Simplified and improved string method for computing the minimum energy paths in barrier-crossing events

Weinan E, Weiqing Ren, and Eric Vanden-Eijnden

Citation: The Journal of Chemical Physics **126**, 164103 (2007); doi: 10.1063/1.2720838
View online: http://dx.doi.org/10.1063/1.2720838
View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/126/16?ver=pdfcov
Published by the AIP Publishing

Simplified and improved string method for computing the minimum energy paths in barrier-crossing events

Weinan E[a)]
Department of Mathematics, Princeton University, Princeton, New Jersey 08544 and PACM, Princeton University, Princeton, New Jersey 08544

Weiqing Ren[b)] and Eric Vanden-Eijnden[c)]
Courant Institute of Mathematical Sciences, New York University, New York, New York 10012

(Received 22 January 2007; accepted 7 March 2007; published online 23 April 2007)

We present a simplified and improved version of the string method, originally proposed by E *et al.* [Phys. Rev. B **66**, 052301 (2002)] for identifying the minimum energy paths in barrier-crossing events. In this new version, the step of projecting the potential force to the direction normal to the string is eliminated and the full potential force is used in the evolution of the string. This not only simplifies the numerical procedure, but also makes the method more stable and accurate. We discuss the algorithmic details of the improved string method, analyze its stability, accuracy and efficiency, and illustrate it via numerical examples. We also show how the string method can be combined with the climbing image technique for the accurate calculation of saddle points and we present another algorithm for the accurate calculation of the unstable directions at the saddle points. © *2007 American Institute of Physics*. [DOI: 10.1063/1.2720838]

I. INTRODUCTION

Rare events have been a topic of great interest in many areas for many years. For systems with simple energy landscapes, the object of interest is the most probable transition path between the local minima of the potential energy. It is known that these paths are the minimum energy paths (MEPs), which are paths in configuration space along which the potential force is everywhere parallel to the path. The MEPs allow us to identify the relevant saddle points which act as the bottlenecks for a particular barrier-crossing event, as well as the unstable directions at these points that enter into the calculation for the prefactor of the transition rates. Several computational methods have been developed for finding the MEPs.[1–8] Most successful among them are the nudged elastic band (NEB) method[4] and the (zero-temperature) string method.[7] Once the MEP is obtained, transition rates can be computed using several strategies (see, for example, Ref. 7).

The present paper is a continuation of the work presented in Ref. 7 on the calculation of MEPs by the string method. Here, we present an improved and simplified version of the string method in which the step of projecting the potential force onto the hyperplane perpendicular to the string is eliminated. Instead the full force is used for the evolution of the string. This not only simplifies the numerical procedure, but also makes the method more stable and accurate, since we no longer need to compute the tangent vectors of the string using up-wind scheme (see Refs. 7 and 8). The overall algorithm is an iterative application of a simple two-step procedure: evolution of the string by standard ordinary differential equation (ODE) solvers, and reparametrization of the string by interpolation. The accuracy of the method depends on the interpolation scheme, whereas its efficiency depends on the ODE solver. High order accuracy and better efficiency can be easily achieved by employing accurate interpolation method and ODE solver. Here we discuss the algorithmic details and the implementation issues of the improved string method, and we analyze its accuracy and stability.

In many cases, we are not interested in the full MEP, but only the saddle points. We show how the climbing image technique developed in Ref. 6 can be naturally and simply combined with the string method in order to identify the location of the saddle points with arbitrary precision. Finally, we show how to compute the unstable direction at the saddle point to arbitrary precision.

II. THE NEW STRING METHOD

Our main objective is to find the minimum energy path for barrier-crossing events. Denote by $V(x)$ the potential energy of the system of interest and assume that $V(x)$ has at least two minima, at a and b. By definition, a MEP is a curve γ connecting a and b that satisfies

$$(\nabla V)^{\perp}(\gamma) = 0, \tag{1}$$

where $(\nabla V)^{\perp}$ is the component of ∇V normal to γ,

$$(\nabla V)^{\perp}(\gamma) = \nabla V(\gamma) - (\nabla V(\gamma), \hat{\tau})\hat{\tau}. \tag{2}$$

Here $\hat{\tau}$ denotes the unit tangent of the curve γ, and $(\cdot,\cdot)$ denotes the Euclidean inner product. In appropriate mathematical setting, one can prove that the MEP is the most probable path that the system will take under the overdamped dynamics to move between a and b, crossing the barriers in-between.[10]

[a)]Electronic mail: weinan@math.princeton.edu
[b)]Electronic mail: weiqing@cims.nyu.edu
[c)]Electronic mail: eve2@cims.nyu.edu

The basic idea of the string method is to find the MEP by evolving a curve connecting a and b, under the potential force field. The simplest dynamics for the evolution of such curves is given abstractly by

$$v_n = -(\nabla V)^{\perp}, \tag{3}$$

where v_n denotes the normal velocity of the curve. We remark that for the evolution of a curve, only the normal component of the velocity matters, tangential velocity only moves points along the curve, changing the parameterization of the curve without changing the curve itself. For the same reason, when we give an explicit representation of the curve, we are free to choose any particular parameterization. The string method makes use of this freedom in an essential way.

To translate (3) to a form that can be readily used in numerical computations, we assume that we have picked a particular parametrization of the curve γ: $\gamma=\{\varphi(\alpha):\alpha \in [0,1]\}$. Then we have $\hat{\tau}(\alpha)=\varphi_\alpha/|\varphi_\alpha|$, where φ_α denotes the derivative of φ with respect to α. The simplest parametrization to think of is equal arc-length parametrization in which α is a constant multiple of the arc length from a to the point $\varphi(\alpha)$. In this case, we also have $|\varphi_\alpha|=$const (this constant being the length of the curve γ).

