



Eindhoven University of Technology
Department of Mathematics and Computer Science
Center for Analysis, Scientific computing and Application

Creation of a Nudged Elastic Band method to determine the activation energy using least-energy reaction paths

Thijs Vernooij
1424890

Supervisors:
Björn Baumeier

This report follows the TU/e Code of Scientific Integrity

Eindhoven, 16-04-2024

Abstract

Chemical reactions are a very important part in many industries so the research in the characteristics is equally important. Therefore the goal of this report is to create a model where the Nudged Elastic Band method can be applied to find quantum chemistry reaction paths as to determine the least-energy reaction path to find the activation energy of said reaction.

The report begins by first explaining the theory of the Nudged Elastic Band method. This is done by going over each part of the method in a bottom up manner where everything culminates in the final method. The main blocks in which it is discussed are the images, the initial guess, the forces, the integration and the iteration, which are all discussed in enough detail for a reader to create a similar model. Then the Hartree-Fock method is discussed, as this will be the method with which the energy of the atomic structures will be calculated. This is very important as the energy of the system is a main part of the Nudged Elastic Band method.

Then the application of all this in the final code is discussed. This begins with a short overview of the final code and the main framework in which it works. All major functions which are meant for user interaction are discussed briefly to acquaint the reader with their functionality. A short piece is also dedicated to the explanation of various conversion functions which are used for compatibility of the model with PySCF. This Python library is heavily used to determine the energy and gradient levels of the molecular structures with the Hartree-Fock method as discussed in the theory.

Then several pitfalls are discussed which may severely impact the performance of the system. The two most important things to avoid of these is the mis ordering of atoms and ridge walking during the simulations. Both of these can impact the results although they are relatively easily circumvent able.

We finally present numerical results which show that our model is not extremely accurate. Several possible explanations for this are given but a reason for the current inaccuracies could not be determined. This results in situations where the model is 60% off compared to the known experimental values. Possible model accuracy improvements are suggested. For now, therefore, it is not recommended to use the model for the determination of the least-energy reaction path to find the activation energies of more complex reactions. The model appeared to produce reliable results for reactions were symmetries and such were used to reduce the amount of dimensions involved in the final system.

3.3	Results	22
3.3.1	Tri-atomic reaction	23
3.3.2	C3H6	23
3.3.3	C2O2H4	25
4	Conclusion & Discussion	27
A	Animations	30

Nomenclature

Name	Domain	Description
λ	$\mathbb{R}^D \rightarrow \mathbb{R}$	Reaction path
L	\mathbb{R}	Length of reaction path
D	\mathbb{N}	Number of dimensions
V	$\mathbb{R}^D \rightarrow \mathbb{R}$	Potential Energy Surface
N	\mathbb{N}	Number of images
\mathbb{R}^D	-	The state space
S_i	\mathbb{R}^D	The i -th image in the state space
v_i	\mathbb{R}^D	The velocity of the i -th image in the state space
k_i	\mathbb{R}	Spring constant of spring between S_i and S_{i+1}
F_g	$\mathbb{R}^D \rightarrow \mathbb{R}^D$	The opposite gradient force
F_s	$\mathbb{R}^D \rightarrow \mathbb{R}^D$	The spring force
f	$\mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$	Proportion of perpendicular spring force used
Δt	\mathbb{R}	Time step
v_i	\mathbb{R}^D	Velocity of the i -th image
a_i	\mathbb{R}^D	Acceleration of the i -th image
F_i	\mathbb{R}^D	Resulting force on the i -th image
θ	\mathbb{R}	Convergence parameter
L^2	-	The space of all square-integrable normalized functions
\hat{H}	$L^2 \rightarrow L^2$	Hamiltonian operator
ψ	L^2	Many-electron wave function
ψ_{Slater}	L^2	Many-electron wave function based on slater determinant
χ	L^2	Single-electron wave function
ϕ	L^2	Basis set function
E	\mathbb{R}	Energy of a quantum system
K	\mathbb{N}	Number of electrons
M	\mathbb{N}	Number of nuclei
M_A	\mathbb{R}	Mass of nucleus A
Z_A	\mathbb{R}	Charge of nucleus A
r_{ij}	\mathbb{R}	Distance between particle i and j
U_i	$L^2 \rightarrow \mathbb{R}$	The operator of the average of the electron-electron interactions on electron i
\mathbb{A}	-	The set containing all atoms
\mathbb{B}	-	The set containing all bases
ρ	$\mathbb{R}^D \rightarrow \mathbb{R}$	Sum of charge densities of single-electron wave functions
ξ	$\mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$	Exchange function
R	\mathbb{R}	Resolution of the plot

1. Introduction

Many processes exist in our current world, each shaping a different part of our lives. Examples of these can be everywhere, from electricity charging our devices to gravity which keeps us with both feet on the ground. Another class of fundamental phenomena are the chemical reactions that give shape to many processes vital not only to our daily life but to our society as a whole [“Chemistry is everywhere - American Chemical Society”, 2024].

One important characteristic of chemical reactions is the rate at which they occur. The rate makes the difference between a reaction releasing gasses in a controlled manner such as a campfire releasing carbon dioxide and water vapor, and a large explosion used in the mining industry releasing extremely large amounts of gas in instants. Therefore, many methods have been constructed to determine the reaction rate [Bartsch et al., 2012; Malmstadt et al., 1972; Pollak and Talkner, 2005].

Several of these methods are based on physical experiments, and while this gives many advantages, it also has many disadvantages. Not every reaction permits for an experimental determinatoin of the reaction rate.

Thus, instead of conducting physical (costly) experiments one can resort to theoretical approaches to derive the reaction rate. For instance, one can use the Arrhenius equation [Laidler, 1984]. It gives a connection between the reaction rate and the activation energy, which results in the activation energy also being a very important parameter in the reaction and thus shifting the parameter that needs to be determined from the reaction rate to the activation energy.

In this report, we derive this activation energy using the least energy reaction path involved with teh reaction. If this path is known, then the energy of each point along the path can be calculated using the energy function of the system.

