

IMPLEMENTING FEW-BODY ALGORITHMIC REGULARIZATION WITH POST-NEWTONIAN TERMS

SEPPO MIKKOLA¹ AND DAVID MERRITT²

¹ Turku Observatory, University of Turku, Väisäläntie 20, Piikkiö, Finland; Seppo.Mikkola@utu.fi

² Center for Computational Relativity and Gravitation and Department of Physics, Rochester Institute of Technology, Rochester, NY 14623, USA;
David.Merritt@rit.edu

Received 2007 September 21; accepted 2008 March 9; published 2008 May 14

ABSTRACT

We discuss the implementation of a new regular algorithm for simulation of the gravitational few-body problem. The algorithm uses components from earlier methods, including the chain structure, the logarithmic Hamiltonian, and the time-transformed leapfrog. This algorithmic regularization code, AR-CHAIN, can be used for the normal N -body problem, as well as for problems with softened potentials and/or with velocity-dependent external perturbations, including post-Newtonian terms, which we include up to order PN2.5. Arbitrarily extreme mass ratios are allowed. Only linear coordinate transformations are used and thus the algorithm is somewhat simpler than many earlier regularized schemes. We present the results of performance tests which suggest that the new code is either comparable in performance or superior to the existing regularization schemes based on the Kustaanheimo–Stiefel (KS) transformation. This is true even for the two-body problem, independent of eccentricity. An important advantage of the new method is that, contrary to the older KS-CHAIN code, zero masses are allowed. We use our algorithm to integrate the orbits of the S stars around the Milky Way supermassive black hole for one million years, including PN2.5 terms and an intermediate-mass black hole. The three S stars with shortest periods are observed to escape from the system after a few hundred thousand years.

Key words: black hole physics – celestial mechanics – Galaxy: center – methods: N -body simulations – relativity – stellar dynamics

1. INTRODUCTION

After the introduction of electronic computers, those who carried out simulations of the gravitational N -body problem soon realized that the classical methods of numerical integration were often not satisfactorily accurate due simply to the strong $1/r^2$ character of the gravitational force.

The situation changed when Kustaanheimo & Stiefel (1965) published their Kustaanheimo–Stiefel (KS) transformation from four-dimensional to three-dimensional space. The special case of a planar system had been known for a long time (Levi-Civita 1920) but it had turned out that a similar transformation in the three-dimensional space was not possible. The situation for the general N -body problem improved only after the KS transformation became well known, due in large part to publication by Stiefel & Scheifele (1971) of their text, which comprehensively discussed the application of the KS transformation to the perturbed two-body problem. Later, Aarseth & Zare (1974), Heggie (1974), Zare (1974), and Mikkola & Aarseth (1993) applied the KS transformation to the general three-body and N -body problems.

An entirely new way of regularizing close encounters was invented simultaneously by Mikkola & Tanikawa (1999a, 1999b) and by Preto & Tremaine (1999). This new method introduced the so-called logarithmic Hamiltonian (LogH). Together with the simple leapfrog algorithm, this new method gives regular results for close encounters; in fact a correct trajectory is obtained for the two-body problem. Other similar methods are described by Huang & Leimkuhler (1997).

The remaining problem was that none of these regularization methods could be easily applied to systems with extremely large mass ratios. In an attempt to solve this problem, Mikkola & Aarseth (2002) introduced the time-transformed leapfrog (TTL). This method is in some cases mathematically equivalent to the LogH method, but it is more general, and arbitrary mass ratios

are allowed. The drawback of this method, however, is that in some cases the roundoff error affects the results considerably. In these new methods (LogH, TTL) the regularization is achieved by using the leapfrog, hence the name “algorithmic regularization.” More details about the thus-far mentioned methods can be found in the book by Aarseth (2003).

Since the leapfrog alone is rarely accurate enough, one must supplement the method with the extrapolation algorithm (Gragg 1964, 1965; Bulirsch & Stoer 1966; Press et al. 1986), or with higher-order leapfrogs (Yoshida 1990), in order to get highly accurate results. The efficiency of the extrapolation procedure requires that the basic algorithm (leapfrog in algorithmic regularization, modified midpoint method in the KS-regularized codes) have a certain symmetry. In the case of the leapfrog this means time reversibility. If the system has velocity-dependent forces, such as relativistic post-Newtonian (PN) terms (Soffel 1989), then the required symmetry is more difficult to obtain. One way to solve this problem was recently obtained by Mikkola & Merritt (2006), who formally doubled the dimensionality of the parameter space and constructed a generalized midpoint method (GAR). This algorithm allows the use of algorithmic regularization and velocity-dependent perturbations. However, these authors discussed the new algorithm essentially only for the case of the perturbed two-body problem.

In this paper, we discuss the details of our most recent implementation of the algorithmic regularization method. This uses the chain structure, the same structure as in the KS-CHAIN algorithm of Mikkola & Aarseth (1993). That device, together with a new time-transformation function, significantly reduces the roundoff problems and makes the new code a good alternative for simulations of strongly-interacting few-body systems. In the final section, we present some applications of the chain routine to the problem of relativistic orbits around the supermassive black hole (SMBH) at the Galactic center.

2. NOTATION

In this paper, we use the following basic symbols:

$G = 1$	gravitational constant
t	time
m_k	mass of body k
\mathbf{p}_k	momentum
\mathbf{r}_k	position
$\mathbf{v}_k = \dot{\mathbf{r}}_k$	velocity
$T = \sum_{k=1}^N \mathbf{p}_k^2 / 2m_k$	kinetic energy
$U = \sum_{0 < i < j \leq N} m_i m_j / \mathbf{r}_i - \mathbf{r}_j $	potential energy
$E = T - U$	total energy
$B = -E = U - T$	binding energy

3. REGULAR ALGORITHMS

In the few-body problem, close approaches of two bodies are common, and these events require highly accurate computations since the energies involved are large. Thus, a method that can accurately advance the motions of few-body systems must be accurate and efficient for the two-body problem. In addition to the classical KS transformation there are some new algorithms that satisfy this requirement.

An additional, and most important, problem is the roundoff error. This becomes a major problem if, e.g., center-of-mass coordinates are used and there are close binaries and/or close encounters in the simulated system. The solution to this problem is the use of the chain structure, originally applied with the KS transformation (Mikkola & Aarseth 1993) in order to KS-regularize all the short(est) distances. Later it became clear that the chain structure was also beneficial in significantly reducing the roundoff error.