The original form of the string method uses the following model to represent (3):

$$\dot{\varphi} = -\nabla V(\varphi)^{\perp} + \lambda\hat{\tau}, \tag{4}$$

where $\dot{\varphi}$ denotes the time derivative of φ. The term $\lambda\hat{\tau} \equiv \lambda(\alpha,t)\hat{\tau}(\alpha,t)$ is a Lagrange multiplier term added to enforce the particular parametrization that we have chosen (for instance, by normalized arc length); as explained before, this term does not affect the evolution of the curve itself, only its parametrization, since it does not contribute to the normal velocity of the curve. In the actual numerical algorithm, the action of $\lambda\hat{\tau}$ is realized by a simple interpolation step, as we discuss below.

It is easy to see that the stationary states of this dynamics (when the time derivative vanishes) satisfies (2).

The main difficulty with this model is in the computation of the projected force. Numerical stability requires changing the way that the tangent vector is computed before and after the saddle points are crossed.[7–9] This step lowers the accuracy of the overall method.[14] This projection step is eliminated in the new method, which amounts to solving

$$\dot{\varphi} = -\nabla V(\varphi) + \bar{\lambda}\hat{\tau}, \tag{5}$$

where $\bar{\lambda}(\alpha,t)\hat{\tau}(\alpha,t)$ is again a Lagrange multiplier term for the purpose of enforcing the particular parametrization of the string. Clearly, (5) is equivalent to (4) with the identification $\bar{\lambda}=\lambda+(\nabla V,\hat{\tau})$, but the representation (5) is better for numerical purposes

As in the original string method, in the improved method the string is discretized into a number of images $\{\varphi_i(t), i=0,1,\ldots,N\}$. The images along the string are evolved by iterating upon the following two-step procedure based on time splitting of the terms at the right hand side of (5).

In the first step, the discrete points on the string are evolved over some time interval Δt according to the full potential force,

$$\dot{\varphi}_i = -\nabla V(\varphi_i). \tag{6}$$

Equation (6) can be integrated in time by any ODE solver, e.g., the forward Euler method or Runge-Kutta methods, as described in Sec. IV A.

In the second step, the points are redistributed along the string using a simple interpolation/reparametrization procedure as described in Sec. IV B.

The scheme above is not only simpler than the original string method (and NEB), it is also more stable and more accurate, as discussed in Sec. IV C.

III. COMPARISON WITH THE NUDGED ELASTIC BAND METHOD

Another successful algorithm for computing the MEP is the NEB method.[4] As the string method, NEB is a chain-of-states method, since the MEP is obtained as a chain of states connected by some spring force. The states on the chain are viewed as different replicas of the original system.

Even though after discretization, the string or elastic band are indeed represented by a chain of states, conceptually it is much more convenient to adopt a continuous viewpoint as was done and expressed in (4) and (5). From this viewpoint, NEB can be thought of as an evolution upon the elastic band method,

$$\dot{\varphi} = -\nabla V(\varphi) + \kappa\varphi_{\alpha\alpha}, \tag{7}$$

where κ is the spring constant. It is well known that the elastic band method converges to a path that, in general, is not a MEP.[11] For this reason, the nudged elastic band method is proposed[4] which takes the form

$$\dot{\varphi} = -\nabla V(\varphi)^{\perp} + \kappa(\varphi_{\alpha\alpha},\hat{\tau})\hat{\tau}. \tag{8}$$

Compared with (4), we see that the difference between the original version of the string method and the NEB is in the second term at the right hand side, which is there to prevent the images along the path from falling into the local minima at the ends, due to the potential force. In the elastic band method, this is done by adding some artificial spring force along the path. In the string method, this is done by enforcing a particular parametrization. From an algorithmic viewpoint, the first approach amounts to penalizing the breaking away images. The second approach uses interpolation to fill up the space created along the string due the effect of the potential force.

The new string method takes away the projection of the potential force. Clearly this idea can also be used for the elastic band method, i.e., we may replace (8) by

$$\dot{\varphi} = -\nabla V(\varphi) + \kappa(\varphi_{\alpha\alpha},\hat{\tau})\hat{\tau}. \tag{9}$$

This "half-nudged" elastic band method is different from both (7) and (8) but, like (8), it has the exact MEPs as steady states. Compared to (8),(9) may also have some of the advantages discussed below of the new string method over the original string method.

Finally, let us note that the string method can also be viewed as the *inextensible* limit ($\kappa \to \infty$) of the elastic band method (nudged or half nudged). This explains the terminology of "string."

IV. ALGORITHMIC DETAILS AND PERFORMANCE OF THE NEW STRING METHOD

A. Step 1: Evolution of the images

Equation (6) can be integrated in time by any suitable ODE solver. If we denote by φ_i^n, $i=0,\ldots,N$, the positions of the images after n iterations of the scheme, the new set of images after step 1 is given by

$$\varphi_i^\star = \varphi_i^n - \Delta t \nabla V(\varphi_i^n) \tag{10}$$

if the forward Euler method is used, or

$$\begin{aligned} k_i^{(1)} &= \Delta t \nabla V(\varphi_i^n), \\ k_i^{(2)} &= \Delta t \nabla V\left(\varphi_i^n + \tfrac{1}{2}k_i^{(1)}\right), \\ k_i^{(3)} &= \Delta t \nabla V\left(\varphi_i^n + \tfrac{1}{2}k_i^{(2)}\right), \\ k_i^{(4)} &= \Delta t \nabla V(\varphi_i^n + k_i^{(3)}), \\ \varphi_i^\star &= \varphi_i^n - \tfrac{1}{6}k_i^{(1)} - \tfrac{1}{3}k_i^{(2)} - \tfrac{1}{3}k_i^{(3)} - \tfrac{1}{6}k_i^{(4)} \end{aligned} \tag{11}$$

if the fourth order Runge-Kutta method is used.