There are several ways in which this reaction path can be determined [Ayala and Schlegel, 1997; Elber and Karplus, 1987; Halgren and Lipscomb, 1977; Simm et al., 2019; Vinu and Broadbelt, 2012]. We select the Nudged Elastic Band method, which is a general method to find the path with the lowest maximum energy between two points [JÓNSSON et al., 1998]. An application will be shown for two molecules and a transition in their structure where the energy of the systems will be determined using the Hartree-Fock method.

The report itself will start with an explanation of the theory and more in detail with the model used. Thereafter the Nudged Elastic Band method is introduced and pros and cons are highlighted. Lastly the theory of the Hartree-Fock method which will be discussed together with additional information regarding bases.

Then we consider our approach for several applications for which we also discuss requirements. Next will several pitfalls be given and how to circumvent them. Then lastly will several results be given, starting with arbitrary mathematical scenarios meant for visualization followed by two quantum-chemistry examples.

We finish with a conclusion and recommendations.

2. Theory

First we describe our model, after which the Nudged Elastic (NEB) Band method is introduced. Thereafter we formulate a surface-level approach of the Hartree-Fock method and finally we connect the Nudged Elastic Band method to a the quantum chemical model.

2.1 Model

We consider a model without random influences and described by a state S which consists of a finite amount, $D \in \mathbb{N}$, of parameters. S is then an element of the space \mathbb{R}^D which contains every variable that makes up the state.

Associated with our model is a Potential Energy Surface (PES) which describes the energy related to the system (of molecules). It is defined as

$$V : \mathbb{R}^D \rightarrow \mathbb{R}. \quad (2.1)$$

To finish with, our model's state changes from the initial state to its final state (both fixed). The transitions between initial and final state is defined by a pathway, which is every intermediate state during the change from the initial state to the final state.

Example

This section will frequently refer to this PES used for this example

$$V_{\text{example}}(x, y) = 2 \cdot \exp(-x^2) + 2 \cdot \exp(-y^2). \quad (2.2)$$

This PES is related to two particles (with position x respectively y) both in the space \mathbb{R}^1 . The state space is \mathbb{R}^2 and the two positions together create a vector (x, y) which represents the system. The two particles do not interact with each other but do interact with a potential $V_{1D}(x) = 2 \cdot \exp(-x^2)$. A figure of the PES can be seen in Figure 2.1. The beginning and end states of the two particles have been chosen as -5 and 5 and 0 and 0 resulting in $(-5, 0)$ and $(5, 0)$.

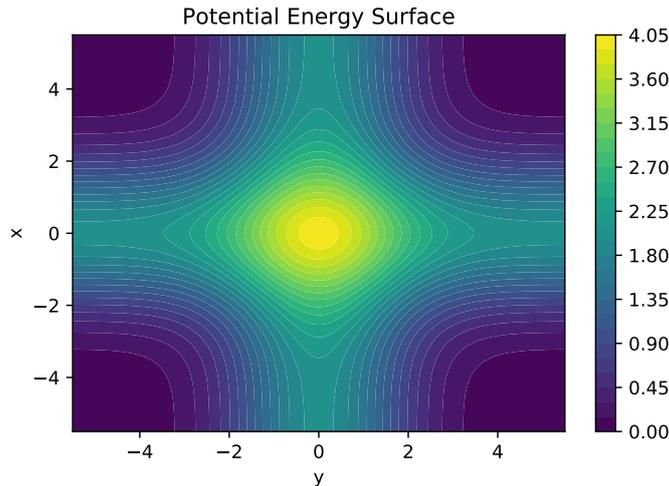


Figure 2.1: The potential energy surface of V_{example} .

2.2 Nudged Elastic Band method

The Nudged Elastic Band method (NEB) is a manner in which a pathway between two different, the beginning and end, states can be determined. The basis lies in the concept of an elastic band

connecting the two states together in the state space with a yet-to-be-explained force nudging it into the correct places to find the result, where the result is defined as the pathway that has the lowest maximum possible energy. Ideally, this 'lowest' maximum is the global minimum for the maximum possible energy, but often it is local.

Then, since every possible pathway shares the same beginning and end point, we can conclude that this result is also the pathway with the lowest activation energy, which is the difference in energy between the beginning state and the final state. This results in this pathway being chosen most often by chemical reactions because it has the highest reaction rate Laidler, 1984.

In practice, the finding of the optimal pathway is an iterative process since forces act on the pathway which leads us to use a force simulation which accounts for velocity by integration steps during each iteration.

Note that the pathway in our model doesn't involve time, only a pathway between two fixed points. How this transition is depended on time is not included in the model. As such, the term initial and end states do not refer to states in time but (fixed) in space. The to be presented method is invariant under a swap of these two states.

The final result of the pathway can be seen in Figure 2.2. The system here is that of the Example of Section 2.1.

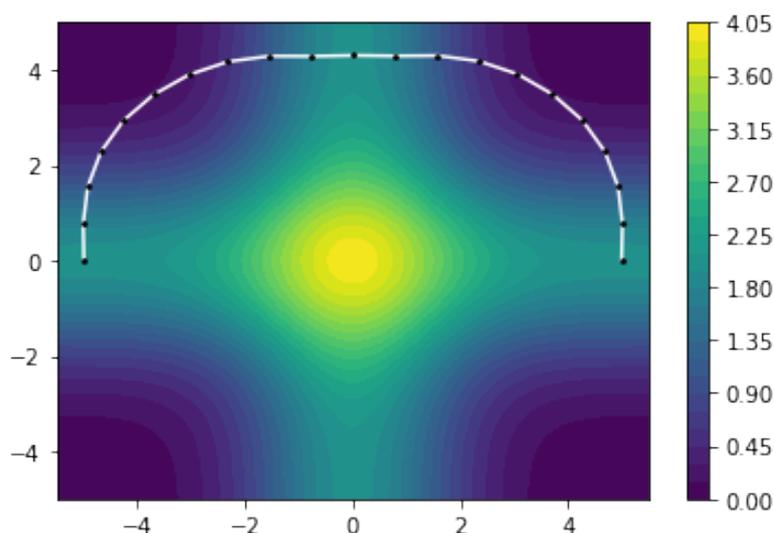


Figure 2.2: The figure shows a contour plot of a Potential Energy Surface, with a solution as found by the NEB method. The white band is the pathway, where the black dots are all the intermediate states.

