A Faster Version of the SWIFT–MVS Integrator and Implementation of the Yarkovsky Force

Miroslav Brož [1,2]

[1] Institute of Astronomy, Charles University, V Holešovičkách 2, 180 00 Prague, Czech Republic

[2] miroslav.broz@email.cz

Abstract: The package SWIFT [1, 2] is a well known and commonly used tool for solar system studies. It allows to integrate a set of massive, mutually interacting bodies, and massless "test particles". SWIFT includes 4 integration methods: Wisdom–Holman Mapping (WHM or MVS), Regularized Mixed Variable Symplectic (RMVS), a fourth order T+U Symplectic (TU4) and Bulirsch–Stoer (BS).

We have modified the original code in the following manner:

1. We have incorporated Yarkovsky (hereinafter such an integrator is referred with the suffix -Y) and Poynting–Robertson (-PR) dissipative forces into the integrators and checked their results againts analytic predictions.
2. We have implemented on-line digital filters based on the Kaiser windows (-F).
3. We have implemented the 2nd order symplectic integration scheme (MVS2) by Laskar and Robutel [3], which seems to be at least 2 times faster than MVS, while keeping the same relative energy error. However, the algorithm is not "regularized", ie. no close encounters are allowed.
4. We have parallelized integrators according to the OpenMP standard [4], which allows to run SWIFT on multiprocessor machines.

We will present results of tests involving the new integrators, the parallelization, the filter and the Yarkovsky implementations, etc. The SWIFT–RMVSY package is available on our web site [5].

Tests of integrators – Table of contents:

1., 2. Yarkovsky, Poynting–Robertson force; on-line digital filter:

- The theory of the Yarkovsky effect was published in [6, 7]. We have modified original SWIFT subroutines and added several new ones to incorporate the Yarkovsky force (and later also the Poynting–Robertson force to study dust particle dynamics). Our method corresponds to [8, 9].
- The addition of a weak dissipative force to the symplectic integrator did NOT affect its stability — it was proven by the comparison of secular changes of semimajor axis $\mathrm{d}\,a/\,\mathrm{d}\,t$ with the analytic estimates from Gauss equations [6, 7]. See also Fig. 1.