In this section, we first review the ingredients we need in the construction of the final N -body algorithm. These basic algorithms produce regular results in close approaches and their results can be improved to high precision using, e.g., the Bulirsch & Stoer (1966) extrapolation method.

3.1. LogH, TTL, and GAR

3.1.1. LogH

Recently, Mikkola & Tanikawa (1999a) and Preto & Tremaine (1999) pointed out that the LogH in extended phase space,

$$\Lambda = \ln(T + B) - \ln(U), \quad (1)$$

where B (the binding energy) is the momentum of time, gives the equations of motion in the form

$$t' = \frac{\partial \Lambda}{\partial B} = 1/(T + B) \quad (2)$$

$$\mathbf{r}'_k = \frac{\partial T}{\partial \mathbf{p}_k} \bigg/ (T + B) \quad (3)$$

$$B' = \frac{\partial U}{\partial t} \bigg/ U \quad (4)$$

$$\mathbf{p}'_k = \frac{\partial U}{\partial \mathbf{r}_k} \bigg/ U. \quad (5)$$

Here we include Equation (4) although this is needed only if there is a time-dependent potential to be added to the N -body potential U . Since the right-hand sides of these equations do not depend on the left-hand-side variables, a leapfrog algorithm is possible. This may be symbolized as

$$X(h/2)V(h)X(h) \cdots V(h)X(h/2), \quad (6)$$

where $X(s)$ means solution for the coordinate Equations (2), (3) with constant B and \mathbf{p}_k over an integration step of length s :

$$\delta t = s/(T + B); \quad t \rightarrow t + \delta t; \quad \mathbf{r}_k \rightarrow \mathbf{r}_k + \delta t \frac{\partial T}{\partial \mathbf{p}_k}. \quad (7)$$

Correspondingly $V(s)$ signifies the operation

$$\tilde{\delta t} = s/U; \quad B \rightarrow B + \tilde{\delta t} \frac{\partial U}{\partial t}; \quad \mathbf{p}_k \rightarrow \mathbf{p}_k + \tilde{\delta t} \frac{\partial U}{\partial \mathbf{r}_k}, \quad (8)$$

which solves Equations (4) and (5) for constant t and \mathbf{r}_k .

For the case of two bodies only, this algorithm produces correct trajectories, with only an $O(h^3)$ phase error. This is true even for the collision orbit and the energy conservation is, in this case, typically at the machine precision level.

3.1.2. TTL

The second important ingredient in our algorithm is the TTL (Mikkola & Aarseth 2002). The basic idea is to introduce a coordinate-dependent time-transformation function $\Omega(\dots, \mathbf{r}_k, \dots)$ and a new variable ω which is supposed to have the same numerical value as Ω , but that value is obtained from the differential equation

$$\dot{\omega} = \dot{\Omega} = \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot \mathbf{v}_k. \quad (9)$$

The equations of motion can be written as

$$t' = 1/\omega \quad (10)$$

$$\mathbf{r}'_k = \mathbf{v}_k/\omega \quad (11)$$

$$\mathbf{v}'_k = \mathbf{A}_k/\Omega \quad (12)$$

$$\omega' = \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot \mathbf{v}_k/\Omega, \quad (13)$$

where \mathbf{A}_k is the acceleration of the k th particle. The structure of these equations allows the construction of a leapfrog algorithm:

$X(s)$:

$$\delta t = s/\omega; \quad t \rightarrow t + \delta t; \quad \mathbf{r}_k \rightarrow \mathbf{r}_k + \delta t \mathbf{v}_k \quad (14)$$

$V(s)$:

$$\tilde{\delta t} = s/\Omega; \quad \mathbf{v}_k \rightarrow \mathbf{v}_k + \tilde{\delta t} \mathbf{A}_k; \quad \omega \rightarrow \omega + \tilde{\delta t} \langle \dot{\Omega} \rangle, \quad (15)$$

where $\langle \dot{\Omega} \rangle$ is the average of this quantity over the step, i.e.,

$$\langle \dot{\Omega} \rangle = \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot (\mathbf{v}_k^{\text{old}} + \mathbf{v}_k^{\text{new}})/2. \quad (16)$$

Here, the superscripts “old” and “new” refer to \mathbf{v}_k values before and after the velocity advancement in the operation (15).

3.1.3. GAR

Here we concisely review the GAR method. Following Mikkola & Merritt (2006), we consider a differential equation

$$\dot{z} = f(z), \quad (17)$$

and an approximation for its solution (over a short step $= h$) written as

$$z(h) \approx z(0) + d(z(0), h). \quad (18)$$

Here the increment $d(z_0, h)$ can be any approximation suitable for the particular equation in question. If one writes, instead of Equation (17), the two equations

$$\dot{x} = f(y); \quad \dot{y} = f(x), \quad (19)$$

and solves this pair with the initial values $x(0) = y(0) = z(0)$, the solution obviously is $x(t) = y(t) = z(t)$. Using the pair of equations one can write a leapfrog as

$$x_{\frac{1}{2}} = x_0 + \frac{h}{2} f(y_0); \quad y_1 = y_{\frac{1}{2}} + h f(x_{\frac{1}{2}}); \quad x_1 = x_{\frac{1}{2}} + \frac{h}{2} f(y_1),$$

which is actually nothing but the well-known modified midpoint method. In the above, one can split the advancement of y into two operations to get

$$x_{\frac{1}{2}} = x_0 + \frac{h}{2} f(y_0); \quad y_{\frac{1}{2}} = y_0 + \frac{h}{2} f(x_{\frac{1}{2}}) \quad (20)$$

$$y_1 = y_{\frac{1}{2}} + \frac{h}{2} f(x_{\frac{1}{2}}); \quad x_1 = x_{\frac{1}{2}} + \frac{h}{2} f(y_1), \quad (21)$$

and this can be readily generalized using the more general increment $d(z, h)$. This results in the generalized midpoint method

$$x_{\frac{1}{2}} = x_0 + d\left(y_0, \frac{h}{2}\right); \quad y_{\frac{1}{2}} = y_0 - d\left(x_{\frac{1}{2}}, -\frac{h}{2}\right), \quad (22)$$

$$y_1 = y_{\frac{1}{2}} + d\left(x_{\frac{1}{2}}, \frac{h}{2}\right); \quad x_1 = x_{\frac{1}{2}} - d\left(y_1, -\frac{h}{2}\right). \quad (23)$$

This method has the great advantage that one can use any special approximation d and the algorithm is time reversible (albeit only in the extended x - y -space) and thus suitable to be used as the basic integrator in an extrapolation method. Specifically, one may stress that the increment d may be computed using LogH or TTL which produce good approximations in the case of close encounters. In addition, the appearance of velocity-dependent forces is not problematic here, as shown by Mikkola & Merritt (2006).

