Available online at www.sciencedirect.com

ScienceDirect

Computer Physics Communications 176 (2007) 292–299

Computer Physics Communications

www.elsevier.com/locate/cpc

Pathfinder: A parallel search algorithm for concerted atomistic events

Aiichiro Nakano

Collaboratory for Advanced Computing and Simulations, Department of Computer Science,
Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science,
University of Southern California, Los Angeles, CA 90089-0242, USA

Received 26 October 2006; received in revised form 5 November 2006; accepted 7 November 2006

Available online 1 December 2006

Abstract

An algorithm has been designed to search for the escape paths with the lowest activation barriers when starting from a local minimum-energy configuration of a many-atom system. The pathfinder algorithm combines: (1) a steered eigenvector-following method that guides a constrained escape from the convex region and subsequently climbs to a transition state tangentially to the eigenvector corresponding to the lowest negative Hessian eigenvalue; (2) discrete abstraction of the atomic configuration to systematically enumerate concerted events as linear combinations of atomistic events; (3) evolutionary control of the population dynamics of low activation-barrier events; and (4) hybrid task + spatial decompositions to implement massive search for complex events on parallel computers. The program exhibits good scalability on parallel computers and has been used to study concerted bond-breaking events in the fracture of alumina.

© 2006 Elsevier B.V. All rights reserved.

PACS: 02.70.-c; 02.70.Ns; 82.20.Db

Keywords: Transition state theory; Molecular dynamics; Parallel computing

1. Introduction

Many important material processes occur through a sequence of infrequent events [1,2]. An example is slow crack growth, such as stress corrosion cracking, in which a sequence of bond-breaking events over years leads to a catastrophic failure of a structure [3]. Enumeration of events with low activation barriers and accurate estimation of their barrier energies are essential for understanding microscopic mechanisms of the long-time dynamics as well as for predicting the lifetime of the structure.

Various computational methods have been proposed for carrying out an exhaustive search of activated events in many-atom systems [4,5], including the activation–relaxation technique [6], the dimer method [7], and a variety of eigenvector-following methods [8–11] especially those using the Lanczos algorithm to obtain the lowest eigenvalue of the Hessian matrix and the corresponding eigenvector [12]. In materials with complex microstructures, however, the search for activated events remains a hard computational problem [13,14], since the events with the lowest activation barriers often involve unexpected combinations of elementary atomistic events [15]. It is thus of great importance to design an efficient algorithm with tractable computational complexity to systematically search for such concerted events.

Discrete abstraction [16,17] of atomic configurations enables the use of combinatorial techniques to systematically enumerate concerted events. For example, an atomic configuration can be abstracted as a graph $G = (S_v, S_e)$, in which atoms constitute the set of vertices S_v, and the edge set S_e consists of chemical bonds [18]. Graph-based topological analysis (e.g., shortest-path circuit analysis) of million-to-billion node chemical bond networks has been used successfully to discover complex atomistic events underlying impact-damage [19] and hardening [20] mechanisms of materials.

Another computational technique that can significantly accelerate the combinatorial search for concerted events is evolutionary computation [21,22]. In evolutionary algorithms, a population of candidate solutions in the search space is maintained, and its dynamics is controlled with various techniques (e.g.,

E-mail address: anakano@usc.edu (A. Nakano).

0010-4655/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2006.11.003

recombination and mutation) to obtain approximate solutions while avoiding the combinatorial complexity of the search.

Advanced parallel and distributed computing technologies are also expected to facilitate massive searches for concerted events. Event-search algorithms are often implemented as loosely-coupled parallel applications, in which multiple search tasks are executed concurrently on distributed computers [23–25]. When each search task becomes computationally demanding, a hybrid task + spatial decomposition approach [26,27] can be implemented using the communicator construct in the message passing interface (MPI) language [28], which is a natural migration path to hybrid Grid remote procedure call (GridRPC) + MPI programming on a Grid of geographically distributed parallel computers [29].

This paper presents the design of a search algorithm for activated events with low barrier energies, starting from a local minimum-energy configuration of a many-atom system. The pathfinder algorithm combines: (1) a steered eigenvector-following (SEF) method that guides a constrained escape from the convex region of the minimum and subsequently climbs to a transition state tangentially to the eigenvector corresponding to the lowest negative Hessian eigenvalue; (2) discrete abstraction of the atomic configuration to systematically enumerate concerted events as linear combinations of atomistic events (LCAE); (3) elitist control of the population dynamics of low activation-barrier events; and (4) hybrid task + spatial decompositions (HTSD) to implement massive searches on parallel computers. The program exhibits good scalability on parallel computers and has been used to study concerted bond-breaking events in the fracture of aluminum oxide.

This paper is organized as follows. The next section describes the pathfinder algorithm for systematic event search, and its parallelization is discussed in Section 3. Numerical results are presented in Section 4, and Section 5 contains summary.

2. Pathfinder algorithm

Consider a system of N atoms with its state specified by a $3N$-dimensional vector $\mathbf{R} = [r_{1x}, r_{1y}, r_{1z}, \ldots, r_{Nx}, r_{Ny}, r_{Nz}]^{\mathrm{T}} \in \mathbb{R}^{3N}$, where $\mathbf{r}_i = [r_{ix}, r_{iy}, r_{iz}]^{\mathrm{T}} \in \mathbb{R}^3$ is the position of the ith atom ($\mathbb{R}$ is a set of real numbers, and the superscript T denotes a transpose). The forces $\mathbf{F}$ on the atoms are computed from the potential energy function $V(\mathbf{R})$ as

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_N \end{bmatrix} = \begin{bmatrix} -\partial V/\partial \mathbf{r}_1 \\ \vdots \\ -\partial V/\partial \mathbf{r}_N \end{bmatrix} = -\frac{\partial V}{\partial \mathbf{R}}. \tag{1}$$

Let $\mathbf{R}^{\mathrm{init}}$ be an initial state, which is a local energy-minimum such that $\mathbf{F}(\mathbf{R}^{\mathrm{init}}) = 0$ and such that all the eigenvalues of the Hessian matrix,

$$\mathbf{H} = \partial^2 V/\partial \mathbf{R}^2 \in \mathbb{R}^{3N\times 3N}, \tag{2}$$

are positive at $\mathbf{R}^{\mathrm{init}}$. (For systems with periodic boundary conditions, we filter out the zero-eigenvalue translational motions [30].)

The problem is to find a set of activated events with the lowest barrier energies, starting from $\mathbf{R}^{\mathrm{init}}$. Within the framework of the transition state theory [2,31], we define an event as a triplet of states, $e = (\mathbf{R}^{\mathrm{init}}, \mathbf{R}^{\mathrm{tst}}, \mathbf{R}^{\mathrm{fin}})$, that are interconnected by a continuous escape path $\mathbf{R}(\tau)$ ($\mathbb{R} \to \mathbb{R}^{3N}$; τ is a real-valued parameter such that $\mathbf{R}^{\mathrm{init}} = \mathbf{R}(\tau_{\mathrm{init}})$, $\mathbf{R}^{\mathrm{tst}} = \mathbf{R}(\tau_{\mathrm{tst}})$, and $\mathbf{R}^{\mathrm{fin}} = \mathbf{R}(\tau_{\mathrm{fin}})$ with $\tau_{\mathrm{init}} < \tau_{\mathrm{tst}} < \tau_{\mathrm{fin}}$). The ascent path $\mathbf{R}(\tau_{\mathrm{init}} \leqslant \tau \leqslant \tau_{\mathrm{tst}})$ connects $\mathbf{R}^{\mathrm{init}}$ to a transition state, taken here to be a saddle point $\mathbf{R}^{\mathrm{tst}}$, at which $\mathbf{F}(\mathbf{R}^{\mathrm{tst}}) = 0$, and at which only the lowest eigenvalue λ_1 of the Hessian matrix is negative. The final state $\mathbf{R}^{\mathrm{fin}}$ is another local energy-minimum that is reached along a steepest-descent path $\mathbf{R}(\tau_{\mathrm{tst}} \leqslant \tau \leqslant \tau_{\mathrm{fin}})$, staring from $\mathbf{R} = \mathbf{R}^{\mathrm{tst}}$ pushed slightly away from $\mathbf{R}^{\mathrm{init}}$. The barrier energy of event e is defined as $b(e) = V(\mathbf{R}^{\mathrm{tst}}) - V(\mathbf{R}^{\mathrm{init}})$.

The pathfinder algorithm generates a set of events with low barrier energies in such a way that concerted events are systematically constructed from elementary events. Each event, in turn, is generated from an event seed based on a steered eigenvector-following algorithm. Section 2.1 first defines the event seed and then describes the generation of a single event by the steered eigenvector-following algorithm. Systematic construction of concerted events through the control of event-population dynamics in the pathfinder algorithm is described in Section 2.2.

2.1. Steered eigenvector-following (SEF) event generator

In order to initiate an ascent path $\mathbf{R}(\tau_{\mathrm{init}} \leqslant \tau \leqslant \tau_{\mathrm{tst}})$ from the initial state, $\mathbf{R}(\tau_{\mathrm{init}}) = \mathbf{R}^{\mathrm{init}}$, to a transition state, $\mathbf{R}(\tau_{\mathrm{tst}}) = \mathbf{R}^{\mathrm{tst}}$, we first define an event seed σ as a parameterized sequence of $(3N-1)$-dimensional surfaces $S(\tau)$, in which the atoms' moves are constrained. A specific example for the slow crack-growth problem is a bond-length constraint imposed on a given atomic pair (i, j),

$$\sigma = \bigl\{S(\tau)\bigr\} = \bigl\{\|\mathbf{r}_{ij}\| = r_{ij}(\tau) = r_{ij}^0 + \dot{r}_{ij}(\tau - \tau_{\mathrm{init}})\bigr\}, \tag{3}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, r_{ij}^0 is their bond length in the initial state, and $\dot{r}_{ij}$ is the bond-stretching rate along the path.

The steered eigenvector-following event generator algorithm consists of three algorithmic phases (see Table 1): (1) steered centrifugal escape from the convex region (in which the Hessian matrix is positive definite) of the initial energy-minimum; (2) eigenvector-following climb to a transition state; and (3) steepest descent to reach a final energy-minimum [32].

The steered centrifugal escape phase starts from the initial state $\mathbf{R}^{\mathrm{init}}$, and performs a sequence of steepest-descent steps,

$$\mathbf{R} \leftarrow \mathbf{R} + \frac{\delta\tau^2}{2\langle m\rangle}\mathbf{F}, \tag{4}$$

where $\delta\tau$ (∼1 fs) is a time-discretization unit, and $\langle m\rangle$ is the average mass of the atoms. (Various energy-minimization methods can be used in this step, such as variable-step steepest-descent [12], conjugate-gradient [11] and quasi-Newton [4] methods.) Each steepest-descent step is followed by the projection of state $\mathbf{R}$ onto the constrained surface,

$$\mathbf{R} \leftarrow P\bigl(S(\tau)\bigr)\mathbf{R}, \tag{5}$$

Table 1
Steered eigenvector-following event generation algorithm

```
Algorithm event_generator
Input:
  $\mathbf{R}^{\rm init} \in \mathbb{R}^{3N}$: an initial local minimum-energy state
  $\sigma = \{S(\tau)\}$: an event seed, i.e. a parameterized sequence of $(3N-1)$-dimensional constraint surfaces
Output:
  $e = (\mathbf{R}^{\rm init}, \mathbf{R}^{\rm tst}, \mathbf{R}^{\rm fin})$: an event, i.e. a triplet of initial, transition, and final states

Steps:
  1. Steered centrifugal escape
    $\tau \leftarrow 0$
    $\mathbf{R} \leftarrow \mathbf{R}^{\rm init}$
    do
      $\tau \leftarrow \tau + \delta\tau$
      $\mathbf{R} \leftarrow \mathbf{R} + (\delta\tau^2/2\langle m\rangle)\mathbf{F}$ // steepest-descent step
      $\mathbf{R} \leftarrow P(S(\tau))\mathbf{R}$ // projection onto the constraint surface
    while $\lambda_1 \geqslant -\Delta\lambda_1$

2. Eigenvector-following climb
  do
    $\mathbf{R} \leftarrow \mathbf{R} - \frac{\delta\tau^2}{2\langle m\rangle}(\mathbf{V}^1\mathbf{V}^{1^{\rm T}})\mathbf{F} + \frac{\delta\tau^2}{2\langle m\rangle}(\mathbf{I} - \mathbf{V}^1\mathbf{V}^{1^{\rm T}})\mathbf{F}$ // eigenvector-following step
  while $\max_{i\alpha}\{|f_{i\alpha}| \mid i = 1, \ldots, N; \alpha = x, y, z\} > \Delta f$
  $\mathbf{R}^{\rm tst} \leftarrow \mathbf{R}$

3. Steepest descent
  $\mathbf{R} \leftarrow \mathbf{R}^{\rm tst} + \delta_{\rm os}(\mathbf{R}^{\rm tst} - \mathbf{R}^{\rm init})$ // push the state over the transition state away from the initial state
  do
    $\mathbf{R} \leftarrow \mathbf{R} + (\delta\tau^2/2\langle m\rangle)\mathbf{F}$ // steepest-descent step
  while $\max_{i\alpha}\{|f_{i\alpha}| \mid i = 1, \ldots, N; \alpha = x, y, z\} > \Delta f$
  $\mathbf{R}^{\rm fin} \leftarrow \mathbf{R}$
```

corresponding to the current time τ, where $P(S(\tau))$ is the projection operator [33]. For the bond-length constraint in Eq. (3), the projection operator is expressed as [34]

$$P\big(S(\tau)\big)\mathbf{r}_k = \mathbf{r}_k + \frac{\delta_{ki} - \delta_{kj}}{2}\left(\frac{r_{ij}(\tau)}{\|\mathbf{r}_{ij}\|} - 1\right)\mathbf{r}_{ij} \quad (k = 1, \ldots, N), \tag{6}$$

where $\delta_{ki} = 1$ (if $k = i$) and 0 (else).

