

# Benchmarks for Characterization of Minima, Transition States, and Pathways in Atomic, Molecular, and Condensed Matter Systems

Samuel T. Chill,<sup>†</sup> Jacob Stevenson,<sup>‡</sup> Victor Ruehle,<sup>‡</sup> Cheng Shang,<sup>‡</sup> Penghao Xiao,<sup>†</sup> James D. Farrell,<sup>‡</sup> David J. Wales,<sup>\*,‡</sup> and Graeme Henkelman<sup>\*,†</sup>

<sup>†</sup>Department of Chemistry and the Institute for Computational and Engineering Sciences, The University of Texas at Austin, Austin, Texas 78712-0165, United States

<sup>‡</sup>University Chemical Laboratories, Lensfield Road, Cambridge CB2 1EW, United Kingdom

**ABSTRACT:** A set of benchmark systems is defined to compare different computational approaches for characterizing local minima, transition states, and pathways in atomic, molecular, and condensed matter systems. Comparisons between several commonly used methods are presented. The strengths and weaknesses are discussed, as well as implementation details that are important for achieving good performance. All of the benchmarks and methods are provided in an online database to make the implementation details available and the results reproducible. While this paper provides a snapshot of the benchmark results, the online framework is structured to be dynamic and incorporate new methods and codes as they are developed.

| Entry                      | <N> | min N | max N | Failed |
|----------------------------|-----|-------|-------|--------|
| optim                      | 176 | 90    | 421   | 0      |
| pele-lbfgs-M100-maxstep0.2 | 179 | 90    | 540   | 0      |
| scipy-lbfgs <sup>1</sup>   | 230 | 105   | 545   | 0      |
| pele-lbfgs-M4-maxstep0.2   | 230 | 100   | 620   | 0      |

<http://OptBench.org>

## 1. INTRODUCTION

Computational tools for geometry optimization of potential energy surfaces (PESs) are ubiquitous in the field of computational chemistry and molecular and materials science. The most appropriate and efficient tools are generally determined by a few limited comparisons between available methods and codes, rather than a systematic consensus of the strengths and weaknesses of different methods. There are several reasons for this. One is that the efficiency of different methods can vary across systems, making it difficult to draw general conclusions in terms of their performance. Another issue is that different research groups have their own codes, often involving methods that they have developed, and are naturally biased toward using them. Furthermore, the barriers associated with distributing and supporting codes for others to use, as well as understanding and implementing different methods, makes systematic comparisons difficult. Finally, the community has not emphasized the importance of standard benchmarks. Such a benchmark database, containing codes and results, facilitates comparisons between methods and implementations, making it easier to draw general conclusions regarding performance. The benefit is for the community of developers and for users who want to understand the similarities and differences of available methods.

The performance of geometry optimization algorithms can be sensitive to the form of the underlying objective function. In this regard, our focus in the present contribution is specifically on systems in chemical physics and materials science, where we want to find structures that are stationary points on the PES. Local minima are usually the first structures to be considered because they provide information about the thermodynamically stable states of the system. We are also interested in the kinetics

that determine the pathways between the stable states. Within the context of transition state theory (TST), the first-order saddle points that connect states via intrinsic reaction coordinates are also important.<sup>1</sup> Accordingly, we also investigate methods for finding these transition states and the approximate steepest-descent paths that connect them, from which energy barriers and reaction rates can be determined.

This manuscript is structured as follows: First, we define our choice of benchmark systems and codes to test. Second, we perform a comparison of minimization methods to find local minima. Third, we compare global optimization methods, involving a long sequence of displacements from local minima and minimizations, designed to find global low-energy structures. Fourth, we investigate single-ended saddle point search methods to find a nearby saddle from a given initial position. Fifth, we compare double-ended saddle point search methods that attempt to locate a saddle between a specified initial and final position. Finally, we look at methods to find all the low energy saddles that lead from an initial minimum in order to calculate the rate of escape from harmonic TST. The paper concludes with a discussion of the results and information about the online benchmark database.

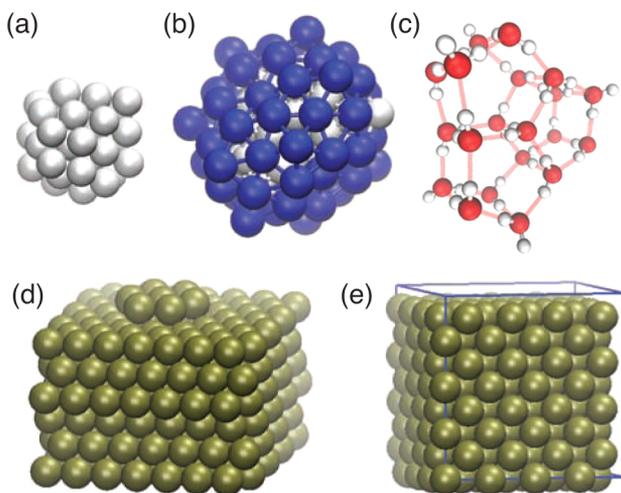
## 2. BENCHMARK SYSTEMS

Five different atomic clusters and condensed phase systems are compared in these benchmarks: (a) a cluster of 38 particles interacting through a pairwise Lennard–Jones potential (LJ<sub>38</sub>), (b) a two-component LJ cluster with 100 particles (BLJ<sub>100</sub>), (c) clusters of 10, 15, and 20 water molecules interacting through

Received: September 28, 2014

Published: November 11, 2014

the pairwise TIP4P potential (TIP4P<sub>N</sub>), (d) a seven-atom heptamer island supported on the (111) surface of a face-centered cubic (FCC) material interacting via a pairwise Morse potential, and (e) a bulk FCC system for the same Morse potential. Figure 1 illustrates these five systems.



