Parallelized Nudged Elastic Band

6.338J/18.337J Final Project

Anubhav Sinha

1 Introduction

1.1 Background

The field of computational chemistry studies chemical systems *in silico*. Many different methods (using classical and/or quantum mechanics) are used in simulations to explain and predict chemical behavior. These simulations can give us the energy at every possible physical arrangement of the particles being studied. An important property of a chemical system is the potential energy barrier between two different states, i.e. the amount of energy that the system needs to go from one stable configuration to another. This energy is called the activation energy and the point where the activation energy is found is called the transition state. See Figure 1 for an example of a system of interest and its transition state. On a potential energy surface (PES), stable states are local minima of the surface. There can be multiple local minima on a surface; typically, one particular starting configuration and one ending configuration are of interest. We would like to find the transition state between these two points. On a PES, transition states are saddle points. This is equivalent to a min-max problem. The reaction path is represented as a continous line from the starting point to the ending point and the transition state is found at the minimum value of the maximum of the path in the space of all possible paths. This path is called the intrinsic reaction coordinate

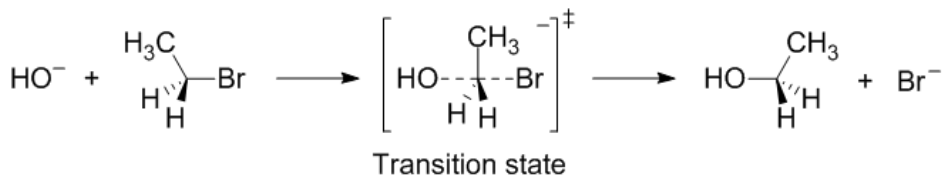

Figure 1: A simple reaction with the transition state shown. Public domain image available at http://en.wikipedia.org/wiki/File:Transition_State.png

(IRC) or minimum energy path (MEP).

1.2 Formalism

The MEP $\vec{q}$ on a PES $V(x_1, x_2, \ldots, x_n)$ is defined as being a path $\vec{q}(s)$ where $s \in [0, 1]$ with endpoints $\vec{q}(0) = \vec{q_i}$ and $\vec{q}(1) = \vec{q_f}$ that are both local minima. For every point along the path, $\frac{d\vec{q}}{ds}$ is parallel to ∇V. Consequently, the minimum energy path passes through at least one saddle point (representing the transition state) since both $\vec{q_i}$ and $\vec{q_f}$ are local minima.

1.3 Methods

There are several existing methods to find the transition state. Some methods are very computationally intensive, requiring calculations to be performed at every single possible physical arrangment of the system. If N particles are in the system of interest and store their locations in (x, y, z) coordinates, this requires searching every possible location of a $3N$-dimensional coordinate space. This is very computationally expensive, since, depending on the method used, each individual energy calculation can take quite a bit of time. These include methods such as the Relaxed PES Scan which constrains the system along one molecular axis (such as fixing a particular bond length) and minimizing the rest of the system around this constraint. Similarly, a hypersphere search constructs a hypersphere with the same dimensionality as the coordinate space and radius equal to the energy of the system. Local minima are found on the surface of the sphere and the radius is varied and the transformation of the paths is traced. Less computationally intensive are methods that

depend on the eigenvectors of the Hessian matrix. At each point searched in this method, the Hessian can be calculated and its eigenvectors followed. These can be used for methods such as the Gradient Extremal Following Method. These are more feasible, but are still difficult since calculating the second derivatives is expensive. Better than these methods are the simpler chain-of-states method. These included the Plain Elastic Band (PEB), Nudged Elastic Band (NEB), and String Method. These methods functionally accomplish the same thing: use several discrete images along a path and view the evolution of the path as a result of the forces applied to it by the PES. These only rely on calculating the gradients at several points along the path. This is more computationally tractable than the other methods. Each state can be handled on a different processor, with a limited amount of communication between processors, making the methods embarrassingly parallel.

1.4 Parallelism

The method fits into the general picture of parallelism as show in Figure 2. The simplest parallelization method is to simply put one image per node. This would involve Send and Recieve calls for every iteration since there would be lots of inter-node communication. If multiple images are kept on one node, they can communicate via direct memory calls, improving performance. There would still be several calls to message passing functions per iteration.

Also note that the entire system must be synchronized. Processors cannot fall behind without slowing down the rest of the calculation. Dynamic load balancing could potentially improve performance further.

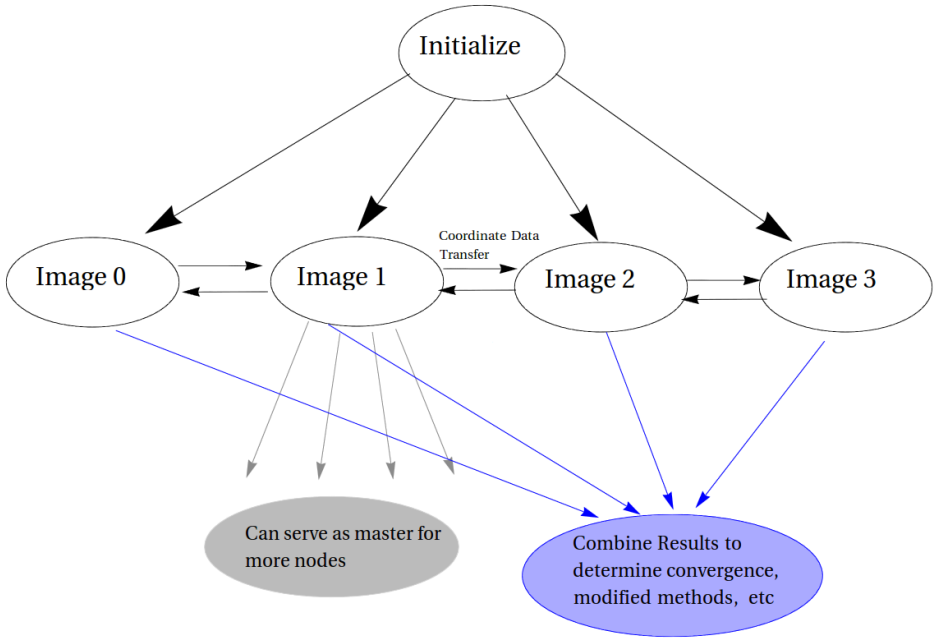

Figure 2: The figure shows how the different parts of the algorithm can be used in the model of parallel computing. The individual images can go on separate processors, each serving as the master node for several other processors that actually carry out complex calcuations for the energies or other properties. These send messages to each other to get the relevant coordinates. Other processors can be involved for reduction operations—quantities can be computed across all images, and then various transformations can occur over all nodes.