2.2.1 State space

An important notion in the NEB method is the state space. While the system itself may exist in any continuous space, the space in which the NEB method achieves it results is the space of each possible state of the system.

Chemistry Example

A very classical example of this is a molecule. Water, H_2O , is made of 3 atoms and this system exists in the space \mathbb{R}^3 . However, the total state of the molecule consists of $3 \cdot 3 = 9$ variables since each atom has 3 spatial variables: x,y,z . Therefore, the state space of the molecule is \mathbb{R}^9 .

2.2.2 Images

The equations to be presented are such that it is unlikely that their solution (the path's final shape) can be analytically determined. Keeping this in mind, we discretize the path. First we choose an initial (smooth) pathway between the initial and end points in the state space. Next we choose $N \in \mathbb{N}$ points on this pathway (called images). In this manner each point (between initial and end point) has two nearest neighbors along the path. We denote the initial point with S_1 and the end point with S_N . $N > 2$ for the pathway to be non-trivial.

There exists a natural trade-off: larger N will lead to more computations since more images need to have their forces calculated and positions updated during each iteration.

It is, therefore, a valid idea, to begin with few images and when both resources permit and accuracy demands we scale up.

2.2.3 Forces

Each image will have 2 forces acting on it at each moment. Firstly, the elastic force, imitated by applying a spring between each nearest neighboring image, $F_s : \mathbb{R}^D \rightarrow \mathbb{R}^D$, and the opposite gradient force $F_g : \mathbb{R}^D \rightarrow \mathbb{R}^D$. These 2 together will determine the final resulting force on each image. One important distinction is the role of each of these forces and therefore both forces will be discussed separately.

Spring force

The spring force aims at distribution images evenly along the pathway and to avoid paths with loops or sharp corners ($> 90^\circ$). If the spring forces were not present then all the images would slowly accumulate in the places where the potential is lowest and sharp corners will not be removed.

The reason that sharp corners in the pathway are unwanted is that sharp corners hint at the presence of either loops and/or kinks or in general too few images. We will now discuss a solution to the first problem such that it can be concluded that the number of images should be increased if there are still kinks in the final pathway.

The elastic force on the pathway is modeled by springs between nearest neighbors on the pathway where $k_i \in \mathbb{R}$ denotes the spring constant between images S_i and S_{i+1} . A change to the spring constant will have an inversely proportional effect on the ideal distance between the images connected by the point in comparison to the other distances between the points.

The elastic force on image S_i is given by

$$F_{\text{spring}}(S_{i-1}, S_i, S_{i+1}) = k_{i-1}(S_{i-1} - S_i) + k_i(S_{i+1} - S_i). \quad (2.3)$$

We'll decompose this force into a normal and a tangential component. The parallel line of the pathway at image S_i is defined as the line between image S_{i-1} and S_{i+1} , translated onto S_i . Perpendicular to the pathway at image S_i is then the plane through S_i with as normal the parallel line.

The parallel component of the spring force is then defined as the projection of the force onto the normalized parallel. This component is purely working on the purpose of keeping all images equally distributed over the pathway. The reason for this is that it only acts in the approximate direction of the pathway. It will not point outward. Therefore it is left as it is and not changed.

The perpendicular component of the spring force is then defined as the spring force minus the parallel component. This component works to remove sharp corners, kinks, and loops. However, care should be taken with this function, because if it is left unperturbed, then it will also try to

straighten out any appropriate slow bend in the pathway which should be present. Acknowledging the fact that this perpendicular part inherently gets smaller if the pathway straightens out, the choice was made to add additional scaling. This scaling function is defined as

$$f : \mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R} \quad (2.4)$$

$$f(S_{i-1}, S_i, S_{i+1}) = \begin{cases} \cos\left(-\frac{\pi}{2} \frac{(S_{i-1}-S_i, S_{i+1}-S_i)}{|S_{i-1}-S_i||S_{i+1}-S_i|}\right) & ; \frac{(S_{i-1}-S_i, S_{i+1}-S_i)}{|S_{i-1}-S_i||S_{i+1}-S_i|} \leq 0 \\ 1 & ; else \end{cases} \quad (2.5)$$

which allows for a more cosine-like 'activation' of the perpendicular component when it is or is not needed. The function itself first computes the cosine of the angle between the nearest neighbors of the image i using the inner product. If the angle is larger than 90° then the force is not scaled at all. Otherwise, the result is again fed into a cosine along with some scaling. This second cosine is applied to flatten the curve just a bit more around π to ensure that small bends do not get unnecessarily straightened out.

An example of this entire decomposition of the spring force and a plot of the scaling function can be seen in Figure 2.3

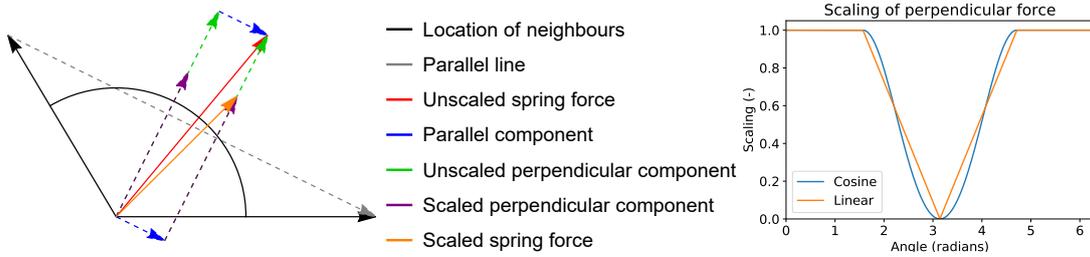


Figure 2.3: The left figure shows a schematic view of the decomposition of the original spring force and the subsequent scaling of the perpendicular component. The black arrows point to both S_{i+1} and S_{s-1} . In the middle can the legend of the left figure be found. The right figure shows the scaling of the perpendicular force as a function of the angle between the vectors. A linear profile is added for reference.

