Accelerated saddle point refinement through full exploitation of partial Hessian diagonalization

Eric D. Hermes, Khachik Sargsyan, Habib N. Najm, and Judit Zádor*

Combustion Research Facility, Sandia National Laboratories, Livermore, CA 94551-0969 USA

E-mail: jzador@sandia.gov

Abstract

Identification and refinement of first order saddle point (FOSP) structures on the potential energy surface (PES) of chemical systems is a computational bottleneck in the characterization of reaction pathways. Leading FOSP refinement strategies require calculation of the full Hessian matrix, which is not feasible for larger systems such as those encountered in heterogeneous catalysis. For these systems, the standard approach to FOSP refinement involves iterative diagonalization of the Hessian, but this comes at the cost of longer refinement trajectories due to the lack of accurate curvature information. We present a method for incorporating information obtained by an iterative diagonalization algorithm into the construction of an approximate Hessian matrix that accelerates FOSP refinement. We measure the performance of our method with two established FOSP refinement benchmarks and find a 50 % reduction on average in the number of gradient evaluations required to converge to a FOSP for one benchmark, and a 25 % reduction on average for the second benchmark.

1 Introduction

There is a growing interest in exploring the properties of complex reaction systems from first principles. New tools that automate the tedious and error-prone task of enumerating all plausible reaction pathways has enabled the development of increasingly complex reaction network models.[1–18] Additionally, the availability of petascale and upcoming exascale computational resources makes it possible to perform the many computationally intensive first principles calculations required by these complex models. These developments have inspired the need for more efficient and reliable software tools, as too many unconverged or failed calculations render these automated frameworks impractical. Unfortunately, many existing software tools are not designed explicitly with automation in mind, and thus occasionally exhibit inconsistent or unreliable behavior. In addition, it is common for software packages to bundle together method implementations for complementary but distinct tasks, such as electronic structure theory and geometry optimization. As a result, it may not be possible to use a particular combination of methods that is best suited to the task at hand unless those methods are both implemented in the same software package. It is therefore beneficial to develop new software that both is amenable to exascale automation and which improves flexibility by decoupling method implementations.

In this work, we focus on the task of saddle point refinement, and propose a new method for this task that can be effectively deployed in an automated computational framework. Locating first order saddle points (FOSP) on the potential energy surface (PES) of chemical systems is a computational bottleneck in the characterization of reaction pathways. A FOSP is a stationary point on the PES where the Hessian matrix $\mathbf{H}$ has precisely one negative eigenvalue. There are several established approaches for the task of locating and refining FOSP geometries, the applicability of which depends on what is already known about the PES. When both the reactant and product geometry are known, double-ended methods such as nudged elastic band (NEB) or quadratic synchronous transport (QST) can be used to refine the minimum energy path (MEP)

connecting the two minimum wells.[19–27] The maximum of the MEP is necessarily a FOSP. If only a single minimum energy geometry is known, certain single-ended methods such as the growing string method (GSM) or the activation-relaxation technique (ART) can be used to drive the geometry in the direction of a desired reaction coordinate.[28–33] With both single- and double-ended methods, it is usually more efficient to switch to a local FOSP refinement method once a sufficiently accurate approximate structure has been found. There are also a growing number of software packages for the determination of approximate FOSP structures using heuristics or machine learning, such as KinBot and AutoTST.[9,10,34] Regardless of how they are obtained, approximate FOSP structures can be efficiently refined using techniques adapted from geometry minimization methods.

Many molecular dynamics and electronic structure theory software packages implement their own FOSP refinement techniques, but the quality of these implementations can vary greatly between packages. This coupling poses a problem to scientists who wish to choose the best software package for solving their particular task, as the software package with the best performing FOSP refinement method may not implement the desired PES. This has also led to a splintering of FOSP refinement strategies between codes designed for small gas-phase molecules and codes designed for larger condensed-phase systems. Many advances that have been developed for molecular FOSP refinement have not been adapted for use in condensed-phase systems and vice versa. The method described in this work begins to close the gap between the methods that are used for these two types of chemical systems.

A common feature of all FOSP refinement strategies is the determination of an ascent direction. In contrast to (local) minimization, for which displacements should always descend the PES, FOSP refinement involves ascending the PES in precisely one direction. The ascent direction is chosen to locally approximate the reaction coordinate corresponding to the desired FOSP. The reaction coordinate can be identified by the leftmost eigenvector of $\mathbf{H}$ at the FOSP. The leftmost eigenvector of $\mathbf{H}$ is then a natural choice for the ascent direction if no other information about the reaction coordinate is

available.

If $\mathbf{H}$ is known, the leftmost eigenvector can be determined at an insignificant computational expense compared to the cost of a single PES evaluation. However, calculating $\mathbf{H}$ requires $3N$ times the cost of a single energy or gradient evaluation for a system of N atoms. Consequently, evaluating $\mathbf{H}$ at every step is prohibitively computationally expensive for all but the smallest systems. Instead, many FOSP refinement strategies evaluate $\mathbf{H}$ sparingly, commonly only for the initial structure. Subsequent geometry refinement steps instead use an approximation $\mathbf{B}$ that is constructed to remain close to the true $\mathbf{H}$.

For sufficiently large systems, evaluating $\mathbf{H}$ even a single time may require more computational resources than the total cost of all subsequent PES evaluations. This problem has led to the development of strategies for identifying the leftmost eigenvector of $\mathbf{H}$ without needing it to be evaluated in full. This can be accomplished through the use of an iterative diagonalization algorithm, such as Lanczos, which does not require any direct knowledge of $\mathbf{H}$. Instead, iterative diagonalization algorithms only require the ability to evaluate Hessian-vector products, which can be approximated by applying finite difference to the gradient vector. This makes it possible to accurately identify the leftmost eigenvector of $\mathbf{H}$ at a significantly lower computational expense.

The approximate Hessian $\mathbf{B}$ is used to determine geometry refinement steps that converge to a FOSP. The accuracy of $\mathbf{B}$ directly affects the number of steps that are required to reach convergence. The change in the gradient from one iteration to the next provides approximate curvature information that can be used to improve the accuracy of $\mathbf{B}$. However, if $\mathbf{B}$ is initially a poor quality approximation to $\mathbf{H}$, then the first few steps may lead away from the desired FOSP. By the time $\mathbf{B}$ has become sufficiently accurate, the geometry may have wandered far from the initially close FOSP.

It is for this reason that $\mathbf{B}$ must be sufficiently accurate at the beginning of FOSP refinement. If it is feasible to evaluate $\mathbf{H}$ for the initial structure, then $\mathbf{B}$ can be made initially exact. This is not possible if an iterative diagonalization algorithm is used, as the full Hessian is never evaluated. However, it is possible to construct $\mathbf{B}$ to be exact

in the subspace searched by the diagonalization algorithm, though to the best of the authors' knowledge, no established method does this. As we will show, constructing $\mathbf{B}$ in this way results in a substantial improvement in performance compared to methods that use an iterative diagonalization algorithm but construct $\mathbf{B}$ in a different way.

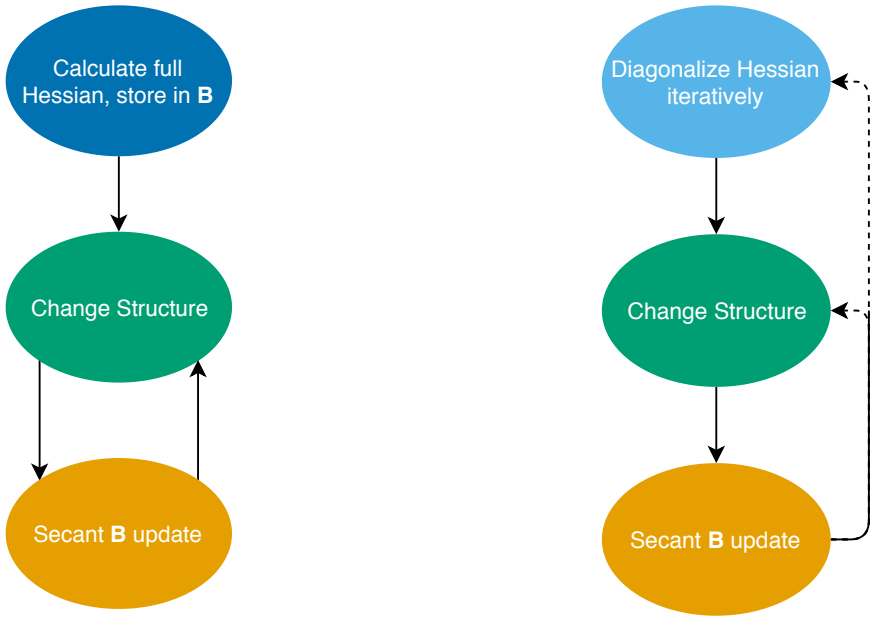
In section 2, we discuss several established computational methods with relevance to FOSP refinement. Our new FOSP refinement method is described in section 3. We analyze the performance of this new method in section 4.

2 Theoretical Background

In this section, we discuss established FOSP refinement methodologies and methodologies for related problems that we use in our FOSP refinement method, which is described in section 3. We use the term "FOSP refinement" instead of "FOSP optimization" to avoid confusion with the saddle point problem that affects gradient descent (ascent) minimization (maximization) methods. This section also provides context to better understand where and why we deviate from established FOSP refinement methodologies. In order to better motivate the method we have developed, we describe these methods and problem spaces in some detail.

Figure 1 compares two established classes of FOSP refinement procedures. Figure 1a illustrates a method that relies on exact calculation of $\mathbf{H}$ for an initial structure (dark blue oval). The approximate Hessian $\mathbf{B}$ is initially exact. After each geometry step (green oval), a secant update is applied to $\mathbf{B}$ (orange oval). In this way, $\mathbf{B}$ remains a reasonably accurate approximation to $\mathbf{H}$ over the course of FOSP refinement. We describe existing approaches for determining geometry steps for FOSP refinement in section 2.2. Additionally, we describe existing secant update methods in section 2.3.

Figure 1b illustrates a method that iteratively diagonalizes $\mathbf{H}$ to determine its left-most eigenvector $\mathbf{v}^{(1)}$ (light blue oval). $\mathbf{B}$ is initialized independently of the iterative diagonalization algorithm, typically as a scaled identity matrix. Secant updates are applied to $\mathbf{B}$ in this approach as well, though $\mathbf{B}$ generally remains a relatively poor



(a) (b)

Figure 1: An illustration of leading methods for FOSP refinement. (a) A method which fully calculates $\mathbf{H}$ for the initial structure, then relies on secant updates to maintain an approximation $\mathbf{B}$. (b) A method which periodically iteratively diagonalizes $\mathbf{H}$ to find its leftmost eigenvector $\mathbf{v}^{(1)}$ and independently maintains an approximation $\mathbf{B}$ for geometry refinement steps.

approximation to $\mathbf{H}$. In this approach, $\mathbf{H}$ must be periodically iteratively diagonalized in order to maintain an accurate estimate of $\mathbf{v}^{(1)}$. We describe existing approaches for diagonalizing the Hessian matrix iteratively in section 2.1. While the method developed in this work takes several inspirations from methods that fully evaluate $\mathbf{H}$, it is most similar to methods that use an iterative diagonalization routine. We describe our method and how it improves upon existing methods in section 3.

In the remainder of this section, and in section 3 below, we shall use the following definitions. The molecular structure at iteration k of FOSP refinement is represented by the coordinate vector $\mathbf{q}_k$. In this context, "iteration" refers to a single step of the FOSP refinement algorithm, not including gradient evaluations invoked by the iterative diagonalization algorithm. In this work, we assume a Cartesian representation, though much of the described work is independent of this choice. The energy, its gradient, and its true Hessian are respectively $\epsilon(\mathbf{q})$, $\mathbf{g}(\mathbf{q})$, and $\mathbf{H}(\mathbf{q})$. The quantities ϵ_k, $\mathbf{g}_k$, and $\mathbf{H}_k$ refer to these same quantities evaluated at point $\mathbf{q}_k$. The approximate Hessian at iteration k of FOSP refinement is represented by $\mathbf{B}_k$.

2.1 Iterative diagonalization of the Hessian

Several existing FOSP refinement methods rely on iterative diagonalization of $\mathbf{H}(\mathbf{q})$.[35–39] Such methods require only the ability to evaluate Hessian-vector products $\mathbf{Hs}$ (the argument $\mathbf{q}$ to $\mathbf{H}$ has been omitted for clarity), where $\mathbf{s}$ is an arbitrary displacement vector. These Hessian-vector products can be approximated by finite difference,

$$\mathbf{Hs} \approx \frac{\mathbf{g}(\mathbf{q} + \eta\mathbf{s}) - \mathbf{g}(\mathbf{q})}{\eta}, \tag{1}$$

where η is a small real number controlling the magnitude of the finite displacement step size. In the following discussion, the argument $\mathbf{q}$ to $\mathbf{H}$ and $\mathbf{g}$ will be omitted for brevity.