2 Algorithm and Implementation

2.1 Nudged Elastic Band

The Nudged Elastic Band (NEB) method is a general method designed to find a MEP on any reasonable potential energy surface. NEB has several steps. On a high level, the nudged elastic band method connects images together using springs to make an elastic band. This keeps the images roughly evenly spaced along the path. The band experiences forces from the potential energy surface. The forces are calculated and the band is iteratively relaxed until it converges to a path. The path that it converges to is a MEP. The advantages to the NEB method are that it converges to the MEP, does not require second derivative information, and is an embarrassingly parallel method. The disadvantages include the number of images needed for clear resolution of the path, the possible convergence to one of multiple valid energy paths between the starting and ending points (i.e., not the path of interest), and the number of iterations needed for convergence.

2.2 Method Outline

1. Calculate initial interpolation $[\vec{q_0}, \vec{q_2}, \ldots, \vec{q_{N-2}}, \vec{q_{N-1}}]$ between $\vec{q_i}$ and $\vec{q_f}$ with N images between. In this implementation of NEB, a straight linear interpolation defined by $\vec{q_p} = \vec{q_i} + \frac{q}{N} \cdot q_f \vec{-} q_i$ was used.

2. Calculate spring forces between adjacent images. These are given by

$$\vec{F}_p^S = k_p(\vec{q}_{p+1} - \vec{q}_p) - k_{p-1}(\vec{q}_p - \vec{q}_{p-1})$$

   In this implementation of NEB, the spring constants were set to be equal. This can be varied if a non-even distribution of images along the band is desired. Also note that the value of the spring constant is not important— this is because of the vector projections described below. The relative values of spring constants are significant, not

the constants themselves.

3. Calculate forces on the images due to the PES. These are given by

$$\vec{F}_p^V = -\nabla V(\vec{q}_p)$$

Depending on the methods used, computing the gradient can require multiple potential evaluations. In this implementation, exact potentials were used, but this can be replaced with a finite difference method.

4. Calculate a modified unit tangent $\vec{\tau}$. There have been several papers published on the value of $\vec{\tau}$. The simple immediate difference method $\vec{\tau}_p = \vec{q_{p+1}} - \vec{q_{p-1}}$ (normalized) works, but not terribly well since it can contribute to the formation of kinks in the path. A more involved defintion $\vec{\tau}_p = \frac{\vec{q_p} - \vec{q_{p-1}}}{|q_p - q_{-1}|} + \frac{\vec{q_{p+1}} - \vec{q_p}}{|\vec{q_{p+1}} - \vec{q_p}|}$ (normalized) works more effectively, but still leads to kink formation. A better defintion is as follows.

$$\vec{\tau}_p = \begin{cases} \vec{\tau}_p^+ \Delta V^{\texttt{MAX}} + \vec{\tau}_p^- \Delta V^{\texttt{MIN}} & V_{p+1} > V_{p-1} \\ \vec{\tau}_p^+ \Delta V^{\texttt{MIN}} + \vec{\tau}_p^- \Delta V^{\texttt{MAX}} & V_{p+1} < V_{p-1} \end{cases}$$

where

$$\Delta V^{\texttt{MAX}} = \max(|V_{p+1} - V_p|, |V_{p-1} - V_p|) \qquad \vec{\tau}_p^+ = \vec{q_{p+1}} - \vec{q_p}$$
$$\Delta V^{\texttt{MIN}} = \min(|V_{p+1} - V_p|, |V_{p-1} - V_p|) \qquad \vec{\tau}_p^- = \vec{q_p} - \vec{q_{p-1}}$$

After normalizing this, the modified tangent helps prevent kinks from forming by weighting the functions used to compute the tangent.

5. Compute the total force on the image.

$$\vec{F}_p = \left(\vec{F}_p^V - (\vec{F}_p^V \cdot \vec{\tau})\vec{\tau}\right) + \left((\vec{F}_p^S \cdot \vec{\tau})\vec{\tau}\right) = \vec{F}_p^V|_\perp + \vec{F}_p^S|_\parallel$$

6. Move images using Steepest Descent; $\vec{Q}_p = \vec{Q}_p + \alpha \cdot \vec{F}_p$ where α is a tunable parameter.

Initialization	MPI_Init MPI_Comm_size MPI_Comm_rank MPI_Barrier MPI_Wtime
Spring Forces	MPI_Send MPI_Recv
Potential Forces, Unit Tangent	Calls to PES
Vector Calculations, Image Movement	No parallelism
Reduction	MPI_Reduce
Convergence	MPI_Bcast MPI_Wtime MPI_Finalize
Optional File I/O	MPI_File_open MPI_File_write_at_all MPI_File_close

Table 1: Important MPI Operations used

7. Reduce operation—compute maximum force magnitude across all nodes. Check if under threshhold value. If under, then converged, Otherwise, loop through steps 2-6 until convergence.

2.3 Implementation

The algorithm was implemented in C++ using MPI for parallelism. Node 0 was the node used for for the reduce operations. The algorithm above corresponds to MPI operations as shown in Table 1.

