Energy Optimization of a Diatomic System

Jane Hung

May 25, 2012

Contents

1 Introduction

Over the past few decades, computers have been used to apply quantum theory on chemical systems [1]. The impact of these calculations reaches many parts of physics, chemistry, and biology. Popular research topics that often incorporate these calculations include nanomaterials, quantum dots, condensed matter, and protein folding. To better understand system properties or particular observed phenomena, the first step is often to find the ground state geometry, the geometry at which the energy of the system is at a minimum on the potential energy surface.

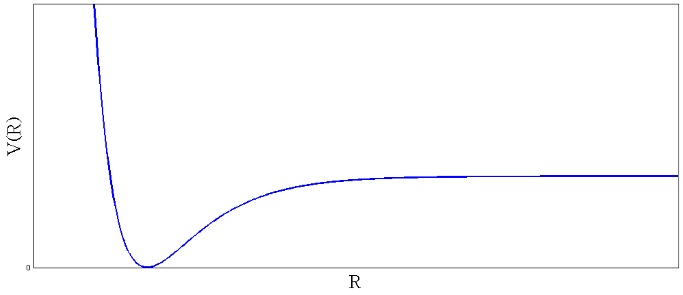
Quantum chemistry calculations that attempt to determine potential energy surface stationary point structures rely on geometry optimization [2-4]. Current geometry optimization methods generally use a quasi-Newton approach. For this method, the next step on a potential energy surface is determined by Newton-Raphson using the gradients and Hessians that are originally calculated in Cartesian coordinates [3]. However, retrieving and manipulating Hessian matrices can make calculations costly, so an energy optimization method that bypasses the need for them is desirable. Here we describe our curve fitting method of optimization for a diatomic system that does not require energy derivatives.

Energies of diatomic systems are well modeled by the Morse potential. For a system with an unknown minimum energy, the general form for the Morse potential energy equation in terms of distance between atoms R is

$$V(R) = D(1 - e^{-a(R-R_e)})^2 + C \tag{1}$$

where $a = \sqrt{F_c/(2D)}$, D is a constant called the well depth, F_c is the force constant, R_e is the equilibrium bond distance between the two atoms, and C is the minimum energy. D, F_c and C are in atomic units (Hartrees) and R_e is in angstroms.

Figure 1: Morse Potential



2 Method Overview

2.1 Newton-Raphson

For the simple Newton-Raphson approach, we start with an initial guess for the ground state geometry $\mathbf{R}_0$. Since the energy function is at least twice continuously differentiable, we can use Taylor's theorem to take an expansion around $\mathbf{R}_i$, a possible geometry. For a two atom system, it is convenient to take one atom as fixed at $(0,0,0)$ and the other to be $(R_x, R_y, R_z) = \mathbf{R}$. Here we define $R = |\boldsymbol{R}|$. Taking the second order Taylor expansion of energy as a function of $\boldsymbol{R}_{i+1}$, we have

$$E(\boldsymbol{R}_{i+1}) = E(\mathbf{R}_i) + (\mathbf{R}_{i+1} - \mathbf{R}_i)^T \boldsymbol{g} + \frac{1}{2}(\mathbf{R}_{i+1} - \mathbf{R}_i)^T \boldsymbol{H}(\mathbf{R}_{i+1} - \mathbf{R}_i) + r_1 \quad (2)$$

where r_1 is the remainder, and $\boldsymbol{g}$ and $\boldsymbol{H}$ are the gradient and Hessian, respectfully, evaluated at $\boldsymbol{R}_i$. Generally, energy functions are well approximated by quadratic functions when R_i is reasonably close to the minimum R_e, so we approximate r_1 as 0. Because we consider the function to be nearly quadratic, we can further simplify the calculation by making $\boldsymbol{H}$ constant. We would want $\mathbf{R}_{i+1}$ to be the minimum, therefore we find the step that needs to be taken from $\boldsymbol{R}_i$ to $\boldsymbol{R}_{i+1}$ in order for the gradient at $\boldsymbol{R}_{i+1}$ to be $\mathbf{0}$ (indicating a critical point). The gradient of E at $\boldsymbol{R}_{i+1}$ is

$$\mathbf{0} = \boldsymbol{g} + \boldsymbol{H}(\boldsymbol{R}_{i+1} - \boldsymbol{R}_i)$$

so

$$\Delta\boldsymbol{R} = -\boldsymbol{H}^{-1}\boldsymbol{g}. \quad (3)$$

The energy function not truly being a quadratic means that R_{i+1} is only an approximation of R_e, but we can iterate using (3) to get better approximations. Starting with $\boldsymbol{R}_0$, we can find $\boldsymbol{R}_1$, then $\boldsymbol{R}_2$, etc.; when $|\boldsymbol{R}_{i+1} - \boldsymbol{R}_i|$ is below a desired tolerance, the iteration ends, and we say that $\boldsymbol{R}_{i+1}$ is the optimal geometry found by the Newton-Raphson method.

