Journal of Mathematical Chemistry, Vol. 41, No. 1, January 2007 (© 2006)
DOI: 10.1007/s10910-006-9060-6

Fast marching method for calculating reactive trajectories for chemical reactions

Bijoy K. Dey,[1,2] Stuart Bothwell,[1] and Paul W. Ayers[1,*]

[1] *Department of Chemistry, McMaster University, 1280 Main St. West, Hamilton, ON, Canada*
[2] *Institut für Theoretische Physik, Freie Universität Berlin, Arimallee 14, 14195 Berlin, Germany*
E-mail: ayers@mcmaster.ca

Received 8 September 2005; revised 8 December 2005

We present a method for computing classical Newtonian trajectories that minimize the path length or transit time from reactant to product. Our approach is based on a generalization of the fast-marching method, which allows us to construct the solution of the Hamilton-Jacobi equation for the action that optimizes the desired quantity. The resulting "reactive paths" can be interpreted as reaction coordinates but, unlike more conventional choices, they contain dynamical information about the chemical system of interest.

KEY WORDS: fast-marching method, Hamilton-Jacobi equation, principle of least action, reaction path

1. Introduction

The chemical reaction path – defined as the sequence of molecular conformations linking reactants and products through reactive intermediates and transition states – is among the most important and fundamental concepts in chemistry. Given a reaction path, one can determine the mechanism of a reaction in atomic detail and infer information about the reaction's thermodynamic and kinetic characteristics.

Atomic detailed simulations of chemical dynamics can be performed either by computing molecular dynamics (MD) trajectories [1–3] or by computing the reaction path (RP) [4–23]. Computing molecular dynamics trajectories is essentially an initial value problem: starting with some initial momentum vector (usually chosen at random from a Boltzman distribution) and initial position vector (usually chosen to be a conformation corresponding to the reactants), one solves the Newton's equation of motion for the position coordinates as a function of time. Dynamical properties can then be extracted from the simulations.

*Corresponding author.

However, MD simulations are limited to time-scales much shorter than the rate of a typical chemical reaction. This is a serious drawback of MD simulation, and it is often difficult to find MD trajectories linking a specified initial (reactant) and final (product) configuration. However, because the actual time a trajectory spends crossing from the reactant to the product well is usually much shorter than the total length of the trajectory, carefully choosing the initial momentum so that one finds a "ballistic" trajectory can overcome the time-scale problem.

The second approach is posed as a two-point boundary value problem where an initial and a final configuration are given and one seeks out a path connecting the two configurations. These boundary points can be specified as a reactant and product configuration or a transition state configuration and reactant (product) configuration (as in the intrinsic reaction coordinate [6–8]). The calculated path provides a qualitative description of the structural changes as function of the reaction coordinate. Parameterizing the reaction path defines the reaction coordinate $\mathbf{Q}(s) = [q_1(s), \ldots, q_N(s)]^T$ where s is the parameter. Although the reaction path approach is immune from the time scale limitation of MD simulation, the ruggedness of the potential energy can severely limit the approach since most algorithms to compute RP rely on the explicit use of a spacial step. In addition, because using reaction-path methods require approximate structures for the reactant, product, and key intermediates, these methods are not directly applicable to the reactions where theoretical approaches would be most helpful – that is, reactions where even the coarsest mechanistic details are not known.

Whereas dynamical properties can be easily extracted from the MD trajectories by averaging over time, the computed reaction paths are difficult to convert to dynamically useful properties. (This would not be true of course, if one ran many reactive trajectories and constructed an ensemble. But this is not commonly done.) The importance of the reaction path comes in determining the reaction rate [24, 25] where the reaction paths describe the mechanism of the reaction and provide critical informations about the reaction intermediates, transition states and barrier heights. Reaction kinetics are based either implicitly (transition state theory [4]) or explicitly (variational transition state theory [4]) on the knowledge of the reaction paths. These theories require only the local information about the potential energy surface (PES) along the reaction path. (This, partially overcomes the dimension problem for a medium-sized or large molecules, since it is impossible to fully calculate their PES).

Both MD simulation and RP approach are useful in chemical dynamics. Hence it is desirable that we find a computational method that could combine the merits of both approaches, so that a computed reaction path (which could be used in transition state theory) would be a solution of Newton's equations of motion for the system, thereby yielding dynamic information. The present paper is in this direction.

There are many algorithms for computing reaction paths. Among the more popular choices are (a) direct geometric search of the PES coupled with energy minimization [6,7,9–12] (they are the geometrically defined paths meaning that only properties of the PES are taken into account, but no dynamic behavior of the molecule is taken into consideration.), (b) the extremization of a functional (e.g., the mean first passage time [13,14], diffusive flux [15,16]) along the path, (c) constructing a trial path and subsequent refinement of it based on the optimization of a functional defining the optimal path [17–20,23], (d) calculating the optimal reaction path [21,22] by maximizing the conditional probability for moving between two different states defined in Langevin dynamics [26]. These methods require approximate knowledge of the initial, final, and transition states. In addition, these reaction paths are not Newtonian trajectories.

Ideally, we would like to start with the knowledge of only the reactants and determine the feasible product state, reactive intermediate(s) (if any), transition state(s) and the reaction mechanisms. We recently proposed an efficient method [27] for computing the reaction path by solving the Hamilton-Jacobi equation for the least-action (and a similar equation for least time) where only an initial configuration (reactant) is known. The key innovation was the use of the fast marching method [28–30], after which the reaction products and mechanism can be found from the evaluated least-action or the least-time level curves. However, the least-time reaction paths are not true trajectories in the Newtonian sense and the least-action paths are high-energy paths. In this paper we generalize the fast marching method to compute reaction paths that are trajectories, thereby obtaining reaction paths that are solutions to Newton's eqautions of motion.

The proposed method works by solving the classical Hamilton-Jacobi (HJ) equation for the action $S(\mathbf{Q}_0, \mathbf{Q})$ function corresponding to the transition from an initial configuration ($\mathbf{Q}_0$) to any other configuration ($\mathbf{Q}$) accessible classically. However, unlike our previous work, which constructed the least action solution to the HJ equation, here we wish to compute a stationary action solution and, in particular, the solution that takes us from $\mathbf{Q}_0$ to $\mathbf{Q}$ in the least time. To accomplish this, from the action function calculated over the discrete grid we evaluate the time function $\tau_n(\mathbf{Q}_0, \mathbf{Q})$ that follows the ray that emanates from $\mathbf{Q}_0$ and ends at $\mathbf{Q}$. A combination of the upwind difference scheme, time sorting (these two define a modified fast marching method, modified because here the sorting is performed based on least-time instead of least-action) and ray tracing (this is used to evaluate the function τ_n) evaluates the action and least-time over the entire grid. The reaction paths are then calculated from the level curves of the action following the direction of unit (outward) normal $-\nabla S/|\nabla S|$ for points on the action level curves (this is termed as back-tracing). Since the action satisfies the HJ equation, the reaction paths constructed by this method are true trajectories and can be obtained, for example, by solving Newton's equations of motion for the forward and backward trajectories whenever the initial position and momentum vector (∇S), is defined on the reaction path. In the next section,

we will present the generalized fast-marching method in more detail, expounding on each facet of the algorithm we have just sketched. Section 3 then presents numerical examples, while section 4 concludes.

2. Theory

2.1. Hamilton-Jacobi equation

The classical action function for a path from configuration $\mathbf{Q}_0$ at time t_0 to configuration $\mathbf{Q}_f$ at time t_f is given by

$$S(\mathbf{Q}_0(t_0), \mathbf{Q}_f(t_f)) = \int_{\mathbf{Q}_0}^{\mathbf{Q}_f} \mathbf{P} \cdot \mathrm{d}\mathbf{Q}, \tag{1}$$

where $\mathbf{Q} = (q_1, q_2, \ldots, q_N)^T$ ($q_i = \sqrt{m_i} x_i$ denotes the atomic positions in mass-weighted coordinates) and $\mathbf{P}$ denotes the corresponding momentum vector. Equation (1) is not of much practical utility since one must know the path along which to evaluate the integral. However, a differential equation for the action can be easily derived from equation (1); it is this equation that we solve using the fast marching method [27–30]. First, rewrite equation (1) as

$$S(\mathbf{Q}_0(t_0), \mathbf{Q}_f(t_f)) = 2\int_{\mathbf{Q}_0}^{\mathbf{Q}_f} T(\mathbf{Q}(t))\mathrm{dt} = 2\int_{\mathbf{Q}_0}^{\mathbf{Q}_f} (E - V(\mathbf{Q}(t)))\mathrm{dt}. \tag{2}$$

In deriving equation (2), we used the relations $T = 1/2\mathbf{P}^T\mathbf{M}^{-1}\mathbf{P}$ for the kinetic energy and $\mathbf{P} = \mathbf{M}d\mathbf{Q}/dt$ for the momentum vector, where $\mathbf{M}$ is the mass matrix, V($\mathbf{Q}$) is the potential function, and E is the total energy of the system. The time variable in the above equation is not an independent variable because it depends on the path. A rigorous definition of path can be made by parameterizing the curve using the line element $\mathbf{s}(\Theta)$, for the trajectory (Θ denotes the parameterization of the path.) [31, 32]. The time it takes for a particle to cross a given point on the curve is related to the velocity, $\mathrm{d}\mathbf{s}(\Theta)/\mathrm{d}t$ at that point. At any point on the curve $2T = (\mathrm{d}\mathbf{s}/\mathrm{d}t)^2$,

$$\mathrm{d}t = \frac{|\mathrm{d}\mathbf{s}(\Theta)|}{\sqrt{2(E - V(\mathbf{Q}(\mathbf{s})))}}. \tag{3}$$

