

Multiscale Newton-Krylov Transition-State Finder: A Quantum-Mechanical/Molecular-Mechanical Algorithm

Aiichiro Nakano,^{a,1} Rajiv K. Kalia,^a Ken-ichi Nomura,^a Fuyuki Shimojo,^{a,b} Priya Vashishta,^a and Weiqiang Wang^a

^a*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*

^b*Department of Physics, Kumamoto University, Kumamoto 860-8555, Japan*

Abstract

An algorithm has been designed for finding transition states with quantum-mechanical (QM) accuracy in a system containing thousands of atoms. The multiscale Newton-Krylov transition-state finder (MNK-TSF) algorithm consists of doubly nested loops. The outer Newton iteration is driven by accurate QM forces in the framework of the density functional theory (DFT), combined with approximate inversion of a Hessian matrix using a less compute-intensive molecular-mechanical (MM) method. The inner Krylov-subspace iteration inverts the MM Hessian matrix in a Jacobian-free manner using the generalized minimal residual method. Furthermore, a QM correction to the MM Hessian matrix is included within a rank-two approximation. The MNK-TSF algorithm has been implemented on parallel computers using hybrid task + spatial decomposition. The algorithm has been used to find transition states for hydrolysis reactions in amorphous silica using a hierarchy of simulation methods. Here, a classical MM force field and a chemically reactive force field have been used in sequence to explore a large solution space and to approximately locate a reaction path, whereas a limited number of DFT-based QM force calculations have been invoked to refine the path.

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

Keywords: Transition state theory; Multiscale simulation; Molecular mechanics; Quantum mechanics; Density functional theory; Jacobian-free Newton Krylov method; Parallel computing

¹*Email address:* anakano@usc.edu (A. Nakano)

1. Introduction

Mechanical processes in materials are often fundamentally influenced by chemical reactions.¹ An example of such mechanochemical processes is stress corrosion cracking, in which corrosive molecules at the crack tip modify the crack growth rate as a function of the stress intensity factor.² However, inseparable coupling of bond-breaking events (the description of which requires an accurate quantum-mechanical method) and long-range stress fields (which requires the consideration of a large number of atoms) makes the computation highly demanding. A promising approach for coupling quantum-mechanical (QM) accuracy with large-scale materials modeling is multiscale simulation,³⁻⁶ in which atomistic molecular dynamics (MD) simulations using force fields of varying accuracy and computational costs—from classical molecular-mechanical (MM) force fields⁷ to chemically reactive force fields (ReaxFF)⁸—invoke accurate QM simulations in the framework of the density functional theory (DFT)⁹⁻¹¹ within a small spatial subsystem.

A major drawback of such a traditional spatial QM/MD approach³⁻⁶ is that the time steps of its QM and MD simulations are identical, so that the accessible time scale is severely limited by the large number of compute-intensive QM force evaluations required for long-time simulations. This paper instead develops a spatiotemporal multiscale simulation scheme,¹² in which accurate QM simulations correct MD solutions only at sparse spatiotemporal points.¹³ This is accomplished in the context of the transition state theory to describe rare events that dictate long-time materials behavior.^{14, 15} After identifying an atomistic event approximately by MD simulation, our multiscale Newton-Krylov transition-state finder (MNK-TSF) algorithm finds its transition state in doubly nested loops that combine QM and MD methods. The outer Newton iteration¹⁶⁻²³ is driven by accurate QM forces, combined with approximate inversion of a Hessian matrix using a less compute-intensive MD method. The inner Krylov-subspace iteration²⁴ inverts the MM Hessian matrix in a Jacobian-free manner using the generalized minimal residual (GMRES) method.^{25, 26} Furthermore, a QM correction to the MM Hessian matrix is included within a rank-two approximation of Broyden family.^{16, 17, 23}

Advanced parallel and distributed computing technologies are also expected to facilitate the quantum-mechanically accurate simulation of large spatiotemporal-scale processes. Multiscale QM/MD simulations have been parallelized with a hybrid task + spatial decomposition approach,^{5, 27} using the communicator construct in the message passing interface (MPI)

language.²⁸ This approach is a natural migration path to hybrid Grid remote procedure call (GridRPC) + MPI programming on a Grid of geographically distributed parallel computers.²⁹

We have tested the MNK-TSF algorithm for finding transition states for hydrolysis reactions in amorphous silica, using a hierarchy of simulation methods. First, a classical MM force field and a ReaxFF have been used in sequence to explore a large solution space and to approximately locate a reaction path, whereas a limited number of DFT-based QM force calculations have been invoked to refine the path.

This paper is organized as follows. The next section describes the MNK-TSF algorithm, and its parallelization is discussed in Section 3. Numerical results are presented in Section 4, and Section 5 contains summary.

2. Multiscale Newton-Krylov transition-state finder (MNK-TSF) algorithm

Consider a system of N atoms with its state specified by a $3N$ -dimensional vector $\mathbf{R} \in \mathbb{R}^{3N}$ (\mathbb{R} is a set of real numbers). The forces \mathbf{F} ($\in \mathbb{R}^{3N}$) on the atoms are computed from the potential energy function $V(\mathbf{R})$ as $\mathbf{F} = -\partial V / \partial \mathbf{R}$, and the Hessian matrix is defined as $\mathbf{H} = \partial^2 V / \partial \mathbf{R}^2 \in \mathbb{R}^{3N \times 3N}$. The key computational problem in the transition state theory¹⁴ is to find a transition state \mathbf{R}^{tst} , at which $\mathbf{F}(\mathbf{R}^{\text{tst}}) = 0$ and at which only the lowest eigenvalue λ_1 of the Hessian matrix is negative.

This section presents a multiscale Newton-Krylov transition-state finder (MNK-TSF) algorithm for finding \mathbf{R}^{tst} . First, Section 2.1 describes how quantum-mechanical (QM) and molecular-dynamics (MD) calculations are combined in the outer loop of MNK-TSF. Section 2.2 then discusses its inner loop based on a Jacobian-free generalized minimum residual (JF-GMRES) algorithm for determining atomic displacements from residual QM forces. Finally, Section 2.3 explains the use of MD calculation to obtain an initial guess for the transition state, based on a steered eigenvector-following (SEF) algorithm.