2.2 New Approach: Least Squares Fitting

In our new approach, we take advantage of our knowledge of the Morse potential and its applicability to the energy of diatomic systems. For a given diatomic system, though the analytical formula for the potential is unknown, we can assume that it closely follows the Morse potential in form and adjust the parameters until a good fit is achieved.

Least squares analysis is a very common tool for finding the best fit curve to a set of data, where there are more data points than unknown parameters. A function $f(x, \gamma)$ is defined with parameters $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_n)$, and the observed data points are $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$. (The x_i and/or y_i may be vectors.) The objective is to find the $\boldsymbol{\gamma}$ that minimizes the sum of errors squared:

$$\min_{\boldsymbol{\gamma}} \sum_{i=1}^{m} (y_i - f(x_i, \boldsymbol{\gamma}))^2. \quad (4)$$

2.2.1 Trust-Region-Reflective Algorithm

The optimization algorithm used for the least squares minimization is the Trust-Region-Reflective (TRR). This is one of the algorithm options for least squares fitting in MATLAB, the others being Levenberg-Marquardt and Gauss-Newton, and the MATLAB implementation is what is described here [5-7]. Though the other algorithms can act as a replacement for TRR, in our experience TRR gives more accurate and less costly results. The reasoning behind the TRR approach is as follows: for the minimization of a function $u(x)$ (x could be a vector), in order to find a point x_{i+1} with a smaller function value than the current point x_i, u can be reasonably approximated by a quadratic (from its Taylor expansion around x_i) in the neighborhood N of x_i. This N is called the trust region, and the improved point x_{i+1} should also be in this region. The step $s_i = x_{i+1} - x_i$ is found by approximately solving the equation

$$\min_{s \in N} \psi_i(s), \tag{5}$$

where

$$\psi_i(s) = \boldsymbol{g}^T s + \frac{1}{2} s^T \boldsymbol{H} s, \tag{6}$$

and $\boldsymbol{g}$ and $\boldsymbol{H}$ are the gradient and Hessian, respectfully, of u evaluated at x_i. In particular, $s \in N$ is equivalent to $||D_i s|| < \Delta_i$ where D_i is a scaling matrix and Δ_i is the positive trust region size. Notice ψ_i can be thought of as the predicted value of Δu.

In order to solve for s_i more quickly and easily, we restrict the neighborhood N to a 2 dimensional subspace V. A good choice for V would induce global convergence and fast local convergence. In our algorithm, we choose V to be spanned by $\boldsymbol{v}_1$, the vector in the direction of the gradient $\nabla u(x_i)$ (negative of the steepest descent direction) and $\boldsymbol{v}_2$, an approximate Gauss-Newton direction. This Gauss-Newton direction is the solution $\boldsymbol{v}_2$ to

$$\boldsymbol{H}\boldsymbol{v}_2 = -\boldsymbol{g}.$$

Considering that our nonlinear least squares problem can be written in the form

$$\min_x \sum_{i=1}^{n} z_i^2(x) = \min_x \sum ||\boldsymbol{Z}(x)||^2$$

$$\text{where } \boldsymbol{Z} \text{ is the vector field } \boldsymbol{Z}(x) = \begin{pmatrix} z_1(x) \\ z_2(x) \\ . \\ . \\ . \\ z_n(x) \end{pmatrix},$$

the approximate Gauss-Newton direction can be found more efficiently by finding the solution $\boldsymbol{v}_2$ to

$$\min_{\boldsymbol{v}} ||\boldsymbol{J}\boldsymbol{v} + \boldsymbol{Z}||^2,$$

where $\boldsymbol{J}$ is the Jacobian of $\boldsymbol{Z}$. This is equivalent to solving the normal equations

$$\boldsymbol{J}^T\boldsymbol{J}\boldsymbol{v} = -\boldsymbol{J}^T\boldsymbol{Z},$$

which we do with the preconditioned conjugate gradient (PCG) method. PCG uses a preconditioner matrix $\boldsymbol{P}$ that makes the equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ easier to solve numerically by turning it into $\boldsymbol{P}^{-1}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{0}$.

Now that the subspace V is known, (5) becomes a two dimensional problem, and so the minimization is more easily done.