After each constrained steepest-descent step, the minimum eigenvalue λ_1 of the Hessian matrix is computed iteratively using the Lanczos algorithm [4,12] in Appendix A. We use a finite-difference method to evaluate the product of the Hessian matrix $\mathbf{H}$ and a vector $\mathbf{Q} \in \mathbb{R}^{3N}$,

$$\mathbf{H}(\mathbf{R})\mathbf{Q} = c_{\rm fd}\big[-\mathbf{F}(\mathbf{R} + \mathbf{Q}/c_{\rm fd}) + \mathbf{F}(\mathbf{R})\big], \tag{7}$$

so that only the forces but not the Hessian matrix need to be computed. We use various divide-and-conquer algorithms to compute the forces in Eq. (7) in $\mathrm{O}(N)$ time. For example, a space–time multiresolution molecular dynamics (MRMD) algorithm [35] and a fast reactive force-field (F-ReaxFF) algorithm [36] are used in cases of classical interatomic potentials and semi-classical reactive force fields, respectively. To compute the forces quantum-mechanically from the Hellmann–Feynman theorem, we use an embedded divide-and-conquer density-functional-theory (EDC-DFT) algorithm [37]. Consequently, the computational complexity of the pathfinder algorithm is $\mathrm{O}(N)$. In Eq. (7), $c_{\rm fd} = \max_{i\alpha}\{|q_{i\alpha}| \mid i = 1, \ldots, N; \alpha = x, y, z\}/\delta_{\rm fd}$ and $\delta_{\rm fd}$ ($\sim 10^{-2}$ Å) is a discretization unit for finite differencing. It typically requires 4–8 force evaluations for λ_1 to converge within a convergence criterion $\Delta_{\rm eigen}$ ($\sim 10^{-3}$). The steered centrifugal escape steps are terminated when λ_1 becomes negative. For systems with a large number of small Hessian eigenvalues (due to floppy oscillations of dangling bonds) such as amorphous solids, we alternatively introduce a control parameter, $-\Delta\lambda_1$ (~ -10 eV/Å^2), to terminate the escape steps when $\lambda_1 < -\Delta\lambda_1$.

Once the minimum Hessian eigenvalue becomes sufficiently negative, the eigenvector-following climb phase performs steepest ascent parallel to the Hessian eigenvector,

$$\mathbf{V}^1 = \begin{bmatrix} \mathbf{v}_1^1 \\ \vdots \\ \mathbf{v}_N^1 \end{bmatrix} \in \mathbb{R}^{3N}, \tag{8}$$

corresponding to λ_1 and steepest descent perpendicular to it [4,11,12]:

$$\mathbf{R} \leftarrow \mathbf{R} - \frac{\delta\tau^2}{2\langle m\rangle}(\mathbf{V}^1\mathbf{V}^{1^{\rm T}})\mathbf{F} + \frac{\delta\tau^2}{2\langle m\rangle}(\mathbf{I} - \mathbf{V}^1\mathbf{V}^{1^{\rm T}})\mathbf{F}, \tag{9}$$

where $\mathbf{I}$ is the $3N$ by $3N$ identity matrix, and $\mathbf{V}^{(1)}$ is normalized as

$$\|\mathbf{V}^1\| = \left(\sum_{i=1}^{N}\|\mathbf{v}_i^1\|^2\right)^{1/2} = 1. \tag{10}$$

At a transition state, the forces are zero, whereas the energy takes a minimum value for all directions except for $\mathbf{V}^1$, along which the energy is instead maximum. Thus the eigenvector-following climb, through steepest ascent parallel to $\mathbf{V}^1$ and

Table 2
Pathfinder algorithm to search for concerted events with low activation barriers

```
**Algorithm** pathfinder
**Input**:
  $\mathbf{R}^{\text{init}} \in \mathbb{R}^{3N}$: an initial local minimum-energy state
  $\{\sigma(k) \mid k = 1, \ldots, N_{\text{seed}}\}$: a set of $N_{\text{seed}}$ elementary event seeds
**Output**:
  $\{e(k) \mid k = 1, \ldots, N_{\text{elite}}\}$: a set of $N_{\text{elite}}$ events with the lowest activation barriers

**Steps**:
1. *Elementary (singly-excited) event generation*
  for $k = 1$ to $N_{\text{seed}}$
    call event_generator: $e(k) \leftarrow G(\sigma(k))$
  $N_{\text{event}} \leftarrow N_{\text{seed}}$

2. *Multiply-excited event generation*
  for *excitation* $= 2$ to *Max_excitation*
    $N_{\text{combination}} \leftarrow 0$
    for $\forall(\sigma(k), \sigma(l))(k, l \in [1, N_{\text{event}}]; k < l)$
      $\sigma \leftarrow \sigma(k) \cup \sigma(l)$ // composite event seed as a union
      if $m(\sigma) =$ *excitation*
         $N_{\text{combination}} \leftarrow N_{\text{combination}} + 1$
         $\sigma(N_{\text{event}} + N_{\text{combination}}) \leftarrow \sigma$
         $b_{\text{estimate}}(\sigma(N_{\text{event}} + N_{\text{combination}})) \leftarrow b(e(k)) + b(e(l))$ // estimated barrier energy
    sort $\sigma(N_{\text{event}} + 1 : N_{\text{event}} + N_{\text{combination}})$ in ascending order of $b_{\text{estimate}}$
    for $k = N_{\text{event}} + 1$ to $N_{\text{event}} + \min(N_{\text{combination}}, N_{\text{add\_event}})$ // generate only $N_{\text{add\_event}}$ new events
        call event_generator: $e(k) \leftarrow G(\sigma(k))$
        $b(e(k)) \leftarrow V(\mathbf{R}^{\text{tst}}) - V(\mathbf{R}^{\text{init}})$ // actual barrier energy
    $N_{\text{event}} \leftarrow N_{\text{event}} + \min(N_{\text{combination}}, N_{\text{add\_event}})$
    sort $e(1 : N_{\text{event}})$ in ascending order of $b$
    $N_{\text{event}} \leftarrow \min(N_{\text{event}}, N_{\text{elite}})$ // retain only $N_{\text{elite}}$ new events
```

steepest descent perpendicular to it, converges to a transition state. The eigenvector-following climb steps are terminated, when the maximum force component of every atom falls below a prescribed threshold value: $\max_{i\alpha}\{|f_{i\alpha}| \mid i = 1, \ldots, N; \alpha = x, y, z\} < \Delta f$ (~ 0.1 eV/Å).

Once the eigenvector-following climb converges to a transition state $\mathbf{R}^{\text{tst}}$, the state is pushed slightly away from $\mathbf{R}^{\text{init}}$ [6],

$$\mathbf{R} \leftarrow \mathbf{R}^{\text{tst}} + \delta_{\text{os}}(\mathbf{R}^{\text{tst}} - \mathbf{R}^{\text{init}}), \tag{11}$$

where the dimensionless overshoot parameter δ_{os} ($\sim$0.1) is an input parameter to the algorithm. The algorithm then performs steepest-descent steps, Eq. (4), until the maximum force component becomes less than Δf, signifying the convergence to a final local energy-minimum $\mathbf{R}^{\text{fin}}$.

2.2. *Concerted event generation by discrete linear combination of atomistic events (LCAE)*

The event generator in Section 2.1 defines a mapping, $e \leftarrow G(\sigma)$, from seed σ to event e. To systematically search for events with low barrier energies, we introduce a discrete indexing scheme, which allows the use of combinatorial search techniques. For a specific example of the bond-length constraint in Eq. (3), we first define a composite seed σ as a set of distinct atomic pairs, $l(\sigma) = \{p_1, \ldots, p_{m(\sigma)}\}$, along with the bond-length constraints, Eq. (3), on the pairs. Here, the excitation level $m(\sigma)$ of seed σ is defined as the number of atomic pairs, p_i ($i = 1, \ldots, m(\sigma)$), which constitute the seed. An event seed σ is thus indexed uniquely by a set l of distinct atomic pairs. For example, $\{(15, 783), (47, 875), (175, 811)\}$ is a seed of excitation level 3 consisting of atomic pairs $(15, 783)$, $(47, 875)$ and $(175, 811)$, where the atoms are indexed by positive integers. Similarly, an event $e = G(\sigma)$ is indexed according to its seed σ, from which it is generated. A population of events is stored as an array of the event data type that consists of the atomic-pair list of its seed, the triplet of its initial-, transition-, and final-state energies, and other attributes such as the estimated and actual barrier energies. In addition, the atomic configurations of the transition and final states are stored in files.

The pathfinder algorithm in Table 2 generates progressively more complex composite events, starting from a set of elementary event seeds, $\{\sigma(k) \mid k = 1, \ldots, N_{\text{seed}}\}$, which is an input to the algorithm. An example of elementary event seeds for the slow crack-growth problem is a set of bond-stretching event seeds for all pairs of atoms that are within a cut-off radius from a crack tip. The algorithm first generates N_{seed} elementary events from the elementary seeds by calling algorithm event_generator in Table 1: $e(k) \leftarrow G(\sigma(k))$ ($k = 1, \ldots, N_{\text{seed}}$).

In order to construct concerted events from these elementary events, we construct composite event seeds as unions of simpler seeds. Here, a union, $\sigma = \sigma(k) \cup \sigma(l)$, of a seed-pair $(\sigma(k), \sigma(l))$ is defined as the union of their corresponding atomic-pair sets, $l(\sigma(k)) \cup l(\sigma(l))$, along with the bond-length constraints, Eq. (3), on all constituent atomic pairs. The corresponding composite event is generated as

$$e = G(\sigma) = G\big(\sigma(k) \cup \sigma(l)\big). \tag{12}$$

The pathfinder algorithm maintains a population of events, $S = \{e(1), \dots, e(N_{\text{event}})\}$, where $N_{\text{event}} = |S|$ is the number of events in the population. At the beginning of the algorithm, $N_{\text{event}} = N_{\text{seed}}$ and all events are singly excited, i.e. $m(\sigma(k)) = 1$ $(k = 1, \dots, N_{\text{event}})$. The algorithm then loops over excitation levels from 2 to *Max_excitation*, where the control parameter *Max_excitation* specifies the maximum excitation level considered by the algorithm. At each excitation level, all pairs of the events (or their seeds) in S are considered as candidates for creating new composite events by the union operation. A composite event (or its seed σ) is counted as a new event, only if its number of pairs $m(\sigma)$ is equal to the excitation level under consideration and its atomic-pair set $l(\sigma)$ is distinct from those of all the events in S.

In order to prune the combinatorial search space, we first define the estimated barrier energy of a composite event seed, $\sigma = \sigma(k) \cup \sigma(l)$, as $b_{\text{estimate}}(\sigma) = b(e(k)) + b(e(l))$. After enumerating all new composite events (let the number of which be $N_{\text{combination}}$), we sort them in ascending order of b_{estimate}. To avoid combinatorial explosion of the number of events to be considered, we retain only the $N_{\text{add\_event}}$ lowest (estimated) barrier-energy events out of $N_{\text{combination}}$, where $N_{\text{add\_event}}$ is one of the control parameters of the algorithm. The pathfinder algorithm calls Algorithm event_generator in Table 1 to generate events for the $N_{\text{add\_event}}$ new seeds and to estimate their actual barrier energies. We then increment the number of event N_{event} by $N_{\text{add\_event}}$ and sort all the events in ascending order of the actual barrier energy b. We retain only the N_{elite} lowest (actual) barrier-energy events for the next excitation level, where N_{elite} is another control parameter.

3. Parallelization by hybrid task + spatial decompositions (HTSD)

The pathfinder algorithm has been implemented on parallel computers by first assigning different events to separate processors (task decomposition) and then using spatial decomposition within each task for further parallelization. The parallel program is written in Fortran 90 and message passing interface (MPI) [28] languages, in which all processors constitute an overall MPI communicator, MPI_COMM_WORLD, and processors are grouped into different event groups by defining multiple MPI communicators as subsets of MPI_COMM_WORLD. (The MPI communicator construct combines a processor group and a context, in such a way that messages with different contexts are not intermixed.) In our program, each event calculation is assigned a dedicated communicator. One advantage of the hybrid task + spatial decomposition approach [26,27] implemented with MPI communicators is that the program can be easily converted to a hybrid Grid remote procedure call (GridRPC) + MPI program to be run on a Grid of distributed parallel computers, in which the number of processors change dynamically on demand and resources are allocated and tasks are migrated adaptively in response to unexpected faults [29].

The total number of processors is given by $P = M_c \times P_c$, where M_c is the number of communicators and P_c is the number of processors in each communicator. The number of events to be generated at each algorithmic step is typically larger than the number of communicators M_c, and thus communicator $c \in [0, M_c - 1]$ is assigned a set of events $\{k \mid (k-1) \bmod M_c = c\}$. In spatial decomposition within each task [35], the total volume of the system is divided into P_c subsystems of equal volume, and each subsystem is assigned to a processor in an array of P_c processors. To calculate the force on an atom in a subsystem, the coordinates of the atoms in the boundaries of neighbor subsystems are "cached" from the corresponding processors. After updating the atomic positions due to a steepest-descent/ascent procedure, some atoms may have moved out of its subsystem. These atoms are "migrated" to the proper neighbor processors. With the spatial decomposition, the computation scales as N/P_c, while communication scales in proportion to $(N/P_c)^{2/3}$. Tree-based algorithms such as the fast multipole method (FMM) [38] incur an O(log P_c) overhead, which is negligible for coarse-grained $(N/P_c \gg P_c)$ applications [39].