2.1. Multiscale Newton-Raphson algorithm for finding a transition state

The Newton-Raphson method¹⁶⁻²³ obtains a stationary point, at which $\mathbf{F} = 0$, through successive quadratic approximations of the potential energy,

$$V(\mathbf{R} + \Delta) = V(\mathbf{R}) - \mathbf{F}(\mathbf{R})^T \Delta + \frac{1}{2} \Delta^T \mathbf{H}(\mathbf{R}) \Delta + O(\|\Delta\|^3), \quad (1)$$

for small atomic displacements $\Delta \in \mathbb{R}^{3N}$ (the superscript T denotes a transpose). The stationary relation, $F(\mathbf{R} + \Delta) = -\partial V(\mathbf{R} + \Delta)/\partial \Delta = 0$, imposed on Eq. (1) is solved for the atomic displacements as

$$\Delta = \mathbf{H}^{-1}(\mathbf{R})\mathbf{F}(\mathbf{R}). \quad (2)$$

Accordingly, the Newton-Raphson method successively updates the atomic positions by iterating Newton steps,

$$\mathbf{R} \leftarrow \mathbf{R} + \mathbf{H}^{-1}(\mathbf{R})\mathbf{F}(\mathbf{R}), \quad (3)$$

until the norm of the forces, $\|\mathbf{F}\|$, falls below a prescribed threshold $F_{\text{threshold}}$.

To reduce the computational cost, we propose to use accurate QM forces $\mathbf{F}_{\text{QM}}(\mathbf{R})$ in the framework of the density functional theory^{9, 10} (DFT) only to drive the Newton step, Eq. (3), while performing the compute-intensive Hessian inversion to determine the system's response (i.e., atomic displacements) with a less compute-intensive MD method. The resulting multiscale QM/MD Newton step is expressed as

$$\mathbf{R} \leftarrow \mathbf{R} + \mathbf{H}_{\text{MD}}^{-1}(\mathbf{R})\mathbf{F}_{\text{QM}}(\mathbf{R}), \quad (4)$$

where \mathbf{H}_{MD} is the Hessian matrix computed with the MD potential energy V_{MD} ,⁷ and the QM forces \mathbf{F}_{QM} are computed with the QM potential energy V_{QM} .¹¹

The simple multiscale QM/MD Newton step, Eq. (4), involves two sources of errors: 1) the modeling error, i.e., the discrepancy between the QM and MD potential energies; and 2) the truncation error in the quadratic approximation of the potential energy in Eq. (1). Furthermore, the iterative inversion of the MD Hessian matrix in Eq. (4) is usually truncated within an error tolerance. These errors manifest themselves in the error vector $\Phi \in \mathbb{R}^{3N}$, which is defined in terms of Newton displacements Δ as

$$\Phi = \mathbf{F}_{\text{QM}}(\mathbf{R} + \Delta) - \mathbf{F}_{\text{QM}}(\mathbf{R}) - \mathbf{H}_{\text{MD}}\Delta. \quad (5)$$

In quasi-Newton methods,^{16, 17, 23} this error information is utilized to correct the Hessian matrix to be used in the next Newton iteration. We employ the Powel symmetric Broyden (PSB) method, which reduces the error with a rank-2 matrix constructed from two vectors Φ and Δ :^{16, 17, 23}

$$\mathbf{H}_{\text{QM/MD}} = \mathbf{H}_{\text{MD}} + \delta\mathbf{H}_{\text{QM/MD}} = \mathbf{H}_{\text{MD}} + \frac{\Phi\Delta^T + \Delta\Phi^T}{\|\Delta\|} - \frac{\Phi^T\Delta}{\|\Delta\|^2}\Delta\Delta^T. \quad (6)$$

Accordingly, Eq. (4) is corrected to be a multiscale MD/QM quasi-Newton step defined as

$$\mathbf{R} \leftarrow \mathbf{R} + \left(\mathbf{H}_{\text{MD}}(\mathbf{R}) + \frac{\Phi \Delta^T + \Delta \Phi^T}{\|\Delta\|} - \frac{\Phi^T \Delta}{\|\Delta\|^2} \Delta \Delta^T \right)^{-1} \mathbf{F}_{\text{QM}}(\mathbf{R}). \quad (7)$$

The resulting MNK-TSF algorithm consists of doubly nested loops (see Table 1). Its outer loop iterates the Newton steps, Eq. (7), using the accurate QM forces as a driving force, whereas the inner loop (in algorithm JF-GMRES) multiplies the inverse of the MD Hessian matrix to the QM force vector by using less compute-intensive MD forces in an iterative manner.

Table 1

Multiscale Newton-Krylov transition-state finder (MNK-TSF) algorithm.

Algorithm MNK-TSF

Input:

$\mathbf{R}_0^{\text{tst}} \in \mathbb{R}^{3N}$: an initial guess for the transition state

Output:

$\mathbf{R}^{\text{tst}} \in \mathbb{R}^{3N}$: a transition state with QM accuracy

Steps:

$\mathbf{R} \leftarrow \mathbf{R}_0^{\text{tst}}$

compute $\mathbf{F}_{\text{QM}}(\mathbf{R})$

$\Delta / \|\Delta\| (\in \mathbb{R}^{3N}) \leftarrow 0$

$\Phi (\in \mathbb{R}^{3N}) \leftarrow 0$

do

call JF-GMRES: $\Delta \leftarrow [\mathbf{H}_{\text{MD}}(\mathbf{R}) + (\Phi \Delta^T + \Delta \Phi^T) / \|\Delta\| - (\Phi^T \Delta / \|\Delta\|^2) \Delta \Delta^T]^{-1} \mathbf{F}_{\text{QM}}(\mathbf{R})$

$c_{\text{fd}} \leftarrow \max_i \{ |\Delta_i| \mid i = 1, K, 3N \} / \delta_{\text{fd}}$

$\Phi \leftarrow \mathbf{F}_{\text{QM}}(\mathbf{R} + \Delta) - \mathbf{F}_{\text{QM}}(\mathbf{R}) - c_{\text{fd}} [-\mathbf{F}_{\text{MD}}(\mathbf{R} + \Delta / c_{\text{fd}}) + \mathbf{F}_{\text{MD}}(\mathbf{R})]$