B. Step 2: Interpolation/reparametrization of the string

Parametrization by equal arc length. In the simplest case when we choose to enforce the equal arc-length parametrization, the problem is simply the following: Given the values $\{\varphi_i^*\}$ on a nonuniform mesh $\{\alpha_i^*\}$, we would like to interpolate these values onto a uniform mesh with the same number of points. This is done in the following two simple steps, with a cost of $O(N)$:

(1) We calculate the arc length corresponding to the current images,

$$s_0 = 0, \quad s_i = s_{i-1} + |\varphi_i^\star - \varphi_{i-1}^\star|, \quad i = 1,2,\ldots,N. \tag{12}$$

The mesh $\{\alpha_i^\star\}$ is then obtained by normalizing $\{s_i\}$,

$$\alpha_i^\star = s_i / s_N.$$

(2) Next we use interpolation to obtain the new points φ_i^{n+1} at the uniform grid points $\alpha_i = i/N$. This can be done, for example, by using cubic spline interpolation for the data $\{(\alpha_i^\star, \varphi_i^\star), i=0,\ldots,N\}$ (see the Appendix).

Parameterization by energy-weighted arc length. Compared to equal arc-length parameterization, the energy-weighted arc length parameterization gives finer resolution around the saddle points, and thus better estimate of the energy barrier and also the unstable direction at those points. In the reparametrization step, we first calculate the energy-weighted arc length corresponding to the current images,

$$s_0^w = 0, \quad s_i^w = s_{i-1}^w + W_{i-(1/2)}|\varphi_i^\star - \varphi_{i-1}^\star|, \quad i = 1,2,\ldots,N. \tag{13}$$

Here $W_{i-(1/2)} = W(V_{i+1/2})$ and $V_{i+1/2}$ is the average of the potential energy at $\varphi_{i-1}^\star$ and $\varphi_i^\star$. The weight function $W(z)$ is some positive, increasing function of $z \in \mathbb{R}$. The mesh $\{\alpha_i^\star\}$ is obtained by normalizing $\{s_i^w\}$: $\alpha_i^\star = s_i^w / s_N^w$. The new points φ_i on $\alpha_i = i/N$ are then calculated by cubic spline interpolation across the points $\{\varphi_i^\star, i=0,\ldots,N\}$ (see the Appendix).

Once the new points $\{\varphi_i^{n+1}, i=0,\ldots,N\}$ are calculated, we go back to step 1 and iterate until convergence.

Let us note that the above procedure of interpolation/reparametrization is only second order in terms of preserving the parameterization of the curve by arc length or energy-weighted arc length since we use linear interpolation in (12) and (13) to compute the length of the curve. However, the procedure is fourth order in terms of the accuracy of the curve since we use cubic spline for the interpolation. This is fine since we care mostly about finding the MEP accurately.

C. Accuracy, stability, and efficiency

The main advantages of the new string method over the original one are the following:

(1) Simplicity.
(2) Better stability, which means larger time steps can be used to evolve the string. Even though the dynamics of the string is artificial, this means that we can march to the stationary states, the MEPs, faster using larger time steps. The new string method is stable provided only the time step Δt used in (10) or (11) is within the stability region of the ODE solver, which is determined by the stiffness of the potential energy V but is independent of N. In contrast, in the original string method or the NEB method, stability imposes an additional constraint on the size of the time step (the Courant-Friedricks-Lewy condition[12]), which gives $\Delta t < C/N$ in the original string method, and $\Delta t \leqslant C/N^2$ in NEB, where C is some constant.
(3) Better accuracy as a function of the number of points along the string. The new method does not involve computation of the tangent vector explicitly. This avoids the up-winding techniques used in the original string method,[9] which was the bottleneck for improving its accuracy. If cubic splines are used for the interpolation, and the time step is chosen appropriately, then the overall accuracy is fourth order, by which we mean that the distance between the converged string (i.e., the curve interpolated through the images $\{\varphi_i^n, i=0,\ldots,N\}$ after convergence) and the MEP scales as $O(N^{-4})$.

To understand the accuracy of the method, note first that the new string method converges to a state such that the images φ_i^n lead to a new set of images $\varphi_i^\star$ via time evolution (step 1) which then lead back to the same set of images $\varphi_i^n \equiv \varphi_i^{n+1}$ after interpolation and reparametrization (step 2). If there were no numerical errors, both the images $\varphi_i^n \equiv \varphi_i^{n+1}$ and $\varphi_i^\star$ would belong exactly to the MEP since the solution of (6) stays on the MEP if its starts on the MEP. In practice,

however, these images remain slightly off the MEP because (i) their number is finite and hence the interpolation in step 2 is only approximate and (ii) the motion along the MEP in step 1 is only approximate. The corresponding numerical errors can be estimated as follows.