4. Numerical results

Scalability of the parallel pathfinder algorithm has been tested on a cluster of dual-core, dual-processor AMD Opteron (at clock speed 2 GHz) nodes with Myrinet interconnect, with 4 GB of memory per 4-core node. We define the speed of a program as a product of the total number of atoms and search steps executed per second. The speedup is the ratio between the speed of P processors and that of one processor. The parallel efficiency is the speedup divided by P.

First, we have performed a strong-scaling (or fixed problem-size) test to measure the efficiency of task decomposition parallelism. Here, the system is a cracked Al_2O_3 crystal consisting of 1920 atoms, and multiple communicators ($M_c = 32, \dots, 512$) of size $P_c = 1$ explore a large search space. We choose *Max_excitation* $= 2$ and $N_{\text{add\_event}} = N_{\text{elite}} = 512$. The test uses all four cores per node. Fig. 1 shows the speedup of the parallel pathfinder program over that on 32 processors (we normalize the speedup on 32 processors as 32). The measured speedup

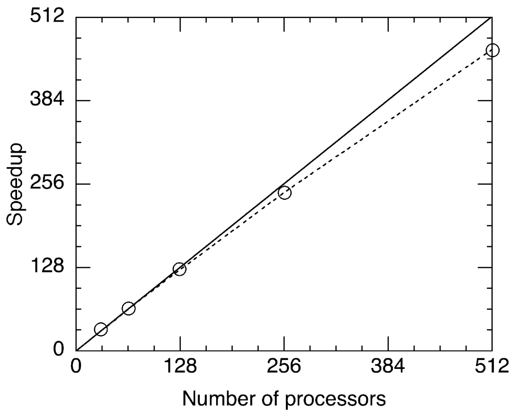

Fig. 1. Strong-scaling (fixed problem-size) speedup of the parallel pathfinder algorithm over 32 processors (normalized so that the speedup is 32 for $P = 32$) as a function of the number of processors P for a 1920-atom cracked Al_2O_3 system on dual-core, dual-processor AMD Opteron nodes. The circles are measured speedups, whereas the solid line denotes the perfect speedup.

on 512 processors is 463.0, and thus the parallel efficiency is 0.904. Although multiple events are generated independently on multiple processors, the parallel algorithm involves sequential bottlenecks such as the sorting of events, and accordingly the parallel efficiency degrades for a larger number of processors.

Next, we have performed a weak-scaling (or isogranular) test to measure the efficiency of spatial decomposition parallelism. In addition to exploring a large number of events for a relatively small number of atoms, the pathfinder program often uses a single communicator to evaluate the barrier energies of a few well-defined events for a larger system. This is the case in multimillion-atom simulations of fracture [40], impact [19], and indentation [20] of materials on a large number of processors P_c. In the weak-scaling test, the number of atoms is scaled linearly with the number of processors. Specifically, we choose $N = 14\,400P_c$, whereas the number of communicators is fixed as $M_c = 1$. Here, we choose $Max\_excitation = 1$ and $N_{add\_event} = N_{elite} = 1$. Fig. 2 shows the total execution and communication times of the parallel pathfinder program on the Opteron cluster for the number of processors $P = 1, \ldots, 512$. (The largest number of atoms is 7 372 800 for $P = 512$.) All four cores per dual-processor, dual-core node are used for the test, except for $P = 1$, where only one core is used. The execution time increases slightly for large P, and the parallel efficiency is 0.764 on 512 processors.

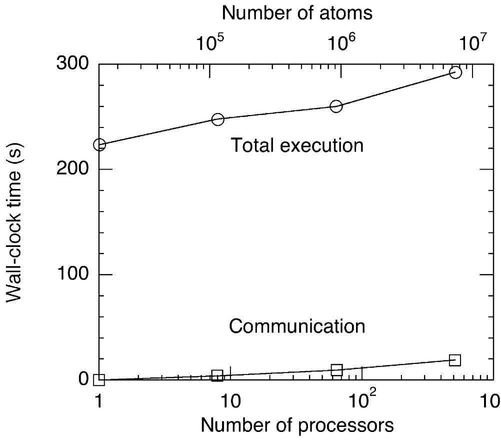

Fig. 2. Weak-scaling (isogranular) test of the parallel pathfinder algorithm on dual-core, dual-processor AMD Opteron nodes. The total execution (circles) and communication (squares) times are plotted as a function of the number of processors P for $14\,400P$-atom Al_2O_3 systems.

The isogranular parallel efficiency is typically used for very large simulations that are performed for a small number of steps. The large granularity, N/P, in such applications makes the parallel efficiency nearly perfect (∼1). For example, we have recently performed benchmark tests including 134 billion-atom space–time multiresolution molecular dynamics (MD) [35], 1.06 billion-atom reactive force-field MD [36], and 11.8 million-atom (1.04 trillion grid points) quantum-mechanical MD in the framework of the divide-and-conquer density functional theory on adaptive multigrids [37], with the parallel efficiency as high as 0.998 on 65 536 dual-processor BlueGene/L processors [41]. We expect the isogranular parallel efficiency of the parallel pathfinder algorithm to become similarly high for such large-scale applications.

To illustrate the use of pathfinder, we simulate a 1920-atom α-crystalline Al_2O_3 with a crack propagating in the $\langle 2\bar{1}\bar{1}0\rangle$ direction in the $\{01\bar{1}0\}$ plane (Fig. 3). The initial state is prepared by first imposing displacements to the atoms according to a linear elastic crack solution corresponding to the stress intensity factor of 1.25 $\mathrm{MPa}\sqrt{\mathrm{m}}$ [3], and then relaxing the atomic configuration to the local energy-minimum, while fixing the positions of the two outer atomic layers in the $\langle 2\bar{1}\bar{1}0\rangle$ and $\langle 01\bar{1}0\rangle$ directions. The periodic boundary condition is applied in the $\langle 0001\rangle$ direction. The simulation uses an interatomic potential consisting of two- and three-body terms, which is similar to those used in previous simulations [19,20,40]. The set of elementary event seeds consists of 43 bonds that are within 2.5 Å from the crack tip. We choose $Max\_excitation = 4$ and $N_{add\_event} = N_{elite} = 128$.

Fig. 3 shows the resulting events with 60 lowest barrier energies, which are a mixture of singly- to quadruply-excited events. Such multiplicity of low activation-barrier events is common in crack growth, which often involves complex events

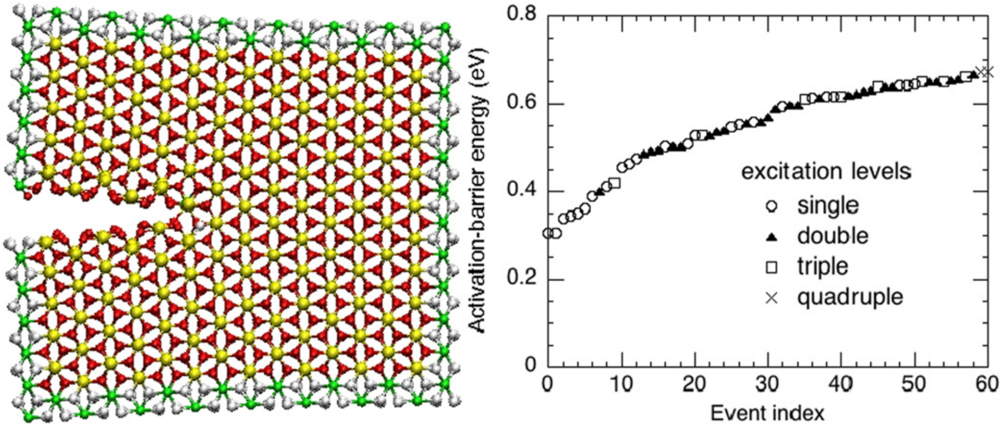

Fig. 3. (Left) The initial state of the 1920-atom cracked Al_2O_3 system, where yellow and red spheres are Al and O atoms, respectively. The positions of the Al (green) and O (grey) atoms at the outer layers are fixed according to a linear-elastic crack solution. (Right) Events with the lowest barrier energies and their excitation levels.

```
**Algorithm** Lanczos
**Input**:
  $\mathbf{R} \in \mathbb{R}^{3N}$: a state
  logical *initialize*: TRUE for the first call in each event generation; FALSE otherwise
**Output**:
  $\lambda_1$: the minimum eigenvalue of the Hessian matrix, $\mathbf{H}(\mathbf{R}) = \partial^2 V/\partial \mathbf{R}^2$
  $\mathbf{V}^1 \in \mathbb{R}^{3N}$: the Hessian eigenvector corresponding to $\lambda_1$
**Steps**:
  if *initialize*
    randomize $\Delta \in \mathbb{R}^{3N}$, such that it contains no translational motion
  $s \leftarrow 0$
  $\beta^s \leftarrow \|\Delta\|$
  $\mathbf{Q}^s\ (\in \mathbb{R}^{3N}) \leftarrow 0$
  do
    $s \leftarrow s+1$
    $\mathbf{Q}^s \leftarrow \Delta/\beta^{s-1}$
    $c_{\text{fd}} \leftarrow \max_{i\alpha}\{|q^s_{i\alpha}| \mid i = 1, \ldots, N; \alpha = x, y, z\}/\delta_{\text{fd}}$
    $\Delta \leftarrow c_{\text{fd}}[-\mathbf{F}(\mathbf{R} + \mathbf{Q}^s/c_{\text{fd}}) + \mathbf{F}(\mathbf{R})] - \beta^{s-1}\mathbf{Q}^{s-1}$
    $\alpha^s \leftarrow \mathbf{Q}^{s\mathrm{T}}\Delta$
    $\Delta \leftarrow \Delta - \alpha^s \mathbf{Q}^s$
    $\beta^s \leftarrow \|\Delta\|$
    diagonalize $\mathbf{T}_s = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{s-2} & \alpha_{s-1} & \beta_{s-1} \\ & & & \beta_{s-1} & \alpha_s \end{bmatrix}$, so that $\tilde{\mathbf{Q}}_s^{\mathrm{T}} \mathbf{T}_s \tilde{\mathbf{Q}}_s = \mathrm{diag}(\tilde{\lambda}_1^s, \ldots, \tilde{\lambda}_s^s)$*
  while $|(\tilde{\lambda}_1^s - \tilde{\lambda}_1^{s-1})/\tilde{\lambda}_1^{s-1}| > \Delta_{\text{eigen}}$
    $\lambda_1 \leftarrow \tilde{\lambda}_1^s$
    $\mathbf{V}^1 \leftarrow \sum_{k=1}^s \mathbf{Q}^k \tilde{q}_k^1$
    $\mathbf{V}^1 \leftarrow \mathbf{V}^1/\|\mathbf{V}^1\|$
```

[*] $\mathrm{diag}(\tilde{\lambda}_1^s, \ldots, \tilde{\lambda}_s^s)$ is an s by s diagonal matrix, with its diagonal elements given by $\tilde{\lambda}_1^s, \ldots, \tilde{\lambda}_s^s$. $\tilde{\mathbf{Q}}^s = [\tilde{\mathbf{q}}^1, \ldots, \tilde{\mathbf{q}}^s]$ is an s by s orthogonal matrix, with $\tilde{\mathbf{q}}^m \in \mathbb{R}^s$ is the mth eigenvector of $\mathbf{T}_s$.

other than individual bond breakings at the crack tip. An example is nanovoid nucleation ahead of the crack tip in glasses, which results from collective atomic motions and long-range stress relaxation [40].

5. Summary

We have designed a search algorithm for escape paths with low activation barriers starting from a local energy minimum configuration of a many-atom system. The pathfinder algorithm combines: (1) a steered eigenvector-following method to generate an escape path tangentially to the eigenvector corresponding to the lowest negative Hessian eigenvalue; (2) systematic combinatorial generation of concerted events as linear combinations of atomistic events; (3) control of population dynamics of low activation-barrier events; and (4) hybrid task + spatial decompositions to implement the algorithm on parallel computers. We have observed reasonable constant problem-size and isogranular parallel efficiencies. The program has been used to study concerted bond-breaking events in the fracture of alumina crystal. The pathfinder algorithm could be combined with other event-population control schemes such as genetic algorithms [21], which could then be used in kinetic Monte Carlo simulations [42,43] that feature on-demand construction of event lists during runtime to explore atomistic mechanisms underlying long-time behavior of materials [44].

Acknowledgements

This work was partially supported by AFOSR—DURINT, ARO—MURI, Chevron—CiSoft, DOE, and NSF. Numerical tests were performed at the University of Southern California using the 5384-processor Linux cluster at the Research Computing Facility and the 2048-processor Linux cluster at the Collaboratory for Advanced Computing and Simulations.

Appendix A. Lanczos algorithm to obtain the minimal Hessian eigenpair

The Lanczos algorithm is used to compute the minimum eigenvalue λ_1 and the corresponding eigenvector $\mathbf{V}^1$ of the Hessian matrix, to be used in the steered eigenvector-following event generator algorithm in Table 1.

References

[1] A.F. Voter, F. Montalenti, T.C. Germann, Annual Review of Materials Research 32 (2002) 321.

[2] D.G. Truhlar, B.C. Garrett, S.J. Klippenstein, Journal of Physical Chemistry 100 (1996) 12771.

[3] B. Lawn, Fracture of Brittle Solids, Cambridge University Press, Cambridge, UK, 1993.

[4] R.A. Olsen, G.J. Kroes, G. Henkelman, et al., Journal of Chemical Physics 121 (2004) 9776.