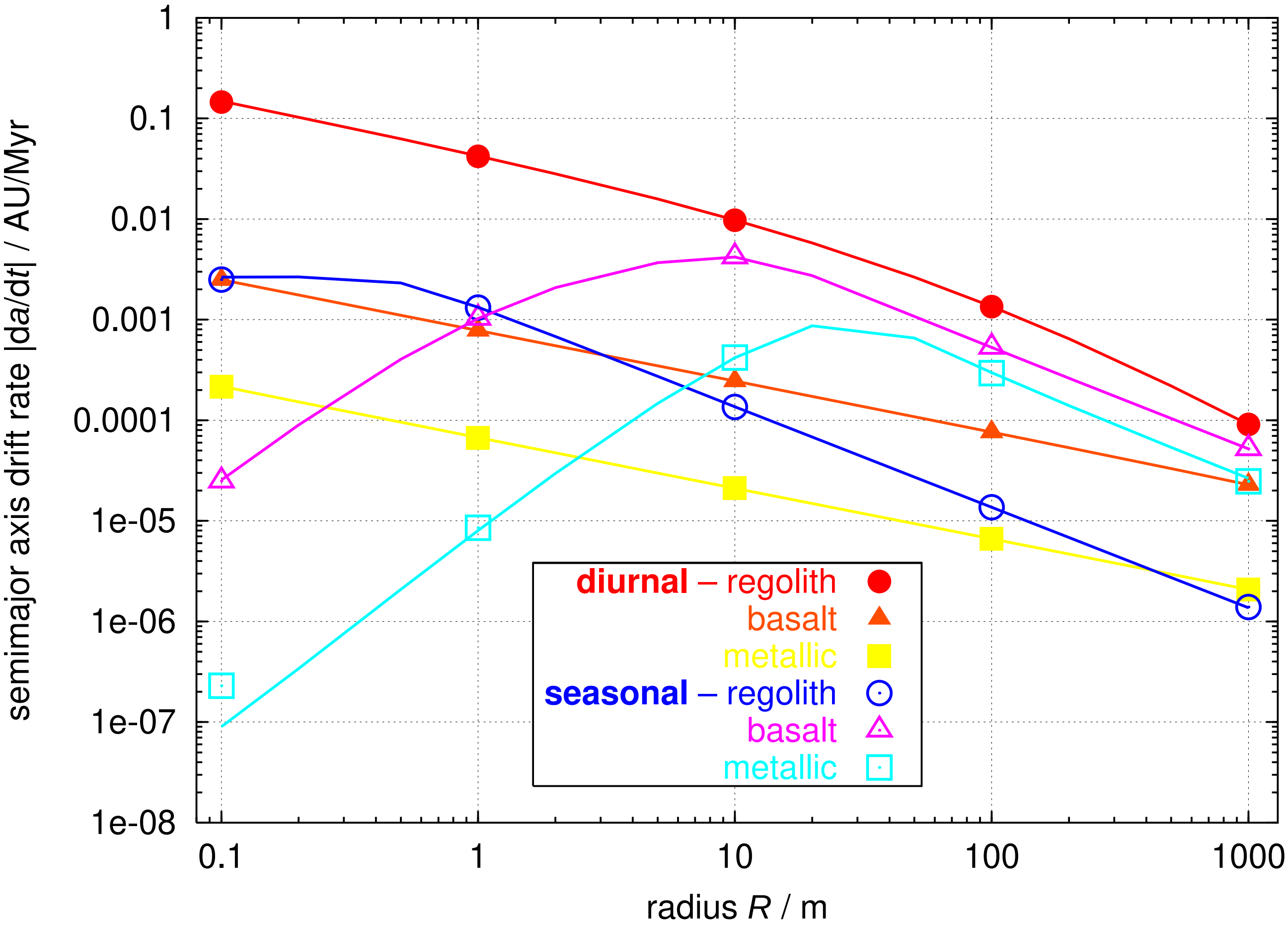


Fig. 1 — The absolute values of semimajor axis drift rates $\mathrm{d}\,a/\,\mathrm{d}\,t$ (in AU/Myr) for particles with several values of radii (from 0.1 m to 1 km) and 3 different types of thermal parameters: regolith, basalt and metallic like. Initial orbits of test particles were $a = 2\,\mathrm{AU}$, $e = 0$, $i = 0^\circ$. The two variants of Yarkovsky force (diurnal and seasonal) are plotted separately. Lines are the analytic estimates, points are computed numerically by LSM fit of the output $a(t)$. (The additional required parameters for the Yarkovsky effect model are: radius, bulk density, surface density, thermal conductivity, thermal capacity, albedo, IR emissivity, rotation period and spin axis orientation.)

- Implementation of the on-line digital filter (based on the Kaiser windows, [10]) enables a user to choose an appropriate sampling timestep, sequence of filters and decimation factors, suitable for his application. The output mean orbital elements are written to a separate binary file. This output file could be filtered later (off-line) by the running-box method (implemented in supplied `mean2prop` program). In a near future we plan to provide full computation of the synthetic proper elements in accordance with definition of [11].

- **Examples of current applications:** (i) N-body simulations of meteoroids (and dust particles) dynamics and their transport from main–belt to Mars (Earth) crossing orbits; (ii) long-term evolution of asteroids and their interaction with gravitational resonances; (iii) diffusion processes in asteroid families. See also Fig. 2 and references [14, 15, 16], where some results calculated with SWIFT–RMVSY were already published.

