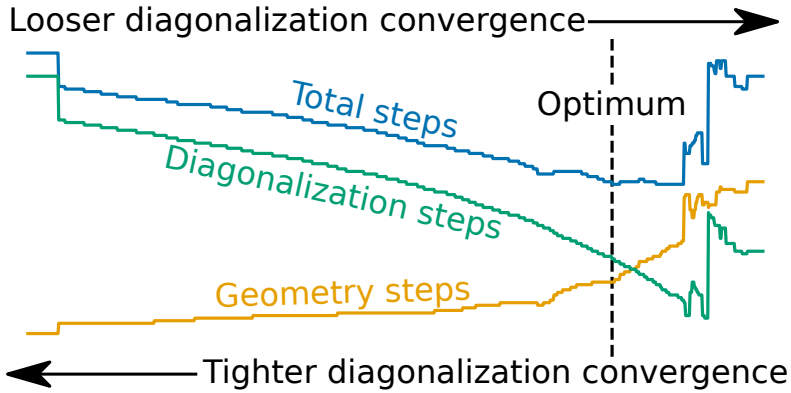
Exceptional service in the national interest

Sandia National Laboratories



Seeking Saddle Points with Sella

Eric D. Hermes

2019-11-19

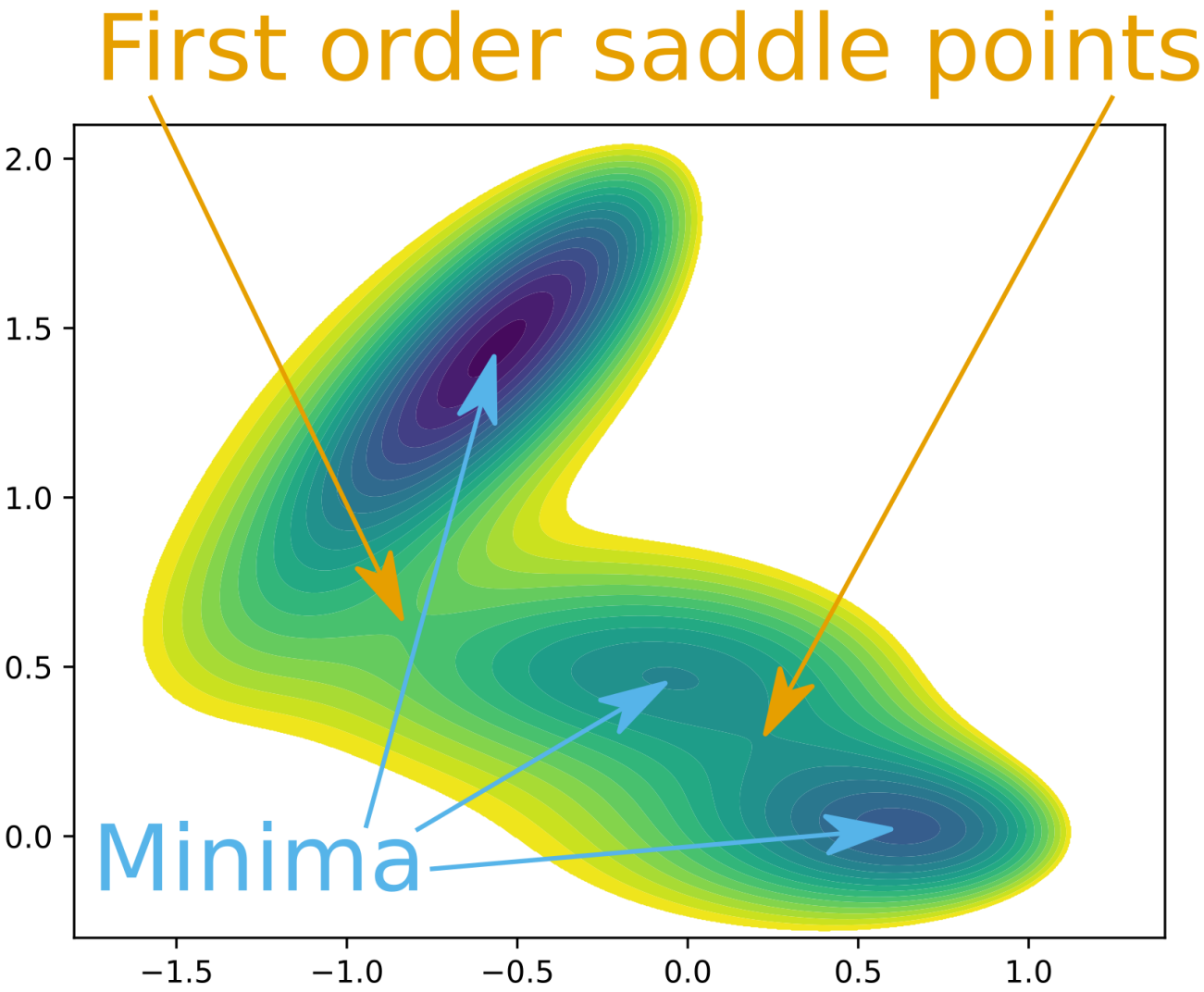
U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

