Journal of Computational Physics 309 (2016) 69–87

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Iterative minimization algorithm for efficient calculations of transition states

Weiguo Gao [a,b], Jing Leng [a], Xiang Zhou [c,*]

[a] School of Mathematical Sciences, Fudan University, Shanghai, 200433, China
[b] MOE Key Laboratory of Computational Physical Sciences, Fudan University, Shanghai, 200433, China
[c] Department of Mathematics, City University of Hong Kong, Tat Chee Ave, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 13 January 2015
Received in revised form 25 September 2015
Accepted 29 December 2015
Available online 4 January 2016

Keywords:
Saddle point
Transition states
Energy landscape
Eigenvector-following
Iterative minimization

ABSTRACT

This paper presents an efficient algorithmic implementation of the iterative minimization formulation (IMF) for fast local search of transition state on potential energy surface. The IMF is a second order iterative scheme providing a general and rigorous description for the eigenvector-following (min-mode following) methodology. We offer a unified interpretation in numerics via the IMF for existing eigenvector-following methods, such as the gentlest ascent dynamics, the dimer method and many other variants. We then propose our new algorithm based on the IMF. The main feature of our algorithm is that the translation step is replaced by solving an optimization subproblem associated with an auxiliary objective function which is constructed from the min-mode information. We show that using an efficient scheme for the inexact solver and enforcing an adaptive stopping criterion for this subproblem, the overall computational cost will be effectively reduced and a super-linear rate between the accuracy and the computational cost can be achieved. A series of numerical tests demonstrate the significant improvement in the computational efficiency for the new algorithm.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Many phenomena in physics, material sciences, chemistry and biology can be abstractly formulated as a system that navigates over a complex potential energy landscape of high or infinite dimensions. Rare events, which correspond to infrequently hops between different local minima of the potential function, have attracted extensive research work to understand quite many important physical processes in natural sciences and engineering. The examples include chemical reactions, phase transitions of condensed matter, etc. To study rare event, one of the fundamental questions is to find energy barriers and transition states. The barrier is the energy difference between a local minimum and its transition state. The transition state is a critical point on the potential energy surface with exactly one negative Hessian eigenvalue. We call this type of saddle points as index-1 saddle points. Furthermore, finding saddle points provides valuable information for many bifurcation problems, especially for subcritical bifurcations, since the unstable manifold of index-1 saddle point connects two locally stable solutions.

While locating potential energy minima can routinely be done, at least for local search, thanks to the significant progress of nonlinear optimization, finding saddle points can be extremely difficult and remains one of the major challenges for large

* Corresponding author.
E-mail address: xiang.zhou@cityu.edu.hk (X. Zhou).

http://dx.doi.org/10.1016/j.jcp.2015.12.056
0021-9991/© 2015 Elsevier Inc. All rights reserved.

systems. Several methods exist to locate the transition states. They can be divided into surface-walking algorithms, which use only local quantities around one point on the potential energy surface, and the chain-of-states methods that connect different points on the surface. The nudged elastic band method [1] is an example of the chain-of-state methods and the string method [2–4] presents an elegant dynamical description and a convenient numerical technique in the path space for such chain-of-states methods. For the surface-walking algorithms, the early developments include quasi-Newton methods introduced by Cerjan and Miller [5] and later modified by Simons and co-workers and Wales [6]. One of these methods widely used for *ab initio* molecular calculations is the partitioned rational function optimization algorithm of Baker [7], which utilizes either an approximate or exact Hessian matrix. For large systems exact Hessians have to be calculated regularly. If the Hessian matrix cannot be determined analytically, the second derivative matrix has to be determined numerically. For large systems this step often becomes prohibitively expensive.

There has been an idea of "eigenvector-following" methodology which uses only the min-mode of the Hessian matrix and moves the system uphill along this min-mode direction. The min-mode is the eigenvector corresponding to the minimal Hessian eigenvalue. This idea of gently ascending potential energy surface following such eigenvectors can be dated back to Crippen and Scheraga in 1971 [8]. There have been significant developments of new algorithms based on eigenvector-following methodology, for instance, the dimer method [9] and its many variants [10–12], the activation–relaxation techniques and the variants [13–15]. Instead of calculating the full Hessian matrix these algorithms calculate only the lowest eigenvalue and the corresponding eigenvector by finite difference scheme or Lanczos methods. Numerous applications for practical problems have proven that these eigenvector-following methods generally run faster and converge better than the previous Newton–Raphson root-finding methods.

Along with the substantial progress in algorithmic developments and applications of this popular eigenvector-following methodology, a first rigorous mathematical analysis for the local convergence to index-1 saddle point is established in [16] by formulating eigenvector-following methodology as a coupled dynamical system, with the name "gentlest ascent dynamics" (GAD). Index-1 saddle point on the potential energy surface becomes a locally stable equilibrium point in the GAD, thus the local (linear) convergence to index-1 saddle points is guaranteed. The applications of the GAD include [17] and [18]. A different but similar dynamical system named shrinkage dimer dynamics is also pursued in [19] by introducing one more dimer length variable. The method in [20] and the application in [21] combined a modified string method technique in which one end of the string follows a modified dynamics and then the inexact Newton method.

Based on the work of the GAD, we recently proposed a new description of the eigenvector-following' methodology in [22]. This new model is an iterative mapping and is named as "iterative minimization formulation", or IMF in short. In each iteration of the IMF, an auxiliary function is constructed as a new objective function, based on the local quantities around the current point on the potential energy. Then a local minimizer of this objective function is assigned as the new position for the next iteration. Theoretically, this iterative scheme can be locally described by a continuously differentiable mapping near the saddle point. We proved that the iterative mapping defined in such ways has quadratic convergence rate. This rate is the best rate for all numeral schemes based on the eigenvector-following methodology by using only the min-mode.

In the work of the IMF [22], we did not specify how to solve the subproblem of minimizing auxiliary objective function at each iteration, which is a crucial step in computations. This article is to address this practical issue of algorithm design for the IMF. It is already observed in [22] that the GAD is equivalent to solving the minimization subproblem in the IMF by using a single steepest descent step. We continue in this article to reveal the connections of other eigenvector-following based methods to the traditional optimization schemes for the subproblem in the IMF. See the detailed discussion in the Appendix A. So, the IMF is a quite general and rigorous mathematical description and offers a unified description of various algorithms related to the eigenvector-following methodology.

The contribution of this paper is to present an efficient and easy-to-implement algorithm for the IMF. To alleviate the computational bottleneck for the IMF, we propose an adaptive stopping rule for solving the subproblem of minimizing the new objective function at each iteration. Meanwhile, we propose an efficient proposal for the calculation of the gradient of the new objective function, which requires the minimal number of force calculations of the original potential function. Therefore, with our stopping rule and gradient calculation scheme, any gradient-based optimization solver can be exploited in principle as a good inexact solver for the subproblem. We call our proposed algorithm as "iterative minimization algorithm" (IMA). The main advantage of the IMA is its efficiency: on the one side, it explores the quadratic iterative rate of the IMF and on the other side, it minimizes the computational cost for the subproblem solver. As a result, the IMA not only needs less number of the computation-intensive rotation steps, but also reduces the overall computational cost to achieve a prescribed accuracy.

The rest of this paper is organized as follows. In Section 2, we review the iterative minimization formulation Our main work of the new algorithm is presented in Section 3. Section 4 includes several numerical examples. Section 5 is the concluding summary. The Appendix A is about the analysis of a series of existing algorithms by the perspective of the IMF.

2. Iterative minimization formulation

2.1. The problem

Let M be a Hilbert space with norm $|\cdot|$ and inner product $\langle\cdot,\cdot\rangle$ in its tangent space T_M. Given a differentiable (and sufficiently smooth) potential function $V : M \to \mathbb{R}$. The vector field ∇V generates the gradient descent flow on M,

$$\frac{d}{dt}\phi_t(x) = -\nabla V(\phi(x,t)),\ \forall t \in \mathbb{R}; \quad \phi_0(x) = x,$$

where ∇V is the (Fréchet) derivative of V. We call a point $x \in M$ critical if $\nabla V(x) = 0$. A critical point x is called a local minimizer for V on M if $V(y) \geq V(x)$ for all $y \in U$ for some neighborhood U of x. If the steepest descent flow $\phi_t(x)$ converges to some local minimizer x^* as $t \to \infty$, x is said to lie in the basin of attraction of x^*.

The transition state of our interest is the index-1 saddle point defined as follows. At each critical point x, the (local) unstable manifold for a neighborhood U of x is

$$\mathcal{M}^u(x) = \{x' \in U : \phi_t(x') \to x \text{ as } t \to -\infty, \text{ and } \phi_t(x') \in U \text{ for all } t \leq 0\}$$

If the dimension of this unstable manifold $\mathcal{M}^u(x)$ is one, the critical point x is called index-1 saddle point. An alternative description of index-1 saddle point is the solution of the min-max problem

$$\inf_{p \in P} \sup_{x \in p} V(x) \tag{1}$$

where P is the collection of all one dimensional paths connecting two points lying in the two neighboring basins of local minima. The minimizing path p_* for $\inf_{p\in P}$ is the unstable manifold of the saddle point on the separatrix. If the potential function V is sufficiently smooth, then linearization around the index-1 saddle point suggests that at the saddle, the Hessian $H := \nabla^2 V$ has only one negative eigenvalue and the corresponding eigen-space is the tangent space of the unstable manifold. We assume that all eigenvalues are distinctive at any fixed point through this paper for simplicity. In the eigenvector-following methodology, the key element is the min-mode: the eigenvector of the Hessian corresponding to the minimal eigenvalue. We use $v(x)$ to denote the min-mode at x, which minimizes the Rayleigh quotient. So,

$$v(x) := \operatorname*{argmin}_{|u|=1} \langle u, H(x)u \rangle .$$

If $V \in C^3(M)$, $v(x)$ is a differential function at x from the space M to its tangent space T_M when there is no crossing of eigenvalues.

2.2. Iterative Minimization Formulation (IMF)

Many existing algorithms based on eigenvector-following methodology focus on the inverted force along the min-mode direction. By contrast, the IMF in [22] regards the modified force in these methods as a certain search direction of an optimization problem. The heuristic viewpoint for the idea of IMF is to interpret (1) near an index-1 saddle x_* as

$$\inf_{p \in P} \ (\inf_{x \in p} \ -V(x)),$$

which could be regarded as an minimization problem for some altered function by using the information of the optimal p_*, whose tangent is the min-mode $v(x_*)$.

Rigorously, the IMF considers the minimization problems locally defined as follows for a given x

$$\min_{y \in U(x)} L(y; x), \tag{2}$$

where the neighborhood $U(x)$ means the solution for y of interest here is the one close to the point x. In designing practical local search algorithms for (2), this means the initial guess y_{init} must be x. The auxiliary function L in (2) is defined by

$$\begin{aligned} L(y; x) = (1-\alpha)V(y) \ &+ \alpha V\big(y - (v(x) \otimes v(x))(y - x)\big) \\ &- \beta V\big(x + (v(x) \otimes v(x))(y - x)\big) \end{aligned} \tag{3}$$

where the unit direction $v(x)$ means the min-mode at x. The notation $u \otimes v$ denotes the tensor defined by $(u \otimes v)x = \langle v, x \rangle u$ for $u, v \in T_M$. Here we require that the two constants α and β satisfy $\alpha + \beta > 1$ for the convergence to index-1 saddle point. We emphasize that $L(y; x)$ is a function of y. Then the iterative mapping $x^{(k)} \to x^{(k+1)}$ in the IMF is defined by

$$\begin{cases} v^{(k+1)} = \operatorname{argmin}_{\|u\|=1} u^{\mathsf{T}} H(x^{(k)}) u, & \text{(a)} \\ x^{(k+1)} = \operatorname{argmin}_{y \in U(x^{(k)})} L\left(y; x^{(k)}\right). & \text{(b)} \end{cases} \tag{4}$$

There are two special choices of α, β:

- $(\alpha, \beta) = (2, 0)$, then $L = L_1(y; x) = -V(y) + 2V\big(y - (v \otimes v)(y - x)\big)$,
- $(\alpha, \beta) = (0, 2)$, then $L = L_2(y; x) = V(y) - 2V\big(x + (v \otimes v)(y - x)\big)$.