Gradient force

The role of the opposite gradient force is to pull each image toward low-energy areas. It applies a force in the direction opposite to the gradient, as the direction of the gradient is always the direction with the largest (local) energy increase while the images need to go to lower energy. It is thus defined as

$$F_{\text{gradient,temp}}(r) = -\nabla E(r) \quad (2.6)$$

The purpose of this force $F_{\text{gradient,temp}}$ is to move the entire path. To avoid it from interfering with the spring force we only consider its 'perpendicular' component which is what is left after subtracting from it its projection onto the normalized parallel of the spring force.

The final force on S_i is then

$$F_i = F_{i,\text{spring,para}}(S_{i-1}, S_i, S_{i+1}) + F_{i,\text{spring,perp}}(S_{i-1}, S_i, S_{i+1})f(S_{i-1}, S_i, S_{i+1}) + F_{i,\text{gradient,perp}}(S_i) \quad (2.7)$$

where $F_{i,\text{spring,para}}$, $F_{i,\text{spring,perp}}$ and $F_{i,\text{gradient,perp}}$ are the parallel component of the spring force, the perpendicular part of the spring force and the perpendicular part of the gradient respectively.

2.2.4 Initial guess

The NEB method does not inherently provide a method to create an initial guess for the states in the pathway for the iterative procedure. This is needed, however, since the iterative procedure uses the given forces to push the pathway to the optimal pathway, but this needs an initial state. This thus needs to be done separately which does allow for some freedom.

A simple guess for the initial positions is to put every image on the line between the start and end images with equal distance between each. This will generally be appropriate, but there may be moments when this will not work.

Two examples of this are as follows:

- The initial guess goes over a location that has either positive or minus infinity energy. When near either of these infinities the gradient naturally becomes extremely large and thus cause instability in the integration method.
- The initial guess is perfectly on top of a ridge of the PES. This is the case when the gradient of the initial guess is everywhere nearly or completely parallel to the guess. In this case, it may be extremely beneficial to give the initial guess a little 'nudge' such that the gradient descent can occur, because if the initial guess is exactly on the ridge then this will not properly occur.

So while a linear distribution of the images between the start and end point is often possible, it may reveal new answers if a disturbance is added. A random disturbance is viable but may cause images to shift to different sides of the ridge which inhibits the method from working correctly. It is therefore advised to choose a disturbance only in one direction.

Example

Back to the example of Section 2.1: if the initial guess is a linear interpolation between the beginning and end points of $(-5,0)$ and $(5,0)$, as shown in the left figure of Figure 2.4, then the gradient will have no perpendicular component over the entire initial guess and thus stay there which is in contrast to the proper solution which, as seen in Figure 2.2, is not on the ridge. Therefore, a quadratic perturbation was added to the initial guess to create the right figure in Figure 2.4. If the NEB method would be applied there, then the correct optimal pathway would be found.

If various initial guesses give different maximum energy values, then the initial guess with the lowest maximum energy has the lowest activation energy and thus highest reaction rate and we thus proclaime this as the best result.

2.2.5 Integration

Just like the fact that the band needs to be split into images due to the complexity of the system, proper integration of the overarching system, the pathway, the images and the forces acting upon it, is also not possible. Therefore, an iterative numerical approach based on the Velocity Verlet algorithm will be used [Swope et al., 1982]. One property of this method is that it only computes the gradient once per iteration. The reason why this is desirable is because calculating the gradient can take a lot of time, depending on circumstances which will be discussed in 2.3. One less ideal but very manageable disadvantage of this method is the fact that the force of a previous iteration needs to be saved until a calculation after the new force has been calculated.

For the actual method, let us consider the fact that the mass of each image is 1 and let us assume for any given time t_0 we know the location and velocity of and the force on image i : S_{i,t_0} , v_{i,t_0} and

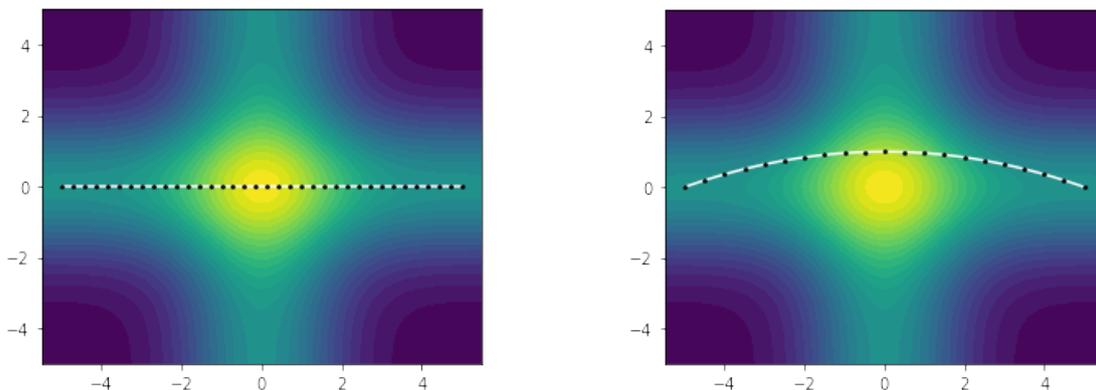


Figure 2.4: Two different possible initial guesses for the same system. The left figure shows a guess that sits on a long saddle point/ridge whereas the right figure has been moved off from it by a perturbation. The right figure will find the correct result when the NEB method is applied, the left will not.

F_{i,t_0} . The i will be dropped from the notation from now on because only image i will be discussed to make it more legible.