4. THE AR-CHAIN ALGORITHM

While in the algorithms discussed above one can use any coordinate system, the roundoff error can be a major problem in the case of close encounters if the coordinates of the approaching bodies are measured from a distant origin. This problem is significantly reduced by utilizing the chain structure. This was originally used in a KS-regularization algorithm (Mikkola & Aarseth 1993) in order to get all the short distances regularized by the KS transformation, but here the sole purpose is the roundoff reduction, for which the device has proved itself.

4.1. Basic Formulation

Let $T = \frac{1}{2} \sum_{k=1}^N m_k v_k^2$ be the kinetic energy and $U = \sum_{i < j \leq N} m_i m_j / r_{ij}$ the potential such that the total energy is $E = T - U$; the binding energy is $B = U - T$.

One forms a chain of particles such that the shortest relative vectors are in the chain (Mikkola & Aarseth 1993). We stress again that the main purpose of using the chain structure in this method is to reduce the (often significant) effects of the roundoff error.

Let us collect the chain coordinates $X_k = r_{i_k} - r_{j_k}$ in the vector $X = (X_1, X_2, \dots, X_{N-1})$ and let the corresponding velocities $V_k = v_{i_k} - v_{j_k}$ be in the vector $V = (V_1, V_2, \dots, V_{N-1})$. Then, the Newtonian equations of motion may be formally written as

$$\dot{X} = V \quad (24)$$

$$\dot{V} = A(X) + f, \quad (25)$$

where A is the N -body acceleration and f is some external acceleration (e.g. due to other bodies).

One may use the two equivalent time transformations (Mikkola & Merritt 2006)

$$ds = [\alpha(T + B) + \beta\omega + \gamma] dt = [\alpha U + \beta\Omega + \gamma] dt, \quad (26)$$

where s is the new independent variable, α, β , and γ are adjustable constants, Ω is an optional function of the coordinates $\Omega = \Omega(X)$, while the initial value $\omega(0) = \Omega(0)$ and the differential equation

$$\dot{\omega} = \frac{\partial \Omega}{\partial X} \cdot V \quad (27)$$

determine the values of ω (in fact, $\omega(t) = \Omega(t)$ along the exact solution).

The time transformation thus introduced regularizes the two-body collisions if one uses the simple leapfrog algorithm as a basic integrator, results from which can, and must, be improved using an extrapolation method (e.g. Bulirsch & Stoer 1966; Press et al. 1986).

It is possible to divide the equations of motion into two categories (where derivatives with respect to the new independent variable s are denoted by a prime):

Coordinate equations:

$$t' = 1/(\alpha(T + B) + \beta\omega + \gamma) \quad (28)$$

$$X' = t' V. \quad (29)$$

Velocity equations:

$$\tilde{t}' = 1/(\alpha U + \beta\Omega + \gamma) \quad (30)$$

$$V' = \tilde{t}'(A + f) \quad (31)$$

$$\omega' = \tilde{t}' \frac{\partial \Omega}{\partial X} \cdot V \quad (32)$$

$$B' = -\tilde{t}' \frac{\partial T}{\partial V} \cdot f. \quad (33)$$

In these equations the right-hand sides do not depend on the variables on the left. Consequently it is possible to construct a regular leapfrog algorithm for obtaining the solutions. The leapfrog results then can easily be improved with the extrapolation method.

4.2. Details

4.2.1. Finding and Updating the Chain

First we find the shortest interparticle vector which is adopted as the first part of the chain. The chain is then augmented by adding the relative vector to the particle nearest to one or the other end of the existing chain. When all particles are included, they are re-numbered along the chain as $1, 2, \dots, N$ for ease of programming.

To reduce roundoff problems, the transformation from the old chain vectors \mathbf{X}_k to the new ones is done directly by expressing the new chain vectors as sums of the old ones as in Mikkola & Aarseth (1993).

4.2.2. Transformations

When the particles are renamed along the chain as $1, 2, \dots, N$ one can evaluate

$$\mathbf{X}_k = \mathbf{r}_{k+1} - \mathbf{r}_k \quad (34)$$

$$\mathbf{V}_k = \mathbf{v}_{k+1} - \mathbf{v}_k. \quad (35)$$

The center-of-mass quantities are

$$M = \sum_k m_k \quad (36)$$

$$\mathbf{r}_{\text{cm}} = \sum_k m_k \mathbf{r}_k / M \quad (37)$$

$$\mathbf{v}_{\text{cm}} = \sum_k m_k \mathbf{v}_k / M. \quad (38)$$

The inverse transformation is done by simple summation

$$\tilde{\mathbf{r}}_1 = \mathbf{0} \quad (39)$$

$$\tilde{\mathbf{v}}_1 = \mathbf{0} \quad (40)$$

$$\tilde{\mathbf{r}}_{k+1} = \tilde{\mathbf{r}}_k + \mathbf{X}_k \quad (41)$$

$$\tilde{\mathbf{v}}_{k+1} = \tilde{\mathbf{v}}_k + \mathbf{V}_k, \quad (42)$$

followed by reduction to the center of mass

$$\tilde{\mathbf{r}}_{\text{cm}} = \sum_k m_k \tilde{\mathbf{r}}_k / M \quad (43)$$

$$\tilde{\mathbf{v}}_{\text{cm}} = \sum_k m_k \tilde{\mathbf{v}}_k / M \quad (44)$$

$$\mathbf{r}_k = \tilde{\mathbf{r}}_k - \tilde{\mathbf{r}}_{\text{cm}} \quad (45)$$

$$\mathbf{v}_k = \tilde{\mathbf{v}}_k - \tilde{\mathbf{v}}_{\text{cm}}. \quad (46)$$

Note that it is not always necessary to reduce the coordinates to the center-of-mass system since accelerations only depend on the differences.

