JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

ELSEVIER Journal of Computational and Applied Mathematics 67 (1996) 173–179

Letter Section

Are Gauss–Legendre methods useful in molecular dynamics?

M.A. López-Marcos[a],*, J.M. Sanz-Serna[a], J.C. Díaz[b]

[a] *Departamento de Matemática Aplicada y Computación, Universidad de Valladolid, Valladolid, Spain*
[b] *Center for Parallel and Scientific Computing, The University of Tulsa, Tulsa, OK 74104-3189, USA*

Received 22 February 1995

Abstract

We apply the two-stage Gauss–Legendre method to the numerical simulation of liquid argon, a typical problem in molecular dynamics. It is found that the scheme is less efficient than the Verlet/leapfrog method, standard in this sort of simulation.

Keywords: Gauss–Legendre methods; Molecular dynamics; Symplectic integration; Parallel integrators

1. Introduction

In this note we study the advisability of using (implicit) Gauss–Legendre Runge–Kutta–Nyström (RKN) methods in the integration of problems of the form

$$\frac{d^2\boldsymbol{q}}{dt} = \boldsymbol{g}(\boldsymbol{q}), \quad \boldsymbol{q} \in \mathscr{R}^D \tag{1}$$

arising in molecular dynamics [1]. For a simulation involving N particles, $D = 3N$ and $\boldsymbol{q}$ consists of N three-dimensional blocks; the ith block contains the three Cartesian coordinates x_i, y_i, z_i of the ith particle. Furthermore, $\boldsymbol{g} = M^{-1}\boldsymbol{f}$, where $\boldsymbol{f}$ is a block vector whose ith block provides the force on the ith particle and M is a diagonal matrix of masses (the mass m_i of the ith particle appears in the $(3i-2)$th, $(3i-1)$th and $3i$th diagonal entries). Thus, (1) represents Newton's second law for the motion of the particles. Alternatively, (1) may be written as first-order system after introducing the velocity vector $\boldsymbol{v}$:

$$\frac{d\boldsymbol{q}}{dt} = \boldsymbol{v}, \qquad \frac{d\boldsymbol{v}}{dt} = \boldsymbol{g}(\boldsymbol{q}). \tag{2}$$

* Corresponding author. E-mail: lopezmar@cpd.uva.es.

In actual molecular dynamics situations, the integration of (1) or (2) is a very expensive task. This is due to the large value of N, to the complexity of the evaluation of each component of the force vector $\boldsymbol{f}$ (cf. Section 3) and to the long time-intervals of interest. Many molecular dynamics simulations are performed by the simple-minded Verlet algorithm [1] and it is of clear interest to investigate whether more sophisticated integrators may provide a better alternative. The Verlet algorithm is symplectic [10] and there is a growing evidence [8] that symplecticness is an attractive feature in molecular dynamics algorithms. For a detailed discussion of symplectic integration see [10]. Okunbor and Skeel have successfully applied a fourth-order symplectic RKN method suggested by Calvo and Sanz-Serna [2] to the simulation of liquid argon. For the same problem, Janežič and Orel [6] have tested the two-stage, fourth-order (implicit) Gauss–Legendre method. The Gauss–Legendre formulae were shown to be symplectic by one of the present authors in [9]; an alternative proof may be seen in [3]. In view of the need for iteratively solving nonlinear equations, implicit formulae may seem of little appear in the numerical simulation of non-stiff molecular dynamics problems. However, when the nonlinear equations are solved by functional iteration in a parallel environment, it is possible to compute in parallel the various stages and this renders implicit formulae potentially interesting. Indeed, van der Howen and his coworkers, see, e.g., [5, 12] have suggested functional iteration in the Gauss–Legendre formulae as a useful way of obtaining parallel integrators. In this connection the Gauss–Legendre formulae have the advantage of their optimally high-order relative to the number of stages. It should be emphasized that explicit, symplectic RKN formulae are inherently sequential [7], [10, Section 8.5] and therefore not specially suited for parallel environments.

In this note we report on our experience with the two-stage Gauss–Legendre RKN method for the liquid argon problem. Our implementation is different from that in [6] and by presenting our result in the form of efficiency plots we are able to reach more definite conclusions than in [6]: it turns out that the Gauss–Legendre method are not useful for this test problem that only involves van der Waals forces. It is in principle possible that in problems involving bond forces the Gauss methods are of some interest.

