JOURNAL OF CHEMICAL PHYSICS VOLUME 120, NUMBER 17 1 MAY 2004

A growing string method for determining transition states: Comparison to the nudged elastic band and string methods

Baron Peters
Department of Chemical Engineering, University of California, Berkeley, California 94720

Andreas Heyden
Department of Chemical Engineering, Hamburg University of Technology, D-21073 Hamburg, Germany

Alexis T. Bell
Department of Chemical Engineering, University of California, Berkeley, California 94720

Arup Chakraborty[a)]
Department of Chemical Engineering, University of California, Berkeley, California 94720
and Department of Chemistry, University of California, Berkeley, California 94720
and Biophysics Graduate Group, University of California, Berkeley, California 94720
and Physical Biosciences and Materials Science Division, Lawrence Berkeley National Laboratory,
University of California, Berkeley, California 94720

(Received 1 December 2003; accepted 9 February 2004)

Interpolation methods such as the nudged elastic band and string methods are widely used for calculating minimum energy pathways and transition states for chemical reactions. Both methods require an initial guess for the reaction pathway. A poorly chosen initial guess can cause slow convergence, convergence to an incorrect pathway, or even failed electronic structure force calculations along the guessed pathway. This paper presents a growing string method that can find minimum energy pathways and transition states without the requirement of an initial guess for the pathway. The growing string begins as two string fragments, one associated with the reactants and the other with the products. Each string fragment is grown separately until the fragments converge. Once the two fragments join, the full string moves toward the minimum energy pathway according to the algorithm for the string method. This paper compares the growing string method to the string method and to the nudged elastic band method using the alanine dipeptide rearrangement as an example. In this example, for which the linearly interpolated guess is far from the minimum energy pathway, the growing string method finds the saddle point with significantly fewer electronic structure force calculations than the string method or the nudged elastic band method. © *2004 American Institute of Physics.* [DOI: 10.1063/1.1691018]

INTRODUCTION

One of the most important contributions of electronic structure theory has been the prediction of potential energy surfaces from which reaction mechanisms and rates can be deduced. Calculations of rates usually employ harmonic transition state theory in which the energy and real vibrational frequencies at the saddle point and at the reactant minimum determine the reaction rate constant. Locating saddle points on *ab initio* potential energy surfaces can be extremely difficult, and remains one of the major challenges in chemical kinetics.

There are two families of algorithms for finding saddle points on potential energy surfaces, the surface walking algorithms and the interpolation algorithms.[1] Surface walking methods[2–5] explore the potential energy surface using local gradient and Hessian (or approximate Hessian) information. Such methods only require the reactant configuration, and are thus able to predict which products will be formed. Unfortunately, surface walking algorithms often work poorly for bimolecular reactions and unimolecular systems with several low frequency vibrational modes. Like more sophisticated transition path sampling methods,[6,7] interpolation algorithms[8–13] require both the reactant and product configurations. These generate a sequence of configurations (nodes) that interpolate between the reactant and product configurations. Interpolating algorithms convert the search for a first order saddle point in configuration space to an optimization problem in a discretized path space. When posed as a path optimization, bimolecular reactions and low frequency modes are easily handled.

The discretized path space dimensionality is larger than configuration space by a factor corresponding to the number of nodes used to represent the path. For most interpolation algorithms, the force is computed at each node of the path each time the path moves. In contrast, some surface walking algorithms require just one force calculation per iteration. Because of this disparity in computational cost per iteration, practical applications often use an interpolation scheme only to generate a good guess for the saddle point.[12] This guess is then optimized using a surface walking method such as

[a)] Author to whom correspondence should be addressed. Electronic mail: arup@uclink.berkeley.edu

Newton–Raphson (NR), a quasi-NR scheme,[2,4] or the Dimer method.[5]

Nudged-elastic-band (NEB) (Refs. 12, 13) is a widely used interpolation algorithm that converges to the minimum energy pathway (MEP). A number of applications have shown that NEB is a robust method for calculating reaction pathways.[14–17] However, NEB is not without disadvantages. The algorithm requires Hooke constants for springs that keep the nodes uniformly spaced along the path. If the spring constants are chosen to be too large, the condition number of the optimization problem can become very large,[18] giving slow convergence to the MEP. If the spring constants are chosen to be too small, the node spacing becomes erratic leading to a poor description of some sections of the path.

Ren *et al.*[19] have recently proposed the ''zero temperature string method'' for finding the MEP. In the zero temperature string method, the discretized string moves in the direction of the normal force at each node on the string. Instead of including tangential spring forces to maintain node spacing, the nodes are redistributed along the string after each move. For brevity the zero temperature string method will be referred to as the string method (SM), although it should be noted that a finite temperature string method has also been proposed.[19,20]

Both SM and NEB require an initial guess for the pathway. Most implementations of NEB employ linear synchronous transit[1] to start the algorithm. While linear synchronous transit is an adequate guess in many cases, for some reactions it may be far from the correct path. A bad guess can result in exceedingly slow convergence, or worse, in severely overlapping atoms. The latter situation may cause failure of the electronic structure calculations used to obtain the potential and forces. Choosing an appropriate set of bond lengths and angles for interpolation can provide a good starting path,[1] but selecting appropriate coordinates requires intuition about the reaction pathway. Intuition may be unavailable when exploring new chemistry or screening new ideas for a desired catalyst. It is in such applications that computational chemistry may prove to be most useful for guiding new experiments.

This paper presents a version of the string method in which the parameterization density evolves so that the string grows from each end of the pathway towards the transition state as each of the growing sections of the string converge. The growing string method thus finds the MEP without requiring an initial guess for the pathway. For systems that are poorly described by the linear synchronous transit guess, the growing string method (GS) obtains a realistic saddle point estimate more rapidly than NEB and SM. The remainder of the paper describes the string method and the growing string method. Two examples compare the convergence properties of NEB, SM, and GS. An Appendix augments the methodology discussion in the main text.