4.2.3. Equations of Motion and the Leapfrog

One writes the equations of motion as

$$\dot{\mathbf{X}}_k = \mathbf{V}_k \quad (47)$$

$$\dot{\mathbf{V}}_k = \mathbf{F}_{k+1} - \mathbf{F}_k + \mathbf{f}_{k+1} - \mathbf{f}_k, \quad (48)$$

where \mathbf{f}_k are the individual external accelerations and the N -body accelerations \mathbf{F}_k are

$$\mathbf{F}_k = - \sum_{j \neq k} m_j \frac{\mathbf{r}_{jk}}{|\mathbf{r}_{jk}|^3}, \quad (49)$$

where, for $j < k$,

$$\mathbf{r}_{jk} = \begin{cases} \mathbf{r}_k - \mathbf{r}_j; & \text{if } k > j + 2 \\ \mathbf{X}_j; & \text{if } k = j + 1 \\ \mathbf{X}_j + \mathbf{X}_{j+1}; & \text{if } k = j + 2. \end{cases} \quad (50)$$

For $k > j$ one uses $\mathbf{r}_{jk} = -\mathbf{r}_{kj}$. In the acceleration computation the use of \mathbf{X}_j and $\mathbf{X}_j + \mathbf{X}_{j+1}$ reduces the roundoff effect significantly. This is one of the most important features of the algorithm.

The kinetic energy is evaluated as usual

$$T = \frac{1}{2} \sum_k m_k \mathbf{v}_k^2, \quad (51)$$

while the potential

$$U = \sum_{i < j} \frac{m_i m_j}{|\mathbf{r}_{ij}|} \quad (52)$$

is obtained along with the accelerations according to Equation (50). For the time transformation function Ω it seems advantageous to use

$$\Omega = \sum_{i < j} \frac{\Omega_{ij}}{|\mathbf{r}_{ij}|}. \quad (53)$$

Here Ω_{ij} are adjustable constants (see below).

Now one is able to evaluate the two time transformation functions

$$t' = 1/(\alpha(T + B) + \beta\omega + \gamma) \quad (54)$$

$$\tilde{t}' = 1/(\alpha U + \beta\Omega + \gamma), \quad (55)$$

which are equivalent along the correct solution, i.e. $t' = \tilde{t}'$. The evolution of ω , with $\omega(0) = \Omega(0)$, is obtained by

$$\dot{\omega} = \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot \mathbf{v}_k. \quad (56)$$

In the presence of external perturbations the binding energy evolves according to

$$\dot{B} = - \sum_k m_k \mathbf{v}_k \cdot \mathbf{f}_k. \quad (57)$$

The leapfrog for the chain vectors \mathbf{X}_k and \mathbf{V}_k can be written as the two mappings:

$\mathbf{X}(s)$:

$$\delta t = s/(\alpha(T + B) + \beta\omega + \gamma) \quad (58)$$

$$t = t + \delta t \quad (59)$$

$$\mathbf{X}_k \rightarrow \mathbf{X}_k + \delta t \mathbf{V}_k \quad (60)$$

$\mathbf{V}(s)$:

$$\tilde{\delta t} = s/(\alpha U + \beta\Omega + \gamma) \quad (62)$$

$$\mathbf{V}_k \rightarrow \mathbf{V}_k + \tilde{\delta}t(\mathbf{F}_{k+1} - \mathbf{F}_k + \mathbf{f}_{k+1} - \mathbf{f}_k) \quad (63)$$

$$\mathbf{B} \rightarrow \mathbf{B} + \tilde{\delta}t \sum_k (-m_k \langle \mathbf{v}_k \rangle \cdot \mathbf{f}_k) \quad (64)$$

$$\omega \rightarrow \omega + \tilde{\delta}t \sum_k \frac{\partial \Omega}{\partial \mathbf{r}_k} \cdot \langle \mathbf{v}_k \rangle, \quad (65)$$

where $\langle \mathbf{v}_k \rangle$ is the average of the initial and final \mathbf{v} (obtained from the \mathbf{V} according to the equations in Section 4.2.2). It is necessary to evaluate the individual velocities \mathbf{v}_k because the expressions for \mathbf{B}' and ω' in terms of the chain vector velocities \mathbf{V}_k are rather cumbersome.

A leapfrog step can be written as

$$\mathbf{X}(h/2)\mathbf{V}(h)\mathbf{X}(h/2)$$

and a sequence of n steps as

$$\mathbf{X}(h/2)[\Pi_{v=1}^{n-1}(\mathbf{V}(h)\mathbf{X}(h))]\mathbf{V}(h)\mathbf{X}(h/2).$$

This is useful with the extrapolation method when advancing the system over a total time interval of length $= nh$.

4.3. Time Transformation Alternatives

If one takes

$$\Omega_j = m_i m_j, \quad (66)$$

then $\alpha = 0$, $\beta = 1$, $\gamma = 0$ is mathematically equivalent to $\alpha = 1$, $\beta = \gamma = 0$ as was shown in Mikkola & Aarseth (2002). However, numerically these are not equivalent, mainly due to roundoff errors in updating the value of Ω , and the LOGH alternative is numerically more stable. However, for proper treatment of small bodies some function Ω is to be used.

For increased numerical stability in the motions of the large bodies, and smoothing of the encounters of small bodies, the recipe is $\alpha = 1$, $\beta \neq 0$ (but small) and

$$\Omega_{ij} = \begin{cases} \tilde{m}^2; & \text{if } m_i * m_j < \epsilon \tilde{m}^2 \\ =0; & \text{otherwise,} \end{cases} \quad (67)$$

where $\tilde{m}^2 = \sum_{i < j} m_i m_j / (N(N-1)/2)$ is the mean mass product and $\epsilon \sim 10^{-3}$. It may be advisable to integrate Equation (56) for ω even if $\beta = 0$, in order to force the integrator (extrapolation method!) to use short steps if $\dot{\omega}$ is large, thus giving higher precision when required. In fact, numerical experiments suggest that in most practical cases the parameters $(\alpha, \beta, \gamma) = (1, 0, 0)$ give the best results. Exceptions are cases with extremely large mass ratios such that the contribution of the small masses in the potential is negligible (e.g., zero masses included). This point, however, needs further investigation. One potential problem in the integration of the quantity ω is that the increments of it can be arbitrarily large if a collision of point masses occurs. In this case the roundoff errors in the value of ω are significant, but do not affect much the results if the value of β is small. One should also realize that the parameters (α, β, γ) can be changed during the integration (but not during a Bulirsch–Stoer integration step).