Since $|\mathrm{d}\mathbf{s}| = \sqrt{\sum(\mathrm{d}s_i)^2}$ and $\mathrm{d}s_i = (\partial s_i/\partial\Theta)\mathrm{d}\Theta$ we obtain

$$\mathrm{d}t = \frac{|\nabla\mathbf{s}(\Theta)|}{\sqrt{2(E - V(\mathbf{Q}(\mathbf{s})))}}\mathrm{d}\Theta, \tag{4}$$

where $|\nabla \mathbf{s}(\Theta)| = \sqrt{\sum_i (\partial s_i / \partial \Theta)^2}$ and s_i corresponds to the component of $\mathbf{s}$ along q_i. Equation (2) now is rewritten as

$$S(\mathbf{Q}_0(\Theta_0), \mathbf{Q}_f(\Theta_f)) = \int_{\Theta_0}^{\Theta_f} \frac{|\nabla \mathbf{s}(\Theta)|}{(2(E - V(\mathbf{Q}(\mathbf{s}))))^{-1/2}} \mathrm{d}\Theta \tag{5}$$

One immediately obtains the differential form for the above integral equation (equation (5)) (see ref. [33]). (From now on, we simply write $\mathrm{S}(\mathbf{Q})$ for $S(\mathbf{Q}_0(\Theta_0), \mathbf{Q}_f(\Theta_f))$ for a fixed initial configuration.)

$$\frac{|\nabla S(\mathbf{Q})|}{\sqrt{2(E - V(\mathbf{Q}))}} = 1, \quad \text{or} \quad \nabla S(\mathbf{Q}) \cdot \nabla S(\mathbf{Q}) = 2(E - V(\mathbf{Q})). \tag{6}$$

This is the Hamilton-Jacobi equation for the action, where ∇S is the generalized momentum vector, $\mathbf{p} = \nabla S$.

Once the path has been determined we evaluate the change in various quantities along the path. For example,

$$\tau_n(\mathbf{Q}_0(t_0), \mathbf{Q}_f(t_f)) = \int_{\Theta_0}^{\Theta_f} |\nabla \mathbf{s}(\Theta)| \left(\sqrt{(2(E - V(\mathbf{Q}(\mathbf{s}))))}\right)^n \mathrm{d}\Theta. \tag{7}$$

The corresponding differential equation is given by

$$\nabla \tau_n(\mathbf{Q}) \cdot \nabla \tau_n(\mathbf{Q}) = (2(E - V(\mathbf{Q})))^n, \tag{8}$$

where n can be any real or integer value. Thus, a series of function can be evaluated along the path for different values of n; e.g., for $n = -1$ the function is the travel time from $\mathbf{Q}_0$ to $\mathbf{Q}_f$ (cf. equation (3)). Some other values of n also have special significance. The $n = 0$ case of equation (7) clearly corresponds to arc-length, so solving equation (8) with $n = 0$ gives the least-distance path. (The level curves of τ_0 are spheres.) The $n = -1$ case corresponds to least-time, so solving equation(8) with $n = -1$ gives brachistochrones. The $n \to -\infty$ case corresponds to least-potential, so solving equation (8) with $n \to -\infty$ gives the minimum energy path (intrinsic reaction coordinate). Although the above formulation is only valid for mass-weighted Cartesian coordinates, one can easily derive the corresponding equations for a general curvilinear coordinate system.

The Hamilton-Jacobi equation (equation (3)) describes a wave front propagating [30] with the local "speed" $1/\sqrt{2(E - V)}$. Wave fronts of the action are represented as level curves, $\Gamma_S(a)$, defined as the set of all $\mathbf{Q} \in R^N$ for $S(\mathbf{Q}) = a$. Symbolically, $\Gamma_S(a) = \{\mathbf{Q} \in R^N; S(\mathbf{Q}) = a\}$. The direction of propagation of the wave front is always perpendicular to the front (along the outward-pointing normal of the level curve) and the speed of a given surface element of the wave front is given by the local value of the speed function, $1/\sqrt{2(E - V(\mathbf{Q}))}$. Sometimes it is more useful to explain results in terms of the "slowness" function, which is simply the reciprocal of the "speed" function.

2.2. Generalized fast marching method

The fast marching method (FMM) [28–30] finds the solution of the eikonal equation, $|\nabla U|/F = 1$, that minimizes $U(\mathbf{Q}) - U(\mathbf{Q}_0)$, where $\mathbf{Q}_0$ is the initial, or source, point. The generalized fast marching method (FMM), also solves the eikonal equation, $|\nabla U|/F = 1$, but instead of constructing the minimum $U(\mathbf{Q}) - U(\mathbf{Q}_0)$ solution, constructs the solution that minimizes $\tau_n(\mathbf{Q}) - \tau_n(\mathbf{Q}_0)$, where $\tau_n(\mathbf{Q})$ is given by equation (7). Clearly, when $\tau_n = U$ the method reduces to the usual FMM.

Here we describe how the generalized FMM algorithm can be used to compute reaction paths that are true molecular dynamics trajectories. We present the case where the configuration of the molecular system is given by a two-dimensional vector, that is, $\mathbf{Q} = [q_1, q_2]^T = [x, y]^T$. (The generalization to an N-dimensional system is straightforward.) The eikonal equation is equivalent to the Hamilton-Jacobi equation (equation (6)), which we write in a general form as

$$\alpha \left(\frac{\partial S}{\partial x}\right)^2 + \beta \left(\frac{\partial S}{\partial y}\right)^2 = 2[E - V(x, y)], \tag{9}$$

where $\alpha = \beta = 1$ for equation (6). The generalized FMM algorithm can be described in three steps: initialization, launching and termination.

2.2.1. Initialization

(a) Discretize the 2-D system as $x_1, \ldots, x_{N_1}, y_1, \ldots, y_{N_2}$ where $x_i = x_L + i \times \mathrm{d}x$ and $y_i = y_L + i \times \mathrm{d}y$, and $\mathrm{d}x$ and $\mathrm{d}y$ denote the step size along the x and y coordinates respectively. The variables $\mathrm{d}x$ and $\mathrm{d}y$ are given as $\mathrm{d}x = (x_R - x_L)/N_1$ and $\mathrm{d}y = (y_R - y_L)/N_2$ respectively, with $(x_L, x_R) \times (y_L, y_R)$ defining the domain where generalized FMM is performed.

(b) Define an initial level curve $\Gamma_S(0)$ as a fixed point $\mathbf{Q}_0 = [x^0, y^0]^T$ representing an initial configuration of the reacting system; here $\mathrm{S} = 0$ and $\tau_n = 0$. Find the point, (i_0, j_0), that is closest to the desired initial configuration, $\mathbf{Q}_0$. For other points, S and τ (for simplicity we ignore the subscript n of τ_n) are assigned a large number.

(c) Construct the first narrow band (see figure 1) (Υ_1) around the point (i_0, j_0). This narrow-band is comprised of eight neighboring points, $\Upsilon_1 : \{(i_0 - 1, j_0 - 1), (i_0 - 1, j_0), (i_0 - 1, j_0 + 1), (i_0, j_0 - 1), (i_0, j_0 + 1), (i_0 + 1, j_0 - 1), (i_0 + 1, j_0), (i_0 + 1, j_0 + 1)\}$. (These eight points are shown as cross marks in figure 1.) The values of S (or τ) in the first narrow-band are assumed known or are calculated from equation (5) (for τ equation (7)) by approximating the path-integrals.

(d) Tag different points as *alive*, *close* and *far* as follows

$$\textit{Tagged points} = \begin{cases} \textit{alive}, & \text{for source} \\ \textit{close}, & \text{for first narrow-band}(\Upsilon_1) \\ \textit{far}, & \text{for others} \end{cases}.$$

