INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2008; **73**:407–426
Published online 2 May 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2078

A heterogeneous space–time full approximation storage multilevel method for molecular dynamics simulations

Haim Waisman and Jacob Fish*, †

Scientific Computation Research Center (*SCOREC*), *Rensselaer Polytechnic Institute*, *Troy*, *NY 12180-3590*, *U.S.A.*

SUMMARY

A heterogeneous space–time full approximation storage (HFAS) multilevel formulation for molecular dynamics simulations is developed. The method consists of a waveform Newton smoothing that produces initial space–time iterates and a coarse model correction. The formulation is coined as heterogeneous since it permits different interatomic potentials to be applied at different physical scales. This results in a flexible framework for physics coupling. Time integration is performed in windows using the implicit Newmark predictor–corrector method that permits larger time integration steps than the explicit method. The size of the time steps is governed by accuracy rather than by stability considerations of the algorithm. We study three different variants of the method: the Picard iteration, constrained dynamics and force splitting. Numerical examples show that FAS based on force splitting provides significant time savings compared to standard explicit methods and alternative implicit space–time schemes. Parallel studies of the Picard iteration on harmonic problems illustrate the time parallelization effect that leads to a superior parallel performance compared to explicit methods. Copyright © 2007 John Wiley & Sons, Ltd.

Received 11 August 2006; Revised 16 March 2007; Accepted 21 March 2007

KEY WORDS: waveform relaxation; space–time multilevel; multigrid; FAS; molecular dynamics integration; Hessian matrix

1. INTRODUCTION

The dynamics of polymer chains has been studied for decades and is still an active field of research. Simplifying theories confine lateral fluctuations of polymers to a tube-like region around some mean conformation [1]. These theories are limited to simple cases since microscopic dynamics and viscoelastic properties of polymers are dominated by entanglements. To investigate the properties of polymer chains more accurately a detailed computer simulations is required. One such form is molecular dynamics (MD).

*Correspondence to: Jacob Fish, Scientific Computation Research Center (SCOREC), Rensselaer Polytechnic Institute, Troy, NY 12180-3590, U.S.A.
†E-mail: fishj@rpi.edu

MD can be viewed as a process by which one generates atomic trajectories of a system of particles by direct numerical integration of Newton's equations of motion with the appropriate initial and boundary conditions [2]. The system can be formulated either by dynamic equilibrium consideration or by means of variational principle; it can be expressed as

$$\begin{aligned} M\ddot{d} &= F^{\text{int}}(d) + F^{\text{ext}} \\ d(0) &= d_0 \\ \dot{d}(0) &= v_0 \end{aligned} \tag{1}$$

where d is a vector of atom positions, M is the mass matrix, F^{ext} is a vector of external forces, and $F^{\text{int}} = -\nabla\Phi(d)$ is the internal force vector defined as a gradient of the potential energy.

Current MD algorithms severely restrict the modeling efforts to relatively small systems and/or short time intervals. The algorithmic challenges facing MD simulations stem from the difficulty of designing methods which are insensitive in terms of the integration time step to rapid fluctuations in bond stretching. Most widely used algorithms in MD are explicit methods, such as Verlet [3], Swope *et al*. [4] and Gear's predictor–corrector methods [5]. A severe limitation in the ability of the explicit methods to propagate numerical trajectories stems from a wide range of time scales spanning many orders of magnitude. For instance, in polymer chains, bond-stretching vibrations are the fastest atomic motions in a molecule, typically in the order of femtoseconds, whereas the relaxation of polymers in the form of segmental motions or terminal relaxations of chains spans time scales in the range of 10^{-2}–10^{4} s [6]. The maximum time step is governed by the smallest oscillation period that can be found in the simulated system. This time step is necessary to maintain the stability of explicit numerical integration schemes [7].

Major research efforts are devoted to alleviation of this severe time-step requirement. The most popular approach is to constrain bond lengths using either SHAKE or RATTLE algorithms [8]. In this approach, bonds are constrained to have a fixed length. Typically, freezing all bond length coordinates enables one to significantly increase the time step compared to unconstrained MD simulation. Another commonly used approach in MD simulations is to employ a variable time step using multiple-time-step (MTS) methods [9–11]. Nevertheless, the increases in the integration time step have been quite modest so far [12].

An alternative approach based on the space–time variational multilevel principle has been recently developed by Waisman and Fish [13]. The method consists of the waveform relaxation (WR) scheme aimed at capturing the high-frequency response of atomistic vibrations and a coarse-scale solution aimed at resolving smooth features of the discrete medium. The method is implicit in space and time and thus allows for larger time steps governed by accuracy considerations of coarse-scale quantities of interest. The evolution of the coarse-scale equations requires force field calculations on the fine scale, which governs the computational cost of the method.

In this paper we propose a new multilevel method where the coarse problem is evolved without force field calculations on the fine scale. The formulation is based on a variant of the non-linear multigrid theory, the approach known as the full approximation storage (FAS) [14]. The proposed variant of FAS allows for the consideration of different force fields at various scales; it results in added flexibility and superior computational performance. We emphasize that the formulation is general and may be applied to various problems, for example, protein folding, solvation of substances in water, simulations of lipids and peptides and more. The paper is organized as follows. In Section 3, we develop the heterogeneous space–time FAS

DOI: 10.1002/nme

(HFAS) formulation for efficient solution of MD equations. We study three formulation variants, the Picard iteration, constrained dynamics and force splitting, which differ in the method of solving the coarse-scale equations. Performance studies on polymer melts are conducted in Section 4.

2. REVIEW OF THE SPACE–TIME MULTILEVEL AND FAS METHODS

In this section we briefly review the space–time variational multilevel approach [13] and the FAS multigrid method, a combination of which serves the foundation for the current method. The space–time variational multilevel approach consists of two phases: smoothing and coarse grid correction. Smoothing, described in Section 2.1 captures the high-frequency response of the atomistic vibrations whereas the coarse-scale correction (see Section 2.2), formulated as a minimization on the subspace of coarse-scale functions, resolves the smooth features of the discrete (atomistic) medium.

2.1. The waveform relaxation scheme for molecular dynamics

The WR is an iterative solution method of evolutionary problems that offers parallelization [15, 16]. Due to its implicit nature, WR provides superior stability and larger time steps compared to explicit time-stepping methods. In the WR algorithm, the space–time domain is partitioned in space into smaller subsystems. Each subsystem is then integrated over a certain time interval called window; the total time integration is the union of all windows. Windows are used to accelerate convergence and to reduce storage. Information transfer between different windows takes place once the integration within a certain window is completed. The main advantage of the method is that it permits simultaneous integration of several subsystems in each window and its ability for unstructured integration. In this paper we adopt a non-linear version of the WR, known as the waveform Newton (WN) [17, 18] scheme. By this approach the internal force in Equation (1) is approximated by

$$F^{\text{int}} = F^{\text{int}}(d^{\nu}) + D(d^{\nu})(d^{\nu} - d^{\nu+1}) \tag{2}$$

where $D(d^{\nu}(t)) = \text{diag}(H(t))$ is a diagonal of the Hessian matrix obtained from the second derivative of the potential function

$$H_{ij}(t) = \frac{\partial^2 \phi(d(t))}{\partial d_i \partial d_j} \tag{3}$$

Alternatively, D can be defined as a block diagonal matrix to include blocks of atoms. Usually, the Hessian matrix can be computed analytically.