4.4. Velocity-Dependent Perturbations

For the case of velocity-dependent perturbations $\mathbf{f} = \mathbf{f}(\mathbf{X}, \mathbf{V})$, which occur, e.g., if one introduces relativistic PN terms, related algorithms were discussed in detail by Mikkola & Merritt (2006) (although mostly for the perturbed two-body problem). Here we present those ideas in the short notation used in Equation (25).

Implicit midpoint method. We have

$$\mathbf{V}' = \tilde{t}'(\mathbf{X})(\mathbf{A}(\mathbf{X}) + \mathbf{f}(\mathbf{X}, \mathbf{V})), \quad (68)$$

which should be solved for constant t , \mathbf{X} . This is often not easy, but it is possible to replace the exact solution by the implicit midpoint method, i.e., one solves (often iteratively) the increment of \mathbf{V} from

$$\Delta \mathbf{V} = \tilde{\delta}t(\mathbf{X})(\mathbf{A}(\mathbf{X}) + \mathbf{f}(\mathbf{X}, \mathbf{V}_0 + \frac{1}{2}\Delta \mathbf{V})), \quad (69)$$

where \mathbf{V}_0 is the value of \mathbf{V} before the update $\mathbf{V} \rightarrow \mathbf{V} + \Delta \mathbf{V}$. This operation thus replaces the one in Equation (63), and the mean velocities (needed in updating \mathbf{B} and ω) are obtained from $\mathbf{V}_0 + \frac{1}{2}\Delta \mathbf{V}$.

Generalized midpoint method. When the generalized midpoint method is used, instead of the leapfrog, one can use a leapfrog step to obtain the increments $\mathbf{d}(\mathbf{X}, \mathbf{V}, s)$ and here one can use for the velocity the most recent available value of \mathbf{V} . One thus simply evaluates

$$\Delta \mathbf{V} = \tilde{\delta}t(\mathbf{X})(\mathbf{A}(\mathbf{X}) + \mathbf{f}(\mathbf{X}, \mathbf{V}_0)). \quad (70)$$

The formulation of the method then guarantees that the final approximation has the correct symmetry (and the correct form of the error expansion) so that the efficient Bulirsch–Stoer extrapolation method can be used.

However, one question to be addressed here is: which of the two above methods is the more efficient? This is problem dependent and numerical experiments may be necessary to answer the question. In the implementation of our code we start with the implicit midpoint method (which is most efficient if the number of iterations remains small) and compute a long time average of the required number of iterations Q recursively as

$$Q_{\text{new}} \rightarrow 0.99Q_{\text{old}} + 0.01Q_{\text{now}}. \quad (71)$$

This slowly “forgets” the old values and does not grow too quickly if there are occasional cases that require many iterations. However, when the average number of iterations exceeds some limit, then there is time to switch to the GAR that does not require iterations (but is otherwise more expensive).

5. NUMERICAL DEMONSTRATIONS

In this section, we demonstrate the performance of the new algorithm and compare it with the celebrated KS-transformed CHAIN code of Mikkola & Aarseth (1993). We first considered a test problem consisting of one massive particle (the “black hole”) with $m = 1$ and seven additional particles (“stars”) with masses $10^{-3} \leq m \leq 10^{-9}$. Relativistic terms were not included. Initial velocities were all zero so the stars moved initially on nearly rectilinear orbits toward the black hole, but their orbits become eccentric ellipses as they experience perturbations from the other stars. We set $(\alpha, \beta, \gamma) = (1, 0, 0)$.

We integrated the initial values given in Table 1 including 2, 3, ..., 8 bodies and carried out integrations up to time 1000 ($G = 1$) using both the old KS-CHAIN code and the new AR-CHAIN. Two sets of experiments were conducted, one in which the Bulirsch–Stoer integrator automatically chose the stepsizes (in a supposedly optimal way) and the second with (iteration to) exact output times of interval $\Delta t = 0.1$.

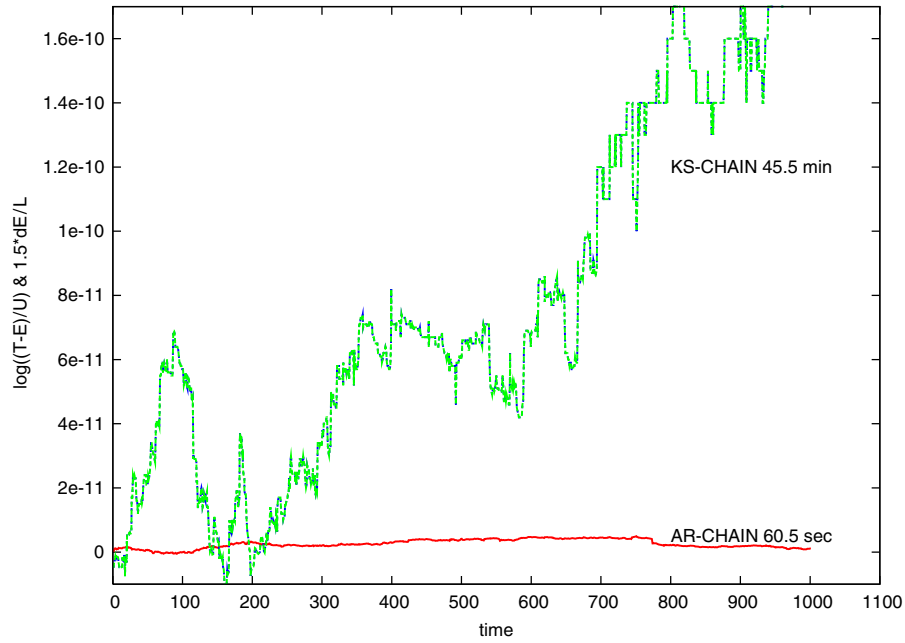


Figure 1. Evolution of $\log((T + B)/U)$ and $1.5 * dE/L$ in the integration of the initial conditions in Table 1. The two lines are for the two different methods: the present method (AR-CHAIN) and the KS-regularized chain method (KS-CHAIN). In both computations, the codes were allowed to automatically choose the stepsize.

