N-body simulations: the performance of eleven integtrators

P.W. Sharp*†

September 16, 2003

Abstract

We compare the performance of the Störmer methods of orders nine and thirteen, two optimized symplectic explicit Runge-Kutta-Nyström methods of orders four and seven, and the integrators `DIVA`, `STEP`, `DXRK8`, `RKSUITE`, `ODEX`, `ODEX2`, `RKNINT` on four challenging N-body simulations of the Solar System.

Keywords: integrators, comparisons, N-body simulations

2000 MSC: Primary - 65L05, secondary - 70F10

1 Introduction

N-body simulations of the Solar System are a rich source of non-stiff initial value problems in ordinary differential equations. The problems range from small simulations of two bodies over short intervals of time to simulations of tens of thousands of bodies over 10^9 years requiring CPU time counted in months on the computer-of-date (see for example [9]).

In [20], we presented four N-body problems, all with long intervals of integration, two with detailed interactions between the bodies, and illustrated their use as test problems by comparing the performance of the integrators `DIVA` [11, 12], `STEP` [19], `RKSUITE` [2] and `RKNINT` [1]. We have since greatly extended this work by finding reference solutions to the problems for intervals of integration up to ten times as long as in [20], and by comparing the performance of a bigger and more diverse collection of integrators, eleven in all. We present a summary of these comparisons here. We have two aims, both to do with establishing how effective the integrators are at finding accurate solutions to the problems. One aim is to compare the relative efficiency of one-step and multi-step non-symplectic integrators. The other is to measure to what extent, if any, symplectic

*Department of Mathematics, University of Auckland, Private Bag 92019, Auckland, NEW ZEALAND, sharp@scitec.auckland.ac.nz

†This research was partly supported by the University of Auckland Research Committee.

explicit Runge-Kutta Nyström (ERKN) methods are more efficient than non-symplectic ERKN methods on N-body simulations with long intervals of integration. On this last point, Calvo and Sanz-Serna [5] said: *From this experiment we conclude that if accurate solutions are needed, even for long integration times, a high-order standard code may easily be a better choice than a symplectic algorithm.*

The first two problems have the bodies acting through Newtonian gravitational forces and we compare the integrators `DIVA` [11, 12], `STEP` [19], `DXRK8` [13, 14], `ODEX` [10] (pp. 484-486), `ODEX2` [10] (pp. 487), `RKSUITE` [2], `RKNINT` [1], `CS4`, `CS7`, `S9` and `S13` on them. The first seven integrators are variable-stepsize, the next two are fixed-stepsize symplectic ERKN integrators, and the last two fixed-stepsize Störmer integrators. We wrote the fixed-stepsize integrators for the current work. We decided against a variable-stepsize implementation for the Störmer methods because most N-body simulations done with them use a fixed stepsize. The remaining two problems have post-Newtonian interactions and we compare `DIVA`, `STEP`, `RKSUITE`, `DXRK8` and `ODEX` on them.

We measured the computational effort using the number of derivative evaluations and the amount of CPU time. The number of evaluations is emphasized when comparing similar methods, the CPU time when comparing dissimilar methods. To assess how the relative CPU time and propagation of the round-off error depended on the computer, we did the comparisons on three different processors: a 500 MHz SGI processor, a 700MHz IBM xSeries 350 processor and a 2.2 GHz Dell PE2650 processor. We found the differences were small and have reported the results for just the SGI processor. All comparisons were in double precision optimized Fortran.

The global error was estimated at one thousand evenly spaced points. The reference solution for the estimation was found by an integration in quadruple precision with a local error tolerance TOL of 10^{-18}. The global error in the reference solution was estimated using a solution calculated with $\mathrm{TOL} = 10^{-20}$. The least squares power law fit for the L_2 norm of the global error in the reference solution was $t^{1.99}$, $t^{1.95}$, $t^{2.00}$ and $t^{2.00}$ for the four problems, where t is the astronomical time. The end-point norm was less than 10^{-9} for each problem.

The integrations in quadruple precision took 50 to 200 times the CPU time of the integrations in double precision and this placed a limit on the length of integration for the test problems. It will be seen below that the length of integration is sufficiently long that much insight about the performance of the integrators is gained.

In §2, 3 and 4, we give a brief description of the eleven integrators, specify the four problems and present our comparisons. The specification of the problems is essentially that in [20] and has been included here for completeness. The comparisons for the first problem are discussed in detail; for the remaining problems we emphasize differences with preceding problems. We end in §5 with a discussion of our results and their possible implications.

2 Integrators

DIVA is a variable-order variable-stepsize (VOVS) Adams integrator for q-th and mixed order problems, $q > 0$, used extensively by NASA when planning their unmanned missions. It uses PECE mode on the first step, PEC mode for the rest of the start phase, and PECE mode thereafter. The predictor and corrector are orders k and $k+1$ respectively, $k \leq 19$, where k may differ from equation to equation. STEP is a VOVS Adams integrator for first order problems. It uses PECE mode with the predictor order k and the corrector order $k+1$, $k \leq 12$, where k is the same for all equations. STEP uses Møller's [15, 16] technique to reduce the systematic round-off error when the local error tolerance TOL is less than 100 times the unit round-off.

DXRK8 is a fixed-order, variable-step explicit Runge-Kutta (ERK) integrator that advances the solution using an order eight formula and controls the stepsize using embedded formulae of orders three and five. Twelve derivative evaluations are used on each step. RKSUITE is a fixed-order, variable-step ERK integrator with 2-3, 4-5 and 7-8 pairs. We used the 7-8 pair because it was more efficient than the other pairs for the stringent local error tolerances we employed. In addition, the order eight formula in the pair is the same as that in DXRK8, simplifying the comparison between the two integrators. The 7-8 pair uses 13 derivative evaluations per step.

ODEX is a VOVS extrapolation integrator for first order problems; ODEX2 is ODEX adapted to second-order differential equations of the form $\ddot{y} = f(x, y)$. RKNINT is a fixed-order, variable-stepsize ERKN integrator with 4-6 and 10-12 pairs. We used the 10-12 pair because it is considerably more efficient than the 4-6 pair at stringent tolerances. The pair uses 17 derivative evaluations per step.

CS4 and CS7 are fixed-stepsize integrators we wrote. They use the order four and seven symplectic ERKN methods of [4] and [5] respectively. The order four method uses five derivative evaluations on the first step, four on later steps; the order seven method uses 13 and 12 evaluations respectively. We chose these symplectic ERKN methods because the error coefficients are optimized and the number of evaluations per step is contrasting. Both integrators use Møller's technique to reduce the systematic round-off error.

S9 and S13 are fixed-stepsize integrators we wrote that use the order nine and 13 Störmer methods respectively. The numerical evidence in [8] suggests the order 13 methods is the most efficient among Störmer methods for N-body simulations of the Solar System; we included the order nine method to provide a contrast. The implementation is that of the N-body integrator NBI [21].

During an integration, DIVA and DXRK8 will recommend increasing TOL if the round-off error is deemed too large. The recommendation can be ignored but doing so runs counter to the design of the integrators and we always followed the recommendation. We used the default values for the optional inputs to all integrators whenever possible.

3 The problems

The initial conditions and parameter values are listed in the Appendix. The units of distance, time and mass are one astronomical unit, one day and one solar mass respectively.

3.1 Jovian problem

The Jovian problem has the Sun, Jupiter, Saturn, Uranus and Neptune interacting through Newtonian gravitational forces. Let $\mathbf{r}_i$ denote the position of the i-th body, where the bodies are ordered Sun to Neptune and the coordinate system is three-dimensional Cartesian with the origin at the barycentre of the bodies. The equations of motion for the Jovian problem are

$$\ddot{\mathbf{r}}_i(t) = \sum_{j=1, j\neq i}^{5} \frac{\mu_j(\mathbf{r}_j(t) - \mathbf{r}_i(t))}{||\mathbf{r}_j(t) - \mathbf{r}_i(t)||_2^3}, \quad i = 1, \ldots, 5, \tag{1}$$

where the dot operator denotes differentiation with respect to time t, $||\cdot||_2$ is the L_2 norm, and $\mu_j = Gm_j$, G being the gravitational constant and m_j the mass of the j-th body.

Except for the omission of Pluto and a change in the coordinate system, equations (1) are those for problem C5 in non-stiff `DETEST` [6]. C5 is an easy problem to solve. It becomes a demanding problem when the interval of integration is long. We used an interval of ten million years. This is equivalent to 843,418 orbital periods of Jupiter, 339,864 of Saturn, 113,963 of Uranus and 61,079 of Neptune.

The total energy $E(t)$ for the Jovian problem is

$$E(t) = \frac{1}{2}\left[\sum_{i=1}^{5}\left(m_i\dot{\mathbf{r}}_i \cdot \dot{\mathbf{r}}_i - \sum_{j=1, j\neq i}^{5} \frac{Gm_im_j}{||\mathbf{r}_j - \mathbf{r}_i||_2}\right)\right]. \tag{2}$$

$E(t)$ is conserved, permitting a second, weaker, assessment of the accuracy of the solution.

3.2 Nine Planets problem

This is the same as the Jovian problem except all nine planets are included. To make the problem consistent with the next two, we used the Earth-Moon system in place of the Earth.

