doi:10.1111/j.1365-2966.2006.10854.x

# Algorithmic regularization with velocity-dependent forces

Seppo Mikkola[1]* and David Merritt[2]*

[1]*Tuorla Observatory, University of Turku, Väisäläntie 20, Piikkiö, Finland*
[2]*Department of Physics, Rochester Institute of Technology, Rochester, NY 14623, USA*

Accepted 2006 July 19. Received 2006 July 18; in original form 2006 May 2

## ABSTRACT

Algorithmic regularization uses a transformation of the equations of motion such that the leapfrog algorithm produces exact trajectories for two-body motion as well as regular results in numerical integration of the motion of strongly interacting few-body systems. That algorithm alone is not sufficiently accurate and one must use the extrapolation method for improved precision. This requires that the basic leapfrog algorithm be time-symmetric, which is not directly possible in the case of velocity-dependent forces, but is usually obtained with the help of the implicit mid-point method. Here, we suggest an alternative explicit algorithmic regularization algorithm which can handle velocity-dependent forces. This is done with the help of a generalized mid-point method to obtain the required time symmetry, thus eliminating the need for the implicit mid-point method and allowing the use of extrapolation.

**Key words:** stellar dynamics – methods: *N*-body simulations – celestial mechanics.

## 1 INTRODUCTION

In some *N*-body problems one has velocity-dependent perturbations. Examples are the relativistic terms, which are important in black hole dynamics (Aarseth 2003), or dissipative terms due to tidal friction or atmospheric friction in satellite orbits. The KS regularization (e.g. basic KS: Kustaanheimo & Stiefel 1965; Stiefel & Scheifele 1971 and the CHAIN-method of Mikkola & Aarseth 1993) can easily handle any additional forces, however in multi-body regularization with the KS transformation, large mass ratios cause problems. Therefore other regularization methods – algorithmic regularizations – such as the logarithmic Hamiltonian method (Mikkola & Tanikawa 1999a,b; Preto & Tremaine 1999) or the time-transformed leapfrog (Mikkola & Aarseth 2002) must be considered. On the other hand, these methods, when combined with the extrapolation method (Gragg 1964, 1965; Bulirsch & Stoer 1966) cannot easily include velocity-dependent forces, except with the help of the implicit mid-point method. Since implicit methods may be inefficient, there is motivation to study ways to make the integrations explicit, while at the same time utilizing the good properties of algorithmic regularization.

Algorithmic regularization is simpler than KS regularization and, what is most important, versions of it work for arbitrary mass ratios. This is especially important in simulations of black hole dynamics in galactic nuclei (Merritt 2006).

In this paper, we first introduce the problem using a perturbed two-body system as an example. Then we suggest a generalized mid-point method to be used as a tool to time-symmetrize any basic algorithm. Finally, the generalization to the *N*-body problem is briefly outlined.

## 2 GENERALIZED ALGORITHMIC REGULARIZATION

Here we discuss the formulation of the basic algorithms, the time-transformed leapfrogs, that are regular in two-body collisions. Then a generalized mid-point method, that can also be used with the Bulirsch–Stoer (BS) extrapolation method (Gragg 1964, 1965; Bulirsch & Stoer 1966), is introduced.

### 2.1 The perturbed two-body problem

We first consider the perturbed two-body problem with velocity-dependent forces. Let $\boldsymbol{r}$ and $\boldsymbol{v}$ be the position and velocity vectors, respectively and $m$ the mass of the two-body system and $t$ the time. Using units in which the gravitational constant $G = 1$ we may write the equation of motion as

$$\dot{\boldsymbol{v}} = -m\frac{\boldsymbol{r}}{r^3} + \boldsymbol{f}(\boldsymbol{r}, t, \boldsymbol{v}), \qquad (1)$$

$$\dot{\boldsymbol{r}} = \boldsymbol{v}. \qquad (2)$$

This case is simple enough for a detailed discussion; generalization to the full *N*-body problem will be straightforward.

As Mikkola & Tanikawa (1999a,b) and Preto & Tremaine (1999) demonstrated, there is a way to make the leapfrog algorithm exact for two-body orbits, and regular for two-body collisions in more complicated problems, if one introduces a time transformation. Here, we concisely re-derive the algorithm and augment it to the case of a general (not necessarily Hamiltonian) perturbation.