After solving for s_i, if it is such that $u(x_i + s_i) < u(x_i)$, then x_{i+1} becomes the current point; otherwise x_i remains the current point but Δ_i is decreased, and we solve for s_i once again.

Because this is an unbounded problem, D_i is always the identity matrix.

The trust region size Δ_i is adjusted whenever the step size is rejected. So, if $u(x_i + s_i) < u(x_i)$, then the step size remains the same as before ($\Delta_{i+1} = \Delta_i$); otherwise, make the new step size $\Delta_{i+1} \in (0, \tau\Delta_i]$ given some $0 < \tau < 1$.

In summary, the TRR algorithm involves the following steps:

1. Consider the problem of finding the trial step size s_i between the ith point x_i and the next point x_{i+1} that minimizes $\psi_i(s) = \boldsymbol{g}^T s + \frac{1}{2}s^T\boldsymbol{H}s$
2. Compute the 2 dimensional subspace V with 2 spanning vectors: one vector in the direction of the gradient at x_i and one in the approximate Gauss-Newton direction
3. Solve for trial step s_i of the 2 dimensional subproblem
4. If for a predefined constant $0 < \tau < 1$, $u(x_i + s_i) < u(x_i)$, then x_{i+1} becomes the current point; otherwise $x_{i+1} = x_i$
5. Update Δ_i
6. If $\nabla u(s_i)$ is below a chosen tolerance, the algorithm ends; otherwise, repeat and increment i

Some notable convergence properties of this algorithm are global and local quadratic convergence.

2.2.2 Trust-Region-Reflective Algorithm with Least Squares

For the purpose of least squares minimization, the u in the TRR method is a function of $\boldsymbol{\gamma}$, in particular

$$u(\boldsymbol{\gamma}) = \sum_{i=1}^{m}(y_i - f(x_i, \boldsymbol{\gamma}))^2.$$

Since our function is in the form of a Morse potential, we set

$$f(x_i, \boldsymbol{\gamma}) = \gamma_1 \left(1 - e^{-\sqrt{\frac{\gamma_2}{2\gamma_1}}(x_i - \gamma_3)}\right)^2 + \gamma_4.$$

The vector of parameters $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3, \gamma_4) = (D, F_c, R_e, C)$ which are the constants from equation (1), and $x_i = R$, the distance between the two atoms.

It is relatively inexpensive computationally to find the energy of a system for a given configuration $\boldsymbol{R}_i$, so the bond lengths and energies can be thought of as the observed data points: configuration $\boldsymbol{R}_i$ with bond length x_i would have energy y_i.

3 Implementation

The Newton-Raphson method was implemented with code written by the author, however the method of least squares fitting with TRR was implemented by the MATLAB function 'lsqcurvefit'.

For each of these methods, we analyze a case study using the respective algorithms. From previous work the formula for the energy of HCl in the form of a Morse potential is known and has parameter values $D = 0.3459$, $F_c = 4.9600$, $R_e = 1.2934$, $a = \sqrt{\frac{F_c}{2D}} = 2.6776$. We use the formula to calculate the numerical values needed for our methods. In addition, we compare results from the following implemented methods with the known values. All the tolerance values were set at 10^{-6}. The number of iterations taken for each calculation was limited to 10^4 in order to cut down on computation time, so in those cases the final geometry output might not fulfill the tolerance condition. A calculation with a number of iterations of that order would not likely reach the optimal solution in any case, but even if it did, the extremely large number of iterations would still render it unhelpful for actual implementation.

In the following results, if R_f is the equilibrium bond length found from the algorithm, we refer to accuracy as being $|R_f - R_e|$.

We use MATLAB version 7 to carry out the calculations.

3.1 Newton-Raphson

3.1.1 Ideal Morse Potential

Gradient and Hessian numerical values were taken from the quadratic approximation of this Morse potential. In terms of the coordinates of the mobile atom $R = (x, y, z)$, we took gradient

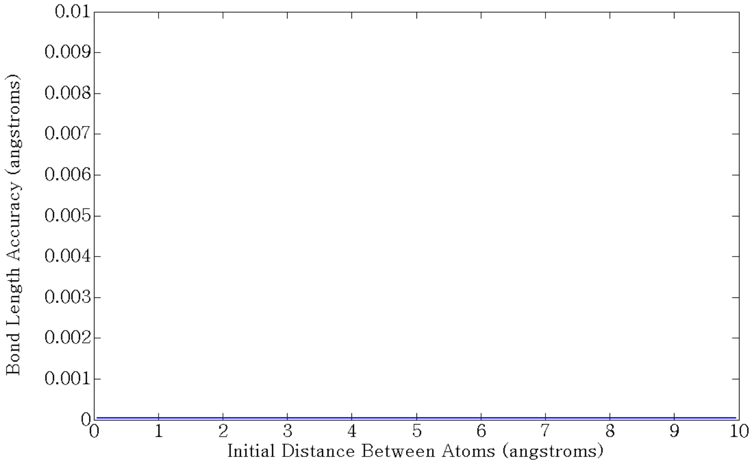
$$\boldsymbol{g} = 2Da^2R(R - R_e)\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