The shortest orbital period for the Jovian and Nine Planets problems is 4331 days (Jupiter) and 88 days (Mercury) respectively. Hence, the average stepsize for the Nine Planets problem should be about 50 times smaller than that for the Jovian problem. This reduction together with the four-fold increase in the cost of evaluating the derivative meant we used an interval of integration of fifty thousand years.

3.3 Spin Axis problem

Quinn, Tremaine and Duncan [18] performed a three million year integration of the planets and the Earth's spin axis. The equations of motion included corrections for the dominant post-Newtonian terms and the quadpole moment of the Earth-Moon system.

Quinn, Tremaine and Duncan used three-dimensional Cartesian coordinates with the origin at the centre of the Sun. The planets were ordered Mercury, Venus, Earth-Moon, Mars, Jupiter, Saturn, Uranus, Neptune and Pluto. The equations of motion for the i-th planet are

$$\begin{aligned}\ddot{\mathbf{r}}_i \;=\; & -\frac{G(m_s+m_i)}{r_i^3}\mathbf{r}_i + \sum_{j=1,j\neq i} \frac{Gm_j}{||\mathbf{r}_j-\mathbf{r}_i||^3}(\mathbf{r}_j-\mathbf{r}_i) - \sum_{j=1,j\neq i}^{9} \frac{Gm_j\mathbf{r}_j}{r_j^3} + \\ & \frac{Gm_s}{r_i^3}\mathbf{r}_i\left[2(\beta+\gamma)\frac{Gm_s}{c^2 r_i} - \gamma\frac{||\dot{\mathbf{r}}_i||_2^2}{c^2}\right] + \\ & (2+2\gamma)\frac{Gm_s}{c^2 r_i^3}(\dot{\mathbf{r}}_i\cdot\mathbf{r}_i)\dot{\mathbf{r}}_i - \frac{3}{4}\delta_{i3}\frac{Gm_s}{r_i^3}\mathbf{r}_i\left(\frac{R}{r_i}\right)^2 f\frac{m_3 m_l}{(m_3+m_l)^2},\end{aligned} \tag{3}$$

where m_s, m_3 and m_l are the mass of the Sun, Earth and Moon respectively, m_j, $j = 1, 2, 4, \ldots, 9$ are the masses of the remaining planets, c is the speed of light, $r_k = ||\mathbf{r}_k||_2$, δ_{i3} is the Kronecker delta, and R and f are constants. The factor $m_3 m_l/(m_3+m_l)^2$ in (3) is evaluated as $(m_3/m_l)^2/(1+m_3/m_l)^2$ since m_3/m_l is known more accurately than m_3 and m_l separately.

The equations of motion for the Earth's spin axis $\widehat{\mathbf{a}}$ are

$$\dot{\widehat{\mathbf{a}}} = K(t)\left[\frac{2}{r_3^3}(\widehat{\mathbf{r}}_3\cdot\widehat{\mathbf{a}})(\widehat{\mathbf{r}}_3\times\widehat{\mathbf{a}}) - B_l(t)(\widehat{\mathbf{b}}\cdot\widehat{\mathbf{a}})(\widehat{\mathbf{b}}\times\widehat{\mathbf{a}})\right] + \frac{3Gm_s}{2r_3^3c^2}(\mathbf{r}_3\times\dot{\mathbf{r}}_3)\times\widehat{\mathbf{a}} - \mathbf{c}\frac{d\epsilon}{dt}, \tag{4}$$

where $K(t) = K_0\left[1 + d_1(t-t_0)\right]$, $B_l(t) = B_{l,0}\left[1 + d_2(t-t_0)\right]$, $\mathbf{c} = [\widehat{\mathbf{b}} - (\widehat{\mathbf{a}}\cdot\widehat{\mathbf{b}})\widehat{\mathbf{a}}](\sin\epsilon)^{-1}$ and $\widehat{\mathbf{a}}$, $\widehat{\mathbf{b}}$ and $\widehat{\mathbf{r}}_3$ are unit vectors. The subscript 3 in (4) refers to the Earth-Moon barycentre and $\widehat{\mathbf{b}} = \widehat{\mathbf{r}}_3\times\widehat{\dot{\mathbf{r}}}_3$.

The present problem differs from the previous two in two important ways: the equations are mixed order and the right side of the second-order equations has terms containing the first derivative.

We used an interval of integration of five thousand years.

3.4 DE102 problem

Newhall, Standish and Williams [17] used a detailed post-Newtonian model for the orbital motion of the Sun and planets as part of their model for the planetary ephemeris DE102. The equations of motion for the i-th planet, $i = 1, \ldots, 9$, are

$$\ddot{\mathbf{r}}_i = \left(\sum_{j=1,j\neq i}^{10} \rho_{ij}\frac{\mu_j(\mathbf{r}_j-\mathbf{r}_i)}{||\mathbf{r}_j-\mathbf{r}_i||_2^3}\right) + A_i, \quad i = 1, \ldots, 9, \tag{5}$$

where $j = 10$ refers to the Sun and the coordinate system is that of the Jovian problem.

The factor ρ_{ij} is

$$1 - \frac{2(\beta+\gamma)}{c^2} \sum_{k=1,k\neq i}^{10} \frac{\mu_k}{||\mathbf{r}_i - \mathbf{r}_k||_2} - \frac{2\beta - 1}{c^2} \sum_{k=1,k\neq j}^{10} \frac{\mu_k}{||\mathbf{r}_j - \mathbf{r}_k||_2} + \gamma \frac{||\dot{\mathbf{r}}_i||_2^2}{c^2} + (1+\gamma)\frac{||\dot{\mathbf{r}}_i||_2^2}{c^2} -$$

$$\frac{2(1+\gamma)}{c^2}\dot{\mathbf{r}}_i \cdot \dot{\mathbf{r}}_j - \frac{3}{2c^2}\left[\frac{(\mathbf{r}_i - \mathbf{r}_j)\cdot \dot{\mathbf{r}}_j}{||\mathbf{r}_i - \mathbf{r}_j||_2}\right]^2 + \frac{1}{2c^2}(\mathbf{r}_j - \mathbf{r}_i)\cdot \ddot{\mathbf{r}}_j,$$

where $\ddot{\mathbf{r}}_j$ is the Newtonian acceleration of the j-th body by the remaining nine bodies.

The term A_i is

$$\frac{1}{c^2}\sum_{j=1,j\neq i}^{10} \frac{\mu_j}{||\mathbf{r}_j - \mathbf{r}_i||_2^3} \left\{[\mathbf{r}_i - \mathbf{r}_j]\cdot[(2+2\gamma)\dot{\mathbf{r}}_i - (1+2\gamma)\dot{\mathbf{r}}_j]\right\}(\dot{\mathbf{r}}_i - \dot{\mathbf{r}}_j) + \frac{3+4\gamma}{2c^2}\sum_{j=1,j\neq i} \frac{\mu_j \ddot{\mathbf{r}}_j}{||\mathbf{r}_i - \mathbf{r}_j||_2}$$

The Sun's position and velocity are found using the definition of the barycentre for the ten bodies

$$\sum_{i=1}^{10} \mu_i^\star \mathbf{r}_i = 0, \quad \sum_{i=1}^{10} \mu_i^\star \dot{\mathbf{r}}_i = 0, \tag{6}$$

where

$$\mu_i^\star = \mu_i \left\{1 + \frac{1}{2c^2}||\dot{\mathbf{r}}_i||_2^2 - \frac{1}{2c^2}\sum_{j=1,j\neq i}^{10} \frac{\mu_j}{||\mathbf{r}_i - \mathbf{r}_j||_2}\right\}. \tag{7}$$

Since $\mu_j^\star$ depends on $\mathbf{r}_j$, the identities in (6) form a system of six nonlinear equations for the Sun's position and velocity. As in [17], we solved the system using fixed-point iteration, this must be done every derivative evaluation.

We used an interval of integration of two thousand years.

4 Results

The long intervals of integration necessitated small local error tolerances for the variable-step integrators. Most of our comparisons were with TOL as 10^{-13} or 10^{-14}; larger values did not realize the full accuracy of the integrators; smaller values often led to little gain in accuracy. On occasions we used larger values of TOL to permit a more direct comparison between integrators. With the fixed-stepsize integrators we chose the stepsizes to produce accurate solutions and to illustrate interesting behaviour of the global error. Throughout the summary below, $\mathcal{E}(t)$ denotes the norm of the global error over all components when expressed as a first order system; this is a fair measure of the global error because the bodies do not make close approaches and hence the velocities are not large.

4.1 Jovian problem

As part of a larger simulation, Grazier, Newman, Kaula and Hyman [9] solved the Jovian problem using the order 13 Störmer method with a stepsize of four days. They chose this stepsize as a compromise between having a small truncation error and performing the integration efficiently. We used this stepsize with `S9` and `S13` as well as a larger one of 20 days. We used stepsizes of 2.5, 10 and 100 days for `CS4` and 10, 20 and 100 days for `CS7`.

Figures 1 and 2 contain $\log_{10} - \log_{10}$ graphs of $\mathcal{E}(t)$ and the magnitude of the relative error $\mathcal{E}_{\mathrm{E}}(t)$ in the energy; Table 1 gives the cost of the integration and the power law fit for $\mathcal{E}(t)$. We have not distinguished between the evaluation of a second-order system and its first order form because there is little difference in CPU time. There is no information for `ODEX` with TOL $= 10^{-13}$ because `ODEX` stopped the integration at $t \approx 50,000$ years.