THE STRING METHOD

The string is a path, $\varphi(\sigma)$, connecting the reactant and product configurations. The parameterization variable, σ, is some monotonic function of the normalized arclength, s, measured from one end of the string. The actual function $\sigma(s)$ is determined from a parameterization density, and is normalized so that $\varphi(0)$ is the reactant configuration, and $\varphi(1)$ is the product configuration. Each iteration of the string method consists of two steps, an evolution step that guides the string toward the MEP, and a reparameterization step that enforces the prescribed parameterization density.[19]

EVOLUTION STEP

The evolution step guides the string toward the MEP, a path joining reactant and product configurations that is parallel to the gradient at each point along the path. If $\hat{\mathbf{t}}(\sigma)$ is the unit tangent vector along the string at $\varphi(\sigma)$, and V is the potential, then the normal force on the string at σ can be defined as

$$\mathbf{f}^{\perp}(\varphi(\sigma)) = -\nabla V(\varphi(\sigma)) + (\hat{\mathbf{t}}(\sigma)^T \nabla V(\varphi(\sigma)))\hat{\mathbf{t}}(\sigma). \quad (1)$$

The MEP by definition satisfies $\mathbf{f}^{\perp}(\varphi_{\mathrm{MEP}})=0$ so the MEP is a global minimum of the functional

$$F[\varphi] = \int_0^1 d\sigma \mathbf{f}^{\perp}(\varphi(\sigma))^T \mathbf{f}^{\perp}(\varphi(\sigma)) \quad (2)$$

over all paths connecting $\varphi(0)$ to $\varphi(1)$. In practice the string is represented by a series of discrete configurations, uniformly spaced in the variable σ, so the objective function becomes

$$F(\varphi(\sigma_0),\varphi(\sigma_1),\ldots,\varphi(\sigma_n)) = \sum_{k=0}^{n} |\mathbf{f}^{\perp}(\varphi(\sigma_k))|^2. \quad (3)$$

A number of schemes can be used to obtain the path tangent. The centered-difference scheme is the most intuitive. However, to prevent the string from growing kinks, the node spacing, R, must satisfy $R > |\nabla V \cdot \hat{\mathbf{t}}|/2\omega^2$, where ω^2 is the smallest eigenvalue of the Hessian in the hyperplane perpendicular to the path.[12] The upwind difference scheme for defining the path tangent at each node stabilizes the string against kink growth for all node spacings.[12] Tangents from the upwind scheme work well when the node spacings are approximately uniform. However, tangents from both finite difference schemes can unrealistically point toward a distant node if the node spacing varies significantly. Figure 1 shows an example of this situation, which is particularly relevant to the growing string method described below.

Using cubic splines to fit the curve $\varphi(s)$ gives realistic tangents and interpolated geometries even for nodes adjacent to a large gap in the interpolated pathway. The cubic spline with free ends provides natural path tangents at the endpoints, $\varphi(0)$ and $\varphi(1)$. These tangents are useful for evolving the endpoints of the string in bimolecular reactions where the endpoints are not necessarily potential energy minima. There is no easily derived stability criterion for the minimum node spacing because the splines depend on the position of every node. However, kink growth instabilities due to small node spacings have not been encountered. The appendix derives a ''dekinking force'' that arises naturally from the gradient of $F[\varphi]$. If a kinking instability arises, the normal force can be augmented with the dekinking force to stabilize the string method and the growing string method.

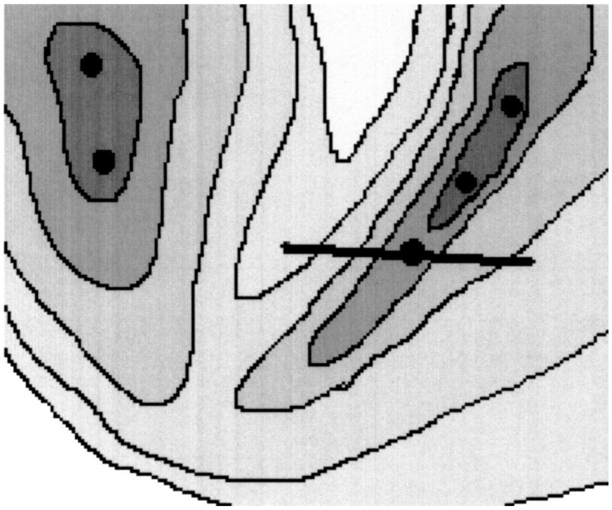

FIG. 1. When the node separation is not uniform, both centered difference and upwind difference schemes can give tangents that are unrealistically influenced by a distant node. The heavy black line illustrates an erroneous tangent for a node adjacent to a large gap between nodes.

There are a number of numerical methods for minimizing functions like $F[\varphi]$. The Appendix presents an analysis of the steepest descent, Newton–Raphson, and perpendicular force directions for minimizing $F[\varphi]$. Ren *et al.* employ Broyden's method,[18] which converges much faster than the steepest descent method and the method used in NEB, but requires an accurate initial guess for the pathway.[19] If an accurate initial guess is available, an estimated saddle point can be interpolated from the initial string. Then a pseudo-Newton–Raphson method (or the Dimer method) can be used to find the exact saddle point.[1,12] If an accurate MEP is needed, it can be determined from the converged saddle point by following the gradient down to the neighboring minima.[21–23] Thus in practical applications, converging the string to the MEP with extremely high accuracy is unnecessary.

REPARAMETERIZATION STEP

The reparameterization step redistributes the nodes along the string after each evolution step. Although reparameterization effectively plays a role similar to that of the springs in NEB, the freedom to choose a parameterization density based on intrinsic properties of the string imparts additional flexibility to the string method. The general mechanics of reparameterization will be discussed first, followed by the scheme for evolving the parameterization density so that the string grows inward from the endpoints to the transition state as the growing ends converge.

The function $\sigma(s)$ disperses nodes along the string at uniform intervals in σ, i.e., at $\sigma=0,1/n,2/n,\ldots,$ and 1 for a string with $n+1$ nodes, such that the density of nodes as a function of the normalized arclength, s, is consistent with a chosen parameterization density, $\rho(s)$. The inverse function $s(\sigma)$ is determined by solving the differential equation[19]

$$\frac{d}{d\sigma}\left(\rho(s)\frac{ds}{d\sigma}\right)=0 \tag{4a}$$

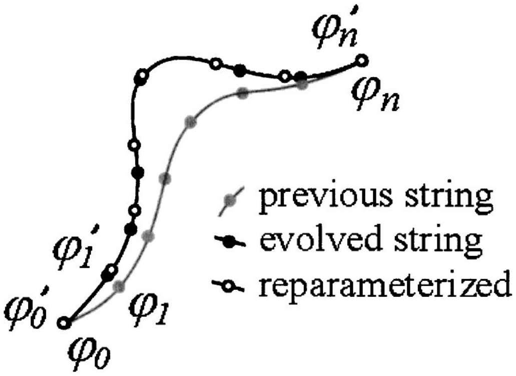

FIG. 2. An iteration of the string method showing the initial string, the evolved string, and the evolved string after reparametrization.

with the boundary conditions,

$$s(0)=0\text{and }\ s(1)=1. \tag{4b}$$

Integrating Eq. (4a) twice effectively normalizes the chosen parameterization density,

$$c=\int_0^1\rho(s)ds. \tag{5}$$

Integrating the density until the integral reaches a fraction $\sigma_k=k/n$ of c gives the new normalized arclength $s_k=s(\sigma_k)$ along the previously evolved string,

$$\int_0^{s_k}\rho(s)ds=c\sigma_k. \tag{6}$$

The reparameterized position $\varphi_k=\varphi(s_k)$ is then determined by following along the evolved string to an arclength s_kS, where S is the true un-normalized arclength of the string. The computational cost associated with the reparameterization procedure is negligible compared to the calculation of normal forces using ab initio methods, so it can be done between each step. The evolution and reparameterization steps are illustrated in Fig. 2.