**Figure 1.** Five benchmark systems: (a) Lennard–Jones 38-particle cluster, LJ<sub>38</sub>, (b) two-component LJ<sub>100</sub> cluster, (c) 20-molecule water cluster, (d) heptamer island on a (111) surface, and (e) bulk Pt as described by a Morse potential.

The LJ potential employs the standard 6–12 form,<sup>2</sup> appropriate for describing the interaction between noble gas atoms

$$V_{\text{LJ}} = 4\epsilon \sum_{i < j} \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^6 \right] \quad (1)$$

where  $r_{ij} = \|r_j - r_i\|$ , and  $r_i$  is the position vector for atom  $i$ . In the TIP4P model,<sup>3</sup> water molecules are represented as rigid monomers with an OH distance of 0.9572 Å and a HOH angle of 104.52°. The LJ site is centered on the oxygen atom, with partial positive charges (+0.52  $e$ ) on the hydrogen atoms and a partial negative charge (−1.04  $e$ ) displaced 0.15 Å along the bisector of the HOH angle, toward the hydrogen atoms. The potential energy of a cluster of TIP4P molecules is given by

$$V_{\text{TIP4P}} = \sum_{n < m} \left[ \frac{B}{r_{\text{OO}}^{12}} - \frac{C}{r_{\text{OO}}^6} + \sum_i^{\text{on } m} \sum_j^{\text{on } n} \frac{k_C q_i q_j}{r_{ij}} \right] \quad (2)$$

where  $r_{\text{OO}}$  is the distance between the oxygen atoms on molecules  $n$  and  $m$ ,  $q_i$  and  $q_j$  are the partial charges,  $k_C = 332.1$  Å kcal mol<sup>−1</sup>,  $A = 6 \times 10^9$  kcal Å<sup>12</sup> mol<sup>−1</sup>, and  $B = 6.1 \times 10^3$  kcal Å<sup>6</sup> mol<sup>−1</sup>. The Morse potential also takes a pairwise additive form<sup>4</sup>

$$V_{\text{Morse}} = \left[ \sum_{i < j} C e^{-2a(r_{ij} - r_0)} - 2C e^{-a(r_{ij} - r_0)} \right] \quad (3)$$

where  $C = 0.7102$  eV,  $a = 1.6047$  Å<sup>−1</sup>, and  $r_0 = 2.8970$  Å; they were chosen to describe Pt.<sup>5</sup>

There are a couple of differences between these systems that can affect the performance of the methods. The LJ<sub>38</sub> and LJ<sub>100</sub> systems are clusters in the gas phase, where rotations and translations need to be explicitly accounted for. The LJ

potential is also quite stiff in terms of producing relatively large forces and curvatures when atoms approach each other. The Morse potential in the present parametrization is softer and more forgiving of second order optimizers, which build up a local approximation to the curvature. The surface system has frozen atoms in the bottom-most layers of the slab, which automatically prevent translation and rotation. These differences affect the performance of the different optimizers, especially those used in searches for transition states.

### 3. METHOD IMPLEMENTATIONS

Four in-house codes, PELE,<sup>6</sup> OPTIM,<sup>7</sup> GMIN,<sup>8</sup> and EON<sup>9,10</sup> were compared, as well as ASE,<sup>11</sup> and SCIPY.<sup>12</sup> These programs implement a variety of optimizers, including limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS),<sup>13</sup> conjugate gradient (CG),<sup>14,15</sup> fast inertial relaxation engine (FIRE),<sup>16</sup> damped dynamics “quick-min” (QM) method,<sup>17</sup> and steepest-descent (SD). Double-ended transition state algorithms include the climbing-image nudged elastic band (CI-NEB)<sup>18,19</sup> and doubly nudged elastic band (D-NEB)<sup>20</sup> methods. Single-ended min-mode following methods include the hybrid eigenvector-following (EF),<sup>21,22</sup> dimer,<sup>23–25</sup> and Lanczos algorithms, as in the activation–relaxation technique (ARTn).<sup>26</sup> Global optimization was performed using the basin-hopping (BH) algorithm<sup>27</sup> and the temperature basin-paving (TBP) algorithm.<sup>28,29</sup>

The details of these methods will not be discussed here; they are documented in the references provided. Instead, we will mention any important implementation details, as well as qualitative differences between the methods that explain their relative performance.