In these two special cases, the additional terms are added to the original potential function V and the sign-reversed potential $-V$, respectively, in order to invert the shape of the potential surface along the min-mode direction. The main properties of this objective function L for any $\alpha + \beta > 1$ are listed as follows. Refer to [22] for the proof and other details.

Theorem 1. *Suppose that x_* is an index-1 saddle point of a C^4-function $V(x)$, and the auxiliary function L is defined by (3) with $\alpha+\beta>1$, then*

(1) *x_* is a local minimizer of $L(y;x_*)$.*
(2) *There exists a neighborhood $\mathcal{U}$ of x_* such that for any $x\in\mathcal{U}$, $L(y;x)$ is strictly convex in $y\in\mathcal{U}$ and thus has a unique minimum in $\mathcal{U}$.*
(3) *Define the mapping $\Phi: x\in\mathcal{U}\to\Phi(x)\in\mathcal{U}$ to be the unique local minimizer of L in $\mathcal{U}$ for any $x\in\mathcal{U}$. Further assume that $\mathcal{U}$ contains no other stationary point of V except x_*. Then the mapping Φ has only one fixed point x_*.*
(4) *$\Phi(x)$ is differentiable in $\mathcal{U}$. The derivative of Φ vanishes at x_*, i.e., the Jacobian matrix $\Phi_x(x_*)=0$.*
(5) *The iterative scheme $x^{(k+1)}=\Phi(x^{(k)})$ has exactly the second order (local) convergence rate, i.e., there is a positive constant c such that if the initial starting point $x^{(0)}$ is sufficiently close to x_*, then for sufficient large k, $|x^{(k+1)}-x_*|\le c|x^{(k)}-x_*|^2$.*

We list some useful calculations for the function L. First, let $g(\cdot)=\nabla V(\cdot)$ and $H(\cdot)=\nabla^2 V(\cdot)$, then the gradient and the Hessian of L (with respect to y) are, respectively,

$$\begin{aligned}\nabla L(y;x)=(1-\alpha)g(y)\ &+\alpha(I-v\otimes v)g\big(y-(v\otimes v)(y-x)\big)\\ &-\beta(v\otimes v)g\big(x+(v\otimes v)(y-x)\big),\end{aligned}\tag{5}$$

and

$$\begin{aligned}\nabla^2 L(y;x)=(1-\alpha)H(y)\ &+\alpha(I-v\otimes v)H\big(y-(v\otimes v)(y-x)\big)(I-v\otimes v)\\ &-\beta(v\otimes v)H\big(x+(v\otimes v)(y-x)\big)(v\otimes v).\end{aligned}\tag{6}$$

The x variable in the min-mode $v(x)$ is dropped out here for convenience. (5) needs the evaluations of the function g at three different locations. Expand the last two terms in g at the frozen position x up to the Hessian terms, then we have that by using the eigen-relation $Hv=\lambda v$,

$$\begin{aligned}\nabla L(y;x)=&(1-\alpha)g(y)+\alpha g(x)-(\alpha+\beta)\langle v(x),g(x)\rangle v(x)\\ &+\alpha(H(x)-\lambda(x)v(x)\otimes v(x))(y-x)-\beta\lambda(x)\langle v(x),y-x\rangle v(x)+O(|y-x|^2).\end{aligned}\tag{7}$$

In particular when $(\alpha,\beta)=(0,2)$ is used for (7), the above approximation is

$$\nabla L_2(y;x)\approx\nabla\tilde{L}_2(y;x):=g(y)-2\langle v(x),g(x)\rangle v(x)-2\lambda(x)\langle v(x),y-x\rangle v(x),\tag{8}$$

which is the summation of the gradient $g(y)$, a constant term and a linear term (x and $g(x)$ are already known and thus fixed). So, the advantage in this $\nabla\tilde{L}_2$ is that only the calculation of the gradient g at the point y is needed.

When evaluated at the initial point $y=x$, the gradient and the Hessian of L become, respectively,

$$\nabla L(x;x)=\big(I-(\alpha+\beta)(v\otimes v)\big)g(x)\tag{9}$$

and

$$\nabla^2 L(x;x)=H(x)-(\alpha+\beta)\lambda(x)v\otimes v.\tag{10}$$

If x_* is the index-1 saddle point, then $\nabla^2 L(x_*;x_*)$ is positive-definite if and only if $\alpha+\beta>1$.

The main challenge to construct efficient practical algorithms based on Theorem 1 is the large cost of solving the subproblem $\min_y L(y;x)$ at each iteration. In the Appendix A, we show that many existing methods, like the GAD and the dimer method, and their improved variants, are different approximation methods to solve the subproblem $\min_y L(y;x)$. These approximations have cheaper computational cost since they basically use one-step solution as the output, but the bad news is that the accuracy is also significantly lost. We comment in the Appendix A about the linear efficiency of these methods.

2.3. Summary

From the analysis in the Appendix A, it is clear that many existing numerical schemes based on the eigenvector-following idea can be analyzed from a unified viewpoint of the IMF, although the original developments of these methods did not follow this line of thought. In practice, these schemes and their variants do increase efficiency and convergence speed in comparison to their original versions (e.g. [9]). However, neither of them have achieved super-linear or quadratic convergence rate. All of their convergent rates are at the linear scale, i.e., $\log|x^{(k)}-x_*|\sim ck$ for sufficient large k, although the constant c varies for different variants. From the viewpoint of the IMF, there is an important common feature for these existing algorithms which may limit their practical performances: only one-step numerical solution is calculated as the output of the subproblem $\min_{y\in U(x)} L(y;x)$. For example, this one-step operation is called the translation step in the dimer method and its variants. After the dimer is translated along the min-mode direction, then in the dimer method, there seems no way to translate it further for improvement unless the min-mode is updated again. However, for the IMF, the so-called translation step is replaced by solving a locally defined optimization subproblem based on the fixed min-mode direction

$v^{(k+1)}$ and the initial guess $x^{(k)}$. It is quite natural to solve this optimization problem *iteratively*, until some stopping criteria is satisfied, rather than just iterate one time. This will utilize the advantage of the quadratic convergence iterative rate in the IMF. The resulting new algorithm, named as "Iterative Minimization Algorithm" (IMA), is based on this idea. Such an iterative minimization procedure is actually quite common in numerical techniques for certain challenging optimization problems, such as in the augmented Lagrangian method for constrained nonlinear optimization, where the augmented Lagrangian function is iteratively minimized after each time the Lagrangian multiplier is updated. It turns out that the IMA's procedure of using a full optimization for the subproblem with good stopping rules will not bring much overhead in overall: instead, it increases practical efficiency for numerical computation. We shall demonstrate this observation in the section of numerical examples. Next, we present the details of our algorithm.

3. Iterative minimization algorithms

The IMF has two-level iterations. The top level is $x \to \Phi(x)$, which consists of a second-level iterative procedure to solve the subproblem for a given x. To avoid the abuse of the notion of "iteration", we call the top level as *cycle*. The k-th cycle means the step of $x^{(k)} \to x^{(k+1)} = \Phi(x^{(k)})$. This cycle has two tasks: solving the min-mode $v^{(k+1)} := v(x^{(k)})$ and minimizing the auxiliary function $L(y; x^{(k)})$ with initial $y_0 = x^{(k)}$. As mentioned above, we regard the task of finding $v(x^{(k)})$ as a standard procedure (rotation step) and we mainly focus on the task of minimizing $L(y; x^{(k)})$. In principle, any main-stream gradient-based nonlinear optimization methods can be applied to this minimization problem for $L(y; x^{(k)})$. Which nonlinear optimization solver in practice generally performs best is left for empirical tests for specific applications. We simply use the conjugate-gradient (CG) method as an example in this paper. When solving the subproblem with the fixed pair x and v, the CG method will produce a sequence $y_0, y_1, y_2, \ldots$ until some stopping rule is reached, say, at y_m. Then we set y_m as the value of $x^{(k+1)}$.

3.1. Choice of step size

We prefer numerical schemes without using line search to accommodate the case that the potential value $V(x)$ is not available. The step size of our choice is as follows. Assume that a certain iteration scheme $y_i \to y_{i+1}$ is used to solve the minimization problem $\min_y L(y; x)$ with initial $y_0 = x$. Suppose that a normalized search direction, p_i, has been determined at iteration i by the optimization solver. We select the step size ρ_i as a Newton step and then let $y_{i+1} = y_i + \rho_i p_i$. So, ρ_i minimizes the quadratic approximation of $l(\rho) := L(y_i + \rho p_i; x)$. Solving the 1D problem

$$\min_{\rho} \left(L(y_i; x) + \rho \langle \nabla L(y_i; x), p_i \rangle + \frac{\rho^2}{2} \left\langle p_i, \nabla^2 L(y_i; x) p_i \right\rangle \right),$$

we then have the formula of our step size:

$$\rho_i = -\frac{\langle p_i, \nabla L(y_i; x) \rangle}{\left\langle p_i, \nabla^2 L(y_i; x) p_i \right\rangle}. \tag{11}$$

As discussed above, the gradient $\nabla L(y, x)$ is usually approximated by (8) in practice to save two calculations of the potential force ∇V.

At the initial step $i = 0$, from (9) and (10), we have the following step size

$$\rho_0 = -\frac{\langle p_0, (I - (\alpha + \beta) v \otimes v) g \rangle}{\langle p_0, (I - (\alpha + \beta) v \otimes v) H p_0 \rangle}, \tag{12}$$

where $\lambda = \lambda(x)$, $g = g(x)$ and $H = H(x)$. This will give the one-step solution $y_1 = x + \rho_0 p_0$.

3.2. Stopping rule and inexact CG solver for the subproblem

We now discuss the key ingredients of the IMA in solving $\min L(y; x^{(k)})$: how to set the stopping rule and how to modify the original conjugate-gradient method. First, our proposal of the stopping rule is based on a simple heuristic argument that when the current $x^{(k)}$ is still far away from the true solution x_*, it is not so useful to solve the subproblem for $L(y; x^{(k)})$ very accurately. However, when $x^{(k)}$ is quite close to the true solution, it is better to solve the subproblem at a higher accuracy. The distance of the current value to the true solution of index-1 saddle point is practically measured by the gradient $|\nabla V(x^{(k)})|$. So, our stopping rule is in the form of

$$|\nabla_y L(y_m; x^{(k)})| \le \delta |\nabla V(x^{(k)})|^{\gamma} \tag{13}$$

where $\delta > 0$ is a constant and the power exponent γ is a positive number with a recommended value between 1 and 2.

The main flow chart of the IMA is illustrated in Fig. 1. The tolerance *tol* is specified by (13). In this figure, the solver for the subproblem is the conjugate-gradient method and the sequence $y_0, y_1, y_2, \ldots$ obtained by this CG optimization solver is written in an abstract iterative form $y_{i+1} = \mathrm{CG}(y_i)$. The specific scheme for this procedure is quite important for the efficiency since different schemes will lead to different numbers of force calculations. This is the second important issue

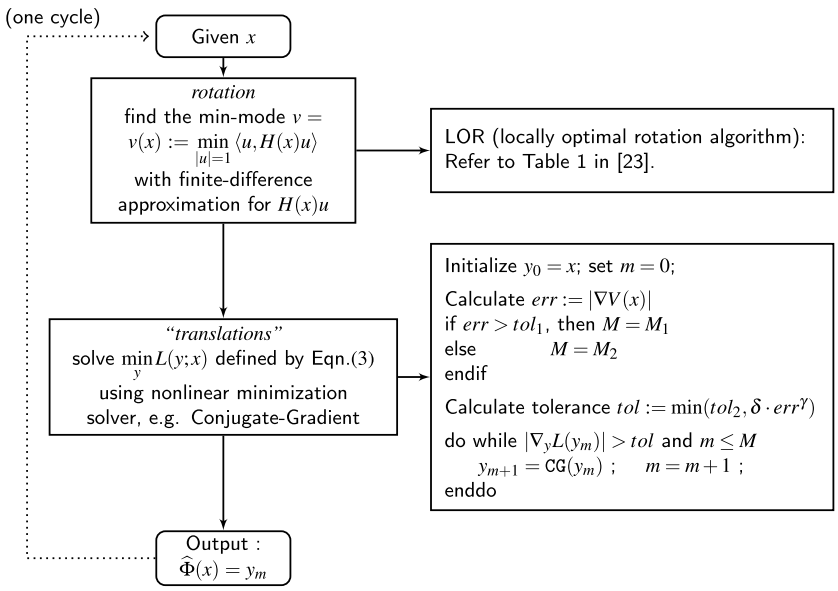


Fig. 1. Flow chart of the numerical method for one cycle in IMA: $x \to \widehat{\Phi}(x)$ ($\widehat{\Phi}$ is the numerical approximation of Φ). $tol_1 > tol_2 > 0$, $\delta > 0$ and $\gamma \in [1, 2]$ are the constant real parameters. The parameter tol_1 is the tolerance to check if the current value x is close to saddle point. The maximal iteration number $M_1 = 3 \sim 5$. The maximal iteration number M_2 is specified by the user, depending on the available computing resources. The power exponent γ is between 1 and 2.

