PAPER AAS 07-XXX

COMPARISON OF FIXED AND VARIABLE TIME STEP TRAJECTORY INTEGRATION METHODS FOR CISLUNAR TRAJECTORIES

Michael W. Weeks and Stephen W. Thrasher
EG/Aeroscience and Flight Mechanics Division
NASA Johnson Space Center
Houston, Texas 77058

17th AAS/AIAA Space Flight Mechanics Meeting

Sedona, AZ January 28 – February 1, 2007

AAS Publications Office, P.O. Box 28130, San Diego, CA 92198

AAS 07-XXX

Comparison of Fixed and Variable Time Step Trajectory Integration Methods for Cislunar Trajectories

Michael W. Weeks[*] and Stephen W. Thrasher[†]

Due to the nonlinear nature of the Earth-Moon-Sun three-body problem and non-spherical gravity, CEV cislunar targeting algorithms will require many propagations in their search for a desired trajectory. For on-board targeting especially, the algorithm must have a simple, fast, and accurate propagator to calculate a trajectory with reasonable computation time, and still be robust enough to remain stable in the various flight regimes that the CEV will experience. This paper compares Cowell's method with a fourth-order Runge-Kutta integrator (RK4), Encke's method with a fourth-order Runge-Kutta-Nyström integrator (RKN4), and a method known as Multi-Conic. Additionally, the study includes the Bond-Gottlieb 14-element method (BG14) and extends the investigation of Encke-Nystrom methods to integrators of higher order and with variable step size.

INTRODUCTION

Because of the nonlinear nature of the Earth-Moon-Sun three-body problem and non-spherical gravity, cislunar targeting algorithms require many cislunar propagations in their search for a desired trajectory. For on-board targeting especially, the algorithm must have a simple, fast, and accurate propagator to calculate a trajectory with reasonable computation time, and still be robust enough to remain stable in the various flight regimes that the CEV will experience (LEO rendezvous, LLO rendezvous, LLO orbit, etc.).

Reference [1] compares Cowell's method with a fourth-order Runge-Kutta integrator (RK4), Encke's method with a fourth-order Runge-Kutta-Nyström integrator (RKN4), and a method known as Multi-Conic. This new study includes the Bond-Gottlieb 14-element method and extends the investigation of Encke-Nystrom methods to integrators of higher order and with variable step size. D'Souza found that Encke's method outperformed Cowell's method and the Multi-Conic method, so these two methods are excluded from this report.

[*] CEV Flight Dynamics Orbit GNC Engineer, NASA-JSC
[†] CEV Flight Dynamics Orbit GNC Engineer, C. S. Draper Laboratory

INTEGRATION METHODS

Encke Method

The classic form of Encke's special perturbation method solves the initial value problem of celestial mechanics by analytically propagating a two-body "reference" orbit and numerically integrating the deviations from two-body motion. These two procedures are performed for the same time interval or step size. Since the two-body solution is "exact", the time interval used is usually a function of the numerical integration technique and the magnitude of the perturbations. After each propagation interval, the errors between the reference orbit and the perturbed orbit are evaluated. When these errors become "large", a process known as rectification of the orbit is performed. This involves adding the integrated deviations to the two-body orbit to produce a new reference orbit. After rectification, the algorithm is reinitialized with the orbital elements of the new osculating orbit and the process is repeated until either the errors become large once more or the final time is reached. The motion of the vehicle with respect to the primary body is governed by

$$\ddot{\mathrm{r}}_{PV} + \frac{\mu_P}{r_{PV}^3}\mathrm{r}_{\mathrm{PV}} = a_d \tag{1}$$

where a_d is the disturbing acceleration from the moon, sun, drag, etc., with respect to the primary body. Instead of integrating this directly, the Encke method involves integrating a perturbation,

$$\delta(t) = \boldsymbol{r_{PV}}(t) - \boldsymbol{r_{PV(C)}}(t) \tag{2}$$

where

$$\ddot{\delta} = -\frac{\mu_P}{r_{PV(C)}^3}\left[f(q_C)\mathrm{r}_{\mathrm{PV}}(t) + \delta(t)\right] + a_d \tag{3}$$

$$q = \frac{(\delta - 2\mathrm{r}_{\mathrm{PV}})\cdot\delta}{r_{PV}^2} \tag{4}$$

$$f(q) = \frac{3 + 3q + q^2}{1 + (1+q)^{3/2}} \tag{5}$$

where $\boldsymbol{r_{PV}}$ is the position of the vehicle (V) with respect to the primary body (P), and $\boldsymbol{r_{PV(C)}}$ denotes the conic two-body solution. Since the Encke method integrates only the perturbations which are much smaller in magnitude than the actual states, the computation precision can be much greater than methods that integrate the full equations of motion. Encke's method works well with Nyström integrators, which are specialized for second-order ordinary differential equations with no first-order terms, like Equation 3.