### 4. RESULTS

**4.1. Local Optimization.** In the first local optimization benchmark, a set of 1000 structures for the LJ<sub>38</sub> cluster are minimized until the magnitude (L<sup>2</sup> norm) of the force is less than 10<sup>−2</sup> reduced units. The initial structures were generated using a cluster-growing algorithm.<sup>30</sup>

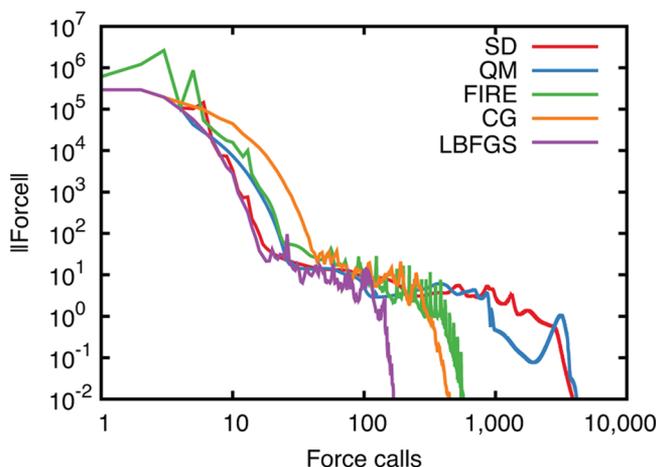
The local optimization benchmark requires that the program being tested load each of these structures and minimize the energy for the LJ potential until the magnitude of the force is less than 10<sup>−2</sup> reduced units. The average number of gradient evaluations of the potential (force calls) is reported, as well as the minimum and maximum number of force calls to find any single minimum. Note that we are comparing force-based optimizers, so it is assumed that the force is evaluated at every iteration; the energy is typically evaluated as well, so each force call also counts these calculations. Table 1 summarizes a

**Table 1.** Minimization of LJ<sub>38</sub> Configurations<sup>a</sup>

| code  | method | force calls |      |      |
|-------|--------|-------------|------|------|
|       |        | avg         | min  | max  |
| OPTIM | LBFGS  | 176         | 90   | 421  |
| PELE  | LBFGS  | 179         | 90   | 540  |
| EON   | LBFGS  | 181         | 90   | 405  |
| ASE   | LBFGS  | 355         | 166  | 9317 |
| EON   | CG     | 453         | 207  | 1154 |
| EON   | FIRE   | 645         | 207  | 2963 |
| EON   | QM     | 3523        | 667  | 9929 |
| EON   | SD     | 4901        | 1355 | 9982 |

<sup>a</sup>Until  $\|\text{Force}\| < 10^{-2}$ .

selected set of benchmark results; when codes have similar performance, only one characteristic value is reported. Figure 2



**Figure 2.** Performance of the different optimizers for  $LJ_{38}$  as implemented in the EON code.

shows how the different algorithms compare in a typical run. A full set of results for these tests and all the other benchmarks reported here can be found online.<sup>31</sup> A comparison of the performance for a typical starting configuration is shown in Figure 2.

In this local optimization benchmark, LBFGS is the clear winner, with CG in second place, followed by the other methods. This result is not surprising;<sup>32</sup> both LBFGS and CG employ numerical curvature information to accelerate optimization within a harmonic approximation of the potential. CG can be thought of as having memory of a single curvature, whereas LBFGS builds up an approximate inverse Hessian with a longer memory. In our implementation, the performance of LBFGS gradually improves with the length of the memory, and a default value of 20 previous steps works well, although additional speed gains can be made with larger values. Perhaps even more important, however, is that LBFGS provides a step length, whereas CG gives only a direction. Some implementations of CG use a bracketing approach to find a minimum along this direction; the implementation in EON evaluates the curvature along the direction with a second force evaluation and does a Newton's step to approximate the location of the zero in the force along the line. This formulation requires a second force call at each step compared to LBFGS, which is evident from the results in Table 2, with CG taking approximately twice as long to converge as LBFGS.

**Table 2. Minimization of Bulk FCC Solid until  $\|Force\| < 10^{-3}$  eV/Å**

| code  | method | force calls |     |     |
|-------|--------|-------------|-----|-----|
|       |        | avg         | min | max |
| OPTIM | LBFGS  | 46          | 21  | 80  |
| PELE  | LBFGS  | 51          | 24  | 84  |
| EON   | LBFGS  | 52          | 35  | 81  |
| ASE   | LBFGS  | 54          | 35  | 91  |
| EON   | CG     | 106         | 67  | 183 |
| EON   | FIRE   | 156         | 107 | 212 |
| EON   | SD     | 196         | 95  | 360 |

The first-order methods, FIRE, QM, and SD, are all slower. The potential advantage of these methods is stability,<sup>32</sup> although stability of the optimizers was not an issue for this benchmark.

The same relative performance for the different optimizers is reflected in a second benchmark, namely, the minimization of 1000 structures randomly displaced from a bulk crystalline FCC structure, modeled with a Morse interatomic potential. Again, LBFGS is the fastest, followed by CG, and then the methods that do not use approximate curvature information. The LBFGS implementation in ASE is a little slower because it uses a fixed initial Hessian (here 25 eV/Å<sup>2</sup>) as opposed to the LBFGS implementations in EON, OPTIM, and PELE, where the initial Hessian is updated at each step based upon the curvature between the current and previous configurations.