	TOL	N_f	N_s	$\bar{h}$	CPU	b	e_b	RMS	r
`DIVA`	10^{-13}	125,888,688	62,940,237	58.0	1186	2.094	0.0060	0.081	0.9960
`DIVA`	10^{-14}	131,723,154	65,855,590	55.5	1275	1.840	0.0026	0.036	0.9990
`STEP`	10^{-13}	235,292,227	117,171,519	31.2	1774	1.989	0.0015	0.020	0.9997
`STEP`	10^{-14}	303,568,678	151,350,267	24.1	2279	1.940	0.0034	0.046	0.9985
`DXRK8`	10^{-13}	873,329,016	72,777,417	50.2	2570	1.998	0.0011	0.014	0.9999
`DXRK8`	10^{-14}	1,063,794,551	88,649,544	41.2	3130	2.001	0.0011	0.014	0.9999
`RKSUITE`	10^{-13}	886,743,136	67,960,095	53.7	3883	2.003	0.0011	0.015	0.9998
`RKSUITE`	10^{-14}	1,219,896,644	91,726,459	39.8	5334	1.962	0.0019	0.025	0.9995
`ODEX`	10^{-14}	583,773,495	7,775,323	469.8	2003	2.007	0.0011	0.015	0.9998
`ODEX2`	10^{-13}	267,583,837	7,219,355	505.9	1126	1.994	0.0012	0.016	0.9998
`ODEX2`	10^{-14}	318,451,746	8,607,267	424.4	1338	2.014	0.0013	0.017	0.9998
`RKNINT`	10^{-13}	300,111,023	17,653,262	206.9	1074	1.944	0.0022	0.029	0.9994
`RKNINT`	10^{-14}	383,553,313	22,561,728	161.9	1373	1.869	0.0042	0.057	0.9975
`CS4`		146,100,001	36,525,000	100	475	0.635	0.0156	0.210	0.7904
`CS4`		1,461,000,001	365,250,000	10	4755	1.040	0.0050	0.068	0.9887
`CS4`		5,844,000,001	1,461,000,000	2.5	19023	1.129	0.0036	0.049	0.9950
`CS7`		438,300,001	36,525,000	100	1432	1.080	0.0100	0.135	0.9599
`CS7`		2,191,500,001	182,622,500	20	7158	2.025	0.0025	0.034	0.9992
`CS7`		4,383,000,001	365,250,000	10	14313	2.000	0.0013	0.018	0.9998
`S9`		182,625,154	182,625,000	20	2080	1.973	0.0013	0.017	0.9998
`S9`		913,125,154	913,125,000	4	10375	1.591	0.0052	0.070	0.9948
`S13`		182,625,210	182,625,000	20	2503	1.033	0.0013	0.017	0.9993
`S13`		913,125,210	913,125,000	4	12539	1.392	0.0055	0.075	0.9922

Table 1: Jovian problem. N_f and N_s are the number of derivative evaluations and accepted steps respectively, $\bar{h}$ is the average stepsize in days, CPU is the CPU time in seconds, b is the exponent in the power law fit for $\mathcal{E}(t)$, e_b is the standard error in b, RMS is the root-mean-square for the fit and r is the coefficient of correlation.

The graphs in Figure 1 permit a comparison of the integrators when the cost of the integration is ignored. The Adams and extrapolation integrators are of similar accuracy, the ERK integrators more accurate than these and `RKNINT` the most accurate of the variable-stepsize integrators. When TOL was decreased from 10^{-13} to 10^{-14}, $\mathcal{E}(t)$ for

STEP, ODEX2 and RKSUITE decreased by an order of magnitude. In contrast, $\mathcal{E}(t)$ decreased little for DIVA and DXRK8, probably because TOL was increased near the start of the two integrations. The decrease in $\mathcal{E}(t)$ for RKNINT was intermediate to that for the other integrators. We did integrations with RKNINT for TOL between 10^{-13} and 10^{-10} and found reasonable proportionality between $\mathcal{E}(t)$ and TOL, further evidence the round-off error for TOL $= 10^{-14}$ was significant.

Except for DIVA and RKNINT with TOL $= 10^{-14}$, the exponent b (see Table 1) in the power law fit for the variable-stepsize integrators is close to two as we would expect. When TOL $= 10^{-13}$, DIVA used fewer derivative evaluations and less CPU time than STEP. Since the solutions for DIVA and STEP are of similar accuracy, the number of evaluations and CPU time both imply DIVA was more efficient than STEP. The relative efficiency of DIVA and STEP for TOL $= 10^{-14}$ is not as easily decided because of the increase of TOL by DIVA. ODEX used approximately twice as many derivative evaluations and twice as much CPU time as ODEX2, a result we could expect from the underlying extrapolation formulae in the integrators. A comparison of the CPU time and $\mathcal{E}(t)$ shows that when CPU time is used to measure computational effort, RKNINT is more efficient than the other variable-stepsize integrators and the two Störmer integrators.

If the global error for a symplectic method is dominated by truncation error, $\mathcal{E}(t) \propto t$ except possibly at extremely large t. This appears the case for CS4 and CS7 with stepsizes of 10 and 100 days respectively. If the global error is dominated by systematic round-off error, $\mathcal{E}(t) \propto t^2$ (see Brouwer [3]). This appears so for CS7 with stepsizes of 10 and 20 days. $\mathcal{E}(t)$ for CS4 with a stepsize of 2.5 days grows faster than linearly suggesting round-off error is contributing significantly to $\mathcal{E}(t)$. When the stepsize is 100 days, $\mathcal{E}(t)$ for CS4 is large enough after one million years that the integration fails in the sense the phase angles are poorly known. $\mathcal{E}(t)$ for CS7 with a stepsize of 20 days grows markedly more slowly than t^2 for approximately the first 900,000 steps, possibly because Möller's technique is used in CS7 (and CS4).

The numerical evidence of Grazier [8] suggests that implementing Störmer methods as in NBI, S9 and S13 ensures systematic round-off error is eliminated or nearly so provided the truncation error is less than machine precision. This elimination leaves stochastic round-off error which should grow asymptotically as $t^{3/2}$ for $\mathcal{E}(t)$ ([3]). The size of the truncation error for the Jovian problem (see [8]) means we can expect quadratic growth for $\mathcal{E}(t)$ when using S9 and S13 with a stepsize of 20 days and $t^{3/2}$ growth when using a stepsize of four days. The values of b (Table 1) tend to confirm this, except for S13 with a stepsize of 20 days where b is close to one, an unexpectedly small value.

From Figure 1, we observe $\mathcal{E}(t)$ at $t = 10^7$ years is approximately 10^{-4} for RKNINT, CS4 with a stepsize of 2.5 days and CS7 with a stepsize of 10 days. Since CS4 and CS7 used more CPU time and, not unrelated, more derivative evaluations than RKNINT, we conclude RKNINT is more efficient than CS4 and CS7 on the Jovian problem out to $t = 10^7$ years. A similar conclusion for the CPU time holds when S13 with a stepsize of 20 days is compared with RKNINT.

Figure 2 contains graphs of the magnitude of the relative error $\mathcal{E}_{\mathrm{E}}(t)$ in the energy. The error was calculated in quadruple precision to reduce loss of significance. The relative efficiency of the variable-stepsize integrators is similar to that in Figure 1 and $\mathcal{E}_\mathrm{E}(t)$ grows steadily with t except for DIVA, STEP and RKNINT with TOL $= 10^{-14}$. $\mathcal{E}_\mathrm{E}(t)$ for CS4 with a stepsize of 100 days is small (less than 2×10^{-7}), confirmation $\mathcal{E}_\mathrm{E}(t)$ is a poor measure of the phase error. For the symplectic integrators with small stepsizes, $\mathcal{E}_\mathrm{E}(t)$ grows approximately as t, suggesting systematic round-off error is dominating $\mathcal{E}_\mathrm{E}(t)$. $\mathcal{E}_\mathrm{E}(t)$ for S9 with a stepsize of 10 days grows approximately linearly; for the remaining integrations with S9 and S13, $\mathcal{E}_\mathrm{E}(t)$ is $O(10^{-12})$ to $O(10^{-11})$.

4.2 Nine Planets problem

We used stepsizes of 1/16, 1/4 and 5/2 days with CS4, 1/4, 1/2 and 5/2 days with CS7, and 1/10 and 1/2 days with S9 and S13. Figures 3, 4 and Table 2 summarize the results.