GROWING STRING METHOD

The reparameterized string in Fig. 2 depicts the dispersion of nodes along the string that results from a uniform parameterization density, $\rho(s)$. As an alternative, the density can be chosen as a non-negative function of curvature or potential along the string to focus nodes near the transition state or near curves in the reaction pathway. Generally, the function $\rho(s)$ need not be continuous and it can change between iterations. Additionally, the number of nodes on the string, n, can change between iterations. These properties allow a systematic evolution of the parameterization density so that the string grows from its endpoints as it evolves until eventually the two ends join into one continuous string. By adaptively growing separate ends of the string toward the unknown interior of a reaction pathway, the string can avoid excessively rugged regions of the potential energy surface where electronic structure calculations may fail.

As the string ends grow into the unknown interior of the pathway, the nodes adjacent to the vacant interior play a

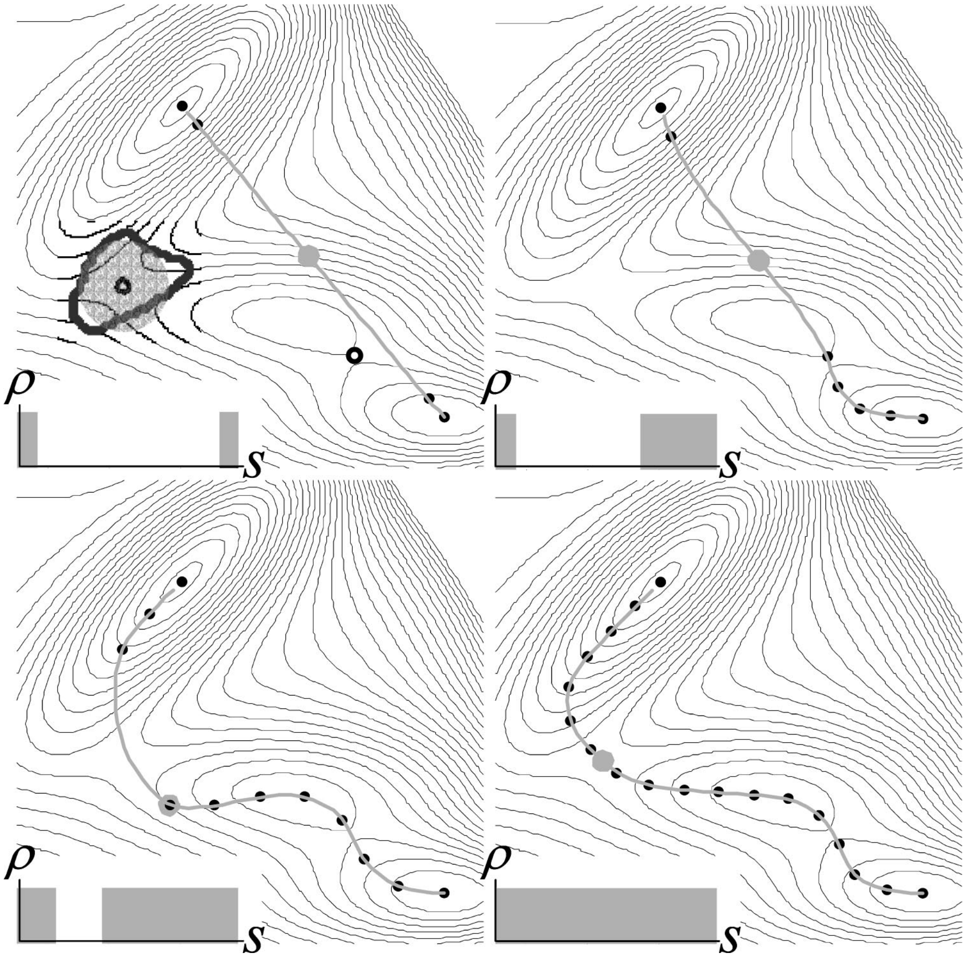

FIG. 3. (a)–(d) Four snapshots of a growing string on the Muller–Brown potential energy surface. Panels (a), (b), (c), and (d) show iterations 0, 4, 8, and 12, respectively. The nodes are shown as black dots, and the splined string is shown between the nodes as a gray curve. The parametrization density $\rho(s)$ is shown in the lower left of each panel. The gray dot on each string is the interpolated point of maximum energy. (a) shows the two saddle points on the potential energy surface as small open circles. The gray outline around the highest energy saddle point is its Newton–Raphson basin of attraction. The large shaded circle has radius 0.25 and approximately covers the basin of attraction.

critical role in the process of advancing each end into the interior. For convenience these nodes will be referred to as the left and right frontiers, or φ_{lf} and φ_{rf}, respectively. The normal force at a frontier point can be used as an indicator of whether it is safe to advance the growing end further into the interior of the reaction pathway.

To implement the growing string (GS) scheme, define the indicator function as

$$i(\mathbf{f}^{\perp})=\begin{cases}1 & \text{if } |\mathbf{f}^{\perp}|\leqslant\text{tolerance}\\ 0 & \text{if } |\mathbf{f}^{\perp}|>\text{tolerance},\end{cases} \tag{7}$$

where the tolerance is an adjustable parameter that can be fixed or adaptively changed like a trust radius. Let $\rho_k(s)$ be the parameterization density for the kth iteration. If n_k is the number of nodes on the string, and $\rho_k(s)$ is of the form

$$\rho_k(s)=\begin{cases}1 & \text{if } s\in[0,a_1^{(k)}]\\ 0 & \text{if } s\in(a_1^{(k)},a_2^{(k)})\\ 1 & \text{if } s\in[a_2^{(k)},1],\end{cases} \tag{8}$$

where $0<a_1^{(k)}<a_2^{(k)}<1$, then reparameterization will uniformly disperse the nodes outside of a vacant section of the string between the normalized arclengths $a_1^{(k)}$ and $a_2^{(k)}$. If ΔS is the desired arclength between nodes on the string, the normalized arclength spacing is $\Delta s_k=\Delta S/S_k$, where S_k is the total arclength of the string at the kth iteration. Beginning at $k=0$ with $n_0=2$, $a_1^{(0)}=0^+$, and $a_2^{(0)}=1-0^+$, rules for adaptively evolving the parameterization density and the number of nodes on the string, n_k, can be written as

$$\begin{aligned}&a_1^{(k+1)}=a_1^{(k)}+\Delta s_k i(\mathbf{f}^{\perp}(\varphi_{\mathrm{lf}})),\\ &a_2^{(k+1)}=a_2^{(k)}-\Delta(s_k i(\mathbf{f}^{\perp}(\varphi_{\mathrm{lf}}))),\\ &n_{k+1}=n_k+i(\mathbf{f}^{\perp}(\varphi_{\mathrm{lf}}))+i(\mathbf{f}^{\perp}(\varphi_{\mathrm{rf}})).\end{aligned} \tag{9}$$

The separate ends of the string merge when the fraction of vacant arclength a_2-a_1 reaches zero. Once the string has merged, the number of nodes remains fixed, and the parameterization density remains uniform. Figure 3 shows four snapshots of a growing string with 18 nodes on the Muller–Brown potential. (See the section on examples for a discussion of the Muller–Brown potential.)