Bond-Gottlieb-Fraietta 14-Element Method

BG14 is a perturbation method that uses variation of parameters to provide differential equations for a set of orbital elements. These equations are uniformly valid for elliptic, parabolic and hyperbolic motion. Based on the work of Sperling in 1961, BG14 produces a linearized and regularized set of differential equations of motion of the two-body problem using a two step process. In the first step the independent variable is changed from time t to fictitious time s using the Sundmann transformation.

$$\frac{dt}{ds} = r \tag{6}$$

Under this transformation, with uniform steps in time *s*, time *t* changes slower near the primary body and faster far from the primary body. Changing the independent variable transforms the two-body equation of motion from (7) to (8) as shown below.

$$\ddot{\boldsymbol{r}} + \frac{\mu}{r^3}\boldsymbol{r} = \boldsymbol{f} = -\frac{\partial V(\boldsymbol{r},t)}{\partial \boldsymbol{r}} + \boldsymbol{\varphi} \tag{7}$$

$$\boldsymbol{r}'' = r^2 \boldsymbol{f} - \mu\, \boldsymbol{p} + \left[\dot{\boldsymbol{r}} \cdot \dot{\boldsymbol{r}} - \frac{2\mu}{r}\right]\boldsymbol{r} \tag{8}$$

where $\boldsymbol{f}$ is the sum of all perturbing forces, V is the sum of perturbing potentials, $\boldsymbol{\varphi}$ is the sum of perturbing forces not derivable from a potential, and $\boldsymbol{p}$ is the eccentricity vector, a conserved quantity. The coefficient of $\boldsymbol{r}$ in (8) is twice the energy integral of Keplerian motion, another conserved quantity. The goal is to embed conserved quantities into the equations of motion so that they remain constant over a range of perturbations. Although there are several options for this, Bond and Gottlieb chose to embed the Jacobian integral here as seen in (9) as an element to replace the total energy in a reformulation of the differential equations of motion.

$$\alpha_J = \dot{\mathrm{r}} \cdot \dot{\mathrm{r}} - \frac{2\mu}{r} + 2V(\mathrm{r},t) - 2\sigma \tag{9}$$

At the end, a transformed set of 14 regularized differential equations is produced.

As References [2] and [3] describe in detail, the process for each integration step is to begin with the initial time and cartesian state. From that, the required constants, Stumpff functions, and BG14 elements can be computed – thus producing the transformed differential equations of the BG14 elements. Once those equations are integrated, the elements can the transformed back into the new cartesian state and the process repeated. Due to the variation of parameters method and setup of the BG14 elements, the BG14 integration takes more time than a standard cartesian integration. However, it is intended to be used with relatively large step sizes that will compensate for the extra computations in each integration step. With sufficiently large step sizes, BG14 propagation can provide a more accurate answer with a reduction in total run time as Gottleib states in References [2] and [3].

One difficulty of BG14 which can be seen immediately is due to the Sundmann transformation of the independent variable as seen in Equation 6. Since the time t is transformed into s (which is a function of the radius vector) we are no longer able to take fixed steps in time, only in s. For this reason, it is more difficult to mesh BG14 easily with other methods, for example, a navigation propagator that requires fixed steps in time. Another problem with operating in s is finding a final state at some desired t, which must involve solving for the final step size in s to yield the correct t or using another propagation method for the final step.

INTEGRATORS

For both propagation methods, once the appropriate differential equations are formulated, an integrator must numerically integrate the set of differential equations from the current step to the next step in order to compute an updated state. Several integrators were chosen for comparison. The integrators were RK45, Adams-Bashforth-Moulton, RK87, RKN4, RKN64, and RKN86. Variable step integrators were tested with both fixed and variable step size.

While the Runge-Kutta integrators are typically used with a constant step size as in D'Souza's work, for this analysis fixed and variable steps were employed. Variable-step integration methods change the step size at each step to meet some tolerance for the local error. Comparing two integrations of different orders at each step gives an estimate of the local error, and the step size is adjusted so that the estimated error at the next step will be approximately equal to the tolerance. For instance, the RK45 method estimates the local error by comparing the result of fourth-order and fifth-order Runge-Kutta formulas.

A brief description of each of the integrators is presented in the following section. Vallado and Battin present more detailed descriptions [4, 5].

ODE45 (RK45)

The *ode45* integrator is a MATLAB® function. It is based on an explicit Runge-Kutta (4,5) formula known as the Dormand-Prince formulas (Reference [6]). It is a single-step method, i.e., in computing *y(tn)*, it needs only the solution at the immediately preceding time point, *y(tn-1)*. In general engineering practice, *ode45* is the usually applied as a first try for most problems.

ODE113 (Adams-Bashforth-Moulton)

The *ode113* integrator is a variable order Adams-Bashforth-Moulton Predictor-Corrector (PECE) solver that comes standard with MATLAB®. It may be more efficient than *ode45* at stringent tolerances and when the ODE function is particularly expensive to evaluate. It is a multistep solver and needs the solutions at several preceding time points to compute the current solution.

ODE87 (RK87)

The *ode87* integrator is an explicit Runge-Kutta single-step method. It integrates the system of differential equations using eighth-seventh order Dorman and Prince formulas (Reference [6]). As such, it is an 8th-order accurate integrator and requires 13 function evaluations per integration step.

Runge-Kutta-Nyström Methods

Runge-Kutta-Nyström methods deal with a special class of second order differential equations whose right hand side is not an explicit function of the first derivative of the independent variable. With RKN integrators, it is possible to achieve a higher order of agreement with the Taylor series expansion of the solution for a given number of evaluations of the right hand side than could be expected for the general RK integrator. Because of this, two evaluations of the RKN method could achieve an error of fourth order, three evaluations could achieve an error of fifth order, four evaluations could achieve an error of sixth order, etc.

RKN86

The *rkn86* integrator is a Runge-Kutta-Nyström solver which integrates a special system of second order ordinary differential equations using an 8-stage Runge-Kutta-Nyström pair of orders 8 and 6. The method advances using the higher order formula (using local extrapolation). The coefficients of the Runge-Kutta-Nyström pair are taken from Reference [7].