Table 1
Initial Conditions for the Integration of Figure 1

m	X	Y	Z	V_x	V_y	V_z
1	0	0	0	0	0	0
1×10^{-3}	-0.432498862	-0.765730892	-0.432498862	0	0	0
1×10^{-4}	0.534279612	0.288862435	0.534279612	0	0	0
1×10^{-5}	-0.383536991	0.601722629	-0.383536991	0	0	0
1×10^{-6}	0.233942789	-0.166401737	0.233942789	0	0	0
1×10^{-7}	0.703086026	0.748854732	0.703086026	0	0	0
1×10^{-8}	0.449061307	0.186538286	0.449061307	0	0	0
1×10^{-9}	0.320791289	-0.848655159	0.320791289	0	0	0

Table 2
Automatic (Free) Stepsize

AR-CHAIN			KS-CHAIN		
NB	s	$\langle dE/U \rangle$	s	$\langle 1.5 * dE/L \rangle$	
2	0.28	9.0×10^{-13}	0.6	4.9×10^{-13}	
3	3.9	1.8×10^{-13}	2.8	4.0×10^{-13}	
4	8.1	2.6×10^{-13}	8.9	4.1×10^{-13}	
5	27.2	5.0×10^{-13}	67.9	5.8×10^{-13}	
6	43.6	5.5×10^{-13}	61.8	1.2×10^{-12}	
7	59.5	1.0×10^{-12}	638.7	2.9×10^{-12}	
8	60.5	3.0×10^{-12}	2730.0	7.8×10^{-11}	

The results are concisely summarized in Tables 2 and 3. There we give for both methods the execution times and average values of the errors. The errors are defined in terms of the Hamiltonians associated with the methods: for AR-CHAIN the average was taken over $|\log((T - E)/U)| \approx \delta E/U$ and for KS-CHAIN $1.5|(T - U - E)/L| \approx 1.5 \delta E/L$. Here $L = T + U$ is the Lagrangian and the factor 1.5 is included because in virial equilibrium $L = 1.5U$. On the other hand the system is chaotic and the solutions are, after some time, often different (as with any method) and so the results must be considered as giving just a general view of the performance.

One sees that the accuracies are comparable, with some advantage in favor of the new method. In particular, we note

Table 3
 $dt = 0.1$ (Iteration to Exact Output Time)

AR-CHAIN			KS-CHAIN		
NB	s	$\langle dE/U \rangle$	s	$\langle 1.5 * dE/L \rangle$	
2	1.3	7.5×10^{-14}	2.2	3.0×10^{-12}	
3	5.9	6.3×10^{-13}	6.1	1.9×10^{-12}	
4	12.4	3.8×10^{-13}	14.0	2.4×10^{-12}	
5	32.9	7.6×10^{-13}	49.6	2.2×10^{-12}	
6	38.8	7.3×10^{-13}	80.5	3.1×10^{-12}	
7	56.6	7.4×10^{-13}	86.8	2.9×10^{-12}	
8	76.5	9.4×10^{-13}	12230.4	1.3×10^{-10}	

that the new method is faster for the case of $N = 2$, i.e. a binary with eccentricity $e = 1$, while the precision is essentially the same. Thus one can conclude that this method works equally well, and perhaps better than the celebrated KS transformation. This is a consequence of the fact that for the two-body system the LogH + leapfrog produces an exact trajectory with only a small error in the time, even for the collision orbit with eccentricity $e = 1$.

The accuracy of the angular momentum is typically similar to that of the energy. This is largely due to the fact that the basic algorithm (leapfrog) used in the method conserves angular momentum exactly.

However, the execution times for cases in which smaller and smaller bodies are included differ considerably in favor of the new AR-CHAIN code. This is not very surprising, since zero masses are a singularity for the KS-CHAIN, but the AR-CHAIN can also handle zero masses.

Figure 1 compares the energy conservation for the two different methods in the eight-particle integration from the initial conditions in Table 1. The system is highly chaotic and the figure shows that in this very difficult case the new method is more accurate and much faster.

Figure 2 shows a second numerical experiment that included all PN terms up to order PN2.5. We considered the prob-

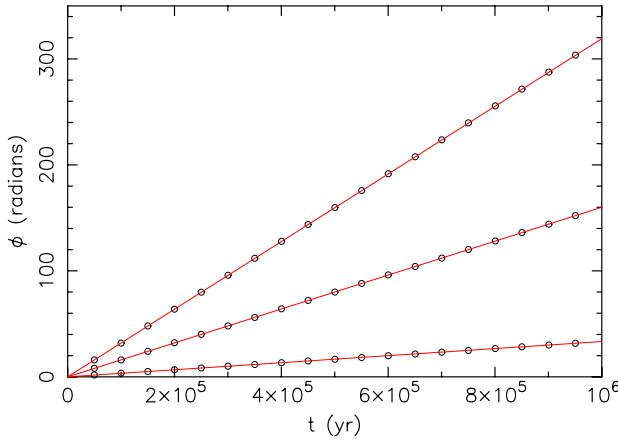


Figure 2. Periastron advancement of a star around the Milky Way SMBH. The semi-major axis is 0.01 pc and three different values of the eccentricity were tried, $e = (0.9, 0.98, 0.99)$. Dots show the major axis orientation every 1000 orbits while the solid (red) lines show the PN2.5 prediction, Equation (72).

lem of periastron shift of a single star orbiting around the SMBH at the center of the Milky Way. The particle masses were $m_1 = 3.5 \times 10^6 M_\odot$ and $m_2 = 10 M_\odot = 2.85 \times 10^{-6} m_1$, and the Keplerian orbit had initial semi-major axis $a = 0.01$ pc, similar to that of the “S” stars (Ghez et al. 2005; Eisenhauer et al. 2005). In units where $G = m_1 = 1$, and adopting $1 \text{ mpc} = 10^{-3} \text{ pc}$ as the length unit, the speed of light is 77.19. Three different eccentricities were tried: (0.9, 0.98, 0.99). The same values of (α, β, γ) were used as in the first experiment. The integrations were continued for 10^6 yr or $\sim 20,000$ orbital periods. The figure plots the orientation of the Laplace–Runge–Lenz vector every 1000 orbits; the solid (red) lines show the expected periastron advance based on the PN2.5 equations, which predict a shift each having a period of

$$\Delta\phi = \frac{6\pi G M_1}{a(1-e^2)c^2} + \frac{3(18+e^2)\pi G^2 M_1^2}{2a^2(1-e^2)^2 c^4}, \quad (72)$$

or (0.095, 0.45, 0.91) degrees for $e = (0.9, 0.98, 0.99)$.