Setion 2 contains a brief discription of the methods tested and Section 3 presents the problem to be solved. The final Section 4 includes the numerical results and our conclusions.

2. Numerical methods

If we denote by $\boldsymbol{q}^n$ and $\boldsymbol{v}^n$ the numerical solution of (2) at time $t^n = n\Delta t$, then a step $(\boldsymbol{q}^n, \boldsymbol{v}^n) \mapsto (\boldsymbol{q}^{n+1}, \boldsymbol{v}^{n+1})$ of (the position form of) Verlet's method is given by the three substeps

$$\boldsymbol{q}^{n+1/2} = \boldsymbol{q}^n + \tfrac{1}{2}\Delta t\, \boldsymbol{v}^n, \tag{3}$$

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \Delta t \boldsymbol{g}(\boldsymbol{q}^{n+1/2}), \tag{4}$$

$$\boldsymbol{q}^{n+1} = \boldsymbol{q}^{n+1/2} + \tfrac{1}{2}\Delta t\, \boldsymbol{v}^{n+1}. \tag{5}$$

If the value $\boldsymbol{q}^{n+1}$ at a particular grid point t^n is not required for output, then it is advantageous to merge the last substep of the current step with the first substep of the next. This yields

$$\boldsymbol{q}^{n+3/2} = \boldsymbol{q}^{n+1/2} + \Delta t\,\boldsymbol{v}^{n+1},$$

a formula that, along with (4), gives the leapfrog version of the method. This involves approximating $\boldsymbol{v}$ at the grid points t^n and $\boldsymbol{q}$ halfway between gridpoints. We implement the leapfrog version and therefore, along the integration, we only perform the substep (3) (respectively (5)) if t^n was an output point (respectively if t^{n+1} is going to be an output point).

The two-stage Gauss–Legendre RKN method for (1) is the result of applying to (2) the standard two-stage Gauss–Legendre Runge–Kutta formula and eliminating the stage vectors for the velocity $\boldsymbol{v}$, see, e.g., [10, Section 3.5]. When functional iteration is used to solve the implicit equations for the stage vectors, a step of the metod is as follows. Given initial approximations $\boldsymbol{Q}_1^{[0]}, \boldsymbol{Q}_2^{[0]}$ to the stage vectors, find, for $i = 0, 1, \dots, \nu - 1$,

$$\boldsymbol{Q}_1^{[i+1]} = \boldsymbol{q}^n + \Delta t \gamma_1 \boldsymbol{v}^n + \Delta t^2 \alpha_{11} \boldsymbol{g}(\boldsymbol{Q}_1^{[i]}) + \Delta t^2 \alpha_{12} \boldsymbol{g}(\boldsymbol{Q}_2^{[i]}),$$

$$\boldsymbol{Q}_2^{[i+1]} = \boldsymbol{q}^n + \Delta t \gamma_2 \boldsymbol{v}^n + \Delta t^2 \alpha_{21} \boldsymbol{g}(\boldsymbol{Q}_1^{[i]}) + \Delta t^2 \alpha_{22} \boldsymbol{g}(\boldsymbol{Q}_2^{[i]}),$$

where [10, Example 3.3] $\gamma_1 = (3 - \sqrt{3})/6$, $\gamma_2 = (3 + \sqrt{3})/6$, $\alpha_{11} = \alpha_{22} = 1/24$, $\alpha_{12} = (3 - 2\sqrt{3})/24$, $\alpha_{21} = (3 + 2\sqrt{3})/24$. After this,

$$\boldsymbol{q}^{n+1} = \boldsymbol{q}^n + \Delta t\,\boldsymbol{v}^n + \Delta t^2 \beta_1 \boldsymbol{g}(\boldsymbol{Q}_1^{[\nu]}) + \Delta t^2 \beta_2 \boldsymbol{g}(\boldsymbol{Q}_2^{[\nu]}),$$

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^n + \tfrac{1}{2} \Delta t \left(\boldsymbol{g}(\boldsymbol{Q}_1^{[\nu]}) + \boldsymbol{g}(\boldsymbol{Q}_2^{[\nu]})\right),$$

where $\beta_1 = (3 + \sqrt{3})/12$ and $\beta_2 = (3 - \sqrt{3})/12$.