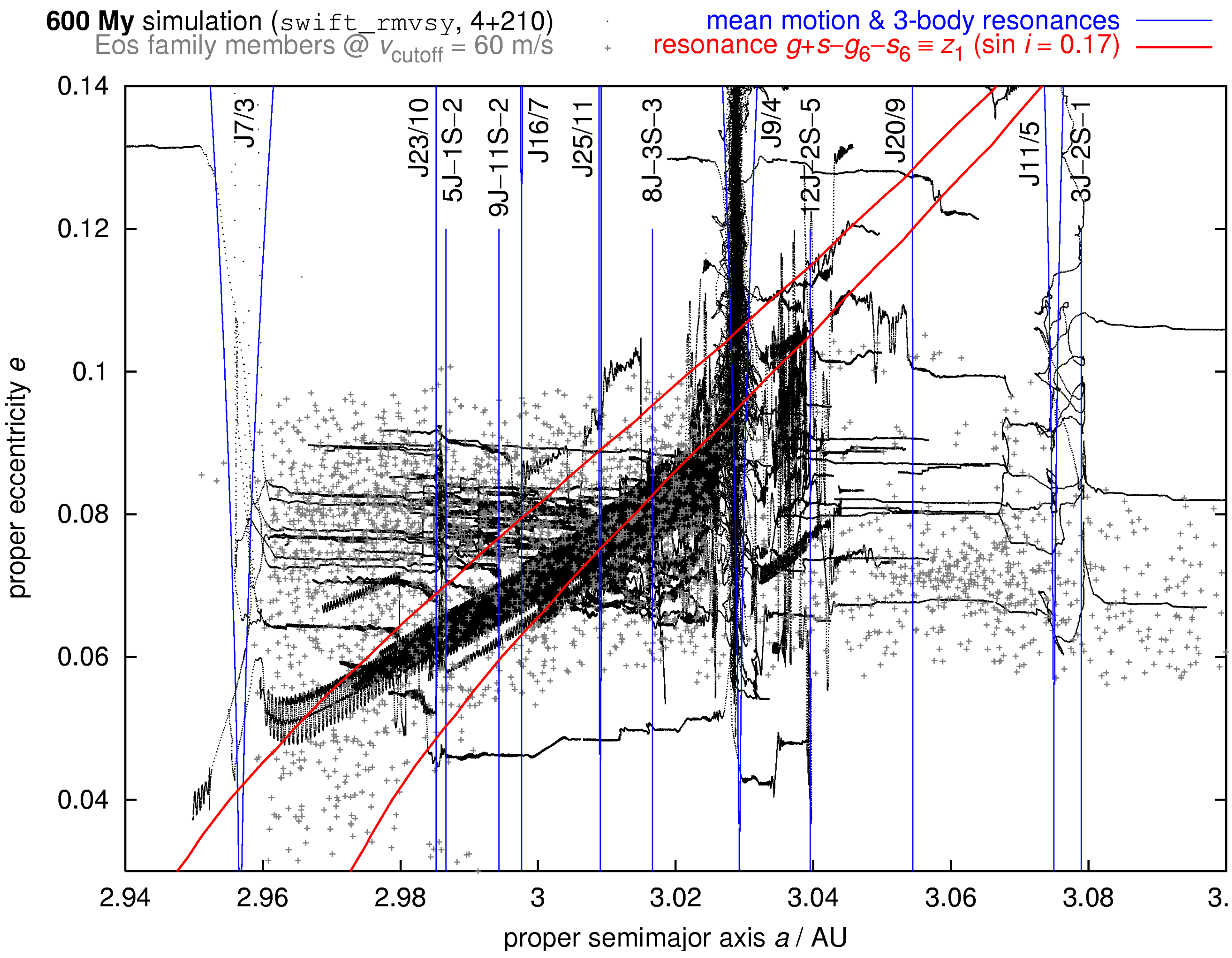
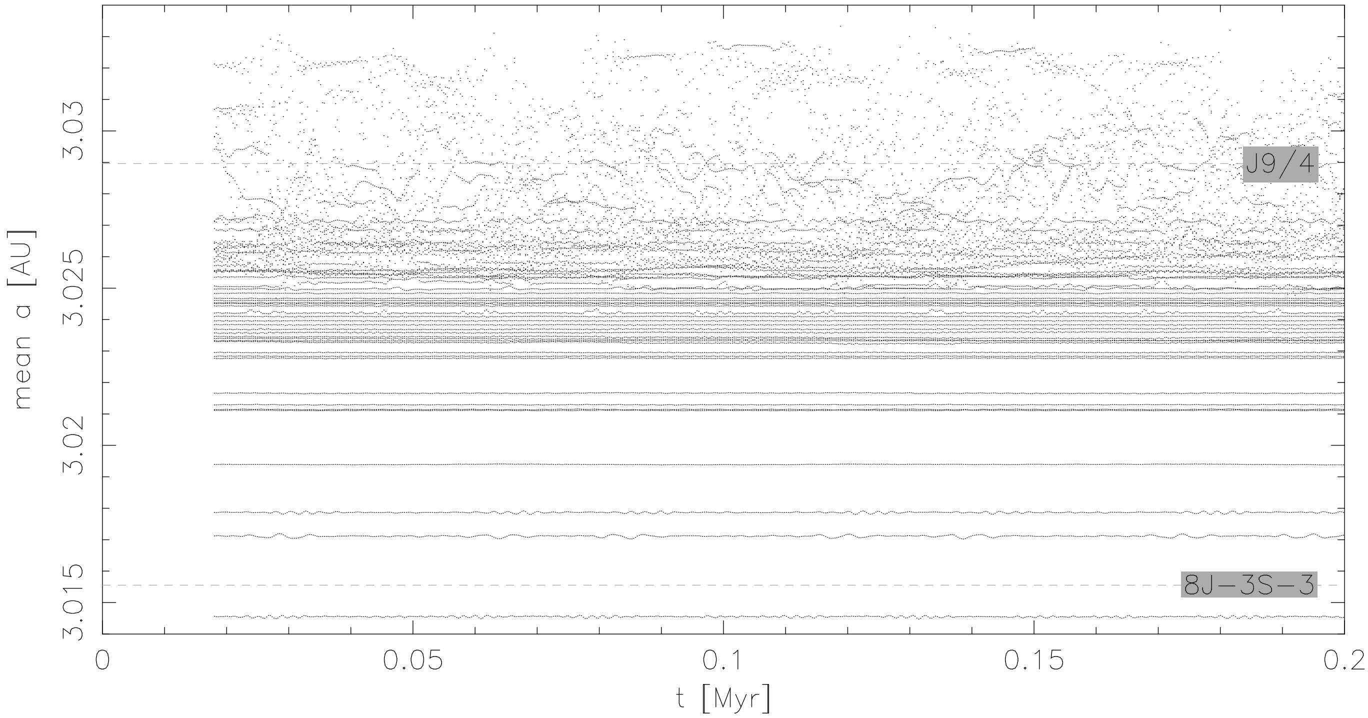