RKN64

The *rkn64* integrator is a Runge-Kutta-Nyström solver which integrates a special system of second order ordinary differential equations using a 6-stage scheme with 4th order accuracy and 5 implicit stages. More information on *rkn64* can also be found from Reference [7].

ASSUMPTIONS AND INITIAL CONDITIONS

The four trajectories which were evaluated for each method under investigation began after the TLI maneuver and were propagated from 0 to 10,000 seconds, 0 to 3.18 days (just prior to a sphere of influence frame switch), 3.18 to 4 days, and 0 to 4 days. Spherical gravity was assumed for both the Earth and the Moon. The initial state and parameters are listed in Table 1.

ECI X-POSITION (m)	544259.156
ECI Y-POSITION (m)	6180337.037
ECI Z-POSITION (m)	2475349.698
ECI X-VELOCITY (m/s)	-10339.931481
ECI Y-VELOCITY (m/s)	-77.810392
ECI Z-VELOCITY (m/s)	3258.388684
μ_E (m^3/s^2)	$3.986004414996968 \times 10^{14}$
μ_M (m^3/s^2)	$4.902799999996766 \times 10^{12}$
μ_S (m^3/s^2)	$1.327124400417518 \times 10^{20}$
ET_0 (s)	$5.853008442 \times 10^{8}$

Table 1: Trajectory Initial Conditions and Parameters

The test runs were performed in MATLAB® (version 7.1.0.246, service pack 3) using a Dell computer (3.19 GHz Intel Pentium 4) running Windows XP (version 2002, service pack 2). The final state truth values were chosen to be the state obtained by a fourth-order Encke-Nyström method with a 1 second step size. For the 3.18 to 4 day trajectory, the truth value after 3.18 days served as the initial condition.

COMPARISON

This study examines the Encke method with RKN4, RKN64, and RKN86 integration and the BG14 method with RK45, RK87, and ABM integration. Nyström integration requires a second-order ODE with no first-order derivatives and therefore does not apply to BG14. Nyström methods are faster than their regular Runge-Kutta counterparts, so RK45 and RK87 were not considered with Encke's method. ABM or some second-order version of ABM might perform well with Encke's method, but it was not included in this study.

To overcome the problem of achieving the state at a desired time while integrating in fictitious time, s, Encke with RKN64 was chosen to complete the final timestep of each BG14 run. This study examines the Gottlieb Constant Energy Method (GCEM) version of BG14 [3]. The BG14 functions did not use the sliding origin technique described in [2], and a brief investigation showed no loss of accuracy for the trajectories of this study.

The Encke propagation algorithm with RKN4 came from Battin [5]. Rectification is performed at every timestep. Encke with RKN64 and RKN86 used a function from the MATLAB® Central File Exchange called rkn86 [8], with RKN64 constants coming from [5]. Fixed-step RKN6 and RKN8 integration involved running RKN64 and RKN86 with a fixed step size, resulting in one extra matrix multiplication per iteration. BG14 used built-in MATLAB® integrators *ode45* and *ode113* for RK45 and ABM, respectively. For RK87, BG14 used a function from the MATLAB® Central File Exchange called *ode87* [9].

Step sizes and variable-step tolerances were chosen to compare the methods across a performance range from fast, inaccurate propagations to slow, accurate propagations. The BG14 algorithms use MATLAB integrators with built-in relative tolerance checking, while the Encke methods check absolute error tolerance. Step size, error tolerance, number of iteration, position error, velocity error, and run time are in the appendix for selected runs. The MATLAB® *cputime* function returned each run's computation time.

The following logarithmic plots summarize the results for easy comparison. These data are meant to give an overall idea of performance. On another trajectory, with a different final time, or with a slightly different step size or error tolerance, the performance might differ.