and Hessian

$$\boldsymbol{H} = 2Da^2\boldsymbol{I}, \text{with } \boldsymbol{I} \text{ being the 3x3 identity matrix,}$$

which are the gradient and Hessian in Cartesian coordinates found from the second order Taylor expansion. Notice $\boldsymbol{H}$ is a constant matrix, so it does not change even when the iteration step changes.

Figure 2: Accuracy of Newton Raphson for Morse Potential at Various Initial Atom Distances



Incrementing the distance between the atoms by 0.1 angstroms for each calculation, we found the accuracy of the final geometry output from the algorithm as well as the number of iterations taken.

The accuracy and number of iterations remained constant for every calculation. No matter how far apart the atoms started, it always took 2 iterations to reach a solution, and the solution was equal to the expected equilibrium bond length.

3.1.2 Modified Morse Potential

The potentials of most diatomic systems are not ideal Morse potentials and have some perturbations. To simulate this, we add the Gaussian functions

$$0.05e^{-25(R-1.7)^2} + 0.02e^{-10(R-2.3)^2}$$

to the ideal potential. This function, like the ideal potential, is monotone increasing after the minimum, but the additional Gaussians make the function less similar to a quadratic. Incrementing the distance between the atoms by 0.1 angstroms for each calculation, we found the accuracy of the final geometry output from the algorithm as well as the number of iterations taken.

Figure 3: Iterations of Newton Raphson for Morse Potential at Various Initial Atom Distances

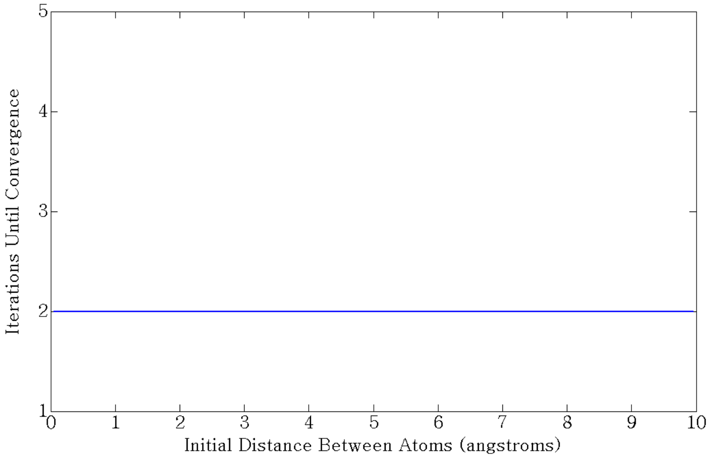


Figure 6: Iterations of Newton Raphson for Modified Morse Potential at Various Initial Atom Distances