The motion of the images is then considered to be Newtonian, then Newton's second law of motion gives that

$$a_{t_0} = F_{t_0} \quad (2.8)$$

where $a_{t_0} \in \mathbb{R}^D$ is the acceleration of image S_i at time t_0 . Then, the second-order Taylor approximation of the position is computed as follows

$$S_{t_0 \pm \Delta t} \approx S_{t_0} + \frac{\partial}{\partial t} S_{t_0} \cdot (t_0 \pm \Delta t - t_0) + \frac{\partial^2}{\partial t^2} S_{t_0} \cdot (t_0 \pm \Delta t - t_0)^2 \quad (2.9)$$

which can be rewritten into, using the earlier mentioned variables and Equation 2.8

$$S_{t_0 \pm \Delta t} \approx S_{t_0} \pm v_{t_0} \cdot \Delta t + F_{t_0} \cdot \Delta t^2 \quad (2.10)$$

which can give

$$S_{t_0 + \Delta t} + S_{t_0 - \Delta t} = F_{t_0} \cdot \Delta t^2 + 2S_{t_0} \quad (2.11)$$

$$S_{t_0 + \Delta t} - S_{t_0 - \Delta t} = 2v_{t_0} \cdot \Delta t \quad (2.12)$$

which together with Equation 2.10 can be rewritten, via algebra and an index shift to

$$v_{t_0 + \Delta t} = v_{t_0} + \frac{\Delta t}{2}(a_{t_0} + a_{t_0 + \Delta t}) \quad (2.13)$$

$$S_{t_0 + \Delta t} = S_{t_0} + v_{t_0} \cdot \Delta t + \frac{a_{t_0}}{2} \cdot \Delta t. \quad (2.14)$$

These are then the two main equations that will be used to integrate the system after each time step. As can be seen, the force needs to be calculated only once but needs to be used later in the next iteration. Another interesting point to make is the time step. As is in general and again here, a smaller time step will yield a more precise result but will result in a slower simulation.

There are however 2 extra additions that will be used to either more precisely steer the images in the correct direction:

- Preventing overshoot.

The fact that velocity is used does add efficiency in case of larger distances to be traveled

over the PES by the images, but it may end up with an image having a too-large velocity such that it overshoots the optimal pathway. Therefore, before the integration equations are used it is first checked if the newly calculated force is in the opposite direction of the old velocity. If this is the case, then first the velocity and the previous force are set to the zero vector before the other steps occur.

- Not making the bends.
if the image has a large velocity, then it may overshoot necessary bends in its trajectory. The gradient force may not be enough on its own to pull the image in the right direction. Therefore, the velocity vector will be projected onto the normalized force vector each time before the new location is calculated but after the new forces have been calculated. This will ensure that the image will be steered in the correct direction.

Both these measures together result in:

$$v_{n,new} = \begin{cases} (F_{n+1}, v_n) \frac{F_{n+1}}{|F_{n+1}|^2} & , (F_{n+1}, v_n) \geq 0 \\ 0 & , else \end{cases} \quad (2.15)$$

2.2.6 Iteration

The entire model thus runs on an iterative basis. However, before that begins must everything be set up in a correct manner. Examples of these include the creation of a 'good enough' initial guess and defining every velocity as zero initially.

Then the iteration itself consists of 3 steps: calculating the forces, calculating the new velocities, and then calculating the new positions. One thing worth noting which will be explored more in Chapter 3 is the fact that for the calculations of the variables of iteration n of a specific image, only information of iteration $n - 1$ is necessary. This will allow for the parallelization of quite some calculations. A piece of pseudo-code of the algorithm will be shown in later in this chapter in Algorithm 1

Convergence

Then there is the matter of convergence. A general convergence criterion is extremely hard to determine due to the incredibly large amount of applications for this model aside from just quantum chemistry. Firstly, the parameters to consider are both the velocity and the forces on the images. If either of these is large enough, then the NEB method has not yet converged to a stable pathway. Therefore, the final convergence parameter $\theta \in \mathbb{R}$ of iteration n will be defined as

$$\theta_n = \max_{i \in \mathbb{N}} (|F_{n,i}|) + \max_{i \in \mathbb{N}} (|v_{n,i}|). \quad (2.16)$$

This variable, the sum of the maximum force magnitude over all images and the maximum velocity magnitude over all images will only be low in 2 circumstances: when it is stable, or when it is on a ridge. Imagine a ball rolling down a hill and then onto a for which it has just enough speed to slowly crest over the top to go down again. Therefore, it is always still preferable to create a plot of this parameter to check if it has truly converged. An example is given in Figure 2.5.

A thing of note is that the convergence parameter will usually start smaller than it is at its peak. The reason for this is that it starts with 0 velocity. Therefore, there are cases where the convergence parameter needs to rise before it can fall to a stable almost non-zero level. This effect can be slightly seen in the right figure of Figure 2.5.

Algorithm

The final algorithm used by the Nudged Elastic Band method can be seen in Algorithm 1. Here, NEB method parameters are the number of images, the time step size and the spring constants.

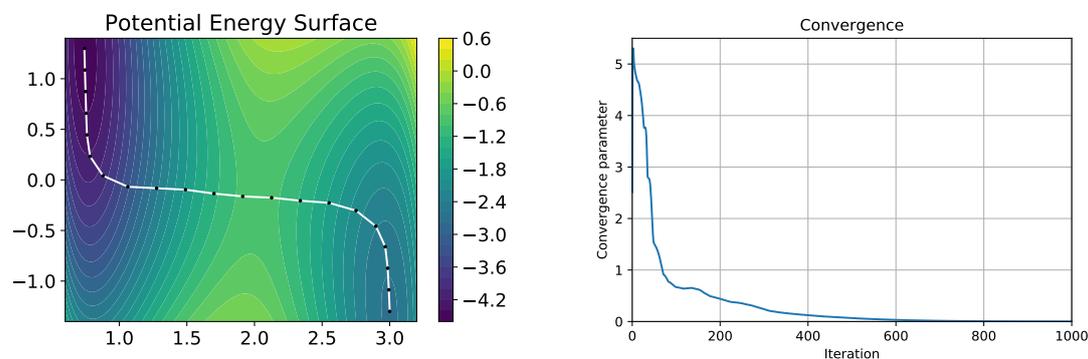


Figure 2.5: The left figure shows a contour plot of a Potential Energy Surface, with a solution as found by the NEB method. On the right can the convergence parameter be seen against the number of the iteration.