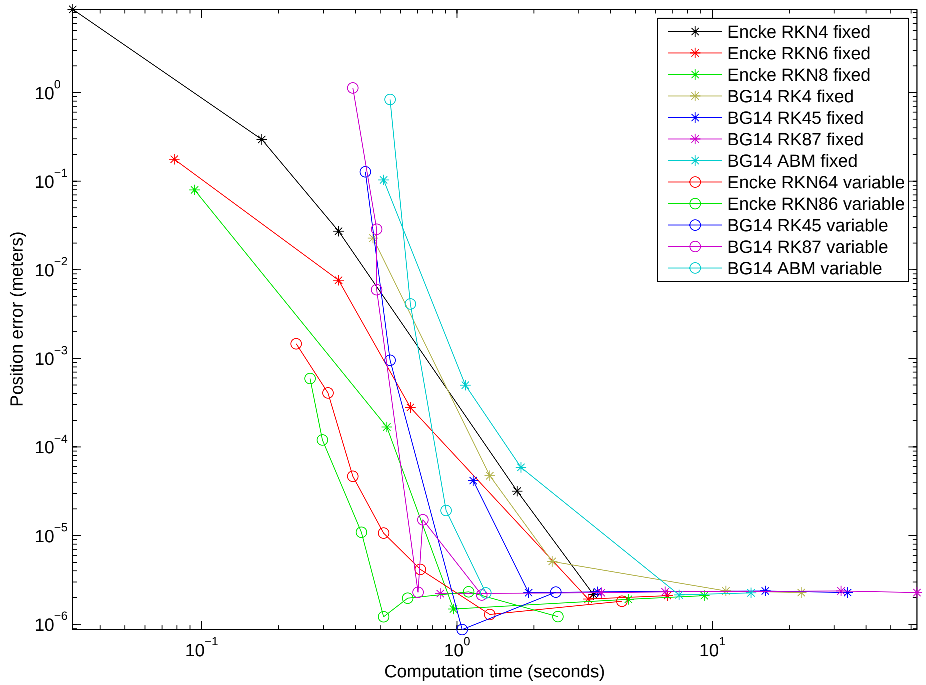

Figure 1: Position error for 0 to 10,000 seconds

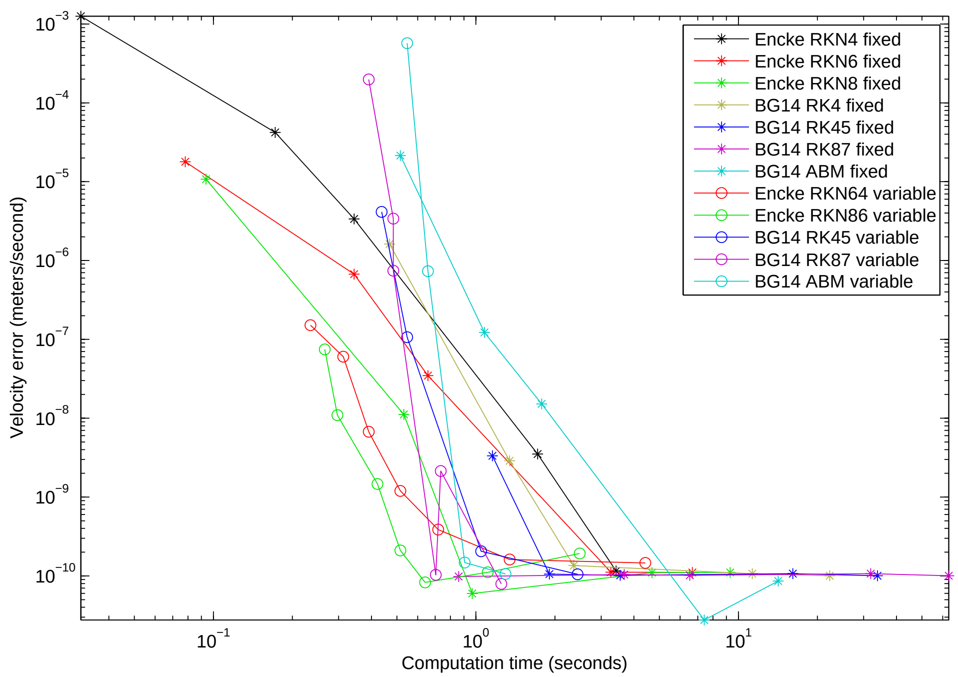

Figure 2: Velocity error for 0 to 10,000 seconds

Encke high-order variable-step performed best on the 10000 second run. For both Encke and BG14 variable-step methods, even very few steps yielded small errors over such a short run. Because BG14 requires more computation per integration step, Encke was faster.

