uni-bonn.de/~dfischer/mirror/285neo041227.html`. (accessed 25.05.2019).

[2] K. E. Atkinson. *An Introduction to Numerical Analysis*. 2nd ed. John Wiley & Sons, 1989.

[3] D. Bancelin et al. "Asteroid (99942) Apophis: new predictions of Earth encounters for this potentially hazardous asteroid". In: *Astronomy and Astrophysics* 544.15 (2012). DOI: `10.1051/0004-6361/201117981`.

[4] P. Bartram, H Urrutxua, and Wittig. A. "Exploring the Parameter Space of Multistep Collocation Methods".

[5] G. Beutler. *Methods of Celestial Mechanics, Volume 1: Physical, Mathematical, and Numerical Principles*. Springer, 2005. ISBN: 3540407499.

[6] H. Brunner. *Collocation methods for Volterra Integral and Related Functional Equations*. Cambridge University Press, 2004. ISBN: 9780521806152.

[7] J. C. Butcher. "A history of Runge-Kutta methods". In: *Applied Numerical Mathematics* 20.3 (1996), pp. 247–260.

[8] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. 2nd ed. John Wiley & Sons, Ltd, 2008. ISBN: 978-0-470-72335-7.

[9] S. R. Chesley. "Potential impact detection for Near-Earth asteroids: the case of 99942 Apophis (2004 MN4)". In: Asteroids, comets, meteors : proceedings of the 229th Symposium of the International Astronomical Union. (Búzios, Rio de Janeiro, Brasil). 2005.

[10] B. Dachwald and R. Kahle. "Head-On Impact Deflection of NEAs: A Case Study for 99942 Apophis". In: Planetary Defense Conference. (Washington, D. C., USA). 2007.

[11] W. M. Folkner et al. "The Planetary and Lunar Ephemerides DE430 and DE431". In: *The Interplanetary Network Progress Report* 42.196 (2014).

[12] J. D. Giorgini et al. “Predicting the Earth encounters of (99942) Apophis”. In: *Icarus* 193.1 (2008), pp. 1–19. DOI: 10.1016/j.icarus.2007.09.012.

[13] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Vol. 14. Springer, 1991.

[14] *JPL Small-Body Database Browser*. URL: https://ssd.jpl.nasa.gov/sbdb.cgi?sstr=99942;cad=1. (accessed 05.04.2019).

[15] A. Prothero and A. Robinson. “On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations”. In: *Mathematics of Computation* 28.125 (1974), pp. 145–162.

[16] C. M. Rumpf, H. G. Lewis, and P. M. Atkinson. “Asteroid impact effects and their immediate hazards for human populations”. In: *Geophysical Research Letters* 44 (2017), pp. 3433–3440. DOI: 10.1002/2017GL073191.

[17] P. K. Seidelmann. *Explanatory Supplement to the Astronomical Almanac*. University Science Books, 1992.

[18] L. L. Sokolov. “On the Keyhole Positions of Apophis”. In: IAA Planetary Defense Conference. (ESA-ESRIN, Frascati, Italy). 2015.

[19] M. Trenti and P. Hut. “N-body simulations (gravitational)”. In: *Scholarpaedia* (2008). DOI: 10.4249/scholarpedia.3930. URL: http://www.scholarpedia.org/article/N-body_simulations.

[20] A. Wittig et al. “Propagation of large uncertainty sets in orbital dynamics by automatic domain splitting”. In: *Celestial Mechanics and Dynamical Astronomy* 122 (3 2015), pp. 239–261. DOI: 10.1007/s10569-015-9618-3.

[21] C. Zhang and S. Vandewalle. “General Linear Methods for Volterra Integro-Differential Equations with Memory”. In: *SIAM Journal on Scientific Computing* 27.6 (2006), pp. 2010–2031.