In the implementation, multiple images can be run on the same node. The number of total images was also variable. The cluster used for running the algorithm was the Evolution cluster; the cluster had 60 nodes with each node having 2 x 2-core Intel Xeon processors with 6 GB RAM/node. Multiple images on one node can communicate directly through memory with much lower latency than inter-node communication.

2.4 Test Systems

There were three main test systems used. These are the Karplus, Muller-Brown, and Wolfe-Quapp surfaces. They were used because they are canonical 2D potential energy surfaces and they are easy to visualize. The implementation was designed to work with n-dimensional potential energy surfaces, and preliminary work suggests that the algorithm converges and gives reasonable paths for 3D and higher PESs. However, futher work needs to be carried out for more conclusive statements. The test systems are all quickly calculatable functions and are representative of simple surfaces that calculations on real systems can give, but they evaluate much more quickly than most real calculations would.

2.4.1 Karplus PES

The Karplus PES is defined by

$$V(x,y) = 0.6(x^2+y^2)^2 + xy - 9(e^{-(x-3)^2-y^2} + e^{-(x+3)^2-y^2})$$

See Figure 3 for a diagram of the PES.

2.4.2 Muller-Brown PES

The Muller-Brown PES is defined by

$$V(x,y) = \sum_{i=0}^{i=3} A_i e^{a_i(x-x_i^0)^2 + b_i(x-x_i^0)(y-y_i^0) + c_i(y-y_i^0)^2}$$

where $A = [-200, -100, -170, 15]$, $a = [-1, -1, -6.5, 0.7]$, $b = [0, 0, 11, 0.6]$, $c = [-10, -10, -6.5, 0.7]$, $x^0 = [1, 0, -0.5, -1]$, $y^0 = [0, 0.5, 1.5, 1]$. See Figure 4 for more details.

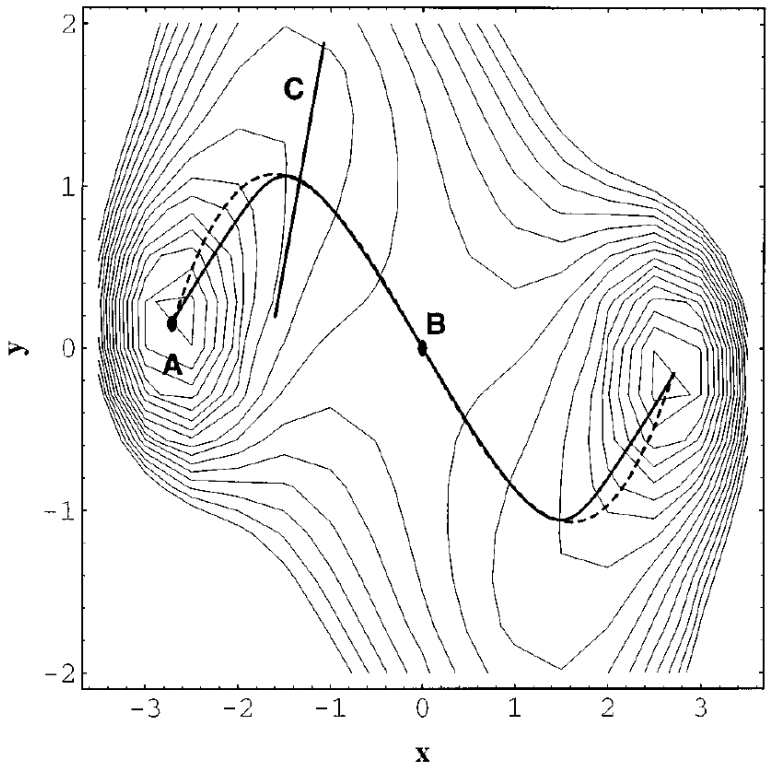

Figure 3: The Karplus PES has a saddle point at the origin and two local minima. The MEP is the solid line on the diagram.

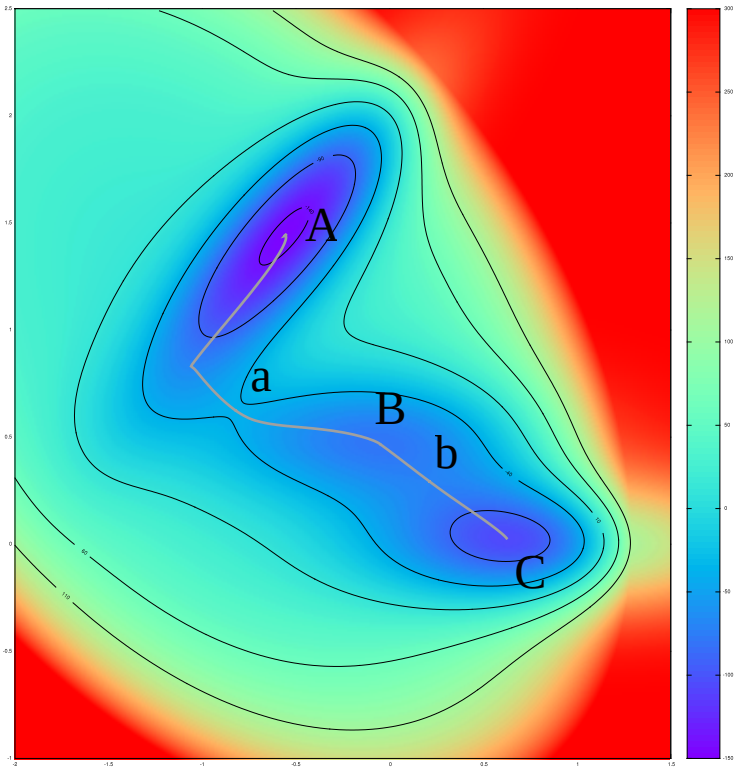

Figure 4: The Muller-Browns PES has two saddle points (points a and b) and three local minima (points A, B, and C). The MEP is the solid line on the diagram; it was calculated using a Newton-Raphson algorithm.