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. SAND NO. 2019-14022 C

Sandia National Laboratories

What is Sella?

Sella is a tool for finding first order saddle points.



How do I use Sella?

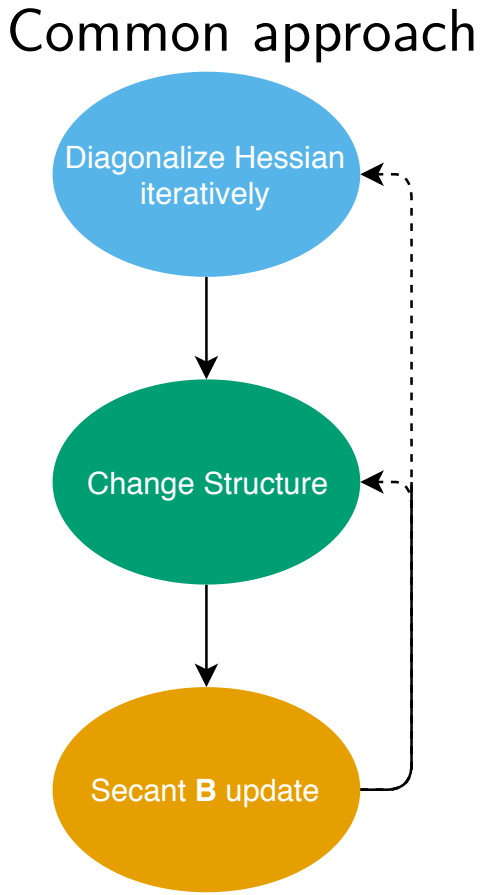
Sandia National Laboratories

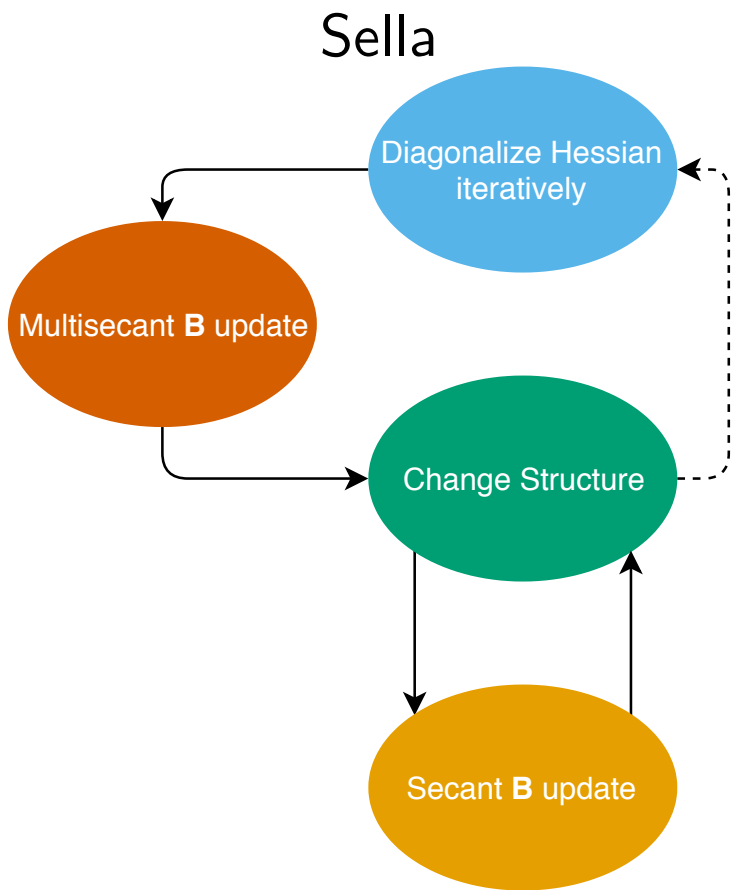
Install Sella with pip: `pip3 install sella --user`, or download from github: `https://github.com/zadorlab/sella`.