With a mth order accurate interpolation scheme ($m=4$ for cubic spline), the interpolation error scales as

$$\text{interpolation error} = O(N^{-m}). \tag{14}$$

On top of this there is the local error of the ODE solver. If the local order of accuracy of the solver is l (i.e., $l=2$ for forward Euler, $l=5$ for fourth order Runge-Kutta), the error introduced scales as

$$\text{evolution error} = O(\Delta t^{-l}). \tag{15}$$

Thus, in order that the total error between the curve interpolated through the images $\{\varphi_i^n, i=0,\ldots,N\}$ and the actual MEP be less than some prescribed accuracy, say TOL, we must choose N and Δt as

$$N = C_1 \mathrm{TOL}^{-1/m}, \quad \Delta t = \min\{C_2 \mathrm{TOL}^{-1/l}, \Delta t'\}, \tag{16}$$

where C_1 and C_2 are two numerical constants and $\Delta t'$ is the largest time step at which the ODE solver remains stable; as explained before, $\Delta t'$ is determined by the stiffness of the potential energy V, but is independent of N.

D. Illustrative example

To illustrate the improved string method, we consider the two-dimensional potential given by

$$V(x,y) = (1-x^2-y^2)^2 + y^2/(x^2+y^2). \tag{17}$$

The contour lines of V are shown in the upper panel of Fig. 1. The potential V has two minima at $A=(-1,0)$ and $B=(1,0)$, respectively. The exact MEPs connecting these two states are the upper and lower branches of the unit circle: $x^2+y^2=1$.

In our calculation, the initial string is the linear interpolation between $(-0.5,0.5)$ and $(0.5,0.5)$ (the dashed line in the upper panel of Fig. 1). Note that the locations of the minima are not required *a priori*. As long as the two end points of the initial string lie in the two basins of attraction of the minima, they are identified automatically since the end points of the string evolve freely according to (6) and their positions are not affected by the reparameterization step. The string is discretized into N points $\{\varphi_i=(x_i,y_i), i=0,1,\cdots,N-1\}$; in the numerical experiments we used N ranging from 4 to 512. Cubic spline interpolation is used to redistribute the discrete points at each time step according to equal arc length. The fourth order Runge-Kutta method is used to evolve the N discrete points until the maximum displacement d of the string is less than some tolerance TOL, $d<\mathrm{TOL}$, where

$$d = \max_i \frac{1}{\Delta t}|\varphi_i^{n+1} - \varphi_i^n|; \tag{18}$$

and

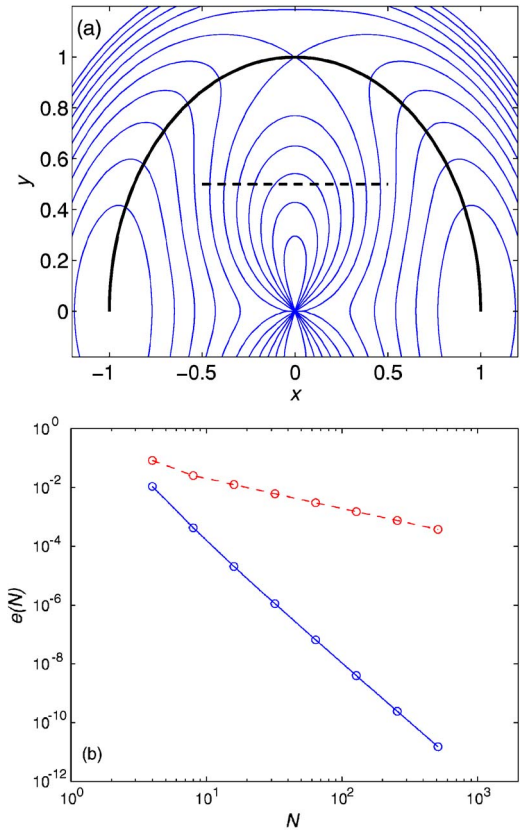

FIG. 1. Upper panel: Initial string (dashed curve)and calculated MEP (thicker solid curve). The background shows the contour lines of the two dimensional potential in (17). Lower panel: Error $e(N)$ of the calculated MEP vs N, the number of points along the string. The solid curve is the result of the improved string method using fourth order Runge-Kutta method and cubic spline interpolation. The dashed curve is the result of the original string method.

$$\mathrm{TOL} = \max\{N^{-4}, 10^{-10}\}. \tag{19}$$

Note that the choice of TOL is consistent with the fourth order accuracy of the numerical scheme. The time step Δt is

$$\Delta t = 0.05 \min\{0.2, N^{-1}\}. \tag{20}$$

For large N this gives $\Delta t = O(N^{-1})$ which is slightly smaller than the optimal choice $\Delta t = O(N^{-4/5})$ [see (16)].

The thicker solid curve in the upper panel of Fig. 1 shows the calculated MEP. The lower panel illustrates its accuracy for various number of points N. The solid line (circles) shows the error of the converged string versus N. Here the error is defined as

$$e(N) = \max_i |(x_i^2+y_i^2)^{1/2} - 1|, \tag{21}$$

where we recall $\varphi_i=(x_i,y_i)$ and φ_i, $i=0,\ldots,N-1$ are the N images along the converged string. The result shows that $e(N)$ scales as N^{-4}, which is consistent with the fourth order accuracy of the cubic spline interpolation and the fact that the time step in the Runge-Kutta method has be chosen accordingly.

Also plotted in the lower panel of Fig. 1 (dashed line) is the numerical result of the original string method as described in Sec. II. The error $e(N)$ scales as N^{-1}. Here the accuracy is limited by the up-winding scheme (28) for the

calculation of tangent vectors, which is only first order accurate. The error in NEB scales as N^{-1} as well.