Substituting approximation (2) for the internal forces into (1) leads to the following system of ordinary differential equations:

$$\begin{aligned} M\ddot{d}^{\nu+1} + D(d^{\nu})d^{\nu+1} &= F^{\text{int}}(d^{\nu}) + D(d^{\nu})d^{\nu} \\ d^{\nu+1}(0) &= d_0 \\ \dot{d}^{\nu+1}(0) &= v_0 \end{aligned} \tag{4}$$

DOI: 10.1002/nme

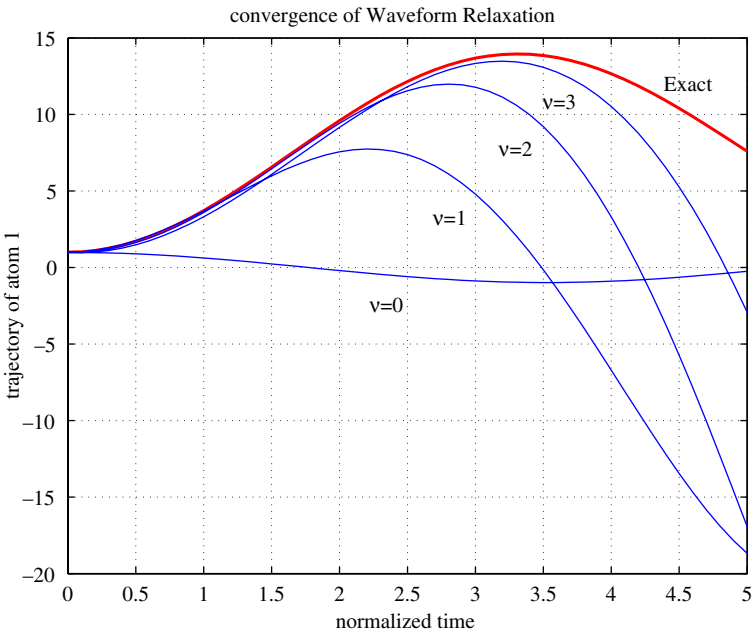

Figure 1. Space–time convergence of waveform relaxation methods.

The above system is integrated over the time window $t \in [t_0, t_n]$ using the Newmark predictor–corrector algorithm [19]. Note that if D is diagonal the WR iteration scheme given in Equation (4) is explicit, even though the overall integrator is implicit. The WR iteration is terminated when the maximum residual in a time window is smaller than a specified tolerance

$$\max\{\|r^{\nu+1}(t)\|_2\} = \max\{\|Md^{\nu+1} - F^{\text{int}}(d^{\nu+1})\|_2\} \tag{5}$$

An illustration of the convergence of the WR method to a trajectory of a single atom is shown in Figure 1. It can be seen that the WR method converges to the entire trajectory as opposed to sequentially advancing in time as in classical explicit and implicit integrators. The major drawback of the WR method is its slow convergence in case of strong coupling between subsystems, sizable windows and large implicit time steps [20, 21].

The convergence of the WR method can be accelerated using a coarse grid correction, in which case the WR takes the role of a smoothing aimed at capturing the high-frequency response of atomistic vibrations.

2.2. *Variational space–time multilevel method*

In the variational scheme the coarse-grained equations are constructed directly from the fine scale using Hamilton's principle on the subspace of the coarse-scale functions. Let $e(t)$ be the coarse-scale correction aimed at updating the fine-scale solution, where $m \ll N$ is the size of the coarse model. The updated fine-scale solution at a certain time step is given by

$$d^{\nu+1}(t) \leftarrow d^{\nu}(t) + Qe(t) \tag{6}$$

DOI: 10.1002/nme

where Q is the prolongation operator (assumed to be constant over a certain period of time). To find the optimal correction we express the Lagrangian in terms of the correction Qe

$$L(Qe, Q\dot{e}) = \tfrac{1}{2}\langle M(\dot{d}^v + Q\dot{e}),\ (\dot{d}^v + Q\dot{e})\rangle - \Phi(d^v + Qe) \tag{7}$$

By Hamilton's principle $e(t)$ is the minimizer of

$$S[e(t)] = \int_{t_1}^{t_2} L(Qe, Q\dot{e})\,\mathrm{d}t \tag{8}$$

written as

$$\frac{\delta S}{\delta e(t)} = 0, \quad t_1 < t < t_2 \tag{9}$$

which is equivalent to solving the following Euler–Lagrange equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{e}}\right) - \frac{\partial L}{\partial e} = \frac{\delta S}{\delta e(t)} = 0 \tag{10}$$

Substituting the Lagrangian into (47) results in the following coarse grid problem:

$$\begin{aligned} Q^{\mathrm{T}} M Q\ddot{e} - Q^{\mathrm{T}} F^{\mathrm{int}}(d^v + Qe) &= -Q^{\mathrm{T}} M \ddot{d}^v \\ e(0) &= 0 \\ \dot{e}(0) &= 0 \end{aligned} \tag{11}$$

Equation (11) depicts the coarse grid correction in space and time. System (11) is integrated implicitly using the Newmark predictor–corrector method. Once the error $e(t)$ is calculated it is prolongated to the fine scale at each time step within a window (see Equation (58)). Algorithmic details of the variational space–time multilevel method can be found in [13]. The formulation has been found to provide a superior rate of convergence in terms of the number of cycles taken to converge, but is computationally expensive since it involves computations of the force fields obtained on the fine scale, i.e. forces acting on the coarse model (the term $Q^{\mathrm{T}} F^{\mathrm{int}}(d^v + Qe)$ in Equation (11)) are computed on the fine scale and then projected to the coarse scale. This is analogous to the Galerkin projection in the classical algebraic multigrid.

2.3. The FAS multilevel algorithm

There are two basic multilevel approaches for solving a non-linear discrete system of equations arising from partial differential equations: (i) the Newton multigrid and (ii) Full Approximation Storage (FAS) multigrid. In Newton multigrid the non-linear problem is solved using Newton's method, where a standard linear multigrid is applied to solve a linearized system of equations. In the following we focus on the latter, the FAS approach, which directly utilizes multigrid principles to solve a non-linear system of equations.

The FAS scheme has been introduced in the seminal paper by Brandt [22] in the 1970s. The algorithm has been developed to solve non-linear discrete problems of the form

$$L(d) = f \quad \text{in } \Omega \tag{12}$$

DOI: 10.1002/nme

arising from partial differential equations, where $d, f \in \mathbb{R}^N$, L is the operator and Ω is a given mesh. We begin by denoting d_f an approximation to the exact solution d and by e the error,

$$e = d - d_\mathrm{f} \tag{13}$$

We will use subscript f to denote the fine grid variables and subscript c for the coarse variables. In the linear case, $L(d) = Kd$, the residual $r = f - Kd_\mathrm{f}$ satisfies the residual equation,

$$Ke = r \tag{14}$$

In the non-linear case, given the approximation d_f, the residual is

$$r = f - L(d_\mathrm{f}) \tag{15}$$

Subtracting Equation (12) from Equation (15) gives,

$$L(d) - L(d_\mathrm{f}) = r \tag{16}$$

This suggests that $L(e) \neq r$ and the coarse grid correction cannot be written in the form of Equation (14). In FAS, multigrid is directly applied to Equation (12) with Equation (16) used as the coarse grid correction. The process begins first by applying some relaxation (smoothing) technique to Equation (12) to obtain the approximate solution d_f. Many linear relaxation schemes have analogs to non-linear systems [23].