*E-mail: Seppo.Mikkola@utu.fi (SM); David.Merritt@rit.edu (DM)

Let

$$b = \frac{m}{r} - \frac{1}{2}\boldsymbol{v}^2 \tag{3}$$

be the binding (Kepler) energy of the two-body system. We have the energy equations

$$\frac{1}{2}\boldsymbol{v}^2 + b = \frac{m}{r}, \tag{4}$$

$$\dot{b} = -\boldsymbol{v} \cdot \boldsymbol{f}. \tag{5}$$

This allows the introduction of the two time transformations

$$\frac{\mathrm{d}t}{\mathrm{d}s} = \frac{1}{\frac{1}{2}\boldsymbol{v}^2 + b}, \tag{6}$$

$$\frac{\mathrm{d}t}{\mathrm{d}s} = \frac{r}{m}, \tag{7}$$

which are equivalent along the solution trajectory. Using the first alternative (6) to transform the equation of motion for the coordinates $(t, \boldsymbol{r})$, one gets

$$t' = \frac{1}{\frac{1}{2}\boldsymbol{v}^2 + b}, \tag{8}$$

$$\boldsymbol{r}' = \frac{\boldsymbol{v}}{\frac{1}{2}\boldsymbol{v}^2 + b}, \tag{9}$$

and the second equation (7) gives for $b$ and $\boldsymbol{v}$

$$b' = -\boldsymbol{v} \cdot \boldsymbol{g}, \tag{10}$$

$$\boldsymbol{v}' = -\frac{\boldsymbol{r}}{r^2} + \boldsymbol{g}, \tag{11}$$

where primes indicate differentiation with respect to the new independent variable $s$ and

$$\boldsymbol{g} = \frac{r}{m}\boldsymbol{f}(\boldsymbol{r}, t, \boldsymbol{v}). \tag{12}$$

If the perturbation $\boldsymbol{f}$ (hence $\boldsymbol{g}$) is independent of the velocity $\boldsymbol{v}$, then the use of (6) for $t$, $\boldsymbol{r}$ and (7) for $b$, $\boldsymbol{v}$ allows the use of the leapfrog algorithm:

$$t_{\frac{1}{2}} = t_0 + \frac{h}{2}\frac{1}{\frac{1}{2}\boldsymbol{v}_0^2 + b_0}, \tag{13}$$

$$\boldsymbol{r}_{\frac{1}{2}} = \boldsymbol{r}_0 + \frac{h}{2}\frac{\boldsymbol{v}_0}{\frac{1}{2}\boldsymbol{v}_0^2 + b_0}, \tag{14}$$

$$\boldsymbol{v}_1 = \boldsymbol{v}_0 - h\frac{\boldsymbol{r}_{\frac{1}{2}}}{r_{\frac{1}{2}}^2} + h\boldsymbol{g}_{\frac{1}{2}}, \tag{15}$$

$$b_1 = b_0 - h\boldsymbol{v}_{\frac{1}{2}} \cdot \boldsymbol{g}_{\frac{1}{2}}, \tag{16}$$

$$t_1 = t_{\frac{1}{2}} + \frac{h}{2}\frac{1}{\frac{1}{2}\boldsymbol{v}_1^2 + b}, \tag{17}$$

$$\boldsymbol{r}_1 = \boldsymbol{r}_{\frac{1}{2}} + \frac{h}{2}\frac{\boldsymbol{v}_1}{\frac{1}{2}\boldsymbol{v}_1^2 + b_1}, \tag{18}$$

where the subscripts 0 and 1 refer to the beginning and the end of the step, and $\boldsymbol{v}_{\frac{1}{2}} = (\boldsymbol{v}_0 + \boldsymbol{v}_1)/2$, which gives the correct solution for the pair of equations (10) and (11) when solved for $b$ and $\boldsymbol{v}$ with constant $t$ and $\boldsymbol{r}$. More generally, it simply gives the right kind of symmetry for the approximation. If the perturbation $\boldsymbol{g} = \boldsymbol{0}$, then the motion is pure Kepler motion and the leapfrog algorithm produces an exact trajectory with only a time error (Mikkola & Tanikawa 1999a,b; Preto & Tremaine 1999).