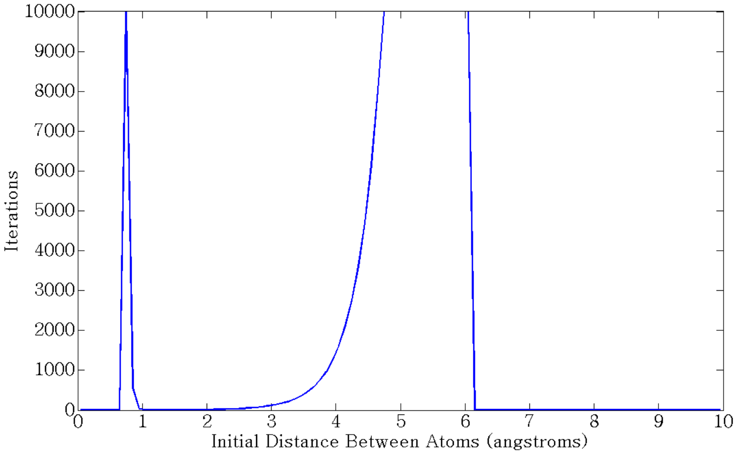


Figure 4: Modified Morse Potential

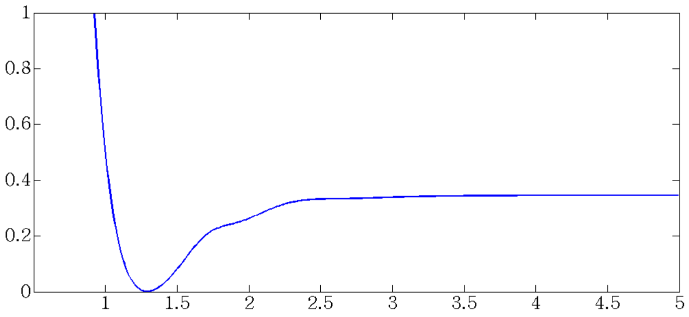
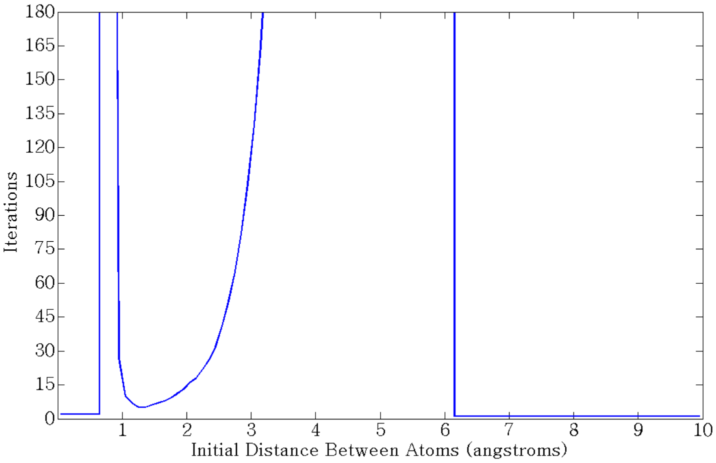


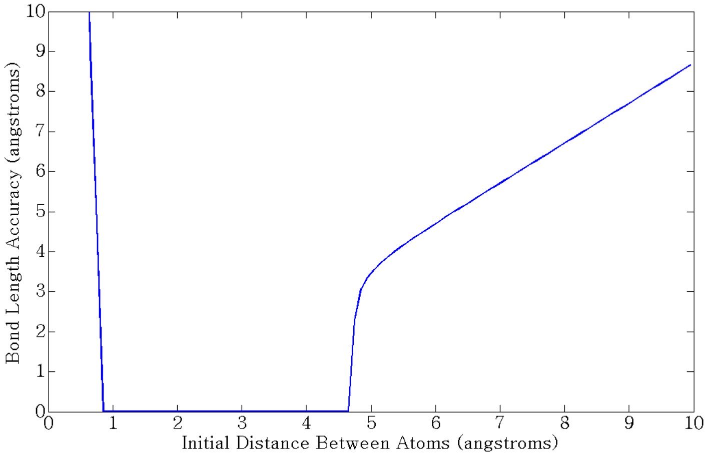
Figure 7: Iterations of Newton Raphson for Modified Morse Potential at Various Initial Atom Distances (Zoomed In)



The accuracy matches that of the ideal potential (Figure 2) for initial distances between about 0.8 to 4.7 angstroms, however for initial distances outside that range, the accuracy greatly suffers. The almost vertical drop down to 0 around 0.8 angstroms indicates the algorithm is very inaccurate for initial distances below 0.8 anstroms. For initial distances above 4.7 angstroms, the algorithm is also inaccurate. Given the normal equilibrium bond lengths, it is fair to say any accuracy value above 1 angstrom already makes the solution useless, so only the solutions found with initial distances in the 0.8 to 4.7 angstrom range would be pertinent.

For some starting distances, the algorithm's iterations were over 10^4, while for others, there were less than 10 iterations. In the regions just below 1 angstrom and between 4 and 6 angstroms, the number of iterations spiked up

Figure 5: Accuracy of Newton Raphson for Modified Morse Potential at Various Initial Atom Distances



by many orders of magnitude. Although after the 6 angstrom mark the iterations were minimal, this is inconsequencial due to the inaccuracy of the solution as discussed above. Looking at the acceptable initial distance range of around 0.8 to 4.7 angstroms, it can seen that the number of iterations reached a local minimum at around 1.2 angstroms (7 iterations) but grew precipitously after that point.