As a third experiment, we investigated the effect of an intermediate-mass black hole (IMBH) on the motion of a star orbiting around the Milky Way SMBH. The IMBH was given a mass of $3500 M_\odot = 10^{-3} M_{\text{SMBH}}$, semi-major axis 0.1 mpc, and eccentricity 0.9. The large eccentricity greatly reduced the gravitational-wave inspiral time, by a factor $\sim 10^3$ compared with a circular orbit; inspiral required ~ 3670 yr or $\sim 10^5$ initial orbital periods. To this binary system was added a third component of mass $10 M_\odot$ on an orbit having $a = 8 \text{ mpc} \approx 1600 \text{ AU}$ and $e = 0.974$ with respect to the center of mass of the SMBH/IMBH binary. These orbital elements are similar to those inferred for the star S0-16 (Ghez et al. 2005). The star and IMBH were coplanar with aligned angular momenta. The star completed $\sim 10^2$ orbits during the time of IMBH inspiral. Figure 3 shows the evolution of the star’s semi-major axis and its orbital orientation during this time, and the configuration-space trajectory is plotted in Figure 4. Initially, the apoastron distance of the IMBH is ~ 0.19 mpc, similar to the periastron distance of the star, ~ 0.21 mpc, so that close interactions are allowed. The primary influence was found to be on the star’s semi-major axis; the eccentricity of the star’s orbit changed only slightly. The star’s periastron advancement was found to be well predicted by Equation (72) if the time dependence of a and e were taken into account (Figure 3).

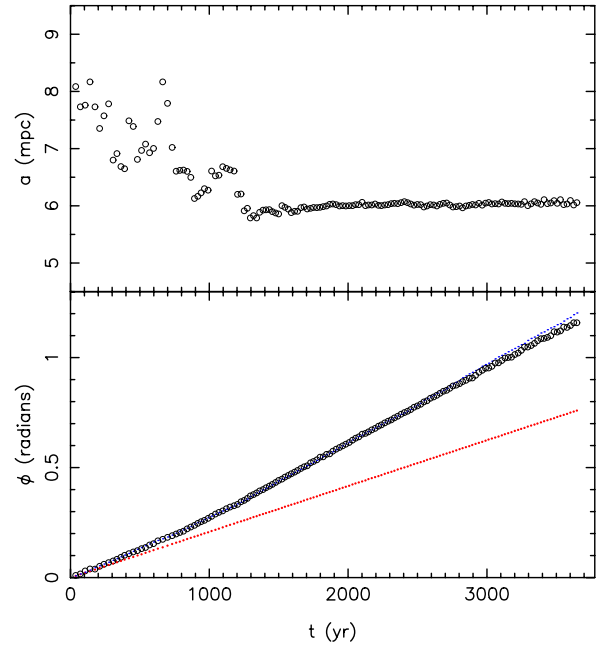


Figure 3. Integration of a star on an eccentric orbit around an IMBH/SMBH binary at the Galactic center. The initial orbit elements are similar to those of the star S0-16 (Ghez et al. 2005). The top panel shows the semi-major axis of the star’s orbit with respect to the massive binary, the lower panel shows the advancement of the periastron. The red (large) filled dots in this panel are the prediction of Equation (72) assuming that a and e remain fixed at their initial values. The blue (small) dots show the predicted advancement if the time dependence of a and e are taken into account. This integration was continued just until the IMBH/SMBH binary had coalesced.

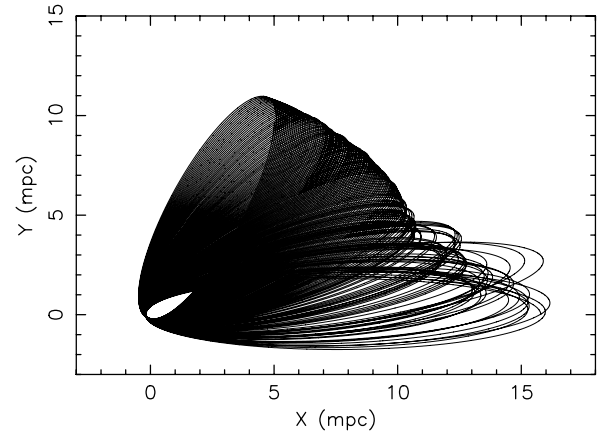


Figure 4. Configuration-space trajectory of the star whose orbital elements are plotted in Figure 3. The orbit remains in the X–Y plane; the semi-major axis is initially parallel to the X-axis. The semi-major axis changes randomly due to perturbations from the IMBH/SMBH binary, but after the latter has shrunk appreciably, a remains nearly fixed and the only evolution is uniform precession due to the PN terms.

A second integration without the PN terms confirmed that the IMBH itself contributes only slightly to the star’s precession rate for this configuration.

Our final numerical experiment was a one-million-year integration of a seven-body system consisting of the MW SMBH, an IMBH of mass $10^{-3} M_{\text{SMBH}}$, and the five, shortest-period S stars: S0-1, S0-2, S0-16, S0-19, and S0-20 (Ghez et al. 2005). Stellar masses were set to $15 M_\odot$ and the initial positions and velocities were determined at year 2000 AD using the Keplerian orbital elements given in Table 3 of Ghez et al. (2005). The IMBH orbit