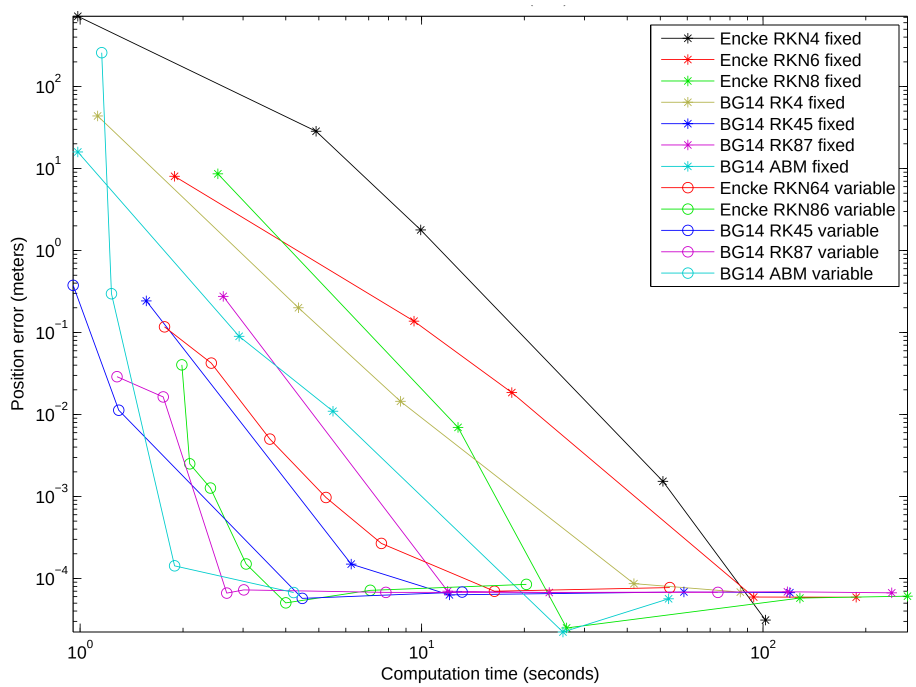

Figure 3: Position error for 0 to 274,871 seconds

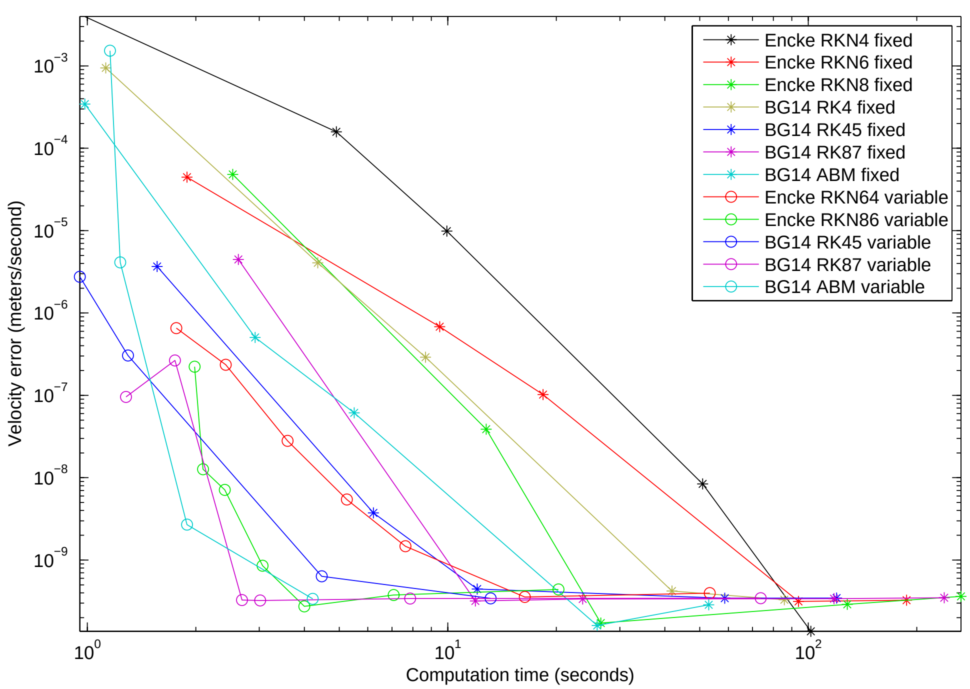

Figure 4: Velocity error for 0 to 274,871 seconds

From initial position to the sphere of influence interface, the picture changes. Over this region, BG14 requires fewer integration steps to achieve the same accuracy as the Encke methods.