V. CALCULATION OF SADDLE POINTS AND UNSTABLE DIRECTIONS

The string method is primarily designed to identify accurately the whole MEP. In many cases, we are not interested in the whole MEP, we are only interested in the saddle point and some local quantities at the saddle point for the computation of the transition rates. We will show in this section that for this purpose, the climbing image technique developed in Ref. 6 can be naturally and easily adapted to the string method. This will allow us to identify the saddle point φ_s to arbitrary precision at a cost of moving a single image. We will then describe a technique in Sec. V B for identifying afterwards the unstable direction $\hat{\tau}_s$ at a cost of moving two images.

A. Calculation of the saddle point

If we are interested in determining the location of the saddle point (but not of the whole MEP) with high precision, instead of increasing the number of points N along the string, it is better to first find a rough approximation of the MEP using the string method with a small number of images and then switch to the following climbing image algorithm in which one integrates (using, e.g., forward Euler or the fourth order Runge-Kutta algorithms)

$$\dot{\varphi}_s = -\nabla V(\varphi_s) + 2(\nabla V(\varphi_s), \hat{\tau}_s^0)\hat{\tau}_s^0, \tag{22}$$

with the initial condition $\varphi_s(0)=\varphi_s^0$. The equilibrium points of (22) satisfy $0=\nabla V$, i.e., they are critical points (minima, saddle points, etc.) of the potential V. Provided only that $\hat{\tau}_s^0$ be close enough to the actual $\hat{\tau}_s$, the saddle point, which is an unstable equilibrium point of (6), is a stable equilibrium point of (22). This is because the force is inverted in the direction of $\hat{\tau}_s^0$: hence this direction, which is unstable for (6), becomes stable for (22). As a result the solution of (22) converges towards the actual saddle point φ_s provided that φ_s^0 is close enough to it. Notice that (22) involves evolving one image only ($\hat{\tau}_s$ is fixed in this equation). Therefore, since the convergence of the solution (22) towards the saddle point is exponential in time, the number of steps n_{step} required to achieved a given accuracy TOL scales as

$$n_{\text{step}} = O(\log \text{TOL}^{-1}). \tag{23}$$

If there is more than one saddle point along the MEP, the procedure above can be straightforwardly generalized to identify them all by using (22) for each of them.

The above is a two-step procedure. One first runs the string method and then the climbing image technique, holding other images fixed. If desired, we can also integrate the climbing image technique into the string method. This requires a modification of the reparametrization step. The climbing image φ_s would evolve according to (22) [whereas the other images evolves according to (6)]. In the reparametrization step, the climbing image is kept fixed, and reparametrization is done separately on both sides of the climbing images. This can be easily achieved by using the following mesh instead of the uniform one at reparametrization:

$$\alpha_0 = 0, \quad \alpha_i = \alpha_{i-1} + h_i, \quad i = 1,2,\ldots,N, \tag{24}$$

where

$$h_i = \begin{cases} \alpha_{i_s}^\star / i_s, & \text{if } i \leqslant i_s \\ (1-\alpha_{i_s}^\star)/(N-i_s), & \text{if } i > i_s. \end{cases} \tag{25}$$

Here i_s is the index of the climbing image, $\alpha_{i_s}^\star$ is the normalized arc length corresponding to the climbing image, i.e., $\varphi_s = \varphi(\alpha_{i_s}^\star)$. The choice of α_i's in (24) and (25) amounts to splitting the string into the two pieces on each side the climbing image and treating the two pieces independently. The discrete points on both pieces are redistributed evenly using cubic spline interpolations.

B. Calculation of the unstable direction

Once the above calculation converged and the saddle point φ_s is accurately located, we can switch to another method to compute accurately the tangent vector $\hat{\tau}_s$ at φ_s. By definition, $\hat{\tau}_s$ is the unique eigenvector with negative eigenvalue of the Hessian tensor at the saddle point, $H(\varphi_s) = \nabla\nabla V(\varphi_s)$, i.e., the unique solution of

$$H(\varphi_s)\hat{\tau}_s = \mu\hat{\tau}_s, \tag{26}$$

with $\mu<0$ and $|\hat{\tau}_s|=1$. For future reference, notice that the solution of (26) with $\mu<0$ and $|\hat{\tau}_s|=1$ is also the steady state solution of

$$\dot{\hat{\tau}}_s = -H(\varphi_s)\hat{\tau}_s + \mu\hat{\tau}_s, \tag{27}$$

where μ can be viewed as a Lagrange multiplier to enforce the constraint that $|\hat{\tau}_s|=1$.

(27) is inconvenient because it involves the Hessian tensor $H(\varphi_s)$. The idea of the method presented in this section is to avoid computing the Hessian by approximating $\hat{\tau}_s$ and $H(\varphi_s)$ in (27) using finite differences. This can be done by introducing two images, φ_r and φ_l, such that

$$\hat{\tau}_s \approx \hat{\tau}_s^a = \frac{\varphi_r - \varphi_l}{|\varphi_r - \varphi_l|}, \quad |\varphi_r - \varphi_s| = |\varphi_l - \varphi_s| = h \tag{28}$$

for some small $h>0$. Initially, φ_r and φ_l can be taken as two points on each side of the saddle point φ_s determined in Sec. V A and lying in the direction of the approximation $\hat{\tau}_s^0$ provided by the string method. The images φ_r and φ_l are then updated by using the following finite-difference approximation of (27):

$$\begin{aligned} \dot{\varphi}_l &= -\nabla V(\varphi_l) + \lambda_l(\varphi_l - \varphi_s), \\ \dot{\varphi}_r &= -\nabla V(\varphi_r) + \lambda_r(\varphi_r - \varphi_s), \end{aligned} \tag{29}$$

where λ_l and λ_r are Lagrange multipliers determined by the constraints

$$|\varphi_l - \varphi_s| = |\varphi_r - \varphi_s| = h. \tag{30}$$

(29) is a discretized version of (27) because

$$\nabla V(\varphi_l) = H(\varphi_s)(\varphi_l - \varphi_s) + O(h^2) \tag{31}$$

and similarly for $\nabla V(\varphi_r)$: here we used $\nabla V(\varphi_s)$ and $|\varphi_l - \varphi_s| = h$.