3.2 Least Squares Fitting with TRR

One important issue in the implementation of this algorithm is the set of points that should be taken as the observed data. Since there are 4 parameters we are trying to find with this algorithm, the minimum number of data points that can be used is 4. We would like to choose the minimal number of points that give the most useful information possible about the function. In this paper, we pick fixed values for R and use the energy function with the known parameters to find the corresponding energy values $V(R)$. It is useful to consider that in real diatomic systems, bond lengths range from about 0.7 angstroms to 4 angstroms. Some prominent features of the potential are the energy's increase to infinity as R decreases to 0, the well shape around the minimum, the asymptotic behavior for sufficiently large R values to some constant value (in the notation of (1), the value is $D + C$). By choosing points that capture these features, we have a better chance of fitting the function accurately. Here we try several different sets of points and compare the results.

Because the minimum of the Morse potential occurs at $\gamma_3 = R_e$, by finding the parameters we can immediately obtain the equilibrium bond length according to the algorithm. The initial values for the parameters are chosen to be

$\boldsymbol{x}_0 = (1, 1, 1, 0)$. Using the minimum 4 data points, we started out with the set of x values (atom distances) $[0.5, 1, 2, 6]$ and changed each to examine the resulting differences in the parameter values, bond length accuracy, and iterations. Since we wanted one value close to 0 and one far away, 0.5 and 6 were chosen; the other two were chosen because based on the range bond lengths found in real systems, 1 and 2 were likely in the well region and at least somewhat near the minimum. Other sets of x data points with more than 4 points were chosen to investigate the effect of the fitting method using a larger set of data.

3.2.1 Ideal Morse Potential

The results for our fitting method using an ideal Morse potential are shown in Table 1. Additional cases are examples of badly picked data points because not all important regions of the potential are represented.

Table 1: TRR Results on Morse Potential with Different Data Points

#	x data points	D	F$_c$	R$_e$	C	Accuracy	Iterations
1	[0.5, 1, 2, 6]	0.3459	4.9599	1.2935	-0.0000	0.0000	24
2	[0.5, 1.3, 2, 6]	0.3459	4.9599	1.2935	-0.0000	0.0000	43
3	[0.5, 1.2, 2, 6]	0.3459	4.9599	1.2935	-0.0000	0.0000	57
4	[0.3, 1.2, 2, 6]	0.3459	4.9599	1.2935	-0.0000	0.0000	109
5	[0.3, 1.3, 2, 6]	0.3459	4.9599	1.2935	-0.0000	0.0000	86
6	[0.3, 1.3, 2.3, 6]	0.3459	4.9599	1.2935	-0.0000	0.0000	155
7	[0.5, 1, 2, 5]	0.3459	4.9599	1.2935	-0.0000	0.0000	26
8	[0.5, 1, 2, 10]	0.3459	4.9599	1.2935	-0.0000	0.0000	28
9	[0.5, 1, 2, 3, 4, 6]	0.3459	4.9599	1.2935	-0.0000	0.0000	28
10	[0.5, 1, 1.5, 2, 2.5, 3, 4, 6]	0.3459	4.9600	1.2935	-0.0000	0.0000	30
	Additional						
11	[0.3, 0.5, 0.7, 0.9]	0.0000	0.0001	4.7436	-0.6150	Did not converge	
12	[0.3, 0.5, 1.5, 2]	0.0000	0.0001	5.2701	0.2099	Did not converge	
13	[1.5, 2, 3, 4, 6]	8.3615	4.9586	-5.9405	-7.9460	7.2340	44

For every set of data points tested, the algorithm found the correct parameters with the best possible accuracy for the equilibrium bond length. In other words, accuracy was not sensitive to the data points chosen. Since the maximum accuracy was achieved with only 4 points, changing the number of points could not improve accuracy, though having more points at least did not negatively affect it.

Despite every case reaching the same solution, the number of iterations varied. Adding data points led to a slight increase in the number of iterations, but modest changes in even just one or two of the data points sometimes led to double the number of iterations (see $\#1, \#3$ and $\#4$ or $\#1, \#2$ and $\#5$). It can be seen that the data points can greatly affect the number of iterations needed in the algorithm, so the method by which the points are chosen in the first place seems to be important in terms of the computational cost. The large differences in iterations stem from changing the closer points in the well region since changing the farthest point, one on the flat part of the potential, has little effect (see #1, #7 and #8).

For badly picked data points such as #11, #12 and #13, the accuracy is extremely far off, and convergence may be difficult.

3.2.2 Modified Morse Potential

The same sets of data points were used for our fitting model with the modified Morse potential as with the ideal Morse potential above. The results are shown in Table 2. Additional cases are examples of sets with a greater number of data points.