[5] H.B. Schlegel, Journal of Computational Chemistry 24 (2003) 1514.
[6] G.T. Barkema, N. Mousseau, Physical Review Letters 77 (1996) 4358.
[7] G. Henkelman, H. Jonsson, Journal of Chemical Physics 111 (1999) 7010.
[8] A. Banerjee, N. Adams, J. Simons, et al., Journal of Physical Chemistry 89 (1985) 52.
[9] J. Baker, Journal of Computational Chemistry 7 (1986) 385.
[10] P. Culot, G. Dive, V.H. Nguyen, et al., Theoretica Chimica Acta 82 (1992) 189.
[11] L.J. Munro, D.J. Wales, Physical Review B 59 (1999) 3969.
[12] N. Mousseau, P. Derreumaux, G.T. Barkema, et al., Journal of Molecular Graphics & Modelling 19 (2001) 78.
[13] F.H. Stillinger, Physical Review E 59 (1999) 48.
[14] J.P.K. Doye, D.J. Wales, Journal of Chemical Physics 116 (2002) 3777.
[15] P.J. Feibelman, Physical Review Letters 65 (1990) 729.
[16] G.T. Barkema, N. Mousseau, Physical Review Letters 81 (1998) 1865.
[17] O. Trushin, A. Karim, A. Kara, et al., Physical Review B 72 (2005) 115401.
[18] C. Zhang, B. Bansal, P.S. Branicio, et al., Computer Physics Communications 175 (2006) 339.
[19] P.S. Branicio, R.K. Kalia, A. Nakano, et al., Physical Review Letters 96 (2006) 065502.
[20] I. Szlufarska, A. Nakano, P. Vashishta, Science 309 (2005) 911.
[21] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1997.
[22] K.A. DeJong, Evolutionary Computation, MIT Press, Cambridge, MA, 2002.
[23] A.F. Voter, Physical Review B 57 (1998) R13985.
[24] M. Shirts, V.S. Pande, Science 290 (2000) 1903.
[25] G. Henkelman, H. Jonsson, Physical Review Letters 90 (2003) 116101.
[26] H. Kikuchi, R.K. Kalia, A. Nakano, et al., in: Proceedings of Supercomputing 2002, IEEE/ACM, 2002.
[27] S. Ogata, E. Lidorikis, F. Shimojo, et al., Computer Physics Communications 138 (2001) 143.
[28] W. Gropp, E. Lusk, A. Skjellum, Using MPI, second ed., MIT Press, Cambridge, MA, 1999.
[29] H. Takemiya, Y. Tanaka, S. Sekiguchi, et al., in: Proceedings of Supercomputing 2006, IEEE/ACM, 2006.
[30] D. Wales, Energy Landscapes: Applications to Clusters, Biomolecules and Glasses, Cambridge University Press, Cambridge, UK, 2004.
[31] G.A. Voth, D. Chandler, W.H. Miller, Journal of Chemical Physics 91 (1989) 7749.
[32] We compute the minimum Hessian eigenvalue of the final state to check if it is a local energy minimum.
[33] D.H. Lu, M. Zhao, D.G. Truhlar, Journal of Computational Chemistry 12 (1991) 376.
[34] D.H. Lu, D.G. Truhlar, Journal of Chemical Physics 99 (1993) 2723.
[35] A. Nakano, R.K. Kalia, P. Vashishta, et al., Scientific Programming 10 (2002) 263.
[36] A. Nakano, R.K. Kalia, K. Nomura, et al., Computational Materials Science (2006), in press.
[37] F. Shimojo, R.K. Kalia, A. Nakano, et al., Computer Physics Communications 167 (2005) 151.
[38] L. Greengard, V. Rokhlin, Journal of Computational Physics 73 (1987) 325.
[39] S. Ogata, T.J. Campbell, R.K. Kalia, et al., Computer Physics Communications 153 (2003) 445.
[40] Z. Lu, K. Nomura, A. Sharma, et al., Physical Review Letters 95 (2005) 135501.
[41] A. Nakano, R.K. Kalia, K. Nomura, et al., International Journal of High Performance Computing Applications (2007), accepted for publication.
[42] K.A. Fichthorn, W.H. Weinberg, Journal of Chemical Physics 95 (1991) 1090.
[43] A.F. Voter, in: K.E. Sickafus, E.A. Kotomin, B.P. Uberuaga (Eds.), Radiation Effects in Solids, Springer, Dordrecht, The Netherlands, 2006.
[44] G. Henkelman, H. Jonsson, Journal of Chemical Physics 115 (2001) 9657.

Available online at www.sciencedirect.com

ScienceDirect

Computer Physics Communications 178 (2008) 280–289

Computer Physics Communications

www.elsevier.com/locate/cpc

A space–time-ensemble parallel nudged elastic band algorithm for molecular kinetics simulation

Aiichiro Nakano

Collaboratory for Advanced Computing and Simulations, Department of Computer Science, Department of Physics & Astronomy, Department of Chemical Engineering & Materials Science, University of Southern California, Los Angeles, CA 90089-0242, USA

Received 1 July 2007; received in revised form 9 September 2007; accepted 29 September 2007

Available online 7 October 2007

Abstract

A scalable parallel algorithm has been designed to study long-time dynamics of many-atom systems based on the nudged elastic band method, which performs mutually constrained molecular dynamics simulations for a sequence of atomic configurations (or states) to obtain a minimum energy path between initial and final local minimum-energy states. A directionally heated nudged elastic band method is introduced to search for thermally activated events without the knowledge of final states, which is then applied to an ensemble of bands in a path ensemble method for long-time simulation in the framework of the transition state theory. The resulting molecular kinetics (MK) simulation method is parallelized with a space–time-ensemble parallel nudged elastic band (STEP-NEB) algorithm, which employs spatial decomposition within each state, while temporal parallelism across the states within each band and band-ensemble parallelism are implemented using a hierarchy of communicator constructs in the Message Passing Interface library. The STEP-NEB algorithm exhibits good scalability with respect to spatial, temporal and ensemble decompositions on massively parallel computers. The MK simulation method is used to study low strain-rate deformation of amorphous silica.

© 2007 Elsevier B.V. All rights reserved.

PACS: 02.70.-c; 02.70.Ns; 82.20.Db

Keywords: Nudged elastic band method; Transition state theory; Molecular kinetics simulation; Parallel computing

1. Introduction

Atomistic mechanisms of material processes often involve a sequence of thermally activated events that occur in complex microstructures. For example, various atomistic events, such as bond switching and double-defect recombination, have been postulated to account for plastic flow in amorphous silica [1,2], but they are yet to be characterized quantitatively. Atomistic simulation of slow material processes (e.g., low strain-rate deformation) is challenging because of the existence of unexpected events such as the creation of nanometer-scale voids ahead of the crack tip during fracture of amorphous silica [3,4]. The major computational challenge here is to couple vast spatiotemporal scales for enumerating a sequence of complex atomistic events and accurately evaluating their activation energies, thereby reliably calculating the rate of the material processes.

To address this challenge, a number of methods have been proposed for long-time dynamics simulations [5]. Among these methods, those based on path integrals [6–8] are computationally advantageous for the study of complex material processes, since they offer excellent scalability on parallel computers. These simulation methods use a sequence of atomic configurations (or states) that interpolate initial and final local minimum-energy states to find transition states [9] and estimate the activation energies of the associated events. The scalability of the path-based simulation methods arises from the temporal concurrency along a path [10], in addition to the spatial decomposition parallelism within each state that constitutes the path [11]. One of the widely used path-based simulation methods is the nudged elastic band (NEB) method [12,13], which is employed in this paper.

E-mail address: anakano@usc.edu.

0010-4655/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2007.09.011

For complex material processes, however, final states of events are often unknown, and thus exhaustive search for events is needed [14–16]. In this paper, a slight modification of the NEB method—directionally heated nudged elastic band (DHNEB) method—is used to search for events without the knowledge of final states, where mutually constrained molecular dynamics (MD) simulations with separate temperature controls are performed for a sequence of states in a band. Such unsupervised search for events can be incorporated into kinetic Monte Carlo simulation [17–20] to study long-time processes, where a requisite event list is generated on the fly [21]. We use a path ensemble method (PEM) to enumerate events by concurrently applying the DHNEB method to an ensemble of bands. The resulting molecular kinetics (MK) simulation provides an additional parallelization axis [22,23], i.e. ensemble parallelism, to the spatiotemporal parallelism of path-based simulation. In this paper, we design a space–time-ensemble nudged elastic band (STEP-NEB) algorithm to implement MK simulation on massively parallel computers.

This paper is organized as follows. The next section describes the DHNEB and MK simulation methods, and their scalable parallelization based on the STEP-NEB algorithm is explained in Section 3. Numerical results are presented in Section 4, and Section 5 contains summary.

2. Simulation method

Consider a set of N atoms and its state represented by a $3N$-dimensional vector $\mathbf{R} \in \mathbb{R}^{3N}$ ($\mathbb{R}$ is a set of real numbers), which contains the 3-dimensional positions of the N atoms. Long-time behavior of the system is often studied in the framework of the transition state theory [9], which describes the time evolution of the state in terms of a sequence of thermally activated events. Each event is defined as a triplet of states $(\mathbf{R}_{\text{init}}, \mathbf{R}_{\text{tst}}, \mathbf{R}_{\text{fin}})$ interconnected by a minimum energy path (MEP) $\mathbf{R}(\lambda)$ ($\mathbb{R} \to \mathbb{R}^{3N}$; λ is a real-valued parameter such that $\mathbf{R}_{\text{init}} = \mathbf{R}(\lambda_{\text{init}})$, $\mathbf{R}_{\text{tst}} = \mathbf{R}(\lambda_{\text{tst}})$, and $\mathbf{R}_{\text{fin}} = \mathbf{R}(\lambda_{\text{fin}})$ with $\lambda_{\text{init}} < \lambda_{\text{tst}} < \lambda_{\text{fin}}$). The potential energy function $V(\mathbf{R})$, which describes how atoms interact with each other, is required to be stationary, $\partial V/\partial \mathbf{R} = 0$, at all three states. In addition, the initial ($\mathbf{R}_{\text{init}}$) and final ($\mathbf{R}_{\text{fin}}$) states are local energy minima such that all eigenvalues of the Hessian matrix, $\mathbf{H} = \partial^2 V/\partial \mathbf{R}^2 \in \mathbb{R}^{3N\times 3N}$, are positive, whereas the transition state ($\mathbf{R}_{\text{tst}}$) is a saddle point, at which only the lowest Hessian eigenvalue is negative.

Section 2.1 summarizes the nudged elastic band (NEB) method to approximately obtain a MEP given initial and final states, and Section 2.2 introduces its extension, directionally heated nudged elastic band (DHNEB) method, which searches for an event starting only from an initial state. Section 2.3 describes a path ensemble method (PEM) that applies the DHNEB method to an ensemble of bands to implement molecular kinetics (MK) simulation of long-time processes.

2.1. Nudged elastic band method

Let a nudged elastic band (NEB) be a sequence of S states, $\beta = (\mathbf{R}_0, \ldots, \mathbf{R}_{S-1})$, where the two ends of the band are the initial ($\mathbf{R}_0 = \mathbf{R}_{\text{init}}$) and final ($\mathbf{R}_{S-1} = \mathbf{R}_{\text{fin}}$) states of an event, respectively (see Fig. 1) [12,13]. The band is a discrete approximation to a path $\mathbf{R}(\lambda)$.

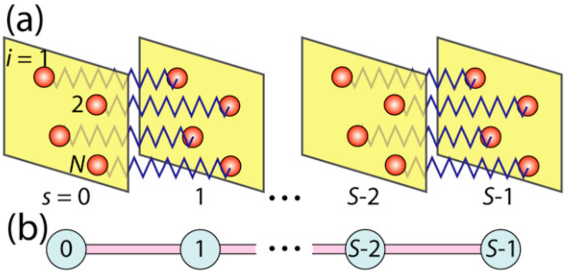

Fig. 1. Schematic of a nudged elastic band. (a) A NEB consists of a sequence of S states (yellow parallelograms), $\mathbf{R}_s$ ($s = 0, \ldots, S-1$), where each state consists of N atoms (red spheres), $i = 1, \ldots, N$. Corresponding atoms in consecutive states interact via harmonic forces represented by blue wavy lines. (b) Abstraction of a NEB consisting of S states (cyan circles) connected by harmonic forces (magenta lines). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The NEB method obtains a MEP iteratively by integrating mutually constrained ordinary differential equations to follow the dynamics of the states:

$$\mathbf{M}\frac{d^2}{dt^2}\mathbf{R}_s = \mathbf{F}_s - \mathbf{M}\gamma_s \frac{d}{dt}\mathbf{R}_s \quad (s = 0, \ldots, S-1), \tag{1}$$

where $\mathbf{M} = \text{diag}(m_1, m_1, m_1, \ldots, m_N, m_N, m_N) \in \mathbb{R}^{3N\times 3N}$ is a diagonal mass matrix (m_i is the mass of the ith atom) and γ_s ($\sim 10^{-2}$ fs^{-1}) is the friction coefficient in the sth state. The force $\mathbf{F}_s$ acting on the sth state is derived from the interatomic potential energy $V(\mathbf{R})$ as

$$\mathbf{F}_s = \begin{cases} -\frac{\partial V}{\partial \mathbf{R}_s}\big|_\perp + \mathbf{F}_s^{\text{spr}}\big|_\parallel & \\ \quad = (\mathbf{I} - \hat{\tau}_s \hat{\tau}_s^{\mathrm{T}})(-\frac{\partial V}{\partial \mathbf{R}_s}) & (1 \leqslant s \leqslant S-2) \\ \qquad + k_{\text{spr}}(\|\tau_s^+\| - \|\tau_s^-\|)\hat{\tau}_s & \\ -\frac{\partial V}{\partial \mathbf{R}_s} & (s = 0, S-1), \end{cases} \tag{2}$$

where $\mathbf{I} \in \mathbb{R}^{3N\times 3N}$ is the identity matrix, $\hat{\tau}_s = \tau_s/\|\tau_s\| \in \mathbb{R}^{3N}$ is a normalized tangential vector along the band ($\|\tau_s\|$ is the norm of vector τ_s), and the superscript T denotes a transpose. The tangential vector τ_s is defined as follows. Let