Algorithm 1 Nudged Elastic Band method

Require: Beginning and end state

Require: Potential Energy Surface

Require: NEB method paramters

Create initial (linear) guess for pathway

while θ not small enough **do**

for every image **do**

 Calculate gradient force

 Project gradient force on normalized parallel to find parallel component

 Subtract parallel component from original force to find perpendicular component

 Calculate spring force

 Project spring force on normalized parallel to find parallel component

 Subtract parallel component from original force to find perpendicular component

 Scale the perpendicular component using f

 Add perpendicular and parallel components back together

 Add the resulting gradient force and spring force together to find the final force

if Angle between old velocity and new force $> 90^\circ$ **then**

 Set velocity to 0

else

 Update velocity by projecting velocity on normalized force

end if

 Update the velocity using first step of Verlet Velocity (Equation 2.13)

 Update the location using second step of Verlet Velocity (Equation 2.14)

end for

 Calculate Convergence parameter

end while

Return (S_1, S_2, \dots, S_N)

2.2.7 Energy graph

Once an acceptably stable band has been found, thus the least-maximum energy pathway, the task of retrieving the important data begins. This is done via a simple plot, where the energy of each image is plotted against its distance towards the initial state along the pathway. The Euclidean distance is used, no matter the amount of dimensions. Then, the difference between the energy of the maximum and the initial state can be read to find the activation energy. An example can be seen in Figure 2.6.

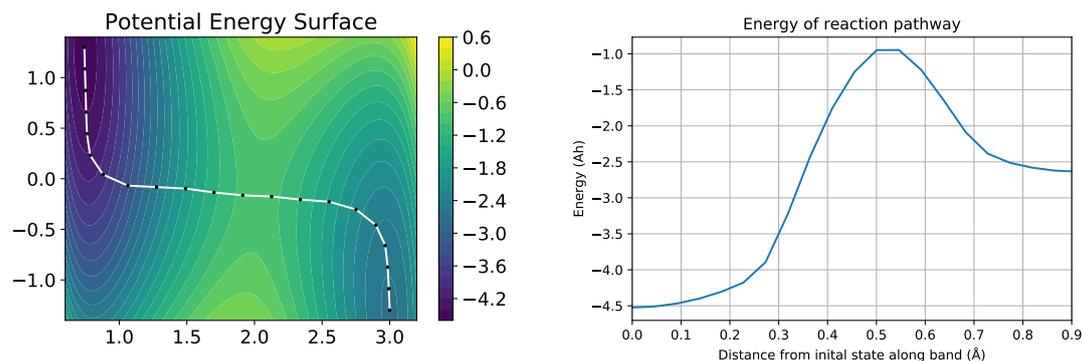


Figure 2.6: The left figure shows a contour plot of a Potential Energy Surface, with a solution as found by the NEB method. On the right can the energy of the reaction pathway be seen plotted against the distance.

2.3 Hartree-Fock

One key component not discussed yet is the function that determines the energy of a state. It is not important for the construction of the system due to the generality of the NEB method, but for our applications, a molecule is the object on which the NEB will act and this molecule will be described by a quantum mechanical system. For this, the Hartree-Fock method [Hajiabadi, 2023] will be used to determine both the energy and the gradient of the energy. The precise connection between the NEB method and the molecule will be discussed in Section 2.4.

Invented by Dr. Hartree and improved by Dr. Fock, the Hartree-Fock method finds the energy of a system by solving the Schrodinger equation for the many-electron wave function using various approximations. This section will go into some detail regarding the thought process and the methods, however, it will not go in-depth in the specific derivations since this method is not the main focus of this report.

2.3.1 System

First, consider the fact that electrons are fermions and thus have half-integer spins, which according to the Pauli exclusion principle, are not allowed to simultaneously occupy the exact same quantum states. The consequence of this is that the total system, consisting only of electrons and thus fermions, is anti-symmetric. Let $\psi(x_1, x_2)$ be some many-electron state, then

$$\psi(x_1, x_2) = -\psi(x_2, x_1). \quad (2.17)$$

Note that this occurs as well for states with more particles: any transposition between two electrons will cause the sign to change. One method to create an ansatz for an anti-symmetric many-electron wave function of the many single-electron wave functions is the Slater determinant. It can be

described by

$$\psi_{\text{Slater}}(x_1, x_2, \dots, x_K) = \frac{1}{\sqrt{K!}} \begin{vmatrix} \chi_1(x_1) & \chi_2(x_1) & \cdots & \chi_K(x_1) \\ \chi_1(x_2) & \chi_2(x_2) & \cdots & \chi_K(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(x_K) & \chi_2(x_K) & \cdots & \chi_K(x_K) \end{vmatrix} \quad (2.18)$$

which, when the determinant is worked out, is both normalized and anti-symmetric.

Next, since we want to find the energy of a stable system will we look mostly at the time-independent Schrodinger equation which is

$$\hat{H} |\psi\rangle = E |\psi\rangle \quad (2.19)$$

where $|\psi\rangle \in L^2$ is the current state of the system with L^2 the space of all square-integrable normalized functions, $\hat{H} : L^2 \rightarrow L^2$ the Hamiltonian which is an operator and $E \in \mathbb{R}$ the energy of the system and the eigenvalue of the Hamiltonian. Furthermore, note that The Hamiltonian can then be determined to be

$$\hat{H} = - \sum_{i=1}^K \frac{1}{2} \nabla_i^2 - \sum_{A=1}^M \frac{1}{2M_A} \nabla_A^2 - \sum_{i=1}^K \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^K \sum_{j>i}^K \frac{1}{r_{ij}} + \sum_{A=1}^M \sum_{B>A}^M \frac{Z_A Z_B}{r_{AB}} \quad (2.20)$$

where $M \in \mathbb{R}$ is the number of nuclei, $M_A \in \mathbb{R}$ the mass of nucleus $A \in \mathbb{N}$, $Z_A \in \mathbb{R}$ the charge of nucleus A and $r_{ij} \in \mathbb{R}$ the distance between particles i and j . The terms are, in order, the kinetic energy of the electrons, the kinetic energy of the nuclei, the electron-nucleus interaction, the electron-electron interaction and the nucleus-nucleus interaction. It can then be split into a part acting on the electrons and a part acting on the others to read

$$\hat{H} = H_{\text{elec}} + E_{\text{nuc}} = \left(- \sum_{i=1}^K \frac{1}{2} \nabla_i^2 - \sum_{i=1}^K \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^K \sum_{j>i}^K \frac{1}{r_{ij}} \right) + \left(- \sum_{A=1}^M \frac{1}{2M_A} \nabla_A^2 + \sum_{A=1}^M \sum_{B>A}^M \frac{Z_A Z_B}{r_{AB}} \right). \quad (2.21)$$