In practice, (29) can be solved by a two-step procedure. At each time step, φ_r and φ_l is first evolved by the potential force to give intermediate values,

$$\varphi_l^\star = \varphi_l^n - \Delta t \nabla V(\varphi_l^n), \tag{32}$$

and similarly for $\varphi_r^\star$; then the constraints in (30) are enforced by projecting $\varphi_l^\star$ and $\varphi_r^\star$ to the sphere $S_{\varphi_s,h}$ with center φ_s and radius h,

$$\varphi_l^{n+1} = \varphi_s + h\frac{\varphi_l^\star - \varphi_s}{|\varphi_l^\star - \varphi_s|} \tag{33}$$

and similarly for $\varphi_r^\star$. The steady-state solution of the procedure above is used in (28) to calculate the tangent vector $\hat{\tau}_s$.

The parameter h in (28) should be chosen as small as possible without impeding the accuracy with round-off errors: if the digital precision is $\mathrm{TOL}_{\min}$, one should choose $h = \mathrm{TOL}_{\min}^{1/2}$, in which case the error due to finite difference in (28) remains $O(h^2) = O(\mathrm{TOL}_{\min})$.

Notice that the time step Δt in (32) can be chosen independently of h without impeding on the accuracy because (31) implies that $\nabla V(\varphi_l) = O(h)$ and $\nabla V(\varphi_r) = O(h)$. As a result $|\varphi_l^\star - \varphi_l^n| = O(h)$ and $|\varphi_r^\star - \varphi_l^n| = O(h)$ and the two steps in the procedure above do not interfere with the accuracy regardless of what Δt is. Since the convergence of the solution of (29) is exponential in time, the number of steps n_{step} required to achieved a given accuracy TOL on τ_s scales as in (23).

Note that the above procedure brings φ_r and φ_l to the minima of the potential energy V on the sphere $S_{\varphi_s,h}$ by steepest descent dynamics. More efficient constrained optimization methods can be used as well to improve the convergence rate and save computational cost.[15]

C. Illustrative example

In this example, we calculate the MEP, one of the saddle point, and the associated unstable direction for the Mueller potential.[13]

In the calculation, we first identify an approximation of the MEP using the improved string method with $N = 10$ images. Cubic splines were used in the reparametrization and the forward Euler method with $\Delta t = 4.5 \times 10^{-4}$ was used in the integration. After 70 time steps when d defined in (18) is less than 0.1, we stop the string calculation, and identify the image of maximum energy along the string, φ_s^0, and the corresponding $\hat{\tau}_s^0$. Then we switch to the climbing image algorithm described in Sec. V A to improve φ_s^0, using again $\Delta t = 4.5 \times 10^{-4}$ in (22). The numerical result is shown in the upper panel of Fig. 2. The figure shows the initial string (dashed line) and the calculated MEP (filled circles). The background shows the contour lines of the Mueller potential. There is an intermediate metastable state along the MEP, and accordingly there are two saddle points. The empty circle on the MEP indicates the location of the saddle point φ_s with higher energy, obtained by the climbing image technique. After convergence, the norm of the potential force at φ_s, $|\nabla V(\varphi_s)|$, is smaller than 10^{-12}. It takes 188 time steps to reach this accuracy. The convergence history for the calculation of the saddle point is shown in the lower panel of Fig. 2. The error decays exponentially with the iteration number or time step n.