Many iterative diagonalization algorithms were originally developed to determine a few eigenvalues and eigenvectors of very large and sparse matrices. These iterative

diagonalization algorithms typically need to balance the number of iterations required to reach convergence against memory storage requirements and the number of linear algebra operations per iteration. However, the matrix $\mathbf{H}$ is neither prohibitively large nor generally sparse. In our application, memory storage requirements are minuscule, and linear algebra operations (excluding those which involve $\mathbf{H}$) are essentially free. We will focus our discussion on methods that use the Rayleigh-Ritz procedure[40] without restart or deflation. Methods such as LOBPCG[41] which employ restarting to reduce memory requirements are not considered.

The Rayleigh-Ritz procedure provides a general template for the refinement of approximate eigenvalues and eigenvectors of a matrix. In the Rayleigh-Ritz procedure, a set of orthonormal displacement vectors form a matrix $\mathbf{S}_m$ that is extended with a new vector at every iteration. The subscript m indicates the diagonalization iteration number and consequently the number of columns in $\mathbf{S}_m$. The product $\mathbf{Y}_m = \mathbf{H}\mathbf{S}_m$ is evaluated column-by-column using eq 1. The matrices $\mathbf{S}_m$ and $\mathbf{Y}_m$ are used to construct the projected Hessian $\mathbf{A}_m = \mathbf{S}_m^T\mathbf{Y}_m = \mathbf{Y}_m^T\mathbf{S}_m = \mathbf{S}_m^T\mathbf{H}\mathbf{S}_m$. The eigenvalues (Ritz values) $\theta_m^{(j)}$ and eigenvectors (primitive Ritz vectors) $\mathbf{c}_m^{(j)}$ of $\mathbf{A}_m$ are determined with a dense diagonalization algorithm. The Ritz values $\theta_m^{(j)}$ and Ritz vectors $\mathbf{x}_m^{(j)} = \mathbf{S}_m\mathbf{c}_m^{(j)}$ approximate the true eigenvalues $\lambda^{(i)}$ and eigenvectors $\mathbf{v}^{(i)}$ of $\mathbf{H}$. Each Ritz pair has a corresponding residual vector $\mathbf{r}_m^{(j)} = \mathbf{Y}_m\mathbf{c}_m^{(j)} - \theta_m^{(j)}\mathbf{x}_m^{(i)}$ which contains the components of $\mathbf{H}\mathbf{x}_m^{(j)}$ that lie outside of the subspace spanned by $\mathbf{S}_m$. These quantities are used to determine a new vector $\mathbf{t}_m$ that will be used to extend $\mathbf{S}_m$. This procedure is repeated until $\mathbf{r}_m^{(j)}$ is deemed sufficiently small for all desired eigenpairs. Diagonalization methods based on the Rayleigh-Ritz procedure differ in how the expansion vector $\mathbf{t}_m$ is chosen.

The Lanczos method is perhaps the simplest iterative diagonalization algorithm that is suitable for finding the leftmost eigenvector of $\mathbf{H}$. Lanczos chooses to expand $\mathbf{S}_m$ with a residual vector $\mathbf{t}_m = \mathbf{r}_m^{(j)}$. All residual vectors at a given iteration of Lanczos are identical up to a constant multiplicative factor, so it is not possible to target which Ritz values are to be improved. Lanczos is known to converge to the largest magnitude

eigenvalues first.

The leftmost eigenvalue of $\mathbf{H}$ is typically smaller in magnitude than the rightmost eigenvalue in the context of chemical PESs. In this case, Lanczos will converge slowly to the leftmost eigenvalue. To solve this problem, the shift-and-invert class of diagonalization algorithms can be used instead. These algorithms are designed to improve the Ritz values closest to a target value.[42] A specific example is Rayleigh quotient iteration (RQI), in which the target value is chosen to be a current Ritz value.[42] RQI determines $\mathbf{t}_m$ by solving the equation

$$\left(\mathbf{H} - \theta_m^{(j)}\mathbf{I}\right)\mathbf{t}_m = -\mathbf{r}_m^{(j)}, \tag{2}$$

where $\mathbf{I}$ is the identity matrix. If $\theta_m^{(j)}$ is sufficiently close to an eigenvalue of $\mathbf{H}$, eq 2 will have no solution for $\mathbf{t}_m$. Unfortunately, this tends to be the case near convergence, and as a result, RQI has difficulty achieving tight convergence. Even if this were not an issue, solving eq 2 exactly requires the use of an iterative linear solver, as $\mathbf{H}$ is not directly known. To use an iterative linear solver would require several additional Hessian-vector products for each iteration of the Rayleigh-Ritz procedure, which would be costly and inefficient.

These additional Hessian-vector products can be avoided if only a single iteration of a preconditioned solver is used instead. Approximately solving eq 2 in this way is equivalent to solving the modified equation

$$\left(\mathbf{B} - \theta_m^{(j)}\mathbf{I}\right)\mathbf{t}_m = -\mathbf{r}_m^{(j)}, \tag{3}$$

where $\mathbf{B} \approx \mathbf{H}$ is the preconditioner. This is known as the generalized Davidson (GD) method.[43,44] Davidson's original method chooses $\mathbf{B}$ to be a diagonal matrix containing the diagonal elements of $\mathbf{H}$.[45] This choice makes sense in the context of electronic structure theory, as the Hamiltonian matrix is diagonally dominant in an atom-centered basis. In contrast, the Hessian matrix is typically not diagonally dominant, and regardless, it is not in general possible to extract only the diagonal elements of $\mathbf{H}$ without

evaluating it in full. GD places no restrictions on how $\mathbf{B}$ is to be constructed. Assuming a reasonably accurate $\mathbf{B}$ is available, this approach can be highly effective. GD has been used previously for the diagonalization of Hessian matrices.[46] Unfortunately, if $\mathbf{B}$ is sufficiently close to $\mathbf{H}$, then GD can experience the same difficulties as RQI near convergence.

Several approaches have been developed to alleviate this particular problem, perhaps the most successful of which is the Jacobi-Davidson (JD) method.[44,47–49] JD determines $\mathbf{t}_m$ by solving the equation

$$\left(\mathbf{I} - \mathbf{x}_m^{(j)}\mathbf{x}_m^{(j)T}\right)\left(\mathbf{H} - \theta_m^{(j)}\mathbf{I}\right)\left(\mathbf{I} - \mathbf{x}_m^{(j)}\mathbf{x}_m^{(j)T}\right)\mathbf{t}_m = -\mathbf{r}_m^{(j)}. \tag{4}$$

Eq 4 is equivalent to eq 2 with the added constraint that $\mathbf{t}_m \perp \mathbf{x}_m^{(j)}$. Provided that $\theta_m^{(j)}$ approximates a simple (non-degenerate) eigenvalue of $\mathbf{H}$, eq 4 can be solved even very near convergence.

However, as with RQI, solving eq 4 would require the use of an iterative linear solver. In the same vein as GD, eq 4 can be solved approximately with a single iteration of a preconditioned solver. This is equivalent to solving the modified equation

$$\left(\mathbf{I} - \mathbf{x}_m^{(j)}\mathbf{x}_m^{(j)T}\right)\left(\mathbf{B} - \theta_m^{(j)}\mathbf{I}\right)\left(\mathbf{I} - \mathbf{x}_m^{(j)}\mathbf{x}_m^{(j)T}\right)\mathbf{t}_m = -\mathbf{r}_m^{(j)}. \tag{5}$$

We refer to this method as JD0, though it is occasionally referred to as Olsen's method.[47,50,51] This is the approach employed by our method.

2.2 Geometry refinement

FOSP refinement methods are largely adaptations of methods developed for geometry minimization. For example, consider the Newton-Raphson (NR) method, in which a displacement direction is selected at iteration k as

$$\mathbf{s}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k. \tag{6}$$

Though NR is sometimes considered a minimization (or maximization) method, it is more accurately described as a root-finding algorithm that can be used to refine stationary points when applied to the gradient of a function. If $\mathbf{H}_k$ is positive definite, the resulting displacement vector $\mathbf{s}_k$ will generally step in the direction of a minimum. In contrast, if $\mathbf{H}_k$ has precisely one negative eigenvalue, then $\mathbf{s}_k$ will likely step in the direction of a FOSP.

As described in section 2.1, we do not generally have the ability to solve equations involving $\mathbf{H}_k^{-1}$ directly. Therefore, a standard approach is to replace $\mathbf{H}_k$ in eq 6 with an approximation $\mathbf{B}_k$,

$$\mathbf{s}_k = -\mathbf{B}_k^{-1}\mathbf{g}_k. \tag{7}$$

This is referred to as the quasi-Newton (QN) method.

It is worth considering the motivation behind eqs 6 and 7. In both cases, the potential energy at point $\mathbf{q}_k$ is expanded in terms of a displacement vector $\mathbf{s}_k$ to second order,

$$\epsilon(\mathbf{q}_k + \mathbf{s}_k) \approx \epsilon_k + \mathbf{g}_k^T\mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k. \tag{8}$$

Eq 8 has a single stationary point given by eq 7 (or eq 6 if $\mathbf{B}_k$ is replaced by $\mathbf{H}_k$). If $\mathbf{s}_k$ is large, then the neglect of third- and higher-order terms in eq 8 will result in significant error. As a result, effort must be taken to ensure that $\mathbf{s}_k$ remains reasonably small. We assume that the quadratic approximation is accurate when $\|\mathbf{s}_k\|_2 \leq \delta_k$, where δ_k is a trust radius that is either imposed *a priori* or determined automatically during geometry refinement. One way to satisfy this constraint is to evaluate eq 7, then reduce the magnitude of $\mathbf{s}_k$ to be equal to δ_k, but this results in a suboptimal step direction.

The standard trust region method (TRM) enforces the trust radius constraint by minimizing a Lagrangian equation adapted from eq 8,

$$\mathcal{L}_{\text{TRM}} = \epsilon_k + \mathbf{g}_k^T\mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k + \frac{1}{2}\xi\left(\mathbf{s}_k^T\mathbf{s}_k - \delta_k^2\right), \tag{9}$$

where ξ is the Lagrange multiplier. Since eq 9 is quadratic with respect to $\mathbf{s}_k$, it too

has a single stationary point,

$$\mathbf{s}_k = -\left(\mathbf{B}_k + \xi\mathbf{I}\right)^{-1}\mathbf{g}_k. \tag{10}$$

TRM is very effective for minimization, but much less effective for FOSP refinement. For minimization, it is always possible to find a value of ξ for any $\mathbf{g}_k$, $\mathbf{B}_k$, and δ_k such that $\mathbf{s}_k$ is a descent direction and $\|\mathbf{s}_k\|_2 \leq \delta_k$. In contrast, it is not always possible to find a value of ξ that results in $\mathbf{s}_k$ stepping towards a FOSP while also satisfying the trust region constraint.

The minimum mode following (MMF) class of methods avoids this flaw by modifying the gradient vector $\mathbf{g}_k$.[35–37,39,52] By definition, $\mathbf{g}_k$ points in the direction of steepest ascent on the PES. This is why $\mathbf{s}_k$ obtained from eq 7 is guaranteed to be a descent direction when $\mathbf{B}_k$ is positive definite. If $\mathbf{g}_k$ is modified to point in a different direction, this guarantee no longer holds. MMF methods replace $\mathbf{g}_k$ in eq 7 with a modified gradient vector

$$\tilde{\mathbf{g}}_k = \left(\mathbf{I} - 2\mathbf{x}_{\text{asc}}\mathbf{x}_{\text{asc}}^T\right)\mathbf{g}_k, \tag{11}$$

where $\mathbf{x}_{\text{asc}} \approx \mathbf{v}_k^{(1)}$ is the ascent direction chosen to approximate the leftmost eigenvector of $\mathbf{H}_k$. Eq 11 inverts the components of $\mathbf{g}_k$ in the ascent direction. Provided that $\mathbf{B}_k$ is positive definite and $\mathbf{x}_{\text{asc}}$ is appropriately chosen, replacing $\mathbf{g}_k$ with $\tilde{\mathbf{g}}_k$ in eq 7 will result in $\mathbf{s}_k$ stepping towards a FOSP.

However, the requirement that $\mathbf{B}_k$ be positive definite means that it will necessarily poorly approximate $\mathbf{H}_k$ near the FOSP, where $\mathbf{H}_k$ is indefinite by definition. This requirement also poses some difficulty for the secant updates to $\mathbf{B}_k$. Any update that would make $\mathbf{B}_{k+1}$ indefinite must be skipped, thereby leaving $\mathbf{B}_k$ unchanged. This occurs frequently during FOSP refinement, particularly near convergence, and as a result $\mathbf{B}_k$ will tend to deteriorate in quality over the course of refinement.

Moreover, MMF requires at least approximate knowledge of $\mathbf{v}_k^{(1)}$ to determine the ascent direction. If $\mathbf{B}_k$ is a good approximation to $\mathbf{H}_k$, then its leftmost eigenvector $\tilde{\mathbf{v}}_k^{(1)}$ should be a good approximation to $\mathbf{v}_k^{(1)}$. If $\mathbf{B}_k$ is positive definite, then it cannot be a

good approximation to $\mathbf{H}_k$ near the FOSP, and it is therefore unlikely that $\tilde{\mathbf{v}}_k^{(1)} \approx \mathbf{v}_k^{(1)}$.