A few comments should be made about our implementations of these optimizers. First, all methods employed a “max move” parameter; if the optimizers ever try to make a step larger than this maximum, the size is reduced in magnitude to the limit. Our limit is generally set between 0.1–0.2 Å or a fraction of a LJ distance unit. Another important issue is that the LBFGS and CG methods can reach ill-conditioned states. When the optimizers operate outside the harmonic region around a minimum, negative Hessian eigenvalues can be introduced into the memory in LBFGS. Following the LBFGS algorithm blindly would take the system to a maximum along such directions. There are different ways to deal with this situation. In EON, a “max move” step is taken parallel to the gradient, and the memory is reset. In PELE, OPTIM, and GMIN, the step is inverted so that the algorithm is forced to move down the potential, and the memory is retained. The LBFGS implementations listed simply accept the LBFGS step or do a minimalistic backtracking line search, but none of them, by default, use a full line search. CG algorithms can also become frustrated if the search direction becomes linearly dependent upon search direction at the previous step. It is essential to check for this condition, which is done implicitly in the Polak–Ribiere update formula<sup>33</sup> and explicitly as in the Powell reset criterion.<sup>34</sup>

**4.2. Global Optimization.** Three benchmarks are defined for global optimization. The first is for the same  $LJ_{38}$  cluster considered above for the local optimization test. The second is a two-component BLJ<sub>100</sub> cluster. The third is a series of TIP4P water clusters, composed of 10, 15, and 20 molecules. The BLJ cluster is composed of 42 A particles and 48 B particles with LJ parameters  $\epsilon_{AA} = \epsilon_{AB} = \epsilon_{BB} = 1$ ,  $\sigma_{AA} = 1$ ,  $\sigma_{BB} = 1.3$ , and  $\sigma_{AB} = (\sigma_{AA} + \sigma_{BB})/2$ . The first example is a relatively simple test in which the benchmark requires all entries to report the number of force calls, on average, required to find the global minimum from 100 initial structures. The initial structures were generated as in the local optimization benchmark, except that they were initially minimized. The second benchmark is significantly harder because of the two components and the frustration in the energy landscape caused by the different sizes of the particles. Because the algorithms are not expected to locate the global minimum within the number of allowed steps, the benchmark reports the average lowest energy, along with the minimum and maximum values, found from 100 initial structures after 2,000,000 force calls.

Both EON and PELE use standard Cartesian trial moves and LBFGS to minimize trial structures; their performance is comparable. The GMIN code is better optimized for this system, using angular moves for surface atoms,<sup>36</sup> as well as symmetrized moves (BH-sym and BH-csm) (Table 3).<sup>37</sup> However, the best

**Table 3. Global Optimization of LJ<sub>38</sub> Clusters**

| code | method | force calls (thousands) |      |      |
|------|--------|-------------------------|------|------|
|      |        | avg                     | min  | max  |
| GMIN | BH-csm | 6.4                     | 0.07 | 27   |
| GMIN | BH-sym | 23                      | 0.7  | 121  |
| GMIN | BH     | 266                     | 6.5  | 1170 |
| EON  | BH     | 508                     | 4.6  | 1796 |
| PELE | BH     | 522                     | 9.4  | 2534 |

results obtained so far (Table 4) correspond to a deterministic search scheme to locate the optimal chemical ordering.<sup>38</sup> This procedure is based on Kernighan and Lin's (KL)<sup>39</sup> heuristic for partitioning graphs.

**Table 4. Global Optimization of a Two-Component BLJ<sub>100</sub> Cluster<sup>a</sup>**

| code | method | lowest energy |        |        |
|------|--------|---------------|--------|--------|
|      |        | avg           | min    | max    |
| GMIN | BH/KL  | -589.5        | -599.0 | -578.6 |
| EON  | BH     | -584.6        | -596.1 | -574.1 |

<sup>a</sup>This reports the average minimum energy reached after 2,000,000 force calls. Previous studies<sup>35</sup> have reported a putative global minimum energy of -604.80 reduced energy units.

Similarly, the third benchmark is more difficult than the first owing to the coupling between orientational and translational degrees of freedom in molecular clusters. Both the BH and TBP algorithms were benchmarked for 10000, 500, and 100 initial structures for 10, 15, and 20 molecules, respectively. We use the linear version presented in the original TBP paper (L-TBP), with the parametrization therein.<sup>28</sup> Uniformly random rotational displacements were employed with both algorithms. Although the L-TBP algorithm performs slightly better in all three cases, the results are equivalent for practical purposes (Table 5).