$$\tau_s^+ = \mathbf{R}_{s+1} - \mathbf{R}_s; \qquad \tau_s^- = \mathbf{R}_s - \mathbf{R}_{s-1} \tag{3}$$

be vectors that connect consecutive images. Then, the tangential vector τ_s is defined as

$$\tau_s = \begin{cases} \text{if} & V(\mathbf{R}_{s+1}) > V(\mathbf{R}_s) > V(\mathbf{R}_{s-1}), \\ & \tau_s^+, \\ \text{else if} & V(\mathbf{R}_{s+1}) < V(\mathbf{R}_s) < V(\mathbf{R}_{s-1}), \\ & \tau_s^-, \\ \text{else if} & V(\mathbf{R}_{s+1}) > V(\mathbf{R}_s), \\ & \tau_s^+ \Delta V_s^{\max} + \tau_s^- \Delta V_s^{\min}, \\ \text{else} & \tau_s^+ \Delta V_s^{\min} + \tau_s^- \Delta V_s^{\max}, \end{cases} \tag{4}$$

where

$$\Delta V_s^{fn} = fn\big(\big|V(\mathbf{R}_{s+1}) - V(\mathbf{R}_s)\big|, \big|V(\mathbf{R}_{s-1}) - V(\mathbf{R}_s)\big|\big) \quad (fn = \max, \min) \tag{5}$$

and the max and min functions return the greater and smaller of the arguments, respectively. In Eq. (2), k_{spr} is a harmonic spring force constant that mutually constrains the motion of consecutive states. The tangential spring forces $\mathbf{F}_s^{\text{spr}}|_\parallel$ thus drive the states $\mathbf{R}_s$ equidistant from each other along the path $\mathbf{R}(\lambda)$, whereas the perpendicular force $-\partial V/\partial\mathbf{R}_s|_\perp$ drives each system toward an energy minimum perpendicular to the path. It has been shown that a band with zero forces, $\mathbf{F}_s = 0$ ($s = 0, \ldots, S-1$), is a discretized approximation to a MEP [12,13]. It should be noted that a number of different definitions have been used for the tangential vectors in the NEB method [24]. In this paper, we adopt the definition given in Ref. [13].

To numerically integrate the second-order ordinary differential equations, Eq. (1), each state $\mathbf{R}_s$ is augmented with corresponding velocities $\mathbf{U}_s = d\mathbf{R}_s/dt \in \mathbb{R}^{3N}$, and accordingly a band is extended to $\beta = ((\mathbf{R}_0, \mathbf{U}_0), \ldots, (\mathbf{R}_{S-1}, \mathbf{U}_{S-1}))$. Integration of Eq. (1) brings the band to a zero-force configuration (hence a MEP) by gradually decelerating the velocities through the frictional forces $-\mathbf{M}\gamma_s\, d\mathbf{R}_s/dt$. For each extended state $(\mathbf{R}_s, \mathbf{U}_s)$, Eq. (1) is equivalent to molecular dynamics (MD) simulation [25], except that the interstate forces $\mathbf{F}_s^{\text{spr}}|_\parallel$ are added and the physical intrastate forces $-\partial V/\partial\mathbf{R}_s$ are projected orthogonal to the path in Eq. (2) for the intermediate ($s = 1, \ldots, S-2$) states. This makes it rather trivial to convert an existing MD program to a NEB program. It is also noteworthy that the dynamics of the initial ($s = 0$) and final ($s = S-1$) ends of the band are not constrained by those of the intermediate states, and thus Eq. (1) brings the initial and final states to physical local minimum-energy configurations even when starting from approximate minima. This is in contrast to the conventional NEB algorithm that does not relax the initial and final ends of the band, which are assumed to be local minimum-energy states.

2.2. Directionally heated nudged elastic band (DHNEB) method

In the standard NEB method described in Section 2.1, the frictional forces $-\mathbf{M}\gamma_s\, d\mathbf{R}_s/dt$ in Eq. (1) are used to quench the band to a zero-force configuration, thereby attaining a MEP, given both initial and final states. Our directionally heated nudged elastic band (DHNEB) method instead generates an event $(\mathbf{R}_{\text{init}}, \mathbf{R}_{\text{tst}}, \mathbf{R}_{\text{fin}})$ starting only from an initial state $\mathbf{R}_{\text{init}}$ (i.e. without the knowledge of a final state $\mathbf{R}_{\text{fin}}$). To do so, the DHNEB method adds a separate heat bath [26] to the MD simulation of each state in the band, so that it is maintained at a desired temperature. For state s, Eq. (1) is thus integrated either: (a) with a heat bath at temperature T_s and zero friction ($\gamma_s = 0$) for thermalization; or (b) without heat bath but with finite friction ($\gamma_s \neq 0$) for quenching.

Given a local minimum-energy state $\mathbf{R}_{\text{init}}$, the DHNEB method starts with an initial band, in which all states $\mathbf{R}_s$ ($s = 0, \ldots, S-1$) are located near $\mathbf{R}_{\text{init}}$. This is achieved by a thermalization phase that duplicates $\mathbf{R}_{\text{init}}$ in all S states $\mathbf{R}_s$, initializes atom velocities $\mathbf{U}_s$ randomly according to the Maxwell–Boltzmann distribution at a common temperature $T_s = T$, and then integrates Eq. (1) for time $t_{\text{therm}} \sim 0.1$ ps, with the force replaced by the physical force $\mathbf{F}_s = -\partial V/\partial\mathbf{R}_s$. Here, no frictional force is applied (i.e. $\gamma_s = 0$), and the temperature of each state is kept at T using a heat bath (see Fig. 2). Note that this is equivalent to performing MD simulations independently for the S states.

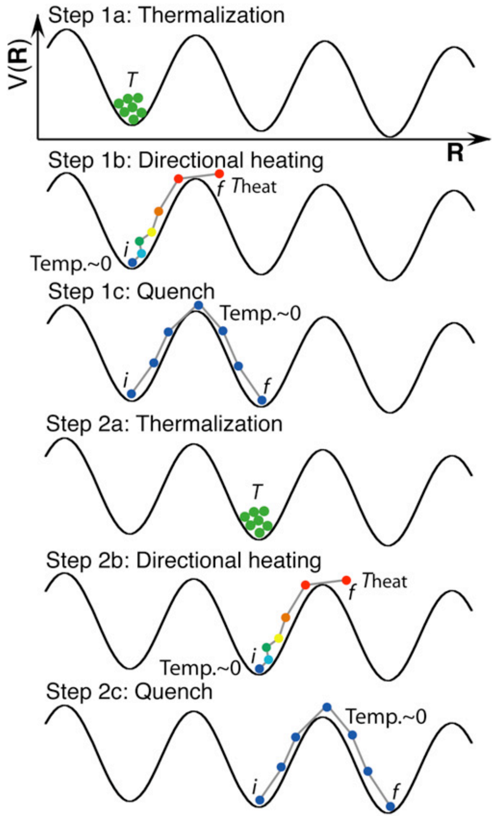

Fig. 2. Algorithmic steps of the directionally heated nudged elastic band method, which consist of thermalization, directional heating, and quench of a band. Black solid curves represent the potential energy surface $V(\mathbf{R})$, whereas circles (with color-coded temperature) are the states interconnected by harmonic forces (gray lines) to form the band. The letters i and f mark the initial and final ends of the band. The figure illustrates two consecutive calls to the DHNEB algorithm, where the final state of the first call is used as the initial state in the second call. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

In the next directional heating phase of the DHNEB method, the final end ($s = S-1$) of the band is heated to $T_{S-1} = T_{\text{heat}}$ ($T_{\text{heat}} \gg T$) using a heat bath, while the initial and intermediate ($s = 0, \ldots, S-2$) states are quenched through the frictional forces using a common friction coefficient $\gamma_s = \gamma$ in Eq. (1) without temperature control. The directional heating allows the final state to escape the concave region (where $\partial^2 V/\partial\mathbf{R}^2 > 0$) of the initial local energy minimum to explore another minimum assisted by high temperature, while the initial state is anchored to $\mathbf{R}_{\text{init}}$ due to the quench. The time duration of the directional heating phase is $t_{\text{heat}} \sim 1$ ps.

After the directional heating locates a new local energy minimum, the quench phase of the algorithm integrates Eq. (1) for $t_{\text{quench}} \sim 2$ ps by turning off the heat bath attached to the final state and applying frictional force $\gamma_{S-1} = \gamma$, so that $\mathbf{R}_{S-1}$ reaches a new local minimum-energy state $\mathbf{R}_{\text{fin}}$. (The initial and intermediate ($s = 0, \ldots, S-2$) states are quenched as well us-

Table 1
Directionally heated nudged elastic band algorithm

```
Algorithm DHNEB
Input:
  $\mathbf{R}_{\text{init}} \in \mathbb{R}^{3N}$: an initial local minimum-energy state
  $T$: temperature at which the system is thermalized
  $T_{\text{heat}}$: temperature to which the final end, $\mathbf{R}_{S-1}$, of the band is heated in the directional heating phase
  $\gamma$: friction coefficient
  $t_{\text{therm}}$: time duration to thermalize the system
  $t_{\text{heat}}$: time duration to directionally heat the band
  $t_{\text{quench}}$: time duration to quench the band
Output:
  event $e = (\mathbf{R}_{\text{init}}, \mathbf{R}_{\text{tst}}, \mathbf{R}_{\text{fin}})$: a triplet of initial, transition and final states
Variable:
  $\beta = ((\mathbf{R}_0, \mathbf{U}_0), \ldots, (\mathbf{R}_{S-1}, \mathbf{U}_{S-1}))$: a nudged elastic band, where $\mathbf{R}_s$ and $\mathbf{U}_s$ are the atomic positions and velocities in the $s$th state
Steps:
  1. initialize band $\beta$ at atomic positions $\mathbf{R}_s \leftarrow \mathbf{R}_{\text{init}}$ with random velocities $\mathbf{U}_s$ according to the Maxwell–Boltzmann distribution at temperature $T$ for all states,
     $s = 0, \ldots, S-1$
  2. thermalization: call NEBDyn($\beta$, $t_{\text{therm}}$, $-\partial V/\partial \mathbf{R}_s$, $T_s = T$ for $s \in [0, S-1]$)
  3. directional heating: call NEBDyn($\beta$, $t_{\text{heat}}$, $\mathbf{F}_s$, $\gamma_s = \gamma$ for $s \in [0, S-2]$ and $T_{S-1} = T_{\text{heat}}$)
  4. quench: call NEBDyn($\beta$, $t_{\text{quench}}$, $\mathbf{F}_s$, $\gamma_s = \gamma$ for $s \in [0, S-1]$)
  5. $\mathbf{R}_{\text{fin}} \leftarrow \mathbf{R}_{S-1}$; $\mathbf{R}_{\text{tst}} \leftarrow \arg\min(V(\mathbf{R}_s))$ $(s = 0, \ldots, S-1)$
```

Table 2
Nudged elastic band dynamics algorithm

```
Algorithm NEBDyn
Input:
  initial band $\beta = ((\mathbf{R}_0, \mathbf{U}_0), \ldots, (\mathbf{R}_{S-1}, \mathbf{U}_{S-1}))$
  $t$: simulated time duration
  $\tilde{\mathbf{F}}_s = \mathbf{F}_s$ or $-\partial V/\partial \mathbf{R}_s$ $(s = 0, \ldots, S-1)$: constrained forces in Eq. (2)
     or physical forces
  $T_s$ or $\gamma_s$ $(s = 0, \ldots, S-1)$: desired temperatures or friction coefficients
Output:
  updated band $\beta = ((\mathbf{R}_0, \mathbf{U}_0), \ldots, (\mathbf{R}_{S-1}, \mathbf{U}_{S-1}))$
Steps:
  for $\forall s \in [0, S-1]$ with a desired temperature $T_s$
     augment the state $(\mathbf{R}_s, \mathbf{U}_s)$ with a heat-bath degree of freedom
     with temperature $T_s$
  compute initial forces $\tilde{\mathbf{F}}_s(\{\mathbf{R}_s | s = 0, \ldots, S-1\})$ for all states,
     $s = 0, \ldots, S-1$
  do $step = 1$ to $\lfloor t/\Delta t \rfloor$ // $\Delta t$ is the time discretization unit
     $\mathbf{U}_s \leftarrow \mathbf{U}_s + (\Delta t/2)\mathbf{M}^{-1}\tilde{\mathbf{F}}_s$ $(s = 0, \ldots, S-1)$ // update velocities
     $\mathbf{R}_s \leftarrow \mathbf{R}_s + \Delta t \mathbf{U}_s$ $(s = 0, \ldots, S-1)$ // update atomic positions
     update forces $\tilde{\mathbf{F}}_s(\{\mathbf{R}_s | s = 0, \ldots, S-1\})$ for all states, $s = 0, \ldots, S-1$
     $\mathbf{U}_s \leftarrow \mathbf{U}_s + (\Delta t/2)\mathbf{M}^{-1}\tilde{\mathbf{F}}_s$ $(s = 0, \ldots, S-1)$ // update velocities
     for $\forall s \in [0, S-1]$ with a friction coefficient $\gamma_s$
        $\mathbf{U}_s \leftarrow \exp(-\gamma_s \Delta t)\mathbf{U}_s$ // damp velocities
```

ing the same friction coefficient $\gamma_s = \gamma$.) The quench phase of the DHNEB algorithm embodies the standard NEB method, and thus the band converges to a MEP connecting the original and new local minima. The transition state $\mathbf{R}_{\text{tst}}$ of the corresponding event is determined as the state with the highest potential energy, i.e. $\mathbf{R}_{\text{tst}} \leftarrow \arg\min(V(\mathbf{R}_s))$ $(s = 0, \ldots, S-1)$. It should be noted that this is an approximate transition state due to the discrete approximation to the MEP. The approximation can be systematically improved by increasing the number of states S in the band or by using the climbing image NEB method [27].