In all the experiments to be reported later, the initial guesses $\boldsymbol{Q}_1^{[0]}, \boldsymbol{Q}_2^{[0]}$ were obtained by evaluating at $t^n + \Delta t \gamma_1$ and $t^n + \Delta t \gamma_2$ the collocation polynomial for the preceding $t^{n-1} \to t^n$ step, see, e.g., [10, Section 3.3.2]. This way of initializing the stage is recommended in [4, p. 133]. In [6] an alternative initialization is employed. We also carried out experiments with the choice $\boldsymbol{Q}_1^{[0]} = \boldsymbol{Q}_2^{[0]} = \boldsymbol{q}^n$; but this alternative degraded the performance of the algorithm.

In the experiments in Section 4, the number ν of iterations per step was prescribed in advance. This, in tandem with our choice of initial guesses, effectively turns the formula into a two-step, explicit one, see [12] for a discussion. It is readily found that, for all values of $\nu \geqslant 1$, the implied explicit formula is consistent of order four. Per step, the number of evaluations of $\boldsymbol{g}$ is $2(\nu + 1)$. However, if two processors are available, say by linking two workstations with PVM, the evaluations at $\boldsymbol{Q}_1^{[i]}$ and $\boldsymbol{Q}_2^{[i]}$ may be carried out in parallel and the cost is effectively of only $\nu + 1$ evaluations per step. Note that with $\nu = 1$ this means order four at only two evaluations per step. On the other hand, note that using a fixed number of iterations per step destroys the symplecticness of the Gauss–Legendre formula; this is an example of what Skeel [11] calls a "symplectic crime". It is also possible to iterate to convergence, i.e., to iterate until two consecutive approximations $(\boldsymbol{Q}_1^{[i]}, \boldsymbol{Q}_2^{[i]})$ and $(\boldsymbol{Q}_1^{[i+1]}, \boldsymbol{Q}_2^{[i+1]})$ are found that differ in less than a small prescribed tolerance. An iteration-to-convergence algorithm was also implemented; its efficiency was found not to differ noticeably from that of the fixed-ν algorithm. With stringent (resp. lax) tolerance the performance was similar to that obtained by giving to ν a fixed, large (resp. small) value.

3. The problem

Consider N particles with masses m_i, positions $\boldsymbol{r}_i = (x_i, y_i, z_i)$ and velocities $\boldsymbol{v}_i = \mathrm{d}\boldsymbol{r}_i/\mathrm{d}t$ attracting each other with a potential $V(r)$. The force $\boldsymbol{f}_{ij}$ exerted on the ith particle by the jth particle is then

$$\boldsymbol{f}_{ij} = -\nabla \boldsymbol{r}_i V(\|\boldsymbol{r}_i - \boldsymbol{r}_j\|) = V'(\|\boldsymbol{r}_i - \boldsymbol{r}_j\|)\frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|}$$

and the equations of motion are

$$m_i \frac{\mathrm{d}^2 \boldsymbol{r}_i}{\mathrm{d}t^2} = \sum_{j \neq i} \boldsymbol{f}_{ij}. \tag{6}$$

The total energy H, obtained by summing the kinetic and potential contributions,

$$H = K + V = \sum_i \tfrac{1}{2} m_i \|\boldsymbol{v}_i\|^2 + \sum_{\substack{i,j \\ i \neq j}} V(\|\boldsymbol{r}_i - \boldsymbol{r}_j\|)$$

is a conserved quantity for the system (6). The temperature T is a measure of the average kinetic energy per particle and is given by

$$\tfrac{3}{2}(N-1)kT = K; \tag{7}$$

k is the Boltzmann constant, 1.380658 J/K. The temperature is not a conserved quantity along the integration because K and V are not individually conserved.

For liquid argon a Lennard–Jones potential [1],

$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right].$$

is suitable with $\varepsilon = 1.65324 \times 10^{-21}$ J and $\sigma = 3.405 \times 10^{-10}$ m. The mass of the argon atom is 6.64×10^{-26} kg.