To use Sella:

```
1  from ase.io import read
2  from ase.calculators.somecalc import SomeCalc
3
4  from sella import Sella
5
6  atoms = read('my_atoms_geometry.xyz')
7  atoms.calc = SomeCalc(...)
8  opt = Sella(atoms)
9  opt.run(fmax=0.01)
```

How does Sella work?





Sandia National Laboratories

Introduction

Theory

- Geometry optimization
- Iterative Hessian diagonalization
- Approximate Hessian updates

Results

Wrap up

Newton-Raphson

Expand the energy ϵ in a 2nd order Taylor series around $\mathbf{q}$:

$$\tilde{\epsilon}(\mathbf{q}+\mathbf{s}) = \epsilon(\mathbf{q}) + \mathbf{s}^T\mathbf{g} + \frac{1}{2}\mathbf{s}^T\mathbf{H}\mathbf{s}$$

Solve for $\frac{\partial\tilde{\epsilon}}{\partial\mathbf{s}} = \mathbf{0}$:

$$\mathbf{s} = -\mathbf{H}^{-1}\mathbf{g}$$

Problems:

- We don't want to calculate $\mathbf{H}$
  - Replace it with an approximation $\mathbf{B}$ (Quasi-Newton)
- $\mathbf{s}$ may be far too big
  - Use a linesearch or a trust region method
- This can converge to a minimum *or* a saddle point

Rational function optimization

RFO replaces the Taylor series with a rational approximation:

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

The stationary points of this function can be found by solving a generalized eigenvalue problem:

$$\begin{bmatrix} \mathbf{B} & \mathbf{g} \\ \mathbf{g}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix} = 2\mu \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}$$

To find a *minimum*, determine $\mathbf{s}$ from the leftmost eigenvector.

Partitioned rational function optimization

P-RFO splits the eigenvalue problem into two eigenvalue problems:

$$\begin{bmatrix} \mathbf{V}^{(\max)T}\mathbf{B}\mathbf{V}^{(\max)} & \mathbf{V}^{(\max)T}\mathbf{g} \\ \mathbf{g}^T\mathbf{V}^{(\max)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}^{(\max)} \\ 1 \end{bmatrix} = 2\mu^{(\max)} \begin{bmatrix} \mathbf{W}^{(\max)} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{s}^{(\max)} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{V}^{(\min)T}\mathbf{B}\mathbf{V}^{(\min)} & \mathbf{V}^{(\min)T}\mathbf{g} \\ \mathbf{g}^T\mathbf{V}^{(\min)} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}^{(\min)} \\ 1 \end{bmatrix} = 2\mu^{(\min)} \begin{bmatrix} \mathbf{W}^{(\min)} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{s}^{(\min)} \\ 1 \end{bmatrix}$$

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

To find a *first order saddle point*, $\mathbf{V}^{(\max)}$ should contain only the leftmost eigenvector of $\mathbf{B}$, and $\mathbf{V}^{(\min)}$ should contain all other eigenvectors.