The disjoint portions of the string in the growing string method (GS) are evolved separately until they unite. The entire string is used to determine a spline that gives string tangents on the left and right ends, but the function $F[\varphi]$ is split into two pieces $F_{\mathrm{left}}[\varphi]$ and $F_{\mathrm{right}}[\varphi]$ that are used to separately determine the stepsize for each of the two portions of string.

To decrease the number of electronic structure calculations, nodes that contribute a negligible fraction to $F[\varphi]$ can be fixed so that the stored gradient remains valid until the rest of the string has evolved sufficiently that the fixed node contributes a significant fraction to $F[\varphi]$. In the examples discussed in the next section, the fixed nodes scheme was used for NEB, SM, and GS calculations.

EXAMPLES

The following examples compare NEB, SM, and GS. In principle the only difference between SM and NEB is that the string is reparameterized after each evolution while NEB augments the normal forces that guide the string with a tangential spring force. The only other difference in our implementation is that NEB uses upwind scheme tangent vectors while SM and GS use cubic spline tangents. The most critical factors in the convergence rate of all three methods are the search directions and stepsizes used to minimize the residual gradients along the path. For a fair comparison, the same method was used to choose the stepsize along the force direction for GS, SM, and NEB. The search direction used for SM and GS, $\mathbf{v}_{\mathrm{SM}}$, is the concatenated normal force vector. The NEB search direction, $\mathbf{v}_{\mathrm{NEB}}$, is the concatenated normal force vector augmented by the tangential spring forces,

$$\mathbf{v}_{\mathrm{SM}}=(\mathbf{f}_0^{\perp},\mathbf{f}_1^{\perp},\ldots,\mathbf{f}_n^{\perp}), \tag{10a}$$

$$\mathbf{v}_{\mathrm{NEB}}=(\mathbf{f}_0^{\perp},\mathbf{f}_1^{\perp},\ldots,\mathbf{f}_n^{\perp})+(\mathbf{f}_0^{\parallel},\mathbf{f}_1^{\parallel},\ldots,\mathbf{f}_n^{\parallel}). \tag{10b}$$

At each iteration, a quadratic polynomial for the function F in distance along these vectors is generated using three points along the search direction vector. The string (or band) φ is moved along the direction $\mathbf{v}_{\mathrm{SM}}$ (or $\mathbf{v}_{\mathrm{NEB}}$) to the minimum of the quadratic polynomial. If the minimum does not exist, or if the distance to the minimum exceeds a maximum allowed displacement, the maximum allowed step is taken.

When using NEB, SM, or GS with an *ab initio* potential energy surface, gradient calculations constitute nearly all of the computational cost. Thus, we compare methods based on the number of gradients required to reach the saddle point or the Newton–Raphson basin of attraction for the saddle point.

THE MULLER–BROWN POTENTIAL

The Muller–Brown potential[24] has a number of interesting features that make it a good test surface. The MEP passes through two saddle points and deviates significantly from the linearly interpolated path. Moreover, a linearly interpolated chain must slide down a dividing surface on which the potential function is not convex. A contour plot of the surface was shown in Fig. 3(a) with the Newton–Raphson basin of attraction around the highest energy saddle point, $\mathbf{x}_{\mathrm{sp}}$. The shaded circle, $|\mathbf{x}-\mathbf{x}_{\mathrm{sp}}|\leqslant 0.25$, approximately covers the basin of attraction.

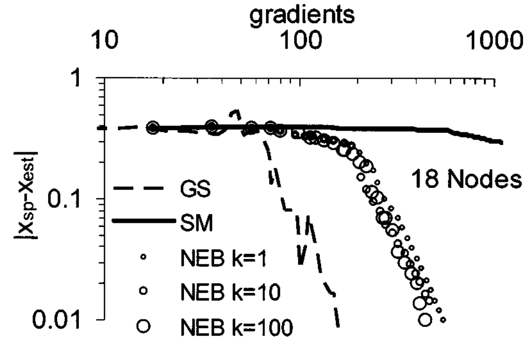

FIG. 4. Error in the saddle point estimate, $\mathbf{x}_{\mathrm{est}}$, as the distance from the true saddle point, $\mathbf{x}_{\mathrm{sp}}$, vs gradients computed. Results are shown at 18 nodes for the growing string method (GS), the string method (SM), and the nudged elastic band (NEB). GS reaches the circle of Fig. 4 with a third as many gradients as NEB. SM performs very poorly with 18 nodes.

An estimated saddle point, $\mathbf{x}_{\mathrm{est}}$, can be obtained from each string by interpolating the point of maximum energy along the string. If the estimated saddle lies within the circle, the exact saddle is easily found using the eigenvector following algorithm.[3,4] The initial Hessian, $\mathbf{K}$, for eigenvector following was constructed with the estimated path tangent, $\mathbf{t}_{\mathrm{est}}$, as an eigenvector with the estimated second directional derivative of the energy, $-\omega_F^2$, as its eigenvalue,

$$\mathbf{K}=\omega_{\max}^2(\mathbf{I}-\mathbf{t}_{\mathrm{est}}\mathbf{t}_{\mathrm{est}}^T)-\omega_F^2\mathbf{t}_{\mathrm{est}}\mathbf{t}_{\mathrm{est}}^T. \tag{11}$$

When started within the circle on this two dimensional surface, the number of eigenvector following steps required to converge on the saddle point, $\mathbf{x}_{\mathrm{sp}}$, was always less than six—much fewer than the number required to generate an estimate, $\mathbf{x}_{\mathrm{est}}$, within the circle. Thus, GS, SM, and NEB can be compared based on the number of gradients required to generate an estimate within the basin of attraction of the saddle point. Figure 4 shows the convergence of the estimated saddle point to the true saddle point for a string with 18 nodes.

GS rapidly converges to the true saddle point, and reaches the basin of attraction with a third of the gradients required for NEB. At 18 nodes SM requires more than 1000 gradients to find the basin of attraction of the saddle point. The relative performances of GS, SM, and NEB depend on

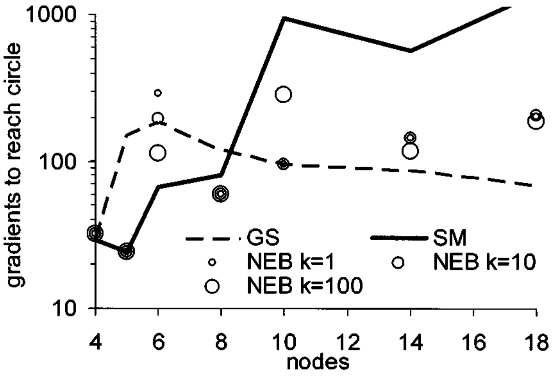

FIG. 5. A summary of results comparing convergence of the growing ends string method (GS), the string method (SM), and nudged-elastic-band (NEB) for the Muller–Brown potential energy surface. The relative performance of the methods depends on the number of nodes.