Our intent in making these observations is not to suggest that MMF methods are somehow ineffective or theoretically unsound. Indeed, MMF methods have been very successful for applications in materials science and heterogeneous catalysis.[53–56] However, the requirement that $\mathbf{B}_k$ be positive definite is at odds with its intended role as an approximation to $\mathbf{H}_k$. The relative success of MMF methods comes at the cost of decoupling the estimation of $\mathbf{v}_k^{(1)}$ from construction of $\mathbf{B}_k$.

In order to enable the usage of an indefinite $\mathbf{B}_k$, we turn to the rational function optimization (RFO) method.[57–59] In RFO, the displacement vector $\mathbf{s}_k$ is obtained by minimizing a rational function,

$$\mu(\mathbf{s}_k) = \frac{\mathbf{g}_k^T \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T \mathbf{B}_k \mathbf{s}_k}{1 + \mathbf{s}_k^T \mathbf{W}_k \mathbf{s}_k}, \tag{12}$$

where $\mathbf{W}_k$ is an unspecified positive definite matrix, typically chosen to be a scaled identity matrix. The denominator in eq 12 has the effect of penalizing large displacement vectors, thereby guaranteeing the existence of a minimum even when $\mathbf{B}_k$ is not positive definite. Minimization of $\mu(\mathbf{s}_k)$ in eq 12 can be recast as an eigenvalue problem,

$$\begin{pmatrix} \alpha^2 \mathbf{B}_k & \alpha \mathbf{g}_k \\ \alpha \mathbf{g}_k^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{s}_k/\alpha \\ 1 \end{pmatrix} = 2\mu \begin{pmatrix} \mathbf{s}_k/\alpha \\ 1 \end{pmatrix}, \tag{13}$$

where we have made the substitution $\mathbf{W}_k = \alpha^{-2}\mathbf{I}$. Eq 13 is a standard eigenvalue problem, though the magnitude and sign of the eigenvector is strategically chosen to illustrate how $\mathbf{s}_k$ is to be evaluated. Eq 13 implies

$$\mathbf{s}_k = -\left(\mathbf{B}_k - \frac{2\mu}{\alpha^2}\mathbf{I}\right)^{-1} \mathbf{g}_k. \tag{14}$$

The relationship between RFO and TRM can be seen by comparing eq 14 with eq 10. For minimization, the displacement vector $\mathbf{s}_k$ is obtained from the leftmost eigenvector of eq 13. This choice guarantees that $\mathbf{B}_k - \frac{2\mu}{\alpha^2}\mathbf{I}$ will be positive definite regardless of

whether $\mathbf{B}_k$ is positive definite.

To reiterate, RFO recasts minimization as an eigenvalue problem. A displacement vector suitable for minimization is found by choosing the leftmost eigenvector of eq 13. If a higher eigenvalue is chosen instead, in theory eq 13 can be used to step towards saddle point structures as well. However, this approach does not reliably converge to saddle point structures. In order to find saddle point structures more reliably, partitioned rational function optimization (PRFO) splits eq 13 into two eigenvalue problems, one for the ascent direction(s) and one for the descent directions.[57] The matrix of eigenvectors $\tilde{\mathbf{V}}_k$ of $\mathbf{B}_k$ can be used as a basis for this partitioning,

$$\begin{pmatrix} \alpha^2 \tilde{\mathbf{V}}_k^{(\max)T} \mathbf{B}_k \tilde{\mathbf{V}}_k^{(\max)} & \alpha \tilde{\mathbf{V}}_k^{(\max)T} \mathbf{g}_k \\ \alpha \mathbf{g}_k^T \tilde{\mathbf{V}}_k^{(\max)} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{s}_k^{(\max)}/\alpha \\ 1 \end{pmatrix} = 2\mu^{(\max)} \begin{pmatrix} \mathbf{s}_k^{(\max)}/\alpha \\ 1 \end{pmatrix} \tag{15}$$

$$\begin{pmatrix} \alpha^2 \tilde{\mathbf{V}}_k^{(\min)T} \mathbf{B}_k \tilde{\mathbf{V}}_k^{(\min)} & \alpha \tilde{\mathbf{V}}_k^{(\min)T} \mathbf{g}_k \\ \alpha \mathbf{g}_k^T \tilde{\mathbf{V}}_k^{(\min)} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{s}_k^{(\min)}/\alpha \\ 1 \end{pmatrix} = 2\mu^{(\min)} \begin{pmatrix} \mathbf{s}_k^{(\min)}/\alpha \\ 1 \end{pmatrix}, \tag{16}$$

where $\tilde{\mathbf{V}}_k^{(\max)}$ contains the leftmost eigenvector(s) of $\mathbf{B}_k$ which span the ascent direction(s) and $\tilde{\mathbf{V}}_k^{(\min)}$ are the remaining eigenvectors which span the descent directions. The rightmost eigenvector of eq 15 and the leftmost eigenvector of eq 16 are used to calculate the displacement vector,

$$\mathbf{s}_k = \tilde{\mathbf{V}}_k^{(\max)} \mathbf{s}_k^{(\max)} + \tilde{\mathbf{V}}_k^{(\min)} \mathbf{s}_k^{(\min)}.$$

When seeking a FOSP, $\tilde{\mathbf{V}}_k^{(\max)}$ will contain only the leftmost eigenvector of $\mathbf{B}_k$. Unlike MMF, PRFO does not require $\mathbf{B}_k$ to be positive definite, or indeed that it have any particular number of negative eigenvalues. Indeed, PRFO will perform best when $\mathbf{B}_k$ accurately approximates $\mathbf{H}_k$. This means it is possible to step towards a FOSP with a highly accurate $\mathbf{B}_k$ even when the true Hessian is positive definite or has multiple negative eigenvalues. PRFO is used extensively by FOSP refinement algorithms that evaluate $\mathbf{H}$ for the initial structure in full. PRFO is not as commonly used by FOSP

refinement algorithms that rely on iterative diagonalization of $\mathbf{H}$.

The magnitude of the displacement vector $\|\mathbf{s}_k\|_2$ can be controlled by changing the value of α. In restricted step PRFO (RS-PRFO), α is chosen such that $\|\mathbf{s}_k\|_2 \leq \delta_k$.[59] This typically requires an iterative procedure in which various values of α are tested until a suitable value is found. This is the method used in this work.

2.3 Approximate Hessian updates

In section 2.2, we described a class of geometry refinement methods that rely on the availability of a matrix $\mathbf{B}_k$ that approximates $\mathbf{H}_k$. Construction of $\mathbf{B}_k$ is not a straightforward task, and much work has gone into the development of methods for constructing a $\mathbf{B}_k$ that is suitable for geometry refinement. Following a displacement in the direction $\mathbf{s}_k = \mathbf{q}_{k+1} - \mathbf{q}_k$, the change in the gradient $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ provides a finite difference approximation to the Hessian-vector product $\mathbf{H}_{k+1}\mathbf{s}_k$. Given the current approximation $\mathbf{B}_k$, an approximate Hessian update strategy must find a correction matrix $\mathbf{E}_k$ which is symmetric and which satisfies the secant condition $(\mathbf{B}_k + \mathbf{E}_k)\mathbf{s}_k = \mathbf{y}_k$. There are infinitely many ways to construct $\mathbf{E}_k$ given these constraints, but one is typically interested in a correction that is in some sense minimal. Greenstadt[60] developed a general formula for $\mathbf{E}_k$ that satisfies the symmetry and secant conditions and which minimizes the weighted norm $\|\mathbf{E}_k\|^2_{\mathbf{M}_k^{-1}}$, defined as

$$\|\mathbf{E}_k\|^2_{\mathbf{M}_k^{-1}} = \mathrm{Tr}\left(\mathbf{M}_k^{-1}\mathbf{E}_k\mathbf{M}_k^{-1}\mathbf{E}_k^T\right). \tag{17}$$

The matrix $\mathbf{M}_k$ must be symmetric positive definite but is otherwise unspecified. While the purpose of $\mathbf{E}_k$ is to ensure that $\mathbf{B}_{k+1}$ satisfies the secant condition, it will also necessarily perturb the components of $\mathbf{B}_k$ that are orthogonal to $\mathbf{s}_k$. The matrix $\mathbf{M}_k$ determines how this correction should be distributed by weighting the norm of $\mathbf{E}_k$. If $\mathbf{M}_k$ is chosen to be indefinite, then a large error may be introduced into $\mathbf{B}_{k+1}$.

The solution which minimizes eq 17 subject to the secant and symmetry constraints

is

$$\mathbf{E}_k = \mathbf{u}_k\mathbf{j}_k^T + \mathbf{j}_k\mathbf{u}_k^T - \mathbf{u}_k\mathbf{j}_k^T\mathbf{s}_k\mathbf{u}_k^T, \tag{18}$$

where

$$\begin{aligned}\mathbf{j}_k &= \mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k \\ \mathbf{u}_k &= \mathbf{M}_k\mathbf{s}_k\left[\mathbf{s}_k^T\mathbf{M}_k\mathbf{s}_k\right]^{-1}.\end{aligned}$$

Greenstadt's eponymous method chooses $\mathbf{M}_k^{\text{Greenstadt}} = \mathbf{B}_k$, though this has proven to not be very effective.[60]

The simplest possible choice for $\mathbf{M}_k$ is the identity matrix, which results in the Powell-symmetric Broyden (PSB) method. PSB tends to perturb all eigenvalues of $\mathbf{B}_k$ essentially equally. This results in larger *relative* errors for lower magnitude eigenmodes. These modes correspond to directions with the largest displacement vectors in geometry refinement steps, so the PSB update can have deleterious effects on geometry refinement performance.

By far the most commonly used and effective Hessian update method for minimization is that of Broyden, Fletcher, Goldfarb and Shanno (BFGS),[61–64]

$$\mathbf{M}_k^{\text{BFGS}} = \frac{\sqrt{\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k}\mathbf{B}_{k+1} + \sqrt{\mathbf{y}_k^T\mathbf{s}_k}\mathbf{B}_k}{\sqrt{\mathbf{y}_k^T\mathbf{s}_k} + \sqrt{\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k}}, \tag{19}$$

which uses a linear combination of $\mathbf{B}_k$ and $\mathbf{B}_{k+1}$ for $\mathbf{M}_k$. Note that the presence of $\mathbf{B}_{k+1}$ in eq 19 is not a concern, as it is only necessary to evaluate $\mathbf{M}_k\mathbf{s}_k$, and $\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k$ by construction. Eq 19 has the effect of preferentially perturbing the larger magnitude eigenmodes of $\mathbf{B}_k$. This preserves the accuracy of lower magnitude eigenmodes. However, $\mathbf{M}_k^{\text{BFGS}}$ may be indefinite if $\mathbf{B}_k$ or $\mathbf{B}_{k+1}$ are indefinite. It is therefore not appropriate to use BFGS when $\mathbf{B}_k$ is expected to be indefinite, as $\mathbf{M}_k$ must be positive definite.

Another somewhat popular secant update method is that of Murtagh-Sargent (MS),[65]

$$\mathbf{M}_k^{\mathrm{MS}} = \mathbf{B}_{k+1} - \mathbf{B}_k \, . \tag{20}$$

Eq 20 is the only choice of $\mathbf{M}_k$ for which $\mathbf{E}_k$ is rank one; consequently, this method is sometimes referred to as the symmetric rank one (SR1) update. As with BFGS, $\mathbf{M}_k^{\mathrm{MS}}$ is not always positive definite. Additionally, if $\mathbf{B}_k\mathbf{s}_k \approx \mathbf{y}_k$, this implies $\mathbf{M}_k^{\mathrm{MS}}\mathbf{s}_k \approx \mathbf{0}$, and so eq 18 will become numerically unstable. One would normally expect $\mathbf{E}_k$ to be very small in this situation, as the secant condition is already close to satisfied by $\mathbf{B}_k$, but MS may introduce a very large correction due to the numerical instability of eq 18.

To alleviate this problem somewhat, the Murtagh-Sargent Powell (MSP) method constructs $\mathbf{E}_k$ as a linear combination of the updates obtained by the PSB and MS methods,[66,67]

$$\mathbf{E}_k^{\mathrm{MSP}} = \phi\mathbf{E}_k^{\mathrm{PSB}} + (1-\phi)\mathbf{E}_k^{\mathrm{MS}},$$

where the scaling factor ϕ is defined as

$$\phi = 1 - \frac{\left(\mathbf{s}_k^T\mathbf{j}_k\right)^2}{\left(\mathbf{s}_k^T\mathbf{s}_k\right)\left(\mathbf{j}_k^T\mathbf{j}_k\right)}. \tag{21}$$

Methods that use alternate definitions of ϕ have also occasionally been referred to as MSP.[68] Similar methods using a linear combination of BFGS and MS updates have also been developed.[69] While these approaches may work in practice, they rely on secant updates that can become numerically unstable even under ideal circumstances, and especially when $\mathbf{B}_k$ is indefinite.