In the above equations, the symbol $\boldsymbol{g}_{\frac{1}{2}}$ indicates $\boldsymbol{g}(\boldsymbol{r}_{\frac{1}{2}})$. However, if $\boldsymbol{g}$ actually depends on the velocity too, then the leapfrog cannot be immediately formed. This problem (or rather an analogous one) was solved by Mikkola & Aarseth (2002), using the implicit mid-point method, i.e. it was necessary to solve the equation

$$\boldsymbol{v}_1 = \boldsymbol{v}_0 - h\frac{\boldsymbol{r}_{\frac{1}{2}}}{r_{\frac{1}{2}}^2} + h\boldsymbol{g}\left(\boldsymbol{r}_{\frac{1}{2}}, t_{\frac{1}{2}}, \frac{\boldsymbol{v}_0 + \boldsymbol{v}_1}{2}\right) \tag{19}$$

for $\boldsymbol{v}_1$. Often this solution is possible only by iteration which can be rather expensive if the perturbation is strong and complicated. This fact motivates a search for ways to find an alternative that is explicit, yet capable of utilizing the algorithmic regularization. This goal can be achieved with the help of the algorithm we next discuss.

### 2.2 Generalized mid-point method

Here, we introduce a generalization to the well-known modified mid-point method. In this algorithm, the basic approximation to advance the solution is not just the evaluation of the derivative at the mid-points, but any method to approximate the solution. Thus, the algorithmic regularization by the leapfrog can be used even when the additional force depends on velocities. That provides a regular basic algorithm, which is made suitable for the extrapolation method by means of the generalized mid-point method, as follows.

Our algorithm is based on the realization that any general initial value problem

$$\dot{z} = \boldsymbol{f}(z) \;\; z(0) = z_0. \tag{20}$$

can be split into two as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{y}), \tag{21}$$

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{x}) \tag{22}$$

with the initial values

$$\boldsymbol{x}_0 = \boldsymbol{y}_0 = z(0).$$

It is simple to see that this pair has the solution

$$\boldsymbol{x}(t) = \boldsymbol{y}(t) = z(t).$$

On the other hand, it is now possible to construct the leapfrog-like algorithm

$$\boldsymbol{x}_{\frac{1}{2}} = \boldsymbol{x}_0 + \frac{h}{2}\boldsymbol{f}(\boldsymbol{y}_0), \tag{23}$$

$$\boldsymbol{y}_1 = \boldsymbol{y}_0 + hf(\boldsymbol{x}_{\frac{1}{2}}), \tag{24}$$

$$\boldsymbol{x}_1 = \boldsymbol{x}_{\frac{1}{2}} + \frac{h}{2}\boldsymbol{f}(\boldsymbol{y}_1), \tag{25}$$

to obtain approximations for the solution of (20). In fact, this is nothing but another way to write the well-known modified mid-point method.

A new interpretation of the above can be obtained by first rewriting it in the form

$$\boldsymbol{x}_{\frac{1}{2}} = \boldsymbol{x}_0 + \left(+\frac{h}{2}\boldsymbol{f}(\boldsymbol{y}_0)\right), \tag{26}$$

$$\boldsymbol{y}_{\frac{1}{2}} = \boldsymbol{y}_0 - \left(-\frac{h}{2}f\left(\boldsymbol{x}_{\frac{1}{2}}\right)\right), \tag{27}$$

$$\boldsymbol{y}_1 = \boldsymbol{y}_{\frac{1}{2}} + \left(+\frac{h}{2} f\left(\boldsymbol{x}_{\frac{1}{2}}\right)\right), \tag{28}$$

$$\boldsymbol{x}_1 = \boldsymbol{x}_{\frac{1}{2}} - \left(-\frac{h}{2} \boldsymbol{f}(\boldsymbol{y}_1)\right). \tag{29}$$

In equation (26), the bracketed term is an (Euler-method) approximation to the increment of $\boldsymbol{x}$ over the time-interval $h/2$ with the initial value $\boldsymbol{y}_0$, while in (27) the initial value is $\boldsymbol{x}_{\frac{1}{2}} \approx \boldsymbol{x}(h/2)$ and the time-interval is $-\ h/2$ Finally, this increment is added – with a minus sign – to $\boldsymbol{y}_0$ to obtain an approximation for $\boldsymbol{y}\ (h/2)$. In the remaining formulae (28) and (29), the idea is the same but the roles of $\boldsymbol{x}$ and $\boldsymbol{y}$ have been changed.