that the IMA will address. We next give an efficient implementation of this CG procedure. For the sake of simplicity, the initial search direction for the first step $y_0 \to y_1$ is set to be the steepest descent direction. Other choices for this direction used in some variants of the dimer method might help better performance in practice. For the step size issue, we use the Newton's step size given by (11) at all iterations. Consequently, the line search for the calculation of the potential function V is avoided and only the calculation of ∇V is required. The most important thing to apply the CG scheme is that at each CG iteration, one needs to calculate the gradient $\nabla L(y;x)$ at $y = y_i$ to obtain the search direction p_i, and to calculate the gradient $\nabla L(y;x)$ at $y = y_i + hp_i$ to calculate the step size ρ_i in (11) (we used the forward finite difference scheme). The calculation of ∇L for any $y \neq x$ by directly following (5) involves three times of the calculations of the gradient ∇V, which are at y, and $y - (v \otimes v)(y - x)$ and $x + (v \otimes v)(y - x)$, respectively. For the latter two points, we seek the first order approximation of ∇L which is stated in (7), or $\nabla \tilde{L}$ in (27). This can reduce the number of force calculation to two if the Hessian involving the coefficient α is approximated again by the forward difference scheme. This calculation involving the Hessian with α can be also avoided by selecting $\alpha = 0, \beta = 2$, i.e., using $\tilde{L}_2$. In a nutshell, for the CG, the gradient $\nabla L(y;x)$ is numerically evaluated by the formula $\nabla \tilde{L}_2(y;x)$ in (8), which only requires one calculation of ∇V at y. Then every single iteration $y_{i+1} = \mathtt{CG}(y_i)$ calculates the force twice. Algorithm 1 shows the details of this CG scheme.

In our IMA, the "translation" step does not carry the traditional meaning as the class of the traditional algorithms based on the dimer method since after one rotation step, the iteration $y_{i+1} = \mathtt{CG}(y_i)$ is carried out repeatedly for $i = 0, 1, 2, \ldots$ until the stopping rule is satisfied with a cutoff at a maximal iteration number M_2. But since the computational cost of one iteration $y_{i+1} = \mathtt{CG}(y_i)$ is comparable to one translation step in the traditional methods, we may roughly state that our proposed algorithm is composed of "(rotation-translation-…-translation)-(rotation-translation-…-translation)-…", while the traditional methods have the pattern "(rotation-translation)-(rotation-translation)-…": here each bracket corresponds to one cycle (Fig. 1). One important fact, which is supported by numerical evidence in the next section, is that our IMA will use a much smaller number of cycles and thus less rotation steps.

3.3. Superlinear efficiency

Before showing numerical examples to demonstrate the efficiency of the IMA, we provide some theoretical arguments why the IMA can achieve super-linear efficiency by using the stopping rule in (13).

Let N be the number of cycles. Denote c_{rot} as the typical cost of calculating the min-mode and $c_V(k)$, $c_g(k)$ as the cost of calculating the potential and its gradient, respectively, during solving the subproblem of minimizing L at the k-th cycle, $k = 1, 2, \ldots, N$. For the problem in high d dimension, $c_g \approx d \cdot c_V \gg c_V$. In addition, many algorithms do not need the potential energy itself, but only use its gradient which is calculated, for instant, from the *ab initio* first principle quantum-chemistry packages. In such cases, $c_V = 0$. The total computational cost for N cycles is then

$$C(N) = Nc_{rot} + \sum_{k=1}^{N} (c_V(k) + c_g(k)) \approx Nc_{rot} + \sum_{k=1}^{N} c_g(k).$$

Algorithm 1: Solve $\min_y L(y;x)$ by conjugate gradient method without line search.

input : the position x, the min-mode v, the minimal eigenvalue λ, the gradient $g=\nabla V(x)$ and the dimer length h
output: inexact solution of $\min_y L(y;x)$
begin

```
y_0 = x ;                                                        // initial value must be x
p'_0 = 2⟨v, g⟩v − g;  p_0 = p'_0/|p'_0| ;                        // −∇L(y_0; x) = 2⟨v, g⟩v − g. Eqn. (9)
f_0 = p'_0 ;                                                     // f†(x) = 2⟨v, g⟩v − g. Eqn. (12)
g_h = ∇V(x + hp_0);  f_h = 2⟨v, g_h⟩v − g_h ;                     // f†(x + hp_0)
ρ_0 = ⟨p_0, f_0⟩h / ⟨p_0, f_h − f_0⟩ ;                            // step size, Eqn. (22)
y_1 = y_0 + ρ_0 p_0
for i = 1, 2, ... do
    if the tolerance is not achieved then
        f_i = −∇V(y_i) + 2⟨v, g⟩v + 2λ⟨v, y_i − x⟩v ;             // −∇L̃_2(y_i; x). Eqn. (8)
        β_i^PR = ⟨f_i, f_i − f_{i−1}⟩ / ⟨f_{i−1}, f_{i−1}⟩ ;      // Polak-Ribière type CG
        p'_i = f_i + β_i^PR p'_{i−1};  p_i = p'_i/|p'_i|;
        f_h = −∇V(y_i + hp_i) + 2⟨v, g⟩v + 2λ⟨v, y_i + hp_i − x⟩v ;   // −∇L̃_2(y_i + hp_i; x)
        ρ_i = ⟨p_i, f_i⟩h / ⟨p_i, f_h − f_i⟩ ;                    // step size, Eqn. (12)
        y_{i+1} = y_i + ρ_i p_i ;
    else
        stop ;
```

The unit to measure the computational cost for benchmarking efficiency can be chosen as one calculation of the force ∇V. We remark that typically, the rotation step has a bigger overhead and requires more numbers of force calculations than the translation step. This means that the cost of rotation c_{rot} is typically larger than the cost of force calculation c_g.

For the linear convergent methods such as the dimer methods and the GAD one has that $|x^{(k)}-x_*|\le c|x^{(k-1)}-x_*|\le c^k|x^{(0)}-x_*|$ for a constant $c\in(0,1)$. Every translation step in these methods involves a fixed number of force computation (usually once or twice). So the cost $C(N)$ is linear in N, say $C(N)\sim N(c_{rot}+c_g)$. This implies that to achieve a prescribed error tolerance ε for $|x^{(N)}-x_*|\le\varepsilon$, the total computational cost one has to pay is around $C_\varepsilon\sim|\log\varepsilon|$. We call this scale of total cost as linear efficiency. If the cost grows slower than $O(|\log\varepsilon|)$, we call the algorithm has superlinear efficiency.

Next we show that our IMA illustrated in Fig. 1 can achieve super-linear sufficiency. The short derivation given below is to present an heuristic explanation, rather than in pursuit of a rigorous and lengthy proof. First, for simplification and to better focus on the effect of inexact solver of the subproblem for L, we assume that the min-mode $v(x)$ appearing in the auxiliary function $L(y;x)$ is precise. This assumption is reasonable because in practical applications of the dimer methods, usually it is important to maintain a quite accurate calculation of the min-mode in the rotation step. This assumption also implies that the dimer length h is sufficiently small.

For the subproblem of solving $\min_y L(y;x)$, we suppose that the solver has a p-th order convergence rate, with $p\in(1,2]$; for instance, $p=2$ for the exact Newton method and $p<2$ for super-linear algorithms like CG or L-BFGS method. Let $\{x^{(k)}:k=1,2,\dots,N\}$ be the sequence of numerical solutions produced during the total N cycles by the IMA. Define the error $\varepsilon_k=|x^{(k)}-x_*|$. Then

$$\varepsilon_{k+1}\le|x^{(k+1)}-\Phi(x^{(k)})|+|\Phi(x^{(k)})-x_*|. \tag{14}$$

Note that $\Phi(x^{(k)})$ is the true solution of the subproblem and $x^{(k+1)}$ is the numerical solution, then by the stopping rule we proposed in the IMA, the following is derived

$$|x^{(k+1)}-\Phi(x^{(k)})|\le c|\nabla V(x^{(k)})|^\gamma=c|\nabla V(x^{(k)})-\nabla V(x_*)|^\gamma\sim|x^{(k)}-x_*|^\gamma=\varepsilon_k^\gamma$$

where c is a generic positive constant. Here the notation $a_k\sim b_k$ for two sequences a and b means $\lim_{k\to\infty}\frac{a_k}{b_k}\in(0,\infty)$. Applying the Taylor expansion to Φ, we have

$$|\Phi(x^{(k)})-x_*|=|\Phi(x^{(k)})-\Phi(x_*)|=|\Phi_x(x^*)|\varepsilon_k+c|\Phi_{xx}(x^*)|\varepsilon_k^2\sim\varepsilon_k^2$$

due to the conclusions $x_*=\Phi(x_*)$ and the Jacobian $\Phi_x(x_*)=0$ in Theorem 1. So, combining the above two results, we have from (14)

$$\varepsilon_{k+1}\sim(\varepsilon_k^\gamma+\varepsilon_k^2)\sim\varepsilon_k^{\min\{2,\gamma\}}=\varepsilon_k^\gamma,\qquad\gamma\le2. \tag{15}$$

This shows that the decay rate of the error is controlled by the second order of Φ's iteration and the exponent γ in stopping criteria. For a prescribed accuracy ε, the total number of cycles N in need to achieve this accuracy is decided by the equation $\varepsilon_N=\varepsilon$. The calculation based on (15) shows that

$$N\gtrsim\frac{1}{\log\gamma}\log\frac{\log\varepsilon}{\log\varepsilon_0}. \tag{16}$$

Therefore, the total number of cycles for the IMA has a scale $(\log|\log\varepsilon|)$, which is much smaller than $|\log\varepsilon|$ for the old methods.

Now we estimate the cost $c_g(k)$ during the k-th cycle for the tolerance $c|\nabla V(x_k)|^\gamma \approx c|x_k - x_*|^\gamma \sim \varepsilon_k^\gamma$. Fix k now. Define the i-th step error $\delta_i := |y_i - \Phi(x^{(k)})|$. Since the order of the solver for the subproblem is p, then $\delta_i \le c\delta_{i-1}^p$, which gives $\log\delta_i \le p^i \log(c\delta_0)$ where $c\delta_0 < 1$. The number of steps to reach the tolerance ε_k^γ, denoted as m, is decided by

$$p^m \log(c\delta_0) \sim \log(\varepsilon_k^\gamma) = \gamma \log\varepsilon_k.$$

Since $y_0 = x^{(k)}$, the initial error $\delta_0 = |x^{(k)} - \Phi(x^{(k)})| \le |x^{(k)} - x_*| + |x_* - \Phi(x^{(k)})| \approx |x^{(k)} - x_*| + 1/2|\nabla^2\Phi_*|^2|x_* - x^{(k)}|^2 \sim \varepsilon_k$. So, we have $m \lesssim \frac{1}{\log p}\log\frac{\gamma\log\varepsilon_k}{\log(c\delta_0)}$. Note that $c_g(k)$, the cost during these m steps for this k-th cycle is, $2m$. Using (15), i.e., $\log\varepsilon = \log\varepsilon_N = \gamma^{N-k}\log\varepsilon_k$, for $\varepsilon_N = \varepsilon$, and using (16) we have that

$$c_g(k) \lesssim \frac{2}{\log p}\log\frac{\gamma\log\varepsilon_k}{\log(c\delta_0)} = \frac{2}{\log p}\log\frac{\gamma^{k+1}\log\varepsilon}{\log(c\delta_0)} \sim \frac{2\log\gamma}{\log p}(k+1) \sim k.$$

The above result implies that $c_g(k)$ grows linearly for large k. This good feature is due to two important features in our solver for the subproblem $\min L(\cdot; x_k)$: (1) the initial guess is $y_0 = x^{(k)}$; (2) the stopping rule is $\delta_m \sim \varepsilon_k^\gamma$.