**Table 5. Global Optimization of TIP4P<sub>N</sub> (N = 10, 15, 20) Clusters<sup>a</sup>**

| code | method | N  | force calls (thousands) |       |         |
|------|--------|----|-------------------------|-------|---------|
|      |        |    | avg                     | min   | max     |
| GMIN | BH     | 10 | 16.236                  | 0.121 | 148.360 |
| GMIN | L-TBP  | 10 | 16.099                  | 0.121 | 145.120 |
| GMIN | BH     | 15 | 670.65                  | 4.55  | 5026.50 |
| GMIN | L-TBP  | 15 | 625.36                  | 2.35  | 6176.90 |
| GMIN | BH     | 20 | 28597                   | 441   | 141910  |
| GMIN | L-TBP  | 20 | 25567                   | 597   | 143400  |

<sup>a</sup>Statistics are reported for runs starting from 10000, 500, and 100 initial structures for 10, 15, and 20 molecules, respectively.

**4.3. Single-Ended Saddle Point Searches.** Single-ended saddle point finding methods start from a single configuration on the PES and aim to converge to a nearby saddle point. Almost all such methods rely on the determination of an uphill direction along which the potential is a local maximum, while minimizing in the space perpendicular to that direction. A common choice is to use the local lowest curvature mode as the maximization direction; this class of methods, which is referred to as "min-mode following" methods, evolved from the earlier eigenvector-following and surface walking methods.<sup>40-44</sup> Over-

all translation and rotation can be projected out in this procedure.<sup>45</sup>

The performance of the min-mode following algorithms can be understood by decomposing them into their two components: (a) identifying the lowest (nonzero) curvature mode and (b) optimizing to the saddle point. We compare three strategies for estimating the lowest curvature mode: minimization of a Rayleigh-Ritz ratio<sup>46</sup> (hybrid EF methods),<sup>21</sup> rotation of a dimer (dimer method),<sup>23-25</sup> and Lanczos (as used in the ARTn method).<sup>26</sup> In fact, the Rayleigh-Ritz approach used in hybrid EF methods is based upon the same finite-difference gradient of the Rayleigh quotient that is used in the dimer method, as we have recently reviewed.<sup>47</sup> Rayleigh-Ritz minimization and dimer rotations both correspond to minimization of the same force equations, although the implementation details can be different.

Table 6 compares these strategies by choosing points near saddles in the LJ<sub>38</sub> cluster and seeing how many force calls are

**Table 6. Determination of Lowest Curvature Modes for Points near Saddles in LJ<sub>38</sub><sup>a</sup>**

| code  | method       | force calls |     |     |
|-------|--------------|-------------|-----|-----|
|       |              | avg         | min | max |
| OPTIM | RR           | 25          | 13  | 58  |
| PELE  | RR           | 25          | 12  | 61  |
| TSASE | Lanczos      | 25          | 13  | 54  |
| TSASE | dimer (BFGS) | 27          | 13  | 65  |
| EON   | dimer (BFGS) | 28          | 14  | 70  |
| EON   | Lanczos      | 28          | 13  | 73  |
| TSASE | dimer (CG)   | 29          | 13  | 86  |
| EON   | dimer (CG)   | 30          | 14  | 93  |

<sup>a</sup>The eigenvector corresponding to the smallest non-zero Hessian eigenvalue is considered converged when the dot product with the exact vector is greater than 0.99.

required to determine the lowest eigenvector direction. These methods all use only first derivatives. The lowest (nonzero) curvature mode could be found by diagonalizing the Hessian matrix, but this approach is often significantly slower, especially for larger systems. No initial information about the mode with the smallest nonzero Hessian eigenvalue is provided; the eigenvector corresponding to the unique negative eigenvalue is randomly initialized. OPTIM, PELE, and EON use LBFGS as the optimizer in the EF and dimer methods.

The second aspect of the single-ended transition state methods is convergence to a saddle, which is achieved by maximizing the potential along the chosen uphill direction and minimization in all other directions. Again, there are different strategies. At each iteration in hybrid EF, the system is moved uphill along the negative mode according to a Newton's method type step.<sup>44,48</sup> The system is then minimized in the space perpendicular to that mode, and the eigenvector corresponding to the uphill direction is reconverged. In the dimer method, the force is inverted along the lowest mode and followed uphill with a standard (force-based) optimizer. Table 7 shows the relative performance of the methods for finding a saddle starting from a point between pairs of adjacent minima in LJ<sub>38</sub>.

A drawback of the Lanczos algorithm is that several force calls are required to determine that the lowest eigenvector is sufficiently converged. This is because convergence of the eigenvector is measured by a change in the estimated lowest

**Table 7.** Transition States from Starting Points near Saddles in LJ<sub>38</sub><sup>a</sup>

| code  | method  | force calls |     |      | failed |
|-------|---------|-------------|-----|------|--------|
|       |         | avg         | min | max  |        |
| OPTIM | EF      | 145         | 57  | 565  | 0      |
| PELE  | EF      | 192         | 59  | 1488 | 0      |
| EON   | Lanczos | 237         | 65  | 1898 | 0      |
| EON   | dimer   | 528         | 92  | 3581 | 0      |

<sup>a</sup>Convergence is defined by  $\|\text{Force}\| < 10^{-3}$ .

eigenvalue. In contrast, the hybrid EF and dimer methods use the root-mean-square gradient for the Rayleigh–Ritz ratio or the rotational force on the eigenvector to determine convergence and, in fact, to determine if any steps or rotations are required. Particularly near the saddle, when the lowest mode is found to sufficient accuracy, the hybrid EF and dimer methods can outperform Lanczos by avoiding any refinement of the uphill direction.