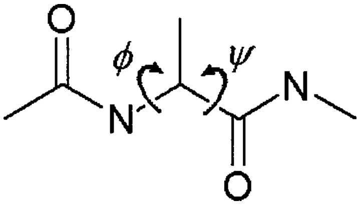

FIG. 6. The alanine dipeptide rearranges via simultaneous rotations about the dihedral angles ϕ and ψ.

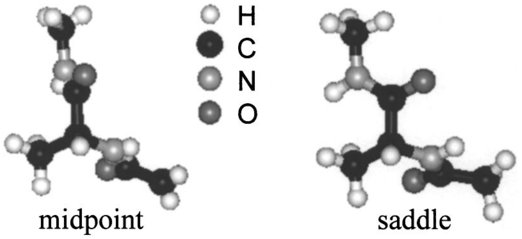

FIG. 8. The interpolated midpoint and saddle point configurations for the alanine dipeptide internal rotation pathway.

the number of nodes. Figure 5 compares the number of gradient calculations required for each method to reach the basin of attraction as a function of the number of nodes.

For a four-node string SM and GS perform slightly better than NEB, presumably because tangent vectors obtained from cubic splines are more accurate than the NEB tangents for widely separated nodes. Note that GS and SM are identical when the maximum number of nodes is four, because the first two interior nodes are automatically grown, making the string complete at four nodes. SM performed well with fewer than 10 nodes, but performed poorly as the number of nodes increased. If instability caused the slow convergence of SM, the number of gradients would grow disproportionately with increasing numbers of nodes, i.e., with decreasing node spacing. A 50 node string required approximately (50/18) times as many gradients as the 18 node string. Thus, it does not appear that instability caused the poor performance of SM.

NEB performed well at all node numbers. NEB with a spring constant of $k=10$ consistently reached the basin of attraction within 210 gradients. GS reached the basin of attraction within 185 gradients at all node numbers. An impressive feature of GS is that, beyond a threshold number of nodes, the number of gradients required to obtain an accurate saddle estimate is only weakly dependent on the number of nodes.

Clear advantages of GS over SM and NEB are not readily apparent from this example. On a two-dimensional potential energy surface the string tangents are entirely determined by the normal force directions. Small changes in the tangent directions can thus cause differences in the efficiency of the algorithm for minimizing $F[\varphi]$. For higher dimensional problems, differences are averaged out by successive minimization steps on the high dimensional hyperplanes perpendicular to the tangent. As a second example we consider the alanine dipeptide rearrangement.

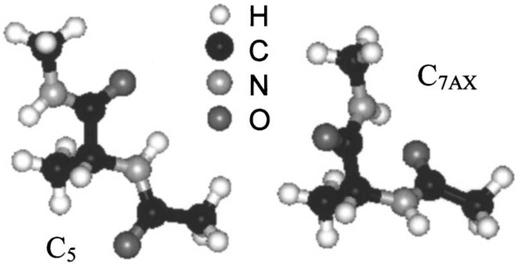

FIG. 7. C_5 and C_{7AX} minima for the internal rearrangement of alanine dipeptide. The structures have been rotated and translated to minimize their relative displacements.

ALANINE DIPEPTIDE REARRANGEMENT

The alanine dipeptide rearrangement has been studied extensively[25–29] and is often used to test theoretical methods for studying reactive systems.[30–35] In the 66-dimensional space of Cartesian atomic coordinates, the alanine dipeptide rearrangement provides an excellent example of the difficulties that can occur when linear synchronous transit is used to obtain an initial path. The reaction consists of simultaneous dihedral rotations about the angles ϕ and ψ shown in Fig. 6.

The gas-phase reactant and product minima as determined using B3LYP/6-31G calculations are shown in Fig. 7. The relative orientations of the molecules were adjusted to have the same center of mass and rotational orientations that minimize the distance between the structures in unmass-weighted Cartesian coordinates.

Figure 8 shows the linearly interpolated midpoint between the structures of Fig. 7. Figure 8 also shows the saddle point on the pathway between these structures. Note that the carbonyl groups and the NH moieties are severely crowded in the interpolated structure. In mass-weighted coordinates, the crowding in the midpoint structure is even more severe than in unmass-weighted coordinates.

Table I shows the energies and dihedral angles for the C_5 minimum, C_{7AX} minimum, and saddle point configurations. Energies and angles obtained by Perczel *et al.*[29] are included for comparison.

TABLE I. Dihedral angles and energies at stationary points along the alanine dipeptide rearrangement pathway.

	ϕ	ψ	V
	B3LYP/6-31G[a]		
C_5	−161.2	165.1	0.0
saddle	107.3	−139.7	7.6
C_{7AX}	71.7	−58.2	0.9
	B3LYP/3-21G[b]		
C_5	−172.8	174.5	0.0
saddle	97.7	−150.9	8.3
C_{7AX}	71.3	−57.8	−0.5

ϕ: (C–N–C_α–C) ψ: N–C_α–C–N
angles in deg, energy in kcal/mol

[a] This work.
[b] Reference 29.

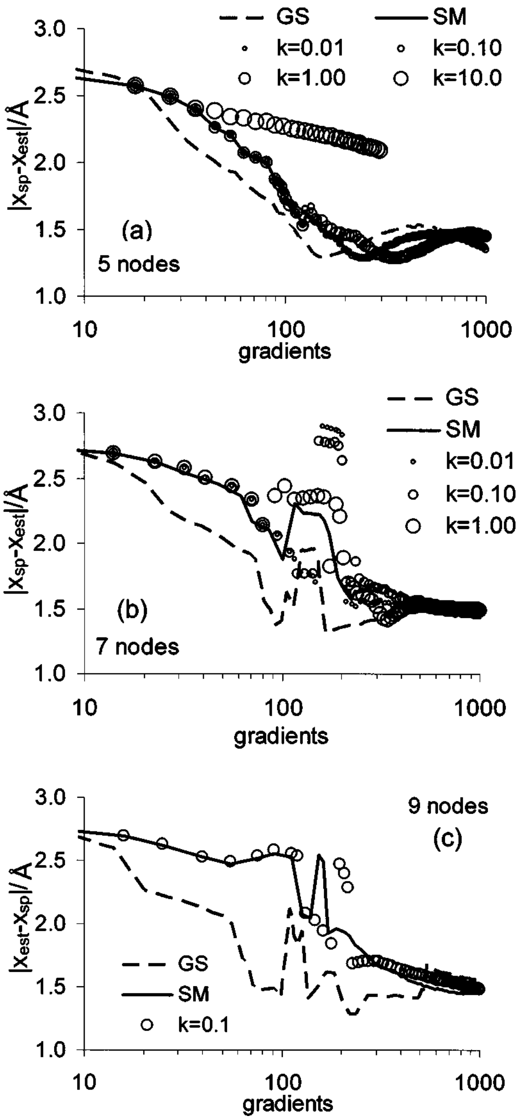

FIG. 9. (a)–(c) Error in the saddle point estimate plotted vs the number of gradients computed in each of the chain of states schemes. (a), (b), and (c) show results for strings with 5, 7, and 9 nodes, respectively. NEB results are labeled according to the spring constant used. With a spring constant of $k = 0.1$ Ha/Bohr^2, the NEB results are similar to the string method (SM) results. The growing string method (GS) rapidly obtains accurate approximations in all cases. Note the logarithmic scale for number of gradients.