$\mathbf{R} \leftarrow \mathbf{R} + \Delta$

while $\|\mathbf{F}_{\text{QM}}(\mathbf{R})\| > F_{\text{threshold}}$

$\mathbf{R}^{\text{tst}} \leftarrow \mathbf{R}$

To reduce the computational cost, the Hessian matrix is computed in a Jacobian-free manner²⁴ in algorithm MNK-TSF. Namely, we use a finite-difference method to evaluate the product of the Hessian matrix \mathbf{H} and vector Δ ,³⁰

$$\mathbf{H}_{\text{MD}}(\mathbf{R})\Delta = c_{\text{fd}} [-\mathbf{F}_{\text{MD}}(\mathbf{R} + \Delta / c_{\text{fd}}) + \mathbf{F}_{\text{MD}}(\mathbf{R})], \quad (8)$$

so that only the forces but not the Hessian matrix need to be computed. In Eq. (8), $c_{\text{fd}} = \max_i \{ |\Delta_i| \mid i = 1, \dots, 3N \} / \delta_{\text{fd}}$ and $\delta_{\text{fd}} (\sim 10^{-2} \text{ \AA})$ is a discretization unit for finite differencing.

We use various divide-and-conquer algorithms to compute the forces in Eq. (8) in $O(N)$ time. For example, a space-time multiresolution molecular dynamics (MRMD) algorithm⁷ and a fast

reactive force-field (F-ReaxFF) algorithm⁸ are used in cases of classical interatomic potentials and semi-classical reactive force fields, respectively. To compute the QM forces in Eq. (5) from the Hellmann-Feynman theorem, we use our $O(N)$ embedded divide-and-conquer density functional theory (EDC-DFT) algorithm.¹¹ Consequently, the computational complexity of the MNK-TSF algorithm is $O(N)$. Note that recent benchmark tests have demonstrated 134 billion-atom MRMD, 1.01 billion-atom F-ReaxFF, and 11.8 million-atom (or 1.04 trillion electronic degrees-of-freedom) EDC-DFT calculations with parallel efficiency as high as 0.998 on 131,072 IBM BlueGene/L processors.³¹

To further reduce the computational cost for evaluating the QM forces in Eq. (5), we optionally employ the traditional spatial QM/MD approach.^{3-6, 32-35} Here, the QM calculation is performed only in small spatial subdomains that are outside the range of validity of the MD approach. Specifically, we use an additive hybridization scheme, in which the QM forces \mathbf{F}_{QM} on the total system in Eq. (5) are replaced by hybrid QM/MD forces,^{5,35}

$$\mathbf{F}(\text{total}) = \mathbf{F}_{MD}(\text{total}) + \sum_{\{\text{domain}\}} [\mathbf{F}_{QM}(\text{domain}) - \mathbf{F}_{MD}(\text{domain})], \quad (9)$$

with QM forces $\mathbf{F}_{QM}(\text{domain})$ evaluated in a set of spatial subdomains using the EDC-DFT algorithm.¹¹ Traditionally, termination atoms are added to the QM and MD domains to minimize artificial boundary effects, but the solution is sensitive to the choice of the termination atoms.⁵ The EDC-DFT algorithm employs a buffer-layer approach to considerably reduce this sensitivity.¹¹ Accordingly, we were able to automate the adaptive domain redefinition during MD simulations.²⁹

2.2. Jacobian-free Krylov subspace algorithm for solving a linear system of equations

In algorithm MNK-TSF in Table 1, a linear system of equations,

$$\left(\mathbf{H}_{MD}(\mathbf{R}) + \frac{\Phi \Delta^T + \Delta \Phi^T}{\|\Delta\|} - \frac{\Phi^T \Delta}{\|\Delta\|^2} \Delta \Delta^T \right) \Delta' = \mathbf{F}_{QM}(\mathbf{R}), \quad (10)$$

needs to be solved to obtain the new atomic displacements Δ' ($\in \mathbb{R}^{3N}$). This is done iteratively using a Krylov subspace method,²⁴⁻²⁶ where an approximate solution to the linear system, $\mathbf{H}_{QM/MD} \Delta' = \mathbf{F}_{QM}$, is obtained by projecting it to an affine subspace, $\Delta_0 + \text{span}\{\Lambda_0, \mathbf{H}\Lambda_0, \mathbf{H}^2\Lambda_0, \mathbf{K}, \mathbf{H}^{M-1}\Lambda_0\}$. Here, Δ_0 ($\in \mathbb{R}^{3N}$) is an initial guess to the solution,

$\Lambda_0 = \mathbf{F}_{QM} - \mathbf{H}\Delta_0$ is the corresponding residual, and *span* denotes a subspace spanned by a set of vectors. The Krylov subspace dimension M ($\ll 3N$) is typically set to ~ 10 . Specifically, we employ the Jacobian-free generalized minimal residual (JF-GMRES) algorithm^{25, 26} in Table 2. Here, the Hessian-vector product is computed again in a Jacobian-free manner as in Eq. (8).

Table 2

Jacobian-free generalized minimal residual (JF-GMRES) algorithm.

Algorithm JF-GMRES

Input:

$\mathbf{R} \in \mathbb{R}^{3N}$: current atomic positions

$\mathbf{F}_{QM} \in \mathbb{R}^{3N}$: quantum-mechanical forces

$\Phi \in \mathbb{R}^{3N}$: error vector

$\Delta \in \mathbb{R}^{3N}$: displacement vector from the previous Newton step

Output:

new displacement vector $\Delta \leftarrow [\mathbf{H}_{MD}(\mathbf{R}) + (\Phi\Delta^T + \Delta\Phi^T)/\|\Delta\| - (\Phi^T\Delta/\|\Delta\|^2)\Delta\Delta^T]^{-1}\mathbf{F}_{QM}$ ($\in \mathbb{R}^{3N}$)

Steps:

$\Delta_0 \leftarrow -(\delta t^2/2\langle m \rangle)(\mathbf{V}_{MD}^1 \mathbf{V}_{MD}^{1^T})\mathbf{F}_{QM} + (\delta t^2/2\langle m \rangle)(\mathbf{I} - \mathbf{V}_{MD}^1 \mathbf{V}_{MD}^{1^T})\mathbf{F}_{QM}$ ($\in \mathbb{R}^{3N}$) // initial guess for the solution