	TOL	N_f	N_s	$\bar{h}$	CPU	b	e_b	RMS	r
DIVA	10^{-13}	43,878,803	21,938,428	0.832	890	1.783	0.0263	0.356	0.9064
DIVA	10^{-14}	44,262,725	22,130,337	0.825	897	2.501	0.0073	0.099	0.9958
STEP	10^{-13}	59,840,088	29,868,168	0.611	1111	1.860	0.0055	0.074	0.9957
STEP	10^{-14}	73,765,441	36,824,979	0.496	1375	1.728	0.0069	0.093	0.9922
DXRK8	10^{-13}	150,481,560	12,540,129	1.46	1470	2.004	0.0047	0.063	0.9973
DXRK8	10^{-14}	175,812,012	14,651,000	1.25	1717	2.003	0.0047	0.063	0.9973
RKSUITE	10^{-13}	254,109,126	19,453,565	0.939	3103	2.225	0.0135	0.183	0.9820
RKSUITE	10^{-14}	348,231,327	26,386,430	0.692	4251	2.022	0.0047	0.063	0.9973
ODEX	10^{-13}	116,870,744	1,454,110	12.6	1251	1.981	0.0047	0.063	0.9972
ODEX	10^{-14}	131,986,413	1,721,043	10.6	1412	2.002	0.0047	0.064	0.9973
ODEX2	10^{-13}	59,585,589	1,447,363	12.6	695	2.033	0.0047	0.064	0.9973
ODEX2	10^{-14}	66,806,460	1,696,001	10.8	779	1.990	0.0047	0.064	0.9972
RKNINT	10^{-13}	95,659,885	5,627,052	3.25	1032	1.764	0.0091	0.123	0.9870
RKNINT	10^{-14}	121,917,318	7,171,605	2.55	1316	1.764	0.0055	0.074	0.9952
CS4		29,220,001	7,305,000	2.5	280	0.102	0.0212	0.287	0.1507
CS4		292,200,001	73,050,000	0.25	2800	1.000	0.0077	0.105	0.9714
CS4		1,168,800,001	292,200,000	0.0625	11197	1.017	0.0073	0.099	0.9753
CS7		87,660,001	7,305,000	2.5	844	0.987	0.0185	0.250	0.8603
CS7		438,300,001	36,525,000	0.50	4217	0.989	0.0164	0.222	0.8856
CS7		898,515,001	73,050,000	0.25	8438	2.017	0.0073	0.099	0.9935
S9		36,525,154	36,525,000	0.50	951	1.751	0.0125	0.169	0.9755
S9		182,625,154	182,625,000	0.10	4747	0.947	0.0045	0.061	0.9890
S13		36,525,210	36,525,000	0.50	1130	1.8618	0.0028	0.038	0.9966
S13		182,625,210	182,625,210	0.10	5630	1.077	0.0028	0.038	0.9669

Table 2: Nine Planets problem. The columns are as in Table 1.

The results for the Nine Planets problem differ from those for the Jovian problem in several ways. The solutions for DIVA are more accurate relative to those for STEP but at the cost of more CPU time (relative to STEP). The same occurred for RKSUITE and DXRK8. The Adams integrators used less CPU time relative to RKNINT, an expected result since the derivative for the Nine Planets problem is more expensive than that for the Jovian problem. $\mathcal{E}(t)$ for DIVA with TOL $= 10^{-13}$ has a troubling dip of over two orders of magnitude at $t \approx 2.2 \times 10^4$ years. The relative error $\mathcal{E}_{\mathrm{E}}(t)$ in the energy has a similar dip but at an earlier time; $\mathcal{E}_{\mathrm{E}}(t)$ appears to be entering another dip just beyond $t = 50,000$ years.

We did integrations with RKNINT for TOL between 10^{-13} and 10^{-10} and found RKNINT used about one third less CPU time than DIVA to produced a solution of similar accuracy to that for DIVA.

Except for the larger oscillations, something possibly caused by the wider range in orbital periods, $\mathcal{E}(t)$ and $\mathcal{E}_{\mathrm{E}}(t)$ for CS4 and CS7 show the same characteristic behaviour as for the Jovian problem. The power law fits to $\mathcal{E}(t)$ for S9 and S13 suggest $\mathcal{E}(t)$ is growing quadratically for a stepsize of 1/2 day and linearly for a stepsize of 1/10 day.

$\mathcal{E}(t)$ for RKNINT with TOL $= 10^{-14}$ and CS7 with a stepsize of 1/2 day is approximately 10^{-6} at $t = 50,000$ years. Since RKNINT uses less than one third the CPU time of CS7 (and fewer than one third the number of derivative evaluations) we conclude RKNINT is more efficient than CS7 on this problem out to $t = 50,000$ years. RKNINT however requires more CPU than S13 with a stepsize of 1/2 day, a result that contrasts with that for the Jovian problem.

4.3 Spin Axis problem

Figure 5 and Table 3 summarize the results.

	TOL	N_f	N_s	$\bar{h}$	CPU	b	e_b	RMS	r
DIVA	10^{-13}	4,232,996	2,116,383	0.863	162	2.081	0.0050	0.068	0.9971
DIVA	10^{-14}	4,464,917	2,232,434	0.818	171	1.712	0.0065	0.087	0.9930
STEP	10^{-13}	6,009,117	2,998,436	0.609	206	2.080	0.0067	0.091	0.9949
STEP	10^{-14}	7,392,881	3,690,680	0.494	254	1.961	0.0059	0.080	0.9955
DXRK8	10^{-13}	15,103,524	1,258,626	1.45	395	1.999	0.0047	0.063	0.9973
DXRK8	10^{-14}	17,643,924	1,470,326	1.24	462	2.002	0.0047	0.063	0.9973
RKSUITE	10^{-13}	25,719.748	1,955,643	0.934	730	1.652	0.0073	0.099	0.9903
RKSUITE	10^{-14}	35,347,316	2,657,191	0.687	1002	2.032	0.0053	0.072	0.9966
ODEX	10^{-13}	11,640,844	146,174	12.5	314	1.998	0.0047	0.064	0.9972
ODEX	10^{-14}	13,265,483	173,774	10.5	357	1.998	0.0047	0.064	0.9972

Table 3: Spin Axis problem. The columns are as in Table 1.

The number of derivative evaluations for DIVA, STEP, DXRK8, RKSUITE and ODEX per unit of astronomical time is very close to that for the Nine Planets problem, evidence the terms for the post-Newtonian interactions, the quadpole of the Earth-Moon system and the Earth's spin axis do not drive the stepsize selection. The relative accuracy of the five integrators is similar to that for the Nine Planets problem except for TOL $= 10^{-14}$ where DIVA is more accurate than STEP. Decreasing TOL from 10^{-13} to 10^{-14} produced an order of magnitude decrease in $\mathcal{E}(t)$ for DIVA, STEP, DXRK8 and ODEX and little decrease for RKSUITE, suggesting round-off error dominates the global error for RKSUITE but not for the other integrators when TOL $= 10^{-14}$.

For TOL = 10^{-13}, DIVA produced a more accurate solution, used less CPU time and performed fewer derivative evaluations than STEP. For TOL = 10^{-14}, DIVA used approximately one third the CPU time of DXRK8 and produced a slightly more accurate solution. We re-solved the Spin Axis problem with RKSUITE using values of TOL between 10^{-13} and 10^{-10} and compared RKSUITE with the remaining integrators using like values of $\mathcal{E}(t)$. RKSUITE used about the same CPU time as DXRK8 and ODEX, one and one half times that of STEP and three times that of DIVA.

4.4 DE102 problem

Figure 5 and Table 2 summarize the results. There are two interesting differences between the results for this problem and the Nine Planets problem. RKSUITE is marginally more accurate than DXRK8 at the cost of nearly twice the CPU time, and decreasing TOL from 10^{-13} to 10^{-14} led to a two orders of magnitude decrease in $\mathcal{E}(t)$ for ODEX.

	TOL	N_f	N_s	$\bar{h}$	CPU	b	e_b	RMS	r
DIVA	10^{-13}	1,748,596	874,252	0.836	288	1.903	0.0052	0.070	0.9963
DIVA	10^{-14}	1,839,853	919,874	0.794	303	1.917	0.0047	0.063	0.9970
STEP	10^{-13}	2,400,555	1,197,821	0.610	395	2.132	0.0054	0.073	0.9968
STEP	10^{-14}	2,936,319	1,466,016	0.498	484	1.977	0.0046	0.062	0.9973
DXRK8	10^{-13}	6,066,888	505,573	1.44	943	2.000	0.0044	0.060	0.9976
DXRK8	10^{-14}	7,086,288	590,523	1.24	1102	1.992	0.0038	0.051	0.9982
RKSUITE	10^{-13}	10,313,055	783,590	0.932	1618	1.984	0.0030	0.041	0.9989
RKSUITE	10^{-14}	14,112,364	1,061,691	0.688	2214	1.986	0.0029	0.040	0.9989
ODEX	10^{-13}	4,650,791	59,015	12.4	729	1.994	0.0047	0.064	0.9972
ODEX	10^{-14}	5,336,263	70,389	10.4	836	1.990	0.0046	0.063	0.9973

Table 4: DE102 problem. The columns are as in Table 1.

5 Discussion

We presented four N-body problems, all with long intervals of integration, two with detailed interactions between the bodies, and investigated how effective the Störmer methods of orders nine and thirteen, two optimized symplectic explicit Runge-Kutta-Nyström methods of orders four and seven, and the variable-stepsize integrators DIVA, STEP, DXRK8, RKSUITE, ODEX, ODEX2, RKNINT were at finding accurate solutions to the problems. The Störmer and symplectic explicit Runge-Kutta-Nyström methods were implemented as fixed-stepsize integrators and incorporated techniques to reduce the systematic round-off error. All comparisons were done in double precision. The reference solutions for the global error estimation were found using accurate integrations in quadruple precision.

Two problems, one with five bodies the other with ten, had the bodies interacting through Newtonian gravitational forces. When computational effort is measured by CPU time, RKNINT is the most efficient of the variable-stepsize integrators on the two problems.