An alternative approach developed by Anglada and Bofill constructs $\mathbf{M}_k$ in a way that is superficially similar to BFGS, but which is guaranteed to be positive definite.[68,70] Their TS-BFGS method uses

$$\mathbf{M}_k^{\mathrm{TS-BFGS}} = \mathbf{y}_k\mathbf{y}_k^T + |\mathbf{B}_k|\mathbf{s}_k\mathbf{s}_k^T|\mathbf{B}_k|, \tag{22}$$

where

$$|\mathbf{B}_k| = \sum_{i=1}^{d} \left|\tilde{\lambda}_k^{(i)}\right| \tilde{\mathbf{v}}_k^{(i)} \tilde{\mathbf{v}}_k^{(i)T}, \tag{23}$$

where $\tilde{\lambda}_k^{(i)}$ and $\tilde{\mathbf{v}}_k^{(i)}$ are the eigenvalues and eigenvectors of $\mathbf{B}_k$, respectively. This method is employed in the current work.

It is also possible to modify Greenstadt's method to enable updates to $\mathbf{B}_k$ using multiple displacement vectors $\mathbf{S}$ and corresponding curvature estimates $\mathbf{Y}$ simultaneously.[71] These multi-secant updates have an expression that is very similar to the standard secant update,

$$\begin{aligned} \mathbf{E}_k &= \mathbf{U}_k \mathbf{J}_k^T + \mathbf{J}_k \mathbf{U}_k^T - \mathbf{U}_k \mathbf{J}_k^T \mathbf{S} \mathbf{U}_k^T \\ \mathbf{J}_k &= \mathbf{Y} - \mathbf{B}_k \mathbf{S} \\ \mathbf{U}_k &= \mathbf{M}_k \mathbf{S} \left[\mathbf{S}^T \mathbf{M}_k \mathbf{S}\right]^{-1}, \end{aligned} \tag{24}$$

In order for $\mathbf{E}_k$ defined in eq 24 to be symmetric, the product $\mathbf{Y}^T\mathbf{S}$ must also be symmetric, which cannot be guaranteed in general. Thus, either $\mathbf{Y}$ or $\mathbf{S}$ must be modified in some way to ensure $\mathbf{E}_k$ is symmetric. Several possible ways of accomplishing this have been developed by Schnabel.[72] In the current work, algorithm 3.1 from reference 72 is used to modify $\mathbf{Y}$. This algorithm describes a procedure for constructing a minimal perturbation $\mathbf{\Gamma}$ such that $(\mathbf{Y} + \mathbf{\Gamma})^T \mathbf{S} = \mathbf{S}^T (\mathbf{Y} + \mathbf{\Gamma})$. In this procedure, the leftmost column of $\mathbf{\Gamma}$ is zero. The relevance of this property will be discussed in section 3.1.

3 Methods

Established FOSP refinement procedures outlined in the introduction and in section 2 present an all-or-nothing choice when it comes to the quality of $\mathbf{B}$. If $\mathbf{H}$ is calculated explicitly for the initial structure, then $\mathbf{B}$ is initially exact, and geometry refinement will converge in relatively few steps. However, this can be prohibitively expensive for systems of many atoms, which instead diagonalize $\mathbf{H}$ iteratively to determine its

leftmost eigenvector. When $\mathbf{H}$ is diagonalized iteratively, it is not possible to construct $\mathbf{B}$ to be initially exact. In general, these methods initialize $\mathbf{B}$ as a scaled identity matrix. As we describe in section 2.3, it is possible to simultaneously update $\mathbf{B}$ with a matrix of displacement vectors. Conveniently, this information can be extracted from the iterative diagonalization method described in section 2.1. This key observation allows us to construct significantly higher quality $\mathbf{B}$ matrices, which results in a substantial improvement in performance for FOSP refinement.

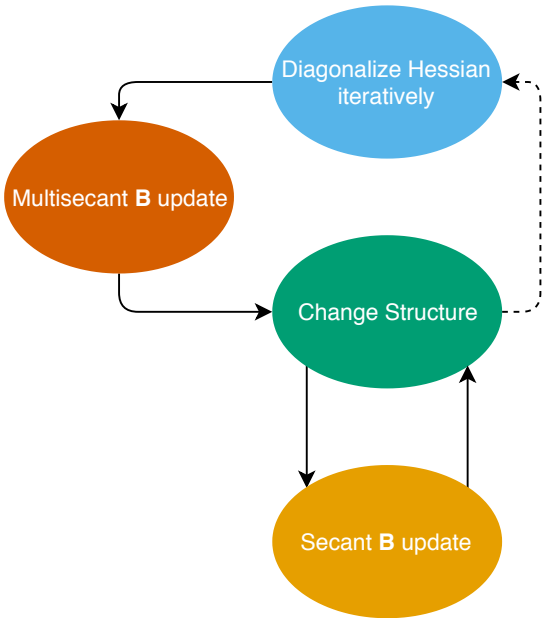


Figure 2: An illustration of the method presented in this work.

The FOSP refinement method we have developed is outlined in figure 2. We iteratively diagonalize $\mathbf{H}_k$ to construct an approximation $\mathbf{B}_k$ using a multi-secant TS-BFGS Hessian update as described in section 2.3. $\mathbf{B}_k$ is then used with the RS-PRFO method described in section 2.2 to determine step direction and magnitude. The leftmost eigenvector of $\mathbf{B}_k$, $\tilde{\mathbf{v}}_k^{(1)}$, is chosen as the ascent direction. After each geometry refinement step, a standard TS-BFGS Hessian update is applied to $\mathbf{B}_k$. All invocations of the iterative diagonalization algorithm (after the first) use $\mathbf{B}_k$ as a preconditioner as described in section 2.1.

3.1 Multi-secant Hessian updates

In section 2.3, we describe several different secant update methods and the more general multi-secant update. Our key observation is that the multi-secant update can be used to incorporate the trial vectors $\mathbf{S}$ and the corresponding Hessian-vector products $\mathbf{Y}$ obtained by the diagonalization algorithm into $\mathbf{B}_k$. For FOSP refinement, $\mathbf{H}_k$ is expected to be indefinite, and the performance of RS-PRFO is predicated on $\mathbf{B}_k$ having a similar structure to $\mathbf{H}_k$. Thus we cannot use any secant update method that expects or requires a positive definite $\mathbf{B}_k$, such as BFGS. MSP and related methods cannot be easily generalized to multi-secant updates, as the definition of ϕ from eq 21 becomes ambiguous. We therefore use the TS-BFGS method, which can be readily generalized to multi-secant updates by using eq 22 with eq 24.

Construction of the initial approximate Hessian $\mathbf{B}_1$ requires special consideration. The iterative diagonalization algorithm does not provide enough information to fully specify $\mathbf{B}_1$, unless it is run to completion, at which point the full Hessian has been evaluated. We must therefore estimate the curvature in the directions not searched by the diagonalization algorithm. We solve this problem by initializing $\mathbf{B}_0$ as a scaled identity matrix,

$$\mathbf{B}_0 = \left(\frac{1}{m} \sum_{j=1}^{m} \left| \theta_m^{(j)} \right| \right) \mathbf{I},$$

where the scaling coefficient is chosen to be the average absolute Ritz value. The multi-secant update is then applied to $\mathbf{B}_0$ to obtain $\mathbf{B}_1$. The choice of how to initialize $\mathbf{B}_0$ has a large impact on overall performance. If the prefactor used to initialize $\mathbf{B}_0$ is chosen to be too small or large in magnitude, then the components of geometry refinement steps orthogonal to $\mathbf{S}$ will be too large or small, respectively. This not only hinders convergence to the FOSP, but may also make it more difficult to update $\mathbf{B}_k$ with accurate curvature estimates. Given the information available, the average absolute Ritz value is a reasonable approximate curvature for directions orthogonal to $\mathbf{S}$.

As we described in section 2.3, the multi-secant update requires that the product $\mathbf{Y}^T\mathbf{S}$ be symmetric. This requirement is generally not satisfied in the current ap-

plication because of finite difference artifacts and noise in the gradient vectors. The procedure described by Schnabel[72] solves this problem by adding a small corrective term $\mathbf{\Gamma}$ to $\mathbf{Y}$. As previously mentioned, the first column of $\mathbf{\Gamma}$ is zero. We exploit this property to preserve the accuracy of the leftmost Ritz value determined by the iterative diagonalization procedure by right-multiplying both $\mathbf{S}$ and $\mathbf{Y}$ by $\mathbf{C}$, the matrix of primitive Ritz vectors. This results in $\mathbf{S}$ becoming the matrix of Ritz vectors, with $\mathbf{Y}$ being composed of the corresponding Hessian-vector products. Applying the procedure described by Schnabel will thus not modify the leftmost Ritz vector while still ensuring that $\mathbf{B}_{k+1}$ remains symmetric.

3.2 Iterative diagonalization algorithm

We use the JD0 approach described in section 2.1. Our testing shows very little difference between JD0 and GD, though JD0 is expected to be more stable when $\mathbf{B}_k$ is a very good approximation to $\mathbf{H}_k$. For the initial call to JD0, no $\mathbf{B}_k$ is available to act as a preconditioner, hence our implementation uses the identity matrix $\mathbf{I}$ as the preconditioner instead. In this case, JD0 and GD both become mathematically equivalent to Lanczos.

In our implementation of the Rayleigh-Ritz procedure, the jth Ritz pair is considered converged if $\|\mathbf{r}_m^{(j)}\|_2 < \gamma|\theta_m^{(1)}|$, where γ is a tunable convergence parameter. Smaller values of γ result in more diagonalization algorithm iterations, which improves the accuracy of $\mathbf{B}_k$ at the cost of added computational expense. Our implementation seeks to converge all negative eigenvalues of $\mathbf{H}_k$, not only the leftmost eigenvalue. This increases the likelihood of converging rapidly to the FOSP when multiple negative eigenvalues are present.

If care is not taken, the columns of $\mathbf{S}_m$ may become partially linearly dependent. This can be avoided if an orthogonalization procedure such as modified Gram-Schmidt is repeatedly applied to the extension vector $\mathbf{t}_m$ until the vector no longer changes. This is the approach used in the current method. For large, sparse matrix applications, this would be prohibitively expensive, but the dimension of problems in the current

application are expected to be relatively small. Additionally, if the length of $\mathbf{t}_m$ is found to be less than 1 % of its original value after a single iteration of Gram-Schmidt, then we consider it to be linearly dependent on $\mathbf{S}_m$, and instead use Lanczos to expand $\mathbf{S}_m$ for the current diagonalization iteration. This helps prevent stagnation which can occur when the component of $\mathbf{t}_m$ that is orthogonal to $\mathbf{S}_m$ is small and dominated by noise.

The procedure developed by Schnabel[72] described in sections 2.3 and 3.1 to symmetrize the product $\mathbf{S}_m^T\mathbf{Y}_m$ is also applied during the Rayleigh-Ritz procedure. This symmetrization is used in the construction of $\mathbf{A}_m$ and for the residual vectors $\mathbf{r}_m^{(j)}$ (see section 2.1). The correction matrix $\mathbf{\Gamma}$ is re-evaluated during every Rayleigh-Ritz iteration and then discarded. This is done to avoid accumulation of error in $\mathbf{Y}_m$.

3.3 Geometry refinement

We use RS-PRFO to determine the step direction and magnitude. At each iteration, a trial displacement vector is found by solving the RS-PRFO eigenvalue equations (eqs 15 and 16) with $\alpha = 1$. If the trial vector is smaller in magnitude than δ_k, then it is accepted, and the energy and gradient are evaluated at the new point. Otherwise, we invoke an iterative procedure to determine the value of α which produces a displacement vector such that $\|\mathbf{s}_k\|_2 = \delta_k$. We accomplish this by applying NR to the residual $\|\mathbf{s}_k\|_2 - \delta_k$ with the analytical gradient $\frac{d\|\mathbf{s}_k\|_2}{d\alpha}$, falling back to the bisection method when NR fails to provide a reasonable value for α.

We use an automated strategy for determining δ_k. After every step, we evaluate the ratio of the predicted change in energy to the true change in energy,

$$\rho = \frac{\mathbf{g}_k^T\mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k}{\epsilon_{k+1} - \epsilon_k}.$$

A value of $\rho \approx 1$ indicates that the quadratic approximation to the energy is accurate, and the trust radius may be safely increased. When ρ is very large, close to zero, or negative, the trust radius should be decreased.