It should then be noted that the nuclei part is always known as the atoms in the molecule are known together with the fact that they are considered stationary due to their immense mass in comparison to the electrons. The interesting part is then the electron Hamiltonian, which can be split into individual electron Hamiltonians by defining

$$\hat{H}_{\text{elec}} = \sum_{i=1}^K \hat{H}_{\text{elec},i} \quad ; \quad \hat{H}_{\text{elec},i} = -\frac{1}{2} \nabla_i^2 - \sum_{A=1}^M \frac{Z_A}{r_{iA}} + U_i \quad (2.22)$$

with $U_i : L^2 \rightarrow L^2$ the yet-to-be-defined operator of the total of the electron-electron interactions acting on electron i . This gives the Fock equations which are

$$\hat{H}_{\text{elec},i} |\chi_i\rangle = \varepsilon_i |\chi_i\rangle. \quad (2.23)$$

Then, the mean field approximation is used to determine U_i . This implies that the average interaction of an electron with another electron is taken by considering the charge density of said electron instead of the original electron-electron interaction. This gives the operator

$$U_i = \iiint \frac{\rho(r')}{|r_i - r'|} d^3 r' \quad ; \quad \rho(r) = \sum_{i=1}^K |\chi_i(r)|^2 \quad (2.24)$$

Then the problem of the electronic energy can be stated as

$$\langle \psi_{\text{Slater}} | \hat{H}_{\text{elec}} | \psi_{\text{Slater}} \rangle = E_{\text{elec}} \quad (2.25)$$

which in quantum mechanics is an expectation value problem. An explicit formula for E_{elec} will be given in Subsection 2.3.3.

2.3.2 Basis

One problem that remains is that the single-electron eigenvalue problem is not easily solved for the general wave functions. One ansatz is to consider each wave function a linear combination of functions with known solutions, mainly Gaussians. A set of these is then called the basis set of the wave functions and is used to approximate the actual single-electron wave function as follows:

$$\chi_i(x_i) \approx \sum_a c_a \phi_a(x_i) \quad (2.26)$$

where $\phi \in L^2$ is a member of the basis set. The size of this basis set is finite. This set often exists only of orthonormal wave functions $\phi_i(x)$, which means that

$$\iiint \overline{\chi_i(x)} \chi_j(x) dV = \delta_{ij}. \quad (2.27)$$

If they are not, then they can always be mapped onto an orthonormal basis set using the Gram-Schmidt process. Several examples of such sets are given in Table 2.1.

Name	Explanation
STO-nG	This family of basis sets consist of n Gaussian primitives of the form $\phi(r) = \left(\frac{2\alpha}{\pi}\right)^{\frac{3}{4}} \exp(-\alpha r^2)$ [Hehre et al., 1969]
A-BCg	This family of basis sets, created by the group of John Pople [Ditchfield et al., 1971] allows for flexibility in the allocation of the number of Gaussians to orbits. The orbits near the nuclei get A Gaussians and the valence orbits get either B or C Gaussians, whatever is necessary. Larger families also exist with names such as A-BCDg or longer where the valence orbits are thus split in more sections with different amounts on Gaussian's.
A-BCg(H,K)	A variation on the previous family with polarization function included. H is normally replaced by an orbit letter such as d,f, or p which are added to the heavy atoms, and K is similar but for light atoms. In front of every orbit letter can also be a number for the number of polarization functions. An example would be 6-31G(3df,3pd).
cc-pVDZ	A family of basis sets taking more things into account [Dunning Jr, 1989]. 'cc-p' stands for correlation-consistent polarized and the V stands for valence-only basis set. It is possible to have a C instead of a V, which also includes core sets. The D stands for Double and can be replaced by either T(riple), Q(uadruple), 5, 6, ... for the number of different possibilities in the Valence orbits.
def2-SVP	A diverse family of sets [Hellweg and Rappoport, 2015]. 'def2-' is a part of the name and does not change. SV stands for triple valence and it can change to T(riple)ZV and Q(uadruple)ZV for either 3 or 4 possibilities for valence electrons instead of only two for SV. P stands for polarization functions, and may be in a bracket for only polarization functions on heavy atoms. Can be repeated for more polarization functions. D can be added for diffuse functions.

Table 2.1: Several basis sets and some explanations regarding them. These are by no means all of them as usually a new basis set is created for a specific problem to fit its specific focus.

Then, with the basis sets in mind, one can calculate the precise coefficients for each of the functions in the basis to approximate the actual wave function. Firstly, the goal should not be to perfectly achieve the actual wave function, since the space L^2 has infinite dimensions while the basis has not. Therefore there will almost always be an error, but this will be minimized as shown in the next Subsection.