Based on Equation (16) we define the coarse grid equation as

$$L_\mathrm{c}(d_\mathrm{c} + e_\mathrm{c}) - L_\mathrm{c}(d_\mathrm{c}) = r_\mathrm{c} \tag{17}$$

where L_c is the coarse grid operator and r_c is the projection of the fine grid residual onto the coarse problem

$$r_\mathrm{c} = Rr_\mathrm{f} = R(f - L_\mathrm{f}(d_\mathrm{f})) \tag{18}$$

where R is the residual restriction operator. Similarly, d_c is the restriction of the solution approximation onto the coarse grid

$$d_\mathrm{c} = \bar{R}d_\mathrm{f} \tag{19}$$

where $\bar{R}$ is the solution restriction operator. Note that in general R and $\bar{R}$ are different operators. Substituting Equation (18) and Equation (19) into Equation (17) yields

$$L_\mathrm{c}(\bar{R}d_\mathrm{f} + e_\mathrm{c}) = L_\mathrm{c}(\bar{R}d_\mathrm{f}) + R(f - L_\mathrm{f}(d_\mathrm{f})) \tag{20}$$

Defining,

$$u_\mathrm{c} = \bar{R}d_\mathrm{f} + e_\mathrm{c} \tag{21}$$

$$f_\mathrm{c} = L_\mathrm{c}(\bar{R}d_\mathrm{f}) + R(f_\mathrm{f} - L_\mathrm{f}(d_\mathrm{f})) \tag{22}$$

DOI: 10.1002/nme

we obtain the following coarse grid equation:

$$L_c(u_c) = f_c \tag{23}$$

which is in the form of the original fine-scale equation. Note that the coarse operator L_c has not been defined so far. For mildly non-linear problems, L_c is often defined to be identical to L_f, whereas for highly non-linear problems a more expensive Galerkin projection

$$L_c(u_c) = RL_f(Qu_c) \tag{24}$$

can be use instead. In (24) Q is the prolongation operator, typically defined as a transpose of the residual restriction operator, R. Once u_c in Equation (23) has been found, the coarse grid error is computed

$$e_c = u_c - \bar{R}d_f \tag{25}$$

and interpolated back to the fine grid

$$d_f^{new} = d_f + Qe_c = d_f + Q(u_c - \bar{R}d_f) \tag{26}$$

This completes a single cycle of the two-level FAS method. The two-level FAS algorithm is summarized in Algorithm 1. A multilevel version of FAS can be obtained by recursive application of the two-level method on the coarse grid.

Algorithm 1 (A two-level FAS scheme)

1: $d_f \leftarrow smooth(L_f, f_f, d_f)$	for some initial guess d_f relax v_1 times
2: $r_f = f_f - L_f(d_f)$	compute fine level residual
3: $r_c = Rr_f$	restrict the residual onto the coarse grid
4: $d_c = \bar{R}d_f$	restrict the solution iterate
5: $L_c(u_c) = L_c(d_c) + r_c$	solve the coarse grid problem
6: $d_f \leftarrow d_f + Q(u_c - d_c)$	prolongate the correction
7: $d_f \leftarrow smooth(L_f, f_f, d_f)$	for some initial guess d_f relax v_2 times

Remark

The method is coined the FAS since the coarse problem is solved for the full approximation (not for the error e_c) [22, 24]. It obviates the need to form and store the Jacobian matrix associated with the Newton method. This is in particular important for large-scale problems, where memory is the limiting factor [25]. Note also that if the operator L_f is linear then FAS reduces to the standard multigrid method for solving linear system of equations.

3. HETEROGENEOUS SPACE–TIME FAS FOR MOLECULAR DYNAMICS SIMULATIONS

In this section, we develop the HFAS approach for MD simulations. The HFAS differs from the original FAS approach in two respects. First, the method is used to solve a space–time problem. Second, different operators are employed at different levels (scales). This is in contrast to the classical FAS theory where the same operators are utilized [26, 27] at different levels. The motivation for the latter is the following. At the fine scale, the interactions between atoms, and therefore the operator L_f, are governed by interatomic force fields. We will denote the interatomic

DOI: 10.1002/nme

potentials on the fine level as 'fine scale potentials.' At the coarse scale (coarse-grained models), the interactions are between the representative blobs in polymers or dislocation lines in metals; these interactions are governed by a completely different set of physical laws, requiring different formulation for L_c.

We now focus on the formulation and algorithmic details of the method. As in the space–time variational multilevel method [13], the first step is pre-smoothing in the space–time domain. This is accomplished using the WN method (see Section 2.1). Similarly to steady-state non-linear problems, we define the space–time problem over a certain time window as

$$\begin{aligned} L_f(\ddot{d}_f^{v+1}(t), d_f^{v+1}(t)) &= f_f \\ d_f^{v+1}(0) &= d_0 \\ \dot{d}_f^{v+1}(0) &= v_0 \end{aligned} \tag{27}$$

where d_f^{v+1} is a vector of positions of atoms after smoothing iteration $v+1$, and

$$L_f(\ddot{d}_f^{v+1}(t), d_f^{v+1}(t)) = M\ddot{d}_f^{v+1}(t) - F_f^{\text{int}}(d_f^{v+1}(t)) \tag{28}$$

$$f_f = F_f^{\text{ext}} \tag{29}$$

The residual over a space–time window is given by

$$r_f^{v+1}(t) = L_f(\ddot{d}_f^{v+1}(t), d_f^{v+1}(t)) - f_f \tag{30}$$

Following Equations (18) and (19) the restriction of the approximate solution and the residual yields

$$r_c(t) = Rr_f^{v+1}(t) = R(L_f(\ddot{d}_f^{v+1}(t), d_f^{v+1}(t)) - f_f) \tag{31}$$

$$d_c(t) = \bar{R}d_f^{v+1}(t) \tag{32}$$

The HFAS scheme is then defined as

$$\begin{aligned} L_c(\ddot{u}_c(t), u_c(t)) &= f_c \\ u_c(0) = \bar{R}d^{v+1}(0) &= \bar{R}d_0 \\ \dot{u}_c(0) = \bar{R}v^{v+1}(0) &= \bar{R}v_0 \end{aligned} \tag{33}$$

where the initial conditions d_0 and v_0 are simply the restriction of the fine-scale initial conditions; u_c and f_c in (33) are defined as (see Equations (21)–(22))

$$u_c(t) = \bar{R}d_f^{v+1}(t) + e_c(t) \tag{34}$$

$$f_c(t) = L_c(R\ddot{d}_f^{v+1}(t), \bar{R}d_f^{v+1}(t)) + R(f_f - L_f(\ddot{d}_h^{v+1}(t), d_h^{v+1}(t))) \tag{35}$$

DOI: 10.1002/nme

Substituting Equations (28) and (35) into Equation (33), we obtain the following coarse-grained model (for simplicity of the notation we drop t and $v+1$ variables):

$$\begin{aligned} L_{\mathrm{c}}(\ddot{u}_{\mathrm{c}}, u_{\mathrm{c}}) &= L_{\mathrm{c}}(R\ddot{d}_{\mathrm{f}}, \bar{R}d_{\mathrm{f}}) + R(F_{\mathrm{f}}^{\mathrm{ext}} - M\ddot{d}_{\mathrm{f}} - F_{\mathrm{f}}^{\mathrm{int}}(d_{\mathrm{f}})) \\ u_{\mathrm{H}}(0) &= \bar{R}d_0 \\ \dot{u}_{\mathrm{H}}(0) &= \bar{R}v_0 \end{aligned} \tag{36}$$