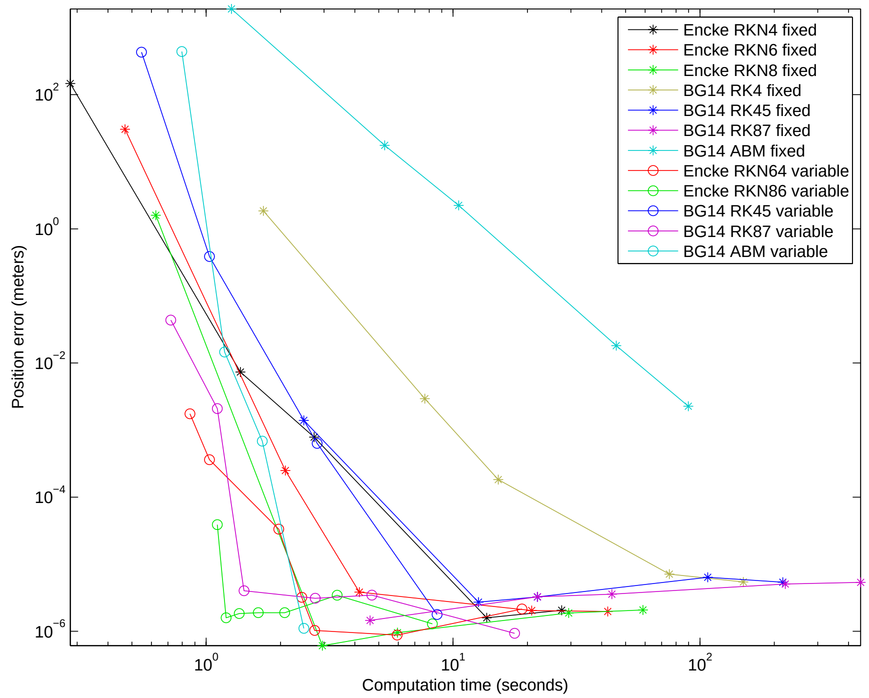

Figure 5: Position error for 274,871 to 345,600seconds

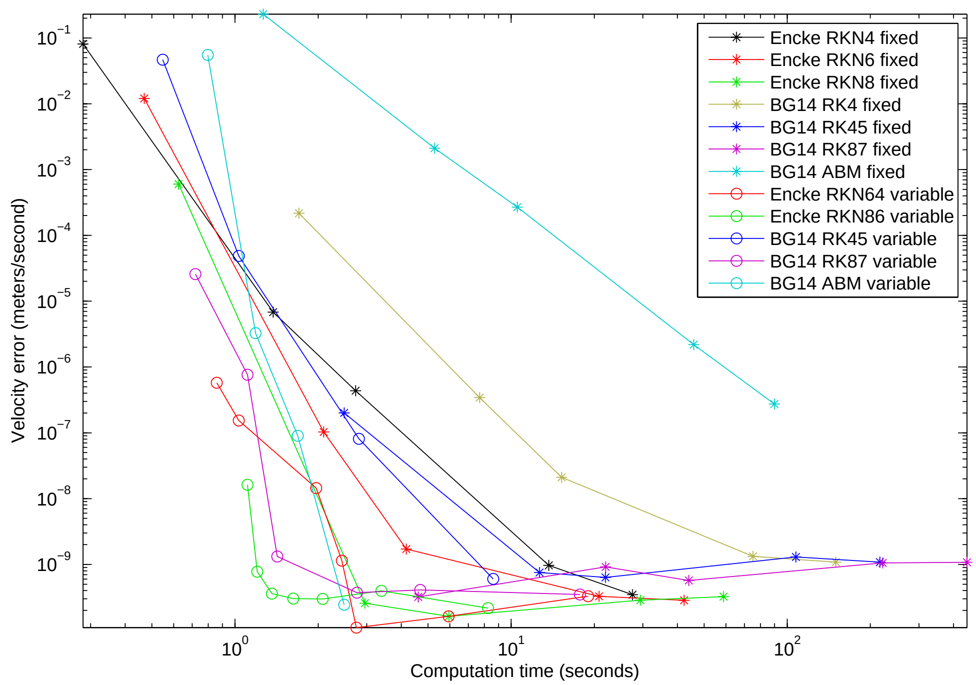

Figure 6: Velocity error for 274,871 to 345,600seconds

From SOI to the end of the 4 day trajectory, the BG14 step size decreased to the point where its performance was once again comparable to Encke methods. BG14 with fixed-step ABM performed poorly here for unknown reasons.

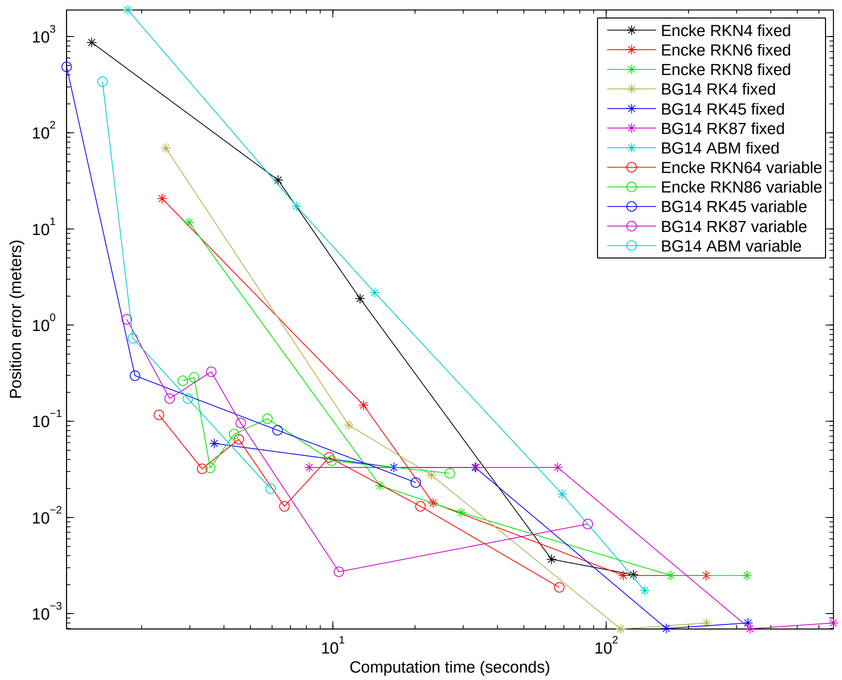

Figure 7: Position error for 0 to 345,600seconds

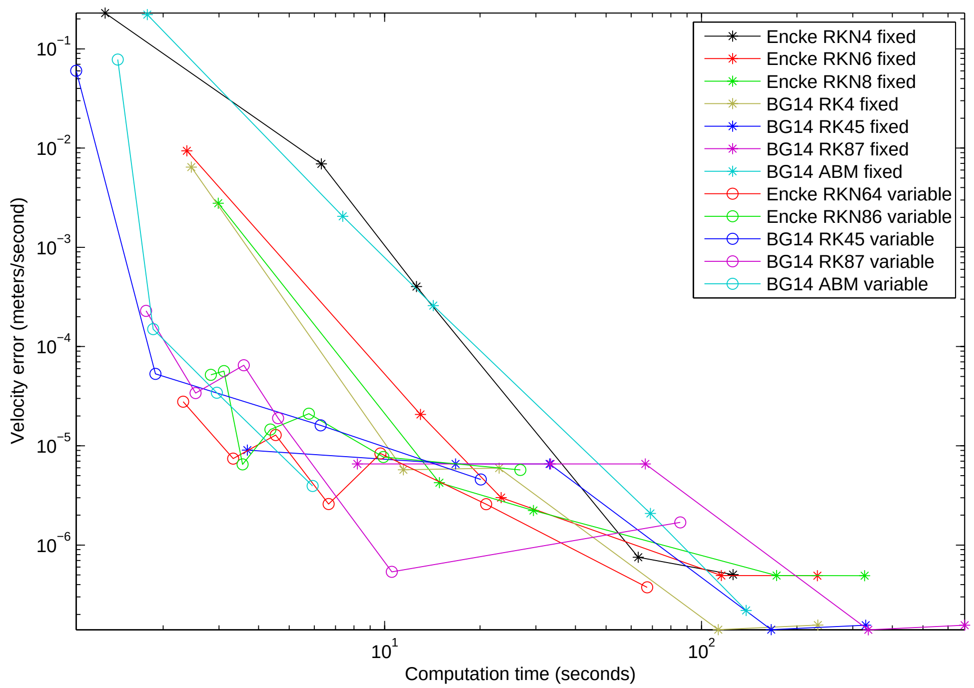

Figure 8: Velocity error for 0 to 345,600seconds

Performance is comparable between high-order variable-step BG14 and Encke methods over the 4 day trajectory. Inaccurate switching of sphere of influence caused the jaggedness of the best performing results. For very short timesteps, Encke with RK4 performs as well as its high-order counterparts.