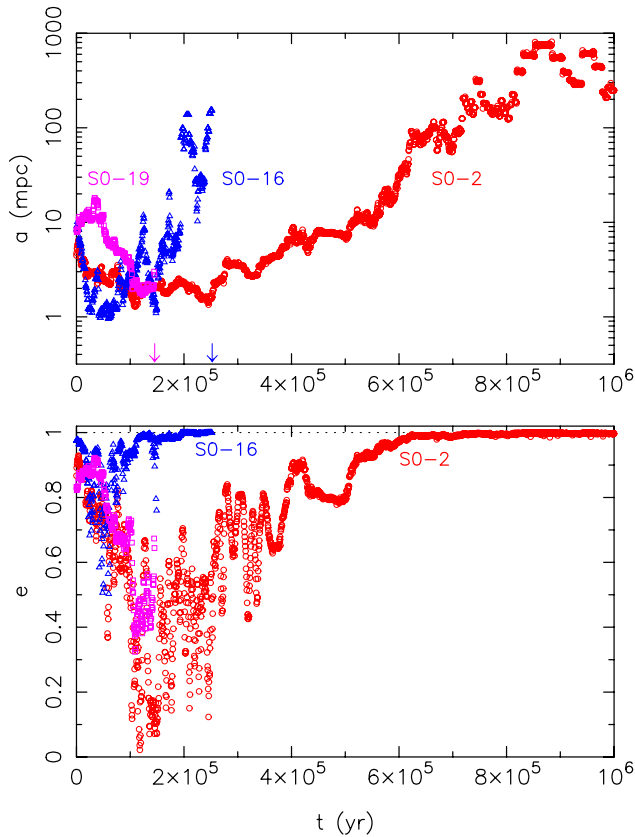


Figure 5. Evolution of the semi-major axes (top) and eccentricities (bottom) of stars S0-2, S0-16, and S0-19 with respect to the Milky Way SMBH in a seven-particle integration that included an IMBH and the five, shortest-period S0 stars. Time zero corresponds to 2000 AD. Arrows in the top panel indicate when S0-19 and S0-16 are ejected; S0-2 remained formally bound to the SMBH/IMBH binary but its semi-major axis gradually increased to ~ 1 pc.

was assigned an initial eccentricity of 0.9 and semi-major axis of 1 mpc, compared with $4 \text{ mpc} \lesssim a \lesssim 25 \text{ mpc}$ for the stars. PN terms were included, causing the orbit of the IMBH to precess rapidly and to fill the annulus $0.1 \text{ mpc} \lesssim r \lesssim 1.9 \text{ mpc}$; for this choice of (a, e, M_{IMBH}) the gravitational-wave inspiral time is $\sim 10^8$ yr, much greater than the length of the integration. Three of the included stars, S0-2, S0-16, and S0-19, have periastron distances that intersect the IMBH's orbital annulus and so each of these stars was able to interact closely with the IMBH, although many orbital periods were required before close encounters occurred. S0-19 was the first to achieve positive energy, at $t \approx 147,500$ AD; before ejection the star's orbit evolved toward small a and e . S0-16 was the next to escape, at $t \approx 254,500$ AD; this star moved into a highly eccentric orbit before being ejected. S0-2 remained bound to the SMBH/IMBH binary but its semi-major axis increased gradually to ~ 1 pc, roughly equal to the radius of influence of the SMBH (if the latter were embedded in the Galactic bulge), so in a practical sense it too may be considered to have escaped. Figure 5 shows the evolution of a and e for the three stars. Experiments like these could be used to constrain the mass and orbital parameters of a putative IMBH near the Galactic center.

6. CONCLUDING REMARKS

We have demonstrated that the LogH method (with leapfrog and Bulirsch–Stoer extrapolation) is at least equally good as, and for some systems better than, the KS-CHAIN method. The optional TTL (which we use only to provide some regularization for the occasional close encounters of very small bodies) seems to have some problems that may have to do with the roundoff error when the function ω first increases then decreases by a large amount in the case of very close encounters. However, we point out that this roundoff problem is reduced almost to non-existence by including in the TTL function Ω only the interactions between small bodies and using only a small factor for this quantity in the time-transformation function. Since the TTL function derivative is explicitly integrated it automatically reduces the stepsize in the case of close approach of small bodies thus providing more careful integration of such events even if the coefficient β is negligible. This is due to the properties (sensitivity) of the Bulirsch–Stoer extrapolation method that sees any tiny irregularity in the data and modifies the stepsize in response. This property of the BS extrapolation method also helps if the leapfrog algorithm happens to evaluate the derivatives too close to the collision of some pair of bodies: the step is rejected and re-evaluated with a different stepsize. This procedure normally avoids the repetition of such a numerical accident. Finally, our code gives the option of not using time transformation at all. Even in this alternative the present code is much better than the straightforward use of center-of-mass coordinates. The reason is that the chain coordinates reduce the roundoff significantly. This is one of the great advantages of the chain structure.

D.M. was supported by grants AST-0420920 and AST-0437519 from the NSF and grant NNX07AH15G from NASA.

REFERENCES

- Aarseth, S. J. 2003, *Gravitational N-Body Simulations* (Cambridge: Cambridge Univ. Press)
- Aarseth, S. J., & Zare, K. 1974, *Celest. Mech.*, **10**, 185
- Bulirsch, R., & Stoer, J. 1966, *Numer. Math.*, **8**, 1
- Eisenhauer, F., et al. 2005, *ApJ*, **628**, 246
- Ghez, A. M., Salim, S., Hornstein, S. D., Tanner, A., Lu, J. R., Morris, M., Becklin, E. E., & Duchêne, G. 2005, *ApJ*, **620**, 744
- Gragg, W. B. 1964, PhD Thesis, Univ. of California, Los Angeles
- Gragg, W. B. 1965, *SIAM J. Numer. Anal.*, **2**, 384
- Heggie, D. C. 1974, *Celest. Mech.*, **10**, 217
- Huang, W., & Leimkuhler, B. 1997, *SIAM J. Sci. Comput.*, **18**, 239
- Kustaanheimo, P., & Stiefel, E. 1965, *J. Reine Angew. Math.*, **218**, 204
- Levi-Civita, T. 1920, *Acta Math.*, **42**, 99
- Mikkola, S., & Aarseth, S. J. 1993, *Celest. Mech. Dyn. Astron.*, **57**, 439
- Mikkola, S., & Aarseth, S. 2002, *Celest. Mech. Dyn. Astron.*, **84**, 343
- Mikkola, S., & Merritt, D. 2006, *MNRAS*, **372**, 219
- Mikkola, S., & Tanikawa, K. 1999a, *Celest. Mech. Dyn. Astron.*, **74**, 287
- Mikkola, S., & Tanikawa, K. 1999b, *MNRAS*, **310**, 745
- Press, W. H., Flannery, B. P., Teukolsky, S. A., & Wetterling, W. T. 1986, *Numerical Recipes* (Cambridge: Cambridge Univ. Press)
- Preto, M., & Tremaine, S. 1999, *AJ*, **118**, 2532
- Soffel, M. H. 1989, *Relativity in Astrometry, Celestial Mechanics and Geodesy* (Berlin: Springer), 141
- Stiefel, E. L., & Scheifele, G. 1971, *Linear and Regular Celestial Mechanics* (Berlin: Springer)
- Yoshida, H. 1990, *Phys. Lett. A*, **150**, 262
- Zare, K. 1974, *Celest. Mech.*, **10**, 207