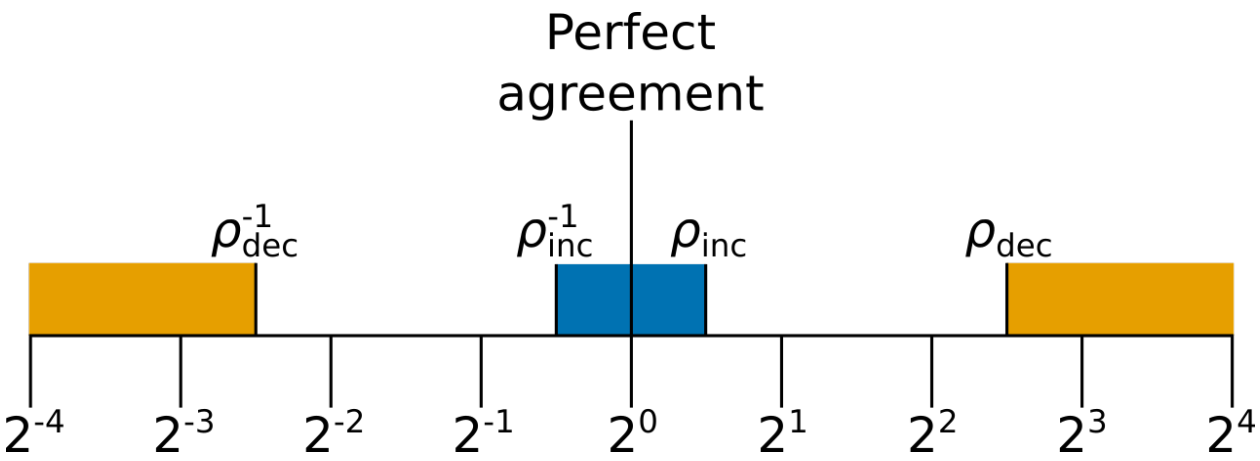


Figure 3: An illustration of how we choose to update the trust radius based on the ratio of the predicted change in energy to the true change in energy, ρ. When $\rho_{\text{inc}}^{-1} < \rho < \rho_{\text{inc}}$ (central blue region), the trust radius is increased. When $\rho < \rho_{\text{dec}}^{-1}$ or $\rho > \rho_{\text{dec}}$ (peripheral orange regions), the trust radius is reduced. In all other cases, the trust radius is left unchanged.

The method we use to update δ_k is illustrated in figure 3. The initial trust radius δ_0 is a parameter to be specified at the beginning of FOSP refinement. Two parameters, ρ_{dec} and ρ_{inc}, are used to determine whether the trust radius should be increased, decreased, or left unchanged. Two additional parameters, σ_{dec} and σ_{inc}, are used to determine the value of the new trust radius. When $\rho_{\text{inc}}^{-1} < \rho < \rho_{\text{inc}}$ (the blue region in figure 3), the trust radius is set to $\delta_{k+1} = \max\left(\sigma_{\text{inc}}\|\mathbf{s}_k\|_2, \delta_k\right)$. When $\rho < \rho_{\text{dec}}^{-1}$ or $\rho > \rho_{\text{dec}}$ (the orange regions), the trust radius is set to $\delta_{k+1} = \max\left(\sigma_{\text{dec}}\|\mathbf{s}_k\|_2, \delta_{\text{min}}\right)$. In all other situations, $\delta_{k+1} = \delta_k$. The value δ_{min} is a lower bound to the values the trust radius is allowed to take. This is necessary because the true change in energy $\epsilon_{k+1} - \epsilon_k$ will become dominated by noise as $\|\mathbf{s}_k\|_2$ approaches zero. In this scenario, ρ will be far from one, resulting in a reduction of δ_k, which will further exacerbate the problem. We choose $\delta_{\text{min}} = \eta$, the finite displacement step size used by the iterative diagonalization routine. The scaling factors σ_{dec} and σ_{inc} should be slightly below and slightly above one, respectively. Oscillation in geometry refinement steps may occur if $\sigma_{\text{dec}} = \sigma_{\text{inc}}^{-1}$, so this should be avoided.

3.4 Additional considerations

The energy of a system of atoms is invariant to net translation in the absence of an external potential. The energy is further invariant to net rotation in the absence of periodic boundary conditions. These zero-curvature modes can complicate determination of the lowest curvature *internal* modes that we are interested in finding. Additionally, if $\mathbf{B}_k$ becomes singular, it may be impossible to solve the RS-PRFO eigenvalue problems. To alleviate these problems, we construct an orthonormal basis that is orthogonal to translational and rotational modes using singular value decomposition. The Hessian is iteratively diagonalized and geometry refinement steps are determined in this reduced basis.

We note that rotational modes are generally only eigenvectors of $\mathbf{H}_k$ at stationary points in a Cartesian representation. Even in the absence of an external potential and for systems without periodic boundary conditions, $\mathbf{H}_k$ will generally only have three eigenvalues that are precisely zero, except at stationary points. This means that the eigenvalues and eigenvectors of $\mathbf{H}_k$ will be modified when employing the procedure we have just described. In practice, we find that projecting out rotational modes improves performance.

4 Results

The method described in this work has been implemented in Sella,[73] an open-source software package written in Python. Sella interfaces with a variety of electronic structure theory packages such as NWChem,[74] Quantum Espresso,[75,76] CP2K,[77] and GPAW[78] through the ASE library.[79] Sella runs on all major operating systems and CPU architectures.

Our algorithm, as implemented in Sella, has six hyperparameters: the Rayleigh-Ritz convergence parameter γ, the initial trust radius δ_0, the trust radius increase and decrease thresholds ρ_{inc} and ρ_{dec}, and the trust radius increase and decrease factors σ_{inc} and σ_{dec}. For all tests below, we choose $\gamma = 0.4$, $\delta_0 = 1.3 \times 10^{-3}$ Å per degree of

freedom, $\rho_{\text{inc}} = 1.035$, $\rho_{\text{dec}} = 5.0$, $\sigma_{\text{inc}} = 1.15$, and $\sigma_{\text{dec}} = 0.65$. Note that the value of the initial trust radius δ_0 scales with the number of degrees of freedom. This is to enable the treatment of systems of very different sizes.

All calculations were performed with Sella. Scripts to reproduce these calculations are available in the SI.

4.1 FOSP refinement benchmarks

We measure Sella's overall FOSP refinement performance using two FOSP refinement benchmarks from optbench.org.[80] The LJ38 benchmark consists of 200 38-atom Lennard Jones clusters, while the Pt-heptamer island benchmark consists of 49 Pt(1 1 1) surfaces with adsorbed 7-atom clusters simulated with a Morse potential. These benchmarks are designed to resemble systems encountered in materials science, such as bulk solids or metal atom clusters. For both of these benchmarks, the goal is to refine each initial structure to a FOSP in the fewest number of gradient evaluations n_{grad} with a convergence criterion of $\|\mathbf{g}_k\|_2 \leq 10^{-3}$. The tests do not stipulate to which FOSP the initial structures must converge. Energy and gradient evaluations for the LJ38 benchmark systems were performed with the ASE implementation of the Lennard-Jones potential.[79] LAMMPS was used to evaluate the energies and gradients of the Pt-heptamer island benchmark systems.[81] We compare Sella's performance on this benchmark to that of the two best-performing codes from the original benchmark publication,[80] Optim[82] and Pele.[83]

Table 1: LJ38 optbench.org FOSP refinement benchmark.[80]

Code	mean(n_{grad})	min(n_{grad})	max(n_{grad})
Sella	70	24	159
Optim	145	57	565
Pele	192	59	1488

Sella's performance on the LJ38 optbench.org FOSP refinement benchmark is outlined in table 1. Sella requires less than half as many gradient evaluations on average for this benchmark compared to next best performing code, Optim. In fact, the aver-

age number of gradient evaluations required by Optim is only slightly less than Sella's worst-performing configuration. Pele requires significantly more gradient evaluations on average than Sella's worst performing configuration.

Table 2: Pt-heptamer island optbench.org FOSP refinement benchmark. [80]

Code	mean(n_{grad})	min(n_{grad})	max(n_{grad})
Sella	53	31	108
Optim	71	43	143
Pele	88	52	198

Sella's performance on the Pt-heptamer island optbench.org FOSP refinement benchmark is outlined in table 2. As with the LJ38 benchmark, Sella outperforms to two current best-performing codes, Optim and Pele. The improvement in performance for this test is less substantial than for the LJ38 test, though Sella still requires 25 % fewer gradient evaluations than the next best performing code, Optim.

These performance improvements can be explained by the two major ways in which our method differs from traditional MMF methods. First, we use the JD0 iterative diagonalization method instead of Lanczos, which accelerates convergence when an approximate Hessian is known. Second, we use a multi-secant Hessian update to construct our approximate Hessian from the information provided by the diagonalization algorithm. The relative performance of our iterative diagonalization algorithm is discussed in section 4.2. The role of the multi-secant Hessian updates is investigated in section 4.3.

4.2 Iterative diagonalization benchmark

We separately measure the performance of the iterative diagonalization algorithms implemented in Sella using the iterative diagonalization benchmark from optbench.org. [80] This benchmark consists of 200 38-atom Lennard-Jones clusters. The goal of this benchmark is to find the leftmost eigenvector of the Hessian for each structure to within a target accuracy of $\left|\mathbf{x}_m^{(1)T}\mathbf{v}^{(1)}\right| \geq 0.99$ in the fewest number of gradient evaluations n_{grad}. The convergence criteria corresponds to an angle of less than $\arccos(0.99) \approx 8^\circ$ between

the leftmost Ritz vector and the true leftmost eigenvector.

Table 3: Optbench.org lowest-eigenvector benchmark

Code	mean(n_{grad})	min(n_{grad})	max(n_{grad})
Sella	18	4	37
Optim	25	13	58
Pele	25	12	61

We compare the performance of Sella's iterative diagonalization routines on this benchmark to that of Optim and Pele in table 3. For this test, Sella requires almost 30 % fewer gradient evaluations on average compared to both Optim and Pele. The source of this improvement in performance is not immediately evident, as both Optim and Pele also employ an iterative diagonalization algorithm based on the Rayleigh-Ritz procedure. No approximate Hessian is available for these test structures, so the JD0 procedure used by Sella is mathematically equivalent to Lanczos.

One possible difference is in the choice of initial guess vector for the iterative diagonalization procedure. This benchmark provides a single initial guess vector $\mathbf{x}_0^{\text{optbench}}$ for the leftmost eigenvector to be used with all 200 test structures. In contrast, Sella by default will use the gradient vector $\mathbf{g}$ as its initial guess vector. Another possible difference is that Sella projects zero-curvature translational and rotational modes out of the space searched by the iterative diagonalization method (see section 3.4). Other codes may instead shift these zero-curvature modes to better separate them from the low-curvature internal modes of interest. To see how these factors affect iterative diagonalization performance, we measure Sella's performance for each combination of initial test vector and zero-curvature mode treatment.

Table 4: Comparison of initial guess vector and zero-curvature mode treatment on optbench.org lowest-eigenvector benchmark

$\mathbf{x}_0$	Zero-curvature modes	mean(n_{grad})	min(n_{grad})	max(n_{grad})
$\mathbf{x}_0^{\text{optbench}}$	Projection	21	10	44
	Shifting	22	11	44
$\mathbf{g}$	Projection	18	4	37
	Shifting	18	4	37

The performance of these different combinations is shown in table 4. These results indicate that $\mathbf{g}$ is always a better choice of initial guess of the leftmost eigenvector than $\mathbf{x}_0^{\text{optbench}}$. Shifting and projection both afford the same performance when $\mathbf{g}$ is used as the initial guess vector. In fact, because $\mathbf{g}$ is necessarily orthogonal to translation and rotation, any reasonable treatment of these zero-curvature modes will result in the same performance. Our tests show that this is true even if diagonalization is performed in the full dimensional representation without shifting, *i.e.* when $\mathbf{H}$ is allowed to remain singular.

In contrast, choosing $\mathbf{x}_0^{\text{optbench}}$ as the initial guess has poorer performance that is slightly affected by the treatment of zero-curvature modes. This is because $\mathbf{x}_0^{\text{optbench}}$ is not strictly orthogonal to translational and rotational modes for all systems. In principle, $\mathbf{x}_0^{\text{optbench}}$ can be orthogonalized against these modes; in fact, this must be done when the translational and rotational modes are projected out. When $\mathbf{x}_0^{\text{optbench}}$ is not orthogonalized in this way, as in the shifted case, more iterations are required to reach convergence.

These factors alone are not sufficient to explain Sella's improved performance on this test relative to Optim and Pele. Even if $\mathbf{x}_0^{\text{optbench}}$ is used as the initial guess vector and if the zero-curvature modes are shifted rather than projected, Sella still outperforms both Optim and Pele. We suspect that this may be a result of the very rigorous numerical routines implemented in Sella, specifically the orthogonalization and symmetrization routines described in section 3.2.