This effect is illustrated in a slightly different benchmark, using the Pt-heptamer island system, where we start with an initial point near a saddle, but this time the reactant minimum structure is also supplied. Information about a minimum is typically known, for example, when single-ended search methods are used to find saddles connected to an initial minimum. The vector between the reactant and the initial search point can be a reasonable guess for the uphill direction if it does not contain any components of Hessian eigenvectors corresponding to overall translation or rotation. Using this information, the performance of the different methods is similar, as shown in Table 8.

**Table 8.** Saddle for Pt-Heptamer Island<sup>a</sup>

| code  | method  | force calls |     |     | failed |
|-------|---------|-------------|-----|-----|--------|
|       |         | avg         | min | max |        |
| OPTIM | EF      | 71          | 43  | 143 | 0      |
| PELE  | EF      | 88          | 52  | 198 | 0      |
| EON   | Lanczos | 106         | 71  | 163 | 0      |
| EON   | dimer   | 116         | 83  | 160 | 0      |

<sup>a</sup>Convergence is defined by  $\|\text{Force}\| < 10^{-4}$  eV/Å.

When fewer force calls are required to find the uphill direction, the advantage of the dimer and hybrid EF methods is apparent.

**4.4. Double-Ended Saddle Searching.** A second class of saddle point methods involves double-ended searches, in the sense that they find a saddle (or a set of saddles and intermediate minima) between specified reactant and product end points. To benchmark these methods, we choose pairs of minima known to be separated by a single transition state and see how quickly the corresponding saddle can be found. Table 9 shows the results for the LJ<sub>38</sub> cluster using selected pairs of adjacent minima. An issue for this system, and all gas-phase molecules, is that overall translation and rotation should be removed. This condition can be achieved (to some extent) by explicitly projecting out these modes in the NEB methods, but the chain-of-states approaches can still suffer from elongation of the path due to displacements between the images along the low or zero-frequency normal modes. In extreme cases, this elongation can lead to slow convergence, low resolution of images around the saddle, and a failure to converge. In this LJ<sub>38</sub>

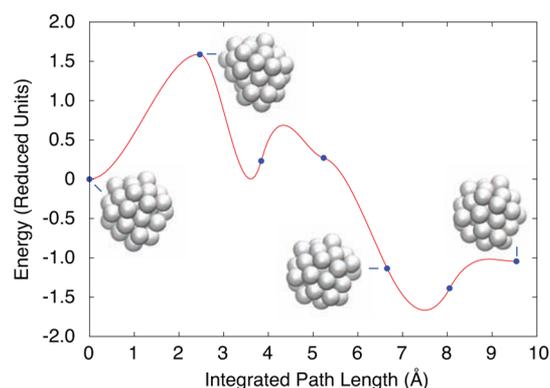
**Table 9.** Transition State between Adjacent Minima in LJ<sub>38</sub> Cluster with a Convergence Condition of  $\|\text{Force}\| < 10^{-3}$ <sup>a</sup>

| code  | method     | force calls |     |      | failed |
|-------|------------|-------------|-----|------|--------|
|       |            | avg         | min | max  |        |
| OPTIM | DNEB+EF    | 131         | 66  | 260  | 0      |
| EON   | Lanczos    | 287         | 85  | 1623 | 0      |
| PELE  | DNEB+EF    | 293         | 107 | 782  | 0      |
| EON   | dimer      | 337         | 89  | 2079 | 0      |
| EON   | CI-NEB(5)  | 827         | 407 | 2347 | 7      |
| EON   | CI-DNEB(5) | 861         | 407 | 2187 | 5      |

<sup>a</sup>Cluster with a Convergence Condition of  $\|\text{Force}\| < 10^{-3}$ .

benchmark, we have chosen a maximum number of force calls (500 iterations) where NEB calculations that start to develop problems, for example, due to long paths through intermediate minima, are considered failures.

An example of a “failed” NEB calculation is shown in Figure 3. While we know that the minima in question are indeed



**Figure 3.** Potential energy profile for a NEB calculation that failed to converge in the LJ<sub>38</sub> system due to the presence of an intermediate minimum and path elongation.

connected by a single transition state (found using approximate steepest-descent following small displacements along the eigenvector corresponding to the unique negative Hessian eigenvalue at the transition state), convergence of the NEB from an initial linear interpolation leads to intermediate minima and an elongated path. Using a single-ended search method to find the first transition state, adding more images, or breaking up the band into separate NEB calculations between all local minima are sensible ways of recovering from this problem.