The coarse-scale operator L_{c} can take various forms. For instance, it could be represented by the coarse-grained molecular dynamics (CGMD) model [28, 29]

$$L_{\mathrm{c}}(\ddot{u}_{\mathrm{c}}, u_{\mathrm{c}}) = M_{\mathrm{c}}\ddot{u}_{\mathrm{c}} - F_{\mathrm{c}}^{\mathrm{int}}(u_{\mathrm{c}}) \tag{37}$$

where M_{H} is the mass matrix of the coarse problem and $F_{\mathrm{c}}^{\mathrm{int}}(u_{\mathrm{c}})$ derived from the coarse-grained Hamilton's equations under fixed thermodynamic conditions. Substituting Equation (37) into (36) yields the coarse problem

$$\boxed{\begin{aligned} M_{\mathrm{c}}\ddot{u}_{\mathrm{c}} - F_{\mathrm{c}}^{\mathrm{int}}(u_{\mathrm{c}}) &= M_{\mathrm{c}}R\ddot{d}_{\mathrm{f}} - F_{\mathrm{c}}^{\mathrm{int}}(\bar{R}d_{\mathrm{f}}) + R(F_{\mathrm{f}}^{\mathrm{ext}} - M\ddot{d}_{\mathrm{f}} - F_{\mathrm{f}}^{\mathrm{int}}(d_{\mathrm{f}})) \\ u_{\mathrm{c}}(0) &= \bar{R}d_0 \\ \dot{u}_{\mathrm{c}}(0) &= \bar{R}v_0 \end{aligned}} \tag{38}$$

Equation (38) relates the fine and coarse-scale physics obtained by the HFAS formulation.

To this end we focus on a variant of HFAS, which constructs an auxiliary model by simplifying force field calculations rather than by spatial coarsening. In the absence of coarsening, $R = \bar{R} = I$ (I is identity), $M_{\mathrm{c}} = M$, and further assuming $F_{\mathrm{f}}^{\mathrm{ext}} = 0$, gives

$$\boxed{\begin{aligned} M\ddot{u}_{\mathrm{c}} - F_{\mathrm{c}}^{\mathrm{int}}(u_{\mathrm{c}}) &= F_{\mathrm{f}}^{\mathrm{int}}(d_{\mathrm{f}}) - F_{\mathrm{c}}^{\mathrm{int}}(d_{\mathrm{f}}) \\ u_{\mathrm{c}}(0) &= d_0 \\ \dot{u}_{\mathrm{c}}(0) &= v_0 \end{aligned}} \tag{39}$$

We will refer to (39) as the temporal version HFAS to emphasize that no coarsening is taking place, but rather an additional relaxation.

Remark

Note that choosing $F_{\mathrm{H}}^{\mathrm{int}} = F_h^{\mathrm{int}}$ results in the original MD equations (see Equation (1)).

A two-level framework is illustrated in Algorithm 2. The definition of the variables is as follows. $X = \{d_{\mathrm{f}}^{v+1}(t_i)\}$ and $A = \{\ddot{d}_{\mathrm{f}}^{v+1}(t_i)\}$ are matrices of the approximate solution and acceleration vectors obtained from WN smoothing as a function of time; n is the number of time steps within the current window. For instance, X_i a column of the matrix X corresponds to the atom positions in Cartesian coordinates, at iteration $v+1$ after i time steps; t_0 and t_n define the window interval. v_1 and v_2 are the number of pre- and post-smoothings, respectively. In the next subsections we focus on the choice of $F_{\mathrm{c}}^{\mathrm{int}}$. We consider a fine-scale potential of the form (see Section 1).

DOI: 10.1002/nme

Algorithm 2 (Two level space–time multilevel method)

1: $[M, d_0, v_0, t_0, t_n] = \text{setup}()$
2: $X \leftarrow [d_1, \ldots, d_n]$ initialize on space–time
3: **while** norm $R \geqslant tol$ **do**
4: $[X, A] = WN(M, X, d_0, v_0, t_0, t_n, v_1)$ pre-smooth v_1 times
5: $[X, A] = FAS(M, X, A, t_0, t_n)$ FAS correction
6: $[X, A] = WN(M, X, d_0, v_0, t_0, t_n, v_2)$ post-smooth v_2 times
7: norm $R \leftarrow \max_i \{\| MA_i - F^{\text{int}}(X_i)\|_2\}$ compute residual
8: **end while**

$$\Phi = \Phi_{\text{stretching}} + \Phi_{\text{LJ}} \tag{40}$$

which results in the following internal forces:

$$\mathbf{F}_{\text{f}}^{\text{int}} = -\frac{\partial \Phi}{\partial d_{ij}} = -\left(\frac{\partial \Phi_{\text{LJ}}}{\partial d_{ij}} + \frac{\partial \Phi_{\text{stretching}}}{\partial d_{ij}} \right) = \mathbf{F}_{\text{LJ}}^{\text{int}} + \mathbf{F}_{\text{stretching}}^{\text{int}} \tag{41}$$

where

$$\mathbf{F}_{\text{LJ}}^{\text{int}}(\mathbf{d}_{ij}) = 24 \frac{\varepsilon}{d_{ij}^2} \left\{ 2 \left[\frac{\sigma}{d_{ij}} \right]^{12} - \left[\frac{\sigma}{d_{ij}} \right]^{6} \right\} \mathbf{d}_{ij} \tag{42}$$

$$\mathbf{F}_{\text{stretching}}^{\text{int}}(\mathbf{r}_{ij}) = \frac{k_b}{d_{ij}} (r_0 - d_{ij}) \mathbf{d}_{ij} \tag{43}$$

Computing the non-bonded (LJ) contribution to the interatomic forces is the main expense in the force field calculations. To reduce the amount of computational work we use a neighbor list [2] approach with a cutoff radius r_{cut} such that

$$\Phi_{\text{LJ}}(d_{ij}) = 0 \quad \text{for } d_{ij} > d_{\text{cut}}$$

3.1. Method I: Picard iteration

The simplest choice is $F_{\text{c}}^{\text{int}} = 0$. Using the notation $d_{\text{f}}^{\text{new}} \leftarrow u_{\text{c}}$ and $d_{\text{f}}^{\text{old}} \leftarrow d_{\text{f}}$, and substituting into Equation (39) we get

$$\begin{aligned} M \ddot{d}_{\text{f}}^{\text{new}} &= F_{\text{f}}(d_{\text{f}}^{\text{old}}) \\ d_{\text{f}}^{\text{new}}(0) &= d_0 \\ \dot{d}_{\text{f}}^{\text{new}}(0) &= v_0 \end{aligned} \tag{44}$$

This is known as Picard iteration, where the forces are obtained from an already known atomistic position. In (44) accelerations are obtained by simply dividing the internal force with corresponding diagonal entry of the mass matrix.

DOI: 10.1002/nme

3.2. *Method II*: *Constrained dynamics*

The second variant is based on constrained dynamics. As in Section 3.1, we choose $F_c^{\text{int}} = 0$ and obtain the system of Equations (44). However, we impose bond-stretch constraints to eliminate the fast oscillating components. The constraints are imposed using the popular RATTLE scheme [8]. The possibility of imposing general Holonomic constraints in MD simulations provides the ability to selectively freeze particular degrees of freedom, without interfering with others.