Table 5: Comparing performance of iterative diagonalization procedures on final structures from LJ38 optbench.org FOSP refinement benchmark

Method	Zero-curvature modes	mean(n_{grad})	min(n_{grad})	max(n_{grad})
Lanczos	Projection	21.3	7	36
	Shifting	22.3	8	39
JD0	Projection	12.0	5	18
	Shifting	12.0	5	18

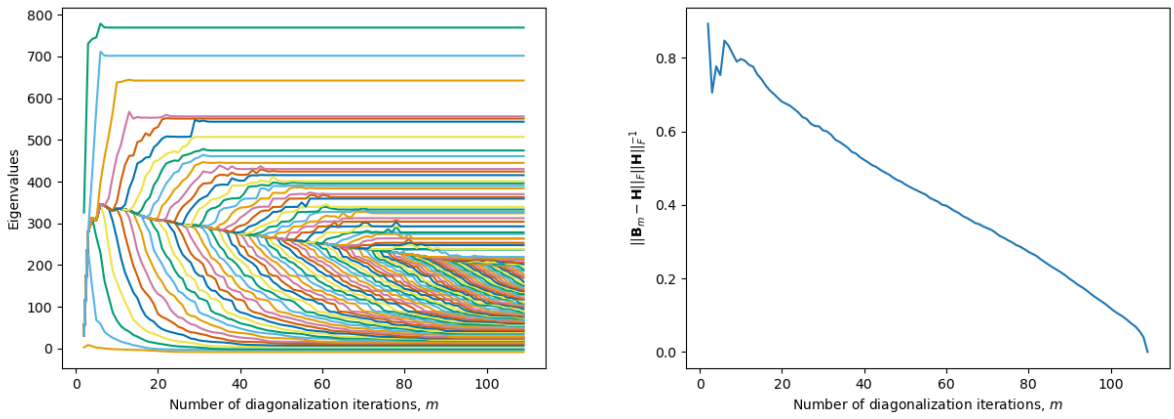
The diagonalization benchmarks established by optbench.org do not provide an opportunity to compare the performance of the preconditioned JD0 diagonalization

routine with that of Lanczos. In order to draw this comparison, we also measure the performance of the diagonalization algorithm for the final converged structures corresponding to the LJ38 FOSP refinement benchmark from table 1. These results are outlined in table 5. As this is not a part of the original optbench.org benchmark suite, we cannot compare Sella's performance against that of Optim and Pele. For these tests, the leftmost eigenvector of the approximate Hessian $\mathbf{B}_k$ following FOSP refinement is used as the initial guess for the leftmost eigenvector of $\mathbf{H}_k$. Additionally, $\mathbf{B}_k$ was used as a preconditioner for the JD0 method. In order to see a measurable difference in performance, the convergence criterion for this test was tightened considerably to $\left|\mathbf{x}_m^{(1)T}\mathbf{v}^{(1)}\right| \geq 0.999\,999$, or an angle of less than $\arccos(0.999\,999) \approx 0.08°$. We see that JD0 consistently requires fewer gradient evaluations to converge to the leftmost eigenvector of $\mathbf{B}_k$. Lanczos is also more strongly affected by the treatment of zero curvature modes, as eigenvalue shifting requires on average one additional gradient evaluation to reach convergence. This effect is absent in JD0, which requires the same number of gradient evaluations to converge for both projection and eigenvalue shifting.

4.3 Hessian update analysis

The key advance of our method is that we incorporate all information of $\mathbf{H}_k$ obtained by the diagonalization algorithm into the approximate Hessian $\mathbf{B}_k$ using a multi-secant update. This generally results in $\mathbf{B}_k$ becoming indefinite, which is not allowed for MMF methods but is permitted if RS-PRFO is used. RS-PRFO is capable of constructing displacement vectors appropriate for FOSP refinement regardless of the number of negative eigenvalues in $\mathbf{B}_k$, unlike leading MMF methods.

In figure 4, we show how the accuracy of the approximate Hessian $\mathbf{B}$ improves as the number of diagonalization iterations increases for an example structure taken from the LJ38 FOSP refinement benchmark (see table 1). Figure 4a depicts the eigenvalue spectrum of $\mathbf{B}$ initialized by the method described in section 3.1 after the given number of diagonalization iterations. For this system, which consists of 38 atoms and thus has 108 degrees of freedom excluding translation and rotation, full diagonalization is



(a) (b)

Figure 4: Comparing the accuracy of the approximate Hessian $\mathbf{B}_m$ as a function of number of diagonalization algorithm iterations m. (a) The eigenvalue spectrum of the approximate Hessian as a function iteration number m. (b) The relative error in the approximate Hessian after m diagonalization iterations, $\|\mathbf{B}_m - \mathbf{H}\|_F \|\mathbf{H}\|_F^{-1}$, where $\|\cdot\|_F$ represents the Frobenius norm.

achieved after 109 gradient evaluations. As expected, the largest magnitude eigenvalues converge more rapidly than eigenvalues closer to zero. However, the leftmost eigenvalues also converge relatively quickly, even though they are quite low in magnitude. Figure 4b shows how the relative error in $\mathbf{B}$ decreases with increasing diagonalization algorithm iterations. Note that even when the extremal eigenvalues are well converged, the error in $\mathbf{B}$ can be quite large.

It is clear that increasing the number of diagonalization algorithm iterations will result in a better approximate Hessian. It is less clear how this increase in the quality of $\mathbf{B}$ translates to performance of the FOSP refinement algorithm. Figure 5 illustrates how the number of geometry refinement steps and the total number of gradient evaluations required are affected by the diagonalization algorithm convergence criterion γ. We selected two structures from the LJ38 FOSP refinement benchmark (see table 1), and for each structure performed a series of full FOSP refinements with values of γ between 10^{-16} and 10^{2}. When $\gamma = 10^{-16}$, the iterative diagonalization method effectively diagonalizes the full Hessian matrix. When $\gamma = 10^{2}$, the iterative diagonalization method is considered converged after a single iteration. Within this range, smaller

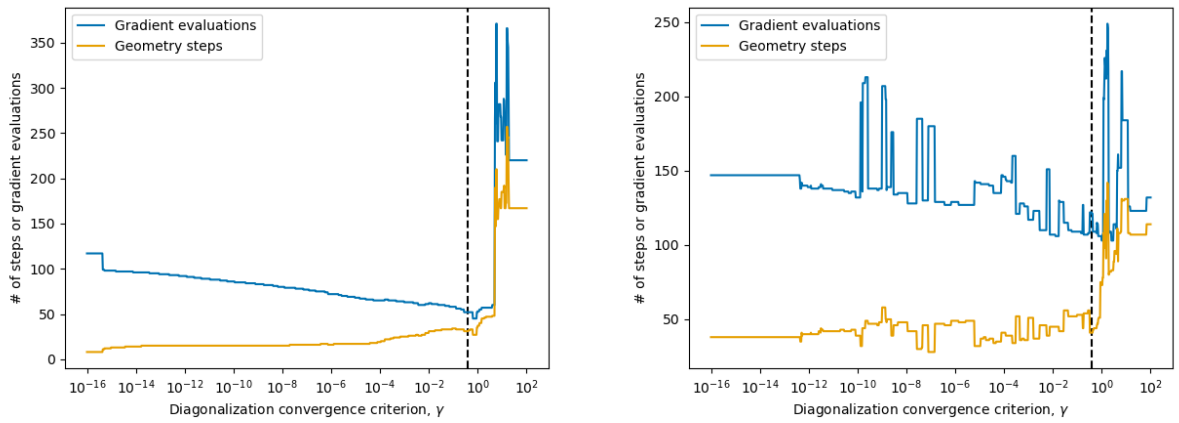


Figure 5: The number of geometry refinement steps (orange) and gradient evaluations (blue) required to reach convergence as a function of the eigensolver convergence criterion γ. (a) A good initial structure for which the diagonalization algorithm is only called once at the beginning of refinement. (b) A poor quality initial structure for which the diagonalization algorithm is called multiple times over the course of refinement. The vertical dashed black line indicates the choice of γ used in the current work.

values of γ result in more steps of the iterative diagonalization algorithm, thereby producing a more accurate $\mathbf{B}$. While one might expect this to reduce the number of geometry steps needed to converge to the FOSP, this is not always the case.

For initial structures close to a FOSP (figure 5a), the more accurate $\mathbf{B}$ afforded by a smaller value of γ results in a decrease in the number of geometry steps required to reach convergence. However, this is counteracted by an increase in the number of gradient evaluations invoked by the eigensolver. For this system, the optimal choice of γ is close to one, as this results in the fewest number of gradient evaluations despite the larger number of geometry refinement steps required to converge to the FOSP. In contrast, refinement of poor quality initial structures (figure 5b) behaves much less consistently when γ is changed. If, during refinement, the leftmost eigenvalue of $\mathbf{H}_k$ becomes positive or close to zero, then standard Hessian updates from geometry refinement steps may result in $\mathbf{B}_k$ becoming positive definite. When this occurs, it becomes necessary to invoke the iterative diagonalization algorithm in order to verify that the leftmost eigenvector of $\mathbf{B}_k$ is accurate. Whether $\mathbf{B}_k$ becomes positive definite depends on the

accuracy of $\mathbf{B}_{k-1}$ and the previous trust radius δ_{k-1}. This explains why the number of gradient evaluations in figure 5b varies non-monotonically with γ across almost the entire range of values. Despite this, the optimal choice of γ is near one for this system as well.

Figure 6 shows how the choice of the Hessian update formula affects the quality of $\mathbf{B}_k$. For two systems taken from the LJ38 FOSP refinement benchmark (see table 1), we perform a full FOSP refinement. For these refinements, $\mathbf{B}_k$ was updated using the TS-BFGS update rule. Three additional approximate Hessians were maintained according to the PSB, SR1 (or MS), and BFGS updates (see section 2.3). These alternate approximate Hessians were not used as a preconditioner to the iterative diagonalization routines or during geometry refinement steps. Figures 6a and 6b plot the relative error in $\mathbf{B}_k$ for each of the four studied Hessian update methods as a function of number of gradient evaluations. Figures 6c and 6d plot the angle between the leftmost eigenvector of $\mathbf{B}_k$ and the leftmost eigenvector of $\mathbf{H}_k$ as a function of number of gradient evaluations. Figures 6a and 6c correspond to a good-quality initial structure, while figures 6b and 6d correspond to a poor-quality initial structure. All four figures begin after the initial call to the iterative eigensolver, which required 20 gradient evaluations for the good-quality initial structure and 50 gradient evaluations for the poor-quality initial structure. Grey regions correspond to gradient evaluations invoked by the iterative eigensolver.

Of the four Hessian update methods tested, only PSB and TS-BFGS choose $\mathbf{M}_k$ to be always positive definite. Figure 6 shows that SR1 and BFGS tend to accumulate very large errors in $\mathbf{B}_k$, both in terms of the matrix as a whole (figures 6a and 6b) and in terms of its leftmost eigenvector (figures 6c and 6d). While SR1 is capable of achieving a lower error in $\mathbf{B}_k$ (figure 6a), it behaves unreliably when $\mathbf{B}_k$ is allowed to become indefinite. PSB maintains a relatively low error in $\mathbf{B}_k$, but is less consistently able to accurately track the leftmost eigenvector of $\mathbf{H}_k$. Of the four methods tested, TS-BFGS is able to produce $\mathbf{B}_k$ with the most consistently low error and which most accurately approximates the leftmost eigenvector of $\mathbf{H}_k$. This is not to say that TS-BFGS is