The single-ended searches are not as sensitive to complex pathways because they search only for a nearby saddle instead of a path. It is still possible to use them with information from two end points. In fact, the dimer and Lanczos methods work very efficiently when initialized half way between two adjacent minima, using the vector between as an initial guess for the negative mode. In PELE and OPTIM the double- and single-ended approaches are combined. First, a DNEB calculation is performed to find an image in the vicinity of the saddle. The double-nudging helps to keep the band short and smooth, as well as helping to prevent elongation of the path. Instead of a fixed number of images, a constant image density was used. When the force on the highest energy image drops below a loose force threshold (here  $\|\text{Force}\| < 0.1$ ), the algorithm switches to hybrid EF, and the transition state is tightly refined. Switching to hybrid EF avoids having to converge the DNEB

images and also improves the stability of the single-ended methods, which is particularly important for gas-phase clusters.

For the Pt-heptamer island diffusion test, frozen atoms at the bottom of the slab hold the surface in place and automatically eliminate rotation and translational degrees of freedom. Removing these modes that correspond to zero frequencies improves the relative performance of the NEB method and reduces the need for double-nudging. Another factor in favor of the NEB is that the paths chosen to investigate, 60 low energy diffusion mechanisms from the compact Pt-heptamer, are better approximated by linear interpolation compared to the LJ<sub>38</sub> cluster.

Table 10 shows the number of force calls required to find the transition state along the path, which works efficiently for only

**Table 10. Saddle between Adjacent Minima for Pt-AHeptamer Island<sup>a</sup>**

| code  | method    | force calls |     |     | failed |
|-------|-----------|-------------|-----|-----|--------|
|       |           | avg         | min | max |        |
| OPTIM | DNEB+EF   | 103         | 59  | 194 | 0      |
| EON   | CI-NEB(1) | 110         | 29  | 594 | 0      |
| EON   | Lanczos   | 165         | 72  | 308 | 0      |
| EON   | dimer     | 182         | 74  | 322 | 0      |
| EON   | CI-NEB(3) | 248         | 86  | 761 | 3      |
| PELE  | DNEB+EF   | 255         | 96  | 782 | 2      |
| EON   | CI-NEB(5) | 391         | 162 | 797 | 0      |

<sup>a</sup>Tolerance of  $\|Force\| < 10^{-4}$  eV/Å.

a single image CI-NEB calculation, labeled as CI-NEB(1). The dimer and Lanczos algorithms also converge but with some extra work because the eigenvector corresponding to the negative eigenvalue is optimized instead of being held fixed, as in the single image CI-NEB calculation. Realistically, one would not generally use a single image for an NEB calculation, so we include the three-image NEB, CI-NEB(3), as a more realistic indication of the cost of the method.

## 5. DISCUSSION AND CONCLUSIONS

The aim of this work is to establish a set of benchmarks for geometry optimization, transition state searches, and characterization of pathways in atomic, molecular, and condensed matter systems. The benchmarks can be accessed at <http://optbench.org/>. All entries are provided with the corresponding source code and scripts to run the benchmark. Hence, the results should be reproducible, and meaningful comparisons are possible. The results presented here compare a limited number of methods and codes and include representative examples from the online database. A longer-term aim is to have a dynamic site, which is updated as new methods and codes are developed. Contributions are welcome, either by email or via the subversion (SVN) repository.

## AUTHOR INFORMATION

### Corresponding Authors

\*E-mail: [dw34@cam.ac.uk](mailto:dw34@cam.ac.uk) (D.J.W.).

\*E-mail: [henkelman@utexas.edu](mailto:henkelman@utexas.edu) (G.H.).

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (CHE-1152342) and Welch Foundation (F-1841) at the University of Texas at Austin, EPSRC, and European Research Council. We thank Rye Terrell and Hannes Jónsson for many helpful discussions, and Hannes especially for providing the initial motivation for this project. Computational resources were provided by the Texas Advanced Computing Center.