Fig. 2 — 600 Myr simulation of 210 Eos family asteroids using the SWIFT–RMVSY integrator. The plot shows their evolution in proper semimajor axis — proper eccentricity plane. The osculating orbital elements were at first on-line filtered by a sequence of 3 Kaiser window filters (→ mean elements) and at second off-line filtered by a running-box 5 Myr wide (→ proper elements). Asteroids decrease or increase semimajor axes (depending on the obliquity) due to Yarkovsky thermal force and subsequently interact with various mean motion and secular resonances, which are able to change significantly their eccentricities and inclinations. The results of the Eos family simulations are discussed by D. Vokrouhlický *et al.*: *Yarkovsky footprints in the Eos family.* (session 12–1 on Wednesday at 11:30)

3. Comparison of BS, MVS, RMVS and -FY integrators:

- Typical test run (if not specified): 4+50 particles, timestep = 20 days, stop time = 0.2 Myr. (Initial orbital elements of TPs are randomly distributed in the region of Eos family, just below the J9/4 mean motion resonance.)
- $a(t)$ plots for individual runs **all** look very similar, as they should; only chaotic orbits differ significantly. (The output of mean orbital elements starts at 0.02 Myr due to the initialization of the digital filter.)



- Integrators BS, BSFY, MVS, RMVS3, RMVSF and RMVSFY were tested; computed on Athlon/1.4 GHz.

integrator (compiler)	run-time / sec	notices
BS (omf77)	4092	local error 10^{-8}
BSFY (omf77)	12200	
MVS (omf77)	273	
RMVS3 (omf77)	356	
RMVSF (omf77)	480	
RMVSFY (omf77)	874	

On-line digital filter slows down the RMVS3 integrator by 30 %. Subroutines for Yarkovsky effect calculation slows integrator further by factor 1.8.

Bulirsch–Stoer integrator (with Yarkovsky subroutines) is approximately 10 times slower than RMVSFY.

4. Comparison of g77, omf77 and other compilers:

- One integrator RMVSFY, but 5 different compilers for Linux OS; computed on Celeron/1 GHz.

RMVSFY (g77)	1652	g77–0.5.24
RMVSFY (omf77)	1496	Omni–1.3 [12]
RMVSFY (fujitsu)	1371	Fujitsu F95
RMVSFY (lf95)	1147	Fujitsu/Lahey F95 Express 6.0
RMVSFY (pgf90)	1133	Portland group Fortran90
	1164	(with `-mp` on 1 CPU)

Compiler omf77 is faster than g77 by approx. 10 %. Commercial Fujitsu/-Lahey and Portland group compilers "beat" g77 by approx. 30 %.

5., 6. Parallel version of SWIFT:

- Parallel program RMVSFY on 1 and 2 processors; computed on PIII/800 MHz. (See [5] for downloads. The specification of OpenMP [4].)