A generalization of this is now obvious. Let

$$z(\Delta t) \approx z_0 + \boldsymbol{d}(z_0, \Delta t) \tag{30}$$

be an approximation to the solution of equation (20) over a time-interval $\Delta t$. In Euler's method,

$$\mathrm{d}(z_0, \Delta t) = \Delta t\, \boldsymbol{f}(z_0), \tag{31}$$

which gives the algorithm described in equations (26)–(29), but in general, $\boldsymbol{d}$ could be obtained from any reasonable method for solving the differential equation (20). We thus choose a method and define

$$\boldsymbol{d}(z_0, \Delta t) = \widetilde{z}(\Delta t) - z_0, \tag{32}$$

where $\widetilde{z}(\Delta t)$ is the approximation for $z(\Delta t)$ obtained with the chosen method. This generalized mid-point algorithm may be especially useful if one uses a special method that is well-suited to the particular problem at hand. For the perturbed two-body problem with velocity-dependent perturbations such a method is the algorithmic regularization leapfrog discussed in the previous section.

One step in the generalized mid-point method can now be written

$$\boldsymbol{x}_{\frac{1}{2}} = \boldsymbol{x}_0 + \boldsymbol{d}\left(\boldsymbol{y}_0, +\frac{h}{2}\right), \tag{33}$$

$$\boldsymbol{y}_{\frac{1}{2}} = \boldsymbol{y}_0 - \boldsymbol{d}\left(\boldsymbol{x}_{\frac{1}{2}}, -\frac{h}{2}\right), \tag{34}$$

$$\boldsymbol{y}_1 = \boldsymbol{y}_{\frac{1}{2}} + \boldsymbol{d}\left(\boldsymbol{x}_{\frac{1}{2}}, +\frac{h}{2}\right), \tag{35}$$

$$\boldsymbol{x}_1 = \boldsymbol{x}_{\frac{1}{2}} - \boldsymbol{d}\left(\boldsymbol{y}_1, -\frac{h}{2}\right), \tag{36}$$

or, if we define the mapping (or 'subroutine')

$$\mathbf{A}(\boldsymbol{x}, \boldsymbol{y}, h): \quad \boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{d}\left(\boldsymbol{y}, +\frac{h}{2}\right) \tag{37}$$

$$\boldsymbol{y} \to \boldsymbol{y} - \boldsymbol{d}\left(\boldsymbol{x}, -\frac{h}{2}\right), \tag{38}$$

we can write the algorithm with many ($N$) steps as

1. Set $\boldsymbol{y} = \boldsymbol{x}$.
2. Repeat $\mathbf{A}(\boldsymbol{x}, \boldsymbol{y}, h)\mathbf{A}(\boldsymbol{y}, \boldsymbol{x}, h)$ $N$ times. (39)
3. Accept $\boldsymbol{x}$ as the final result.

Thus, one simply calls the subroutine $\mathbf{A}$ alternately with arguments $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{y}, \boldsymbol{x})$ such that the sequence is time-symmetric (starts and stops with $\boldsymbol{x}$ in equation 39).

This basic algorithm has the correct symmetry – because it was derived from a leapfrog-like treatment – such that the error in integration over a fixed time-interval with different time-steps $h$ can be written

$$\text{error} = A_1 h^2 + A_4 h^4 + \,.\,., \tag{40}$$

and thus the Gragg–Bulirsch–Stoer extrapolation method can be used to obtain high accuracy.