Therefore, for the IMA, we have that the total cost $C(N) = Nc_{rot} + \sum_{k=1}^N c_g(k) \sim N^2$, thus, the cost to reach accuracy ε is

$$C_\varepsilon^{\mathrm{IMA}} \sim N^2 \sim (\log|\log\varepsilon|)^2. \tag{17}$$

Compared to the scale $C_\varepsilon = O(|\log\varepsilon|)$ for the old methods discussed in Section 5, (17) suggests the efficiency of the IMA improves much.

3.4. Rotation algorithm

We give a few remarks about the rotation step. First, since the Hessian-vector multiplication Hv along a direction v is calculated by finite difference scheme with a dimer length h, with a cost independent of the value of h, it is always preferred to use h as small as possible to reduce the finite differencing error. We suggest practically using h much smaller than the ultimate desired accuracy ε. The use of a time-varying dimer length in [19] does not benefit much, compared to the use of the final and the smallest h. For a numerical analysis including the effect of h, refer to [12].

When the gradient ∇V is obtained from *ab initio* quantum-chemistry calculation or molecular dynamical simulation (such as the free energy surface calculation in [24]), the numerical value of the gradient with statistical errors cannot achieve arbitrary accuracy and this error of the numerical gradient ∇V imposes a natural lower bound on the size of h. The use of too small h in such cases does not help improve accuracy. Refer to the examples in [23].

The use of which finite difference, the central scheme or the forward scheme, is a trade-off in practice. The central scheme has a smaller error $O(h^2)$, but it needs two force calculations at $x^{(k)} \pm hv$ respectively. For the forward scheme, the error is a bit large $O(h)$, but the calculation of force is only required for one time at $x^{(k)} + hv$ since $\nabla V(x^{(k)})$ is already prepared. Hence, the central scheme is probably preferred if there is no way to use arbitrarily small h in practice for the reason mentioned above.

In principle, our adaptive control of the tolerance for the subproblem should also be applied to the rotation step for the calculation of the min-mode. This could further reduce some computational cost of rotations in first few cycles. We did not fully test this effectiveness numerically here. Yet, our numerical experience seems to suggest that to maintain a sufficient small numerical error for the rotation step is beneficial for the robustness of the algorithm. We recommend to use the same numerical scheme and accuracy control as the existing dimer methods do. The only change for the code of the existing dimer methods is in the part of translation.

In the end, we recommend a rotation algorithm proposed in [23]. This locally optimal search direction finding algorithm, namely "LOR", based on the extension of the traditional conjugate-gradient method, has proved to be one of the most efficiency rotation algorithms by several dozens of real applications in chemical calculations. The detail algorithm can be found in Table 1 of [23].

3.5. Choice of initial state for IMA

The eigenvector-following methodology for surface-walking algorithms relies on the local features around the index-1 saddle points. The convergence property is only local and we expect the good convergence behavior of these algorithms only when there does exist an index-1 saddle point not far away from the initial guess. However, the most common initial state used in practice is a local minimum of the potential energy surface (if it is not a local minimum, it is easy to find one from this point), which is in a convex region of the potential surface V where $\lambda(x) > 0$. In many algorithmic implements, e.g., Eqn. (20) in [10], the modified force to evolve the dimer is simply the negative of the potential force along the direction $v(x)$ in the convex region and only when $\lambda(x) < 0$ is it switched to the dimer force or the GAD force to start to apply the eigenvector-following idea. A maximal step length is practically imposed to avoid instability in the convex region.

Algorithm 2: The dimer method with CG search direction based on [10].

input : initial guess x_0, dimer length h
begin
  Solve the min-mode v_0 and the minimal eigenvalue λ_0;
  for $k = 0, 1, 2, 3, \ldots$ **do**
    if *not converge* **then**
      $g_k = \nabla V(x_k)$;
      if $\lambda_k > 0$ **then**
        $f_k^\dagger = \langle g_k, v_k \rangle v_k$;
      else
        $f_k^\dagger = -g_k + 2\langle g_k, v_k \rangle v_k$;
      if $k = 0$ **then**
        $p_k = f_k^\dagger / |f_k^\dagger|$;
      else
        $\beta_k = \frac{\left\langle f_k^\dagger, f_k^\dagger - f_{k-1}^\dagger \right\rangle}{|f_{k-1}^\dagger|^2}$;
        $p_k = (f_k^\dagger + \beta_k p_{k-1}) / |f_k^\dagger + \beta_k p_{k-1}|$;
      $g_h = \nabla V(x_k + h p_k)$;
      $f_h^\dagger = -g_h + 2\langle g_h, v_k \rangle v_k$;
      $\rho_k = \frac{\left\langle p_k, f_k^\dagger \right\rangle}{\left\langle p_k, (f_h^\dagger - f_k^\dagger)/h \right\rangle}$;
      $x_{k+1} = x_k + \rho_k p_k$;
    else
      stop;

The IMA also suffers from this subtle issue of choosing initial states. The above separate treatments for different signs of the curvature $\lambda(x)$ can be easily understood mathematically in the IMF through equation (10): $\lambda(x)$ controls the definitiveness of the Hessian of the auxiliary function at x. When x in the convex region of V, the Hessian $\nabla^2 L$ is not positive definitive due to one negative eigenvalue $(1-\alpha-\beta)\lambda(x)$; then at least locally, there is no minimizer for $L(y;x)$. The mapping $\Phi(x)$ is well-defined only when $\lambda(x) < 0$ and all other eigenvalues of $H(x)$ are positive: we name the connected set of points x satisfying these two conditions as the "index-1 region". Then we need an initial state lying in an index-1 region to apply the IMF mapping Φ. Therefore, all existing ad hoc techniques used in other methods to handle the convex region should also be applied to serve the IMA with feasible initial starting points in the index-1 region. The intersecting set of the convex region and the index-1 region is the collection of branching points (inflexion points) where $\lambda = 0$. So, it is quite natural right now to propose various numerical techniques to move from $\lambda(x) > 0$ to $\lambda(x) < 0$ by exploring the boundary between convex region and index-1 region. For example, one can first look for a branching point z (satisfying $\lambda(z) = 0$), by a minimization procedure for $|\lambda(z)|^2$ from a local minimum of V. Or, if one has two available points z_1 and z_2 in the convex region and an index-1 region respectively, then a binary search along the line $(1-t)z_1 + tz_2$ through z_1 and z_2 will also give some branching point $\lambda(z) = 0$ since the eigenvalue continuously depends on the Hessian. Unfortunately, we have no satisfied theoretic evidence to compare these different procedures to understand which point on the boundary of the convex and index-1 regions is the best candidate. We have developed (in another forthcoming paper) a semi-local convergence result of the mapping Φ in the IMF to describe the convergence domain similarly to the convergence theorem for the traditional Newton method. Yet this theoretic result, as a sufficient condition for convergence, is still difficult to verify in real applications for complex systems.

4. Numerical examples

In this section, we present the numerical results of our algorithms for several examples. The comparisons to the other methods such as dimer methods [10,12] are also provided. To show the performance of various algorithms, we plot the figures with the horizontal axis being the indicator of actual computational cost measured by the number of force calculations. The numerical error in the vertical axis is measure by the gradient $|\nabla V(x^{(k)})|$. We remark that it is not appropriate to use the energy difference $|V(x^{(k)}) - V(x_*)|$ as the measurement of errors in [11].

The version of the dimer method we used for the benchmark follows the flow sheet detailed in [10], except that the method of calculation of the min-mode is replaced by the LOR. Compared with other versions of the dimer method, this variant uses analogy of CG method to update the search direction. The detail of this variant is specified in Algorithm 2. Note that here we use the step size based on the Newton step (22) (i.e., Eqn. (21) in [10]) rather than the linear inter/extrapolation approach in the improved version of [10], for the consistency with the procedure of Algorithm 1.

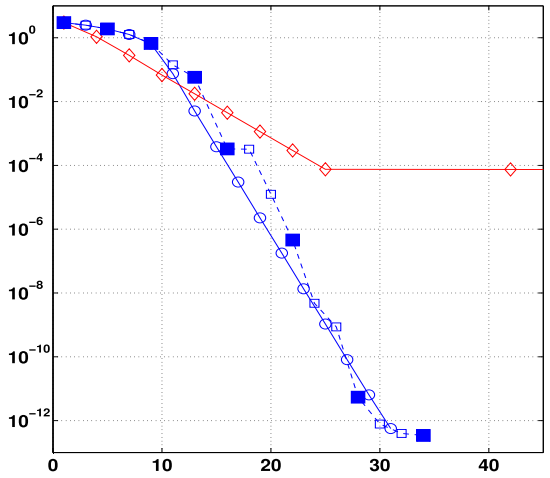


Fig. 2. The "$\circ, \diamond, \square$" markers correspond to the dimer method in Algorithm 2, the other variant of dimer method in [12], and the IMA ($\gamma = 2$), respectively. The solid squares (■) for the IMA are used to highlight the output after each complete cycle, i.e., at $x^{(0)}, x^{(1)}, \cdots, x^{(N)}$. The horizontal axis is the number of force calculation *excluding* the cost for rotations. The vertical axis is the L_2 norm of the gradient $|\nabla V|$.

4.1. A 2D example

We first consider a two dimensional energy function studied in [22]. The formula of this energy function is

$$V(x, y) = 3e^{-x^2-(y-\frac{1}{3})^2} - 3e^{-x^2-(y-\frac{5}{3})^2} - 5e^{(x-1)^2-y^2} - 5e^{(x+1)^2-y^2} + 0.2x^4 + 0.2\,(y - 1/3)^4\,,$$

which has three minima and three saddle points. Refer to [22] for the contour plot of this function and the positions of these fixed points. Since the Hessian is 2×2, we use MATLAB's '`eig`' subroutine for the calculation of the min-mode with the analytic expression of the Hessian. So, the min-mode is precise. But in the translations steps or solving the subproblem in the IMA, we strictly follow Algorithm 2 and Algorithm 1 for comparison and use a very small dimer length h in calculation with a forward finite difference scheme instead of using the known Hessian matrix and matrix–vector multiplication. Therefore, we exclude the computational cost solely from the rotations steps and truly count the costs from the translations in the dimer method and the total number of cycles in the IMA. In this way, we can use this 2D example to illustrate how the algorithms affect the total number of cycles, and the total computational cost spent on the subproblem calculations or the translation steps.

The saddle we calculated is located at $(-0.6172723079, 1.1027345175)$. We pick up a nearby initial value $(-0.77178, 0.62721)$. The parameters in Fig. 1 are $tol_2 = 0.02$ and $\delta = 0.1$. The max number of steps in CG solver is $M_1 = 2$ and $M_2 = 3$. The dimer length for the forward difference in calculating the search direction is $h = 10^{-13}$.

Fig. 2 shows the decay of the errors (vertical axis) versus the growth of force calculations excluding the rotation costs (horizontal axis). The circle markers correspond to the dimer method, Algorithm 2, in which every translation step needs two force calculations, at x_k and $x_k + hp_k$, respectively. The square markers (filled and unfilled) correspond to the IMA with $\gamma = 2$. The filled square markers mean that the results are plotted where each IMF cycle has completed, i.e., at $\{x^{(k)}\}$; the unfilled square markers show the detailed information inside each cycle, i.e., including $\{y_0, y_1, \ldots, y_m\}$. The min-modes are thus calculated only at the points of filled squares and it is unchanged during two successive filled square markers. But for the dimer method the min-modes are recalculated at each circle markers. Fig. 2 shows that the practical performance of the two methods are almost the same for the cost coming from the translation steps. However, the number of cycles of the IMA is only one half or one third of the iteration number of the dimer method, which implies the same amount of the reduced cost for the rotations steps. Although the cost reduction for the rotations is not counted for this toy example, it will have a significant impact on real problems in higher dimensions where the rotation for the min-mode is non-trivially computation-intensive.

The result from another variant of the dimer method [12] is also shown in diamond markers. In this method, when the error reaches around 10^{-4}, it stops decreasing and we observed that the initial step size of line search scheme specified in [12] is always accepted. This could be improved by resetting initial guess if such undesired cases occur. It is clear from this figure that its linear decay cannot beat the dimer method with Newton's step size we used here at the benchmark for our IMA, but this method seems more numerically robust in the first few iterations.