We performed the numerical integration of (6) in nondimensional form. The nondimensional time t^* and the nondimensional distance r^* are defined by

$$r = \sigma r^*, \qquad t = \sqrt{\frac{m\sigma^2}{48\varepsilon}}\, t^*,$$

and the nondimensional equations of motion read

$$\frac{\mathrm{d}^2 \boldsymbol{r}_i^*}{\mathrm{d}t^2} = \sum_{j \neq i}\left(\frac{1}{\|\boldsymbol{r}_i^* - \boldsymbol{r}_j^*\|^{14}} - \frac{1}{2\|\boldsymbol{r}_i^* - \boldsymbol{r}_j^*\|^{8}}\right)(\boldsymbol{r}_i^* - \boldsymbol{r}_j^*).$$

Note that the computation of each component of the force requires a cost that grows like $O(N)$ so that the cost of evaluating $\boldsymbol{g}$ in (1) is $O(N^2)$. It is then reasonable to measure the cost of the different integrators by the number of $\boldsymbol{g}$ evaluations they need.

To avoid boundaries, we assume that the liquid fills the whole three-dimensional space. In order to have a system with a finite number of degrees of freedom, we impose periodicity as follows.

We consider a box

$$\mathscr{B} = \{0 \leqslant x^* \leqslant L^*, 0 \leqslant y^* \leqslant L^*, 0 \leqslant z^* \leqslant L^*\}, \quad L^* = 6.75$$

containing 256 atoms of argon and imagine that this box is replicated periodically, with period L^* in the x^*, y^*, z^* directions. Thus, the index i in (6) runs from 1 to $N = 256$. However, the summation index j in (6) still takes infinitely many values. Each of the N atoms in the primary box $\mathscr{B}$ is subject to forces not only from the other $N - 1$ atoms in $\mathscr{B}$ but also by all atoms in the periodic copies of $\mathscr{B}$. In order to deal with finite sums we use the "minimum image" [1] convention. Given two atoms i and j in $\mathscr{B}$, we assume that, amongst the infinitely many periodic copies of j including j itself, only that closest to i exerts force on i. This truncation has little effect on the equations of motion because the Lennard–Jones forces are virtually zero at long distances. A similar truncation is performed in the computation of the potential energy.

The above values of N and L^* correspond to a density of $1.4\,\mathrm{g\,cm^{-3}}$. We set the temperature of the simulation at 86.4956 K. To get an initial condition compatible with that temperature, a preliminary integration was carried out by the Verlet algorithm. For this preliminary run the atoms are initially located at the nodes of a face-centered cubic lattice and given small random initial velocities. At constant time intervals, all the velocities $\boldsymbol{v}_i$ are rescaled and become $\beta \boldsymbol{v}_i$, where β is a scalar dynamically chosen for (7) to hold with T equal to the target temperature. Each rescaling may be seen as a sudden burst of heat given to the system so as to bring it to the target temperature. After each of these such sudden changes the temperature does not remain constant because part of the kinetic energy is transformed into potential energy. Nevertheless, in successive rescalings the scaling factor β approaches 1, so that after a while no further rescaling is necessary. This produces the initial state to be used in the simulations.

4. Results and conclusions

The liquid argon was simulated for a time interval $0 \leqslant t^* \leqslant 65.536$. This represents 20.31 ps in physical time t. The nondimensional time-step was chosen from the sequence 0.256, 0.128, 0.064, ... With the largest time-step 0.256, 256 steps are needed to span the time-interval. The step $\Delta t^* = 0.064$ corresponds to 0.02 ps of physical time and may be considered as a typical step to be used with the Verlet algorithm.

We have measured the standard deviation of the errors in the total energy H at 64 equally spaced points along the time interval. This is a standard test for molecular dynamics methods, which are typically used to obtain estimates of macroscopic magnitudes rather than to provide a detailed description of the evolution of each individual atom. Fig. 1 is an efficiency plot; the vertical axis corresponds to the relative energy error and the horizontal axis is work as measured by the number of $\boldsymbol{g}$ evaluations. This figure refers to an idealized parallel environment where the evaluation of $\boldsymbol{g}(\boldsymbol{Q}_1^{[i]})$ and $\boldsymbol{g}(\boldsymbol{Q}_2^{[i]})$ may be performed in the same computer time as the evaluation of one of them and is therefore counted as a single force evaluation. The following runs are reported.