When the basic method to obtain the increments $\boldsymbol{d}$ is the leapfrog (equations 13–18), the algorithm (39) involves more work than the conventional leapfrog and thus it cannot be recommended in cases where it is not necessary. However, the great advantage of this generalized mid-point method is that the leapfrog with the implicit mid-point method can be replaced by a method that is not exactly time-symmetric. Consequently even velocity-dependent forces can be treated with an explicit algorithm. The computation of the quantity $\boldsymbol{g}_{\frac{1}{2}}$, when it depends on velocity, can be done in a straightforward way, e.g. by

$$\boldsymbol{g}_{\frac{1}{2}} = \boldsymbol{g}\left(\boldsymbol{r}_{\frac{1}{2}}, t_{\frac{1}{2}}, \boldsymbol{v}_{\frac{1}{2}}\right), \tag{41}$$

where one may approximate $\boldsymbol{v}_{\frac{1}{2}}$ either by $\boldsymbol{v}_{\frac{1}{2}} \approx \boldsymbol{v}_0$ or preferably by

$$\boldsymbol{v}_{\frac{1}{2}} \approx \boldsymbol{v}_0 - \frac{h}{2}\frac{\boldsymbol{r}_{\frac{1}{2}}}{r^2_{\frac{1}{2}}} \tag{42}$$

after which

$$\boldsymbol{v}_1 = \boldsymbol{v}_0 - h\frac{\boldsymbol{r}_{\frac{1}{2}}}{r^2_{\frac{1}{2}}} + h\boldsymbol{g}\left(\boldsymbol{r}_{\frac{1}{2}}, t_{\frac{1}{2}}, v_{\frac{1}{2}}\right) \tag{43}$$

can be used instead of (15) (or 19). Here, it is necessary to stress that only the increments of the variables from the algorithm (13)–(18) are to be used as the quantities $\boldsymbol{d}$ in the algorithm (37)–(38).

## 3 SOME EXPERIMENTS

Using a simple perturbed two-body code, written according to the above theory, we carried out some experiments to compare the new alternative with the implicit mid-point method.

Tests with an (initially) circular orbit of unit radius and with the perturbing (frictional) force $\boldsymbol{f} = -\epsilon \boldsymbol{v}$ suggest that for very small $\epsilon$ ($\leqslant 10^{-6}$) the implicit mid-point method is faster, but for stronger perturbations, the new method is favourable.

Tests with the relativistic PPN2.5 terms from Soffel (1989) are illustrated in Fig. 1. Here the system was a two-body system with masses $m_1 = 0.9, m_2 = 0.1$, initial semimajor-axis $a_0 = 1$, initial eccentricity $e_0 = 0$ and the velocity of light was set to $c = 20$. Due to the gravitational radiation term, the semimajor axis shrinks and the computational effort ($\mathrm{d}N/\mathrm{d}t$ = number of perturbation evaluations per unit of time) increases. The figure illustrates the evolution of $\mathrm{d}N/\mathrm{d}t$ (averages over 100 steps with BS extrapolation) during the computation (until final merger of the bodies) for three different methods. The results are plotted as a function of the shrinking semimajor axis (measured in terms of the Schwarzschild radius for the combined mass). In these integrations the one-step relative error tolerance was set to $10^{-13}$ and the errors, measured via the quantity $\frac{r}{m}(\frac{1}{2}\boldsymbol{v}^2 + b) - 1$, were $\sim 10^{-11}$ for the new method and the mid-point method with iteration to convergence (corresponding to the '△' and '+' experiments in Fig. 1). For the restricted iteration method the error was, however, about $10^{-9}$ suggesting that this method is not to be recommended. The errors grew secularly, as can be seen from Fig. 2; the numbers given above refer to the values just at merger, i.e. when the two particles approach more closely than the sum of

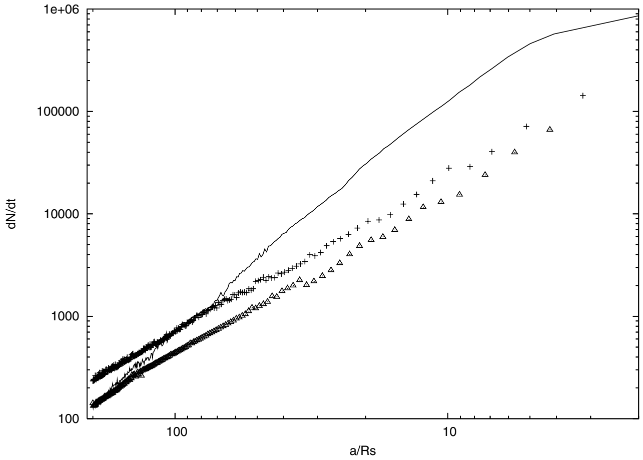

**Figure 1.** Number of perturbation evaluations per unit of time ($\mathrm{d}N/\mathrm{d}t$) in a two-black-hole system, with masses and speed of light $m_1 = 0.9, m_2 = 0.1, c = 20$, integrated from the initial values $a_0 = 1, e_0 = 0$ until the final merger of the two black holes. The $x$-axis is the semimajor axis in units of the Schwarzschild radius. The symbols $\triangle$ are for the new method while $+$ and the solid line illustrate two varieties of the implicit mid-point method (as described in the text).