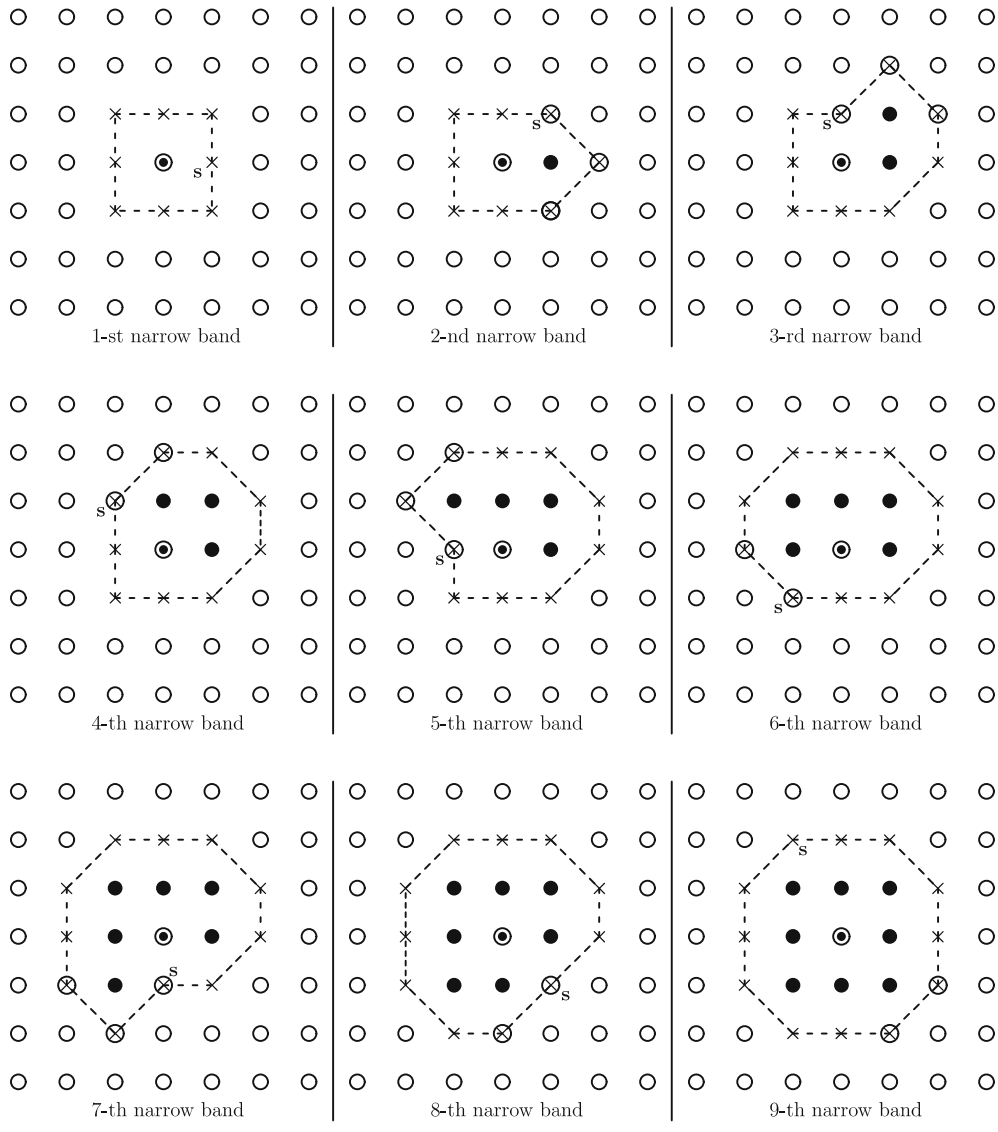

Figure 1. Schematic diagram depicting narrow bands during nine stages. The grid points (crosses and crosses in a circle) on the broken line constitute the narrow band. The first narrow band (first column, first row) is the one obtained at the initialization. Here the grid points with open circle, cross, cross in the circle, black dot and black dot in the circle represent *far* points, *close* points where the function value is not updated, *close* points where the function value will be updated, *alive* points, and the initial point (reactant conformation). The label "s" refers to the point with the smallest function value among the *close* points. At the end of every stage the point labeled "s" is tagged as an *alive* point.

Thus, we have one *alive* point, eight *close* points and $(N_1 \times N_2 - 9)$ *far* points in the initialization step.

2.2.2. Launching

The launching step expands the narrow-band beyond Υ_1, successively adding *far* points to *close* and *close* points to *alive*. This is done as follows:

(a) Select the *trial* point (i_t, j_t) from Υ_1 for which the value of τ is the smallest (in figure 1, label s denotes the trial point with the smallest value of τ) and tag it as *alive*. (In the conventional FMM [30] the *trial* point is the one for which the value of S is the smallest. [27] Here, however, we wish to construct least-τ curves (and not least-action curves), and so we take the point with the smallest value of τ instead.)
(b) Construct the next narrow-band, Υ_2. Υ_2 consists of all the points in Υ_1 except the *trial* point we just added to *alive*; in addition, Υ_2 includes any of the four nearest neighbors that are classified as *far* points. (Points that are nearest-neighbors of the *trial* point but which are already classified as *alive* are not included. Points that are nearest-neighbors of (i_t, j_t) that are already classified as *close* are "inherited" by Υ_2 because they are part of Υ_1. The points in Υ_2 are the *close* points. A schematic of the second narrow band is shown in the first row and second column of figure 1. Here the *trial* point is tagged as *alive*, marked as black dot, and removed from first narrow band. As seen in figure 1 among the four neighbors of this *trial* point, one is *alive*, one is *far* and two are *close*. Hence in this example, the second narrow band consists of $8 - 1 + 1 = 8$ *close* points (figure 1).
(c) Update S and τ at $(i_t - 1, j_t)$, $(i_t + 1, j_t)$, $(i_t, j_t - 1)$, $(i_t, j_t + 1)$ if they are *far* or *close* points. The updating procedure is described in section 2.3.
(d) Iterate (a) and (b) and (c) by replacing Υ_{k-1} of step (a) by Υ_k of step (b), where k indicates iteration number.

2.2.3. Termination

(a) The process terminates when Υ_M is totally empty. At this stage every grid points is either *alive* or classically forbidden. As an alternative, we could designate some stopping condition. For example, when we know the product of the chemical reaction, we might stop once the molecular conformation corresponding to this product has been added to *alive*. Alternatively, we could stop the procedure when we have located a well in the potential energy surface with a sufficiently low energy, as this would probably indicate the formation of the product or, at least, the formation of a very stable reaction intermediate, which might be of independent interest.

As described above, the *trial* point is chosen to have the smallest τ among all values in a given Υ_k, $k = 1, \ldots, M$. This implies that the values of τ in a narrow band, Υ_k, should be sorted from the smallest to the largest. This can be

computationally expensive since the sorting needs to be done for each narrow band and both the number of points in each narrow band and the number of narrow bands are usually large numbers. When a binary tree sorting algorithm is applied [34], the cost of sorting the points is $O(\log_2 N_B)$ where N_B is the number of grid points in the narrow band. The total cost of the algorithm is thus of order $N \log_2 N$, where N is the number of conformations (in this implementation, the number of grid points). This is (at least to within a factor of $K \log_2 N$, where K is a constant) the least cost attainable for this problem.

2.3. *Updating* (*Step* 2(*c*))

2.3.1. *Updating S*

Now we elaborate on the updating of S and τ mentioned in Step 2 (c). For this, we consider a point (i,j) where S and τ will be recomputed. As discussed in the previous section and illustrated in figure 1, such points will be (a) adjacent to the *trial* point (i.e., one of the four points $(i_t \pm 1, j_t)$, $(i_t, j_t \pm 1)$), (b) classically allowed (so $V_{ij} < E$), and (c) downwind from (i_t, j_t) (i.e., (i, j) is *close*, and not *alive*). As mentioned in the introduction, in the updating procedure we first compute S by solving the Hamilton-Jacobi equation (equation (6)) and then we use the information about S and its gradient to calculate τ using a ray-tracing scheme.

To update the action, S, we rewrite the Hamilton-Jacobi equation in discrete form:

$$\alpha(\hat{D}_x S_{ij})^2 + \beta(\hat{D}_y S_{ij})^2 = 2[E - V(x_i, y_j)]. \tag{10}$$

Here S_{ij} is the action at (i, j), and $\hat{D}_x$ and $\hat{D}_y$ are the upwind finite difference formulas [30], defined as

$$|\hat{D}_x S_{ij}| = \max(D_x^- S_{ij}, -D_x^+ S_{ij}, 0), \tag{11}$$

$$|\hat{D}_y S_{ij}| = \max(D_y^- S_{ij}, -D_y^+ S_{ij}, 0). \tag{12}$$

Here $D_x^{\pm}$ and $D_y^{\pm}$ are the forward (+) and backward (−) difference operators defined as

$$D_x^{\pm} S_{ij} = \pm \frac{S_{i\pm1,j} - S_{ij}}{\Delta x} \tag{13}$$

$$D_y^{\pm} S_{ij} = \pm \frac{S_{i,j\pm1} - S_{ij}}{\Delta y}. \tag{14}$$

Only four points appear when solving the above HJ equation (equation (10)) which are $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ and $(i, j + 1)$. These four

points connected by grid line segments form four quadrants (a triangle) as Δ_1: $[(i+1, j), (i, j+1), (i, j)]$, Δ_2: $[(i-1, j), (i, j+1), (i, j)]$, Δ_3: $[(i-1, j), (i, j-1), (i, j)]$ and Δ_4: $[(i+1, j), (i, j-1), (i, j)]$. The conventional FMM solves the quadratic equation (equation (10)) given by each quadrant and then chooses the smallest of these values as the updated value of S. That is,

$$S_{ij}^{updated} = \min[S_{ij}(\Delta_1), S_{ij}(\Delta_2), S_{ij}(\Delta_3), S_{ij}(\Delta_4)], \tag{15}$$

where $S_{ij}(\Delta_k)$, $k = 1, \ldots, 4$ is the largest real solution of the quadratic equation for the kth quadrant. Often, some quadrants will not contribute to the updating of S because the upwind difference formulas (equations (11) and (12)) require that only quadrants with vertices that are upwind to (i, j) can affect the value of the action. That is, because only vertices whose action is smaller than the action at (i, j) can be used in the updating procedure, sometimes one or more of the quadrants does not furnish a valid update. (An alternative method, sometimes preferable, is to only allow *alive* points to be used for updating the value. This gives the same results as the procedure presented here, but the binary sort procedure takes longer because the points are more "out of order" with this method than they are with the usual procedure).