$c_{fd} \leftarrow \max_i\{\|\Delta_{0,i}\| | i = 1, \dots, 3N\}/\delta_{fd}$

$\Lambda_0 \leftarrow \mathbf{F}_{QM} - c_{fd}[-\mathbf{F}_{MD}(\mathbf{R} + \Delta_0/c_{fd}) + \mathbf{F}_{MD}(\mathbf{R})] - (\Phi\Delta^T\Delta_0 + \Delta\Phi^T\Delta_0)/\|\Delta\| + (\Phi^T\Delta/\|\Delta\|^2)\Delta\Delta^T\Delta_0$ ($\in \mathbb{R}^{3N}$)
// residual, $\Lambda_0 = \mathbf{F}_{QM} - \mathbf{H}_{MD/QM}\Delta_0$, of the initial guess

$\beta \leftarrow \|\Lambda_0\|$

$\mathbf{U}_1 \leftarrow \Lambda_0/\beta$ ($\in \mathbb{R}^{3N}$)

\mathbf{H} ($\in \mathbb{R}^{(M+1) \times M}$) = $[h_{ij}] \leftarrow 0$

for $j = 1$ to M

$c_{fd} \leftarrow \max_i\{\|U_{j,i}\| | i = 1, \dots, 3N\}/\delta_{fd}$

$\mathbf{W} \leftarrow c_{fd}[-\mathbf{F}_{MD}(\mathbf{R} + \mathbf{U}_j/c_{fd}) + \mathbf{F}_{MD}(\mathbf{R})] + (\Phi\Delta^T\mathbf{U}_j + \Delta\Phi^T\mathbf{U}_j)/\|\Delta\| - (\Phi^T\Delta/\|\Delta\|^2)\Delta\Delta^T\mathbf{U}_j$ ($\in \mathbb{R}^{3N}$)
// $\mathbf{W} \leftarrow \mathbf{H}_{MD/QM}\mathbf{U}_j$

for $i = 1$ to j

$h_{ij} = \mathbf{W}^T\mathbf{U}_i$

$\mathbf{W} \leftarrow \mathbf{W} - h_{ij}\mathbf{U}_i$

$h_{j+1,j} = \|\mathbf{W}\|$

$\mathbf{U}_{j+1} \leftarrow \mathbf{W}/h_{j+1,j}$

$\mathbf{Y}_* \leftarrow \arg \min_{\mathbf{Y}} \|\beta\mathbf{E}_1 - \bar{\mathbf{H}}\mathbf{Y}\|$ ($\in \mathbb{R}^M$) // $\mathbf{E}_1 = [10 \mathbf{L} \ 0]^T$ ($\in \mathbb{R}^{M+1}$)

$\Delta \leftarrow \Delta_0 + [\mathbf{U}_1 \mathbf{L} \ \mathbf{U}_M]\mathbf{Y}_*$

In algorithm JF-GMRES, the least-square problem, $\mathbf{Y}_* \leftarrow \arg \min_{\mathbf{Y}} \|\beta\mathbf{E}_1 - \bar{\mathbf{H}}\mathbf{Y}\|$ ($\in \mathbb{R}^M$), is solved by the singular value decomposition (SVD) method.³⁶ Here, \mathbf{Y}_* is the vector that minimizes the norm of $\beta\mathbf{E}_1 - \bar{\mathbf{H}}\mathbf{Y}$, where $\beta = \|\Lambda_0\|$ with $\Lambda_0 = \mathbf{F}_{QM} - \mathbf{H}_{MD/QM}\Delta_0$ ($\in \mathbb{R}^{3N}$) being

the residual of the linear system for the initial guess Δ_0 ($\in \mathbb{R}^{3N}$) of its solution, $\mathbf{E}_1 = [10\mathbf{L} \ 0]^T \in \mathbb{R}^{M+1}$, and the (i, j) element of matrix $\bar{\mathbf{H}} \in \mathbb{R}^{(M+1) \times M}$ is denoted as h_{ij} . For obtaining the initial guess Δ_0 of the solution in algorithm JF-GMRES, as well as for obtaining the initial guess $\mathbf{R}_0^{\text{tst}}$ for the transition state in algorithm MNK-TSF, we use a steered eigenvector-following algorithm, which is described in Section 2.3.

2.3. Steered eigenvector-following algorithm for initializing the Newton-Raphson solution

The Hessian matrix of the input state $\mathbf{R}_0^{\text{tst}}$ in Section 2.1 is assumed to have only one negative eigenvalue λ_1 , so that the Newton iteration converges to a transition state. At a transition state, the forces are zero, whereas the energy takes a minimum value for all directions except for the Hessian eigenvector \mathbf{V}^1 corresponding to λ_1 , along which the energy is instead maximal. Thus, the correct iterative step toward the transition state is ascendant along \mathbf{V}^1 and descendant for all the other directions. To see that the Newton-Raphson method provides the correct step in such a case, we substitute in Eq. (2) a completeness relation in the $3N$ -dimensional vector space,

$$\mathbf{I} = \sum_{k=1}^{3N} \mathbf{V}^k \mathbf{V}^{k\text{T}}, \quad (11)$$

where λ_k and \mathbf{V}_k are the k -th smallest eigenvalue and the corresponding eigenvector of the Hessian matrix (the norm of each eigenvector is normalized to unity) and \mathbf{I} is the $3N$ by $3N$ identity matrix. The Newton step in Eq. (2) is then rewritten as

$$\mathbf{R} \leftarrow \mathbf{R} + \sum_{k=1}^{3N} \frac{\mathbf{V}^k \mathbf{V}^{k\text{T}} \mathbf{F}(\mathbf{R})}{\lambda_k}. \quad (12)$$

For positive Hessian eigenvalues ($\lambda_k > 0$), Eq. (12) performs steepest descent (i.e., the atoms are displaced parallel to the projected forces), whereas for negative Hessian eigenvalues ($\lambda_k < 0$), it instead performs steepest ascent (i.e., the displacements are anti-parallel to the forces).