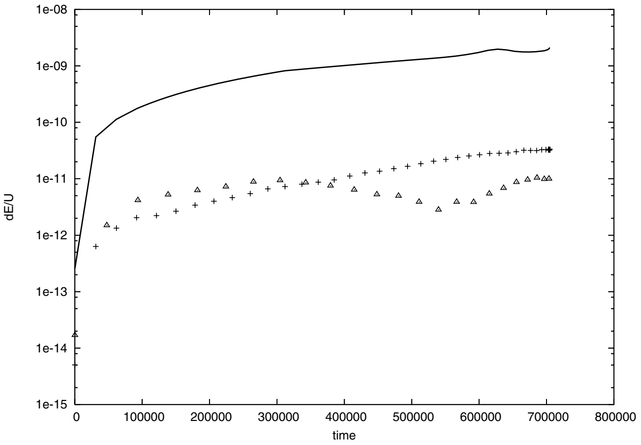

**Figure 2.** Error evolution in the experiments. The measure of error, plotted as a function of time, is $|(\frac{1}{2}\boldsymbol{v}^2 + b)\frac{r}{m} - 1|$. Symbols are the same as in the previous figure.

Schwarzschild radii. It may be seen that in all cases, the new method is somewhat more efficient.

## 4 $N$-BODY FORMULATION

The generalization of the algorithm to the $N$-body problem is simple in principle. One may use the leapfrog algorithms introduced by Mikkola & Tanikawa (1999a,b) or Mikkola & Aarseth (2002) and simply add the necessary velocity-dependent forces. A new formulation that effectively unifies the above cited works may be constructed as follows. Let $T = (1/2)\sum_k m_k \boldsymbol{v}_k^2$ be the kinetic energy, $U = \sum_{i<j} m_i m_j |\boldsymbol{r}_i - \boldsymbol{r}_j|^{-1}$ be the potential energy, and $\Omega$ an (in principle) arbitrary function of of the coordinates, often

$$\Omega = \sum_{i<j} |\boldsymbol{r}_i - \boldsymbol{r}_j|^{-1}. \tag{44}$$

Then one may define, in analogy with (6) and (7), the two time transformations

$$t' = \frac{1}{\alpha T + B} = \frac{1}{\alpha U + \beta\Omega + \gamma}, \tag{45}$$

where $\alpha, \beta$ and $\gamma$ are adjustable constants. Since $T = U + E$, we have $B = -\alpha E + \beta\Omega + \gamma$, which expression is used only for the initial value of $B$ and later this quantity must be obtained by solving the differential equation

$$\dot{B} = -\alpha \sum_k \boldsymbol{v}_k \cdot \boldsymbol{f}_k + \beta \sum_k \frac{\partial\Omega}{\partial \boldsymbol{r}_k} \cdot \boldsymbol{v}_k. \tag{46}$$

In the above, $\boldsymbol{v}_k, \boldsymbol{r}_k$ are the velocity and position of the body with mass $m_k$, correspondingly, and the forces additional to $\partial U/\partial \boldsymbol{r}_k$ are denoted by $\boldsymbol{f}_k$.

The equations of motion that can be used to construct the leapfrog that provides algorithmic regularization are, for time and coordinates, respectively,

$$t' = \frac{1}{\alpha T + B}, \tag{47}$$

$$\boldsymbol{r}'_k = t' \boldsymbol{v}_k \tag{48}$$

and for velocities and $B$

$$\tau' = \frac{1}{\alpha U + \beta\Omega + \gamma}, \tag{49}$$

$$\boldsymbol{v}'_k = \frac{\tau'[(\partial U/\partial \boldsymbol{r}_k) + \boldsymbol{f}_k]}{m_k}, \tag{50}$$

$$B' = \tau' \sum_k \left(-\alpha \boldsymbol{f}_k + \beta \frac{\partial\Omega}{\partial \boldsymbol{r}_k}\right) \cdot \boldsymbol{v}_k. \tag{51}$$