Consider a system of N interacting atoms subjected to bond-stretch constraints

$$\sigma_k(d_i, d_j) = [d_j(t) - d_i(t)]^2 - d_{ij}^2 = 0, \quad k = 1, \ldots, n_s \tag{45}$$

where n_s are the number of constraints, i and j are two atoms involved in the particular constraint σ_k; d_i and d_j are the atom coordinates and d_{ij} is a given distance between them. The Lagrangian of a constrained system may be written using the Lagrange multipliers λ_k

$$L(d, \dot{d}) = \frac{1}{2}\dot{d}^{\mathrm{T}} M \dot{d} - \Phi(d) - \sum_k^{n_s} \lambda_k \sigma_k(d) \tag{46}$$

The constrained equations of motion is then obtained by solving the following Euler–Lagrange equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{d}}\right) - \frac{\partial L}{\partial d} = 0 \tag{47}$$

Substituting Equation (46) into (47) yields,

$$M\ddot{d} = F^{\text{int}}(d) - F_c^{\text{int}}(d) \tag{48}$$

where $F_c^{\text{int}}(d) = \sum_k^{n_s} \lambda_k \partial\sigma_k(d)/\partial d$ is the force resulting from the constraints. Equation (48) can be rewritten according to the FAS framework

$$\begin{aligned} M\ddot{d}_f^{\text{new}} + \sum_k^{n_s} \lambda_k \frac{\partial \sigma_k(d_f^{\text{new}})}{\partial d_f^{\text{new}}} &= F_f(d_f^{\text{old}}) \\ d_f^{\text{new}}(0) &= d_0 \\ \dot{d}_f^{\text{new}}(0) &= v_0 \end{aligned} \tag{49}$$

Note that $F_f(d_f^{\text{old}})$ is a known force, independent of the integration, obtained from the WR process.

We apply the Newmark predictor–corrector scheme to integrate Equations (49). The RATTLE scheme [8] is used to enforce the constraints. The coordinates of atoms i and j are then computed as

$$\begin{aligned} d_i(t_0 + \Delta t) &= d_i'(t_0 + \Delta t) - \frac{\Delta t^2}{m_i}\lambda_k [d_i(t_0) - d_j(t_0)] \\ d_j(t_0 + \Delta t) &= d_j'(t_0 + \Delta t) - \frac{\Delta t^2}{m_j}\lambda_k [d_j(t_0) - d_i(t_0)] \end{aligned} \tag{50}$$

where d_i' and d_j' indicate the position of atoms i and j prior to the application of the constraints; and m_i and m_j are the masses of the atoms. The Lagrange multiplier λ_k is obtained iteratively for

DOI: 10.1002/nme

every bond length from (see [30] for more details)

$$\lambda_k = \frac{[d'_j(t_0+\Delta t) - d'_i(t_0+\Delta t)]^2 - d_{ij}^2}{2\Delta t^2 \left(\dfrac{1}{m_i}+\dfrac{1}{m_j}\right)[d_j(t_0)-d_j(t_0)][d'_j(t_0+\Delta t)-d'_i(t_0+\Delta t)]} \tag{51}$$

In RATTLE one has also to enforce the constraints [8, 31] on the time derivatives of the bond-stretch constraint in Equation (45)

$$\dot{\sigma}_k(d_i, d_j) = 2[d_j(t) - d_i(t)][v_j(t) - v_i(t)] = 0, \quad k = 1, \ldots, n_s \tag{52}$$

The constraints in (52) improve the accuracy of the Velocity-Verlet algorithm. The constrained velocities of atoms i and j are computed in a similar way

$$\begin{aligned} v_i(t_0+\Delta t) &= v'_i(t_0+\Delta t) - \frac{\Delta t}{m_i}\mu_k[d_i(t_0+\Delta t) - d_j(t_0+\Delta t)] \\ v_j(t_0+\Delta t) &= v'_j(t_0+\Delta t) - \frac{\Delta t}{m_j}\mu_k[d_j(t_0+\Delta t) - d_i(t_0+\Delta t)] \end{aligned} \tag{53}$$

where v'_i and v'_j indicate the velocities of atoms i and j before the application of the constraints. The Lagrange multiplier μ_k is obtained iteratively for every bond pair similarly to λ_k (see [30] for more details)

$$\mu_k = \frac{[d'_j(t_0+\Delta t) - d'_i(t_0+\Delta t)][d_j(t_0+\Delta t) - d_i(t_0+\Delta t)]}{\Delta t \left(\dfrac{1}{m_i}+\dfrac{1}{m_j}\right) d_{ij}^2} \tag{54}$$

In HFAS, the constraint distances d_{ij} in Equation (45) are computed from the iterates $d_{\mathrm{f}}^{\mathrm{old}}$ obtained from the WN process.

3.3. Method III: Force field splitting

This variant of the method approximates the force field calculations in an attempt to reduce the computational work. In the present case, the internal forces are calculated from the harmonic approximation of the potential

$$F_{\mathrm{c}}^{\mathrm{int}} = F_{\mathrm{stretching}} \tag{55}$$

where $F_{\mathrm{stretching}}$ is the force field due to bond stretching. Substituting the terms $F_{\mathrm{c}}^{\mathrm{int}}$ in Equation (55) and $F_{\mathrm{f}}^{\mathrm{int}}$ in Equation (41) into the coarse operator Equation (39), and using the notation $d_{\mathrm{f}}^{\mathrm{new}} \leftarrow u_{\mathrm{c}}$ and $d_{\mathrm{f}}^{\mathrm{old}} \leftarrow d_{\mathrm{f}}$ yields

$$\begin{aligned} M\ddot{d}_{\mathrm{f}}^{\mathrm{new}} - F_{\mathrm{stretching}}^{\mathrm{int}}(d_{\mathrm{f}}^{\mathrm{new}}) &= F_{\mathrm{LJ}}(d_{\mathrm{f}}^{\mathrm{old}}) \\ d_{\mathrm{f}}^{\mathrm{new}}(0) &= d_0 \\ \dot{d}_{\mathrm{f}}^{\mathrm{new}}(0) &= v_0 \end{aligned} \tag{56}$$

We note that different splitting strategies may be required for other potential types.

DOI: 10.1002/nme

4. NUMERICAL RESULTS

We study performance of various HFAS formulations applied to MD simulation of polymer melts. The MD simulations are performed under conditions of constant NVE (the microcanonical ensemble) with consideration of periodic boundary conditions. Reduced order units are used for all the simulations. To assess the accuracy of algorithms we track the following ensemble properties: absolute temperature, kinetic energy, configurational (potential) energy, total energy (Hamiltonian) and mean square displacements (self-diffusion coefficients). The time integration of all implicit schemes is performed using the Newmark predictor–corrector algorithm with parameters $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$.