## REFERENCES

- (1) Murrell, J. N.; Laidler, K. J. *Trans. Faraday Soc.* **1968**, *64*, 371–377.
- (2) Jones, J. E.; Ingham, A. E. *Proc. R. Soc. A* **1925**, *107*, 636–653.
- (3) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D. *J. Chem. Phys.* **1983**, *79*, 926.
- (4) Morse, P. M. *Phys. Rev.* **1929**, *34*, 57.
- (5) Bassett, D. W.; Webber, P. R. *Surf. Sci.* **1978**, *70*, 520.
- (6) Python Energy Landscape Explorer. GitHub. <https://github.com/pele-python/pele> (accessed 2014).
- (7) OPTIM: A Program for Optimizing Geometries and Calculating Reaction Pathways. <http://www.wales.ch.cam.ac.uk/OPTIM> (accessed 2014).
- (8) GMIN: A Program for Finding Global Minima and Calculating Thermodynamic Properties from Basin-Sampling. <http://www.wales.ch.cam.ac.uk/GMIN> (accessed 2014).
- (9) Chill, S. T.; Welborn, M.; Terrell, R.; Zhang, L.; Berthet, J.-C.; Pedersen, A.; Jónsson, H.; Henkelman, G. *Model. Simul. Mater. Sci. Eng.* **2014**, *22*, 055002.
- (10) EON: Long Timescale Dynamics. <http://theory.cm.utexas.edu/eon> (accessed 2014).
- (11) Atomic Simulation Environment. <https://wiki.fysik.dtu.dk/ase> (accessed 2014).
- (12) SciPy. <http://www.scipy.org/> (accessed 2014).
- (13) Nocedal, J. *Math. Comput.* **1980**, *35*, 773–782.
- (14) Hestenes, M. R.; Steifel, E. *J. Res. Natl. Bur. Stand.* **1952**, *49*, 409–436.
- (15) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computation*, 2nd ed.; Cambridge University Press: Cambridge, 1992; p 420.
- (16) Bitzek, E.; Koskinen, P.; Fähler, F.; Moseler, M.; Gumbusch, P. *Phys. Rev. Lett.* **2006**, *97*, 170201–1–4.
- (17) Jónsson, H.; Mills, G.; Jacobsen, K. W. In *Classical and Quantum Dynamics in Condensed Phase Simulations*; Berne, B. J., Ciccotti, G., Coker, D. F., Eds.; World Scientific: Singapore, 1998; pp 385–404.
- (18) Henkelman, G.; Jónsson, H. *J. Chem. Phys.* **2000**, *113*, 9978–9985.
- (19) Henkelman, G.; Uberuaga, B. P.; Jónsson, H. *J. Chem. Phys.* **2000**, *113*, 9901–9904.
- (20) Trygubenko, S. A.; Wales, D. J. *J. Chem. Phys.* **2004**, *120*, 2082–2094.
- (21) Munro, L. J.; Wales, D. J. *Phys. Rev. B* **1999**, *59*, 3969.
- (22) Kumeda, Y.; Wales, D. J.; Munro, L. J. *Chem. Phys. Lett.* **2001**, *341*, 185–194.
- (23) Henkelman, G.; Jónsson, H. *J. Chem. Phys.* **1999**, *111*, 7010–7022.
- (24) Heyden, A.; Bell, A. T.; Keil, F. J. *J. Chem. Phys.* **2005**, *123*, 224101–224114.
- (25) Kästner, J.; Sherwood, P. J. *Chem. Phys.* **2008**, *128*, 014106–014116.
- (26) Malek, R.; Mousseau, N. *Phys. Rev. E* **2000**, *62*, 7723–7728.
- (27) Wales, D. J.; Scheraga, H. A. *Science* **1999**, *285*, 1368–1372.
- (28) Shanker, S.; Bandyopadhyay, P. *J. Phys. Chem. A* **2011**, *115*, 11866–11875.
- (29) Rakshit, A.; Bandyopadhyay, P. *Comp. Theor. Chem.* **2013**, *1021*, 206–214.
- (30) Lloyd, L. D.; Johnston, R. L.; Salhi, S. *J. Comput. Chem.* **2005**, *26*, 1069–1078.
- (31) OptBench. <http://optbench.org/> (accessed 2014).

- (32) Asenjo, D.; Stevenson, J. D.; Wales, D. J.; Frenkel, D. J. *Phys. Chem. B* **2013**, *117*, 12717–12723.
- (33) Polak, E.; Ribiere, G. *Rev. Fr. Inform. Rech. Oper.* **1969**, *16*, 35–43.
- (34) Powell, M. J. D. *Math. Program.* **1977**, *12*, 241–254.
- (35) Kolossváry, I.; Bowers, K. J. *Phys. Rev. E* **2010**, *82*, 056711.
- (36) Wales, D. J.; Doye, J. P. K. *J. Phys. Chem. A* **1997**, *101*, 5111–5116.
- (37) Oakley, M. T.; Johnston, R. L.; Wales, D. J. *Phys. Chem. Chem. Phys.* **2013**, *15*, 3965–3976.
- (38) Schebarchov, D.; Wales, D. J. *J. Chem. Phys.* **2013**, *139*, 221101.
- (39) Kernighan, B. W.; Lin, S. *Bell Syst. Technol. J.* **1970**, *49*, 291–307.
- (40) Cerjan, C. J.; Miller, W. H. *J. Chem. Phys.* **1981**, *75*, 2800–2806.
- (41) Simons, J.; Jørgensen, P.; Taylor, H.; Ozment, J. *J. Phys. Chem.* **1983**, *87*, 2745–2753.
- (42) Bell, S.; Crighton, J. S. *J. Chem. Phys.* **1984**, *80*, 2464–2475.
- (43) Nichols, J.; Taylor, H.; Schmidt, P.; Simons, J. *J. Chem. Phys.* **1990**, *92*, 340–346.
- (44) Wales, D. J. *J. Chem. Soc. Faraday Trans.* **1993**, *89*, 1305–1313.
- (45) Page, M.; McIver, J. W. *J. Chem. Phys.* **1988**, *88*, 922.
- (46) Horn, R. A.; Johnson, C. R. *Matrix Analysis*; Cambridge University Press: Cambridge, 1985.
- (47) Zeng, Y.; Xiao, P.; Henkelman, G. J. *J. Chem. Phys.* **2014**, *140*, 044115.
- (48) Wales, D. J. *J. Chem. Soc. Faraday Trans.* **1992**, *88*, 653–657.