We also test the results for the various exponents γ and maximum number of steps M_2 in Fig. 3. The result gets a little worse if γ decreases from 2 to 1 for the reason that the stopping tolerance $|\nabla V(x_k)|$ is too big, and there is no much change for the result if M_2 increases from 3 to 4. We suggest using the typical values $\gamma = 2$.

To show the overall performance of the algorithms, we present some statistical results by choosing 100 random initial states on a circle with radius 0.2 whose center is the local minimum $(-1, 0)$. We set a very large $M_2 = 100$ for this example

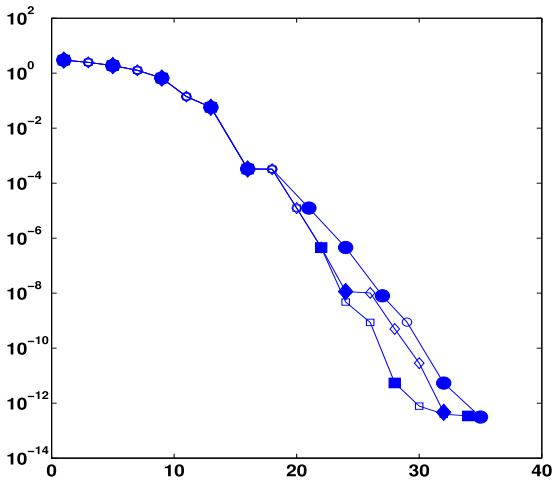


Fig. 3. The markers of "□, ○, ◇" (as well as their solid versions) correspond to different parameters in the IMA ($\gamma = 2$, $M_2 = 3$), ($\gamma = 1, M_2 = 3$) and ($\gamma = 2, M_2 = 4$) respectively. The quantities representing of two coordinate axes in this plot are the same as those in Fig. 2.

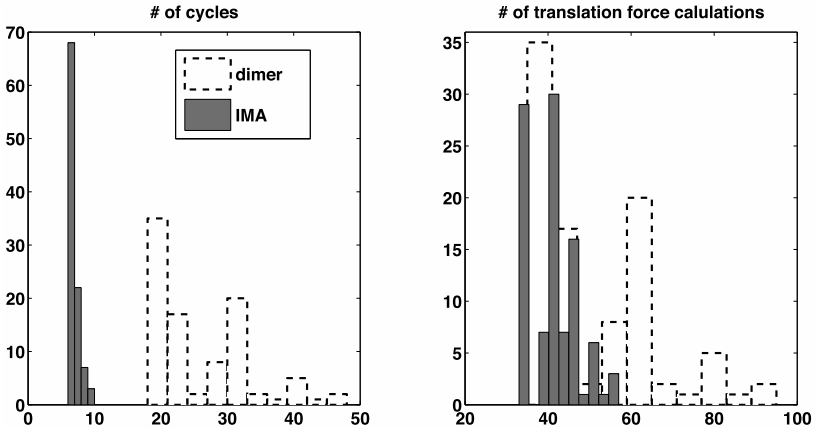


Fig. 4. The histograms of the number of cycles and the number of force calculations (excluding rotation) for the comparison of the IMA and the dimer method for the three-hole example from 100 random initial states. The dark and light colors are from the results from the IMA and the dimer method, respectively.

and we calculate the required number of cycles and the number of force evaluations (excluding rotation) for the error to reach 10^{-10}. First, we observed there are 7 initial states where the dimer method does not converge to any saddle point while none fails to converge in the IMA. For all states which have convergence, we plot the histogram of the data in Fig. 4. In the IMA, the average number of cycles for the IMA is 7.2 and the number of force calculations is 41.3, so the average number of iterations in the subproblem is $41.3/7.2/2 \approx 2.9$, which is much smaller than $M_2 = 100$ here. This also explains why the previous values $M_2 = 3$ and $M_2 = 4$ in Fig. 3 do not differ much. In the dimer method, the average number of iterations is 26 and the number of force calculations is $26 \times 2 = 52$, which is slightly higher than the number of force calculation in the IMA. The comparison of two average numbers of IMA cycles and dimer iterations, 7.2 versus 26, supports the claim that for this example, the IMA only requires around 30% of the number of rotation steps in the dimer method. Finally, one interesting observation of how the data points spread out in Fig. 4 suggests that the computational cost of the IMA seems less sensitive to the initial guesses than the dimer method.

Fig. 5 shows the domains of attraction of our method and the classic Newton's method for this 2-D example. We choose initial guesses from 50×50 grid points uniformly in the rectangular region $[-1.5, 1.5] \times [-1.5, 2.0]$. These points are labeled in Fig. 5 by three different marks in three colors, according to which saddle point (shown by a cross symbol in the same color as that of its initial guesses) they converge to, respectively. The grid point is left blank if no convergence. To see the effect of the adaptive stopping rule, we shut off this rule of *tol* in Fig. 5A and 5B but use the fixed number of iterations. The result from the adaptive stopping rule is shown in Fig. 5C. The result of Newton's method is in Fig. 5D. The observation is as follows. First, the inexact solver in Fig. 5A and Fig. 5C gives a significantly larger domain of attraction than the exact solver and the Newton's method. Second, the Newton's method is a little better than the exact solver for this example. This

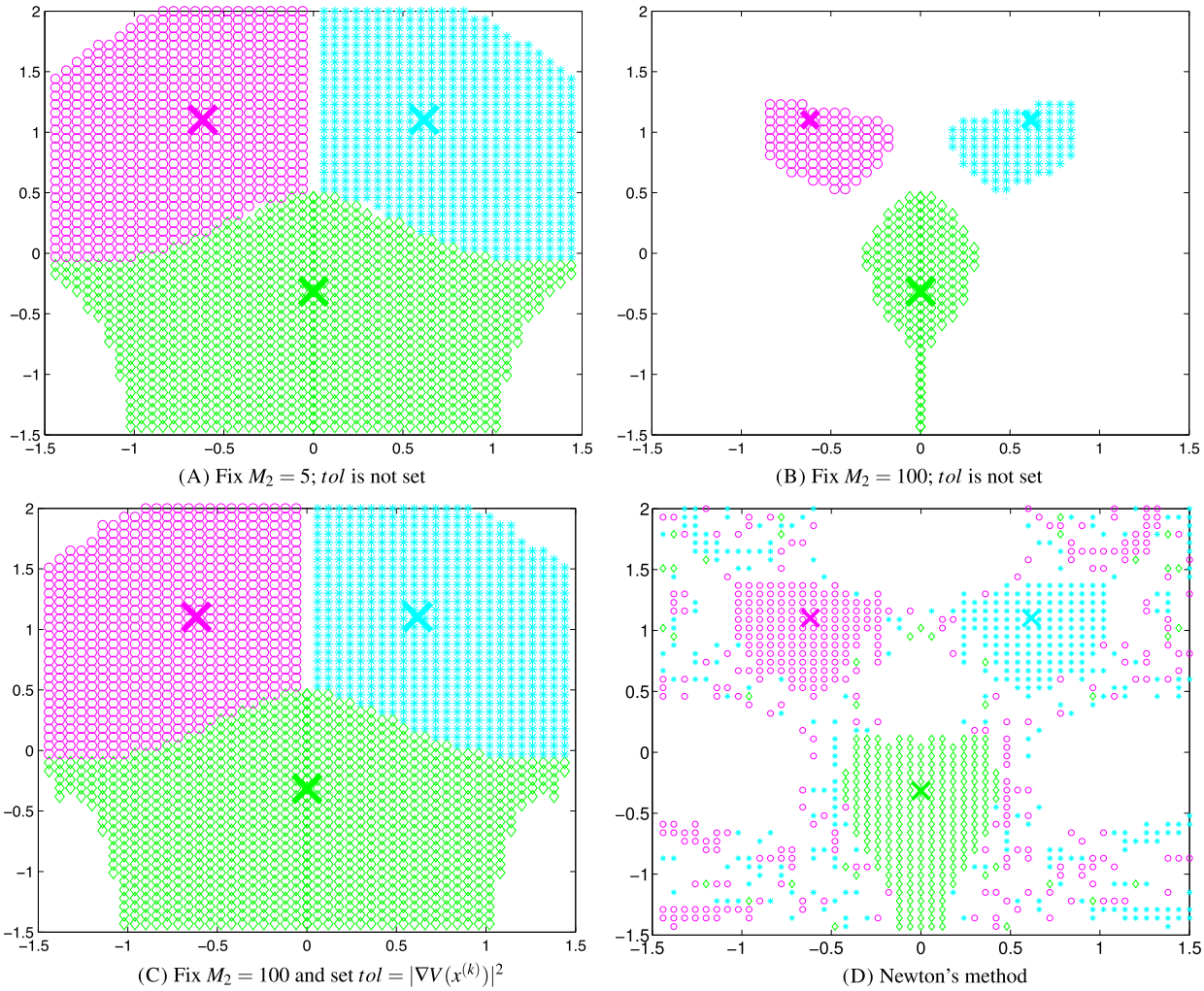


Fig. 5. Comparison of domains of attraction of saddle points for our IMA and the Newton's method. The cross symbols are three index-1 saddle points. (A) and (B): we do not apply the adaptive stopping rule associated with *tol*, but solve the subproblem with a fixed number of iterations: 5 and 100, respectively. For (B), the solution of the subproblem can be viewed as the exact solution since $M_2 = 100$ is very large; (C): the adaptive stopping rule $tol = |\nabla V(x)|^2$ is applied for the subproblem and in this case the solver stops only after $3 \sim 5$ iterations (consequently, $M_2 = 100$ does not take effect); (D): the Newton's method of solving $\nabla V(x) = 0$ is applied to search one of the three index-1 saddle points.

comparison suggests that the inexact solver does not only improve the efficiency by saving costs, but also increases stability of the algorithm because of a larger domain of attraction.

4.2. An atomistic model

This model has been studied extensively in the literature of calculating the transition states for a FCC metal. The atomistic system has a 6-layer slab, each of which contains 56 atoms, and 7 atoms at the top of the slabs. The bottom three layers in the slab are frozen. There are $56 \times 3 + 7 = 175$ identical atoms free to move. The interaction between the atoms is the simple pairwise additive Morse potential

$$V(R) = A[e^{-2a(R-R_0)} - 2e^{-a(R-R_0)}]$$

with parameters chosen to reproduce diffusion barriers on Pt surfaces ($A = 0.7102$ eV, $a = 1.6047$ Å^{-1}, $R_0 = 2.8970$ Å). This potential is cut and shifted by V(R_C) where $R_C = 9.5$ Å. The minimum energy lattice constant 2.74412 Å is used. Detail descriptions can be found in [25] and [22].

The dimension of this problem is $175 \times 3 = 525$. The rotation step in this example carries some computational overhead. We set the dimer length $h = 10^{-10}$. In our LOR algorithm for rotations, it was observed that the typical number of force calculation is around 4, that is $c_{rot} \approx 4$. The total maximal number of iterations for the subproblem is set as $M_2 = 50$. The initial guess has a 0.5 (Euclidean) distance from the true solution. Fig. 6 demonstrates the comparison of efficiency between

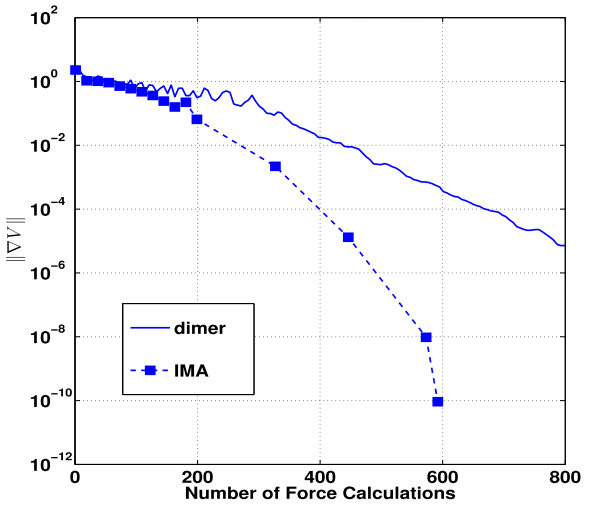


Fig. 6. The 175-atom model.

Table 1
The comparison of IMA and the dimer method from 13 different initial states for the 175-atom model. IMA_cyc is the number of cycles; dimer_iter is the number of iterations; IMA_f and dimer_f are the total number of force calculations.