RKNINT is more efficient than the order 13 Störmer on the problem with five bodies but less efficient on the problem with ten bodies. As could be expected, going from five to ten bodies reduces the efficiency of RKNINT relative to the Adams integrators DIVA and STEP.

The remaining two problems had post-Newtonian interactions between the bodies and we compared DIVA, STEP, DXRK8, RKSUITE and ODEX on them. We found DIVA used the least amount of CPU time to achieve the same end-point global error.

Calvo and Sanz Serna [5] compared one non-symplectic and three symplectic explicit Runge-Kutta Nyström methods on the two-body problem with an interval of integration of 21,870 periods. They found the non-symplectic method required fewer evaluations than the symplectic methods to achieve the same end-point global error. Calvo and Sanz Serna concluded non-symplectic methods could easily be more efficient when an accurate solution is required, even when the interval of integration is long.

The smallest period in our problem with the Sun and Jovian planets interacting through Newtonian gravitational forces is that of Jupiter. If we take this as the period in the two-body problem, the interval of integration for our problem is 40 times that in [5]. We found the non-symplectic integrator RKNINT used less CPU time and fewer derivative evaluations than the two symplectic integrators CS4 and CS7 to achieve the same end-point global error.

Our testing showed that even with techniques to reduce round-off error employed, the accumulated round-off error severely limits the accuracy achievable on long simulations. Fast higher precision arithmetic could clearly be used to advantage on such simulations.

The implementation of the Störmer methods meant the norm of the global error, when the truncation error was less than machine precision, should have grown approximately as $t^{3/2}$. We found the norm grew linearly or nearly so for some integrations. This contradiction could possibly be explained by the stochastic nature of the error growth. The fact the actual growth can differ from the expected growth by so much could affect the choice of integrator.

For each integration, we used least squares to fit the power law at^b to the norm of the global error. On many of the integrations the standard error in b was small and the coefficient of correlation r was close to one. In these cases it is tempting to extend our conclusions to larger intervals of integration. For example, on the problem with the Sun and Jovian planets the end-point global error for RKNINT with TOL $= 10^{-14}$ was approximately that for CS7 with a stepsize of 20 days. Since b for RKNINT was smaller than b for CS7 and RKNINT used less CPU time than CS7 we might conclude RKNINT will use less CPU time for $t > 10^7$ years. Unfortunately, the assumption $\mathcal{E}(t)$ will continue to grow in the same way for larger t is fraught with uncertainty.

Appendix

Tables 5 to 10 list the initial conditions and parameter values for the four test problems. We used $\mu_s = (0.01720209895)^2$.

	x	y	z
Sun	0.9209498686328694E-03	0.2304166030756204E-02	0.9127217048523883E-03
Jupiter	0.3350285173564643E+01	-0.3471457282981824E+01	-0.1571236964688948E+01
Saturn	-0.8971584118413477E+01	0.2281986174163616E+01	0.1331251331416312E+01
Uranus	-0.1002083045458687E+01	0.1732581263930256E+02	0.7605737768120762E+01
Neptune	-0.2919365978874257E+02	-0.7716981025967714E+01	-0.2426332656583918E+01
Sun	-0.4665246664531984E-05	-0.3149154564335707E-05	-0.1269852543206254E-05
Jupiter	0.5580977902917778E-02	0.4959111982658174E-02	0.1991007074196164E-02
Saturn	-0.1862917203661242E-02	-0.4987007735981776E-02	-0.1981527265350456E-02
Uranus	-0.3959919409682252E-02	-0.3790629396772767E-03	-0.1101198438310003E-03
Neptune	0.8160828111535530E-03	-0.2775247414144566E-02	-0.1157385882979126E-02

Table 5: Jovian problem: rows 1 to 5 - initial position, rows 6 to 10 - initial velocity.

References

[1] R.W. Brankin, J.R. Dormand, I. Gladwell, P. Prince and W.L. Seward, *ALGORITHM 670: A Runge-Kutta-Nystrom Code, ACM Trans. Math. Soft.*, **15** (1989), 31-40.

[2] R.W. Brankin, I. Gladwell, *ALGORITHM 771: rksuite_90: software for ordinary differential equation initial value problems, ACM Trans. Math. Soft.*, **23** (1997), 402-415.

[3] D. Brouwer, *On the accumulation of errors in numerical integration*, Astron. J., **30** (1937), 149-153.

[4] M. P. Calvo, J. M. Sanz-Serna, *The development of variable-step symplectic integrators with application to the two-body problem*, SIAM J. Sci. Comput., **14**, 4 (1993), 936-952.

[5] M. P. Calvo, J. M. Sanz-Serna, *High-order symplectic Runge-Kutta-Nyström methods*, SIAM J. Sci. Comput., **14**, 5 (1993), 1237-1252.

[6] W.H. Enright, J.D. Pryce, *Two FORTRAN packages for assessing initial value methods*, ACM Trans. Math. Soft., **13** 1 (1987), 1-27.

[7] J.M. Fine, *On the efficiency of Nystrom methods*, Report EW-27-86, Division of Engineering and Weapons, US Naval Academy, Annapolis, Maryland, 1986.

[8] K.R. Grazier, *The stability of planetesimal niches in the outer solar system: a numerical study*, Ph.D. thesis, University of California, Los Angeles, 1997.

[9] K. Grazier, W.I. Newman, W.M. Kaula, J.M. Hyman, *Dynamical evolution of planetesimals in the outer solar system: I. the Jupiter/Saturn zone*, Icarus **140** (1999), 341-352.

[10] E. Hairier, S.P. Nørsett, G. Wanner, *Solving ordinary differential equations I: nonstiff problems*, Springer-Verlag, 1ed, 1987.

	x	y	z
Sun	0.9301259103994515e-03	0.2292733100662641e-02	0.9059057664779422e-03
Mercury	0.3448565760800415e+00	0.4790821305397614e-01	-0.1001813144545456e-01
Venus	0.1438953102536455e+00	0.6492977991345496e+00	0.2833883064268579e+00
Earth-Moon	-0.1354345700443955e+00	0.8956906559576626e+00	0.3883642504058149e+00
Mars	-0.1368903850273021e+01	0.8454279811185666e+00	0.4247388123779079e+00
Jupiter	0.3350294349606409e+01	-0.3471468715911917e+01	-0.1571243780627322e+01
Saturn	-0.8971574942371711e+01	0.2281974741233523e+01	0.1331244515477938e+01
Uranus	-0.1002073869416921e+01	0.1732580120637246e+02	0.7605730952182388e+01
Neptune	-0.2919365061270080e+02	-0.7716992458897807e+01	-0.2426339472522292e+01
Pluto	-0.2623272065610510e+02	0.2056426815315656e+02	0.1444546303354718e+02
Sun	-0.4559774360194479e-05	-0.3150250493626429e-05	-0.1274328432609927e-05
Mercury	-0.8471091819370054e-02	0.2561145505678817e-01	0.1458557100780699e-01
Venus	-0.1989837205370269e-01	0.3109969215624964e-02	0.2658171477313190e-02
Earth-Moon	-0.1732455862288979e-01	-0.2247454982261186e-02	-0.9746354441906539e-03
Mars	-0.7389123605631364e-02	-0.9480508889767826e-02	-0.4152929465094740e-02
Jupiter	0.5581083375222116e-02	0.4959110886728884e-02	0.1991002598306760e-02
Saturn	-0.1862811731356904e-02	-0.4987008831911066e-02	-0.1981531741239860e-02
Uranus	-0.3959813937377914e-02	-0.3790640356065674e-03	-0.1101243197204039e-03
Neptune	0.8161882834578905e-03	-0.2775248510073856e-02	-0.1157390358868530e-02
Pluto	-0.1320448472641354e-02	-0.2623278455987146e-02	-0.4283576834589079e-03

Table 6: Nine-Planets problem: rows 1 to 10 - initial position, rows 11 to 20 - initial velocity.

[11] F. Krogh, *diva*, `http://www.mathalacarte.com`

[12] F. Krogh, *Changing stepsize in the integration of differential equations using modified divided differences*, in Dale G. Bettis, editor, Proceedings of the conference on the numerical solution of ordinary differential equations, Lecture notes in Mathematics 362, 22-71, Springer-Verlag, 1974.

[13] F. Krogh, *dxrk8*, `http://mathalacarte.com/fkrogh`

[14] F. Krogh, *An adams guy does the Runge-Kutta*, Internal Computing Memo. 554, Jet Propulsion Laboratory, March 1997, available from `http://mathalacarte.com`

[15] O. Møller, *Quasi double-precision in floating point addition*, BIT, **5** (1965), 37-50.

[16] O. Møller, *Note on quasi double-precision*, BIT, **5** (1965), 251-255.

[17] X X Newhall, E.M. Standish, J.G. Williams, *DE 102: a numerically integrated ephemeris of the Moon and planets spanning forty-four centuries*, Astron. Astrophys. **125** (1983), 150-167.

[18] T.R. Quinn, S. Tremaine, M. Duncan, *A three million year integration of the Earth's orbit*, Astron. J., **101**, 6 (1991), 2287-2305.