For this approach to work, the leftmost eigenvector of $\mathbf{B}$ must be close to the leftmost eigenvector of $\mathbf{H}$!

Iterative Hessian diagonalization

For P-RFO to perform well, the leftmost eigenvector of **B** should closely match the leftmost eigenvector of **H**.

But how can we determine the leftmost eigenvector of **H**?

Iterative diagonalization with Rayleigh-Ritz requires only Hessian-vector products (e.g. **Hs**), which can be approximated with finite difference:

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

Practical Hessian-vector products

```python
import numpy as np
from scipy.sparse.linalg import LinearOperator

class NumericalHessian(LinearOperator):
    dtype = np.dtype('float64')
    def __init__(self, atoms, eta=1e-4):
        self.atoms = atoms
        n = 3 * len(self.atoms)
        self.shape = (n, n)
        self.eta = eta
        self.x0 = self.atoms.positions.copy()
        self.g0 = -self.atoms.get_forces().ravel()
    def _matvec(self, v):
        self.atoms.positions = self.x0 + self.eta * v.reshape((-1, 3))
        gplus = -self.atoms.get_forces().ravel()
        self.atoms.positions = self.x0.copy()
        return (gplus - self.g0) / self.eta
```

Diagonalization algorithm

```python
import numpy as np
from scipy.linalg import eigh
from sella.utilities.math import modified_gram_schmidt

def rayleigh_ritz(H, gamma, B):
    S = eigh(B)[1][:, :1]  # start w/ leftmost eigenvector of B
    Y = H.dot(S)
    for _ in range(n - 1):
        vals, vecs = eigh(Y.T @ S)
        x = S @ vecs[:, 0]
        Hx = Y @ vecs[:, 0]
        r = Hx - vals[0] * x
        if np.linalg.norm(r) < gamma * np.abs(vals[0]):
            return S @ vecs, Y @ vecs
        S = np.hstack((S, modified_gram_schmidt(jd0(x, r, vals[0], B), S)))
        Y = np.hstack((Y, H.dot(S[:, -1])))
```

Hessian updates

Given an approximate Hessian $\mathbf{B}$, a displacement vector $\mathbf{s} = \mathbf{q}^+ - \mathbf{q}$, and the change in the gradient vector $\mathbf{y} = \mathbf{g}^+ - \mathbf{g}$, find $\mathbf{B}^+ = \mathbf{B} + \mathbf{E}$ such that:

- $\mathbf{B}^+\mathbf{s} = \mathbf{y}$
- $\mathbf{E}^T = \mathbf{E}$
- $\|\mathbf{E}\|^2_{\mathbf{M}^{-1}} = \mathrm{Tr}\left[\mathbf{E}\mathbf{M}^{-1}\mathbf{E}^T\mathbf{M}^{-1}\right]$ is minimized

General solution for any symmetric positive definite $\mathbf{M}$:

$$\mathbf{B}^+ = \mathbf{B} + \begin{bmatrix}\mathbf{j} & \mathbf{u}\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & -\mathbf{j}^T\mathbf{s}\end{bmatrix}\begin{bmatrix}\mathbf{j}^T\\ \mathbf{u}^T\end{bmatrix}$$

$$\mathbf{j} = \mathbf{y} - \mathbf{B}\mathbf{s}$$

$$\mathbf{u} = \mathbf{M}\mathbf{s}\left[\mathbf{s}^T\mathbf{M}\mathbf{s}\right]^{-1}$$

Multisecant Hessian updates

Given an approximate Hessian $\mathbf{B}$, a matrix of displacement vectors $\mathbf{S}$, and the Hessian-vector products $\mathbf{Y} = \mathbf{HS}$, find $\mathbf{B}^+ = \mathbf{B} + \mathbf{E}$ such that:

- $\mathbf{B}^+\mathbf{S} = \mathbf{Y}$
- $\mathbf{E}^T = \mathbf{E}$
- $\|\mathbf{E}\|^2_{\mathbf{M}^{-1}} = \mathrm{Tr}\left[\mathbf{E}\mathbf{M}^{-1}\mathbf{E}^T\mathbf{M}^{-1}\right]$ is minimized

General solution for any symmetric positive definite $\mathbf{M}$:

$$\mathbf{B}^+ = \mathbf{B} + \begin{bmatrix}\mathbf{J} & \mathbf{U}\end{bmatrix}\begin{bmatrix}\mathbf{0} & \mathbf{I} \\ \mathbf{I} & -\mathbf{J}^T\mathbf{S}\end{bmatrix}\begin{bmatrix}\mathbf{J}^T \\ \mathbf{U}^T\end{bmatrix}$$

$$\mathbf{J} = \mathbf{Y} - \mathbf{BS}$$

$$\mathbf{U} = \mathbf{MS}\left[\mathbf{S}^T\mathbf{MS}\right]^{-1}$$

Sandia National Laboratories

Sella paper

JCTC Journal of Chemical Theory and Computation

Article

Cite This: J. Chem. Theory Comput. 2019, 15, 6536–6549

pubs.acs.org/JCTC

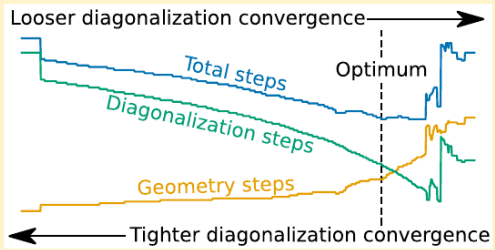
Accelerated Saddle Point Refinement through Full Exploitation of Partial Hessian Diagonalization

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

Combustion Research Facility, Sandia National Laboratories, Livermore, California 94551-0969, United States

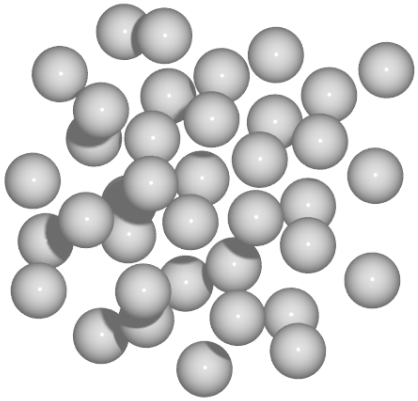
Supporting Information

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 for modestly sized molecular systems 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.