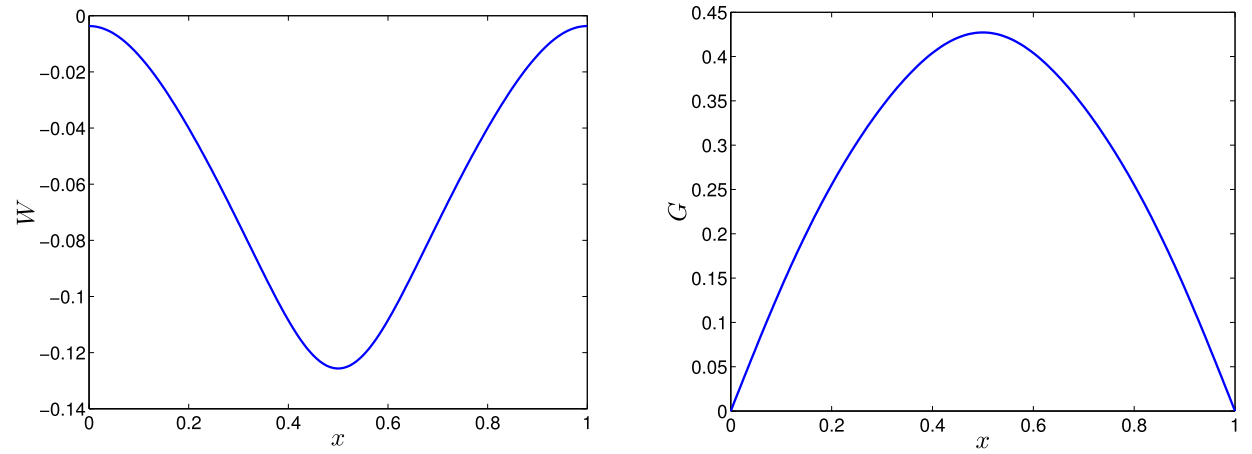
init	IMA_cyc	IMA_f	dimer_iter	dimer_f
min1	12	417	283	1412
min2	11	439	327	1632
min3	17	530	187	932
min4	13	444	346	1727
min5	19	537	203	1012
min6	12	381	220	1097
min7	18	569	209	1042
min8	10	440	273	1362
min9	15	534	178	887
min10	10	381	206	1027
min11	10	383	185	922
min12	9	321	127	632
min13	17	561	214	1067
average	13.3	457	228	1135

the dimer method for benchmarking and the IMA ($\gamma = 2$). Here the number of force calculations in the horizon is the overall cost including the rotations. The errors of the dimer method decay linearly in computational cost while the IMA shows the super-linear decay speed. Only four cycles are needed in the IMA to decrease the error from 10^{-2} to 10^{-10}.

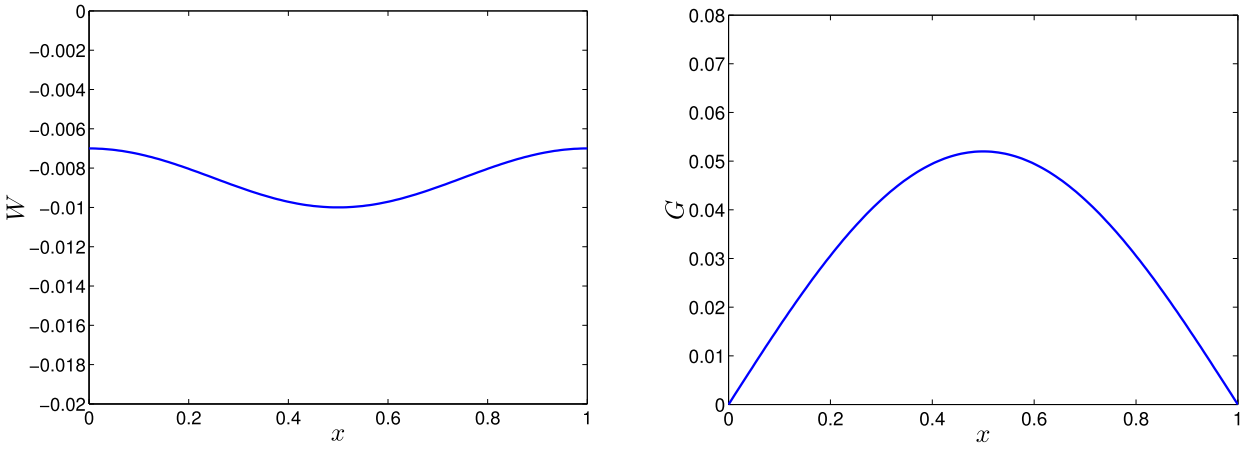
Since there are more than hundreds of local minima and saddle points for this atomic model, we next test our algorithm for a set of 13 local minima and 13 index-1 saddle points. The minima used here at initial guesses correspond to the online files from `min1a.con` to `min13a.con` at [26]. We add a small perturbation (with size 0.2) on these 13 minima as initial states. When the error is less than 10^{-9}, we stop the computation and measure the number of cycles and the total number of force calculations. Table 1 shows the comparison of the IMA and the dimer method. For all these initial states, both methods converge. The average number of cycles in the IMA (including the iteration number when x is in the convex region) is around 13. The total number of force calculations in the IMA is reduced by more than half than the dimer method.

4.3. Strain field of hyper-elastic plate

We consider the model in [27] arising from the bulking problem of hyper-elastic plate. The bifurcation analysis and post-buckling solutions under different stress values are of great interest for the study of mechanical failure. The model describes uni-axial compression of an incompressible hyper-elastic rectangular plate with sliding ending condition. The plate has the small thickness a and infinite length. The leading-order strain only depends on the width $x \in [0, 1]$. Let two continuous functions $W(x)$ and $G(x)$ be the leading-order axial strain and shear strain, respectively. The boundary conditions are $W_x = 0$ and $G = 0$ at two boundary ends $x = 0, 1$. The total potential energy for the reduced 1D model is the integration of the strain energy function $e(x)$ and has the following form (equivalent to Eqn. (4.3) in [27])



(A) index-1 saddle point



(B) post-bucking solution (local minimum)

Fig. 7. (A): W and G plots of the saddle point. (B): W and G plots of the post-bucking solution.

$$E(W,G)=\int_0^1 e(x)\mathrm{d}x=\int_0^1 \mathrm{d}x\Big\{-\hat{\gamma}W-qG+\left(\frac{1}{2}G^2-\frac{3}{4}G^4\right)W+\left(\frac{1}{2}-\frac{5}{4}G^2\right)W^2 +\left(-\frac{1}{2}+\left(\frac{5}{2}+4b\right)G^2\right)W^3+\left(\frac{5}{8}+2b\right)W^4-\left(\frac{3}{4}+4b\right)W^5+\frac{1}{8}G^4 +a^2\left(\frac{1}{6}G_x^2+\frac{1}{6}W_x^2+\frac{16}{9}bWW_x^2-\frac{4}{3}WG_x^2\right)\Big\}, \tag{18}$$

where $\hat{\gamma}$ and q are the axial resultant force and shear resultant force, respectively. The parameter b is a constant depending on the shear modules. We suppose that $q=0$ (no shear) and $\hat{\gamma}<0$ (axial compression). In addition, we set $a=0.05$ and $b=0.25$ as in [27].

There is a trivial stable solution $W(x)\equiv W_0$ and $G\equiv 0$. At $b=0.25$, then the constant W_0 satisfies $W-\frac{3}{2}W^2+\frac{9}{2}W^3-\frac{35}{4}W^4=\hat{\gamma}$, which has two real roots. The smaller root is the stable solution W_0 and its value is negative for $\hat{\gamma}<0$. The trivial solution pair $(W_0,0)$ corresponds to the uniform compression along the axis and no shear induced to bend the plate. When the load $\hat{\gamma}$ is above some critical value, the bucking will occur as new local minima emerge. Apparently, as a new local minimum arises, an index-1 saddle point is created too. We focus on the computation of the index-1 saddle point at $\hat{\gamma}=-8.7651\times 10^{-3}$. This $\hat{\gamma}$ value slightly passes the first critical value $\hat{\gamma}_1$ predicted in [27].

The functions W and G are discretized on a uniform mesh and their first derivatives and the second derivatives are approximated by the fourth order and the second order finite difference scheme, respectively. The initial guess is based on a reduced system with three parameters from the asymptotic result in [27, Eqn. (5.52), $k=1$]. More specifically, we apply the IMA to search an index-1 saddle point in the reduced system and then use it as the initial guess in the space of continuous

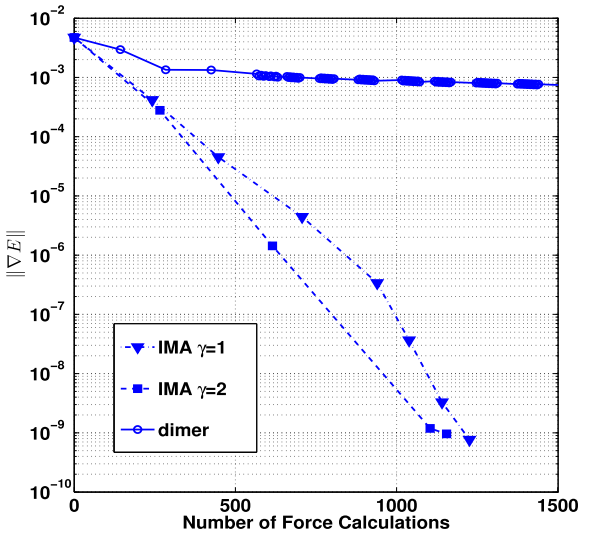


Fig. 8. Hyper-elastic plate problem. Two exponent parameters $\gamma = 1, 2$ are used for comparison. $M_2 = 200$. c_{rot} is observed around 100 in the IMA. The uneven distances between two neighboring circles for the dimer method indicate that sometimes the rotation step in the dimer method needs around 100, sometimes much less.

functions. In this way, we set a good initial guess and the convergence is observed. We set the maximal iteration number $M_2 = 200$. This means the maximum number of force calculations for solving the subproblem $\min_y L(y; x)$ is 400.

The profiles of the saddle point are shown in Fig. 7. The nontrivial local minimizer, which the unstable manifold of this saddle point leads to, is also plotted in this figure. The convergence plot for the efficiency of the IMA is shown in Fig. 8. The error $\|\nabla E\|$ is the L_2 norm of the derivative $(\frac{\delta E}{\delta W}, \frac{\delta E}{\delta G})$. The average number of force calculation for the rotation is around 100 for this example, which is much larger than the cost for one translation step. For the IMA, the data points in filled markers of Fig. 8 again correspond to the results at each cycle. For the dimer method, the data points in open circles correspond to the results after each pair of rotation and translation. Fig. 8 clearly shows that the reduction of the number of cycles significantly reduces the cost coming from the rotations, which is a big portion of the overall computational overhead.

5. Summary

We have provided the iterative minimization algorithm (IMA), based on the iterative minimization formulation (IMF), for finding an index-1 saddle point. This algorithm is specified in Fig. 1 and Algorithm 1. The advantage of our algorithm is the computational efficiency. The higher computational cost for the rotation step, the more significant advantage the IMA has over the traditional methods. The efficiency improvement in the IMA comes not only from our adaptive control for the stopping tolerance for the subproblem at each iteration, but also from our numerical scheme for the gradient of the auxiliary objective function by exploiting its proper approximation, which only requires one force calculation at each CG iteration. Certainly, the underlying theoretic foundation relies on the quadratic convergence rate in the IMF.

There still remain many practical challenges of searching transition states on potential energy surface. For example, in some quantum-chemistry or free energy calculations, one can only calculate the numerical values of gradient forces up to a certain accuracy, thus this source error limits the final accuracy. The more significant challenge is the local convergence for the existing saddle-point methods based on the eigenvector-following methodology, which imposes a difficult task of selecting initial starting points. It would be interesting to explore the possibility of combining the GAD or the IMF with some random sampling techniques.

Acknowledgements

The research of XZ was supported by CityU Start-Up Grant (7200301) and grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 11304314, 109113). WG is supported by the National Natural Science Foundation of China under grants 91330202, Shanghai Science and Technology Development Funds 13dz2260200 and 13511504300, and Special Funds for Major State Basic Research Projects of China (2015CB858560003). JL gratefully acknowledges financial support from China Scholarship Council.