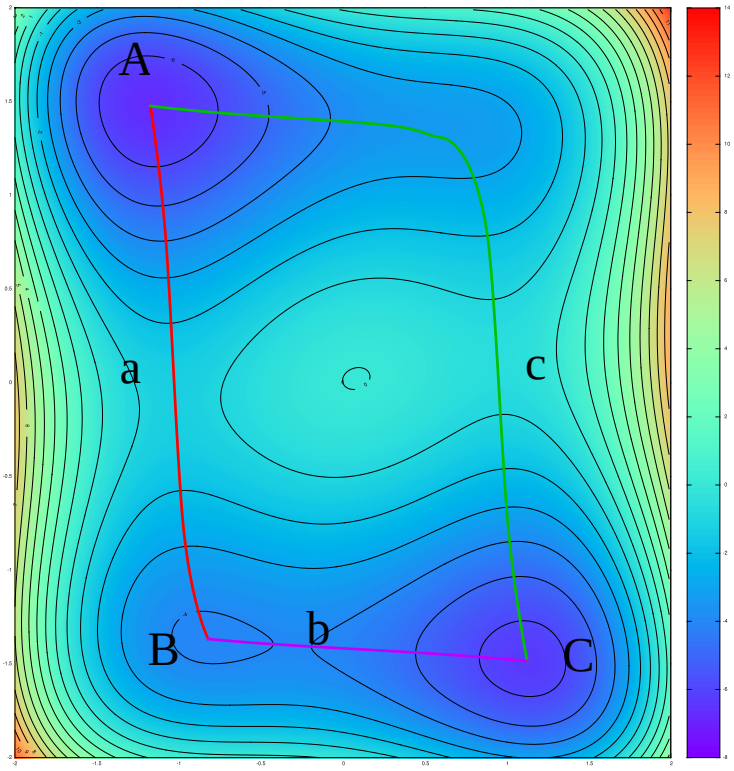

Figure 5: The Wolfe-Quapp PES has three saddle points (points a, b, and c) and three local minima (points A, B, and C). The MEPs are the solid lines on the diagram; they were again calculated using a Newton-Raphson algorithm. There are two paths from A to C, one through local minima B and the other without passing through a local minima. These are both MEPs that the algorithm can converge to.

2.4.3 Wolfe-Quapp PES

The Wolfe-Quapp PES is defined by

$$V(x, y) = x^4 + y^4 - 2x^2 - 4y^2 + xy + 0.3x + 0.1y$$

See Figure 5 for more details.

3 Accuracy Results

The method converges to the following paths for each problem. Figure 6 is for the Karplus PES, Figure 7 is for the Muller-Brown PES, and Figures 8 and 9 are for the Wolfe-Quapp

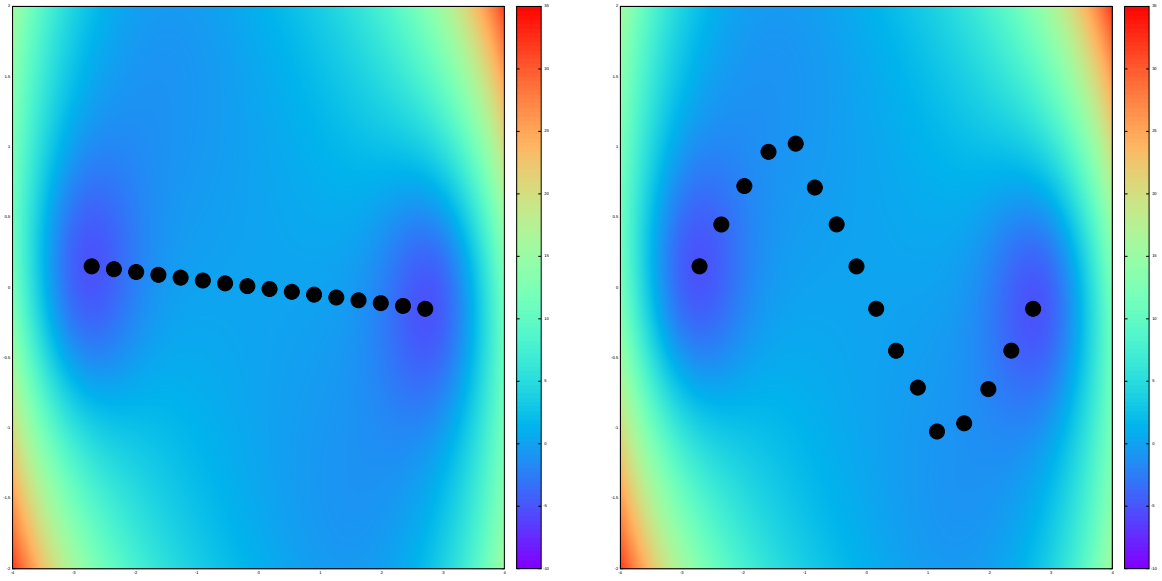

Figure 6: Starting and converged paths for the Karplus PES.

Surface.

3.0.4 Karplus PES

See Figure 6.

3.0.5 Muller-Brown PES

See Figure 7.

3.0.6 Wolfe-Quapp PES

See Figures 8 and 5. The Wolfe-Quapp surface is more interesting. Figures 8 and 9 show two different paths the NEB converged to. NEB is deterministic—the input interpolation is what determines whether or not it converges to one path or the other. This suggests the importance of the initial interpolation. A good heuristic for selecting path to seed NEB with is important, as is testing multiple paths to determine whether or not an MEP exists.