CONCLUSIONS

In general, both Encke-Nyström and BG14 propagation benefit from higher-order integrators with variable step size. While the simpler Encke methods performed better over small timescales, BG14 performed better with long trajectories near Earth. Overall, high-order variable-step-size Encke and BG14 were both comparable over the entire cislunar reference trajectory.

With large step sizes, some runs over-stepped the edge of the moon's sphere of influence by as much as 0.1% of the Earth-moon distance. This was significant enough to increase the final trajectory error noticeably on the 4 day trajectory, causing the jaggedness in the error vs. computation time plots. These results show that within a cislunar targeting algorithm, accurately determining the SOI interface time could reduce error for integration involving large steps.

In performing this analysis, it was our intent to look at the "big picture" as it relates to CEV and propagators. Obviously, propagators and integrators will be needed in both the Ground Control software and the Mission Planning software, as well as the actual CEV GNC flight software. Therefore, as a metric for evaluating the various options that exist, we considered more than just the speed and accuracy of a particular method. Consideration was also given to the simplicity of each method, the robustness of each method, and the applicability of each method to be implemented into future flight software. Ideally, the optimal solution would be to have the same fast, accurate, and robust propagator which could be used on both the flight software and the ground software for all flight regimes. For guidance and navigation flight software, it is necessary for propagators to work in synch with other flight software modules, and they must therefore run with fixed time steps.

As shown earlier, BG14 operates in fictitious time, making it impractical for fixed steps in time. It can be seen in the results displayed in Figures 1 – 8 that BG14 is indeed an efficient propagator and can achieve an accurate solution with perhaps fewer iterations than Encke. However, we feel that the differences in performance are not significant enough to warrant the complexities involved with BG14 with regards to the fictitious time. In light of these facts, Encke with an RKN86 is the recommended choice for cislunar propagation. While Encke with a RKN4 performs as well as RKN6 or RKN8 when step sizes are small (less than 100 seconds), an eighth-order variable time step Encke-Nyström method is recommended for step sizes above 100 seconds.

Several avenues would benefit from future study. Several other integrators exist, and in particular, Encke with Gragg-Bulirsch-Stoer integration and BG14 with Bulirsch-Stoer integration merit study. Because ephemeris calculation dominated computation time, optimizing by pre-loading the Earth-moon and Earth-sun positions and velocity or interpolating their positions in some efficient way might significantly reduce computation. Finally, BG14's equations can be tailored for higher-order gravity models of the primary body. With more efficient ephemeris calculation or higher-order gravity, the comparison of this study should be repeated.

APPENDIX

Selected entries from 0 to 274871 second (SOI) run

Propagator	Pos err (m)	Vel err (m/s)	Compute time (s)	Number of steps	Stepsize
BG14 ABM fixed	9.00E-02	5.04E-07	2.9	175	1.00E-05
					Tolerance
Encke RKN64 variable	5.01E-03	2.80E-08	3.6	108	1.00E-01
	4.22E-02	2.36E-07	2.4	74	1.00E+00
Encke RKN86 variable	5.04E-05	2.75E-10	4.0	86	1.00E-03
	1.51E-04	8.53E-10	3.1	65	1.00E-02
	1.27E-03	7.11E-09	2.4	51	1.00E-01
	2.50E-03	1.27E-08	2.1	44	1.00E+00
	4.03E-02	2.23E-07	2.0	40	1.00E+01
BG14 RK45 variable	5.72E-05	6.34E-10	4.5	110	1.00E-11
	1.13E-02	3.04E-07	1.3	27	1.00E-08
BG14 RK87 variable	7.25E-05	3.23E-10	3.0	35	1.00E-14
	6.63E-05	3.27E-10	2.7	21	1.00E-12
	1.63E-02	2.66E-07	1.8	12	1.00E-10
	2.89E-02	9.58E-08	1.3	8	1.00E-08
BG14 ABM variable	6.74E-05	3.39E-10	4.2	262	2.23E-14
	1.42E-04	2.69E-09	1.9	98	1.00E-11

Selected entries from 4 day run

Propagator	Pos err (m)	Vel err (m/s)	Compute time (s)	Number of steps	Stepsize
BG14 RK45 fixed	5.89E-02	9.06E-06	3.7	108	5.00E-05
BG14 RK87 fixed	3.32E-02	6.57E-06	8.2	108	5.00E-05
					Tolerance
Encke RKN64 variable	4.21E-02	8.35E-06	9.7	322	1.00E-03
	1.31E-02	2.59E-06	6.7	220	1.00E-02
	6.49E-02	1.29E-05	4.5	150	1.00E-01
	3.20E-02	7.43E-06	3.3	103	1.00E+00
	1.17E-01	2.79E-05	2.3	72	1.00E+01
Encke RKN86 variable	3.89E-02	7.71E-06	9.9	231	1.00E-05
	1.06E-01	2.11E-05	5.8	131	1.00E-03
	7.35E-02	1.46E-05	4.4	99	1.00E-02
	3.28E-02	6.50E-06	3.6	77	1.00E-01
	2.86E-01	5.66E-05	3.1	64	1.00E+00
	2.64E-01	5.20E-05	2.8	56	1.00E+01
BG14 RK45 variable	8.08E-02	1.61E-05	6.3	191	1.00E-11
	2.98E-01	5.31E-05	1.9	48	1.00E-08
BG14 RK87 variable	9.57E-02	1.90E-05	4.6	64	1.00E-14
	3.27E-01	6.48E-05	3.6	37	1.00E-12
	1.72E-01	3.40E-05	2.5	22	1.00E-10
BG14 ABM variable	1.99E-02	3.95E-06	5.9	431	2.23E-14
	1.73E-01	3.42E-05	3.0	191	1.00E-11
	7.29E-01	1.50E-04	1.9	112	1.00E-08