RMVSFY (omf77)	1857	run only on 1 CPU
	1263	2 CPUs (SMP architecture)

On two processors the RMVSFY is (only) 1.5 faster than on single one.

- MVSFY integrator (without any regularization, no close encounters!), also compared with integration involving one half of test particles.

MVSFY (omf77)	1613	run only on 1 CPU (50 TPs)
	1044	2 CPUs (SMP architecture)
	846	1/2 of TPs
MVSFY (omf77)	1537	1 CPU, NTP = 500, t_{stop} = 0.02 Myr
	945	2 CPUs

MVSFY is only 1.2 times faster than RMVSFY due to Yarkovsky part of the code (as compared to factor 1.3 between pure MVS and RMVS3).

On 2 CPUs the integration by MVSFY takes 1.55 times shorter time span. (In case of 10 times larger number of TPs, ie. 500, the ratio is even slightly better: 1.62.)

When you manually split the job into two pieces and run two separate jobs, it will take 1.9 shorter time than the single run on 1 CPU.

(It means: you loose 20 % of the computation speed on 2 CPUs, compared with 2 single runs, but you gain a simple manipulation with output data files — there is no need to merge 2 output binary files.)

7., 8. Tests of MOSIX cluster:

- A heterogenous cluster with 8 CPUs, slow 10 Mb LAN; a single migrated job (on 1 CPU, Celeron/850 MHz), using MVSFY integrator. (See [13] for Mosix.)

MVSFY (omf77)	1624	run on home node, migration forbidden
	2030	started elsewhere, migrated to the same
	1654	the same run, but very low I/O

The speed of the migrated process is almost the same (98 %), but only for low I/O (luckily, it is a typical case for our runs).

- Test of parallel version of MVSFY on Mosix.

MVSFY (omf77)	5078	2 threads on single CPU!

It is NOT yet possible to use parallel version of SWIFT on Mosix, because Mosix does not support distributed shared memory. $\rightarrow$ One have to split the calculation to several separate runs (eg. with `swiftsplit` *script, included in the SWIFT–RMVSY package).*

(However, authors of Mosix annouce the future support of DSM. Mosix cluster is now still useful for a comfortable job management.)

9. New symplectic integrator SWIFT–MVS2FY(PR) ($\tau^4\varepsilon + \tau^2\varepsilon^2$):

- A new implementation of the integrator $\mathcal{SBAB}_2$ [3], which is of the order $\tau^4\varepsilon + \tau^2\varepsilon^2$ and of index 2 (ie. the number of evaluations of Hamiltonian parts $\mathcal{A}$ and $\mathcal{B}$). The intregrator SWIFT–MVS2FY is yet without regularization! Used **timestep** = 80 days (ie. **4 times larger**, but still more precise than MVS).

MVS2FY (omf77)	940	1 CPU
	588	2 CPUs
	945	1 CPU, `-omp` (without OpenMP pragma)

MVS2FY integrator is approx. 1.7 times faster than MVSFY and 2 times faster than RMVSFY; one can still enlarge the timestep (up to 100 days?).

- **Total energy error**, which depends on an integrator timestep. We integrated Sun – Jupiter – Saturn system for 0.1 Myr, with different timesteps ranging from 0.1 day up to 1000 days. (See Fig. 3.)
- **CPU time** dependce on a relative energy error (which might be selected above by the appropriate timestep d t); computed on Athlon 1.4 GHz. (See Fig. 4.)