Table 1 shows the DHNEB algorithm that finds an event $e = (\mathbf{R}_{\text{init}}, \mathbf{R}_{\text{tst}}, \mathbf{R}_{\text{fin}})$, given an initial local minimum-energy state $\mathbf{R}_{\text{init}}$. The thermalization, directional heating, and quench phases of the DHNEB algorithm are implemented by calling the NEB dynamics algorithm, NEBDyn, in Table 2, which integrates Eq. (1) numerically with time discretization unit Δt ($\sim$1 fs) using a symplectic, reversible time integrator [28]. The inputs to the NEBDyn algorithm are: (a) initial band augmented with velocities, $\beta = ((\mathbf{R}_0, \mathbf{U}_0), \ldots, (\mathbf{R}_{S-1}, \mathbf{U}_{S-1}))$; (b) simulated time duration t, for which Eq. (1) is integrated; (c) forces—either the mutually constrained forces $\mathbf{F}_s$ in Eq. (2) or the physical forces $-\partial V/\partial \mathbf{R}_s (s = 0, \ldots, S-1)$; and (d) either a friction coefficient γ_s or a desired temperature T_s for each state $s \in [0, S-1]$. The output from the NEBDyn algorithm is the updated band β after the time integration for t. The DHNEB algorithm can be applied repeatedly to search for the next event by starting a new thermalization–directional heating–quench cycle, using the previous $\mathbf{R}_{\text{fin}}$ as a new $\mathbf{R}_{\text{init}}$. Fig. 2 illustrates two thermalization–directional heating–quench cycles of the DHNEB algorithm.

As a method to explore the energy landscape, the DHNEB algorithm is equivalent to other simulation methods such as simulated annealing [29] and temperature-accelerated dynamics [30]. An advantage of the DHNEB algorithm is the integration of activation-barrier estimation into event search, which makes the programming straightforward when a massive number of events and their activation barriers need to be enumerated. The quality of the obtained energy landscape by all these methods can be improved by increasing the size of the statistical ensemble, which will be addressed in the next subsection.

2.3. Path ensemble method (PEM) for molecular kinetics (MK) simulation

The DHNEB method is a stochastic simulation method that finds an event assisted by temperature, starting from a given initial state. For many-atom systems, however, a large number of events may exist for each initial state [31,32]. To con-

Table 3
Molecular kinetics simulation algorithm

```
Algorithm MK
Input:
    $\mathbf{R}_{\text{init}} \in \mathbb{R}^{3N}$: an initial local minimum-energy state
    $N_{\text{step}}$: total number of simulation steps
    $T$: temperature
    $T_{\text{heat}}$: temperature for directional heating in the DHNEB algorithm in Table 1
    $t_{\text{therm}}$: time duration to thermalize the system
    $t_{\text{heat}}$: time duration to directionally heat the band
Output:
    $(e_*^{(1)}, e_*^{(2)}, \ldots, e_*^{(N_{\text{step}})})$: a sequence of events that are selected according to Eq. (7), one at a simulation step
    $(t_*^{(1)}, t_*^{(2)}, \ldots, t_*^{(N_{\text{step}})})$: a sequence of times when the events occur
Variable:
    $E = \{\beta_b \mid b = 0, \ldots, B-1\}$: ensemble of $B$ nudged elastic bands
    $t$: simulated time
Steps:
    $t \leftarrow 0$
    do $l = 1$ to $N_{\text{step}}$
        for $\forall \beta_b \in E$
            $e_b = (\mathbf{R}_{\text{init}}^{(b)}, \mathbf{R}_{\text{tst}}^{(b)}, \mathbf{R}_{\text{fin}}^{(b)}) \leftarrow \text{DHNEB}(\mathbf{R}_{\text{init}})$ // generate an event by calling the DHNEB algorithm
            $\Delta_b \leftarrow V(\mathbf{R}_{\text{tst}}^{(b)}) - V(\mathbf{R}_{\text{init}}^{(b)})$ // calculate the activation energy
            $r_b = \{t_{\text{therm}} + t_{\text{heat}} \exp[\frac{\Delta_b}{k_B}(\frac{1}{T} - \frac{1}{T_{\text{heat}}})]\}^{-1}$ // estimate the reaction rate of event $b$
        $r \leftarrow \sum_{b=0}^{B-1} r_b$
        for $\forall \beta_b \in E$
            $P_b \leftarrow r_b / r$ // calculate the probability that event $b$ occurs
        select an event $b^* \in [0, B-1]$ according to probability distribution $P_b$
        $e_*^{(l)} \leftarrow e_{b^*}$
        $t \leftarrow t - \ln(\xi)/r$, where $\xi$ is a uniform random number in the range [0, 1]
        $t_*^{(l)} \leftarrow t$
        $\mathbf{R}_{\text{init}} \leftarrow \mathbf{R}_{\text{fin}}^{(b^*)}$
```

struct a list of multiple events, we introduce a path ensemble method (PEM), which applies the DHNEB method to an ensemble of B bands, $E = \{\beta_b \mid b = 0, \ldots, B-1\}$, starting from a common initial state $\mathbf{R}_{\text{init}}$ but with different random velocities. This generates a list of possible events $\{e_b = (\mathbf{R}_{\text{init}}^{(b)}, \mathbf{R}_{\text{tst}}^{(b)}, \mathbf{R}_{\text{fin}}^{(b)}) \mid b = 0, \ldots, B-1\}$ with the associated barrier energies $\{\Delta_b = V(\mathbf{R}_{\text{tst}}^{(b)}) - V(\mathbf{R}_{\text{init}}^{(b)}) \mid b = 0, \ldots, B-1\}$. In the framework of the transition state theory [9], we calculate the rate of the bth event as [30]

$$r_b = \left\{ t_{\text{therm}} + t_{\text{heat}} \exp\left[\frac{\Delta_b}{k_B} \left(\frac{1}{T} - \frac{1}{T_{\text{heat}}} \right) \right] \right\}^{-1}, \tag{6}$$

where t_{therm} and t_{heat} are the time durations of the thermalization and directional heating phases of the DHNEB method, respectively, k_B is the Boltzmann constant, and T and T_{heat} are the temperatures used for the thermalization and directional heating phases of the DHNEB method, respectively (see Section 2.2).

In kinetic Monte Carlo simulation [17–20], these rates are used to numerically integrate the master equation that governs transitions among local minimum-energy states to describe the long-time dynamics of the system. Specifically, we select one event b^* from the list of possible events according to probability

$$P_b = \frac{r_b}{r} = \frac{r_b}{\sum_{b=0}^{B-1} r_b}, \tag{7}$$

and the state is changed to the corresponding final state $\mathbf{R}_{\text{fin}}^{(b^*)}$. According to the probability density, $p(t_{\text{evt}}) = r\exp(-rt_{\text{evt}})$, of the time span between two successive events in a Poisson process, the simulated time is advanced by

$$t_{\text{evt}} = -\ln(\xi)/r, \tag{8}$$

where ξ is a uniform random number in the range [0, 1]. This procedure is repeated by using $\mathbf{R}_{\text{fin}}^{(b^*)}$ as the initial state $\mathbf{R}_{\text{init}}$ in the next simulation step.

Table 3 shows the resulting molecular kinetics (MK) simulation algorithm to study long-time processes. It starts with a given local minimum-energy state $\mathbf{R}_{\text{init}}$, and returns a sequence of events $(e_*^{(1)}, e_*^{(2)}, \ldots, e_*^{(N_{\text{step}})})$, where N_{step} is the total number of MK simulation steps. At the lth simulation step, an event $e_*^{(l)}$ is selected according to the probability, Eq. (7). In addition, the algorithm returns the sequence of times $(t_*^{(1)}, t_*^{(2)}, \ldots, t_*^{(N_{\text{step}})})$, at which these events occur.

3. Space–time-ensemble parallel nudged elastic band (STEP-NEB) algorithm

The MK simulation method is implemented on parallel computers based on a space–time-ensemble parallel nudged elastic band (STEP-NEB) algorithm. It first assigns the B bands in the ensemble to separate groups of processors (ensemble decomposition), each of which is in turn decomposed into S processor

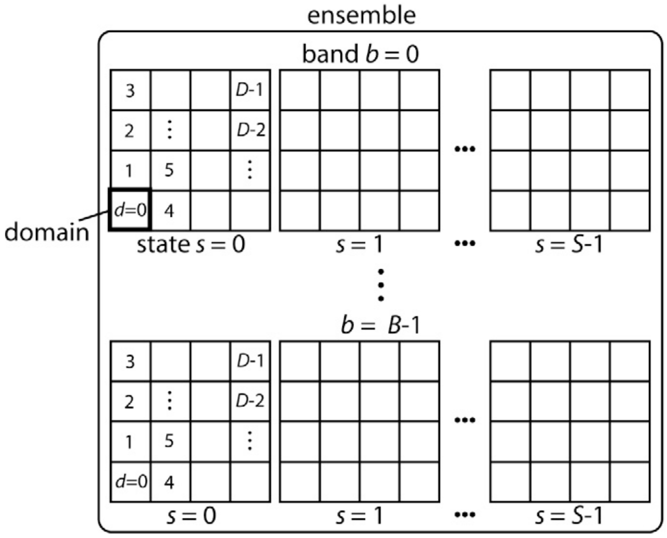

Fig. 3. Processor indexing in the space–time-ensemble parallel nudged elastic band algorithm. Ensemble decomposition first divides the processors into B groups $comm_b$ ($b = 0, \ldots, B-1$), each assigned a band in the ensemble. Temporal decomposition in turn divides each $comm_b$ into S subgroups $comm_{b,s}$ ($s = 0, \ldots, S-1$), each for a state in band b. Spatial decomposition then divides each state into D spatial domains.

subgroups (temporal decomposition) representing the S states that constitute each band (see Fig. 3). We then use spatial decomposition within each state for further parallelization.

The parallel program is written in Fortran 90 and Message Passing Interface (MPI) [33] languages, in which all processors constitute an overall MPI communicator, MPI_COMM_WORLD. (The MPI communicator construct combines a processor group and a context, in such a way that messages with different contexts are not intermixed.) The processors are first grouped into B processor groups, each consisting of SD processors, by defining B MPI communicators $comm_b$ ($b = 0, \ldots, B-1$) as subsets of MPI_COMM_WORLD. The communicator $comm_b$, which performs the computation associated with the bth band, is in turn grouped into S sub-communicators $comm_{b,s}$ ($s = 0, \ldots, S-1$), each consisting of D processors. The communicator $comm_{b,s}$ performs parallel MD simulation for the sth state in the bth band using D processors. Due to this hierarchical decomposition (see Fig. 3), the total number of processors in the STEP-NEB algorithm is given by

$$P = BSD. \tag{9}$$

One advantage of the hybrid task (ensemble and time) + spatial decomposition approach [34] implemented with MPI communicators is that the program can be easily converted to a hybrid Grid remote procedure call (GridRPC) + MPI program to be run on a Grid of distributed parallel computers, in which the number of processors changes dynamically on demand, resources are allocated adaptively, and tasks are migrated automatically in response to unexpected faults [35].

We employ spatial decomposition to parallelize the computation within each $comm_{b,s}$ [36]. Here, the total volume of the simulated system is divided into D domains of equal volume, and each domain is assigned to a processor in an array of D processors. Specifically, we use regular 3-dimensional mesh topology that maps atom i at position $\mathbf{r}_i = (r_{ix}, r_{iy}, r_{iz})$ to processor $d(\mathbf{r}_i)$ in an array of $D = D_x D_y D_z$ processors:

$$\begin{cases} d(\mathbf{r}_i) = d_x(r_{ix}) D_y D_z + d_y(r_{iy}) D_z + d_z(r_{iz}), \\ d_\alpha(r_{i\alpha}) = \lfloor r_{i\alpha} D_\alpha / L_\alpha \rfloor \quad (\alpha = x, y, z), \end{cases} \tag{10}$$

where L_α is the simulation box size in the α direction ($\alpha = x, y, z$). The P processors are globally indexed in such a way that the dth domain of the sth state in the bth band is assigned to processor

$$p = bSD + sD + d. \tag{11}$$

Fig. 4 shows the resulting tree structure of the processor organization in the space–time-ensemble parallelism.

To calculate the intrastate force $-\partial V/\partial \mathbf{R}_s$ acting on an atom in a spatial domain, the coordinates of the atoms in the boundaries of neighbor domains are "cached" from the corresponding processors within the same communicator $comm_{b,s}$ [36]. After updating the atomic positions due to time stepping, some atoms may have moved out of its domain. These atoms are "migrated" to the proper neighbor processors within the same $\mathrm{comm}_{b,s}$ [36]. With the spatial decomposition, the computational cost scales as N/D, while communication scales in proportion to $(N/D)^{2/3}$. For long-range interatomic potentials used in MD simulations, tree-based algorithms such as the fast multipole method (FMM) [37] incur an $O(\log D)$ overhead, which is negligible for coarse-grained ($N/D \gg D$) applications [38].