Figure 9 shows the error in the saddle point estimate, $|\mathbf{x}_{\text{est}} - \mathbf{x}_{\text{sp}}|$, in angstroms as a function of the number of gradients computed in each of the interpolation schemes.

At seven and nine nodes, the error in the saddle point estimate occasionally increases sharply with additional gradients. These perturbations are not related to the addition of nodes to the growing string because they occur for GS, SM, and NEB. The perturbations occur when the estimated saddle point shifts tangentially along the string. This phenomenon occurs each time a node far from the saddle temporarily becomes the highest energy point on the string.

Figure 9 shows that the various methods behave similarly in the limit of many iterations. The advantages of the growing string method are most apparent in the early stages of the string optimization. In practical applications, NEB and SM generate a good estimate of the transition state which is then reoptimized using a pseudo Newton–Raphson scheme such as eigenvector following. For reactions like the alanine dipeptide rearrangement, for which the linearly interpolated guess is far from the MEP, the growing string method can facilitate the process of quickly obtaining a good saddle point estimate. On the two dimensional Muller–Brown surface, the distance from the true saddle point is a sufficient measure of the quality of the saddle point estimate. For the 66-dimensional alanine dipeptide, finding the true saddle point from the estimated saddle point depends on more than the distance between the two geometries. Most estimates of the saddle point have more than one imaginary frequency. In fact, the most unstable mode at the saddle estimate often does not even correspond to the path tangent.

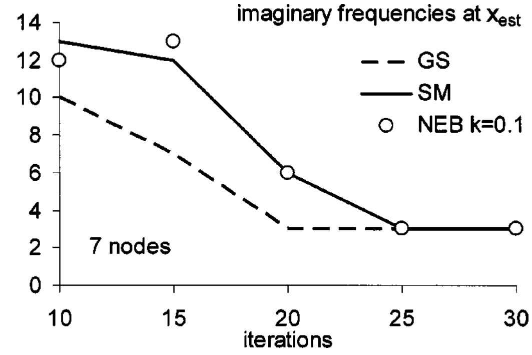

FIG. 10. The number of imaginary frequencies at the interpolated saddle point estimate as a function of the number of iterations. Extraneous imaginary frequencies are rapidly eliminated from the growing string saddle estimate. Note that while the string is growing, GS also requires fewer gradients per iteration than SM and NEB.

An important factor that influences whether the path tangent can be followed to a transition state using an eigenvector following scheme is the number of imaginary frequencies at the starting geometry. If the reaction coordinate direction is known, a few extraneous imaginary frequencies can usually be eliminated by mode following, but a large number of imaginary frequencies can be problematic. Figure 10 shows the number of imaginary frequencies at the estimated saddle point geometry from GS, SM, and NEB for the case of seven nodes.

Symmetric-rank-one (SR1) updates[11] with a maximum stepsize of 0.075 Bohrs were used for the Hessian during the pseudo-Newton–Raphson searches for the saddle point. When the initial Hessian of Eq. (11) was updated, but not periodically reset to the exact Hessian, all saddle point searches required more than 800 gradients or failed, except for three successful searches begun from GS estimates of the saddle point, which required between 200 and 300 total gradients. The pseudo-Newton–Raphson searches were more uniformly successful when started with an exact initial Hessian. The most reliable procedure was to periodically reset the Hessian to the exact Hessian during the eigenvector following procedure. The Hessian was reset to the exact Hessian every 20 iterations with SR1 updates used for the intermediate iterations. Resetting the Hessian periodically also allowed a larger maximum stepsize of 0.10 Bohrs.

Figure 11 shows the total number of gradients required to converge to a transition state as a function of the number

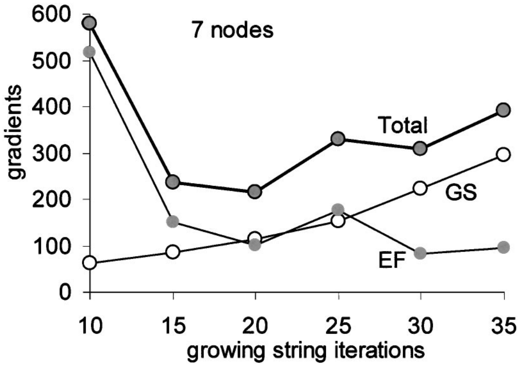

FIG. 11. Gradients (adjusted for periodic Hessian calculations) required to converge to the saddle point with a gradient norm of 0.0003 Ha/Bohr. The eigenvector following (EF) contribution to the total number of gradients generally decreases with increasing number of growing string method (GS) iterations.

of GS iterations used in generating the saddle point estimate. The number of gradients in the eigenvector following (EF) portion of the transition state search was scaled by 1.5 to account for the Hessian calculations which for this system take approximately 10 times as long as a gradient calculation, and therefore constitute one third of the CPU time.

The analysis of Fig. 11 was repeated to determine the minimum number of gradients required to find the saddle point for each method at 5, 6, 7, 8, and 9 nodes. Based on the saddle point estimates in Fig. 9, a uniform NEB spring constant of $k=0.1$ Ha/Bohr^2 was used for this comparison. Figure 12 summarizes the performance of the methods. The gradient count reported in Fig. 12 was adjusted to account for the Hessian calculations as described above. Generally, the GS method performed the best. SM and NEB performed similarly when the NEB spring constant was 0.1 Ha/Bohr^2.

The difference in performance is much more pronounced in mass-weighted coordinates where the atomic overlap along the linearly interpolated guess is more severe. After 100 iterations with a spring constant of 1.0 Ha/Bohr^2 amu, NEB calculations with 5, 6, and 8 nodes still had at least one node with a potential of 380 kcal/mol above the reactant configuration. The string method with a uniform parameterization density failed in a similar fashion. In contrast, at 100 iterations the maximum potential along the growing string was within 2 kcal/mol of the correct barrier. This indicates that only GS converges to the correct pathway in mass-weighted coordinates.