[19] L.F. Shampine, M.K. Gordon, *Computer solution of ordinary differential equations; the initial value problem*, W.H. Freeman & Co., San Francisco, 1975.

	x	y	z
Mercury	0.3439264501696420E+00	0.4561547995331350E-01	-0.1092403721193250E-01
Venus	0.1429651843432460E+00	0.6470050660338870E+00	0.2824824006603800E+00
Earth-Moon	-0.1363646959547950E+00	0.8933979228570000E+00	0.3874583446393370E+00
Mars	-0.1369833976183420E+01	0.8431352480179040E+00	0.4238329066114300E+00
Jupiter	0.3349364223696010E+01	-0.3473761449012580E+01	-0.1572149686393800E+01
Saturn	-0.8972505068282110E+01	0.2279682008132860E+01	0.1330338609711460E+01
Uranus	-0.1003003995327320E+01	0.1732350847327180E+02	0.7604825046415910E+01
Neptune	-0.2919458073861120E+02	-0.7719285191998470E+01	-0.2427245378288770E+01
Pluto	-0.2623365078201550E+02	0.2056197542005590E+02	0.1444455712778070E+02
Axis	0.1519208290000000E-06	-0.2570604820000000E-05	0.9999999999966840E+00
Mercury	-0.8466532045009860E-02	0.2561460530728180E-01	0.1458684533623960E-01
Venus	-0.1989381227934250E-01	0.3113119466118590E-02	0.2659445805745800E-02
Earth-Moon	-0.1731999884852960E-01	-0.2244304731767560E-02	-0.9733611157580440E-03
Mars	-0.7384563831271170E-02	-0.9477358639274200E-02	-0.4151655136662130E-02
Jupier	0.5585643149582310E-02	0.4962261137222510E-02	0.1992276926739370E-02
Saturn	-0.1858251956996710E-02	-0.4983858581417440E-02	-0.1980257412807250E-02
Uranus	-0.3955254163017720E-02	-0.3759137851129410E-03	-0.1088499912877940E-03
Neptune	0.8207480578180850E-03	-0.2772098259580230E-02	-0.1156116030435920E-02
Pluto	-0.1315888698281160E-02	-0.2620128205493520E-02	-0.4270833550262980E-03

Table 7: Spin Axis problem: rows 1 to 10 - initial position, rows 11 to 19 - initial velocity.

[20] P.W. Sharp, *N-body simulations: the performance of eleven integtrators*, Report Series 468, Department of Mathematics, University of Auckland, July 2001.

[21] F. Varadi, `http::/astrobiology.ucla.edu/ varadi/NBI/NBI.html`.

	x	y	z
Mercury	0.3448565760799925e+00	0.4790821305404569e-01	-0.1001813144542399e-01
Venus	0.1438953102535965e+00	0.6492977991346192e+00	0.2833883064268885e+00
Earth-Moon	-0.1354345700444445e+00	0.8956906559577322e+00	0.3883642504058455e+00
Mars	-0.1368903850273070e+01	0.8454279811186362e+00	0.4247388123779385e+00
Jupiter	0.3350294349606360e+01	-0.3471468715911848e+01	-0.1571243780627291e+01
Saturn	-0.8971574942371760e+01	0.2281974741233592e+01	0.1331244515477969e+01
Uranus	-0.1002073869416970e+01	0.1732580120637253e+02	0.7605730952182419e+01
Neptune	-0.2919365061270085e+02	-0.7716992458897738e+01	-0.2426339472522261e+01
Pluto	-0.2623272065610515e+02	0.2056426815315663e+02	0.1444546303354721e+02
Mercury	-0.8471091819370170e-02	0.2561145505678807e-01	0.1458557100780695e-01
Venus	-0.1989837205370281e-01	0.3109969215624858e-02	0.2658171477313146e-02
Earth-Moon	-0.1732455862288991e-01	-0.2247454982261292e-02	-0.9746354441906976e-03
Mars	-0.7389123605631480e-02	-0.9480508889767932e-02	-0.4152929465094784e-02
Jupiter	0.5581083375222000e-02	0.4959110886728778e-02	0.1991002598306716e-02
Saturn	-0.1862811731357020e-02	-0.4987008831911172e-02	-0.1981531741239904e-02
Uranus	-0.3959813937378030e-02	-0.3790640356066732e-03	-0.1101243197204476e-03
Neptune	0.8161882834577752e-03	-0.2775248510073962e-02	-0.1157390358868574e-02
Pluto	-0.1320448472641470e-02	-0.2623278455987252e-02	-0.4283576834589516e-03

Table 8: DE102 problem: rows 1 to 9 - initial position, rows 10 to 18 - initial velocity.

Mercury	Venus	Earth-Moon	Mars	Jupiter	Saturn	Uranus	Neptune	Pluto
$\frac{\mu_s}{6023600}$	$\frac{\mu_s}{408523.5}$	$\frac{\mu_s}{328900.53}$	$\frac{\mu_s}{3098710}$	$\frac{\mu_s}{1047.355}$	$\frac{\mu_s}{3498.5}$	$\frac{\mu_s}{22869}$	$\frac{\mu_s}{19314}$	$\frac{\mu_s}{3000000}$

Table 9: The μ for the planets.

$c = 173.144633484$ AU d^{-1}	$f = 0.9473$	$K_0 = 2.306559456 \times 10^{-7}$ d^{-1}
$R = 0.0025696$ AU	$\beta = 1$	$d_1 = -6.39235695008 \times 10^{-8}$ d^{-1}
$d_2 = -7.9488 \times 10^{-13}$ d^{-1}	$\gamma = 1$	$\epsilon = 23.43929 - 3.60288 \times 10^{-14}(t - t_0)$
$m_3/m_l = 81.3007$	$B_{l,0} = 2.161225$	

Table 10: The parameters for the Spin Axis problem.

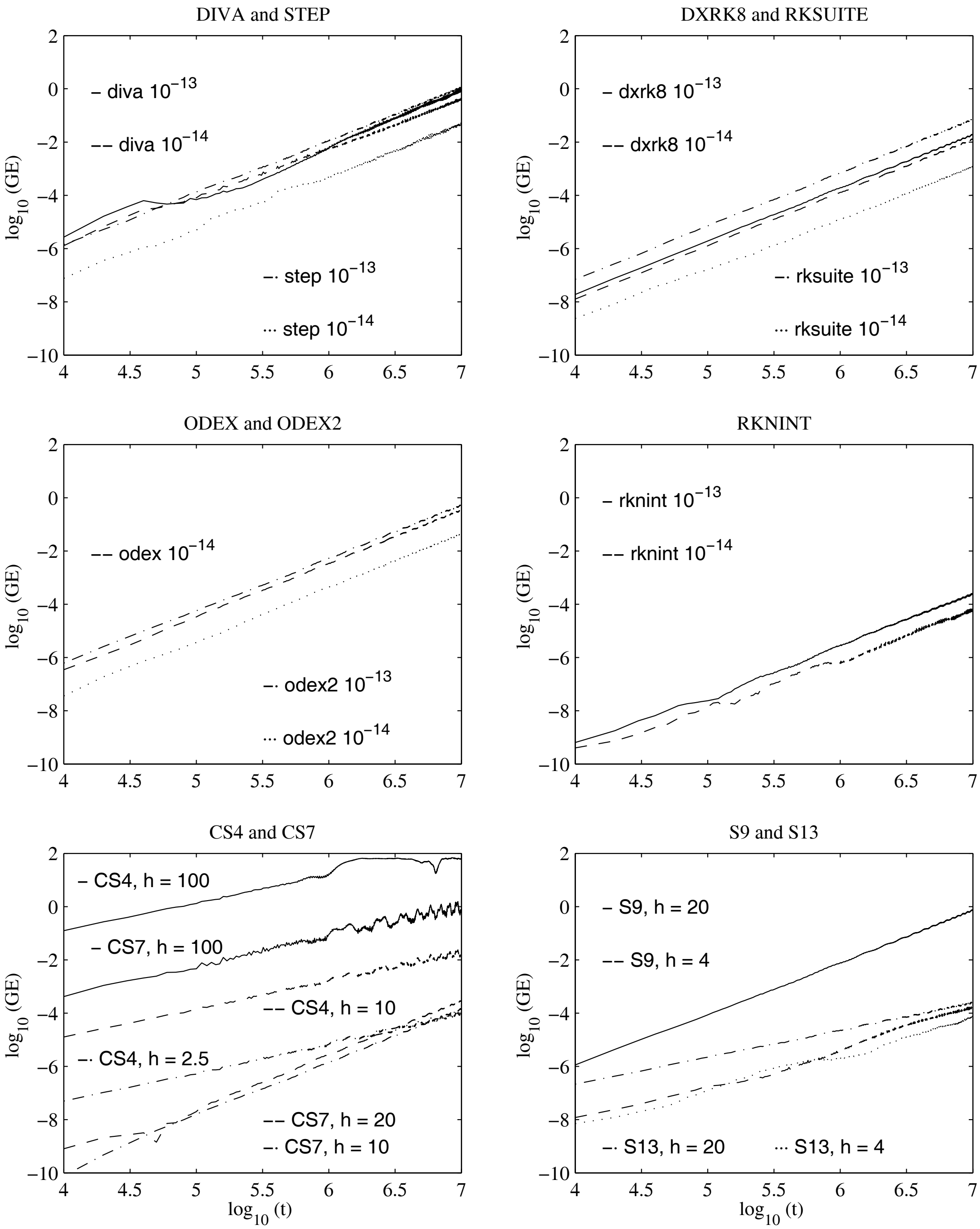

Figure 1: Jovian problem. $\log_{10} - \log_{10}$ graphs of $\mathcal{E}(t)$.