Table 2: TRR Results on Modified Morse Potential with Different Data Points

#	x data points	D	F_c	R_e	C	Accuracy	Iterations
1	[0.5, 1, 2, 6]	0.3144	4.7356	1.2892	0.0315	0.0043	24
2	[0.5, 1.3, 2, 6]	0.3487	5.4233	1.2606	-0.0028	0.0329	45
3	[0.5, 1.2, 2, 6]	0.3342	5.1313	1.2722	0.0117	0.0213	56
4	[0.3, 1.2, 2, 6]	0.3331	5.0725	1.2704	0.0128	0.0231	118
5	[0.3, 1.3, 2, 6]	0.3494	5.3867	1.2564	-0.0035	0.0371	137
6	[0.3, 1.3, 2.3, 6]	0.3856	6.9739	1.1679	-0.0396	0.1256	175
7	[0.5, 1, 2, 5]	0.3144	4.7354	1.2892	0.0315	0.0043	25
8	[0.5, 1, 2, 10]	0.3144	4.7356	1.2892	0.0315	0.0043	23
9	[0.5, 1, 2, 3, 4, 6]	0.3134	4.7262	1.2893	0.0321	0.0042	26
10	[0.5, 1, 1.5, 2, 2.5, 3, 4, 6]	0.3275	4.8425	1.2900	0.0210	0.0035	31
	Additional						
11	[0.3, 1.3, 2, 2.5, 3, 4, 5, 6]	0.3516	5.4740	1.2507	-0.0049	0.0428	130
12	[0.3, 1.3, 1.6, 2, 2.2, 2.5, 3, 6]	0.3576	5.8700	1.2232	-0.0122	0.0703	175
13	[0.3, 1.6, 2, 2.2, 2.5, 3, 4, 6]	0.3039	4.6535	1.2830	0.0432	0.0105	196
14	[0.3, 1.3, 2.3, 3, 4, 5, 6]	0.3877	7.0640	1.1639	-0.0428	0.1296	189
15	[0.3, 1.3, 2, 2.3, 3, 4, 6]	0.3538	5.5961	1.2422	-0.0071	0.0413	145

The accuracy in all cases was nonzero, so the modification of the potential led to worsened accuracy. Accuracy ranged from around 0.004 (see #1, #7, #8, #9, #10) to 0.1256 angstroms (see #6), and all the equilibrium bond distance estimates underestimated the true value (although this is likely due to the data points chosen and not a property of all solutions). Here, accuracy was sensitive to the data points chosen, and even modest changes could lead to almost 10 times worse accuracy (see #1 vs. #2). Changing the points that were in the well region seemed to affect the accuracy the most, while changes to the farthest point (on the flat region of the potential) yielded little or no change (see #1, #7 and #8). It also appears that changing the point closest to 0 angstroms does not have a large effect (see #3 vs. #4 and #2 vs. #5). The sets with more data points showed similar accuracy as the sets with 4 data points. In many instances, adding points to a data set actually worsened accuracy (e.g. #12, #15 and #11 vs. #5).

For the most part (the exception being #5) the number of iterations for each data point set remained roughly the same. Compared to the corresponding results in Table 1, the iterations of #5 increased 60%, while the iterations of the rest differed by 13% or less. Also, observations made about iterations for Table 1 largely apply here as well: the data points can greatly affect the number of iterations needed in the algorithm, but changing the farthest point, one on the flat part of the potential, has little effect. Sets with more data points sometimes had a larger number of iterations (e.g. #14 vs. #6) but other times had a smaller number (e.g. #11 vs. #5). Although the cases with the smallest number of iterations also had the best accuracy, a greater number of iterations does not necessarily mean worse accuracy (see #4).

Sidenote: Another modification that may be done to the Morse potential is to alter D and F_c. This is a good for testing how well this method can handle different systems. Here, we set $D = 0.1$ and $F_c = 0.1$.

Table 3: TRR Results on Morse Potential with Different Parameters

x data points	R_e	Accuracy	Iterations
[0.5, 1, 2, 6]	1.2935	0.0000	26
[0.5, 1.3, 2, 6]	1.2935	0.0000	33
[0.5, 1, 2, 3, 4, 6]	1.2935	0.0000	26
[0.5, 1, 1.5, 2, 2.5, 3, 4, 6]	1.2935	0.0000	26

For the points calculated, the accuracy is the same as in Table 1, and the number of iterations is similar.

4 Discussion

For an ideal Morse potential, both Newton-Raphson and TRR gives equally accurate results. An advantage of Newton-Raphson in this case is that it is very consistent: it can give the correct solution in 2 iterations with virtually any initial geometry. On the other hand, our fitting method requires a set of data points, which are the energies at certain bond distances, and some guess work is involved in deciding the best points to use. It is thus more straightforward in this case to use the Newton-Raphson algorithm.