Here, the (possible) velocity dependence of the additional forces $\boldsymbol{f}_k$ can be handled as in our two-body example above. However, to account for the (explicitly written) $\boldsymbol{v}$-dependence of $B'$ one must follow Mikkola & Aarseth (2002), i.e. first the $\boldsymbol{v}_k$ are advanced and then the average $[\boldsymbol{v}_k(0) + \boldsymbol{v}_k(h)]/2$ is used to evaluate $B'$. Thus, the leapfrog can be constructed in obvious analogy with the perturbed two-body case. However, in $N$-body integrations, the roundoff can be a serious source of error and relative coordinates of close bodies must be used to reduce that effect (Mikkola & Tanikawa 1999a; Mikkola & Aarseth 2002).

Some additional remarks follow.

(i) If one takes $(\alpha, \beta, \gamma) = (1, 0, 0)$ then the method obtained is the logarithmic Hamiltonian method (Mikkola & Tanikawa 1999a).

(ii) If $(\alpha, \beta, \gamma) = (0, 1, 0)$ then we have the time transformed leapfrog (TTL) (Mikkola & Aarseth 2002).

(iii) If $(\alpha, \beta, \gamma) = (0, 0, 1)$ then the method is just the normal basic leapfrog.

(iv) If there are no velocity-dependent perturbations, then the normal leapfrog can be used and it is in fact faster. This is because our alternative algorithm then does some (unnecessary) calculations back and forth.

(v) The question of which combination of the numbers $(\alpha, \beta, \gamma)$ is best cannot be answered in general, but experimentation is necessary. For $N$-body systems with very large mass ratios, however, it seems that one must have $\beta \neq 0$, which means a form of the TTL method. This is because low-mass bodies do not contribute significantly to the energies (kinetic and/or potential) and consequently, if $\beta = 0$, there is no significant reduction in stepsize during a close encounter.

(vi) The experiments discussed in Section 3 correspond to the alternative (i), i.e. $(\alpha, \beta, \gamma) = (1, 0, 0)$. Note that for the case of only two bodies, there should be not much difference between alternatives (i) and (ii) since in this case they are mathematically equivalent (Mikkola & Aarseth 2002).

## 5 CONCLUSION

We have demonstrated that the generalized mid-point algorithm can be used to time-symmetrize the algorithmic regularization leapfrog even when the forces depend on velocities. This permits efficient use of the extrapolation method. For very small perturbations, the implicit mid-point method may still be better, and the new method can be recommended only when the velocity dependence of the forces is significant. Finally, we note that the generalized mid-point method can be used with any special low-order approximation to the differential equations under consideration. Thus, it is not restricted to *N*-body problems.

## ACKNOWLEDGMENTS

DM was supported by grants AST-0420920 and AST-0437519 from the NSF and grant NNG04CJ48G from NASA.

## REFERENCES

Aarseth S. J., 2003, Ap&SS, 285, 367

Bulirsch R., Stoer J., 1966, Num. Math., 8, 1

Gragg W. B., 1964, PhD thesis, Univ. California, Los Angeles

Gragg W. B., 1965, SIAM J. Numer. Anal., 2, 384

Kustaanheimo P., Stiefel E., 1965, J. Reine Angew. Math., 218, 204

Merritt D., 2006, Rept. Prog. Phys., 69, 2513

Mikkola S., Aarseth S. J., 1993, Cel. Mech. Dyn. Astron., 57, 439

Mikkola S., Aarseth S., 2002, Cel. Mech. Dyn. Astron., 84, 343

Mikkola S., Tanikawa K., 1999a, MNRAS, 310, 745

Mikkola S., Tanikawa K., 1999b, Cel. Mech. Dyn. Astron., 74, 287

Preto M., Tremaine S., 1999, AJ, 118, 2532

Soffel M. H., 1989, Relativity in Astrometry, Celestial Mechanics and Geodesy. Springer, Berlin, p. 141

Stiefel E. L., Scheifele G., 1971, Linear and Regular Celestial Mechanics. Springer, Berlin

This paper has been typeset from a TEX/LATEX file prepared by the author.