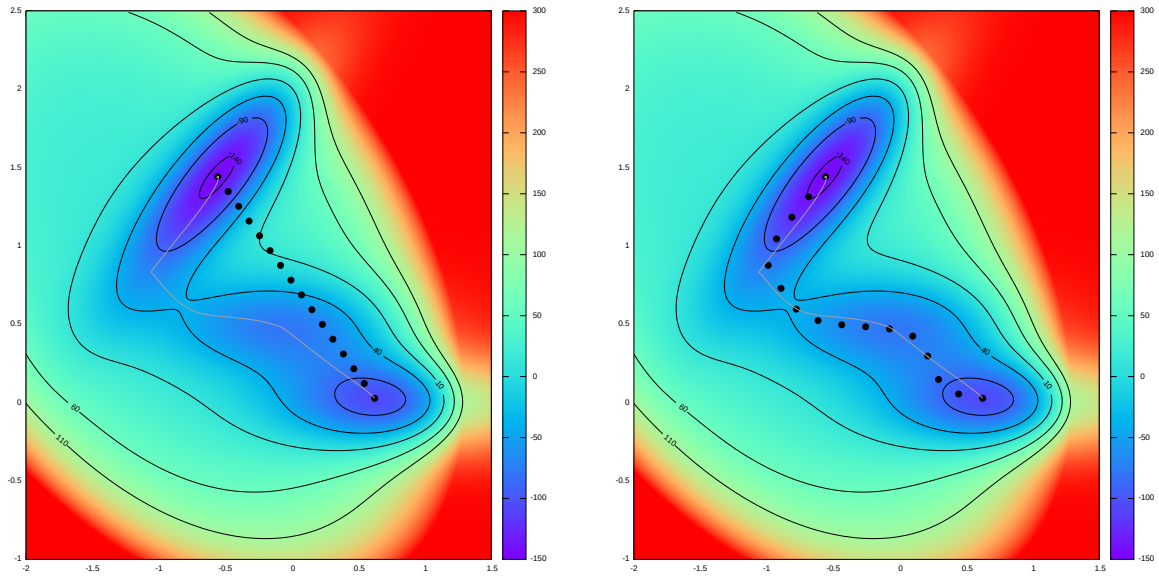

Figure 7: Starting and converged paths for the Muller-Brown PES.

3.1 Summary

The images show that the NEB method very accurately converges to the MEP. The potential vs. arclength can also be plotted for another view of the accuracy of the method. Doing this in Figure 10 for the upper Wolfe-Quapp curve, it can be seen that the curves are very close to each other. This is an additional demonstration that the method is both accurate and precise.

4 Parallelism Results

The parallelism was benchmarked using a 24-image band on the Muller-Brown potential surface. As mentioned earlier, the potentials used are exact functions that evaluate quickly. This is not a very realistic test case–a single energy calculation can take hours on a complex system. The 24-images were run on 1, 2, 4, 8, 12, 16, 20 and 24 processors and the data were plotted in Figure 11. As one might expect for a problem like this, when the functions calculate very quickly, running the code with all images on one processor as a serial code is the fastest. At two processors, the time jumped, since messages had to be sent between nodes,