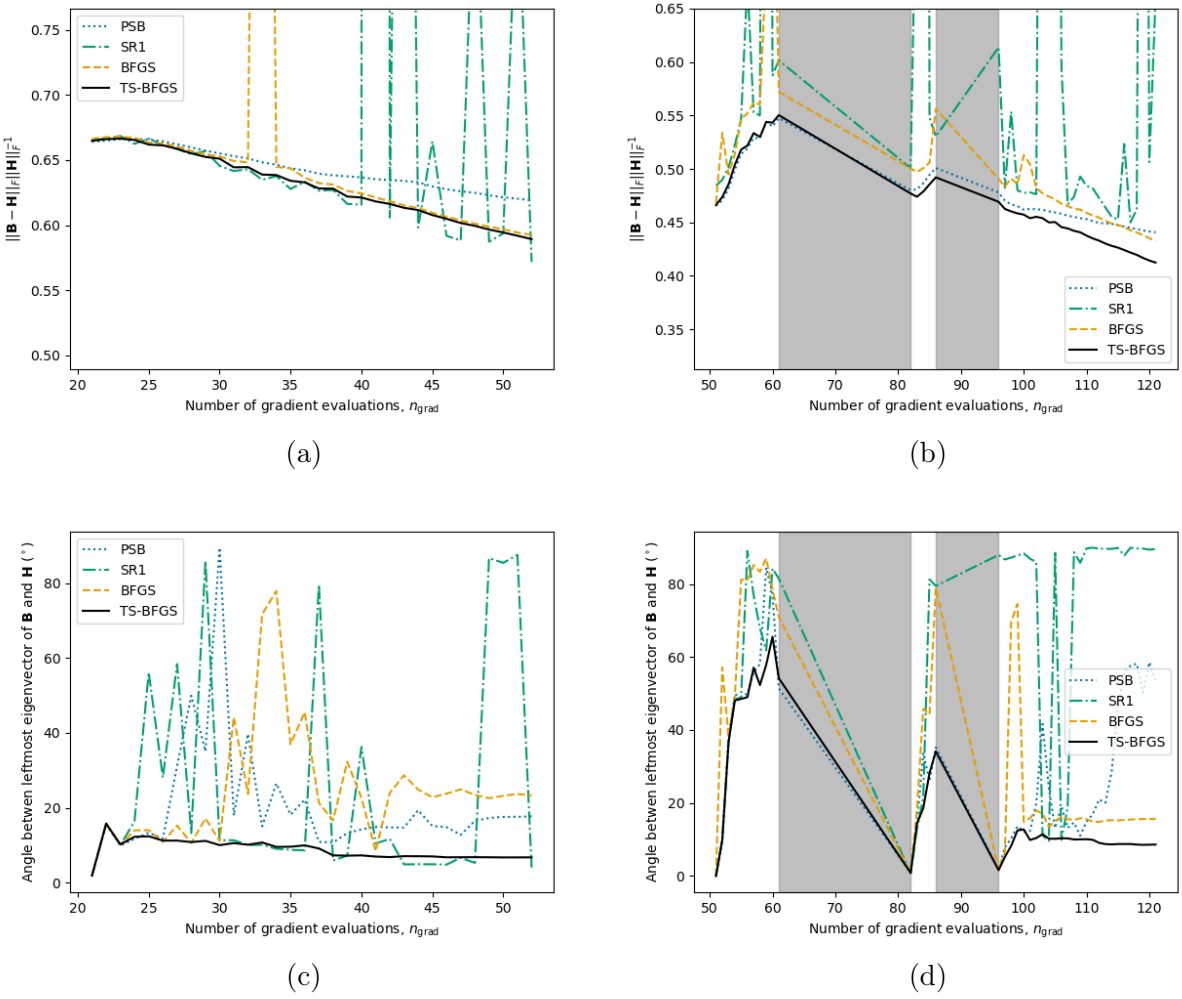


Figure 6: A comparison of various Hessian update schemes for a good initial structure ((a) and (c)) and a poor quality initial structure ((b) and (d)) using four different Hessian update schemes. Grey regions correspond to gradient evaluations invoked by the iterative diagonalization algorithm. (a), (b) A comparison of the error in $\mathbf{B}$ over the course of FOSP refinement. (c), (d) Angle between the leftmost eigenvector of $\mathbf{B}$ and $\mathbf{H}$ over the course of FOSP refinement. All other calculations in this work use the TS-BFGS update. Each plot begins after the first call to the iterative eigensolver returns, which requires 20 gradient evaluations for the good initial structure ((a) and (c)) and 50 gradient evaluations for the poor quality initial structure ((b) and (d)).

guaranteed to always track the leftmost eigenvector of $\mathbf{H}_k$ accurately (see figure 6d). Indeed, this is why it is occasionally necessary to invoke the iterative diagonalization algorithm partway through refinement.

We note that the errors in figures 6a and 6b do not approach zero even at the FOSP. This is to be expected, as $\mathbf{B}_k$ plays the role of a preconditioner for both the diagonalization algorithm and the geometry refinement steps. With the exception of its leftmost eigenvector, which must be somewhat close to the leftmost eigenvector of $\mathbf{H}_k$, $\mathbf{B}_k$ does not need to be highly accurate to guarantee *eventual* convergence to a FOSP. As is typical with preconditioners, convergence is achieved more rapidly when $\mathbf{B}_k$ more accurately approximates $\mathbf{H}_k$, but this must be balanced against the cost required to construct a more accurate $\mathbf{B}_k$.

While our approach is more reliable than some existing FOSP refinement methods, some pathological initial structures remain difficult to converge. Even if $\mathbf{B}_k$ is initially exact, it is possible to lose track of the leftmost eigenvalue of $\mathbf{H}_k$ using only standard secant updates. When this occurs, it is necessary to call the diagonalization algorithm again to correct $\mathbf{B}_k$. However, it is not simple to detect when $\mathbf{B}_k$ needs to be corrected. If it were possible to detect that the leftmost eigenvector of $\mathbf{B}_k$ is in poor agreement with the leftmost eigenvector of $\mathbf{H}_k$, it would be possible to determine when the diagonalization algorithm should be called. Currently, the diagonalization algorithm is only called during refinement if $\mathbf{B}_k$ becomes positive definite, but this may fail to correct $\mathbf{B}_k$ when, for example, $\mathbf{H}_k$ has multiple negative eigenvalues.

5 Conclusion

We present a novel approach for refining first order saddle point structures. In this approach, the Hessian $\mathbf{H}$ is partially diagonalized using an iterative diagonalization algorithm based on the Jacobi-Davidson method. In addition to providing an accurate approximation to the leftmost eigenvector of $\mathbf{H}$, the curvature information obtained by the diagonalization algorithm is used to construct an approximate Hessian $\mathbf{B}$. $\mathbf{B}$ is used

both to determine geometry refinement steps using RS-PRFO and as a preconditioner for any subsequent calls to the iterative diagonalization algorithm.

Our approach is suitable for refining first order saddle point geometries of large systems with many atoms for which it is unfeasible to calculate $\mathbf{H}$ even a single time. The number of steps performed by the diagonalization algorithm and the accuracy of $\mathbf{B}$ can be controlled by a single tunable convergence parameter. In the limit of very tight convergence, our method becomes equivalent to full diagonalization of $\mathbf{H}$ using finite difference. This means our method is also applicable to smaller systems which may benefit from full knowledge of $\mathbf{H}$.

Our method requires on average 50 % fewer gradient evaluations to converge to first order saddle point structures on one saddle point refinement benchmark from optbench.org relative to the current best performing codes, Optim and Pele.[80] On another saddle point refinement benchmark, Sella requires on average 25 % fewer gradient evaluations to converge relative to Optim and Pele. In addition, our preconditioned iterative diagonalization routine converges to the leftmost eigenvector of $\mathbf{H}$ with a high degree of accuracy in just over half the number of gradient evaluations required by Lanczos. Even in the absence of a preconditioner, the iterative diagonalization routine used in our method requires on average almost 30 % fewer gradient evaluations to converge relative to Optim and Pele.

We provide evidence that the observed increase in performance for first order saddle point refinement is a result of an improvement in the accuracy of the approximate Hessian $\mathbf{B}$. Our key insight is that the information provided by the iterative diagonalization routine can be used to construct a highly accurate $\mathbf{B}$, or to update an existing $\mathbf{B}$ to be more accurate. Practically, this innovation results in a $\mathbf{B}$ that is potentially indefinite, precluding the use of standard minimum mode following approaches that require a positive definite $\mathbf{B}$. However, this does not pose a problem if RS-PRFO is used to determine geometry refinement steps, an approach which is used extensively for the refinement of first order saddle points on molecular potential energy surfaces.

There is still significant room for improvement on the method we describe. Partic-

ularly, our method is likely to underperform for molecules relative to methods implemented in leading electronic structure theory packages. This is because our method represents the molecular geometry in Cartesian coordinates, rather than using internal coordinates. We intend to extend our method to other representations including internal coordinates in the near future.

Acknowledgement

This work was supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Chemical Sciences, Geosciences and Biosciences Division, as part of the Computational Chemistry Sciences Program (Award Number: 0000232253).

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Supporting Information Available

Python scripts for generating the data in all tables and figures 4, 5, and 6 using Sella are available in the SI.

References

(1) Dewyer, A. L.; Zimmerman, P. M. Finding reaction mechanisms, intuitive or otherwise. *Organic & Biomolecular Chemistry* **2017**, *15*, 501–504.

(2) Dewyer, A. L.; Arguelles, A. J.; Zimmerman, P. M. Methods for exploring reac-

tion space in molecular systems. *Wiley Interdisciplinary Reviews-Computational Molecular Science* **2018**, *8*.

(3) Simm, G. N.; Vaucher, A. C.; Reiher, M. Exploration of Reaction Pathways and Chemical Transformation Networks. *The Journal of Physical Chemistry A* **2019**, *123*, 385–399.

(4) Zimmerman, P. M. Automated discovery of chemically reasonable elementary reaction steps. *Journal of Computational Chemistry* **2013**, *34*, 1385–1392.

(5) Jafari, M.; Zimmerman, P. M. Uncovering reaction sequences on surfaces through graphical methods. *Physical Chemistry Chemical Physics* **2018**, *20*, 7721–7729.

(6) Suleimanov, Y. V.; Green, W. H. Automated Discovery of Elementary Chemical Reaction Steps Using Freezing String and Berny Optimization Methods. *Journal of Chemical Theory and Computation* **2015**, *11*, 4248–4259.

(7) Maeda, S.; Harabuchi, Y.; Takagi, M.; Taketsugu, T.; Morokuma, K. Artificial Force Induced Reaction (AFIR) Method for Exploring Quantum Chemical Potential Energy Surfaces. *The Chemical Record* **2016**, *16*, 2232–2248.

(8) Maeda, S.; Taketsugu, T.; Morokuma, K. Exploring transition state structures for intramolecular pathways by the artificial force induced reaction method. *Journal of Computational Chemistry* **2014**, *35*, 166–173.

(9) Bhoorasingh, P. L.; West, R. H. Transition state geometry prediction using molecular group contributions. *Physical Chemistry Chemical Physics* **2015**, *17*, 32173–32182.

(10) Bhoorasingh, P. L.; Slakman, B. L.; Seyedzadeh Khanshan, F.; Cain, J. Y.; West, R. H. Automated Transition State Theory Calculations for High-Throughput Kinetics. *The Journal of Physical Chemistry A* **2017**, *121*, 6896–6904.

(11) Van de Vijver, R.; Van Geem, K. M.; Marin, G. B. On-the-fly ab initio calculations toward accurate rate coefficients. *Proceedings of the Combustion Institute* **2018**,

(12) Kim, Y.; Kim, J. W.; Kim, Z.; Kim, W. Y. Efficient prediction of reaction paths through molecular graph and reaction network analysis. *Chemical Science* **2018**, *9*, 825–835.

(13) Yang, M.; Yang, L.; Wang, G.; Zhou, Y.; Xie, D.; Li, S. Combined Molecular Dynamics and Coordinate Driving Method for Automatic Reaction Pathway Search of Reactions in Solution. *Journal of Chemical Theory and Computation* **2018**, *14*, 5787–5796.

(14) Grambow, C. A.; Jamal, A.; Li, Y.-P.; Green, W. H.; Zádor, J.; Suleimanov, Y. V. Unimolecular Reaction Pathways of a γ-Ketohydroperoxide from Combined Application of Automated Reaction Discovery Methods. *Journal of the American Chemical Society* **2018**, *140*, 1035–1048.

(15) Cavallotti, C.; Pelucchi, M.; Georgievskii, Y.; Klippenstein, S. J. EStokTP: Electronic Structure to Temperature- and Pressure-Dependent Rate Constants—A Code for Automatically Predicting the Thermal Kinetics of Reactions. *Journal of Chemical Theory and Computation* **2019**, *15*, 1122–1145.

(16) Martínez-Núñez, E. An automated transition state search using classical trajectories initialized at multiple minima. *Physical Chemistry Chemical Physics* **2015**, *17*, 14912–14921.

(17) Martínez-Núñez, E. An automated method to find transition states using chemical dynamics simulations. *Journal of Computational Chemistry* **2015**, *36*, 222–234.

(18) Varela, J. A.; Vázquez, S. A.; Martínez-Núñez, E. An automated method to find reaction mechanisms and solve the kinetics in organometallic catalysis. *Chemical Science* **2017**, *8*, 3843–3851.

(19) Jónsson, H.; Mills, G.; Jacobsen, K. W. *Classical and Quantum Dynamics in Condensed Phase Simulations*; WORLD SCIENTIFIC, 1998; pp 385–404.

(20) Henkelman, G.; Jónsson, H. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. *The Journal of Chemical Physics* **2000**, *113*, 9978–9985.

(21) Henkelman, G.; Uberuaga, B. P.; Jónsson, H. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. *The Journal of Chemical Physics* **2000**, *113*, 9901–9904.

(22) Trygubenko, S. A.; Wales, D. J. A doubly nudged elastic band method for finding transition states. *The Journal of Chemical Physics* **2004**, *120*, 2082–2094.

(23) Sheppard, D.; Terrell, R.; Henkelman, G. Optimization methods for finding minimum energy paths. *The Journal of Chemical Physics* **2008**, *128*, 134106.

(24) Sheppard, D.; Xiao, P.; Chemelewski, W.; Johnson, D. D.; Henkelman, G. A generalized solid-state nudged elastic band method. *The Journal of Chemical Physics* **2012**, *136*, 074103.

(25) Halgren, T. A.; Lipscomb, W. N. The synchronous-transit method for determining reaction pathways and locating molecular transition states. *Chemical Physics Letters* **1977**, *49*, 225–232.

(26) Peng, C.; Schlegel, H. B. Combining Synchronous Transit and Quasi-Newton Methods to Find Transition States. *Israel Journal of Chemistry* **1993**, *33*, 449–454.

(27) Govind, N.; Petersen, M.; Fitzgerald, G.; King-Smith, D.; Andzelm, J. A generalized synchronous transit method for transition state location. *Computational Materials Science* **2003**, *28*, 250–258.