Appendix A. Connection of the IMF with other methods

A.1. Gentlest ascent dynamics

GAD is a dynamical system coupling the position variable x and the unit direction variable v, which locally converges to index-1 saddle points [16]. It can be seen as analogy with the steepest descent dynamics which converges to local minima. The connection of GAD to IMF has already been discussed in [22]. Here we recapitulate this result that GAD corresponds to one of the simplest algorithmic implementation of the IMF that the subproblem is solved by one single steepest descent step.

Denote by $\varphi_t(x)$ the steepest descent flow for $L(y;x)$, which is $\varphi_t(x) = y(t)$ where

$$\dot{y} = -\nabla_y L(y;x), \qquad y(0) = x.$$

Assume that one small step size δt is used to evolve this steepest descent flow for one step and to approximate the true solution of $\min_y L(y;x)$, then $\Phi(x) \approx \varphi_{\delta t}(x)$. As $\delta t \to 0$, the approximation of the discrete time map $\Phi(x)$: $x \to \varphi_{\delta t}(x)$ becomes a flow with velocity

$$\lim_{\delta t \to 0} \frac{\varphi_{\delta t}(x) - x}{\delta t} = \dot{\varphi}_0(x) = -\nabla_y L(y;x)|_{y=x} = -\nabla V(x) + (\alpha+\beta)\,\langle v(x), \nabla V(x)\rangle\, v(x),$$

where (5) is applied. This dynamics is the limiting case of the following GAD [16] when $\tau \to +\infty$,

$$\begin{cases} \dot{x} = -\nabla V(x) + (\alpha+\beta)\langle \nabla V(x), v\rangle v, & \text{(a)} \\ \tau^{-1}\dot{v} = -\nabla^2 V(x) v + \frac{\langle v, \nabla^2 V(x) v\rangle}{\langle v, v\rangle} v. & \text{(b)} \end{cases} \tag{19}$$

For any finite τ, [16] proved the local convergence of $(x(t), v(t))$ to a nearby index-1 saddle point and its min-mode, respectively, if $\alpha + \beta > 1$. The same conclusion holds when $\tau = +\infty$ which is the case that v in (19)(a) is exactly equal to the min-mode $v(x)$. See Theorem 2 below.

Theorem 2. *Assume that the potential function $V \in C^3$. For any point x, let $v(x)$ be the min-mode of the Hessian $H(x) = \nabla^2 V(x)$ at x. If x_* is an index-1 saddle point of V, then x_* is a linear stable equilibrium point of the following dynamical system in $\mathbb{R}^d$, i.e., the limiting case of GAD (19) at $\tau = \infty$,*

$$\dot{x} = -\nabla V(x) + c(v(x) \otimes v(x))\nabla V(x) \tag{20}$$

for any constant $c > 1$.

Proof. Since $\nabla V(x_*) = 0$, x_* then also nullifies the right hand side of (20). To prove the linear stability, it is sufficient to show that the Jacobian of the GAD flow is $-H(x) + c\lambda(x)v(x) \otimes v(x)$ at the saddle x_* where $H(x)$ is the Hessian and $\lambda(x)$ is the minimal eigenvalue, because this matrix $-H(x) + c\lambda(x)v(x) \otimes v(x)$ is negative definite at the index-1 saddle x_* for $c > 1$. Indeed, the Jacobian matrix, the derivative of the right hand side of (20), denoted as $f^{\mathrm{GAD}}(x) := -g(x) + c(v(x) \otimes v(x))g(x)$, is calculated as follows

$$\begin{aligned} f_x^{\mathrm{GAD}}(x) &= -H(x) + c(v(x) \otimes g(x) + \langle v(x), g(x)\rangle) J(x) + cv(x) \otimes (H(x)v(x)) \\ &= -H(x) + c(v(x) \otimes g(x) + \langle v(x), g(x)\rangle) J(x) + c\lambda(x)(v(x) \otimes v(x)) \end{aligned}$$

where $g(x) = \nabla V(x)$ and $J(x)$ is the derivative of $v(x)$. Then the desired result follows from $g(x_*) = 0$. □

A.2. Other eigenvector-following methods

Since the first proposal of dimer method [9], it has been popularly applied to quite a few chemical reaction problems. Following this work, there are several improved versions from different contributors, such as modified dimer by Heyden et al. [10], L-BFGS improvement by Kästner et al. [11] and a line-search improvement by Gould et al. [12]. Any dimer method or its variant consists of two steps, "rotation" step and "translation" step. The rotation step is to find the min-mode and it can be regarded as a classic numerical eigenvector problem. The translation step is to move the position variable using the min-mode calculated from the rotation step. Different translation schemes were proposed and work better than the original dimer method to certain extent. In this subsection, we take the viewpoint of the IMF to examine several major schemes in these methods and reveal their underlying approximations as classic optimization methods. Our discussions below are labeled into four sequential parts for distinctness.

(1). In the iteration $y_i \to y_{i+1}$ for the subproblem $\min_y L(y;x)$, where x is fixed, suppose that initially a normalized search direction, p_0, has been selected. Since the initial $y_0 = x$, the one-step solution for the subproblem is then $y_1 = x + \rho_0 p_0$. If we select the step size ρ_0 as the Newton's step in (12)

$$\rho_0 = -\frac{\langle p_0, (I - (\alpha+\beta)v \otimes v)g\rangle}{\langle p_0, (I - (\alpha+\beta)v \otimes v)Hp_0\rangle} = \frac{\langle p_0, f^\dagger\rangle}{\langle p_0, (I - (\alpha+\beta)v \otimes v)Hp_0\rangle}, \tag{21}$$

where $g = g(x)$, $H = H(x)$. The term $f^\dagger(\cdot) = -(I - (\alpha+\beta)v \otimes v)g(\cdot)$ is the "modified" force [9,10], i.e., the force in GAD. Note that by (9), $f^\dagger(x) = -\nabla L(x; x)$, which is the steepest descent direction of $L(y; x)$ at $y = x$. If this $f^\dagger$ is selected as the initial direction, then $p_0 = f^\dagger(x)/|f^\dagger(x)|$. Consequently, the resulting iteration $y_0 = x \to y_1 = y_0 + \rho_0 p_0$ gives the following mapping $\hat{\Phi}_1$

$$\begin{aligned}\hat{\Phi}_1(x) &:= x + \rho_0 p_0 = x + \frac{|f^\dagger(x)|^2}{\left\langle f^\dagger(x), \big(H(x) - (\alpha+\beta)\lambda(x)(v(x)\otimes v(x))\big) f^\dagger(x)\right\rangle} f^\dagger(x) \\ &= x + \frac{|f^\dagger(x)|^2}{\left\langle g(x), \big(H(x) - c\lambda(x)v(x)\otimes v(x)\big) g(x)\right\rangle} f^\dagger(x),\end{aligned}$$

where the constant $c = (\alpha+\beta)((\alpha+\beta)^2 - 3(\alpha+\beta) + 3)$. Note that $c > 1$ since $\alpha + \beta > 1$, and $c = 2$ when $\alpha + \beta = 2$. It is not difficult to verify that $\hat{\Phi}_1(x_*) = \lim_{x\to x_*} \hat{\Phi}_1(x) = x_*$ for index-1 saddle x_*. However, we do not generally have the vanishing Jacobian $D\hat{\Phi}_1(x_*)$. For instance, one can use the special 1D example $V(x) = (x^2-1)^2/4$ to check that $D\hat{\Phi}_1(x_*)$ ($x_* = 0$ for this example) monotonically decreases from 1 to -1 when $\alpha + \beta$ changes from 1 to 2.45. This result suggests that using the above one-step solution y_1 in solving the subproblem $\min_y L(y; x)$ only gives the linear convergence rate.

We point out that the above scheme $\hat{\Phi}_1$ corresponds to the dimer method reviewed in [10]. Choose α, β such that $\alpha + \beta = 2$. To avoid the use of the whole Hessian matrix $H(x)$, the term involving the multiplication Hp for the step size given in equation (12) is numerically calculated by the finite difference scheme with spacing h (length of the dimer) along the direction p; for example, the forward finite difference scheme gives

$$(I - 2v\otimes v)Hp \approx (I - 2v\otimes v)\frac{g(x+hp) - g(x)}{h} = -\frac{f^\dagger(x+hp) - f^\dagger(x)}{h}.$$

Therefore,

$$y_1 = y_0 + \rho_0 p_0 \approx x - \frac{h\langle p_0, f^\dagger(x)\rangle}{\langle p_0, (f^\dagger(x+hp_0) - f^\dagger(x))\rangle} p_0 \tag{22}$$

which is exactly Eqn. (21) for the dimer method in [10] if the normalized search direction is set as the steepest descent direction $p_0 = f^\dagger(x)/|f^\dagger(x)|$ and the value of $f^\dagger(x)$ in the numerator of (22) is replaced by the average of $f^\dagger(x)$ and $f^\dagger(x+hp)$.

We remark that the improved dimer method in [10] actually involves a line search by linear inter/extrapolation from a predefined step size, which is a further approximated version of (21): replacing p_0 in the denominator by the min-mode v (which implies $\rho_0 = 0.5\langle p_0, f^\dagger\rangle/|\lambda|$).

(2). Another Newton-type scheme in 2D was introduced in [15] for the activation–relaxation method. The interpretation of this scheme via the IMF is as follows. Now consider the quadratic approximation of $L(y; x)$ at $y_0 = x$ in the 2D space spanned by two orthogonal directions, $v(x)$ and $g^\perp(x) := (I - v(x)\otimes v(x))g(x)$. Solving the following 2D quadratic approximation of $L(y; x)$ with $\alpha + \beta = 2$

$$\min_{a,b\in\mathbb{R}} \left\{ L(x;x) + \nabla L(x;x)\Delta x + \frac{1}{2}\left\langle \Delta x, \nabla^2 L(x;x)\Delta x\right\rangle : \Delta x = av(x) + bg^\perp(x)\right\},$$

we have

$$(a, b) = \left(-\frac{\langle v(x), g(x)\rangle}{\lambda(x)}, -\frac{\langle g^\perp(x), g^\perp(x)\rangle}{\langle g^\perp(x), H(x)g^\perp(x)\rangle}\right). \tag{23}$$

The solution b is non-positive since $\langle g^\perp(x), H(x)g^\perp(x)\rangle = \langle g(x), (H(x) - \lambda(x)v(x)\otimes v(x))g(x)\rangle$ is positive near the index-1 saddle.

Notice that [15] used the following scheme for the displacement

$$\Delta x = -\frac{\langle v, g\rangle}{\min(\lambda, -\bar{\lambda})} v - \mu(I - v\otimes v)g,$$

where $\bar{\lambda} > 0$ is used to avoid instability when λ crosses zero value and the step size $\mu > 0$ is practically set as some constant. The formula of (23) based on the IMF recovers their result and furthermore gives the suggestion for the coefficient at the other direction $g^\perp$.

The formula (23) defines a mapping

$$\widehat{\Phi}_2(x) = x - \frac{\langle v(x), g(x)\rangle}{\lambda(x)} v(x) - \frac{\langle g^\perp(x), g^\perp(x)\rangle}{\langle g^\perp(x), H(x)g^\perp(x)\rangle} g^\perp(x). \tag{24}$$

Again, we have $\widehat{\Phi}_2(x_*) = x_*$, but the Jacobian $D\widehat{\Phi}_2(x_*)$ does not vanish. We tested $\widehat{\Phi}_2$ for a 2D example $V(x, y) = 2xy(x^2 + y^2 - 1)$ for the saddle point $(0, 1)$ and verified that such a scheme also only has the linear iteration rate.

Remark 1. The convergence result proved in [15] for the improved algorithm proposed there is that "One can then prove (see the Appendix A) that algorithm converges to the saddle point quadratically in the principal direction of negative curvature and linearly in the perpendicular directions" (page 114711-2, [15]). In the Appendix A of this article, the quadratic convergence is about the projected sequence "$z_k = (x_k - x_*) \cdot v_1^*$" in $\mathbb{R}^1$, not about the true error "$e_k = x_k - x_*$". Our result here is about the quadratic convergence rate of the true error e_k in $\mathbb{R}^n$.