The input state $\mathbf{R}_0^{\text{tst}}$ to the MNK-TSF algorithm in Table 1 is prepared in such a way that it has only one negative Hessian eigenvalue, by using a steered eigenvector-following (SEF) algorithm in Table 3. The SEF algorithmic step is also used to initialize the atomic displacements Δ_0 in the JF-GMRES algorithm in Table 2. The SEF algorithm starts from a local minimum-

energy configuration \mathbf{R}^{init} , such that $\mathbf{F}(\mathbf{R}^{\text{init}}) = 0$ and such that all the eigenvalues of the Hessian matrix $\mathbf{H}(\mathbf{R}^{\text{init}})$ are positive. It then constructs a continuous escape path $\mathbf{R}(\tau)$ ($\mathbb{R} \rightarrow \mathbb{R}^{3N}$; τ is a real-valued parameter) from $\mathbf{R}^{\text{init}} = \mathbf{R}(0)$ to an approximate transition state, $\mathbf{R}_0^{\text{tst}} = \mathbf{R}(\tau_{\max} > 0)$.

In order to initiate the path, 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.³⁷ A specific example is a bond-length constraint imposed on a given atomic pair (i, j) ,³⁸

$$\sigma = \{S(\tau)\} = \left\{ \|\mathbf{r}_{ij}\| = r_{ij}(\tau) = r_{ij}^0 + \dot{r}_{ij}(\tau - \tau_{\text{init}}) \right\}, \quad (13)$$

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.

Table 3

Steered eigenvector-following (SEF) algorithm for preparing an approximate transition state.

Algorithm SEF

Input:

$\mathbf{R}^{\text{init}} \in \mathbb{R}^{3N}$: an initial local minimum-energy state
 $\sigma = \{S(\tau)\}$: an event seed, i.e., a parameterized sequence of $(3N-1)$ -dimensional constraint surfaces

Output:

$\mathbf{R}_0^{\text{tst}}$: an approximate transition state

Steps:

1. *Steered centrifugal escape*

```

 $\tau \leftarrow 0$ 
 $\mathbf{R} \leftarrow \mathbf{R}^{\text{init}}$ 
do
   $\tau \leftarrow \tau + \delta\tau$ 
   $\mathbf{R} \leftarrow \mathbf{R} + (\delta\tau^2 / 2\langle m \rangle) \mathbf{F}_{\text{MD}}$  // steepest-descent step
   $\mathbf{R} \leftarrow P(S(\tau)) \mathbf{R}$  // projection onto the constraint surface
  while  $\lambda_1 \geq -\Delta\lambda_1$  //  $\lambda_1$  is the minimum eigenvalue of the MD Hessian matrix

```

2. *Eigenvector-following climb*

```

do
   $\mathbf{R} \leftarrow \mathbf{R} - \frac{\delta\tau^2}{2\langle m \rangle} \left( \mathbf{V}^1 \mathbf{V}^{1\text{T}} \right) \mathbf{F}_{\text{MD}} + \frac{\delta\tau^2}{2\langle m \rangle} \left( \mathbf{I} - \mathbf{V}^1 \mathbf{V}^{1\text{T}} \right) \mathbf{F}_{\text{MD}}$  // eigenvector-following step
  while  $\|\mathbf{F}_{\text{MD}}(\mathbf{R})\| > F_{\text{threshold}}$ 
   $\mathbf{R}_0^{\text{tst}} \leftarrow \mathbf{R}$ 

```

The SEF algorithm consists of two algorithmic phases (see Table 3): 1) steered centrifugal escape from the convex region (in which the Hessian matrix is positive definite) of the initial

energy-minimum; and 2) eigenvector-following climb to a transition state. The steered centrifugal escape phase starts from the initial state \mathbf{R}^{init} , and performs a sequence of steepest-descent steps,

$$\mathbf{R} \leftarrow \mathbf{R} + \frac{\delta\tau^2}{2\langle m \rangle} \mathbf{F}_{\text{MD}}, \quad (14)$$

where $\delta\tau$ ($\sim 10^{-15}$ s) is a time-discretization unit and $\langle m \rangle$ is the average mass of the atoms. Each steepest-descent step is followed by the projection of state \mathbf{R} onto the constrained surface,

$$\mathbf{R} \leftarrow P(S(\tau))\mathbf{R}, \quad (15)$$

corresponding to the current time τ , where $P(S(\tau))$ is the projection operator. For the bond-length constraint in Eq. (13), the projection operator is expressed as

$$P(S(\tau))\mathbf{r}_k = \mathbf{r}_k + \frac{\delta_{ki} - \delta_{kj}}{2} \left(\frac{r_{ij}(\tau)}{\|\mathbf{r}_{ij}\|} - 1 \right) \mathbf{r}_{ij} \quad (k = 1, K, N), \quad (16)$$

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 with $O(N)$ operations using the Lanczos algorithm^{39, 40} in the Appendix. We again use a finite-difference method, so that only the forces but not the Hessian matrix need to be computed. It typically requires 4–8 force evaluations for λ_1 to converge within a convergence criterion Δ_{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/Å²), to terminate the escape steps when $\lambda_1 < -\Delta\lambda_1$.

Once the minimum Hessian eigenvalue becomes negative, the eigenvector-following climb phase performs steepest ascent parallel to the Hessian eigenvector \mathbf{V}^1 and steepest descent to all the other directions,³⁹⁻⁴¹

$$\mathbf{R} \leftarrow \mathbf{R} - \frac{\delta\tau^2}{2\langle m \rangle} \left(\mathbf{V}^1 \mathbf{V}^{1\text{T}} \right) \mathbf{F}_{\text{MD}} + \frac{\delta\tau^2}{2\langle m \rangle} \left(\mathbf{I} - \mathbf{V}^1 \mathbf{V}^{1\text{T}} \right) \mathbf{F}_{\text{MD}}. \quad (17)$$

The eigenvector-following climb steps are terminated, when the norm of the forces falls below a prescribed threshold value: $\|\mathbf{F}_{\text{MD}}(\mathbf{R})\| \leq F_{\text{threshold}}$.

The eigenvector-following steps, Eq. (17), converge to a transition state as explained at the beginning of this subsection and are related to the Newton step, Eq. (12), as follows. We can

control the magnitude of the atomic displacements (and also whether to perform descent or ascent for different eigenmodes) by introducing a control parameter μ in Eq. (12):^{19,20}

$$\mathbf{R} \leftarrow \mathbf{R} + \sum_{k=1}^{3N} \frac{\mathbf{V}^k \mathbf{V}^{kT} \mathbf{F}(\mathbf{R})}{\lambda_k - \mu}. \quad (18)$$

By choosing μ such that $\lambda_1 < \mu < \lambda_2 < \dots < \lambda_{3N}$ and by approximating $1/|\lambda_1 - \mu| = 1/(\lambda_2 - \mu) = \dots = 1/(\lambda_{3N} - \mu) = (\delta\tau)^2 / 2\langle m \rangle$, Eq. (18) reduces to Eq. (17).