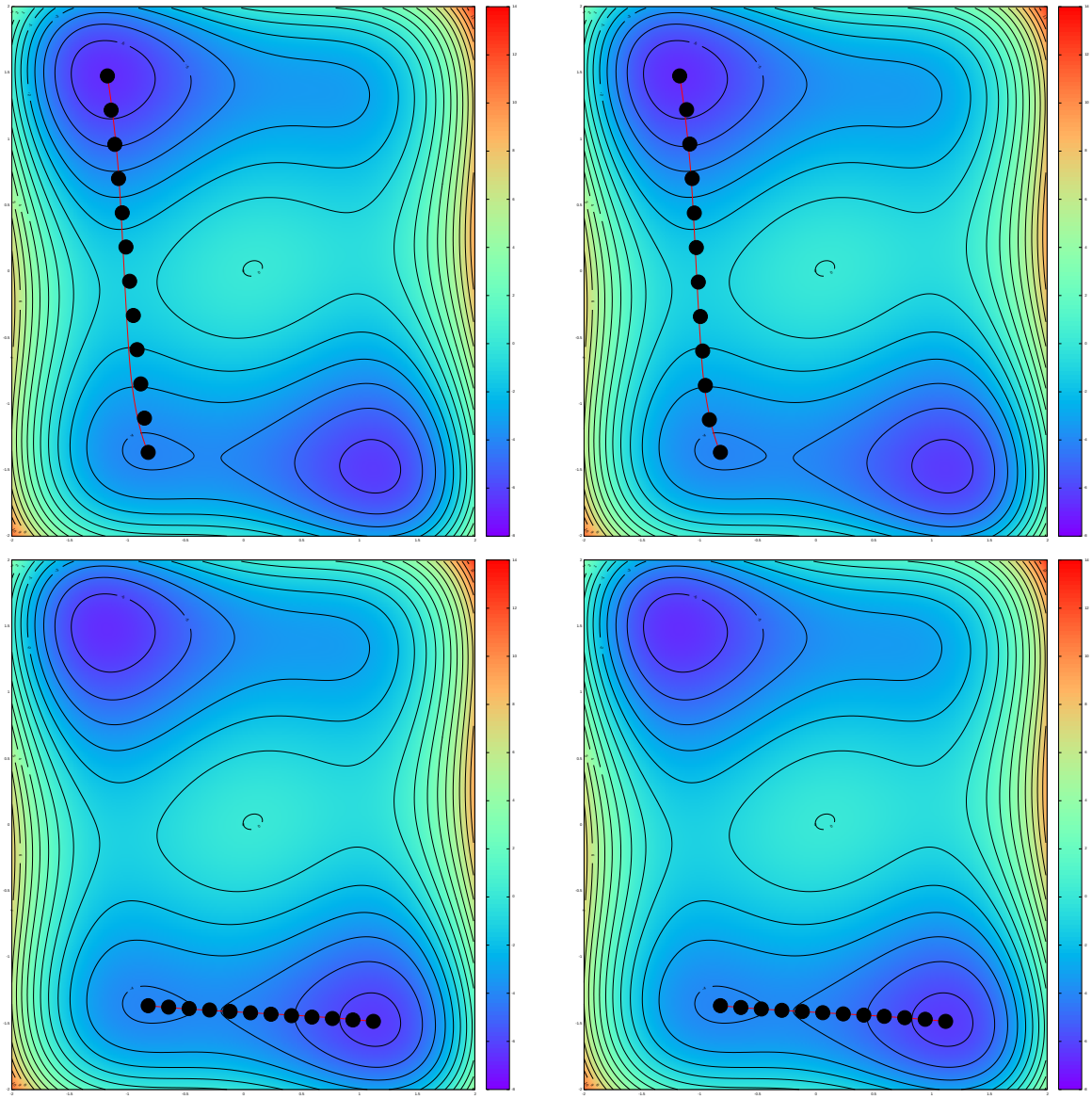

Figure 8: Two simpler problems for the Wolfe-Quapp PES.

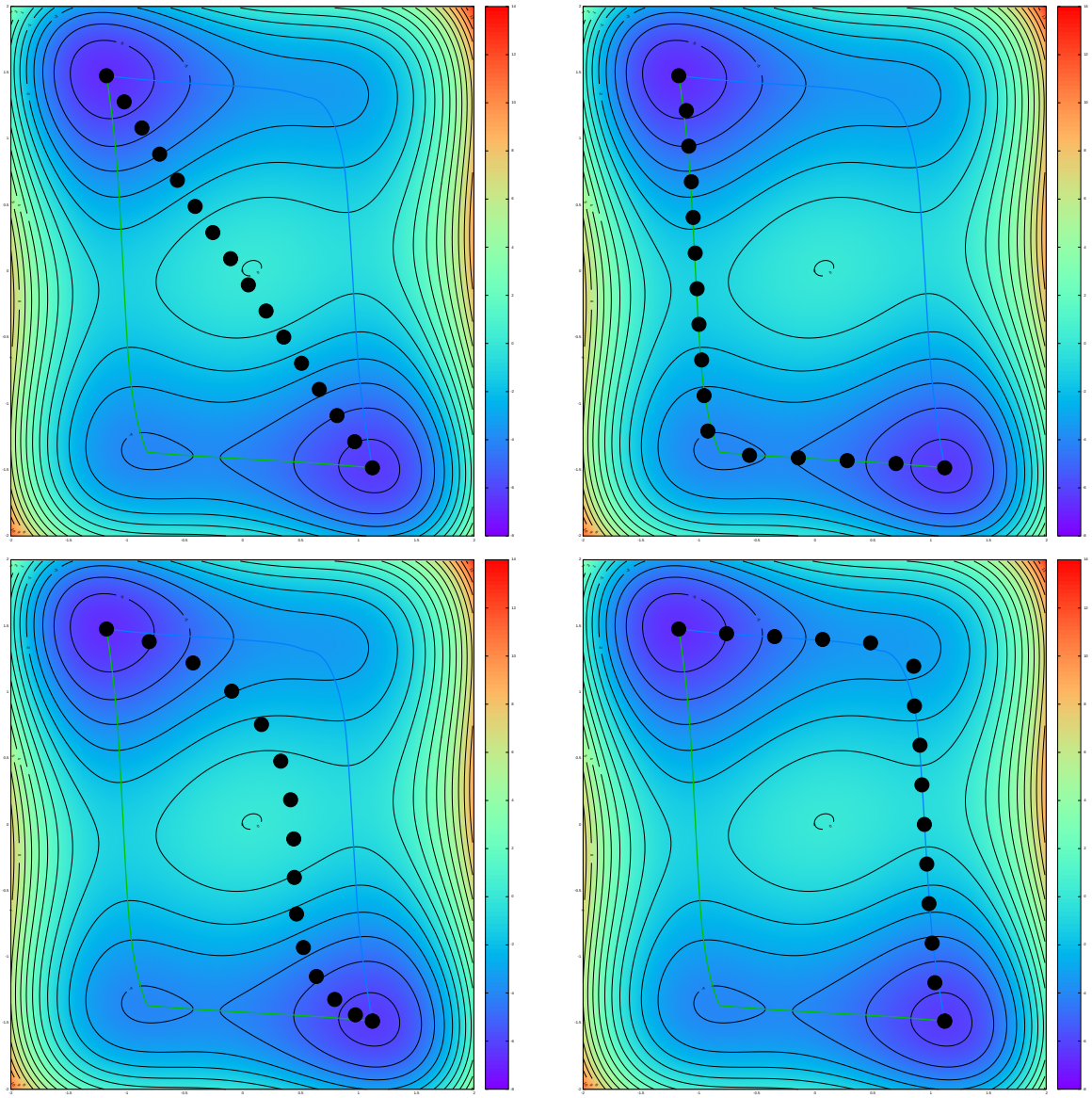

Figure 9: The path from A to C is more complicated. With an initial linear interpolation, the path converges to the lower MEP. With a slightly shifted interpolation, the path converges to the upper MEP. The upper MEP was constructed by computing the Hessian matrix at the endpoints and initially following the eigenvectors with a parametrizable strength. This shows the importance of the initial path selection in determining which path NEB converges to.