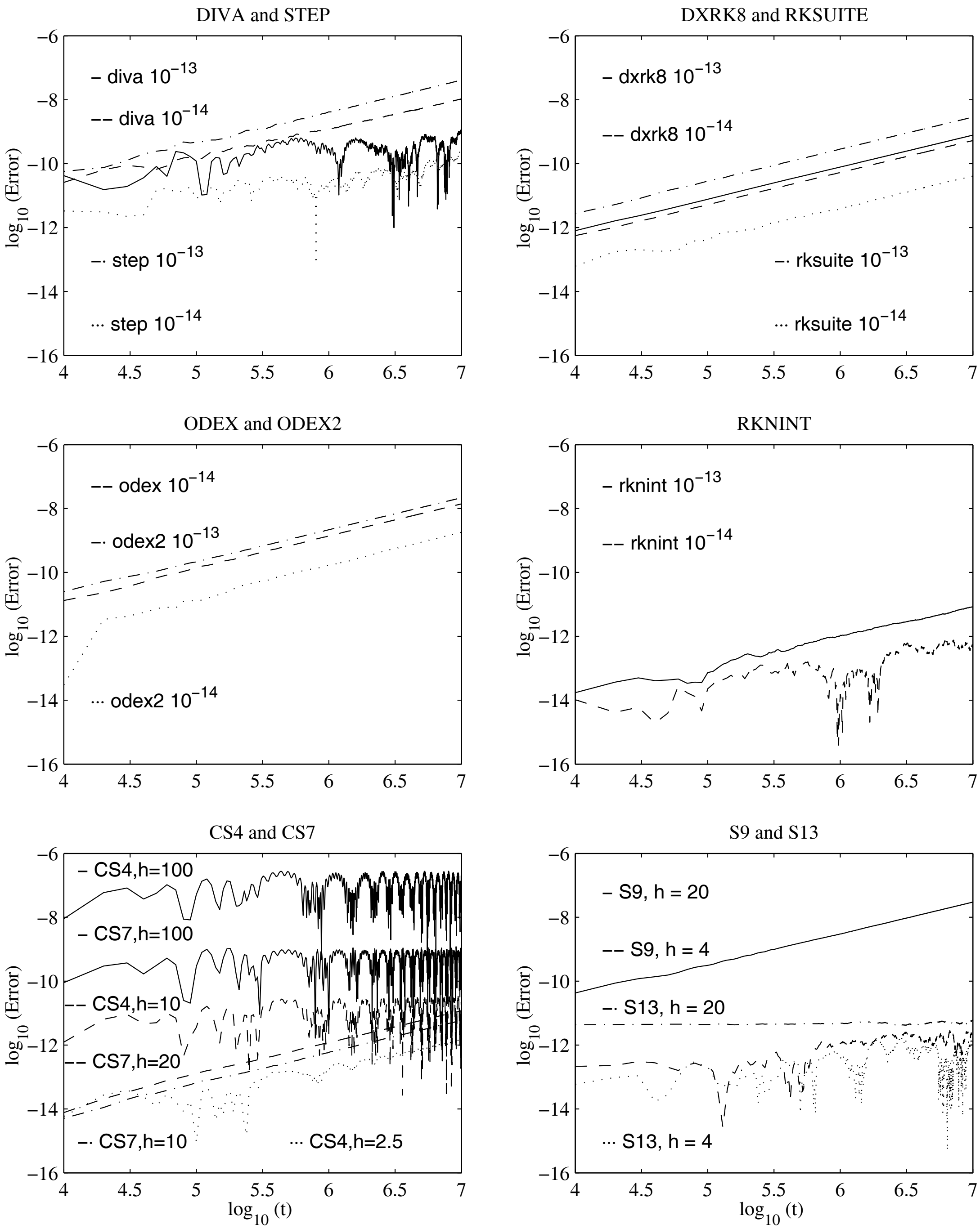

Figure 2: Jovian problem. $\log_{10} - \log_{10}$ graphs of the magnitude of the relative error in the energy.

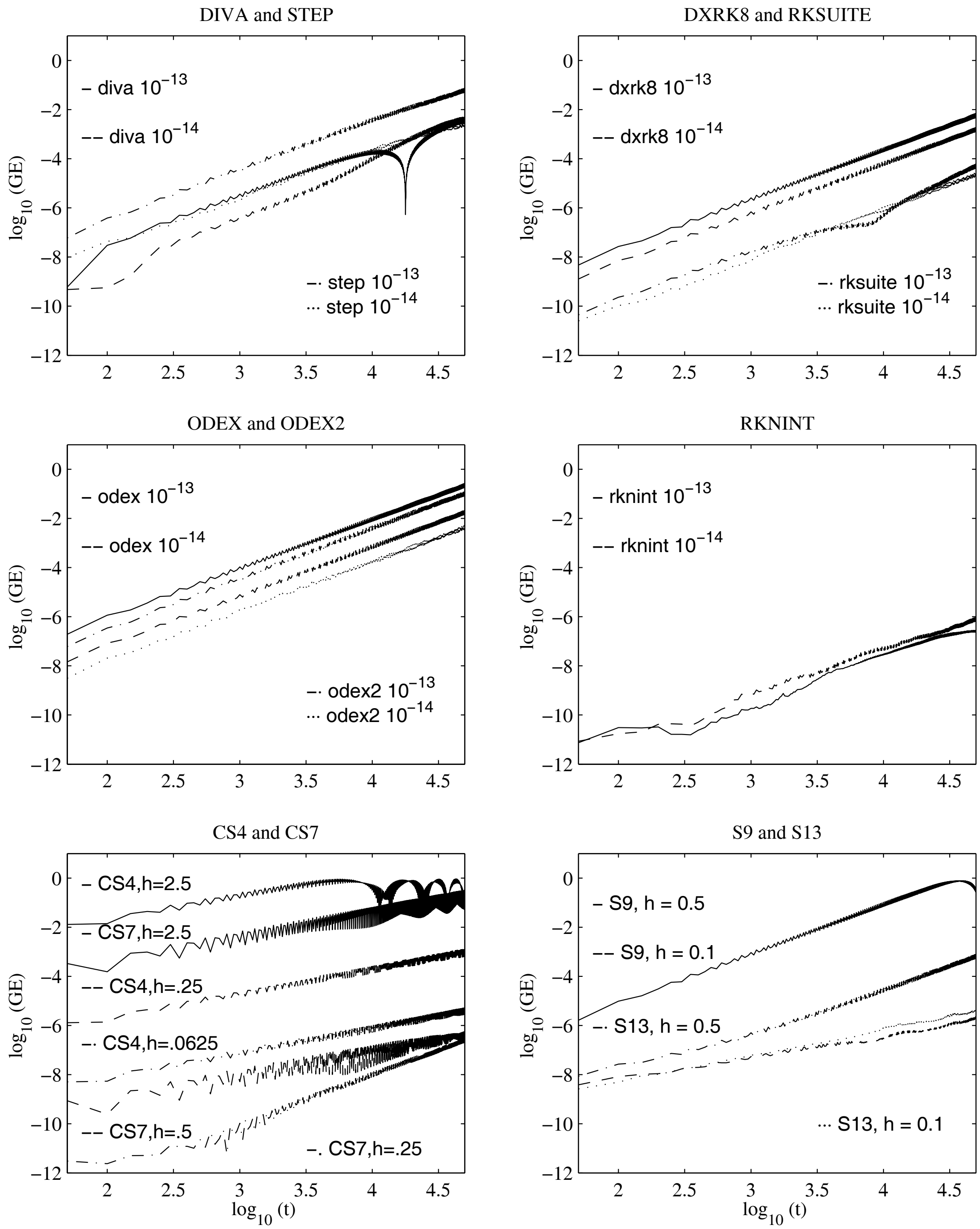

Figure 3: Nine Planets problem. $\log_{10} - \log_{10}$ graphs of $\mathcal{E}(t)$.