The modified Morse potential gives more complex results. There are many issues that may come up if Newton-Raphson is used; in order to get accurate results, the initial atom distance, which is usually a guess, must be in a certain range. From Figure 1, we see even starting at a point just outside that range can lead to a totally inaccurate solution; secondly, even given an initial distance in the correct range, the number of iterations that the algorithm takes can vary by several orders of magnitude, so the calculation cost may be impractical. Although the former would not be a major issue in a diatomic system because the range covers most of the common equilibrium bond lengths, if this were used for a more complicated system where intuition cannot be used to make a good guess, then this problem might arise. The latter problem might be dealt with by increasing the tolerance, but that also adds some inconvenience. In contrast, our fitting model gives solutions on the order of 10^{-1} angstroms or less within the true equilibrium bond distance, and the iterations are on the order of 10 or 10^2. In order to work well, the model does require data points from different regions of the potential, but some of the guesswork that goes into picking the points can be alleviated by choosing more than the necessary number of points within a wide range (e.g. 0.5 to 8 angstroms). The user's tolerance for inaccuracy would differ depending on the situation, but more research should be done to determine whether there is a bound for the accuracy and if so, under what conditions it can be changed. Both methods have unusable results if initial conditions do not conform to certain rules, but our fitting method can get around this by having enough data points over a wide range.

The range of accuracy of the fitting method may be too large for some situations, but determining the best answer is an issue since there are many different data points that can be used to get varying solutions. Two immediate techniques to solve this problem are 1) redo the fitting with more data points, and 2) redo the fitting multiple times with different data point sets each time. From the data presented in Table 2, it does not appear that option 1 is very effective. Adding more points often leads to worse accuracy and/or iterations, and there doesn't seem to be an obvious pattern that indicates which will be worse or by how much. Though it may be a problem of choosing the points more strategically, there is no clear method for choosing the additional points that would be best so more development would be needed. It's also possible the algorithm would be very accurate with a data set with more points than were tested here, and in that case this would be a good option to consider. For option 2, there is a problem of how to use all the different solutions to get to one solution. They could be averaged, but as we saw in Table 2, most solutions could underestimate or overestimate the real value, and so the average would as well. Also, the cost of doing this for many sets of data would probably be greater than fitting a larger set of data only once.

5 Conclusion

We studied two methods for finding the minimum energy geometry of a diatomic system. Diatomic systems in general follow the Morse potential energy, so we use that for the development and application of the methods. The first was Newton-Rapshon, which is an iterative method that uses the gradient and Hessian to find the next geometry step. The other was one we developed that uses least squares fitting to fit a set of data points and find the best Morse potential parameters. Using information about the Morse potential for HCl, we compared equilibrium bond length results of the Newton-Rapshon method and our new fitting method. Although Newton-Raphson performs extremely well for virtually all starting configurations if the potential is an ideal Morse potential, it performs much more poorly with a slightly perturbed Morse potential. The initial atom distance must be within a certain range in order to for the algorithm find a useful solution, so it is not very robust. However, our fitting method gives reasonable results without very large computational cost given data points that span a suitably wide range of distance values. Another advantage to the fitting method is that it is not necessary to compute or use energy derivatives, which can be troublesome. With further development, this new fitting method may be a useful alternative to the current Newton or quasi-Newton optimization methods that need to use and manipulate gradient and Hessian matrices.

References

[1] Mueller, M. R., Fundamentals of quantum chemistry: molecular spectroscopy and modern electronic structure computations, Springer, 2001.

[2] Burger, S. K.; Ayers, P. W., Quasi-Newton parallel geometry optimization methods, J. Chem. Phys. 133, 034116 (2010)

[3] Pulay, P.;Fogarasi, G., Optimization in redundant internal coordinates, J. Chern. Phys. 96 (4), 15 February 1992

[4] Liang, W., et. al. Eigenspace Update for Molecular Geometry Optimization in Nonredundant Internal Coordinate, J. Chem. Theory Comput., 6, 2034-2039, 2010

[5] Coleman, T.F. and Y. Li, "An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds," SIAM Journal on Optimization, Vol. 6, pp. 418-445, 1996.

[6] Mathworks. R2012 Documentation. Optimization Toolbox- Least Squares (Model Fitting) Algorithms. http://www.mathworks.com/help/toolbox/optim/ug/brnoybu.html

[7] Branch, M; Coleman, T.F.; Li, Y. "A Subspace, Interior, and Conjugate Gradient Method fo Large-scale Bound-Constrained Minimization Problems" SIAM J. SCI. COMPUT. Vol. 21, No. 1, pp. 1-23, 1999