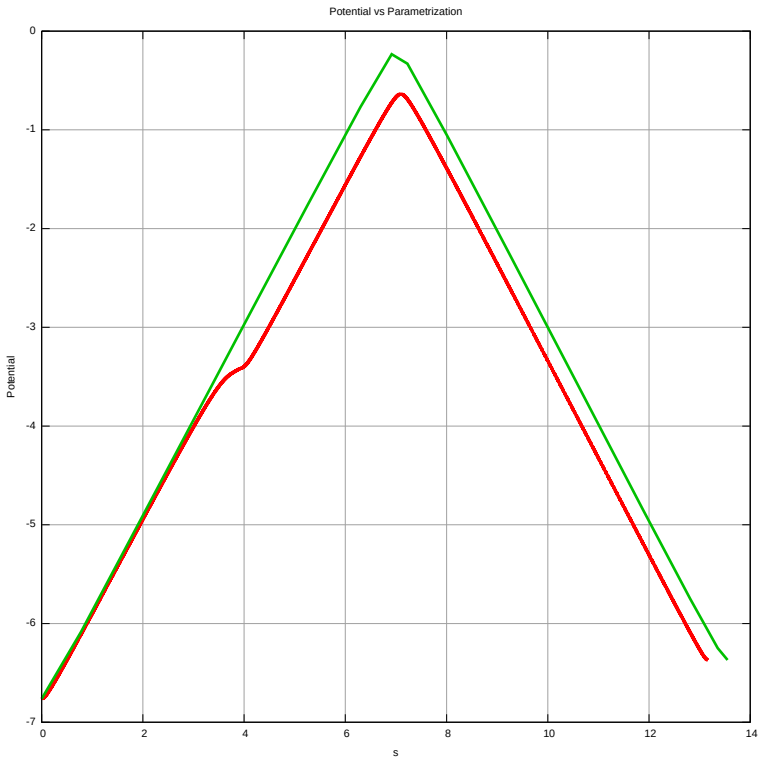

Figure 10: The potential vs. arclength plot for the Newton-Raphson Curve (red) and the NEB curve (green). The resolution of the NEB curve can be improved by increasing the number of images.

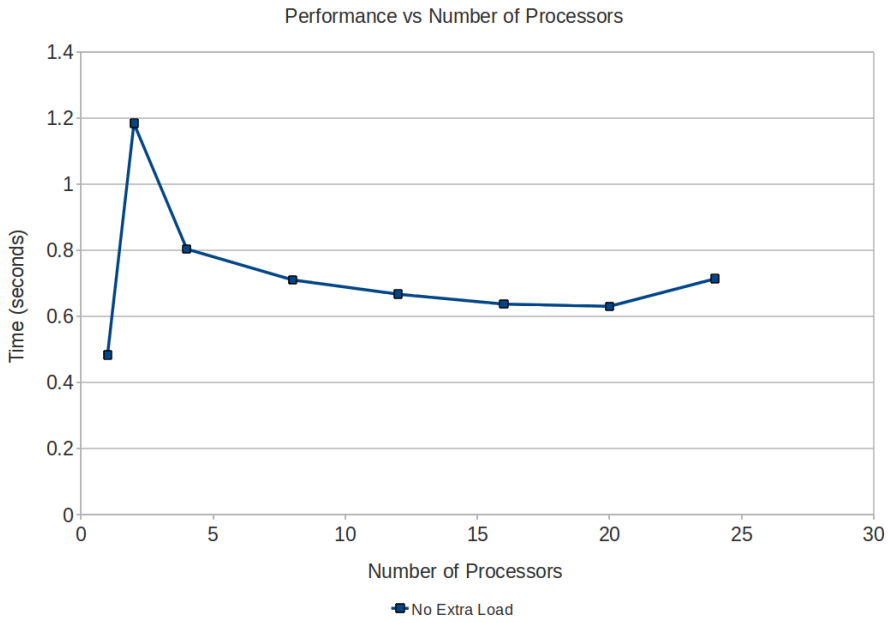

Figure 11: Time vs Number of Processors used under the non-realistic test case.

the time increased. For more nodes, the time decreased again, even though there would be more messages passed per iteration and more possible points of delays. The slowness of the two-node case compared to the many-nodes cases was consistent across many trials; this may be a result of load issues on the machine or of a poor connection. Despite this, the result is clear: parallelism for simple problems decreases performance as compared to the reference implementation. The existence of internode communication slowed the calculation down. This is reasonable since direct memory accesses are very fast compared to the communication time between nodes.

To make the test cases more realistic, a load was added to every potential energy call. A for loop was added to the potential energy function that took about 0.5 seconds on the nodes of the cluster to run. This simulated a realistic test case in which every function evaluation would cost a non-trivial amount of time. The results here match what is expected for an embarrasingly parallel program. See Figure 12 for the data. In Figure 13, it can be seen that the runtimes decreased approximately proportionally to $\frac{1}{n}$ where n is the number of processors used. At a certain point, a performance limit was reached where further

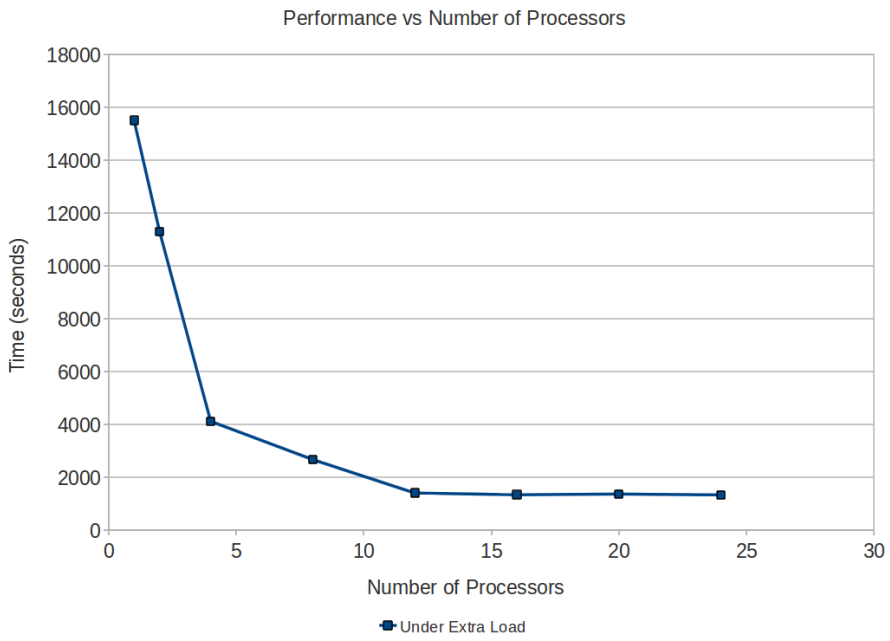

Figure 12: Time vs Number of Processors used under the realistic test case.

parallelization did not improve performance.