Selected entries from 274871 seconds (SOI) to 4 days

Propagator	Pos err (m)	Vel err (m/s)	Compute time (s)	Number of steps	Stepsize
Encke RKN4 fixed	7.84E-04	4.35E-07	2.7	142	500
	7.33E-03	6.81E-06	1.4	71	1000
Encke RKN6 fixed	2.49E-04	1.03E-07	2.1	71	1000
Encke RKN8 fixed	6.11E-07	2.57E-10	3.0	71	1000
BG14 RK45 fixed	1.39E-03	2.01E-07	2.5	74	5.00E-05
					Tolerance
Encke RKN64 variable	1.03E-06	1.10E-10	2.8	94	1.00E-03
	3.21E-06	1.14E-09	2.4	64	1.00E-02
	3.32E-05	1.45E-08	2.0	44	1.00E-01
	3.61E-04	1.53E-07	1.0	30	1.00E+00
	1.75E-03	5.76E-07	0.9	21	1.00E+01
Encke RKN86 variable	1.89E-06	2.99E-10	2.1	47	1.00E-03
	1.89E-06	3.02E-10	1.6	35	1.00E-02
	1.84E-06	3.60E-10	1.4	27	1.00E-01
	1.59E-06	7.72E-10	1.2	21	1.00E+00
	3.89E-05	1.62E-08	1.1	17	1.00E+01
BG14 RK45 variable	6.35E-04	8.09E-08	2.8	82	1.00E-11
BG14 RK87 variable	3.11E-06	3.71E-10	2.8	29	1.00E-14
	4.02E-06	1.32E-09	1.4	17	1.00E-12
	2.09E-03	7.65E-07	1.1	10	1.00E-10
BG14 ABM variable	1.10E-06	2.44E-10	2.5	172	2.23E-14
	6.83E-04	9.02E-08	1.7	94	1.00E-11

Encke RKN4 1-second step size state at 10000 seconds

ECEI X Position (m)	-35585619.555396
ECEI Y Position (m)	-33776924.129816
ECEI Z Position (m)	-3146585.266364
ECEI X Velocity (m)	-1604.28631476683
ECEI Y Velocity (m)	-3317.33506222760
ECEI Z Velocity (m)	-910.94818262126

Encke RKN4 1-second step size state at 274871 seconds (SOI)

ECEI X Position (m)	-88827555.502063
ECEI Y Position (m)	-342708315.002066
ECEI Z Position (m)	-119444456.552843
ECEI X Velocity (m)	3.40170646523
ECEI Y Velocity (m)	-530.35692824082
ECEI Z Velocity (m)	-241.83889468881

Encke RKN4 1-second step size state at 4 days

ECEI X Position (m)	-97240190.506115
ECEI Y Position (m)	-370110277.221061
ECEI Z Position (m)	-129807748.308172
ECEI X Velocity (m)	-119.59826232210
ECEI Y Velocity (m)	681.70531103231
ECEI Z Velocity (m)	745.06798043964

REFERENCES

[1] D'Souza, C., "*An Analysis of Trajectory Integration Methods for Cislunar Trajectories*," EG-CEV-06-32, May 2006.

[2] Bond, V.R. and Fraietta, M.F., "*Elimination of Secular Terms from the Differential Equations for the Elements of Perturbed Two-Body Motion*." NASA - CP 3123, May 1991, pp. 265-283.

[3] Gottlieb, R. G., "*Variation of Parameters Methods Embedding the Jacobi Integral Resulting from Energy Parameter Selection*." AAS Paper AAS99-202, AAS/AIAA Space Flight Mechanics Meeting, Breckenridge, CO, February 1999.

[4] Vallado, D. A., "Fundamentals of Astrodynamics and Applications," Space Technology Series, McGraw-Hill, New York, NY, 1997.

[5] Battin, R.H., *An Introduction to the Mathematics and Methods of Astrodynamics*, American Institute of Aeronautics and Astronautics, 1987.

[6] Prince, P. J. and Dorman, J.R. "*High order embedded Runge-Kutta Formulae*." Journal of .Computational & Applied Mathematics., Vol. 7. 1981, pp. 67-75.

[7] Papakostas, S. N. and Tsitouras, C.' "*High phase-lag order Runge-Kutta and Nystrom pairs*", SIAM Journal of Scientific Computing Vol. 21, 1999, pp. 747-763.

[8] Tsitouras, T., Deveopler, *rkn86* integration function for MATLAB, Matlab Central File Exchange website, http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3153&objectType=file

[9] Govorukhin, V., Developer, *ode87* integration function for MATLAB, Matlab Central File Exchange website, http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3616