It now remains to describe how the value of the action is updated from each quadrant. Taking the first quadrant as an example, we see that there are two cases:

(a) if only $(i+1, j)$ vertex is upwind, then we use the largest solution of

$$\alpha \left(\frac{S - S_{i+1,j}}{\Delta x} \right)^2 = 2[E - V(x_i, y_j)] \tag{16}$$

for the updated action.

(b) If both $(i+1, j)$ and $(i, j+1)$ are upwind, then the updated S is given by the largest real solution of

$$\alpha \left(\frac{S - S_{i+1,j}}{\Delta x} \right)^2 + \beta \left(\frac{S - S_{i,j+1}}{\Delta y} \right)^2 = 2[E - V(x_i, y_j)]. \tag{17}$$

2.3.2. Updating τ

In the "generalized" FMM, we wish to update the action so that some other quantity, τ, is minimized. Our notation is as follows: the action at (i, j) is computed using the value from at most two upwind points; in figure 2, the point at which we are computing the action is point C and the upwind points are A and B). For example, for quadrant Δ_1, A is $(i+1, j)$ and B is $(i, j+1)$. In figure 2 $\mathbf{Q}_0$ is a fixed initial configuration. A ray emanates from the initial point, passes through the line segment AB defined by the points $[x_A, y_A]^T$ and $[x_B, y_B]^T$ and passes through the point C (where the value of τ will be calculated). We assume

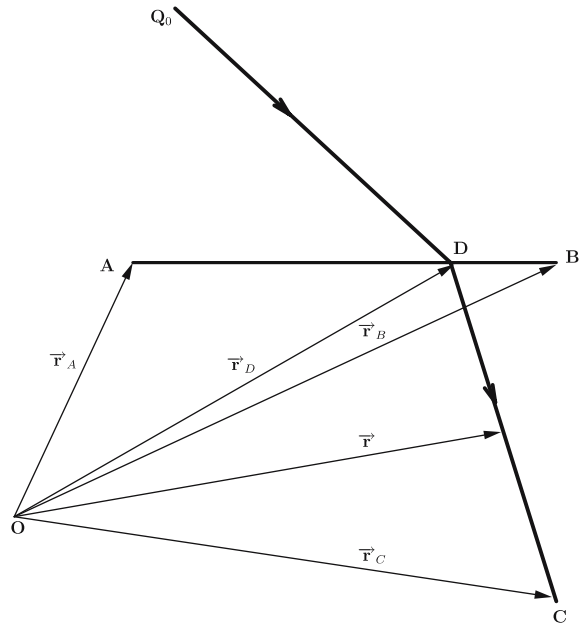

Figure 2. Schematic diagram for evaluating τ. Here A and B are *near* and/or *alive* points and C is a near (but not *alive*) point. The ray from D to C is a stationary-action path in the finite difference representation.

that the ray from $\mathbf{Q}_0$ crosses the line segment AB at the point D. Using Taylor series expansion (up to first order) we now write τ at C as

$$\tau_C = \tau_D + |CD||\nabla\tau_C. \tag{18}$$

Using equation (8) we rewrite this as

$$\tau_C = \tau_D + |CD| \left(\sqrt{2(E - V_C)}\right)^n, \tag{19}$$

where V_C is the potential at the point D. Thus, from the value of τ_D at point D and the Euclidean length of CD line segment we can calculate τ_C. To do this we parameterize the function value at D by a linear interpolation formula

$$\tau_D = w\tau_A + (1 - w)\tau_B \quad (0 \leqslant w \leqslant 1). \tag{20}$$

Define $\vec{r}$ as the vector (figure 2) tracing the point in the line segment CD so that when the point is D we have $\vec{r} = \vec{r}_D$. This vector $\vec{r}$ can be expressed as

$$\vec{r} = \vec{r}_C + l\hat{g}_C, \tag{21}$$

where l defines the distance between C and D ($l = |CD|$) and $\hat{g}_C = [g_C^x, g_C^y]^T$ with $g_C^x = -D_x S/|\nabla S||_C$ and $g_C^y = -D_y S/|\nabla S||_C$ is the local unit normal vector

along the line segment CD. (D_x and D_y are given by equations (13) and (14) respectively.) Since we have parameterized the line segment AB with respect to the parameter w the vector $\vec{r}_D$ is given by

$$\vec{r}_D = w\vec{r}_A + (1 - w)\vec{r}_B. \tag{22}$$

We now evaluate l and w so that $\vec{r} = \vec{r}_D$. This is done by solving the simultaneous equations

$$x_C + lg_C^x = wx_A + (1 - w)x_B \tag{23}$$

$$x_C + lg_C^y = wy_A + (1 - w)y_B. \tag{24}$$

Knowing w and l we can immediately compute τ_D and then τ_C from equation (18). Note that if either A or B has a larger value of the action than we find at C, then we should set w so that this point is not used to compute τ_C. Thus, if A is not upwind then $w = 0$ and $l = (y_B - y_C)/g_C^y$ and if B is not upwind then $w = 1$ and $l = (x_A - x_C)/g_C^x$.

In contrast to the preceding method for computing the action at the point (i, j), the generalized fast-marching method proceeds as follows. While previously (cf. section 2.3.1) we computed the value of the action from each quadrant, now we will compute both the value of the action and the value of τ from each quadrant. Also unlike the previous procedure, we select the update with the smallest value of τ (as opposed to the smallest value of S). This constructs a stationary-action solution (stationary-action because the path is selected so that the Hamilton-Jacobi finite-difference equation is satisfied) with least-τ (because both the tentative value from the search over quadrants and the heap search is done in a way that minimizes τ).

2.4. Back-tracing: finding true dynamical paths

Having obtained the stationary action and least-τ surfaces we calculate reaction paths connecting a final state configuration $\mathbf{Q}_f$ to $\mathbf{Q}_0$ by following the direction of a series of normal vectors, $-u\nabla S/|\nabla S|$, of length u. Thus, for calculating the path we calculate the vector $u\nabla S/|\nabla S|$ of length u at point $\mathbf{Q}_f$ followed by the subsequent evaluation of the action at $\mathbf{Q} = \mathbf{Q}_f - u\nabla S/|\nabla S|$, where at every step $\mathbf{Q}_f$ is replaced with $\mathbf{Q}$. This sequence of vectors of length u define the desired path. Choosing different values of n in equations (8) and (18) we can calculate different reaction paths, all of which correspond to true dynamical paths since they follow from solving the Hamilton-Jacobi equation. Therefore, the reaction paths computed by this generalized FMM can also be computed by

solving Newton's equation of motion

$$\ddot{Q} = -\mathbf{h}(\mathbf{Q}), \quad \mathbf{h}(\mathbf{Q}) = \left[\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}\right]^T \tag{25}$$

with the initial conditions (the initial conditions are defined at the final configuration for the path $\mathbf{Q}_f \longrightarrow \mathbf{Q}_0$)

$$\mathbf{Q}(t_f) = \mathbf{Q}_f, \quad \text{and} \quad \dot{\mathbf{Q}}(t_f) = |\nabla S|_{\mathbf{Q}_f}. \tag{26}$$

From these initial conditions, MD trajectories, $\mathbf{Q}$(t), are easily calculated using standard integrators like the velocity-Verlet algorithm. As we shall see in the next section, the trajectories obtained by solving Newton's equations match the reaction paths evaluated by the generalized FMM.

3. Numerical examples

3.1. Application to a four-well potential surface

We now examine the effectiveness of our method in computing true reaction paths for a two-dimensional potential surface (figure 3) which has four local minima $(\alpha_i, \beta_i), i = 1, \ldots, 4$. This potential was introduced as a nontrivial test for reaction path calculation in our previous work [27]. Here we are mainly interested in the reaction path (and the corresponding MD trajectory) between two neighboring local minima on the energy surface. Stringing together such paths provides a reaction path between any pair of minima on the energy surface. In figure 3, A and B are two extreme minima where $V_B < V_A$ and so A and B can be taken to represent reactant and product, respectively. C and D have higher energy than either A or B and serve as reactive intermediates. Any two neighboring minima are separated by a barrier which serves as the transition state for the neighboring minima. Thus, the potential has all the features necessary to demonstrate the effectiveness of the present method.