To calculate the interstate force $\mathbf{F}_s^{\mathrm{spr}}$ acting on the ith atom in the dth domain of the sth state, we need to know the position of the same atom in states $s-1$ and $s+1$ within the same band b. However, these corresponding atoms may not reside in the same spatial domain d, since the atoms migrate among processors in the spatial decomposition scheme. Under an assumption that the distance between any corresponding atoms between consecutive states is bounded by a cutoff distance r_c, we search for the corresponding atoms in the consecutive states with the aid of global indexing of the N atoms as follows. In the first step, we augment the set of atomic positions, $A_{b,s,d} = \{\mathbf{r}_i \mid d(\mathbf{r}_i) = d\}$, that reside in each spatial subsystem d of state s in band b, by caching atoms that are in the nearest-neighbor domains in the same state s but are within distance r_c from domain d: $A'_{b,s,d} = A_{b,s,d} \cup \{\mathbf{r}_i \mid (d(\mathbf{r}_i) \neq d) \wedge (\|\mathbf{r}_i - \partial d\| < r_c)\}$, where ∂d is the 2D boundary that encloses the 3D spatial domain d. Each processor $p = bSD + sD + d$ then copies the augmented atomic positions, $A'_{b,s-1,d}$ and $A'_{b,s+1,d}$, of the same spatial domain d in the lower ($s-1$) and upper ($s+1$) states from processors $p_- = bSD + (s-1)D + d$ and $p_+ = bSD + (s+1)D + d$, respectively. (Note that the copying is not required for the initial ($s = 0$) and final ($s = S-1$) states, since there is no interstate force on them.) For each atom i in $A_{b,s,d}$, we search for the corresponding atoms in $A'_{b,s-1,d}$ and $A'_{b,s+1,d}$ by way of its global index. Using the linked-list cell method, the computational time for the search is $O(N/D)$ [36]. The communication cost of the temporal decomposition is $O(N/D)$ for copying $A'_{b,s-1,d}$ and $A'_{b,s+1,d}$.

Ensemble decomposition duplicates the above band calculation, each involving SD processors, B times on $P = BSD$

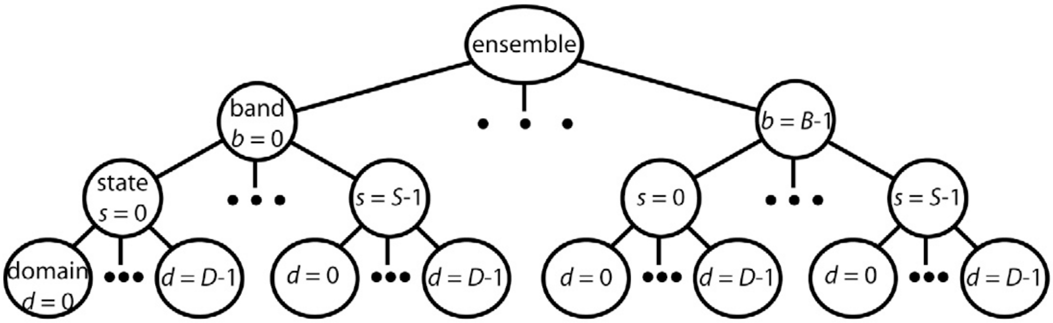

Fig. 4. Tree-structured processor organization in the hierarchical space–time-ensemble parallelization. An ensemble consists of B bands, each consisting of S states. Each state in turn contains D spatial domains.

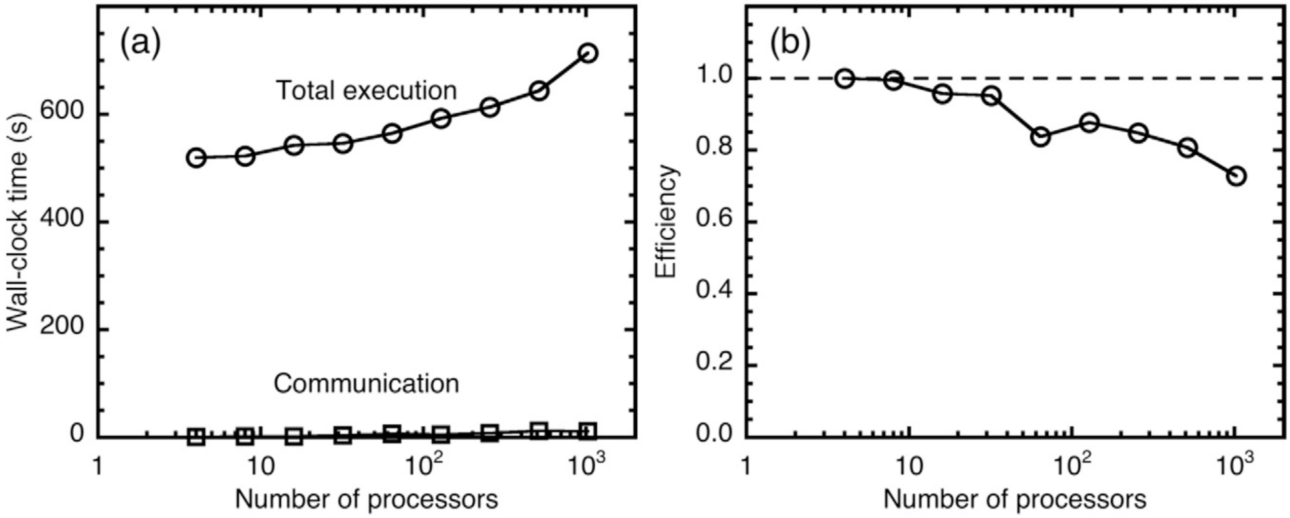

Fig. 5. Weak-scaling (isogranular) test of spatial decomposition in the STEP-NEB algorithm on dual-core, dual-processor AMD Opteron nodes. The total execution (circles) and communication (squares) times (a), as well as the parallel efficiency (b), are plotted as a function of the number of processors $P = BSD$ for $648{,}000D$-atom $\mathrm{SiO_2}$ systems, where $B = 1$, $S = 4$, and $D = 1$–256.

processors. It involves $O((N/D)\log(BS))$ overhead to multicast the new initial state $\mathbf{R}_{\text{init}}$ among processors assigned the same spatial domain, i.e. those with the same p mod D. The multicast cost at the beginning of each MK simulation step is greatly amortized over 10^3–10^4 MD steps in the DHNEB method per MK iteration.

4. Numerical results

4.1. Scalability on parallel computers

Scalability of the STEP-NEB algorithm is tested on a cluster of dual-core, dual-processor AMD Opteron (at clock speed 2 GHz) nodes with Myrinet interconnect, with 4 GB of memory per 4-core node. We define the speed of a program as a product of the total number of atoms and MK simulation steps executed per second. The speedup is the ratio between the speed of P processors and that of one processor. The parallel efficiency is the speedup divided by P.

First, we perform a scalability test associated with spatial decomposition. Here, we fix the number of bands in the ensemble $B = 1$ and that of states per band $S = 4$, and vary the number of spatial domains per state D from 1 to 256 (i.e. the total number of processors $P = BSD = 4$–1024). In our weak-scaling (or isogranularity) test, the number of atoms is scaled linearly with the number of spatial domains: $N = 648{,}000D$. We set $N_{\text{step}} = 1$ and $t_{\text{therm}} = t_{\text{heat}} = t_{\text{quench}} = 10\Delta t$. All the numerical tests in this paper are performed for silica ($\mathrm{SiO_2}$) material, using a many-body interatomic potential [39]. The isogranular scaling test corresponds to a situation, in which the MK program is used to evaluate the barrier energies of a few well-defined events for a large system. This is the case in million-to-billion atom simulations of fracture [40], impact [41], and indentation [42] of materials on a large number of processors D. Fig. 5(a) shows the total execution and communication times per MK simulation step of the STEP-NEB program on the Opteron cluster for the number of processors $P = 4, \ldots, 1024$. (The largest number of atoms per state is 165,888,000 for $D = 256$.) All four cores per dual-processor, dual-core Opteron node are used for the test. The execution time increases only slightly for large P, showing excellent scalability. Fig. 5(b) shows the parallel efficiency as a function of the number of processors for the same test; it is 0.728 on 1024 processors.

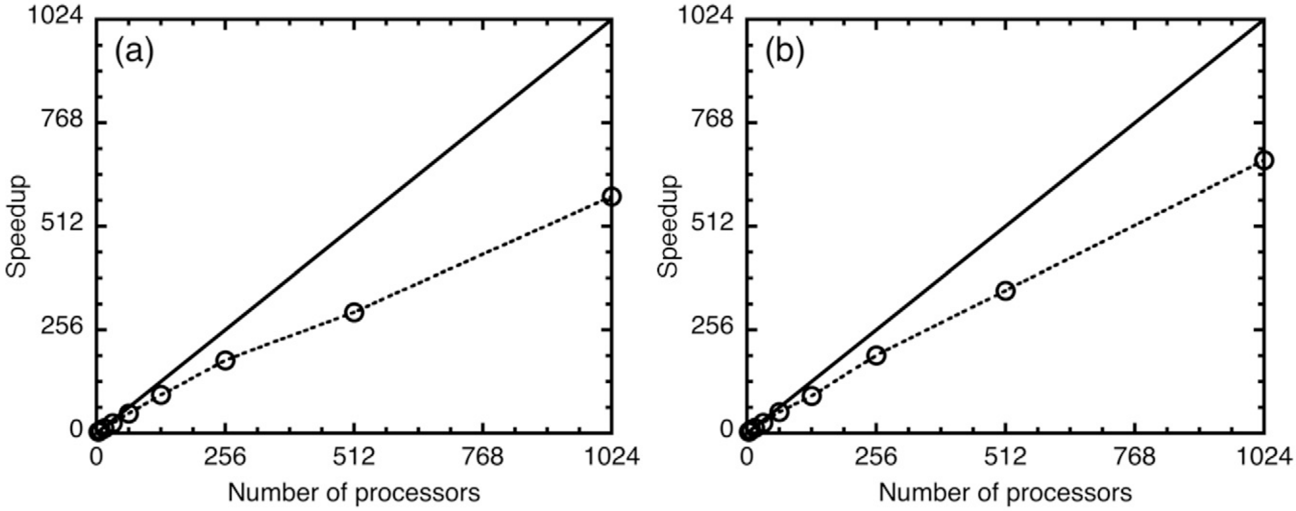

Fig. 6. (a) Speedup of temporal decomposition in the STEP-NEB algorithm (normalized so that the speedup is 4 for $P = 4$) as a function of the number of processors P ($P = 4$–1024) for a 192-atom amorphous SiO_2 system on dual-core, dual-processor AMD Opteron nodes, where we fix $B = D = 1$. The circles are measured speedups, whereas the solid line denotes the perfect speedup. (b) Speedup of ensemble decomposition in the STEP-NEB algorithm as a function of the number of processors P ($= 4, \ldots, 1024$) for silica material ($N = 192$ atoms). Here, we fix the number of states per band $S = 4$ and the number of spatial domains per state $D = 1$, while the number of bands is varied from $B = 1$ to 256.

The isogranular parallel efficiency is typically used for very large simulations, and their large granularity, N/D, makes the parallel efficiency nearly perfect ($\sim$1). For example, we have recently performed benchmark tests including 134 billion-atom space–time multiresolution MD [36], 1.06 billion-atom chemically reactive force-field MD [43], and 11.8 million-atom (1.04 trillion electronic degrees-of-freedom) quantum-mechanical MD in the framework of the divide-and-conquer density functional theory on adaptive multigrids [44], with the parallel efficiency as high as 0.998 on 131,072 BlueGene/L processors [45]. We expect the isogranular parallel efficiency of the STEP-NEB algorithm to become similarly high for such large-scale applications.

Next, we test the scalability of temporal decomposition, where we fix the number of bands $B = 1$ and the number of domains per state $D = 1$. We vary the number of states per band $S = 4$ to 1024. Here, the simulated system is amorphous SiO_2 consisting of $N = 192$ atoms, and we set $N_{\text{step}} = 1$ and $t_{\text{therm}} = t_{\text{heat}} = t_{\text{quench}} = 20\Delta t$. The test uses all four cores per node. Fig. 6(a) shows the speedup of the STEP-NEB program (we normalize the speedup on 4 processors as 4). The measured speedup on 1024 processors is 586.3, and thus the parallel efficiency is 0.573.

The relatively low efficiency shown in Fig. 6(a) is partly due to the small granularity, $N/D = 192$, as well as the non-dedicated operation of the Linux cluster, on which we have observed the reduction of parallel efficiency by $\sim$0.2 on 1024 processors for MD simulation [45]. In addition, the efficiency is expected to be higher in production runs, where the number of MD steps per MK simulation step, $N_{\text{MD}}/N_{\text{step}} = (t_{\text{therm}} + t_{\text{heat}} + t_{\text{quench}})/(\Delta t N_{\text{step}})$, is well over 10^3 (see Section 4.2), compared with 60 in this test. In fact, the parallel efficiency on 1024 processors decreases from 0.573 to 0.450, when $N_{\text{MD}}/N_{\text{step}}$ is further reduced from 60 to 30. With constant B and N/D, the algorithm has a small $O(\log S)$ overhead for multicasting the new initial state at every MK step, and this multicast cost is amortized over $N_{\text{MD}}/N_{\text{step}}$ steps. Hence, a larger number of MD steps per MK step leads to higher efficiency.

Finally, we test the scalability of ensemble decomposition, where we fix the number of states per band $S = 4$ and the number of spatial domains per state $D = 1$. The number of bands per ensemble is varied from $B = 1$ to 256. The simulated system is amorphous SiO_2 consisting of $N = 192$ atoms, and we set $N_{\text{step}} = 1$ and $t_{\text{therm}} = t_{\text{heat}} = t_{\text{quench}} = 20\Delta t$. Although multiple events are generated independently by different processor groups, the parallel algorithm involves sequential bottlenecks such as the selection of an event that occurs, and accordingly the parallel efficiency degrades for a larger number of processors. Fig. 6(b) shows the speedup of the STEP-NEB program on the Opteron cluster as a function of the number of processors (normalized to be 4 on 4 processors). On 1024 processors, the measured speedup is 675.6, and thus the parallel efficiency of ensemble decomposition is 0.660, which is slightly higher than that of temporal decomposition on the same number of processors.

4.2. *Molecular kinetics simulation of low strain-rate deformation of amorphous silica*

We use the MK simulation method to study low strain-rate deformation of amorphous silica (a-SiO_2) at temperature T and shear strain rate $\dot{\varepsilon}$. To do this, we slightly extend the MK simulation algorithm in Table 3: at the end of each MK step, we increment the shear strain imposed on the system by $\dot{\varepsilon} t_{\text{evt}}$, where t_{evt} is defined in Eq. (8). Here, the shear is applied by deforming the MD simulation box tensor [46].