4.1. Polymer melts with Lennard-Jones and harmonic potentials

We consider a unit cell of a polymer melt as shown in Figure 2. Various colors correspond to different chains where every chain consists of 200 atoms. Lennard-Jones (LJ) potentials are used to model the interaction between all pairwise atoms in the system (all atoms including atoms in different chains), and harmonic (stretching) potentials are added along the polymer axis as given in Equation (40). In our simulations we use normalized LJ units with $\varepsilon_{ij} = \sigma_{ij} = 1$ and the stretching potential units $k_{\mathrm{b}}^{\mathrm{bond}} = 270$ and $r_0 = 1$. The resulting polymer system is stiff and thus the explicit Velocity-Verlet algorithm, due to stability consideration, is limited by the fast vibrating components. For the multilevel and waveform methods considered, the length of the time step is governed by the accuracy of the coarse fields of interest, selected here both as temperature and self-diffusion. The temperature is related to the average kinetic energy of the system and is

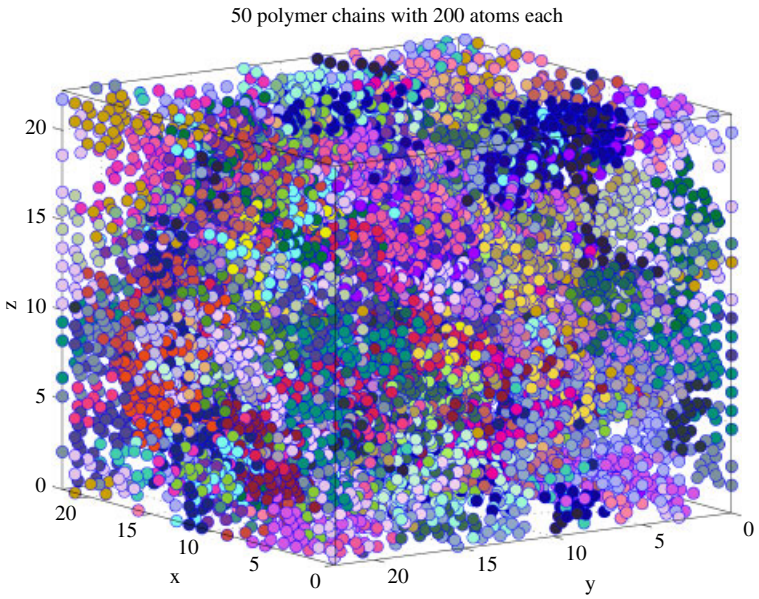

Figure 2. Unit cell of polymer chains.

DOI: 10.1002/nme

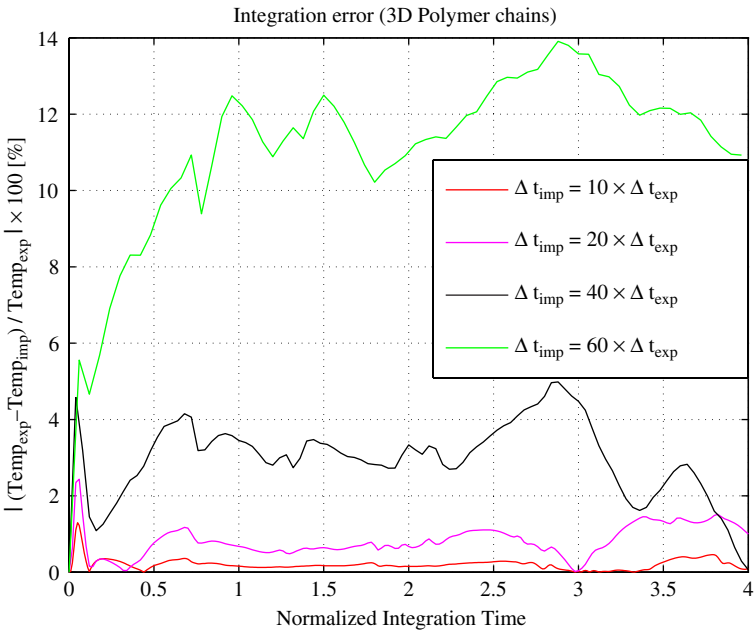

Figure 3. Error in temperature as a function of the time step (simulation of polymer melts). Twenty polymer chains consisting of 16 atoms per chain are used for the plot.

written as

$$\text{Temp} = \frac{2}{3Nk_\text{b}}\langle E_\text{k}\rangle \tag{57}$$

where N is the number of atoms, k_b Boltzmann's constant (here normalized as $k_\text{b} = \frac{2}{3}$) and $\langle E_\text{k}\rangle$ the time average kinetic energy which is a function of atom velocities (see [2] for more details). Figure 3 depicts the relative error in the implicit methods computed as

$$\text{Er}[\%] = \frac{|\text{Temp}_\text{exp} - \text{Temp}_\text{imp}|}{|\text{Temp}_\text{exp}|} \times 100 \tag{58}$$

where Temp_exp is the temperature obtained by the Velocity-Verlet method and Temp_imp is the temperature obtained by HFAS for various time steps. The study is conducted on 20 polymer chains that consist of 16 atoms per chain. We use the explicit method with $\Delta t_\text{exp} = 1 \times 10^{-3}$ as the reference solution. The allowable error in the temperature is selected to be 1.5%. This corresponds to the implicit time step that is 20 times larger than the explicit time step. The adequacy of the selected time step is also verified by inspecting the accuracy of the self-diffusion coefficient. The diffusion coefficient is obtained from the mean square displacements as

$$D_\text{f} = \frac{1}{2n_\text{sd}} \lim_{t\to\infty} \frac{\langle [d(t_0 + t) - d(t_0)]^2\rangle}{t} \tag{59}$$

DOI: 10.1002/nme

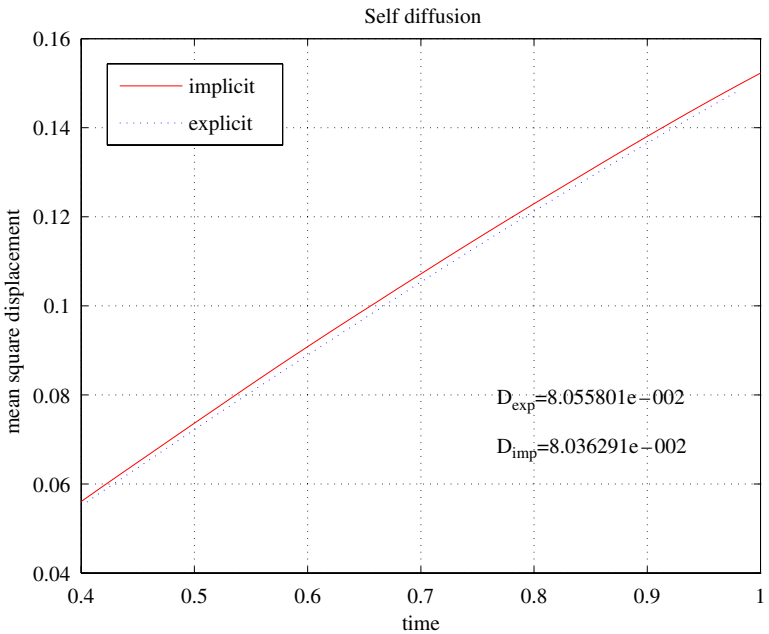

Figure 4. Diffusion coefficient for both explicit and implicit methods. $\Delta t_{\text{imp}} = 20 \times \Delta t_{\text{exp}}$.

where n_{sd} is the number of space dimensions and t_0 the time origin for the ensemble time averages. For more details on diffusion coefficients we refer to [2]. Figure 4 illustrates the diffusion behavior (the slope) for both the explicit and implicit methods taken after long times. The comparison of the diffusion coefficient for $\Delta t_{\text{imp}} = 20 \times \Delta t_{\text{exp}}$ is given in Figure 4. The system is integrated over 40 000 explicit time steps. It can be seen that the multilevel method predicts the diffusion coefficient with only 1% of error compared to the explicit method.