To start the generalized FMM and back-tracing we calculate the action and τ at the first narrow band by approximating the integrals $\int_{\mathbf{Q}_0}^{\Upsilon_1} \sqrt{2(E-V)}\mathrm{d}l$ and $\int_{\mathbf{Q}_0}^{\Upsilon_1} [\sqrt{2(E-V)}]^n \mathrm{d}l$, respectively, where the upper limit Υ_1 represents any point in the first narrow band. Once the initialization stage is complete, we proceed to the launching step where we construct the next level of narrow band by updating S and τ. This process continues until the last narrow band is completely empty, and S and τ are known over the classically allowed portion of the discrete grid. From the level curves of S and τ we then evaluate the reaction path. In figure 4 and 5 we have identified three different paths connecting two nearest local minima, viz., C $\longrightarrow$ A, D $\longrightarrow$ C and D $\longrightarrow$ B. The computation of these paths requires selecting a point (the final state configuration) in the potential energy

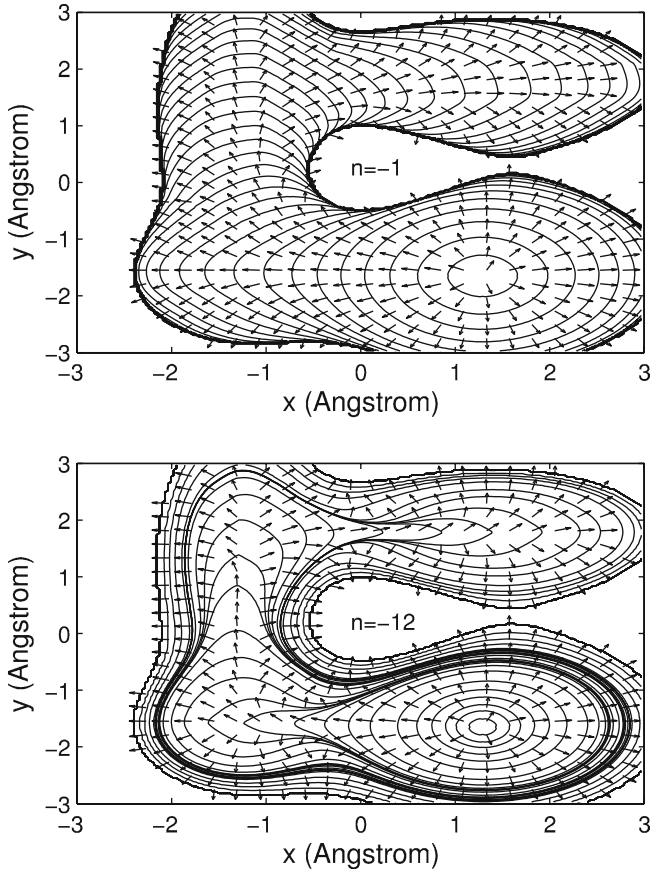

Figure 3. A contour plot of the potential that we employed as one of the demonstrations of the generalized FMM calculations. The potential is of the form: $V(x, y) = V_0 + a_0 e^{-(x-b_x)^2-(y-b_y)^2} - \sum_{l=1}^{4} a_i e^{-\sigma_i^x (x-\alpha_i)^2-\sigma_i^y (y-\beta_i)^2}$, where the parameters are $V_0 = 5.0$ kcal/mol, $a_0 = 0.6$ kcal/mol, $b_x = 0.1$ Å, $b_y = 0.1$ Å, $a_1 = 3.0$ kcal/mol, $a_2 = 1.5$ kcal/mol, $a_3 = 3.2$ kcal/mol, $a_4 = 2.0$ kcal/mol, $\sigma^x = [0.3, 1.0, 0.4, 1.0]^T$ Å^{-2}, $\sigma^y = [0.4, 1.0, 1.0, 0.1]^T$ Å^{-2}, $\alpha = [1.3, -1.5, 1.4, -1.3]^T$ Åand $\beta = [-1.6, -1.7, 1.8, 1.23]^T$ Å. The four local minima are A $(1.29, -1.65)$, B $(1.4, 1.78)$, C $(-1.29, -1.53)$ and D $(-1.17, 1.56)$ and T_1, T_2, T_3 and T_4 represent four transition states connecting the local minima.

surface and evaluating the normal vector $\nabla S/|\nabla S|$ at that point. ∇S was evaluated with the Shepard interpolation method [35, 36].

The paths in figure 4 are true least-time ($n = -1$ in equation (8)) paths because they correspond to solutions of Newton's equation of motion. Figure 4 shows that the paths calculated using the generalized FMM (upper panel) and Newton's equation (lower panel) are very similar. The same holds for the true least-τ ($n = -12$) reaction paths (figure 5): the generalized FMM paths are very similar to the MD paths. Thus, we can interpret the generalized FMM as

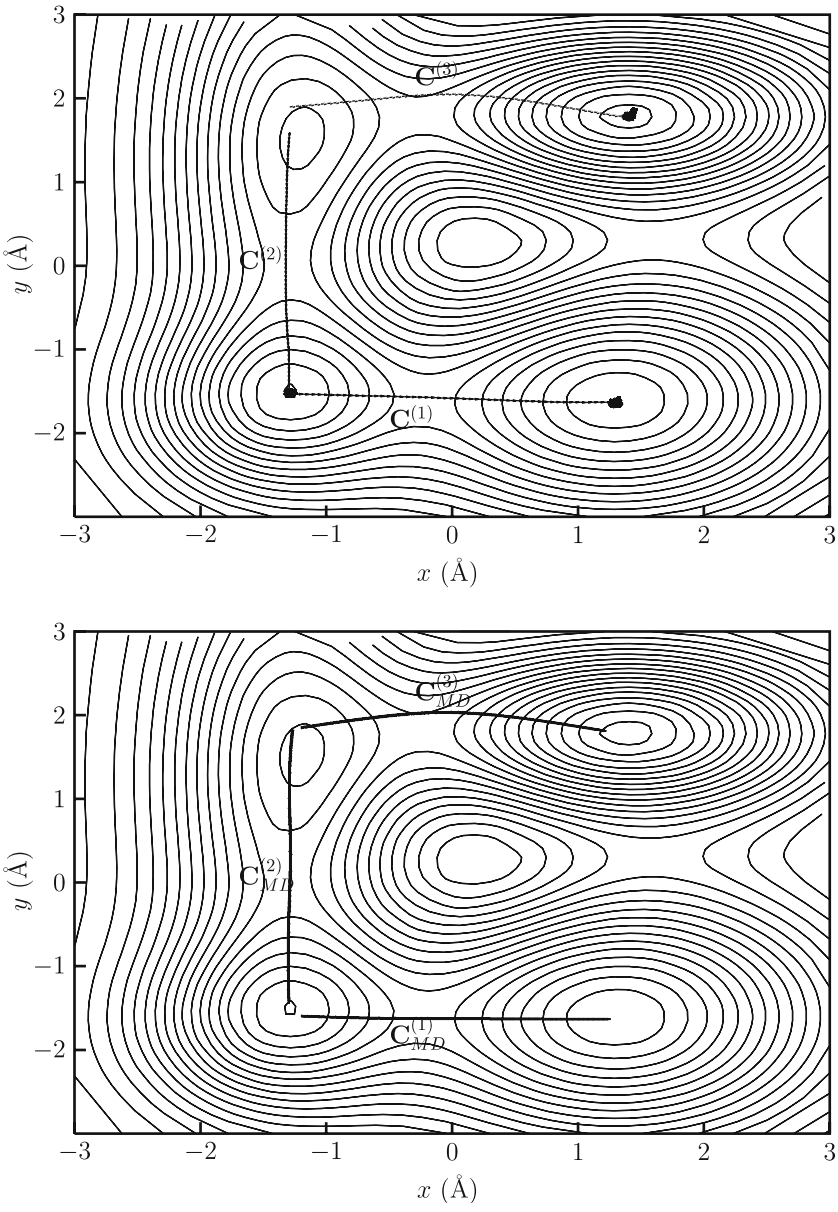

Figure 4. Reaction paths ($n = -1$ (equation (8))) at E = 4.0 kcal/mol calculated by the generalized FMM (top panel) and the MD trajectories (bottom panel) plotted on the potential energy surface. Reaction paths are labeled as $C^{(1)}$, $C^{(2)}$ and $C^{(3)}$ which correspond to the path C $\longrightarrow$ A, D $\longrightarrow$ C and D $\longrightarrow$ B (see figure 1). We have used the level curves corresponding to the initial points **A**, **C** and **B** for calculating the RPs $\mathbf{C}^{(1)}$, $\mathbf{C}^{(2)}$ and $\mathbf{C}^{(3)}$ respectively. The corresponding MD trajectories (bottom panel) are labeled as $\mathbf{C}^{(1)}_{MD}$, $\mathbf{C}^{(2)}_{MD}$ and $\mathbf{C}^{(3)}_{MD}$.