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

The MNK-TSF algorithm has been implemented on parallel computers by first assigning the MD and QM tasks to separate processor groups (task decomposition) and then using spatial decomposition within each task for further parallelism, see Fig. 1. The parallel MNK-TSF 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 process groups by defining multiple MPI communicators as subsets of MPI_COMM_WORLD. (The MPI communicator construct combines a process group and a context, in such a way that messages with different contexts are not intermixed.) In our program, the MD task, as well as the QM task, is assigned a dedicated communicator. One advantage of the hybrid task + spatial decomposition approach^{5, 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 changes dynamically on demand and resources are allocated and tasks are migrated adaptively in response to unexpected faults.²⁹

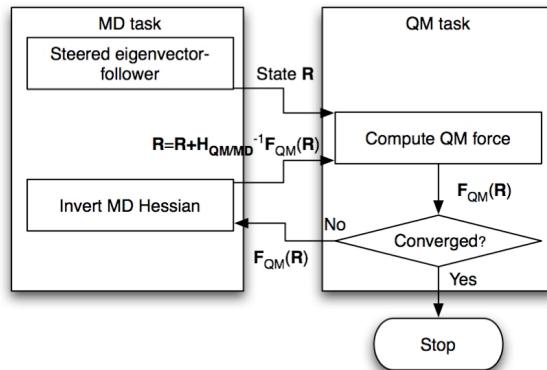


Fig. 1. Flow chart of the parallel QM/MD MNK-TSF algorithm.

In spatial decomposition within each task,⁷ the total volume of the system is divided into P subsystems of equal volume, and each subsystem is assigned to a processor in an array of P 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 during a steepest-descent/ascent or Newton step, 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 , while communication scales in proportion to $(N/P)^{2/3}$. Tree-based algorithms such as the fast multipole method (FMM)⁴² incur an $O(\log P)$ overhead, which is negligible for coarse-grained ($N/P \gg P$) applications.⁴³

4. Numerical results

We test the MNK-TSF algorithm for the reaction of a water (H_2O) molecule in amorphous silica (a- SiO_2). We first prepare an a- SiO_2 structure containing 192 atoms through a melt-quench procedure using a classical MD interatomic potential that consists of two- and three-body terms (see Fig. 2).⁴⁴⁻⁴⁶ The potential has been validated by comparing MD results with experimental measurements of structural correlations, elastic moduli, fracture toughness, and phonon density of states.⁴⁴⁻⁴⁶ We have also confirmed that structural optimization using QM forces in the framework of DFT¹¹ does not change the a- SiO_2 structure (the maximal QM force per atom in the MD local minimum-energy structure is 0.01 eV/ \AA ?, and the maximal atomic displacement by switching to the QM forces is less than 0.1 \AA ?).

We next introduce a water molecule with a random orientation at the center of a large cage composed of Si-O bonds in the a- SiO_2 configuration (see Fig. 2), where the interatomic potential is switched to a chemically reactive force field (ReaxFF) to allow reactions between H_2O and SiO_2 .⁴⁷ The first principles-based ReaxFF approach significantly reduces the computational cost of simulating chemical reactions through environment-dependent bond orders^{48, 49} and dynamic charge transfers based on a charge-equilibration scheme.⁵⁰⁻⁵⁴ The SEF algorithm in Table 3 is then used to locate a transition state approximately using the ReaxFF, where the distance between one of the hydrogen atoms in the water molecule and the oxygen atom in one of the Si-O-Si group in the cage surrounding the water is used as the constraint in Eq. (13).

Finally, the MNK-TSF algorithm using DFT-based QM forces¹¹ is used to refine the transition state found with ReaxFF. Figure 2 shows the resulting transition state along with its ReaxFF and QM energies.

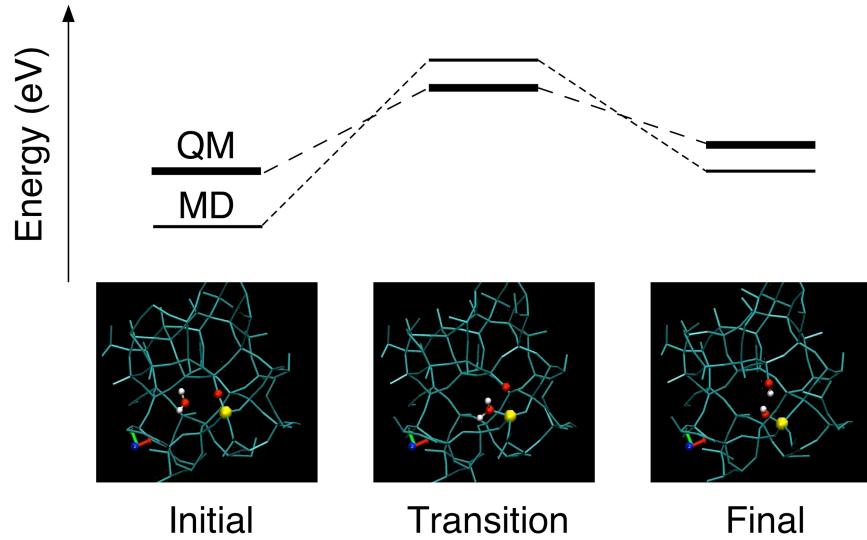


Fig. 2. MD and QM energies of the initial (left), transition (middle), and final (right) states of a hydrolysis reaction in amorphous silica.

Figure 3 shows the convergence of the MNK-TSF algorithm, i.e., the norm of the QM forces as a function of the Newton step. Despite a small number of QM force evaluations, the algorithm achieves quadratic convergence.