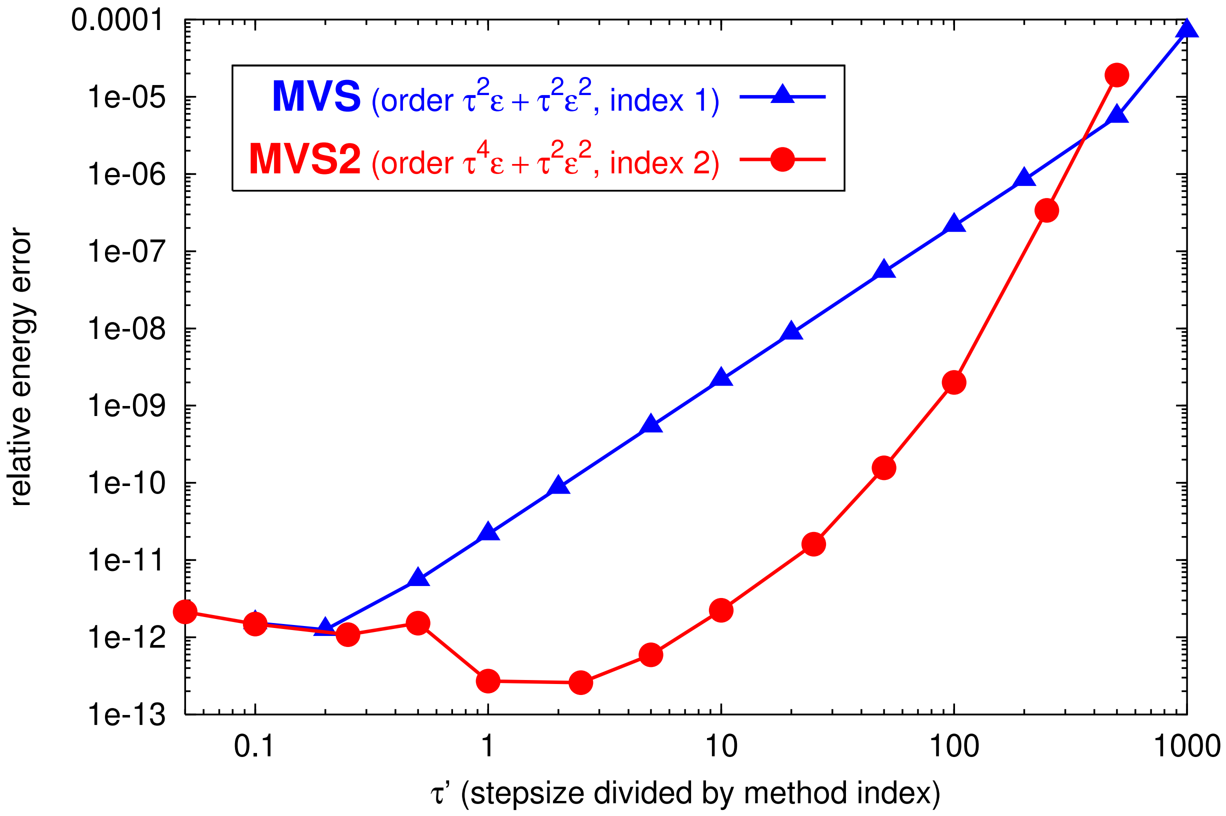


Fig. 3 — Plot relative timestep τ' — relative energy error. (τ' is timestep d t divided by method index (ie. 1 for the classical leap-frog MVS and 2 for the new MVS2. Relative energy error is computed as standard deviation of total energy σ_E, in the interval of integration, divided by mean total energy E.)