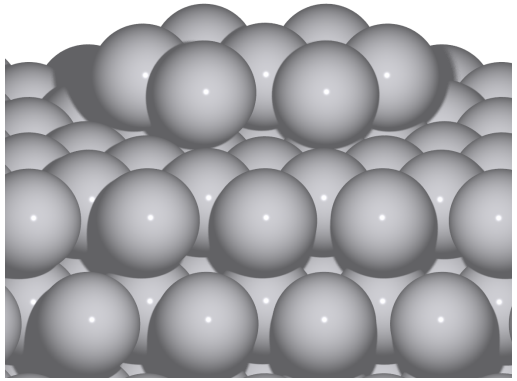


DOI: 10.1021/acs.jctc.9b00869

Optbench.org benchmarks



Code	mean	min	max
Sella	70	24	159
Optim	145	57	565
Pele	192	59	1488

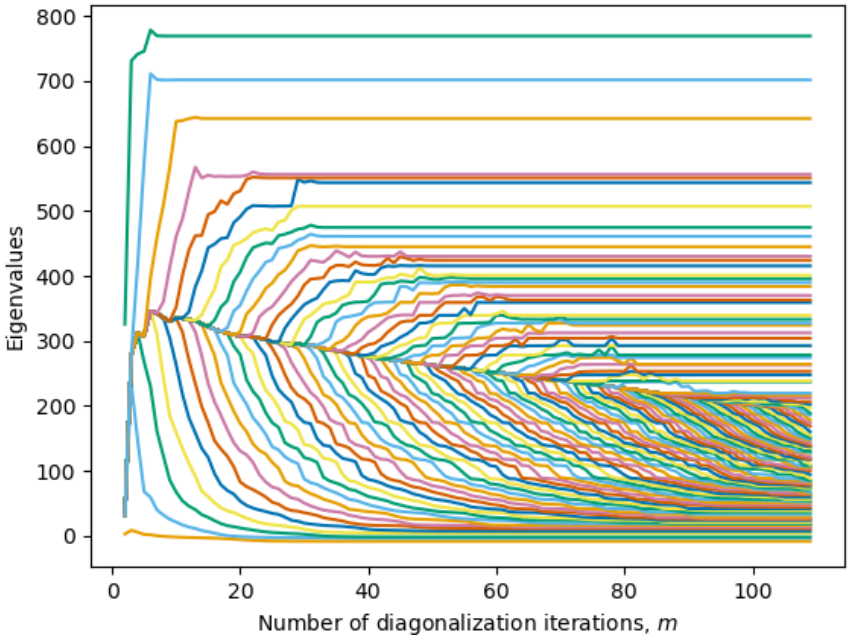


Code	mean	min	max
Sella	53	31	108
Optim	71	43	143
Pele	88	52	198

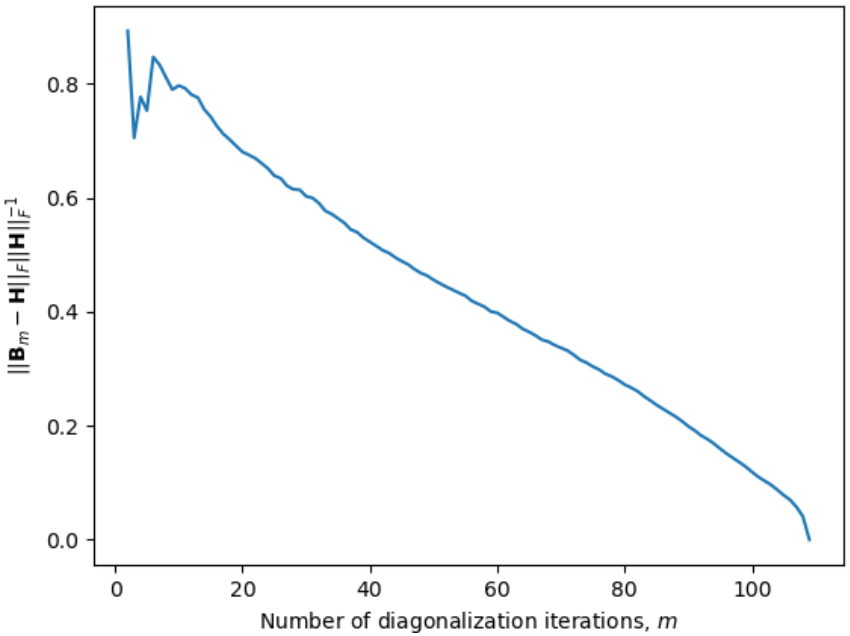
DOI: 10.1021/acs.jctc.9b00869

Accuracy of the approximate Hessian

Approx. Hessian eigenvalues



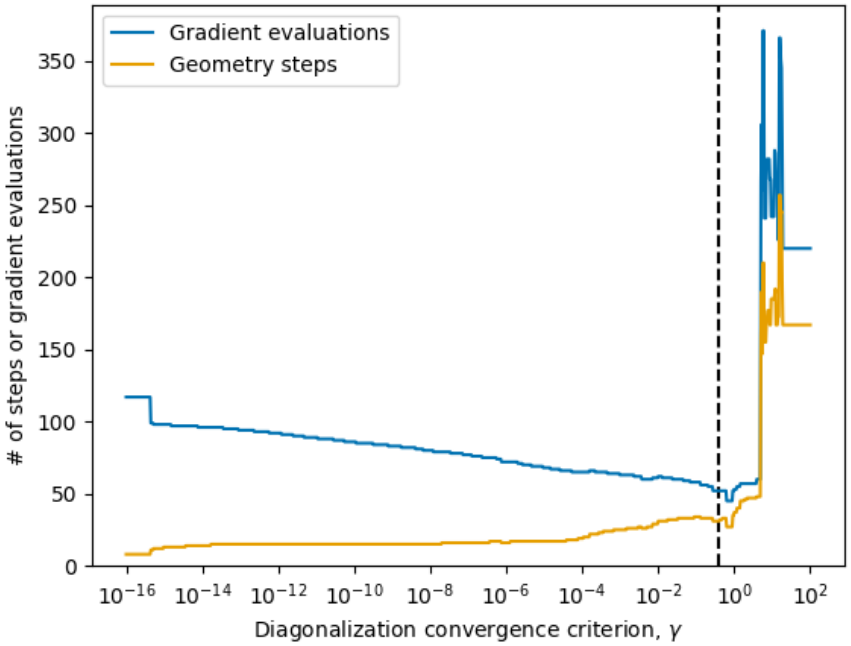
Error in approx. Hessian



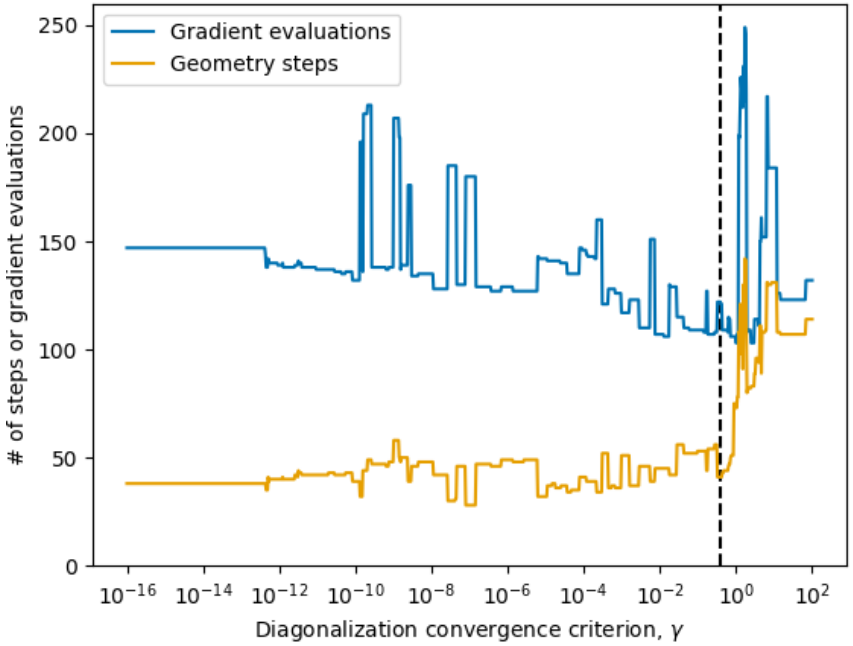
DOI: 10.1021/acs.jctc.9b00869

How eigensolver convergence affects performance

Starting close to the saddle point



Starting far from the saddle point



DOI: 10.1021/acs.jctc.9b00869

Sandia National Laboratories

Introduction

Theory

- Geometry optimization
- Iterative Hessian diagonalization
- Approximate Hessian updates

Results

Wrap up

Other features and future work

Other features of Sella:

- Minimization (though with somewhat poor performance)
- Intrinsic Reaction Coordinate (IRC) calculations
- Constrained optimization (minima and saddle points)
  - Fixed atom, bond distance, bending angle, and dihedral angle constraints currently implemented
  - Translational/rotational modes are excluded during optimization using constraints

Future work and work in progress

- Implement internal coordinate optimization
- Improve minimization performance
- Implement surrogate-accelerated optimization (e.g. using Gaussian process regression)

Funding

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).
The views expressed in this presentation do not necessarily represent the views of the U.S. Department of Energy or the United States Government.