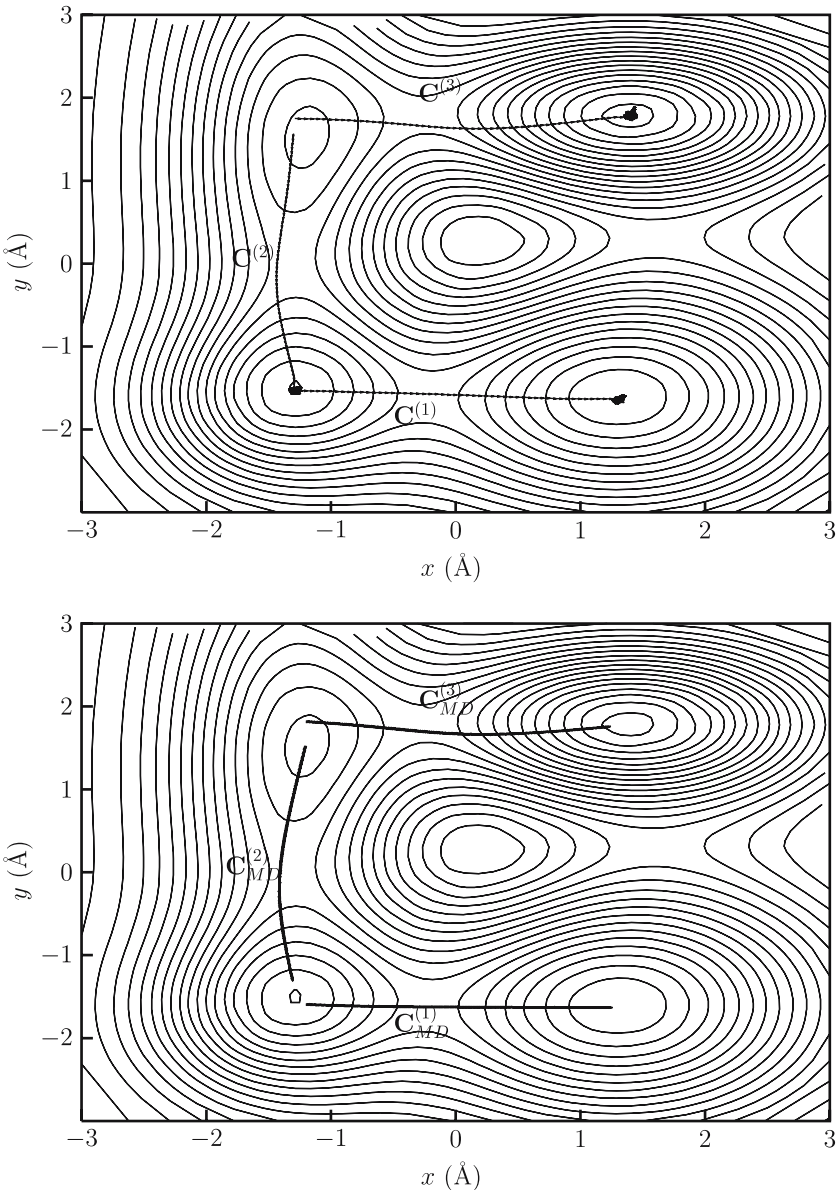

Figure 5. Reaction paths ($n = -12$ (equation (8))) at E = 4.0 kcal/mol calculated by the generalized FMM (top panel) and the MD trajectories (bottom panel) plotted on the potential energy surface. Path labels are as in figure 4.

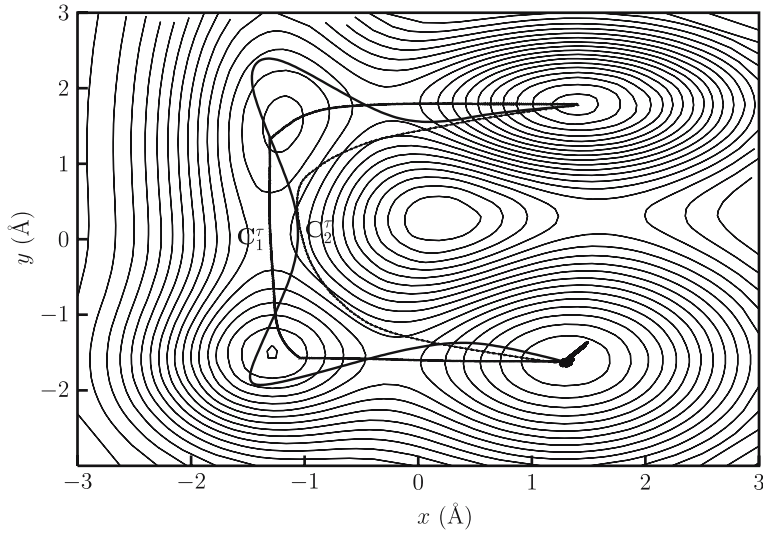

Figure 6. Least-τ reaction paths for $n = -1$ (least-time) (label $\mathbf{C}_1^\tau$) and $n = -12$ (least potential) (label $\mathbf{C}_2^\tau$) at E = 4.0 kcal/mol. Path $\mathbf{C}_1^\tau$ is related to the intrinsic reaction coordinate. The unlabeled path, C_3^τ, is a sketch of what a least-time trajectory that passed through multiple wells might look like.

a method for identifying the "optimum initial conditions" – the value of the position and the momentum that causes the MD trajectory to proceed from reactant to product in the least-τ manner. The MD paths shown in the lower panel of figures 4 and 5 were calculated by solving Newton's equations of motion with an initial position, $\mathbf{Q}(0)$ and momenta vector, $\mathbf{p}(0) = \nabla S(0)$ calculated on a point very close to $\mathbf{Q}_0$ (only the magnitude of ∇S is defined at the initial point).

As a by-product of the generalized FMM we calculate the least-time (C_1^τ in figure 6) and least-τ ($n = -12$) (C_2^τ in figure 6) paths which do not correspond to MD trajectories. These paths were calculated by following the normal $\nabla\tau/|\nabla\tau|$ on the level curves of τ. The least-time path (C_1^τ in figure 6) (brachistochrone) cuts the corner of the energy barrier in order to get to one point from the other in the potential energy surface. The least-τ path with the most negative n (e.g., $n = -12$ in our calculation for C_2^τ in figure 6) approaches the least-potential path and approximates the intrinsic reaction coordinate [37]. Figure 6 reveals that the least-τ ($n = -12$) path from A to B passes through the local potential minima (reaction intermediates, D and C) and the saddle points in the PES (transition states T_4, T_3 and T_1).

Reaction paths ($C^{(1)}$, $C^{(2)}$, $C^{(3)}$, C_1^τ, C_2^τ) calculated using the generalized FMM filter out the unwanted high frequency motions and help us identify the important coordinates associated with a given dynamical change. This concept of important coordinates has been frequently used by many workers to describe the long time-scale dynamics [38–40]. Filtering out the high frequency

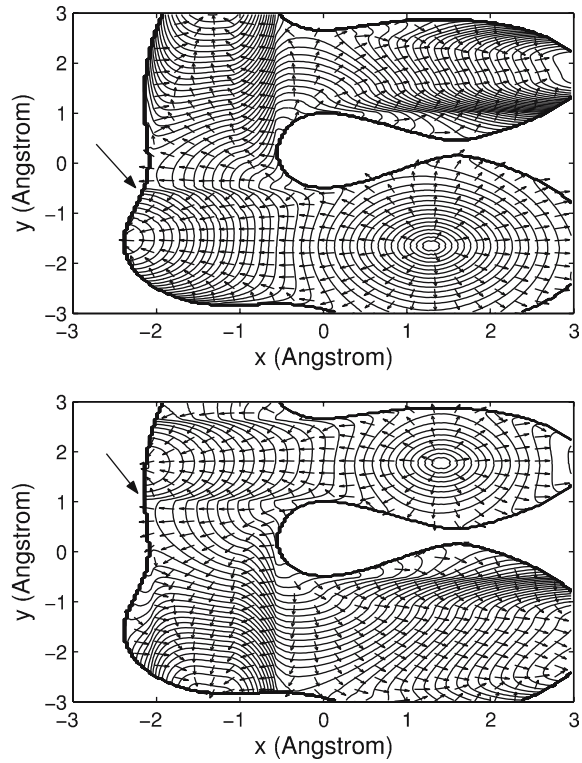

Figure 7. Action level curves for $n = -1$ at $E = 4.0$ kcal/mol and the normal vectors plotted in the (x,y) plane. The upper panel is for the initial state $\mathbf{Q}_0 = [1.29, -1.65]^T$ and the lower panel for $\mathbf{Q}_0 = [1.4, 1.78]^T$. The long vector line shows the crest formation in the level curves.

oscillations and creating a reduced-coordinate space for studying reaction events has always been a physically appealing approximation. For example, in bond stretching, the spatial deviation of bonds from their equilibrium position is comparatively small and during the time-scale of a reactive event these rapid oscillations nearly average out. Using the reaction paths described above one can expect to see some quantitative deviation of some of the properties, however, many of the properties related to kinetics (e.g., reaction rate, activation energy) can be easily calculated with these reaction paths since the computed paths are qualitatively similar to an average over MD trajectories.

In figures 4 and 5 we have shown the true reaction paths (that is, paths which can be obtained as the MD trajectories) which only connect the two nearest local minima through a transition state. In this example, the method is unable to find true reaction paths connecting more than two minima. To see why this is so, in figure 7 we show the action level curves at $E = 4.0$ kcal/mol for $n = -1$ (equation (8)) for two different initial points namely, $\mathbf{Q}_0 = [1.29, -1.65]^T$ (point A in figure 3) and $\mathbf{Q}_0 = [1.4, 1.78]^T$ (point B in figure 3). Using the projected

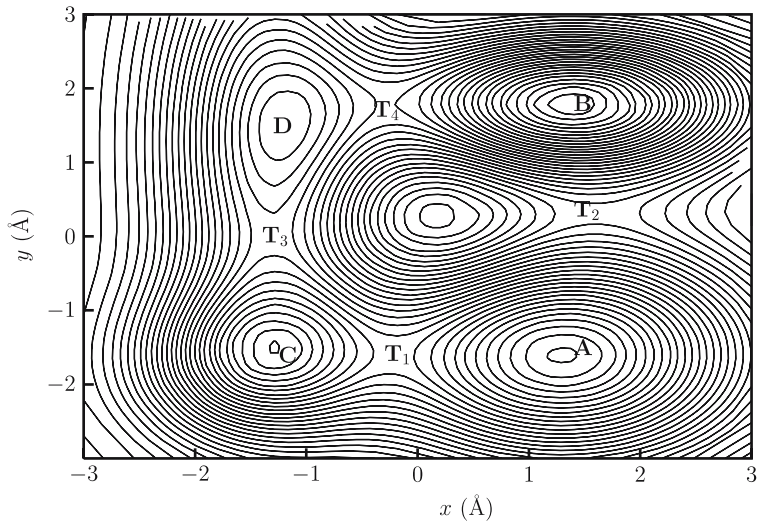

Figure 8. Least-τ level curves at E = 4.0 kcal/mol and the normal vectors plotted in the (x, y) plane. The curves are for the initial point $\mathbf{Q}_0 = [1.29, -1.65]^T$.

normal vectors in figure 7, one can visualize possible paths connecting any two points: start from the final state, and trace backwards (from tail to head) along the projected normal vectors, thereby connecting the final to the initial state. Using this procedure, it is easy to visualize paths connecting C $\longrightarrow$ A (upper panel of figure 7) and D $\longrightarrow$ B (lower panel of figure 7).