Tables I–III illustrate the performance of the WN and various multilevel methods as compared to the popular Velocity-Verlet scheme. We study 10 polymer chains of short, medium and long lengths consisting of 10, 50 and 200 atoms, respectively. The WN and the variational multilevel scheme were reviewed in Section 2 (see Reference [13] for more details). Various HFAS schemes were presented in Section 3. The system is first equilibrated over 1000 explicit steps by velocity scaling to a preset temperature. The methods are then compared in the production phase. We use the following parameters in normalized units: box dimension of 22 units; the system is integrated over 100 implicit steps (2000 explicit steps). The neighbor list [2] is updated every step for the implicit methods and every 20 explicit steps. A cutoff radius of 8 units is employed for all simulations. We adopt the notation given in Table IV for all methods considered. The diagonal terms of the Hessian matrix in Equations (3)-(4) are computed analytically. One presmoothing of Jacobi WN is applied for all multilevel methods. The iteration is terminated when the residual in Equation (18) is less than 10^{-4} for all times within a window. For the numerical experiments considered the HFAS-III method outperformed the explicit, WN and the other multilevel methods. The best performance is obtained when the window size is equal to the size of the time step.

Finally, we demonstrate stability properties of the implicit multilevel methods by considering a long time interval. We integrate the system over 40 000 explicit time steps (2000 implicit steps).

DOI: 10.1002/nme

Table I. CPU time and iteration summary for 10 polymer chains consisting of 10 atoms per chain (short chains).

Method	Winds	dt	Iteration	Non-linear iteration	Functional evaluation	CPU (s)
VV	1	1×10^{-3}	—	—	2001	611.19
IN	1	20×10^{-3}	—	1441	1442	639.469
WN	50	20×10^{-3}	460	—	750	325.94
WN	100	20×10^{-3}	650	—	660	301.88
ML-var	50	20×10^{-3}	200	760	1050	470.94
ML-var	100	20×10^{-3}	218	654	873	383.906
HFAS-I	50	20×10^{-3}	220	—	750	351.25
HFAS-I	100	20×10^{-3}	263	—	527	235.203
HFAS-II	50	20×10^{-3}	300	—	1075	498.83
HFAS-II	100	20×10^{-3}	425	—	875	409.76
HFAS-III	50	20×10^{-3}	220	720	730	340.78
HFAS-III	100	20×10^{-3}	212	424	425	205.109

Table II. CPU time and iteration summary for 10 polymer chains consisting of 50 atoms per chain (medium length chains).

Method	Winds	dt	Iteration	Non-linear iteration	Functional evaluation	CPU (s)
VV	1	1×10^{-3}	—	—	2001	6625.9
IN	1	20×10^{-3}	—	1670	1671	7802.8
WN	50	20×10^{-3}	480	—	840	4061.3
WN	100	20×10^{-3}	780	—	790	3966.4
ML-var	50	20×10^{-3}	200	900	1750	8309.4
ML-var	100	20×10^{-3}	260	260	530	2644.8
HFAS-I	50	20×10^{-3}	260	—	890	4219.4
HFAS-I	100	20×10^{-3}	280	—	570	2858.6
HFAS-II	50	20×10^{-3}	500	—	1750	8696.1
HFAS-II	100	20×10^{-3}	390	—	790	4063
HFAS-III	50	20×10^{-3}	170	580	590	2874.7
HFAS-III	100	20×10^{-3}	230	460	470	2391.6

Figure 5 depicts fluctuations in the configurational (potential) energy U and the total energy E_{tot} (Hamiltonian). It can be seen that the proposed multilevel approach is stable as the Hamiltonian fluctuates around an average value. This average value differs by only %0.18 from the value obtained by the explicit method. We note that the explicit fluctuations are much smaller as compared to the implicit fluctuations.

Finally, we consider implementation of the explicit method and Picard iteration on a parallel machine for Harmonic potentials. In the explicit method, matrix–vector operations are performed at every explicit time step, and communicated between the processors. In the Picard case, the

DOI: 10.1002/nme

Table III. CPU time and iteration summary for 10 polymer chains consisting of 200 atoms per chain (long chains).

Method	Winds	dt	Iteration	Non-linear iteration	Functional evaluation	CPU (s)
VV	1	1×10^{-3}	—	—	2001	16 499.3
IN	1	20×10^{-3}	—	2275	2300	33 378
WN	50	20×10^{-3}	555	—	1115	16 805
WN	100	20×10^{-3}	835	—	840	13 055
ML-var	50	20×10^{-3}	200	700	1050	10 213
ML-var	100	20×10^{-3}	310	310	625	9978
ML-FAS-I	50	20×10^{-3}	300	—	975	14 664
ML-FAS-I	100	20×10^{-3}	345	—	695	6747.1
ML-FAS-II	50	20×10^{-3}	50	—	1850	17 736
ML-FAS-II	100	20×10^{-3}	650	—	1350	13 658
ML-FAS-III	50	20×10^{-3}	180	580	585	9811.7
ML-FAS-III	100	20×10^{-3}	275	550	551	5804.9

Table IV. Notation used in the Tables I–III.

VV	Velocity-Verlet
IN	implicit Newmark (average acceleration)
WN	waveform Newton
ML-var	space–time multilevel variational scheme
HFAS-I	heterogeneous temporal multilevel FAS—Picard
HFAS-II	heterogeneous temporal multilevel FAS—constraints
HFAS-III	heterogeneous temporal multilevel FAS—force field approximation

system is integrated implicitly and local matrix–matrix operations are performed at the end of every window (several time steps) and communicated between the processors. We use the libraries BLACS, PBLAS and ScaLAPACK for our parallel implementation. Figure 6 shows the time parallelization effect on the speed-up factor over the explicit method. The results clearly show that as the number of processors increases, the speed-up factor between the Picard and Velocity-Verlet methods increases. The main reason for the increase in the speed-up factor is due to the processors' communication effect. In the case of standard explicit or implicit methods, processors are communicating after every time step. However, in the Picard case the communication between processors only takes place at the end of each window. This results in superior parallel performance.

5. CONCLUSIONS

A heterogeneous full approximation storage (HFAS) multilevel formulation for molecular dynamics simulations has been developed. The formulation combines the basic principles of the full approximation storage (FAS) multigrid and the space–time variational multigrid approach developed by Waisman and Fish [13]. It allows for different mathematical models to be considered at

DOI: 10.1002/nme

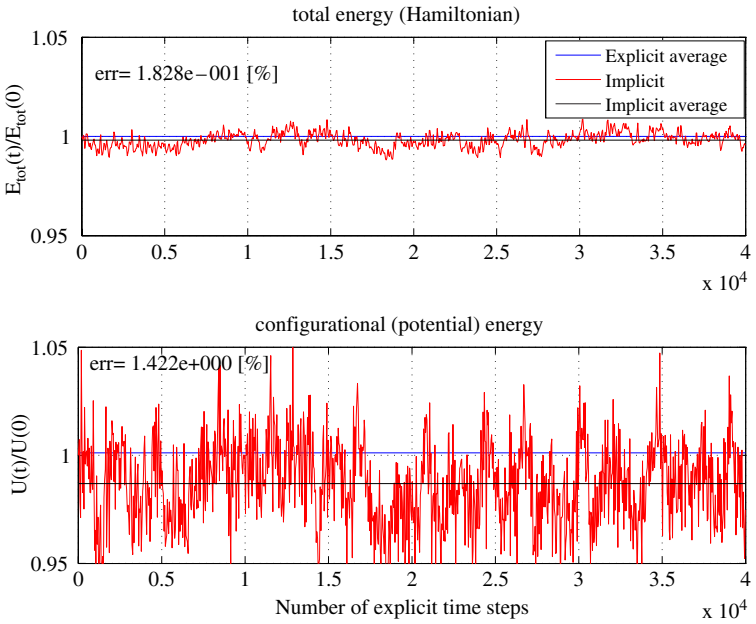

Figure 5. Stability of multilevel method on a long time interval.