5 Conclusions and Future Work

The parallelized NEB method was implemented and benchmarked against standard 2D energy surfaces. The algorithm converged for functions of higher dimensionality, but the accuracy of those paths has yet to be checked. The runtime increased on non-realistic test cases, but on realistic test cases, decreased approximately as $1/n$, demonstrating the effectiveness of parallelizing NEB. An easily modifiable framework also was written for working with alternative chain of states methods. For example, the Plain Elastic Band method is identical to the NEB except for the vector projections used to evaluate the forces. This can easily be tested and it was; the convergence was poor. This is a result of the spring force of the method changing the effective potential energy surface that the images experience—this changed the results nontrivially. The String method only uses the forces from the PES, and artificially reparametrizes the path that its images follow. Other variants of the NEB

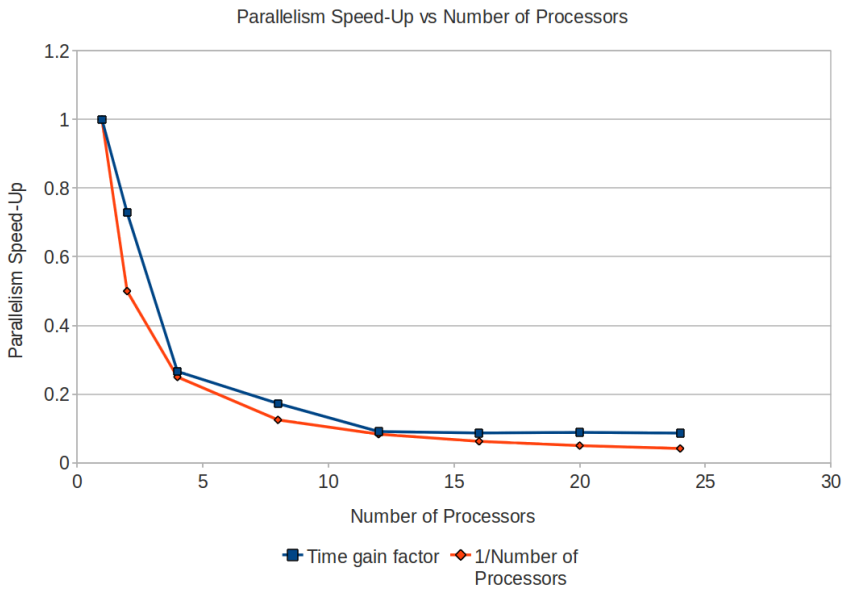

Figure 13: Parallel Speed-Up Factor and $1/n$ on the same axis. The plots are very close, demonstrating the effectiveness of the parallelized NEB method.

method and other methods are easy to implement.

Future work includes making the code work with more complex potential energy surfaces. The code is general in n dimensions, but hasn't been confirmed to converge to the correct MEP in such a space. Adding this makes the code applicable to real chemical problems. The first few steps to achieve this would be to add the Lennard-Jones force

$$V_{LJ} = \epsilon \left[\left(\frac{r_m}{r} \right)^{12} - 2 \left(\frac{r_m}{r} \right)^{6} \right]$$

and to use it for simple cluster rearrangements. The next step with this would be to interface the code with chemistry software packages that compute energies on chemical systems. Parallelism can be used further by making each image the master node for several nodes that carry out that energy calculation. Variants of the NEB method can also be implemented easily. These include the Climbing Image NEB and the Doubly Nudged NEB. Convergence using the Steepest Descent method is slow; a faster Conjugate Gradient, Fast Inertial Relaxation Engine or L-BGFS method might be faster. Initial interpolations other than a simpler

linear interpolation might also be used to determine which path (if several exist) that the MEP converges to as well as to improve the runtime. The nonlinear interpolation on the Wolfe-Quapp Surface was close to its MEP and converged very quickly.

6 Acknowledgements

I'd like to thank graduate students Jeff Bezanson and Laken Top as well as Prof. Alan Edelman and Prof. Troy Van Voorhis.

References

[1] Sheppard, Daniel, Rye Terrell, and Graeme Henkelman. 2008. Optimization methods for finding minimum energy paths. The Journal of chemical physics 128, no. 13: 134106.

[2] Jonsson, Hannes, Greg Mills, and Karsten W Jacobsen. 1998. Nudged elastic band method for finding minimum energy paths of transitions. Ed. B J Berne, G Ciccotti, and D F Coker. Classical and Quantum Dynamics in Condensed Phase Simulations Proceedings of the International School of Physics: 385-404. http://eproceedings.worldscinet.com/9789812839664/9789812839664_0016.html.

[3] E, Weinan, Weiqing Ren, and Eric Vanden-Eijnden. 2002. String Method for the Study of Rare Events. Physical Review B 66, no. 5: 4. http://arxiv.org/abs/cond-mat/0205527.

[4] Neria, Eyal, Stefan Fischer, and Martin Karplus. 1996. Simulation of activation free energies in molecular systems. The Journal of Chemical Physics 105, no. 5: 1902-1921. http://link.aip.org/link/JCPSA6/v105/i5/p1902/s1&Agg=doi.

[5] Sheppard, Daniel, and Graeme Henkelman. 2011. Letter to the Editor Paths to which the Nudged Elastic Band Converges. Physical Review B, no. 3.

[6] Henkelman, Graeme, and Hannes Jonsson. 2000. Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points. The Journal of Chemical Physics 113, no. 22: 9978. http://link.aip.org/link/JCPSA6/v113/i22/p9978/s1&Agg=doi.