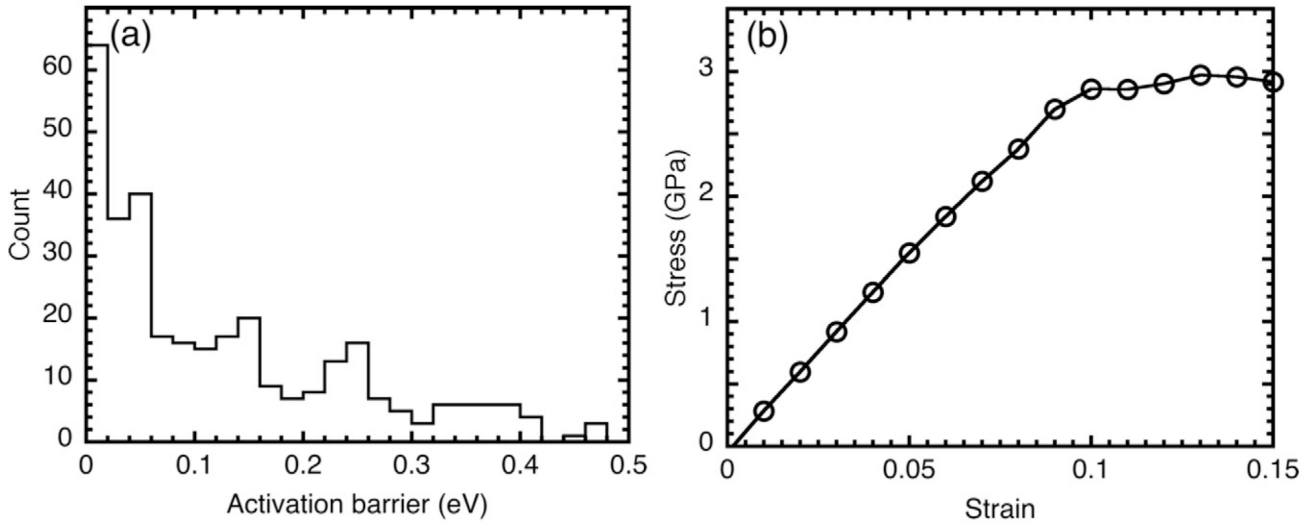

Fig. 7. Molecular kinetics simulation of amorphous silica at strain rate $10^6\ \mathrm{s}^{-1}$ and temperature 300 K. (a) Histogram of activation barriers for the events at strain between 0.1 and 0.15. (b) Stress-strain relation of the same simulation.

We first prepare an initial a-SiO_2 state containing 192 atoms by a melt-quench procedure, with the periodic boundary condition applied to all directions [47,48]. We then perform MK simulations at strain rate $\dot{\varepsilon} = 10^6\ \mathrm{s}^{-1}$, using the number of bands per ensemble $B = 8$, the number of states per band $S = 16$, and the number of spatial domains per state $D = 1$. We set the temperatures $T = 300$ K and $T_{\mathrm{heat}} = 900$ K, and the MD simulation times $t_{\mathrm{therm}} = 0.1$ ps, $t_{\mathrm{heat}} = 1$ ps, and $t_{\mathrm{quench}} = 2$ ps. Though we have not introduced explicit convergence criteria to stop the NEB minimization procedure, the quenching phase of the DHNEB algorithm is equivalent to 2000 steepest-descent quench steps of the standard NEB method (with a unit time step of 1 fs), with which forces are typically converged within 10^{-2} eV/Å. For taking statistical averages, multiple MK simulations are performed with the same parameters but different random number sequences. Fig. 7(a) shows a histogram of the activation barriers of the events that are selected according to the probability distribution, Eq. (7), during the simulation (only events for strain range $[0.1, 0.15]$ are included). The histogram demonstrates the existence of many low activation-barrier events in a-SiO_2, which makes the application of accelerated MD simulation techniques [5] to this system difficult.

Fig. 7(b) shows the calculated stress-strain relation. It is an average over eight MK simulation runs with different random number sequences. Such calculation will be useful for the study of yield stress as a function of the strain rate $\dot{\varepsilon}$, when the rate is less than $10^6\ \mathrm{s}^{-1}$. In contrast, MD simulations are usually applicable only to higher strain rates above $\dot{\varepsilon} = 10^9\ \mathrm{s}^{-1}$.

To estimate the error of the activation barriers in Fig. 7, we have used a quasi Newton method [49] to obtain the fully converged transition state starting from each approximate transition state (i.e. the state with the highest energy in the band). The estimated error is typically 10^{-3} eV/atom, which will not change the results in Fig. 7. However, other quantities such as the lowest Hessian eigenvalue at the approximate transition state have larger uncertainties. The lowest Hessian eigenvalue has an error on the order of 10%, which necessitates the refinement step for the computation of this quantity.

5. Summary

We have designed a scalable parallel algorithm to study long-time dynamics of many-atom systems. A directionally heated nudged elastic band method searches for thermally activated events without the knowledge of final states, which is then applied to an ensemble of bands in a path ensemble method in the framework of the transition state theory. We have parallelized the resulting molecular kinetics simulation method using a space–time-ensemble parallel nudged elastic band algorithm, which employs spatial decomposition within each state, while temporal parallelism across the states within each band and band-ensemble parallelism are implemented using a hierarchy of communicator constructs in the Message Passing Interface library. The STEP-NEB algorithm has exhibited good scalability with respect to spatial, temporal and ensemble decompositions on a massively parallel computer. Finally, we have demonstrated the use of MK simulation for low strain-rate deformation of amorphous silica.

MK simulation of a physical phenomenon produces a massive catalogue of complex atomistic events, and understanding of the phenomenon requires the elucidation of the catalogued events. This necessitates automated (or computerized) identification of primary events as well as their annotation and classification. Such “computational thinking” will be greatly facilitated by discrete abstraction [11,50], such as graph-theory based data mining [51,52], combined with creative massive-date visualization [53] techniques.

Acknowledgements

This work was partially supported by ARO—MURI, DOE—SciDAC, DTRA, and NSF—ITR/PetaApps. Numerical tests were performed at the University of Southern California using the 5472-processor Linux cluster at the Research Computing Facility and the 2048-processor Linux cluster at the Collaboratory for Advanced Computing and Simulations.

References

[1] R.H. Doremus, Journal of Applied Physics 92 (2002) 7619–7629.

[2] N.F. Mott, Philosophical Magazine B: Physics of Condensed Matter Statistical Mechanics Electronic Optical and Magnetic Properties 56 (1987) 257–262.

[3] F. Celarie, S. Prades, D. Bonamy, L. Ferrero, E. Bouchaud, C. Guillot, C. Marliere, Physical Review Letters 90 (2003) 075504.

[4] R.K. Kalia, A. Nakano, P. Vashishta, C.L. Rountree, L. Van Brutzel, S. Ogata, International Journal of Fracture 121 (2003) 71–79.

[5] A.F. Voter, F. Montalenti, T.C. Germann, Annual Review of Materials Research 32 (2002) 321–346.

[6] R. Olender, R. Elber, Journal of Chemical Physics 105 (1996) 9299–9315.

[7] P.G. Bolhuis, D. Chandler, C. Dellago, P.L. Geissler, Annual Review of Physical Chemistry 53 (2002) 291–318.

[8] D. Passerone, M. Parrinello, Physical Review Letters 87 (2001) 108302.

[9] D.G. Truhlar, B.C. Garrett, S.J. Klippenstein, Journal of Physical Chemistry 100 (1996) 12771–12800.

[10] V. Zaloj, R. Elber, Computer Physics Communications 128 (2000) 118–127.

[11] A. Nakano, Computer Physics Communications 176 (2007) 292–299.

[12] H. Jonsson, G. Mills, K. Jacobsen, Classical and Quantum Mechanics in Condensed Phase Simulations, World Scientific, Singapore, 1998.

[13] G. Henkelman, H. Jonsson, Journal of Chemical Physics 113 (2000) 9978–9985.

[14] G.T. Barkema, N. Mousseau, Physical Review Letters 77 (1996) 4358–4361.

[15] L.J. Munro, D.J. Wales, Physical Review B 59 (1999) 3969–3980.

[16] R.A. Olsen, G.J. Kroes, G. Henkelman, A. Arnaldsson, H. Jonsson, Journal of Chemical Physics 121 (2004) 9776–9792.

[17] A.B. Bortz, M.H. Kalos, J.L. Lebowitz, Journal of Computational Physics 17 (1975) 10–18.

[18] D.T. Gillespie, Journal of Computational Physics 22 (1976) 403–434.

[19] K.A. Fichthorn, W.H. Weinberg, Journal of Chemical Physics 95 (1991) 1090–1096.

[20] A.F. Voter, in: K.E. Sickafus, E.A. Kotomin, B.P. Uberuaga (Eds.), Radiation Effects in Solids, Springer, Dordrecht, The Netherlands, 2006.

[21] G. Henkelman, H. Jonsson, Journal of Chemical Physics 115 (2001) 9657–9666.

[22] A.F. Voter, Physical Review B 57 (1998) R13985–R13988.

[23] M.R. Shirts, V.S. Pande, Physical Review Letters 86 (2001) 4983–4987.

[24] N. Gonzalez-Garcia, J.Z. Pu, A. Gonzalez-Lafont, J.M. Lluch, D.G. Truhlar, Journal of Chemical Theory and Computation 2 (2006) 895–904.

[25] A. Rahman, Physical Review 136 (1964) A405–A411.

[26] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, Oxford University Press, Oxford, UK, 1987.

[27] G. Henkelman, B.P. Uberuaga, H. Jonsson, Journal of Chemical Physics 113 (2000) 9901–9904.

[28] G.J. Martyna, M.E. Tuckerman, D.J. Tobias, M.L. Klein, Molecular Physics 87 (1996) 1117–1157.

[29] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Science 220 (1983) 671–680.

[30] M.R. Sorensen, A.F. Voter, Journal of Chemical Physics 112 (2000) 9599–9606.

[31] F.H. Stillinger, Physical Review E 59 (1999) 48–51.

[32] J.P.K. Doye, D.J. Wales, Journal of Chemical Physics 116 (2002) 3777–3788.

[33] W. Gropp, E. Lusk, A. Skjellum, Using MPI, second ed., MIT Press, Cambridge, 1999.

[34] S. Ogata, E. Lidorikis, F. Shimojo, A. Nakano, P. Vashishta, R.K. Kalia, Computer Physics Communications 138 (2001) 143–154.

[35] H. Takemiya, Y. Tanaka, S. Sekiguchi, S. Ogata, R.K. Kalia, A. Nakano, P. Vashishta, in: Proceedings of Supercomputing 2006, IEEE/ACM, 2006.

[36] A. Nakano, R.K. Kalia, P. Vashishta, T.J. Campbell, S. Ogata, F. Shimojo, S. Saini, Scientific Programming 10 (2002) 263–270.

[37] L. Greengard, V. Rokhlin, Journal of Computational Physics 73 (1987) 325–348.

[38] S. Ogata, T.J. Campbell, R.K. Kalia, A. Nakano, P. Vashishta, S. Vemparala, Computer Physics Communications 153 (2003) 445–461.

[39] P. Vashishta, R.K. Kalia, J.P. Rino, I. Ebbsjo, Physical Review B 41 (1990) 12197–12209.

[40] Z. Lu, K. Nomura, A. Sharma, W.Q. Wang, C. Zhang, A. Nakano, R. Kalia, P. Vashishta, E. Bouchaud, C. Rountree, Physical Review Letters 95 (2005) 135501.

[41] P.S. Branicio, R.K. Kalia, A. Nakano, P. Vashishta, Physical Review Letters 96 (2006) 065502.

[42] I. Szlufarska, A. Nakano, P. Vashishta, Science 309 (2005) 911–914.

[43] A. Nakano, R.K. Kalia, K. Nomura, A. Sharma, P. Vashishta, F. Shimojo, A.C.T. van Duin, I.W.A. Goddard, R. Biswas, D. Srivastava, Computational Materials Science 38 (2007) 642–652.

[44] F. Shimojo, R.K. Kalia, A. Nakano, P. Vashishta, Computer Physics Communications 167 (2005) 151–164.

[45] A. Nakano, R.K. Kalia, K. Nomura, A. Sharma, P. Vashishta, F. Shimojo, A.C.T. van Duin, I.W.A. Goddard, R. Biswas, D. Srivastava, L.H. Yang, International Journal of High Performance Computing Applications (2007), in press.

[46] M. Parrinello, A. Rahman, Journal of Applied Physics 52 (1981) 7182–7190.

[47] J.P. Rino, I. Ebbsjo, R.K. Kalia, A. Nakano, P. Vashishta, Physical Review B 47 (1993) 3053–3062.

[48] A. Nakano, R.K. Kalia, P. Vashishta, Physical Review Letters 73 (1994) 2336–2339.

[49] H.B. Schlegel, Journal of Computational Chemistry 24 (2003) 1514–1527.

[50] G.T. Barkema, N. Mousseau, Physical Review Letters 81 (1998) 1865–1868.

[51] D.J. Cook, L.B. Holder, Mining Graph Data, John Wiley & Sons, Hoboken, NJ, 2007.

[52] C. Zhang, B. Bansal, P.S. Branicio, R.K. Kalia, A. Nakano, A. Sharma, P. Vashishta, Computer Physics Communications 175 (2006) 339–347.

[53] A. Sharma, A. Nakano, R.K. Kalia, P. Vashishta, S. Kodiyalam, P. Miller, W. Zhao, X.L. Liu, T.J. Campbell, A. Haas, Presence-Teleoperators and Virtual Environments 12 (2003) 85–95.