Sometimes, this procedure does not work. In some places, level curves of the action form a crest, and the tails of the projected normal vectors on one side of the crest do not connect with the heads of vectors on the other side of the crest. This indicates that there is no way to find a path across the crest. This situation does not arise in the least-τ level curves: as shown in figure 8 for $n = -1$ and $n = -12$, the least-τ paths connecting the points A and B (see figure 3 for the definition of points on the PES) are easily computed.

An apparent limitation of the present method is the inability to find paths connecting more than two local minima in the PES. This is attributed to the formation of a crest and the momenta mismatch across it, as shown in figure 7. The crest arises because of a weakness in the fast-marching algorithm. Our generalized fast-marching algorithm assumes that each configuration is visited exactly once, and assigns the τ value to that configuration accordingly. When the reaction path goes around a corner, like in figure 7, this is not true and we would need to keep track of multiple values of τ at each grid point. That is, we have a trajectory-crossing problem. Distinct least-time trajectories (each with the same initial point, but different final points) may cross. At the point where trajectories cross, there are multiple values of τ, one for each trajectory. The fast-marching

method only keeps track of the smallest value of τ, which makes it impossible to construct the other (and slower) trajectories.

As an example, consider the least-time trajectory from the initial point (A) to the final point (B); the path is sketched as path C_3^τ in figure 6 and the level curves for the action computed by the generalized fast-marching method are plotted in the upper panel of figure 7. In passing from A to B, the trajectory must pass through the first transition state (T_1), visit the repulsive wall at the bottom of the potential, go back through the vicinity of the first reactive intermediate (C), pass through the second transition state region (T_3), and hit the repulsive wall on the far left of the potential, which redirects the trajectory through the region of the second reactive intermediate (D) and over the final barrier (T_4) to the product well (B). In the vicinity of the first transition state, T_1, this path crosses the least-τ trajectory between the reactant (A) and the first reactive intermediate (C). The generalized fast-marching method only constructs the least-τ path to this "crossing point", so the path C_3^τ cannot be constructed with the present algorithm. (The values of τ along the C_3^τ trajectory are usually greater than those on other least-τ paths, because this trajectory visits high-potential regions when it "turns corners." Classical trajectories may be likened to vehicles without steering systems—the only way to change direction is via a "banked" curve, which is what the repulsive regions of the potential provide.) In order to reproduce such trajectories, one must be able to keep track of not only the least-τ value at each grid point, but also the values relevant for other trajectories.

An alternative perspective is directly relevant to the interpretation of the crest. Due to the need for trajectories to "swerve" in order to go around corners, it is possible that the least-τ paths to two adjacent grid points are qualitatively different. The crests in figure 7 occur where there is a discontinuity between "direct" trajectories to a given grid point and "indirect" (i.e., "bank shots") trajectories: on the lower side of the crest in figure 7a the trajectories are direct, on the upper side, the least-τ trajectories (which are not found by the present method) are indirect.

To extend the method, then, we need to include the possibility of such "indirect" trajectories. The method for doing this is conceptually simple, although it is so tedious that we have not implemented it. Denoting the path from $\mathbf{Q}_0$ to $\mathbf{Q}_f$ as $\mathbf{s}(\Theta)$, we minimize

$$\tau_n(\mathbf{s}) = \int_{\Theta_0}^{\Theta_f} |\nabla \mathbf{s}(\Theta)| \cdot [2(E - V(\mathbf{s}(\Theta)))]^{n/2} \, \mathrm{d}\Theta \qquad \text{(A)} \quad (27)$$

with respect to the path $\mathbf{s}(\Theta)$, subject to the constraint that the action,

$$S(\mathbf{s}) = \int_{\Theta_0}^{\Theta_f} \frac{|\nabla \mathbf{s}(\Theta)|}{[2(E - V(\mathbf{s}(\Theta)))]^{-1/2}} \mathrm{d}\Theta, \qquad \text{(B)} \quad (28)$$

is stationary. This is a non-standard minimization problem. The fast-marching technique for τ_n (as sketched above, with the only change being that τ_n is now determined by equation (27), instead of using equations (17)–(20)) will often give a good guess for $\mathbf{s}(\Theta)$ (excepting regions near crests, we expect that least-τ_n trajectories to adjacent points will be similar, and thus provide good "initial guesses" for $\mathbf{s}(\Theta)$). Still, the task of minimizing $\tau_n(\mathbf{s})$ subject to the stationary action constraint will not always be easy (it would probably be particularly challenging near the crests in figure 7) and it is not clear how one should formulate this problem computationally. Accordingly, this approach should not be regarded, at present, as anything more than a formal solution to the problem of finding least-τ_n trajectories. The prohibitive cost of such a technique (it would require evaluating and optimizing a number of path integrals, as opposed to the fast equation solving and sorting associated with fast-marching methods) suggests that such studies should not be lightly undertaken.

4. Discussion

The determination of chemical reaction paths is a central problem in mechanistic chemistry and chemical kinetics. In this paper, we have presented a new technique for calculating reaction paths. Like reaction paths from molecular-dynamics simulation, these paths are trajectories – solutions of Newton's equations of motion for the atomic nuclei. Unlike molecular-dynamics trajectories, these paths proceed directly from reactant to product, with a minimum of "rattling around" in the reactant well; in this sense, the paths resemble minimum-energy paths and other, similar, definitions of the reaction coordinate. The reaction paths constructed by our approach combine the desirable features of both these approaches: they possess the conceptual simplicity of reaction-coordinate definitions yet, because they are trajectories, still contain dynamical information.

Our computational method is based on solving the Hamilton-Jacobi equation for the action using a generalized fast-marching method. This is the first algorithm of which we are aware which, starting from the reactant, finds the path that minimizes one observable (e.g., the transit time) while satisfying the Hamilton-Jacobi equation associated with a different observable (e.g., the action). (This contrasts to existing fast-marching algorithms, where the minimization and the Hamilton-Jacobi equation are associated with the same observable.) Not only is the generalized fast-marching method computationally efficient, it has several desirable features. First and foremost, it relies only on the reactant configuration: one does not need to know the product, transition state, or mechanism of the reaction in order to describe the path. In this sense the proposed method is more like molecular dynamics simulation (which can "predict" the product of a chemical reaction) and less like most extant methods for finding chemical reaction

coordinates (which usually require knowledge of reactive intermediates and/or transition states). A second advantage of this method, useful in some contexts, is that there is no time-step; instead, there is a spatial step (determined, in the present paper, by the grid spacing). This has potential advantages in contexts where different molecular motions span a broad range of time scales, as short time steps would correspond to a very fine spatial resolution of the slower molecular processes [27].

After constructing a trajectory with the generalized fast marching method presented here, it is easy to confirm that the trajectory is, in fact, a solution of Newton's equations of motion. In effect, the fast-marching method has determined the "best" value for the initial momentum and the resulting "ballistic trajectory" will start from the reactant conformation and pass to the product state with minimal "rattling around" in the reactant well of the potential energy surface. The sense in which the trajectory is optimal is changed by adjusting the value of n in equation (19). For example, for $n = 0$, the shortest (least-distance) trajectory is found. The most interesting case is probably $n = -1$, which gives the least-time trajectory. Because the reaction occurs faster along this trajectory than any other, more product is formed along the least-time trajectory than along any other path. (We assume that all directions for the initial momentum vector are equally likely.) Such paths are of fundamental interest in chemical kinetics.

Determining the reactive intermediates, products, and paths necessitates exploration of the entire conformational space; the size of the space (and the number of potential intermediates) grows exponentially with increasing dimensionality. There can be no algorithm with subexponential scaling, because the inherent difficulty of the problem grows exponentially with the dimension. One advantage of the present approach is that it provides a "guaranteed best path", as opposed to approaches formulating the dynamics as an initial value problem, which provide a probabilistic guarantee of a full sampling of all the conformational possibilities only in the long-simulation limit (or, alternatively, using an arbitrarily large number of shorter simulations, each with different initial conditions). (That is, one must construct an "ensemble".) Even for small systems, practical calculations never reach these limits.