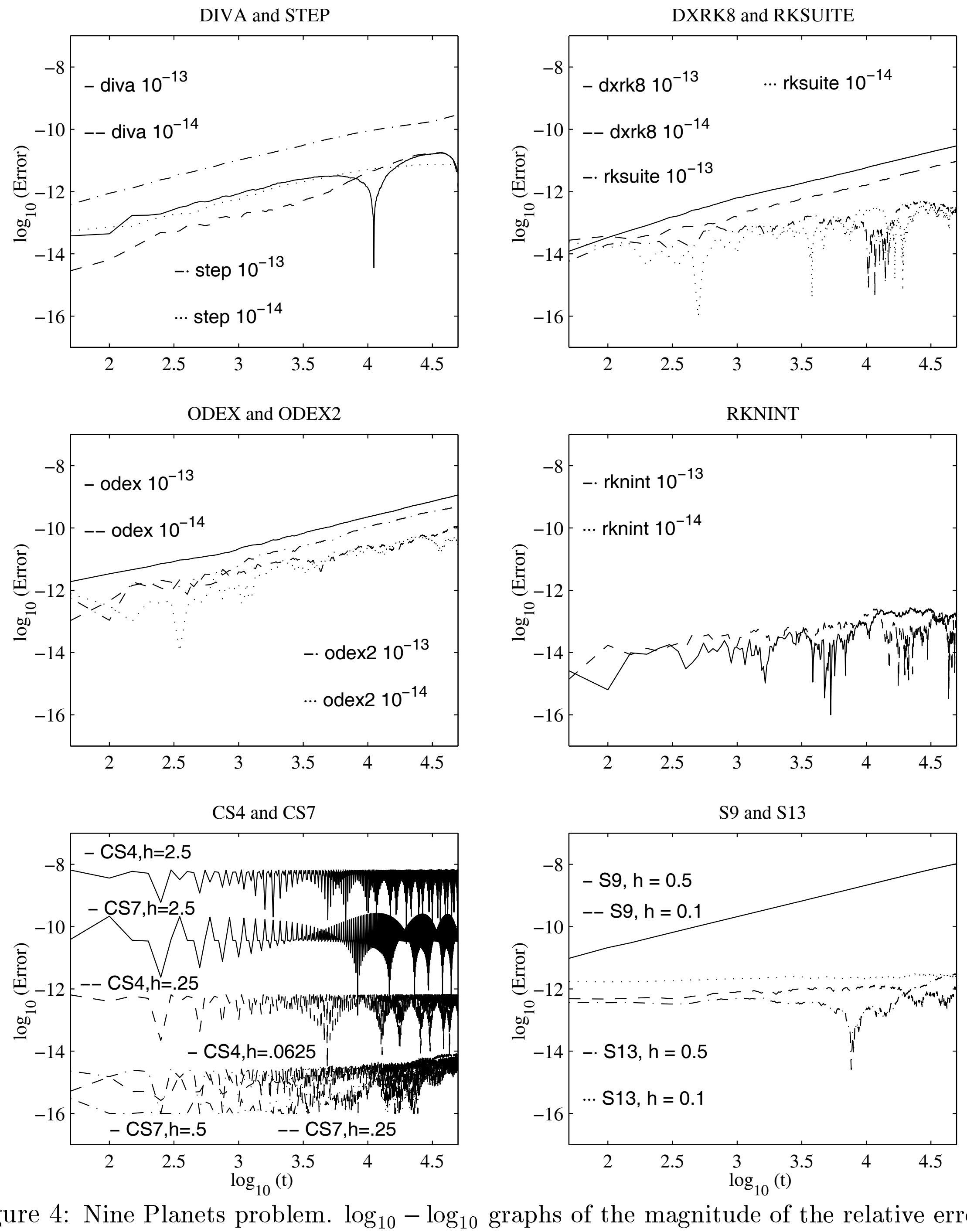

Figure 4: Nine Planets problem. $\log_{10} - \log_{10}$ graphs of the magnitude of the relative error in the energy.

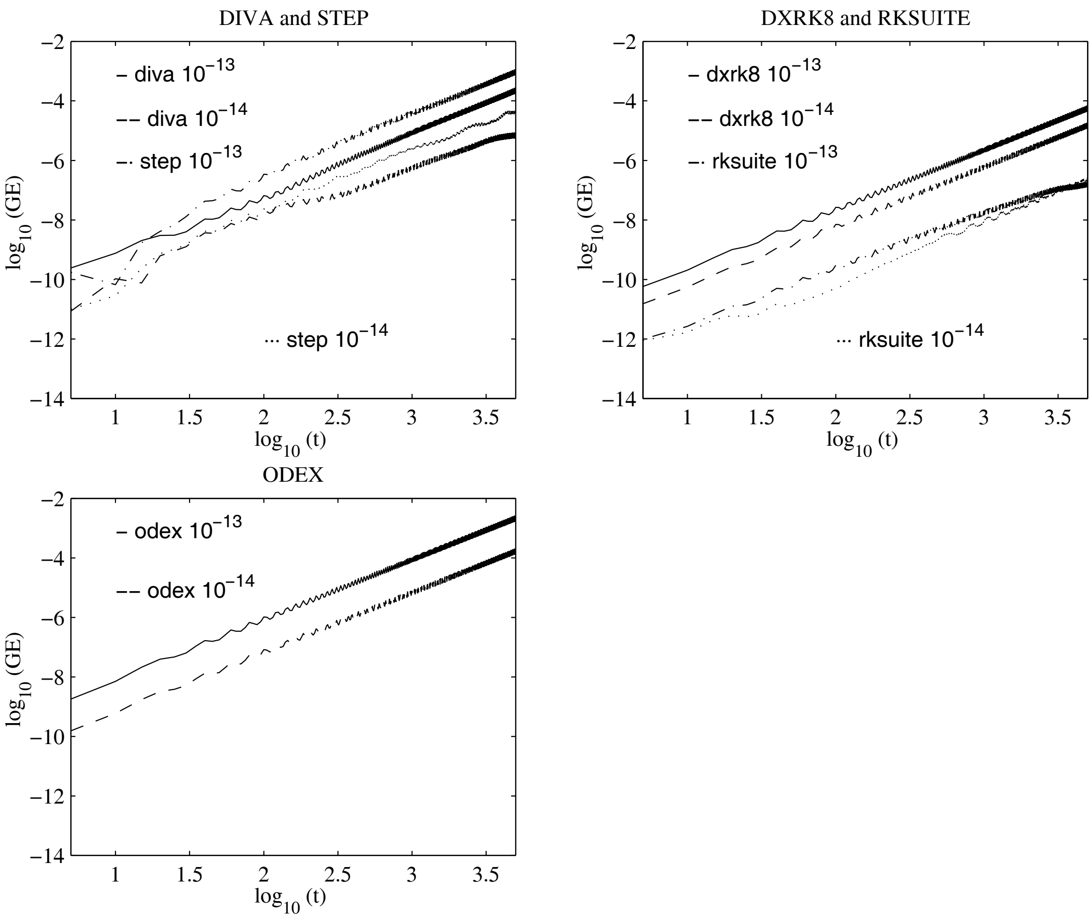

Figure 5: Spin axis problem. $\log_{10} - \log_{10}$ graphs of $\mathcal{E}(t)$.

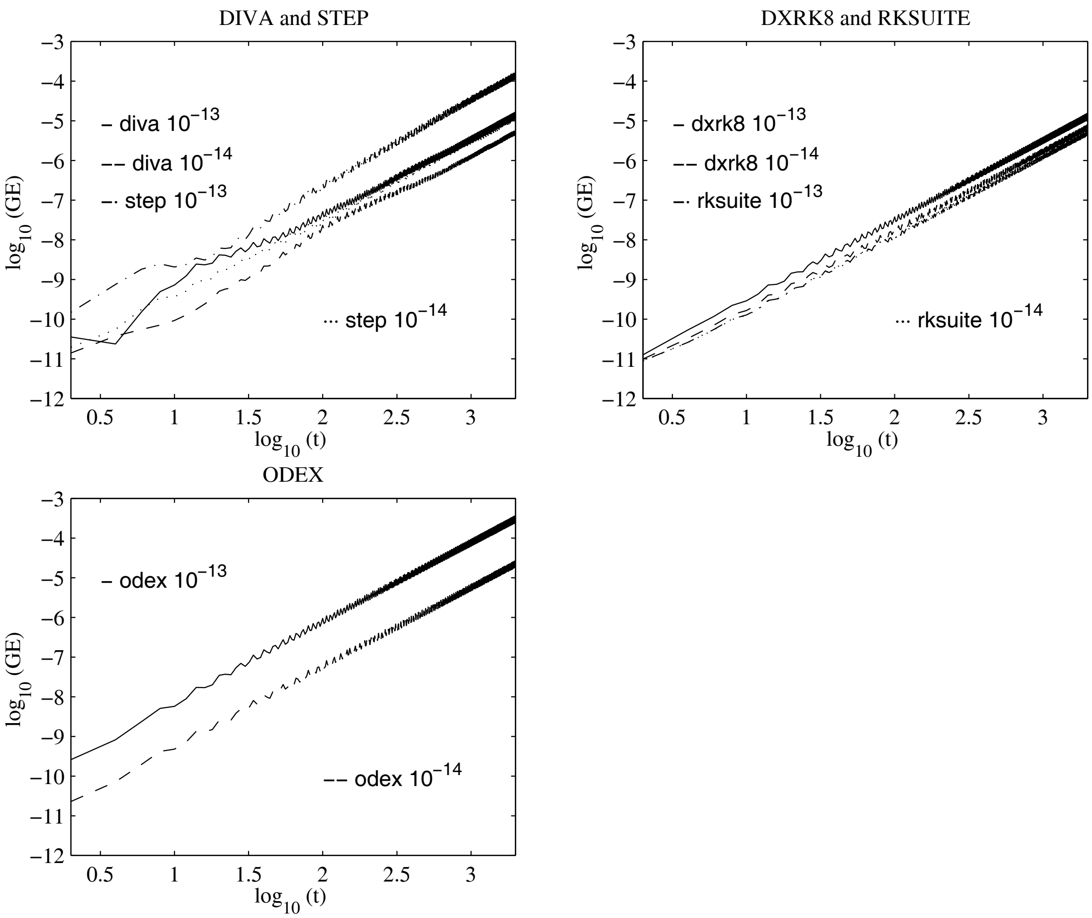

Figure 6: DE102 problem. $\log_{10} - \log_{10}$ graphs of $\mathcal{E}(t)$.