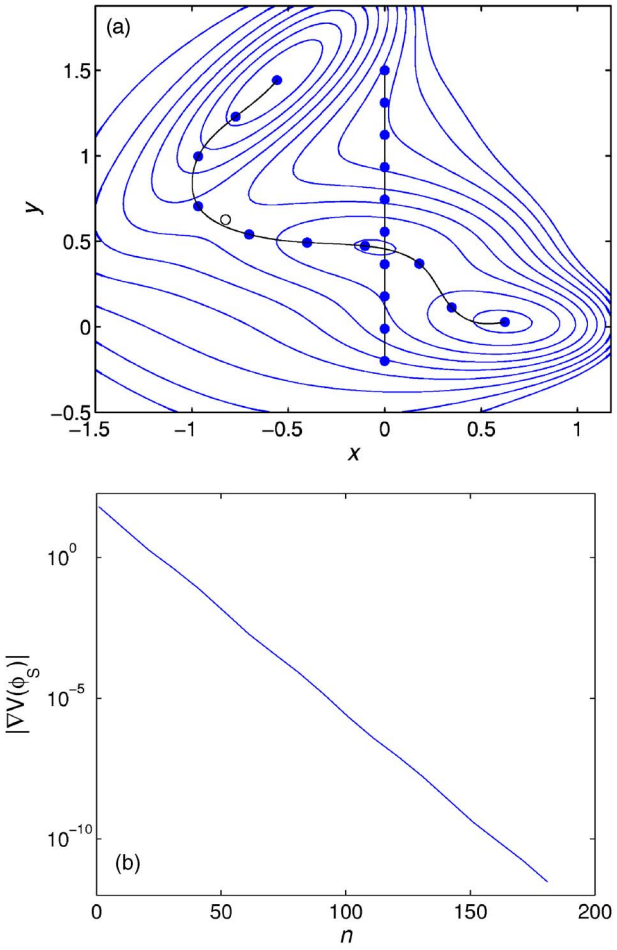

FIG. 2. Upper panel: Initial string and calculated MEP using the string method with ten images (the images are shown as filled circles; the lines are the curves interpolated across these images; the vertical line is the initial string and the other one is the calculated MEP). The empty circle indicates the saddle point identified by combining the string method with the climbing image technique. The norm of the residual potential force at φ_s is smaller than 10^{-12}, $|\nabla V(\varphi_s)| < 10^{-12}$. The background shows the contour lines of the Mueller potential. Lower panel: The norm of the force on the climbing image $|\nabla V(\varphi_s)|$ vs the number n of iterations or time steps. The convergence is exponential in time.

We then proceeded to calculate the unstable direction at φ_s using the algorithm described in Sec. V B. We compared the accuracy of the numerical results for different choices of h.[2,3,5,15] The numerical result is shown in the upper panel of Fig. 3. Here the error is calculated by

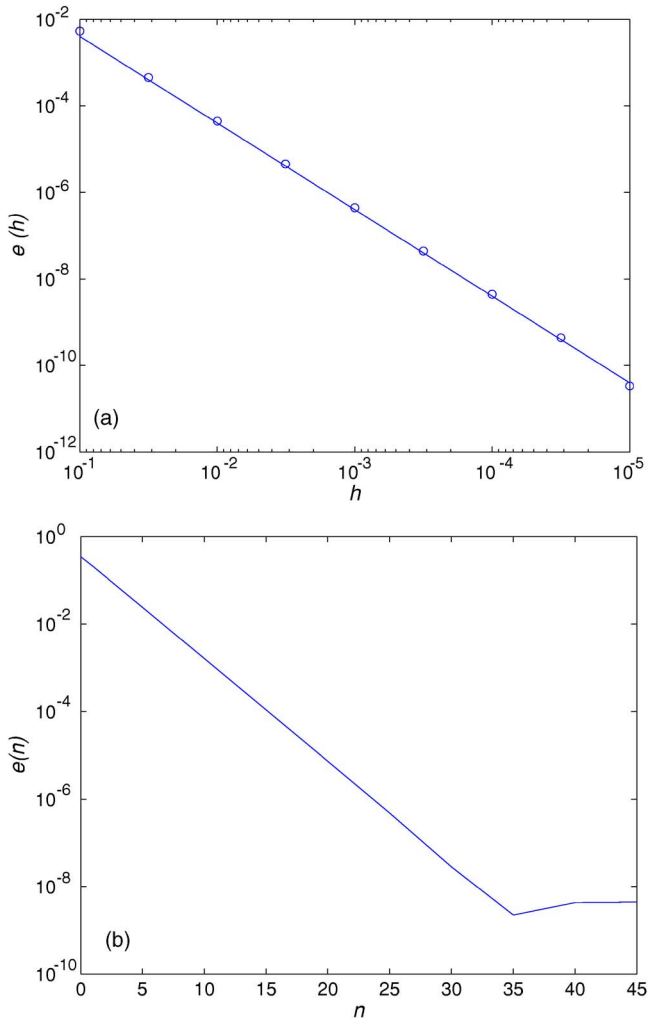

FIG. 3. Upper panel: The error $e(h)$ in (34) of the computed unstable direction at the saddle point vs h after convergence in the Mueller potential example. The error decays as h^2. Lower panel: The error $e(n)$ in (35) of the computed unstable direction vs the number of n of iterations or time steps when $h=10^{-4}$. The convergence is exponential in time until it reaches its optimal value $O(h^2)$.

$$e(h) = |\hat{\tau}_s^a(h) - \hat{\tau}_s^{\text{exact}}|, \tag{34}$$

where $\hat{\tau}_s^a(h)$ is the numerical solution calculated from (28) after convergence of the two-step procedure and $\hat{\tau}_s^{\text{exact}}$ is the exact value computed directly from the Hessian of the potential at φ_s. The discrete circles in the figure are the errors at different h. The solid curve is a plot of a function which is proportional to h^2. Clearly the error of the numerical result decays as h^2.

In the lower panel of Fig. 3 we show the error

$$e(n) = |\hat{\tau}_s^a(n) - \hat{\tau}_s^{\text{exact}}| \tag{35}$$

as a function of the iteration number n (which is also the number of force field evaluations); here $\hat{\tau}_s^a(n)$ is the numerical solution calculated from (28) after n iterations and we took $h=10^{-4}$. The error decreases exponentially with time until it reaches a plateau when $e(n)=O(h^2)$.

VI. CONCLUDING REMARKS

In this paper the string method originally proposed in Ref. 7 was improved. The main components of the improved string method are an ODE solver and an interpolation scheme, both of which are standard numerical techniques. Compared to the original string method and other MEP-finding methods such as NEB, the new method is simpler and easier to implement. More importantly, the numerical scheme does not need the projection of the potential force and thus eliminates the stablility issue. High order accuracy can be easily achieved by employing higher order ODE solvers and more accurate interpolation schemes.