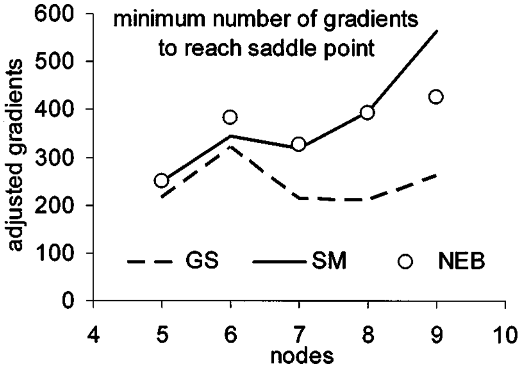

FIG. 12. The minimum number of gradients required to converge to the saddle point (interpolating scheme plus eigenvector following scheme) for strings with 5, 6, 7, 8, and 9 nodes.

CONCLUSIONS

Iterative interpolation schemes for finding the minimum energy pathway are powerful methods for locating transition states. The principle disadvantage of interpolation methods such as nudged elastic band and the string method is that they require an initial guess for the reaction pathway. In the absence of intuition about the reaction pathway, one usually uses a linearly interpolated initial pathway. This paper demonstrates that the speed, and even the viability of these methods, can depend strongly on the quality of the initial guess for the pathway. The growing string method avoids the need for an initial guess. The interpolated pathway grows inward from the product and reactant state toward the transition state as the growing ends converge. Eventually the separate ends join, and the growing string method iterations become identical to those of the string method.

Our results for the two-dimensional Muller–Brown surface do not clearly favor the growing string method over the nudged elastic band and string methods. However, for the alanine dipeptide, a clear hierarchy of methods emerges. The string method and the nudged elastic band method perform similarly when a spring constant of 0.1 Ha/Bohr^2 is used for nudged elastic band. The growing string method consistently outperforms both the string method and nudged elastic band.

The advantages of the growing string method are most apparent in the early stages of string optimization. In practical applications, methods like nudged elastic band and the string method generate a good estimate of the saddle point which is then reoptimized using a pseudo-Newton–Raphson scheme. For reactions such as alanine dipeptide rearrangement where the linearly interpolated guess is far from the MEP, the growing string method generates a suitable saddle point estimate much more quickly than nudged elastic band or the string method.

When mass-weighted coordinates were used to describe alanine dipeptide, the string method and the nudged elastic band method both failed to find the correct rearrangement pathway. In contrast, the growing string method rapidly converged to a realistic saddle point estimate. This clear difference in outcomes emphasizes the advantage of using the growing string method to eliminate the need for an initial guess. We hope it will prove to be useful for chemical applications.

APPENDIX: MINIMIZING $F[\varphi]$

The MEP minimizes the function

$$F(\varphi_0,\varphi_1,\ldots,\varphi_N)=\sum_k |\mathbf{f}_k^{\perp}|^2. \tag{A1}$$

The gradient of F is

$$\frac{\partial F}{\partial \varphi_j}=-\mathbf{K}_j\mathbf{f}_j^{\perp}+\sum_k (\nabla V_k\cdot\hat{\mathbf{t}}_k)\left[\frac{\partial \hat{\mathbf{t}}_k}{\partial \varphi_j}\right]\mathbf{f}_k^{\perp}. \tag{A2}$$

The last term describes how F increases as kinks grow on the string. Note the coefficient of the kinking term is the magnitude of the tangential force. This coefficient also appears in the stability criterion of Henkelman and Jonsson.[12] The tensor $[\partial\hat{\mathbf{t}}_k/\partial\boldsymbol{\varphi}_j]$ is the derivative of the tangent at the kth node with respect to the position of the jth node. It can be determined numerically by finite differencing the tangents with respect to small changes in the positions of each node on the string,

$$\left[\frac{\partial\hat{\mathbf{t}}_k}{\partial\varphi_j}\right]=\begin{bmatrix}\dfrac{\partial\hat{\mathbf{t}}_{k1}}{\partial\varphi_{j1}} & \cdots & \dfrac{\partial\hat{\mathbf{t}}_{k3N}}{\partial\varphi_{j1}}\\ \vdots & \iddots & \vdots\\ \dfrac{\partial\hat{\mathbf{t}}_{k1}}{\partial\varphi_{j3N}} & \cdots & \dfrac{\partial\hat{\mathbf{t}}_{k3N}}{\partial\varphi_{j3N}}\end{bmatrix}. \tag{A3}$$

If stability becomes a problem the kinking terms can be included.

If the nodes are far apart the kinking terms can be neglected, giving

$$\frac{\partial F}{\partial\varphi_j}\approx-\mathbf{K}_j\mathbf{f}_j^{\perp}. \tag{A4}$$

$F[\varphi]$ can have local minima where an eigenvalue of $\mathbf{K}$ becomes zero, but the perpendicular force along the corresponding mode is nonzero. This situation is analogous to the pathologies in square gradient minimization to find saddle points.[1] The approximate steepest descent direction may lead to these local minima

$$\mathbf{v}_{\mathrm{SD}}=\begin{bmatrix}[\mathbf{I}-\mathbf{t}_1\mathbf{t}_1^T]\mathbf{K}_1\mathbf{f}_1^{\perp}\\ \vdots\\ [\mathbf{I}-\mathbf{t}_n\mathbf{t}_n^T]\mathbf{K}_n\mathbf{f}_n^{\perp}\end{bmatrix}. \tag{A5}$$

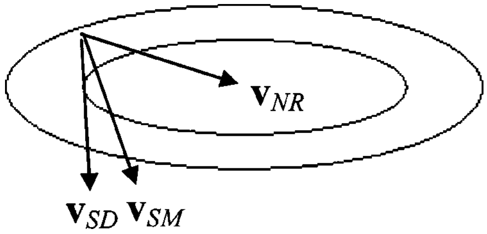

FIG. 13. Three search directions, $\mathbf{v}_{\mathrm{SD}}$, $\mathbf{v}_{\mathrm{SM}}$, and $\mathbf{v}_{\mathrm{NR}}$ on a hyperplane perpendicular to the MEP. The ellipses represent energy contours on the hyperplane. As the smaller Hessian eigenvalue on this surface tends to zero, the $\mathbf{v}_{\mathrm{SD}}$ search direction no longer points downward on the eigenvector corresponding to the zero mode. Note that the NEB search direction would appear identical to the SM search direction if projected onto this hyperplane, because the spring force is perpendicular to the hyperplane.