- The Verlet method ran with $\Delta t = 0.128, \ldots, 0.008$. For $\Delta t^* = 0.256$ the scheme is unstable. These runs are displayed by crosses joined by a solid line.
- The Gauss method with $\nu = 1$ ran with $\Delta t^* = 0.128, \ldots, 0.032$ ($\times$ signs). The choice $\Delta t^* = 0.256$ lead to instability.

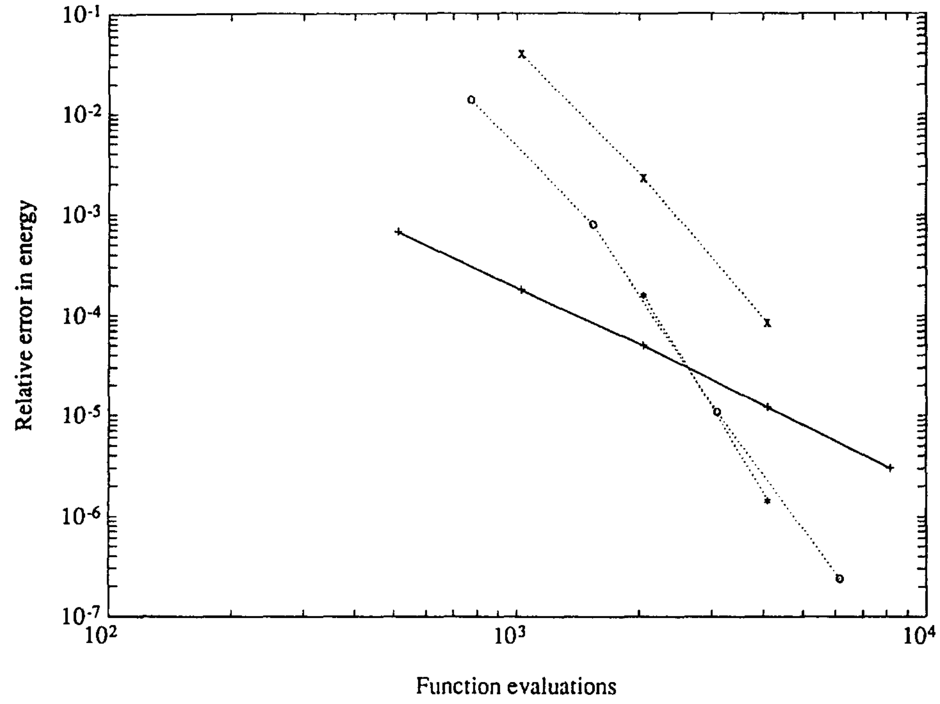

Fig. 1. Efficiency in a parallel setting.

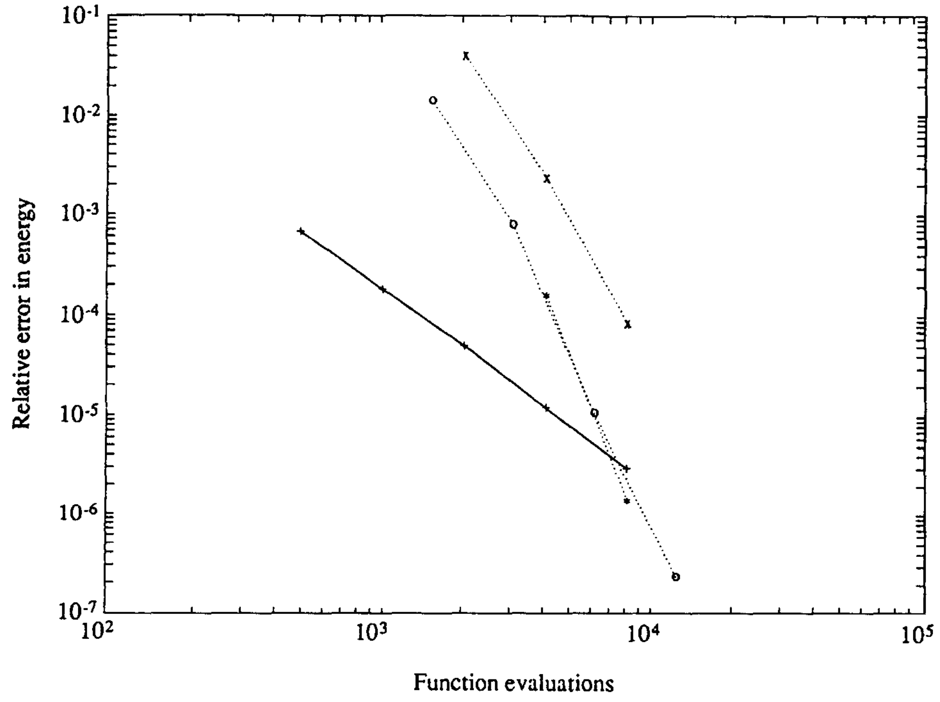

Fig. 2. Efficiency in a sequential setting.