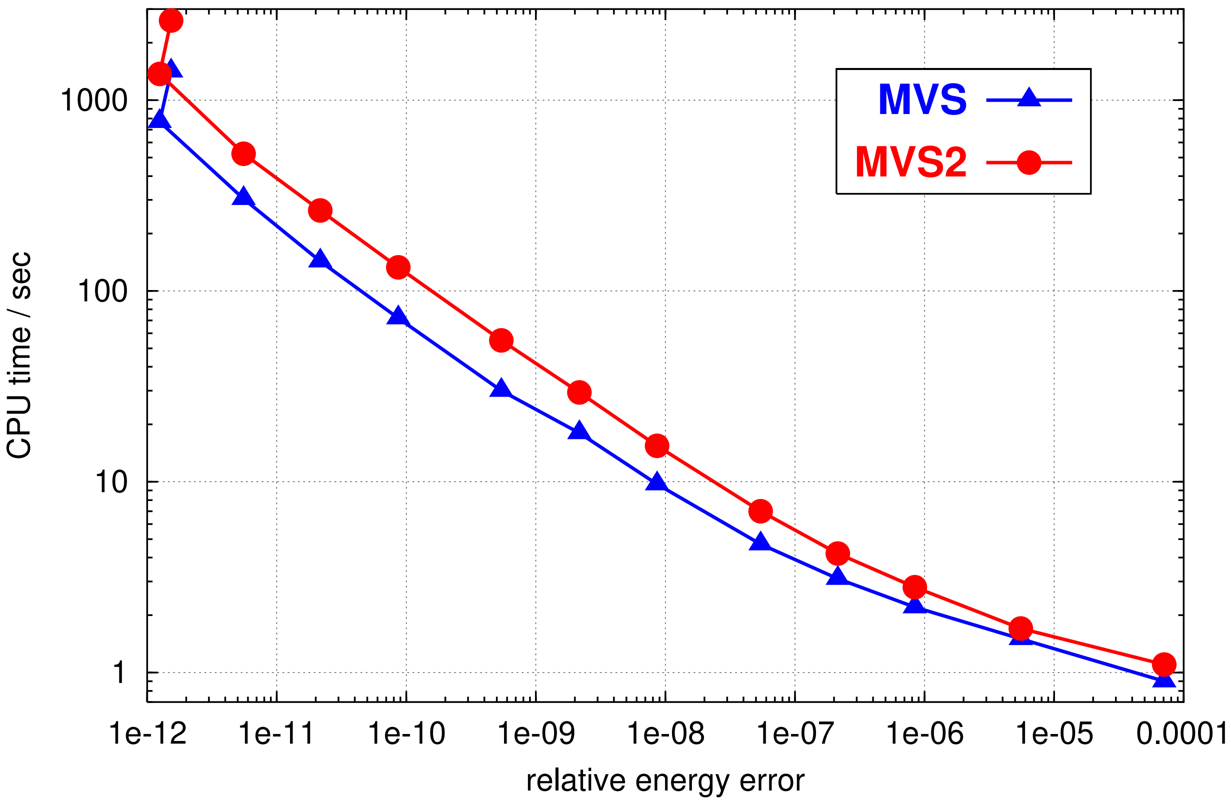


Fig. 4 — Dependence of CPU time on relative energy error for both integrators MVS2F and MVSF. **Timestep for MVS2F was selected 4 times larger than for MVSF**.

MVS2F seems to be more precise (at least by two orders of magnitude!) in a wide range of timesteps — from 2 up to 200 days. It is on average 1.8 times slower than MVSF for a given timestep.

Usual timestep used with MVS (or RMVS3) integrator is 10 or 20 days, while we can safely use 100 days with MVS2F, keeping the same relative energy error.

Conclusions:

- We have presented results of tests of SWIFT numerical integrators under various conditions (different compilers, multiprocessor machines).
- We have modified the SWIFT code to account for Yarkovsky and Poynting–Robertson non-gravitational effects and ensured that these (weak) dissipative forces did not affected the stability of the integrator.
- SWIFT–MVS2FY seem to be a more efficient integrator than original MVS; we will work on the regularization algorithm to account for close encounters.
- The new integrators might be useful tools for Solar system studies, especially in case of long-term evolution of small bodies.

References:

[1] Levison, H. F., Duncan, M. J., Icarus **108**, 18 (1994).
[2] *SWIFT.* `http://www.boulder.swri.edu/~hal/swift.html`
[3] Laskar, J., Robutel, P., Celest. Mech. Dyn. Astr. **80**, 39 (2001).
[4] *OpenMP specification.* `http://www.openmp.org`
[5] *Yarko–site.* `http://sirrah.troja.mff.cuni.cz/yarko-site/`
[6] Vokrouhlický, D., Astron. Astrophys. **335**, 1093 (1998).
[7] Vokrouhlický, D., Farinella, P., Astron. J. **118**, 3049 (1999).
[8] Codeiro, R. R. *et al.*, Celest. Mech. Dyn. Astr. **65**, 407 (1997).
[9] Mikkola, S., Celest. Mech. Dyn. Astr. **68**, 249 (1998).
[10] Quinn, T. R. *et al.*, Astron. J. **101**, 2287 (1991).
[11] Kneževiċ, Z., Milani, A., Celest. Mech. Dyn. Astr. **78**, 17, (2000).
[12] *Omni compiler.* `http://phase.etl.go.jp/Omni/`
[13] *Mosix cluster.* `http://www.mosix.org`
[14] Vokrouhlický, D., Brož, M., Farinella, P., Kneževiċ, Z.: *Yarkovsky-Driven Leakage of Koronis Family Members.* Icarus **150**, 78 (2001).
[15] Bottke, W. F., Vokrouhlický, D., Brož, M., Nesvorný, D., Morbidelli, A.: *Dynamical Spreading of Asteroid Families by the Yarkovsky Effect.* Science **294**, 1693 (2001).
[16] Bottke, W. F., Vokrouhlický, D., Rubincam, D. P., Brož, M.: *Dynamical evolution of asteroids and meteoroids using the Yarkovsky effect.* In: Asteroids III, eds. W. F. Bottke *et al.*, in press (2002).