The approximate Newton–Raphson search direction resembles minimization of the potential on each hyperplane normal to the path. The NEB force direction interpolates between the steepest descent and Newton–Raphson directions for $F[\varphi]$. It has the desirable property that it is not susceptible to getting trapped in local minima of $F[\varphi]$. If we also assume negligible third derivatives of the potential energy

$$\left[\frac{\partial_2 F}{\partial\varphi_k\partial\varphi_j}\right]\approx\delta_{jk}\mathbf{K}_k[\mathbf{I}-\hat{\mathbf{t}}_k\hat{\mathbf{t}}_k^T]\mathbf{K}_k, \tag{A6}$$

so the second derivative matrix for F is approximately block diagonal with blocks given by Eq. (A6). Each block of the second derivative matrix is singular, but we can invert the matrix in the subspace of the hyperplanes to obtain

$$\left[\frac{\partial^2 F}{\partial\{\varphi_k\}\partial\{\varphi_k\}}\right]^{-1}\approx\begin{bmatrix}[\mathbf{I}-\mathbf{t}_1\mathbf{t}_1^T]\mathbf{K}_1^{-2}[\mathbf{I}-\mathbf{t}_1\mathbf{t}_1^T] & & \\ & \ddots & \\ & & [I-t_nt_n^T]\mathbf{K}_n^{-2}[I-t_nt_n^T]\end{bmatrix}. \tag{A7}$$

Thus, an approximate Newton–Raphson search direction for F is

$$\mathbf{v}_{\mathrm{NR}}=\begin{bmatrix}[\mathbf{I}-\mathbf{t}_1\mathbf{t}_1^T]\mathbf{K}_1^{-1}\mathbf{f}_l^{\perp}\\ \vdots\\ [\mathbf{I}-\mathbf{t}_n\mathbf{t}_n^T]\mathbf{K}_n^{-1}\mathbf{f}_n^{\perp}\end{bmatrix}. \tag{A8}$$

The Newton–Raphson search direction is approximately the direction for minimization on each independent hyperplane normal to the path.

The perpendicular force direction used in NEB and SM interpolates between the steepest descent and Newton–Raphson directions for $F[\varphi]$. The normal forces in NEB and SM give the search direction

$$\mathbf{v}_{\mathrm{SM}}=\begin{bmatrix}\mathbf{f}_l^{\perp}\\ \vdots\\ \mathbf{f}_n^{\perp}\end{bmatrix}. \tag{A9}$$

This direction has the desirable property that it does not get trapped in local minima of $F[\varphi]$. Note however, that there may be multiple paths that globally minimize $F[\varphi]$, and these paths may have very different barrier sizes.

Figure 13 shows the $\mathbf{v}_{\mathrm{NR}}$, $\mathbf{v}_{\mathrm{SM}}$, and $\mathbf{v}_{\mathrm{SD}}$ search directions on a hyperplane perpendicular to the minimum energy pathway. This figure suggests that $\mathbf{v}_{\mathrm{NR}}$ may be a better direction than $\mathbf{v}_{\mathrm{SM}}$. In the limit of negligible kinking terms and small third derivatives, the $\mathbf{v}_{\mathrm{NR}}$ search direction could be approximated with a BFGS or Broyden's. Method scheme to provide a superlinear version of the string method while still using only gradients.

ACKNOWLEDGMENTS

The authors appreciate helpful discussions with Professors Eric Vanden-Eijnden and Hannes Jonsson. The authors would like to thank Professor Martin Head-Gordon and Q-Chem (Ref. 36) for supplying the electronic structure code used in this work. This work was supported by BP and the National Science Foundation.

[1] F. Jensen, *Introduction to Computational Chemistry* (Wiley, Chichester, 1999).
[2] C. J. Cerjan and W. H. Miller, J. Chem. Phys. **75**, 2800 (1981).
[3] A. Banerjee, N. Adams, J. Simons, and R. Shepard, J. Phys. Chem. **89**, 52 (1985).
[4] J. Baker, J. Comput. Chem. **7**, 385 (1986).
[5] G. Henkelman and H. Jonsson, J. Chem. Phys. **111**, 7010 (1999).
[6] P. Bolhuis, D. Chandler, and C. Dellago, Annu. Rev. Phys. Chem. **59**, 291 (2002).
[7] P. Geissler, C. Dellago, P. G. Bolhuis, and D. Chandler, Adv. Chem. Phys. **123**, 1 (2002).
[8] R. Elber and M. Karplus, Chem. Phys. Lett. **139**, 375 (1987).
[9] P. Maragakis, S. A. Andreev, Y. Brumer, D. R. Reichman, and E. Kaxiras, J. Chem. Phys. **117**, 4651 (2002).
[10] S. Fischer and M. Karplus, Chem. Phys. Lett. **194**, 252 (1992).
[11] P. Y. Ayala and H. B. Schlegel, J. Chem. Phys. **107**, 375 (1997).
[12] G. Henkelman and H. Jonsson, J. Chem. Phys. **113**, 9978 (2000).
[13] G. Mills and H. Jonsson, Phys. Rev. Lett. **72**, 1124 (1994).
[14] Y. Xu and M. Mavrikakis, J. Phys. Chem. B **107**, 9298 (2003).
[15] M. Neurock, V. Pallassana, and R. A. van Santen, J. Am. Chem. Soc. **122**, 1150 (2000).
[16] R. Dittrich, T. Schrefl, D. Suess, W. Scholz, H. Forster, and J. Fidler, J. Magn. Magn. Mater. **250**, L12 (2002).
[17] J. W. Chu and B. L. Trout, J. Am. Chem. Soc. (to be published).
[18] J. Nocedal and S. J. Wright, *Numerical Optimization*. (Springer-Verlag, New York, 1999).
[19] W. Ren, Ph.D. thesis, New York University, 2002.
[20] W. E. W. Ren and E. Vanden-Eijnden, Phys. Rev. B **66**, 052301 (2002).
[21] M. Page and J. W. McIver, J. Chem. Phys. **88**, 922 (1988).
[22] J.-Q. Sun and K. Ruedenberg, J. Chem. Phys. **99**, 5269 (1993).
[23] C. Gonzalez and H. B. Schlegel, J. Chem. Phys. **95**, 5853 (1991).
[24] K. Muller and L. D. Brown, Theor. Chim. Acta **53**, 75 (1979).
[25] T. J. Marrone, M. K. Gilson, and J. A. McCammon, J. Phys. Chem. **100**, 1439 (1996).
[26] T. Lazaridis, D. J. Tobias, C. L. Brooks, and M. E. Paulaitis, J. Chem. Phys. **95**, 7612 (1991).
[27] B. M. Pettitt and M. Karplus, Chem. Phys. Lett. **121**, 194 (1985).
[28] D. J. Tobias and C. L. Brooks, J. Phys. Chem. **96**, 3864 (1992).
[29] A. Perczel, O. Farkas, I. Jakli, I. A. Topol, and I. G. Csizmadia, J. Comput. Chem. **24**, 1026 (2002).
[30] J. W. Chu, B. L. Trout, and B. R. Brooks, J. Chem. Phys. (in press).
[31] P. E. Smith, B. M. Pettitt, and M. Karplus, J. Phys. Chem. **97**, 6907 (1993).
[32] C. Bartels and M. Karplus, J. Comput. Chem. **18**, 1450 (1997).
[33] T. A. McCormick and D. Chandler, J. Phys. Chem. B **107**, 2796 (2003).
[34] G. Hummer and I. Kevrikidis, J. Chem. Phys. **118**, 10762 (2003).
[35] J. Apostolakis, P. Ferrara, and A. Caflisch, J. Chem. Phys. **110**, 2099 (1999).
[36] J. Kong *et al.*, J. Comput. Chem. **21**, 1532 (2000).