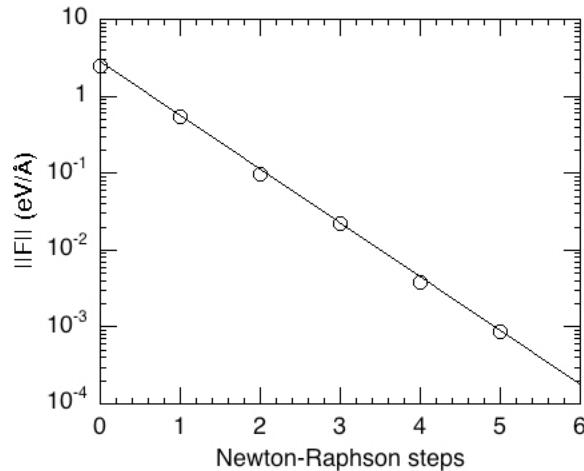


Fig. 3. Convergence of the quantum-mechanical force as a function of the Newton-Raphson step, without and with the QM correction to the Hessian matrix.

To estimate the computational cost of the MNK-TSF algorithm, Table 4 compares the

numbers of QM and MD force evaluations, as well as the corresponding wall-clock times. The computation is performed on a Linux cluster of dual-core, dual-processor AMD Opteron nodes, where the MD computation is performed on one processor whereas the QM computation uses 128 processors. If the QM forces were used both in the outer (Newton) and inner (Krylov) loops of the MNK-TSF algorithm, the resulting wall-clock time to achieve the same convergence would be xxx s compared with the present yyy s, i.e., the multiscale QM/MD scheme has reduced the computational time by a factor of zzz.

Table 4

Computational cost of the MNK-TSF algorithm. The numbers of QM and MD force evaluations, as well as the corresponding wall-clock times, are compared.

	MD	QM
Number of force calls	124	6
Wall-clock time (s)	1.2	10,250.0

5. Summary

We have designed an algorithm for finding transition states with quantum-mechanical accuracy in a system containing thousands of atoms. The multiscale Newton-Krylov transition-state finder (MNK-TSF) algorithm consists of doubly nested loops. The outer Newton iteration is driven by accurate QM forces in the framework of the density functional theory, combined with approximate inversion of the Hessian matrix using a less compute-intensive molecular-mechanical method. The inner Krylov-subspace iteration inverts the MM Hessian matrix in a Jacobian-free manner using the generalized minimal residual method. Furthermore, a QM correction to the MM Hessian matrix is added within the rank-two Powel symmetric-Broyden approximation. The MNK-TSF algorithm has been implemented on parallel computers using hybrid task + spatial decomposition. The parallel algorithm has then been used to find transition states for hydrolysis reactions in amorphous silica using a hierarchy of simulation methods. Here, a classical MM force field and a chemically reactive force field have been used in sequence to explore a large solution space and to approximately locate a reaction path, whereas a limited number of DFT-based QM force calculations have been invoked to refine the path.

Though the MNK-TSF algorithm has been presented for the determination of a transition state in this paper, it can easily be extended to determine an entire atomistic pathway with QM accuracy, when it is formulated variationally,^{55,56}.

Acknowledgments

This work was partially supported by AFOSR—DURINT, ARO—MURI, DOE—SciDAC, DTRA, and NSF. Numerical tests were performed at the University of Southern California using the 5,384-processor Linux cluster at the Research Computing Facility and the 2,048-processor Linux cluster at the Collaboratory for Advanced Computing and Simulations.

Appendix. Lanczos algorithm to obtain the minimal Hessian eigenpair

The Lanczos algorithm below 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 3.

Algorithm lanczos

Input:

$\mathbf{R} \in \mathbb{R}^{3N}$: a state
logical *initialize*: TRUE for the first call in each eigenvector-following procedure; FALSE otherwise

Output:

λ_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 λ_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

$s \leftarrow s + 1$

$\mathbf{Q}_s \leftarrow \Delta / \beta_{s-1}$

$c_{\text{fd}} \leftarrow \max_i \{ |Q_{s,i}| \mid i = 1, \dots, 3N \} / \delta_{\text{fd}}$

$\Delta \leftarrow c_{\text{fd}} [-\mathbf{F}(\mathbf{R} + \mathbf{Q}_s / c_{\text{fd}}) + \mathbf{F}(\mathbf{R})] - \beta_{s-1} \mathbf{Q}_{s-1}$

$\alpha_s \leftarrow \mathbf{Q}_s^T \Delta$

$\Delta \leftarrow \Delta - \alpha_s \mathbf{Q}_s$

$\beta_s \leftarrow \|\Delta\|$

diagonalize $\mathbf{T}_s = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \beta_2 & \\ & 0 & 0 & 0 \\ & & \beta_{s-2} & \alpha_{s-1} & \beta_{s-1} \\ & & & \beta_{s-1} & \alpha_s \end{bmatrix}$, so that $\tilde{\mathbf{Q}}_s^T \mathbf{T}_s \tilde{\mathbf{Q}}_s = \text{diag}(\tilde{\lambda}_1^s, \dots, \tilde{\lambda}_s^s)^*$

while $|(\tilde{\lambda}_1^s - \tilde{\lambda}_1^{s-1}) / \tilde{\lambda}_1^{s-1}| > \Delta_{\text{eigen}}$