(28) Peters, B.; Heyden, A.; Bell, A. T.; Chakraborty, A. A growing string method for determining transition states: Comparison to the nudged elastic band and string methods. *The Journal of Chemical Physics* **2004**, *120*, 7877–7886.

(29) Quapp, W. A growing string method for the reaction pathway defined by a Newton trajectory. *The Journal of Chemical Physics* **2005**, *122*, 174106.

(30) Behn, A.; Zimmerman, P. M.; Bell, A. T.; Head-Gordon, M. Incorporating Linear Synchronous Transit Interpolation into the Growing String Method: Algorithm and Applications. *Journal of Chemical Theory and Computation* **2011**, *7*, 4019–4025.

(31) Zimmerman, P. M. Growing string method with interpolation and optimization in internal coordinates: Method and examples. *The Journal of Chemical Physics* **2013**, *138*, 184102.

(32) Zimmerman, P. Reliable Transition State Searches Integrated with the Growing String Method. *Journal of Chemical Theory and Computation* **2013**, *9*, 3043–3050.

(33) Malek, R.; Mousseau, N. Dynamics of Lennard-Jones clusters: A characterization of the activation-relaxation technique. *Phys. Rev. E* **2000**, *62*, 7723–7728.

(34) Van de Vijver, R.; Zádor, J. KinBot: Automated stationary point search on potential energy surfaces. *Computer Physics Communications* **2019**, Accepted.

(35) Henkelman, G.; Jónsson, H. A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives. *J. Chem. Phys.* **1999**, *111*, 7010–7022.

(36) Heyden, A.; Bell, A. T.; Keil, F. J. Efficient methods for finding transition states in chemical reactions: Comparison of improved dimer method and partitioned rational function optimization method. *The Journal of Chemical Physics* **2005**, *123*, 224101.

(37) Kästner, J.; Sherwood, P. Superlinearly converging dimer method for transition state search. *The Journal of Chemical Physics* **2008**, *128*, 014106.

(38) Xiao, P.; Wu, Q.; Henkelman, G. Basin constrained κ-dimer method for saddle point finding. *The Journal of Chemical Physics* **2014**, *141*, 164111.

(39) Zeng, Y.; Xiao, P.; Henkelman, G. Unification of algorithms for minimum mode optimization. *J. Chem. Phys.* **2014**, *140*, 044115.

(40) Saad, Y.; Lehoucq, R.; Sorensen, D. *Templates for the Solution of Algebraic Eigenvalue Problems*; Software, Environments and Tools; Society for Industrial and Applied Mathematics, 2000; pp 37–44.

(41) Knyazev, A. Toward the Optimal Preconditioned Eigensolver: Locally Optimal Block Preconditioned Conjugate Gradient Method. *SIAM Journal on Scientific Computing* **2001**, *23*, 517–541.

(42) Gu, M.; Ruhe, A.; Lehoucq, R.; Sorensen, D.; Freund, R.; Sleijpen, G.; van der Vorst, H.; Bai, Z.; Li, R. *Templates for the Solution of Algebraic Eigenvalue Problems*; Software, Environments and Tools; Society for Industrial and Applied Mathematics, 2000; pp 45–107.

(43) Morgan, R.; Scott, D. Generalizations of Davidson's Method for Computing Eigenvalues of Sparse Symmetric Matrices. *SIAM Journal on Scientific and Statistical Computing* **1986**, *7*, 817–825.

(44) G. Sleijpen, G.; Van der Vorst, H. A Jacobi–Davidson Iteration Method for Linear Eigenvalue Problems. *SIAM Journal on Matrix Analysis and Applications* **1996**, *17*, 401–425.

(45) Davidson, E. R. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics* **1975**, *17*, 87–94.

(46) Reiher, M.; Neugebauer, J. Convergence characteristics and efficiency of mode-tracking calculations on pre-selected molecular vibrations. *Physical Chemistry Chemical Physics* **2004**, *6*, 4621–4629.

(47) Olsen, J.; Jørgensen, P.; Simons, J. Passing the one-billion limit in full

configuration-interaction (FCI) calculations. *Chemical Physics Letters* **1990**, *169*, 463–472.

(48) Stathopoulos, A.; Saad, Y.; Fischer, C. F. Robust preconditioning of large, sparse, symmetric eigenvalue problems. *Journal of Computational and Applied Mathematics* **1995**, *64*, 197–215.

(49) Sleijpen, G. L.; Van der Vorst, H. A. The Jacobi-Davidson method for eigenvalue problems as an accelerated inexact Newton scheme. IMACS Conference proceedings. 1995.

(50) Sleijpen, G. L. G.; Wubs, F. W. Effective preconditioning techniques for eigenvalue problems. *Preprint 1117* **1999**,

(51) Stathopoulos, A. Nearly Optimal Preconditioned Methods for Hermitian Eigenproblems under Limited Memory. Part I: Seeking One Eigenvalue. *SIAM Journal on Scientific Computing* **2007**, *29*, 481–514.

(52) Schlegel, H. B. Optimization of equilibrium geometries and transition structures. *J. Comput. Chem.* **1982**, *3*, 214–218.

(53) Olsen, R. A.; Kroes, G. J.; Henkelman, G.; Arnaldsson, A.; Jónsson, H. Comparison of methods for finding saddle points without knowledge of the final states. *The Journal of Chemical Physics* **2004**, *121*, 9776–9792.

(54) Mei, D.; Xu, L.; Henkelman, G. Dimer saddle point searches to determine the reactivity of formate on Cu(111). *Journal of Catalysis* **2008**, *258*, 44–51.

(55) Chia, M.; Pagán-Torres, Y. J.; Hibbitts, D.; Tan, Q.; Pham, H. N.; Datye, A. K.; Neurock, M.; Davis, R. J.; Dumesic, J. A. Selective Hydrogenolysis of Polyols and Cyclic Ethers over Bifunctional Surface Sites on Rhodium–Rhenium Catalysts. *Journal of the American Chemical Society* **2011**, *133*, 12675–12689.

(56) Zhao, Y.-F.; Yang, Y.; Mims, C.; Peden, C. H. F.; Li, J.; Mei, D. Insight into methanol synthesis from CO2 hydrogenation on Cu(111): Complex reaction network and the effects of H2O. *Journal of Catalysis* **2011**, *281*, 199–211.

(57) Banerjee, A.; Adams, N.; Simons, J.; Shepard, R. Search for stationary points on surfaces. *J. Phys. Chem.* **1985**, *89*, 52–57.

(58) Anglada, J. M.; Bofill, J. M. A reduced-restricted-quasi-Newton–Raphson method for locating and optimizing energy crossing points between two potential energy surfaces. *J. Comp. Chem.* **1997**, *18*, 992–1003.

(59) Besalú, E.; Bofill, J. M. On the automatic restricted-step rational-function-optimization method. *Theoretical Chemistry Accounts* **1998**, *100*, 265–274.

(60) Greenstadt, J. Variations on variable-metric methods. (With discussion). *Mathematics of Computation* **1970**, *24*, 1–22.

(61) Broyden, C. G. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics* **1970**, *6*, 76–90.

(62) Fletcher, R. A new approach to variable metric algorithms. *The Computer Journal* **1970**, *13*, 317–322.

(63) Goldfarb, D. A family of variable-metric methods derived by variational means. *Mathematics of Computation* **1970**, *24*, 23–26.

(64) Shanno, D. F. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation* **1970**, *24*, 647–656.

(65) Murtagh, B. A.; Sargent, R. W. H. Computational experience with quadratically convergent minimisation methods. *The Computer Journal* **1970**, *13*, 185–194.

(66) Bofill, J. M. Updated Hessian matrix and the restricted step method for locating transition structures. *Journal of Computational Chemistry* **1994**, *15*, 1–11.

(67) Bofill, J. M.; Comajuan, M. Analysis of the updated Hessian matrices for locating transition structures. *Journal of Computational Chemistry* **1995**, *16*, 1326–1338.

(68) Bofill, J. M. Remarks on the updated Hessian matrix methods. *Int. J. Quantum Chem.* **2003**, *94*, 324–332.

(69) Farkas, O.; Schlegel, H. B. Methods for optimizing large molecules. II. Quadratic search. *The Journal of Chemical Physics* **1999**, *111*, 10806–10814.

(70) Anglada, J. M.; Bofill, J. M. How good is a Broyden–Fletcher–Goldfarb–Shanno-like update Hessian formula to locate transition structures? Specific reformulation of Broyden–Fletcher–Goldfarb–Shanno for optimizing saddle points. *Journal of Computational Chemistry* **1998**, *19*, 349–362.

(71) Gower, R. M.; Gondzio, J. Action constrained quasi-Newton methods. *arXiv:1412.8045 [cs, math]* **2014**, arXiv: 1412.8045.

(72) Schnabel, R. B. Quasi-Newton Methods Using Multiple Secant Equations ; CU-CS-247-83. *Computer Science Technical Reports* **1983**, *224*, 1–40.

(73) Hermes, E. Sella. 2019; `https://doi.org/10.5281/zenodo.3379094`.

(74) Valiev, M.; Bylaska, E. J.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Van Dam, H. J. J.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T. L.; de Jong, W. A. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications* **2010**, *181*, 1477–1489.

(75) Giannozzi, P.; Baroni, S.; Bonini, N.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Chiarotti, G. L.; Cococcioni, M.; Dabo, I.; Corso, A. D.; Gironcoli, S. d.; Fabris, S.; Fratesi, G.; Gebauer, R.; Gerstmann, U.; Gougoussis, C.; Kokalj, A.; Lazzeri, M.; Martin-Samos, L.; Marzari, N.; Mauri, F.; Mazzarello, R.; Paolini, S.; Pasquarello, A.; Paulatto, L.; Sbraccia, C.; Scandolo, S.; Sclauzero, G.; Seitsonen, A. P.; Smogunov, A.; Umari, P.; Wentzcovitch, R. M. QUANTUM

ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter* **2009**, *21*, 395502.

(76) Giannozzi, P.; Andreussi, O.; Brumme, T.; Bunau, O.; Nardelli, M. B.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Cococcioni, M.; Colonna, N.; Carnimeo, I.; Corso, A. D.; Gironcoli, S. d.; Delugas, P.; DiStasio, R. A.; Ferretti, A.; Floris, A.; Fratesi, G.; Fugallo, G.; Gebauer, R.; Gerstmann, U.; Giustino, F.; Gorni, T.; Jia, J.; Kawamura, M.; Ko, H.-Y.; Kokalj, A.; Küçükbenli, E.; Lazzeri, M.; Marsili, M.; Marzari, N.; Mauri, F.; Nguyen, N. L.; Nguyen, H.-V.; Otero-de-la Roza, A.; Paulatto, L.; Poncé, S.; Rocca, D.; Sabatini, R.; Santra, B.; Schlipf, M.; Seitsonen, A. P.; Smogunov, A.; Timrov, I.; Thonhauser, T.; Umari, P.; Vast, N.; Wu, X.; Baroni, S. Advanced capabilities for materials modelling with Quantum ESPRESSO. *Journal of Physics: Condensed Matter* **2017**, *29*, 465901.

(77) Hutter, J.; Iannuzzi, M.; Schiffmann, F.; VandeVondele, J. cp2k: atomistic simulations of condensed matter systems. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, *4*, 15–25.

(78) Enkovaara, J.; Rostgaard, C.; Mortensen, J. J.; Chen, J.; Du\lak, M.; Ferrighi, L.; Gavnholt, J.; Glinsvad, C.; Haikola, V.; Hansen, H. A.; Kristoffersen, H. H.; Kuisma, M.; Larsen, A. H.; Lehtovaara, L.; Ljungberg, M.; Lopez-Acevedo, O.; Moses, P. G.; Ojanen, J.; Olsen, T.; Petzold, V.; Romero, N. A.; Stausholm-Møller, J.; Strange, M.; Tritsaris, G. A.; Vanin, M.; Walter, M.; Hammer, B.; Häkkinen, H.; Madsen, G. K. H.; Nieminen, R. M.; Nørskov, J. K.; Puska, M.; Rantala, T. T.; Schiøtz, J.; Thygesen, K. S.; Jacobsen, K. W. Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method. *Journal of Physics: Condensed Matter* **2010**, *22*, 253202.

(79) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.;

Jennings, P. C.; Jensen, P. B.; Kermode, J.; Kitchin, J. R.; Kolsbjerg, E. L.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Maronsson, J. B.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W. The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter* **2017**, *29*, 273002.

(80) Chill, S. T.; Stevenson, J.; Ruehle, V.; Shang, C.; Xiao, P.; Farrell, J. D.; Wales, D. J.; Henkelman, G. Benchmarks for Characterization of Minima, Transition States, and Pathways in Atomic, Molecular, and Condensed Matter Systems. *J. Chem. Theory Comput.* **2014**, *10*, 5476–5482.

(81) Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics* **1995**, *117*, 1–19.

(82) OPTIM: A Program for Optimizing Geometries and Calculating Reaction Pathways. 2019; `http://www-wales.ch.cam.ac.uk/OPTIM`.

(83) Python energy landscape explorer. 2019; `https://github.com/pele-python/pele`.