The present approach is an alternative to the usual initial value formulation of molecular dynamics. The more complete and systematic exploration of the conformational space that is achieved by the fast marching approach, seems desirable. Clearly, any method like this one depends on the ability to describe systems with reduced dimensionality. This is an active area of research, and we note that the dimension reduction technique of Rabitz and coworkers [40] and the essential molecular dynamics approach of Berendsen [38] are key developments. In order to reduce the dimensionality of the system, one must use chemical principles to select a few key coordinates, which are believed to be potential reactive modes. (The identification of reactive modes merely requires that one follow the familiar paths of reasoning associated with proposing

plausible reaction mechanisms. Computational approaches also exist [45].) Such insights are the foundation of much recent work on dynamics using "reduced potential energy surfaces", with the work of Parrinello meriting special mention. The key idea, which is very old, is that a "reduced potential energy surface" or a "potential of mean force" surface can be defined for a few key coordinates by minimizing the energy with respect to the remaining coordinates (for the minimum potential energy surface) or performing a statistical average over the remaining coordinates (for the potential of mean force). Dynamics can then be performed for the "key" chemical coordinates that remain, using the "effective potential energy surface" for the system with reduced dimensionality.

The "hills" method, as developed by Parrinello and coworkers [41–44], is an excellent recent example of this approach. Our approach is an alternative to that approach because it uses the true potential energy surface and a boundary-value formulation of the dynamical problem. Unlike a trajectory, which grows one point at a time, a "front" moves forward many points at a time, so we believe our method is more amenable to parallelization than molecular-dynamics based approaches on reduced potential energy surfaces. Unlike the trajectories located by methods that perturb the potential energy surface, the trajectories in our model are "true", insofar as they are exact Newtonian trajectories on the potential energy surface. (Although, if one is using a "reduced-dimension" surface, the trajectories would not quite be the solutions to the Newton's equations for the full-dimensional system.)

A few additional words about the present method, as opposed to the approaches based on molecular dynamics or statistical sampling, bear mentioning. The great advantage of such "stochastic" methods for characterizing the ensemble of possible reaction paths and reactive intermediates is that they are directly applicable to larger systems. However, such methods give only probabilistic errors, and converge exponentially slowly in the worst-case, reflecting the inherent exponential difficulty of the underlying problem. A deterministic algorithm like this one is directly applicable only to small systems because it scales exponentially even for favorable potential energy surfaces where stochastic methods work satisfactorily. Because this deterministic approach rigorously converges to the correct answer even in the worst case, it can be used to establish the robustness of more efficient non-deterministic approaches. As already mentioned, for larger systems the fast-marching approach is not directly applicable. However, by using reduced-dimensionality potential energy surfaces, it can be applied to large systems. For large systems, however, the computational philosophy of deterministic and stochastic methods is more complementary, so direct comparison of the results is not practical. (However, results from one type of calculation could certainly be used to inform the initialization of alternative techniques.)

In this work, however, we are not focused on large molecules whose dimensionality has been reduced, but the theoretical and mathematical underpinnings of our method. For this reason, we are concentrating on a simple example to

elucidate the key ideas and demonstrate the validity of our approach. It is certainly possible to extend the method to more than two dimensions, and this will be necessary for our targeted applications in enzyme catalysis (where there are often several important collective coordinates). Quite generally, the cost of exploring the conformational space grows exponential with the dimension of the space (k^d-type scaling). If there are $N = k^d$ important conformations, then our method scales as $N \log N$ (which depends on dimensionality like $d(k^d)$). This limit is larger than the theoretical limit by a factor of $\log N$, but we doubt that the theoretical limit can be practically achieved. (It should be noted that even though the present method scales exponentially with increasing dimension, non-fast marching approaches are exponentially worse, with typical scaling $N^2 = k^{2d}$.) We are currently exploring various schemes for dimensional reduction, so that we can consider only the essential degrees of freedom to our system. In addition, though we can never hope to eliminate the exponential scaling of this (or any other) approach, we are attempting to mitigate it using adaptive grids, and more sophisticated upwind differentiation formulas. Finally, we are starting to build interfaces between this program and commercial computational chemistry programs, so that we can begin to apply this approach to realistic chemical systems. We will report on all these additional developments as the results become available.

Acknowledgments

The authors thank NSERC, McMaster University, the Canada Research Chairs, the Canada Foundation for Innovation, and the Ontario Innovation Trust for research support. B.K.D acknowledges support from a Humboldt fellowship and by the DPG Sonderforshungsbereich 450 at the Free University of Berlin, where part of this work was performed.

References

[1] D.C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 2004).
[2] J.M. Haile, *Molecular Dynamics Simulation: Elementary Methods* (Wiley Inter-science, New York, 1997).
[3] J.A. McCammon and S.C. Harvey, *Dynamics of Proteins and Nucleic Acids* (Cambridge University Press, Cambridge, 1987).
[4] D. Heidrich (Ed.), *The Reaction Path in Chemistry: Current Approaches and Perspectives* (Kluwer Academic Publication, Dordrecht 1995).
[5] K. Kukui and H. Fujimoto, *Frontier Orbitals and Reaction Paths* (World Science Publication New York 1997).
[6] K. Fukui, Acc. Chem. Res. 14 (1981) 363.
[7] M. Page and J.W. McIver Jr, J. Chem. Phys. 88 (1988) 922.
[8] W. Quapp, J. Comput. Chem. 25 (2004) 1277.

[9] C.J. Cerjan and W.H. Miller, J. Chem. Phys. 75 (1981) 2800.
[10] R.S. Berry, H.L. Davis and T.L. Beck, Chem. Phys. Lett. 147, (1988) 13.
[11] A. Ulitsky and D. Shalloway, J. Chem. Phys. 106 (1997) 10099.
[12] S. Fischer and M. Karplus, Chem. Phys. Lett. 96 (1992) 5272.
[13] S. Park, M. K. Sener, D. Lu and K. Schulten, J. Chem. Phys. 119 (2003) 1313.
[14] C.W. Gardiner, *Handbook of Stochastic Methods* (Springer, New York, 1983).
[15] S. Huo and J.E. Straub, J. Chem. Phys. 107 (1997) 5000.
[16] S. Huo and J.E. Straub, PROTEINS: Structure, Function, and Genetics 36 (1999) 249.
[17] R. Elber and M. Karplus, Chem. Phys. Lett. 139 (1987) 375.
[18] R. Czerminiski and R. Elber, Int. J. Quant. Chem. 24 (1990) 167
[19] R. Czerminiski and R. Elber, J. Chem. Phys. 92 (1990) 5580.
[20] A. Ulitsky and E. Elber, J. Chem. Phys. 92 (1990) 1510.
[21] R. Elber, in *Recent Developments in Theoretical Studies of Proteins*, ed. R. Elber (World Science Singapore, 1996).
[22] R. Olender and R. Elber, Chem. Phys. 105 (1996) 9299.
[23] L.R. Pratt, J. Chem. Phys. 85 (1986) 5045.
[24] J.E. Straub, in: *Computational Biochemistry and Biophysics*, eds. O.M. Becker, A.D. MacKerell Jr., B. Roux and M. Watanabe (Dekker, New York, 2001).
[25] P. Hanggi, P. Talkner and M. Borkovec, Rev. Mod. Phys. 62 (1990) 251.
[26] L. Onsager and S. Machlup, Phys. Rev. 91 (1953) 1505, 1512.
[27] B.K. Dey, M.R. Janicki and P.W. Ayers, J. Chem. Phys. 121 (2004) 6667.
[28] J.A. Sethian, Proc. Natl. Acad. Sci. USA 93 (1996) 1591.
[29] J.A. Sethian, SIAM Rev. 41 (1999) 199.
[30] J.A. Sethian, *Level Set Methods and Fast Marching Methods* (Cambridge University Press, Cambridge, 1999).
[31] W. Dittrich and M. Reuter, *Classical and Quantum Dynamics: From Classical Paths to Path Integrals* (Springer-Verlag, New York, 1992).
[32] L.M. Landau and E.M. Lifshitz, *Classical Mechanics* (Butterworth-Heinenann, Oxford, 2000).
[33] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, II (Wiley Science New York, 1989).
[34] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes: The Art of Parallel Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, 1996).
[35] R.J. Renka, ACM Trans. Math. Software 14 (1988) 139.
[36] R. Franke and G. Nielson, Int. J. Numer. Methods Eng. 15 (1980) 1691.
[37] B.K. Dey and P.W. Ayers, Mol. Phys. 104 (2006) 541.
[38] A. Amadei, A.B.M. Linssen and H.J.C. Berendsen, Proteins 17 (1993) 412.
[39] C. Schutte, A. Fischer, W. Huisinga and P. Deuflhard, J. Comput. Phys. 151 (1999) 146.
[40] B.K. Dey, H. Rabitz and A. Askar, J. Chem. Phys. 119 (2003) 5379.
[41] A. Laio and M. Parrinello, Proc. Natl. Acad. Sci. USA 99 (2002) 12562.
[42] B. Ensing, A. Laio, M. Parrinello and M. L. Klein, J. Phys. Chem. B 109 (2005) 6676.
[43] A. Stirling, M. Iannuzzi, A. Laio and M. Parrinello, Chemphyschem, 5 (2004) 1558.
[44] B. Ensing, A. Laio, F.L. Gervasio, M. Parrinello and M.L. Klein, J. Am. Chem. Soc. 126 (2004) 9492.
[45] P. Geerlings, F. De Proft and W. Langenaeker, Chem. Rev. 103 (2003) 1793.