- The Gauss method with $v = 2$ ran with $\Delta t^* = 0.256, \ldots, 0.032$ (circles).
- The Gauss method with $v = 3$ ran with $\Delta t^* = 0.128$ and $\Delta t^* = 0.064$ (stars). The choice $\Delta t^* = 0.256$ leads to instability.

The higher rate of convergence of the Gauss integrators is clearly seen in the figure. Note also that, in terms of efficiency, the choice $v = 2$ improves on the choice $v = 1$, but $v = 3$ is no better than $v = 2$. Experiments not given in the figure further reveal that the efficiency for $v = 4$ is the same as that for $v = 2$ or 3. When comparing the Verlet and the Gauss runs it is apparent that the Verlet methods is more efficient than the Gauss methods except if one insists in energy errors too small to be of practical interest. This conclusion is at variance with that in [6] where it is claimed that "the proposed implicit Runge–Kutta method should become the choice for molecular dynamics integrations". Probably, the discrepancy with [6] is created because in that paper the reader's attention is directed to a comparison between errors of different algorithms operating with a given Δt^*; here we compare errors per unit of work.

Fig. 2 is identical to Fig. 1 except for the fact that it refers to a sequential computer environment where the computation of $g(Q_1^{[i]})$ and $g(Q_2^{[i]})$ requires twice as much time as the evaluation of one of them. As expected, the advantages of the Verlet method are in this case still are more marked.

Acknowledgements

The research of the first two authors has been supported by project DGICYT PB92-254. This work was done while J.C. Díaz was visiting the University of Valladolid as an Iberdrola Visiting Professor. We are thankful to Prof. R.D. Skeel for reading the manuscript.

References

[1] M.P. Allen and D.J. Tildesley, *Computer Simulations of Liquids* (Clarendon Press, Oxford, 1987).
[2] M.P. Calvo and J.M. Sanz-Serna, The development of variable step symplectic integrators, with application to the two-body problem, *SIAM J. Sci. Comput.* **14** (1993) 936–952.
[3] E. Hairer, S.P. Nørsett and G. Wanner, *Solving Ordinary Differential Equations I, Nonstiff Problems* (Springer, Berlin, 2nd ed., 1993).
[4] E. Hairer and G. Wanner, *Solving Differential Equations II, Stiff and Differential-Algebraic Problems* (Springer, Berlin, 1991).
[5] P.J. van der Houwen and B.P. Sommeijer, Parallel iteration of high-order Runge–Kutta methods with stepsize control, *J. Comput. Appl. Math.* **29** (1990) 111–127.
[6] D. Janežič and B. Orel, Implicit Runge–Kutta method for molecular dynamics integration, *J. Chem. Inform. Comput. Sci.* **33** (1993) 252–257.
[7] D.I. Okunbor and R.D. Skeel, Explicit canonical methods for Hamiltonian systems, *Math. Comput.* **59** (1992) 439–455.
[8] D.I. Okunbor and R.D. Skeel, Canonical numerical methods for molecular dynamics simulations, *J. Comput. Chem.* **15** (1994) 72–79.
[9] J.M. Sanz-Serna, Runge–Kutta schemes for Hamiltonian systems, *BIT* **28** (1988) 877–883.
[10] J.M. Sanz-Serna and M.P. Calvo, *Numerical Hamiltonian Problems* (Chapman and Hall, London, 1994).
[11] R.D. Skeel, J.J. Biesiadecki and D. Okunbor, Symplectic integration for macromolecular dynamics, in: *Proc. Internat. Conf. Computation of Differential Equations and Dynamical Systems* (World Scientific, Singapore, 1993) 49–61.
[12] B.P. Sommeijer, Explicit, high-order Runge–Kutta–Nyström methods for parallel computers, *Appl. Numer. Math.* **13** (1993) 221–240.