We also described how the climbing image technique can be easily incorporated into the string method for accurate calculation of saddle points. Furthermore, we presented an algorithm for accurate calculation of the unstable directions at the saddle points.

These various methods were illustrated here via simple low-dimensional examples but there is no obstacle in applying them to high-dimensional systems of practical interest.

ACKNOWLEDGMENTS

The authors thank Giovanni Ciccotti, Bob Kohn, and Mitch Luskin for useful discussions and comments. One of the authors (W.E.) was partially supported by ONR Grant No. N00014-01-1-0674 and DOE Grant No. DE-FG02-03ER25587. One of the authors (W.R.) was supported by NSF Grant No. DMS-0604382. One of the authors (E.V.-E.) was partially supported by NSF Grant Nos. DMS02-09959 and DMS02-39625, and by ONR Grant No. N00014-04-1-0565.

APPENDIX: CUBIC SPLINE INTERPOLATION

Consider the one-dimensional cubic spline for N points $\{(x_i,y_i), i=0,1,2,\cdots,N\}$. In the string method, the parameters α_i corresponds to x_i, and each component of $\varphi_i^\star$ corresponds to y_i. A cubic spline is piecewise made of third-order polynomials which pass through these N points. The ith piece of the spline on $[x_i,x_{i+1}]$ is represented by the following function $s(x)$ which ensures the continuity of $s(x)$ and also its second derivative $s''(x)$ at the interior points $\{x_i, i=1,2,\cdots,N-1\}$:[15]

$$s(x) = wy_{i+1} + \bar{w}y_i + h_i^2((w^3-w)\sigma_{i+1} + (\bar{w}^3-\bar{w})\sigma_i), \tag{A1}$$

where $h_i=x_{i+1}-x_i$, $w=(x-x_i)/h_i$, and $\bar{w}=1-w$; the parameters σ_i's in (A1) are determined by requiring the continuity of the first derivative of $s(x)$ at the interior points, i.e.,

$$s'_-(x_i) = s'_+(x_{i+1}), \quad i=2,3,\ldots,N-1, \tag{A2}$$

plus two additional conditions at the two end points. In our calculation, for these additional conditions we chose to set the third derivative $s'''(x_1)$ equal to the third derivative of the cubic polynomial which interpolates $\{x_i, i=0,1,2,3\}$, and similarly for $s'''(x_N)$.

The equations in (A2) and the two conditions at the end points form N linear equations for the N unknowns σ_i. This linear system is symmetric and tridiagonal and can be solved easily by forward and backward substitutions.[15]

[1] A. Ulitsky and R. Elber, J. Chem. Phys. **96**, 1510 (1990).
[2] S. Fischer and M. Karplus, Chem. Phys. Lett. **194**, 252 (1992).
[3] R. Olender and R. Elber, J. Chem. Phys. **105**, 9299 (1996).
[4] H. Jónsson, G. Mills, and K. W. Jacobsen, *Classical and Quantum Dynamics in Condensed Phase Simulations*, edited by B. J. Berne, G. Ciccoti, and D. F. Coker (World Scientific, Singapore, 1998).
[5] G. Henkelman and H. Jónsson, J. Chem. Phys. **111**, 7010 (1999).
[6] G. Henkelman, B. P. Uberuaga, and H. Jónsson, J. Chem. Phys. **113**, 9901 (2000).
[7] W. E, W. Ren, and E. Vanden-Eijnden, Phys. Rev. B **66**, 052301 (2002).
[8] G. Henkelman and H. Jónsson, J. Chem. Phys. **113**, 9978 (2000).
[9] To be more precise, if the string is discretized into a number of points (or images) $\{\varphi_i(t), i=0,1,\cdots,N\}$, an up-winding scheme must be used to calculate the discretized approximation to the tangent vector along the string in order that the scheme for integrating (4) be stable. This amounts, e.g., to using the following:

$$\hat{\tau}_i = \begin{cases} \dfrac{\varphi_{i+1}-\varphi_i}{|\varphi_{i+1}-\varphi_i|}, & \text{if } V(\varphi_{i+1}) > V(\varphi_i) > V(\varphi_{i-1}) \\ \dfrac{\varphi_i-\varphi_{i-1}}{|\varphi_i-\varphi_{i-1}|}, & \text{if } V(\varphi_{i+1}) < V(\varphi_i) < V(\varphi_{i-1}) \\ \dfrac{\varphi_{i+1}-\varphi_{i-1}}{|\varphi_{i+1}-\varphi_{i-1}|}, & \text{if } V(\varphi_{i+1}) < V(\varphi_i) > V(\varphi_{i-1}) \\ & \text{or } V(\varphi_{i+1}) > V(\varphi_i) < V(\varphi_{i-1}). \end{cases}$$

[10] M. I. Freidlin and A. D. Wentzell, *Random Perturbations of Dynamical Systems*, 2nd ed. (Springer, New York, 1998).
[11] R. E. Gililan and K. R. Wilson, J. Chem. Phys. **97**, 1757 (1992).
[12] K. W. Morton and D. F. Mayers, *Numerical Solution of Partial Differential Equations*, 2nd ed. (Cambridge University, Cambridge, 2005).
[13] K. Mueller, Angew. Chem. **19**, 1 (1980).
[14] W. Ren, Commun. Math. Sci. **1**, 377 (2003).
[15] G. E. Forsythe, M. A. Malcolm, and C. B. Moler, *Computer Methods for Mathematical Computations* (Prentics-Hall, Englewood Cliffs, NJ, 1977).