(3). The work in [11] uses L-BFGS to perform the translation step. The corresponding Δx is thus not along the direction $f^\dagger$, but a Newton type in the form of $T^{-1} f^\dagger$ for some matrix T. These are actually quasi-Newton solvers for the quadratic approximation of L. Thus, the L-BFGS [11] or Broyden method [28] can be applied to approximate the inverse of the Hessian $\nabla^2 L = (H - 2\lambda v \otimes v)$ at $x^{(k)}$ using information at $x^{(k-1)}$, $x^{(k-2)}$, etc.

(4). Lastly, we look at a line-search variant of the dimer method recently proposed in [12]. In this work, an auxiliary function is also proposed. In the limit that the dimer length h goes to zero, their auxiliary function F, in [12, Eqn. (12)], is (written in style of this paper)

$$F(y;x) = V(y) - 2\langle v(x), \nabla V(x)\rangle \langle v(x), y - x\rangle - \lambda(x)\langle v(x), y - x\rangle^2 . \tag{25}$$

This auxiliary function F is an approximation of $L_2(y;x)$ by applying a quadratic expansion only for the second term in $L_2(y;x)$ at $y = x$ and keeping the first term $V(y)$ unchanged. In their work, a line search scheme by binary search with certain stopping criteria to select the step size ρ_0 was proposed for this auxiliary function. It is noted that their line search requires to calculate the energy V, while many other methods only need the gradient force ∇V. The line search requiring V is not applicable for quantum-chemistry first principle calculation. We point out that their algorithm still takes the only first step solution y_1 as the new $x^{(k+1)}$ as the previous variant of the dimer method did. Thus, it is indicated that this approach still only has linear convergence efficiency, which was confirmed by Theorem 7 and the numerical results in [12].

Motivated by the function (25), we can also consider the following partially quadratic approximation of the original auxiliary function (3),

$$\begin{aligned}\tilde{L}(y;x) &= (1-\alpha)V(y) \\ &+ \alpha\left[V(x) + \langle (I - v(x)\otimes v(x))g(x), y - x\rangle + \frac{1}{2}\langle (H(x) - \lambda(x)v(x)\otimes v(x))(y-x), y - x\rangle\right] \\ &- \beta\left[V(x) + \langle v(x), g(x)\rangle \langle v(x), y - x\rangle + \frac{\lambda(x)}{2}\langle v(x), y - x\rangle^2\right].\end{aligned} \tag{26}$$

The gradient of $\tilde{L}(y;x)$ is

$$\begin{aligned}\nabla\tilde{L}(y;x) &= (1-\alpha)g(y) + \alpha g(x) - (\alpha+\beta)\langle v(x), g(x)\rangle v(x) \\ &\quad + \alpha(H(x) - \lambda(x)v(x)\otimes v(x))(y-x) - \beta\lambda(x)\langle v(x), y - x\rangle v(x),\end{aligned} \tag{27}$$

which is the same as the approximation expression on the right hand side of equation (7).

The conclusions in Theorem 1 associated with L are still true for $\tilde{L}$. Refer to Theorem 3 below.

Theorem 3. *Assume that the potential function $V \in C^4$. Suppose that x_* is an index-1 saddle point of the function $V(x)$, and the auxiliary function $\tilde{L}$ is defined by (26) together $\alpha + \beta > 1$, then the results (1) to (5) in Theorem 1 hold for $\tilde{L}$ and the corresponding mapping $\tilde{\Phi}$.*

Proof. The proof is similar to the proof of Theorem 1 in [22]. We only show here the proof of the key conclusion *(4)*. Others are straightforward by following [22]. Note that $\tilde{\Phi}(x)$ satisfies the extremal condition $\nabla_y \tilde{L}(y;x) = 0$ at $y = \tilde{\Phi}(x)$. Thus,

$$\begin{aligned}0 = \nabla_y \tilde{L}(y;x)\big|_{y=\tilde{\Phi}(x)} &= (1-\alpha)g(\tilde{\Phi}(x)) + \alpha\Big[(I - v\otimes v)g + (H - \lambda v\otimes v)(\tilde{\Phi}(x) - x)\Big] \\ &\quad - \beta\Big[(v\otimes v)g + \lambda(v\otimes v)(\tilde{\Phi}(x) - x)\Big],\end{aligned} \tag{28}$$

where the functions without explicit arguments means the evaluation at x. We then evaluate the derivative with respect to x of this equality and use $g(x^*) = 0$ as well as $\tilde{\Phi}(x_*) = x_*$. So, at $x = x_*$, the following holds

$$\begin{aligned}&(1-\alpha)H(\tilde{\Phi})\tilde{\Phi}_x + \alpha\Big[(I - v\otimes v)H + (H - \lambda v\otimes v)(\tilde{\Phi}_x - I)\Big] \\ &\quad - \beta\Big[(v\otimes v)H + \lambda(v\otimes v)(\tilde{\Phi}_x - I)\Big] = 0.\end{aligned}$$

It follows then that $\Big(H(x_*) - (\alpha+\beta)\lambda(x_*)(v(x_*)\otimes v(x_*))\Big)\tilde{\Phi}_x(x_*) = 0$. Therefore $\tilde{\Phi}_x(x_*) = 0$ for $\alpha + \beta > 1$. □

A.3. String method

We digress a little from the eigenvector-following methods in this paragraph and remark the interesting connection between the idea behind the string method and the optimization viewpoint of our work of IMF. The string method solves a dynamical equation for minimum energy path (MEP) $\phi(s), s \in [0,1]$, such that the steady state equation $(\nabla V(\phi))^{\perp} = 0$ is satisfied. Here the projection $^{\perp}$ is onto the hyperplane $\{y : \langle y - \phi(s), \tau(s)\rangle = 0\}$ with $\tau(s) = \dot{\phi}(s)/|\dot{\phi}(s)|$ being the unit tangent of ϕ at $\phi(s)$. The steady state equation is equivalent to the self-consistence condition that $\phi(s)$ is a local solution of $\min_y\{V(y) : \langle y - \phi(s), \tau(s)\rangle = 0\}$, which is a constraint minimization problem defined by the point $\phi(s)$ itself and the tangent direction $\tau(s)$. This fact has been exploited in [29] to construct an optimization-based string method. If we make analogy of the min-mode v in the eigenvector-following methodology with the path tangent τ in the string method (they do coincide at index-1 saddle point), then our minimization for $\min_y L(y;x)$ in (3) restricted on the hyperplane $\langle y - x, v\rangle = 0$, is almost the same as the constraint minimization problem in the string method, by noting that $L(y;x) = V(y) - \beta V(x)$ on the hyperplane $\langle y - x, v\rangle = 0$. If one solves the constraint minimization problem $\min_y L(y;x)$ subject to the hyperplane constraint $\langle y - x, v\rangle = 0$ by one step steepest descent dynamics in the same spirit of the GAD, then the following flow arises: $\dot{x} = -(I - v \otimes v)(-\nabla L(x,x)) = -(I - v(x) \otimes v(x))g(x)$. The limiting set of this dynamics at $t \to \infty$ is a curve containing index-1 saddle point and local minimum (but it is not the MEP ϕ since $v(x)$ is defined by the Hessian, not the path tangent). On this curve it is satisfied that $g(x) \parallel v(x)$, i.e., the gradient direction parallels to the min-mode. This condition means that $\lambda(x)g(x) = H(x)g(x)$, which is equivalent to $\lambda(x)\nabla V(x) = \nabla|\nabla V(x)|^2$. This means that $|\nabla V|^2$ and V share the same gradient orbit (the time direction depends on the sign of $\lambda(x)$).

References

[1] H. Jònsson, G. Mills, K.W. Jacobsen, Nudged elastic band method for finding minimum energy paths of transitions, in: B.J. Berne, G. Ciccotti, D.F. Coker (Eds.), Classical and Quantum Dynamics in Condensed Phase Simulations, LERICI, Villa Marigola, in: Proceedings of the International School of Physics, World Scientific, New Jersey, 1998, p. 385.
[2] W. E, W. Ren, E. Vanden-Eijnden, String method for the study of rare events, Phys. Rev. B 66 (2002) 052301.
[3] W. E, W. Ren, E. Vanden-Eijnden, Simplified and improved string method for computing the minimum energy paths in barrier-crossing events, J. Chem. Phys. 126 (2007) 164103.
[4] M. Cameron, R. Kohn, E. Vanden-Eijnden, The string method as a dynamical system, J. Nonlinear Sci. 21 (2) (2011) 193–230.
[5] C.J. Cerjan, W.H. Miller, On finding transition states, J. Chem. Phys. 75 (6) (1981) 2800–2806.
[6] D.J. Wales, Finding saddle points for clusters, J. Chem. Phys. 91 (1989) 7002.
[7] J. Baker, An algorithm for the location of transition states, J. Comput. Chem. 7 (4) (1986) 385–395.
[8] G.M. Crippen, H.A. Scheraga, Minimization of polypeptide energy: XI. the method of gentlest ascent, Arch. Biochem. Biophys. 144 (2) (1971) 462–466.
[9] G. Henkelman, H. Jónsson, A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives, J. Chem. Phys. 111 (15) (1999) 7010–7022.
[10] A. Heyden, A.T. Bell, F.J. Keil, Efficient methods for finding transition states in chemical reactions: comparison of improved dimer method and partitioned rational function optimization method, J. Chem. Phys. 123 (2005) 224101.
[11] J. Kästner, P. Sherwood, Superlinearly converging dimer method for transition state search, J. Chem. Phys. 128 (2008) 014106.
[12] N. Gould, C. Ortner, D. Packwood, An efficient dimer method with preconditioning and linesearch, arXiv e-prints, arXiv:1407.2817, 2014.
[13] N. Mousseau, G. Barkema, Traveling through potential energy surfaces of disordered materials: the activation–relaxation technique, Phys. Rev. E 57 (1998) 2419.
[14] R. Malek, N. Mousseau, Dynamics of Lennard–Jones clusters: a characterization of the activation–relaxation technique, Phys. Rev. E 22 (6) (2000) 7723.
[15] E. Cancès, F. Legoll, M.-C. Marinica, K. Minoukadeh, F. Willaime, Some improvements of the activation–relaxation technique method for finding transition pathways on potential energy surfaces, J. Chem. Phys. 130 (2009) 114711.
[16] W. E, X. Zhou, The gentlest ascent dynamics, Nonlinearity 24 (6) (2011) 1831.
[17] A. Samanta, W. E, Atomistic simulations of rare events using gentlest ascent dynamics, J. Chem. Phys. 136 (2012) 124104.
[18] A. Samanta, M. Chen, T.-Q. Yu, M. Tuckerman, W. E, Sampling saddle points on a free energy surface, J. Chem. Phys. 140 (16) (2014).
[19] J. Zhang, Q. Du, Shrinking dimer dynamics and its applications to saddle point search, SIAM J. Numer. Anal. 50 (2012) 1899–1921.
[20] W. Ren, E. Vanden-Eijnden, A climbing string method for saddle point search, J. Chem. Phys. 138 (2013) 134105.
[21] W. Ren, Wetting transition on patterned surfaces: transition states and energy barriers, Langmuir 30 (10) (2014) 2879–2885.
[22] W. Gao, J. Leng, X. Zhou, An iterative minimization formulation for saddle point search, SIAM J. Numer. Anal. 53 (4) (2015) 1786–1805.
[23] J. Leng, W. Gao, C. Shang, Z.-P. Liu, Efficient softest mode finding in transition states calculations, J. Chem. Phys. 138 (9) (2013) 094110.
[24] A. Samanta, M.E. Tuckerman, T.-Q. Yu, W. E, Microscopic mechanisms of equilibrium melting of a solid, Science 346 (6210) (2014) 729–732.
[25] G. Henkelman, G. Jóhannesson, H. Jónsson, Methods for finding saddle points and minimum energy paths, in: S.D. Schwartz (Ed.), Progress on Theoretical Chemistry and Physics, Kluwer Academic Publishers, 2000, pp. 269–300.
[26] http://theory.cm.utexas.edu/henkelman/research/saddle/configurations.zip.
[27] H.-H. Dai, F.-F. Wang, Analytical solutions for the post-buckling states of an incompressible hyperelastic layer, Anal. Appl. 10 (01) (2012) 21–46.
[28] C. Shang, Z.-P. Liu, Constrained Broyden minimization combined with the dimer method for locating transition state of complex reactions, J. Chem. Theory Comput. 6 (4) (2010) 1136–1144.
[29] A. Samanta, W. E, Optimization-based string method for finding minimum energy path, Commun. Comput. Phys. 14 (2) (2013) 265–275.