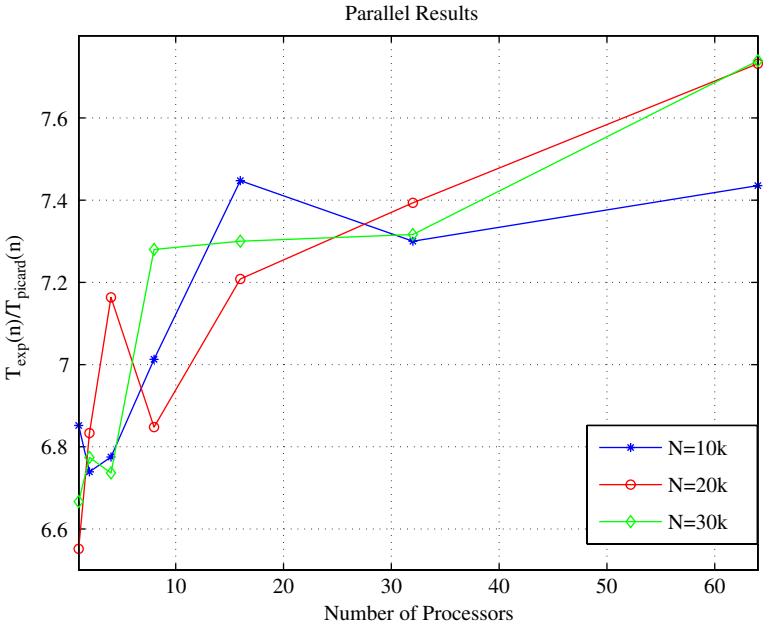

Figure 6. Speed-up ratio of Picard iteration over explicit method.

DOI: 10.1002/nme

different scales. The temporal variant of the method, effectively uses two relaxation schemes: the waveform Newton (WN) scheme, and the implicit integrator employing approximate force field calculations. We study three variants of the method: Picard iteration, constrained dynamics and force splitting. The methods are implicit in space and time, possess superior stability properties and consequently enable larger time steps governed by accuracy considerations of coarse-scale quantities of interest (e.g. temperature, energy, diffusion etc.). Performance studies on polymer melts have shown significant speed-up over the classical explicit methods and the variational space–time scheme. A parallel version of the Picard iteration has been developed for harmonic potentials. Significant speed-ups over the standard explicit method have been observed primarily due to reduced communication time between the processors. Finally, it is important to note that the formulation has been validated for interatomic potentials used to model polymer melts. While the formulation is general it has not been tested for other molecular systems.

REFERENCES

1. de Gennes PG. *Scaling Concepts in Polymer Physics*. Cornell University Press: Ithaca, NY, 1979.
2. Haile JM. *Molecular Dynamics Simulation*: *Elementary Methods*. Wiley: New York, 1992.
3. Verlet L. Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review* 1967; **159**(1):98–103.
4. Swope WC, Anderson HC, Berens PH, Wilson KR. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: application to small water clusters. *The Journal of Chemical Physics* 1982; **76**(1):637–649.
5. Gear CW. *Numerical Initial Value Problems in Ordinary Differential Equations*, Chapter 9. Prentice-Hall: Englewood Cliffs, NJ, 1971.
6. Bennemann C, Paul W, Binder K, Dünweg B. Molecular-dynamics simulations of the thermal glass transition in polymer melts: α-relaxation behavior. *Physical Review E* 1998; **57**(1):843–851.
7. Rapaport DC. *The Art of Molecular Dynamics Simulations*. Cambridge University Press: Cambridge, MA, 1995.
8. Anderson HC. Rattle: a velocity version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics* 1983; **52**:24–34.
9. Tuckerman M, Berne BJ, Martyna GJ. Reversible multiple time scale molecular dynamics. *The Journal of Chemical Physics* 1992; **97**(3):1990–2001.
10. Garcia-Archilla B, Sanz-Serna JM, Skeel RD. The mollified impulse method for oscillatory differential equations. *SIAM Journal on Scientific Computing* 1998; **20**:930–963.
11. Hairer E, Lubich C, Wanner G. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer: Berlin, 2002.
12. Barash D, Yang L, Qian X, Schlick T. Inherent speedup limitations in multiple time step/particle mesh Ewald algorithms. *Journal of Computational Chemistry* 2003; **24**:77–88.
13. Waisman H, Fish J. A space–time multilevel method for molecular dynamics simulations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(44–47):6542–6559.
14. Brandt A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 1977; **31**(138):333–390.
15. Lelarasmee E, Ruehli AE, Sangiovanni-Vincentelli AL. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems* 1982; **CAD-1**(3):131–145.
16. Burrage K. *Parallel and Sequential Methods for Ordinary Differential Equations*. Oxford University Press: Oxford, 1995.
17. Saleh R, White J. Accelerating relaxation algorithms for circuit simulation using waveform-Newton and step-size refinement. *IEEE Transactions on Computer-aided Design* 1990; **9**(9):951–958.
18. Lumsdaine A, Reichelt MW. Waveform iterative techniques for device transient simulation on parallel machines. *Sixth SIAM Conference on Parallel Processing for Scientific Computing*, Norfolk, VA, 1993.
19. Hughes TJR. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1987.

DOI: 10.1002/nme

20. Miekkala U, Nevanlinna O. Convergence of dynamic iteration methods for initial value problems. *SIAM Journal on Scientific and Statistical Computing* 1987; **8**(4):459–482.
21. Giladi E, Keller HB. Space time domain decomposition for parabolic problems. *Numerische Mathematik* 2002; **93**(2):279–313.
22. Brandt A. Multilevel adaptive solutions to boundary-value problems. *Mathematics of Computations* 1977; **31**: 333–390.
23. Ortega JM, Rheinboldt WC. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press: New York, 1970.
24. Briggs WL, Henson VE, McCormick SF. *A Multigrid Tutorial* (2nd edn). SIAM: Philadelphia, PA, 2000.
25. Mavriplis DJ. An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers. *Journal of Computational Physics* 2002; **175**(1):302–325.
26. Dumett MA, Vassilevski P, Woodward CS. A multigrid method for nonlinear unstructured finite element elliptic equations. *Technical Report UCRL-JC-150513*, Lawrence Livermore National Laboratory, 2002.
27. Henson VE. Multigrid methods for nonlinear problems: an overview. *Technical Report UCRL-JC-150259*, Lawrence Livermore National Laboratory, 2003.
28. Rudd RE, Broughton JQ. Coarse-grained molecular dynamics and the atomic limit of finite elements. *Physical Review B* 1998; **58**(10):R5893–R5896.
29. Rudd RE. Coarse-grained molecular dynamics for computer modeling of nanomechanical systems. *International Journal for Multiscale Computational Engineering* 2004; **2**(2):203–220.
30. Kutteh R. Rattle recipe for general holonomic constraints: angle and torsion constraints. *CCP5 Newsletter* 1998; **46**:9–17.
31. Ryckaert JP, Ciccotti G, Berendsen HJC. Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of *n*-alkanes. *Journal of Computational Physics* 1977; **23**:327–341.

DOI: 10.1002/nme