$\lambda_1 \leftarrow \tilde{\lambda}_1^s$

$\mathbf{V}_1 \leftarrow \sum_{k=1}^s \mathbf{Q}_k \tilde{q}_k^1$

$\mathbf{V}_1 \leftarrow \mathbf{V}_1 / \|\mathbf{V}_1\|$

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

References

- ¹ J. J. Gilman, *Science* **274**, 65 (1996).
- ² B. Lawn, *Fracture of Brittle Solids* (Cambridge University Press, Cambridge, UK, 1993).
- ³ J. Q. Broughton, F. F. Abraham, N. Bernstein, et al., *Physical Review B* **60**, 2391 (1999).
- ⁴ A. Nakano, M. E. Bachlechner, R. K. Kalia, et al., *Computing in Science & Engineering* **3**(4), 56 (2001).
- ⁵ S. Ogata, E. Lidorikis, F. Shimojo, et al., *Computer Physics Communications* **138**, 143 (2001).
- ⁶ S. Ogata, F. Shimojo, R. K. Kalia, et al., *Journal of Applied Physics* **95**, 5316 (2004).
- ⁷ A. Nakano, R. K. Kalia, P. Vashishta, et al., *Scientific Programming* **10**, 263 (2002).
- ⁸ A. Nakano, R. K. Kalia, K. Nomura, et al., *Computational Materials Science*, in press (2007).
- ⁹ P. Hohenberg and W. Kohn, *Physical Review* **136**, B864 (1964).
- ¹⁰ W. Kohn and P. Vashishta, *Inhomogeneous Electron Gas*, eds. N.H. March and S. Lundqvist, 79 (Plenum, 1983).
- ¹¹ F. Shimojo, R. K. Kalia, A. Nakano, et al., *Computer Physics Communications* **167**, 151 (2005).
- ¹² I. G. Kevrekidis, C. W. Gear, and G. Hummer, *AICHE Journal* **50**, 1346 (2004).
- ¹³ M. H. M. Olsson, G. Y. Hong, and A. Warshel, *Journal of the American Chemical Society* **125**, 5025 (2003).
- ¹⁴ D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein, *Journal of Physical Chemistry* **100**, 12771 (1996).
- ¹⁵ A. F. Voter, F. Montalenti, and T. C. Germann, *Annual Review of Materials Research* **32**, 321 (2002).
- ¹⁶ J. E. Dennis and R. B. Schnable, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (SIAM, Philadelphia, PA, 1996).
- ¹⁷ R. Fletcher, *Practical Methods of Optimization*, 2nd Ed. (John Wiley & Sons, Chichester, UK, 1987).
- ¹⁸ D. Wales, *Energy Landscapes: Applications to Clusters, Biomolecules and Glasses* (Cambridge University Press, Cambridge, UK, 2004).
- ¹⁹ C. J. Cerjan and W. H. Miller, *Journal of Chemical Physics* **75**, 2800 (1981).
- ²⁰ A. Banerjee, N. Adams, J. Simons, et al., *Journal of Physical Chemistry* **89**, 52 (1985).
- ²¹ J. Baker, *Journal of Computational Chemistry* **7**, 385 (1986).
- ²² P. Culot, G. Dive, V. H. Nguyen, et al., *Theoretica Chimica Acta* **82**, 189 (1992).
- ²³ H. B. Schlegel, *Journal of Computational Chemistry* **24**, 1514 (2003).

- ²⁴ D. A. Knoll and D. E. Keyes, Journal of Computational Physics **193**, 357 (2004).
- ²⁵ Y. Saad and M. H. Schultz, SIAM Journal on Scientific and Statistical Computing **7**, 856 (1986).
- ²⁶ Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd Ed. (SIAM, Philadelphia, 2003).
- ²⁷ H. Kikuchi, R. K. Kalia, A. Nakano, et al., Proceedings of Supercomputing 2002 (IEEE, 2002).
- ²⁸ W. Gropp, E. Lusk, and A. Skjellum, *Using MPI*, 2nd Ed. (MIT Press, Cambridge, 1999).
- ²⁹ H. Takemiya, Y. Tanaka, S. Sekiguchi, et al., Proceedings of Supercomputing 2006 (IEEE/ACM, 2006).
- ³⁰ For chemically nonreactive force fields, the Hessian-vector product is calculated analytically.
- ³¹ A. Nakano, R. K. Kalia, K. Nomura, et al., International Journal of High Performance Computing Applications, in press (2007).
- ³² A. Warshel and M. Levitt, Journal of Molecular Biology **103**, 227 (1976).
- ³³ U. C. Singh and P. A. Kollman, J. Comput. Chem. **7**, 718 (1986).
- ³⁴ M. J. Field, P. A. Bash, and M. Karplus, J. Comput. Chem. **11**, 700 (1990).
- ³⁵ S. Dapprich, I. Komáromi, K. S. Byun, et al., J. Mol. Struct. (Theochem) **461-462**, 1 (1999).
- ³⁶ W. H. Press, S. A. Teukolsky, W. T. Vetterling, et al., *Numerical Recipes*, 2nd Ed. (Cambridge University Press, Cambridge, UK, 1992).
- ³⁷ D. H. Lu, M. Zhao, and D. G. Truhlar, Journal of Computational Chemistry **12**, 376 (1991).
- ³⁸ D. H. Lu and D. G. Truhlar, Journal of Chemical Physics **99**, 2723 (1993).
- ³⁹ N. Mousseau, P. Derreumaux, G. T. Barkema, et al., Journal of Molecular Graphics & Modelling **19**, 78 (2001).
- ⁴⁰ R. A. Olsen, G. J. Kroes, G. Henkelman, et al., Journal of Chemical Physics **121**, 9776 (2004).
- ⁴¹ L. J. Munro and D. J. Wales, Physical Review B **59**, 3969 (1999).
- ⁴² L. Greengard and V. Rokhlin, Journal of Computational Physics **73**, 325 (1987).
- ⁴³ S. Ogata, T. J. Campbell, R. K. Kalia, et al., Computer Physics Communications **153**, 445 (2003).
- ⁴⁴ P. Vashishta, R. K. Kalia, J. P. Rino, et al., Physical Review B **41**, 12197 (1990).
- ⁴⁵ T. J. Campbell, R. K. Kalia, A. Nakano, et al., Physical Review Letters **82**, 4018 (1999).
- ⁴⁶ Z. Lu, K. Nomura, A. Sharma, et al., Physical Review Letters **95**, 135501 (2005).

- ⁴⁷ A. van Duin, A. Strachan, S. Stewman, et al., Journal of Physical Chemistry A **107**, 3803 (2003).
- ⁴⁸ J. Tersoff, Physical Review B **37**, 6991 (1988).
- ⁴⁹ D. W. Brenner, Phys. Stat. Sol. (b) **217**, 23 (2000).
- ⁵⁰ A. K. Rappe and W. A. Goddard, Journal of Physical Chemistry **95**, 3358 (1991).
- ⁵¹ F. H. Streitz and J. W. Mintmire, Physical Review B **50**, 11996 (1994).
- ⁵² S. W. Rick, S. J. Stuart, and B. J. Berne, Journal of Chemical Physics **101**, 6141 (1994).
- ⁵³ A. Nakano, Computer Physics Communications **104**, 59 (1997).
- ⁵⁴ T. J. Campbell, R. K. Kalia, A. Nakano, et al., Physical Review Letters **82**, 4866 (1999).
- ⁵⁵ G. Henkelman and H. Jonsson, Journal of Chemical Physics **113**, 9978 (2000).
- ⁵⁶ D. Passerone and M. Parrinello, Physical Review Letters **87** 10, 108302 (2001).