2.3.3 Variational method

Consider the concept of a stable many-electron state: no electron is excited and the system cannot go to lower energy values. Therefore, by definition, any other function which is not the ground state will always have a higher energy. So the basis functions, which are valid functions by design, and every linear combination of these functions, the single-electron wave functions, will always have a many-electron wave function, as given by Equation 2.18, which has an equal or higher energy than the ground state. By varying the various weights for each separate basis function in the single-electron wave function, can the total energy be changed. This is then a problem that can be solved via the Lagrange Multiplier method, to find the lowest possible energy state for the many-electron wave function. The normalization of the single-electron wave functions can then be considered the constraint. Lastly, do note that the Hamiltonian still depends on the charge density of the many-electron wave function. This results in the explicit energy equation, using the equations given in Subsection 2.3.1,

$$E_{\text{elec}} = \sum_{i=1}^K \varepsilon_i - \frac{1}{2} \left(\int \frac{\rho(r)\rho(r')}{|r-r'|} d^3r d^3r' - \int \frac{\xi(r,r')\xi(r',r)}{|r-r'|} d^3r d^3r' \right) \quad (2.28)$$

with ε_i the solution of the eigenvalue problem

$$\left(-\frac{\Delta_i}{2} - \sum_{i=1}^K \sum_{A=1}^M \frac{Z_A}{|r_i - r_A|} \right) \chi_i(r) + \int \frac{\rho(r')}{|r-r'|} \chi_i(r) d^3r' - \int \frac{\xi(r',r)}{|r-r'|} \chi_i(r') d^3r' = \varepsilon_i \chi_i(r) \quad (2.29)$$

and

$$\xi(r',r) = \sum_{j=1}^K \chi_j^*(r') \chi_j(r). \quad (2.30)$$

This last equation can be interpreted as follows: the many-electron wave function energy is equal to the sum of the single-electron wave functions minus two integrals of which the first can be considered equivalent to a Coulomb interaction while the second has no classical equivalent. It is called the Exchange interaction.

Then, after various more steps, which fall out of the scope of this report both due to subject and complexity, can it be shown that this problem is equivalent to the matrix problem

$$\underline{H} \underline{c}^i = \varepsilon_i \underline{S} \underline{c}^i \quad (2.31)$$

which can be solved 'comparatively' easily by computers compared to the original problem. However, one problem still persists which is the fact that the electron-electron potential depends on the solution for which we are trying to solve. This can be solved via iteration via the algorithm as shown in Algorithm 2.

The resulting energy is one of the things needed for the NEB method, however the gradient is more important. One possibility to use much more complex calculations to get an analytical form for the gradient of the energy. The other option is via numerical methods such as the forward difference equation [Sauer, 2012] or via other complex methods or algorithms [Szabo and Ostlund, 2012].

The first method obviously has advantages because it prevents additional computational work, however, another difficult method may need to be implemented. The disadvantages of the second

Algorithm 2 Hartree-Fock method

Require: Nuclei coordinates

Require: Basis sets

Guess initial single-electron wave functions

while $\Delta E \not\approx 0$ **do**

 Calculate total charge density from single-electron wave functions

while Energy keeps getting lower **do**

 Vary basis functions of single-electron wave functions

 Calculate energy of single electron-wave functions

 Calculate energy of many-electron wave functions

end while

 Calculate ΔE compared to previous iteration

end while

Return Energy

method are more severe, however. In the computationally best case, the forward difference method, it is found that

$$\nabla E(r) = \frac{1}{|d_1|} \begin{pmatrix} E(r + d_1) - E(r) \\ E(r + d_2) - E(r) \\ \vdots \\ E(r + d_D) - E(r) \end{pmatrix} \quad (2.32)$$

where d_i is the zero vector except for dimension i where it has a very small value independent of i . This function converges to the analytical gradient if $|d_i| \rightarrow 0$ for a smooth function. Note that this function calculates the energy $D + 1$ times, whereas the central difference method as given by

$$\nabla E(r) = \frac{1}{|2d_1|} \begin{pmatrix} E(r + d_1) - E(r - d_1) \\ E(r + d_2) - E(r - d_2) \\ \vdots \\ E(r + d_D) - E(r - d_D) \end{pmatrix} \quad (2.33)$$

calculates the energy $2D$ times. Also note that smaller $|d_i|$ may lead to smaller inaccuracies but that it will introduce numerical instability due to floating point errors.

2.4 Application to quantum chemistry

In this short section will the connection between the Nudged Elastic Band method and quantum chemistry be made. The system to be used is a molecule which is in the ground state. All the electrons are thus in positions that result in the lowest energy for that specific configuration. In this system will we consider chemical reactions between two different states of the molecule. The results given in the later half of Chapter 3 will be mostly internal structure changes but general chemical reactions should be feasible.

The system space can then be considered to be \mathbb{R}^3 , as that is the space in which the molecule exists. However, the state space itself is \mathbb{R}^D where $D = 3 * M$. For a molecule that will be used later: C_3H_6 , this leads to a space of \mathbb{R}^{27} .

The energy function as given by the Hartree-Fock method is then the Potential Energy Surface and the gradient must be calculated using it.

It is then possible to determine the precise pathway a chemical reaction will take through the entire state space.

3. Final model

As discussed earlier, the Nudged Elastic Band method from Chapter 2 is able to determine the reaction pathway between a given beginning and end point. One other important function is the energy function, which in this case will be fulfilled using the Hartree-Fock method. Thus, all the building blocks are ready to implement this in a piece of code.

This code is written in the programming language known as Python due to its ease of use and its many libraries which can be imported. Two specific libraries that will be used heavily are the PySCF library and the multiprocessing library, both of which would take too significant effort to recreate for their final result on the program. The PySCF library [Sun, 2015; Sun et al., 2018; Wang and Song, 2016] will be used to perform the RHF method by outsourcing it to multiple C sub-functions which are faster due to C being a compiled language instead of an interpreted language. The multiprocessing library is used to enable the outsourcing of certain function calls to other processor cores. This heavily speeds up the final program since the forces on each image can be computed in parallel and this is by far the slowest part of the code. The intrinsic mechanics of this library are far above my current understanding due to the fact that Python inherently does not support multiprocessing.

The code itself will be explained more in Section 3.1. Here an overview will be given of the general process and loop present. It will also be explained how code can be converted to apply to many other situations. Next, several possible pitfalls will be discussed in Section 3.2 and how to prevent them or circumvent their consequences. Lastly, will some results be discussed in Section 3.3. The discussed reactions are two triatomic chemical reactions and two single-atomic structure changes.

3.1 Code

First note that the code itself can be found attached to this report or can be received at request from the author. A list of which functions are supposed to be used by users and which are supposed to be just used by the code can be found in Table 3.1. Also, see all the functions in *Italics*. These are the functions that are specifically created to fit the current simulation to the PySCF library. If this simulation were to be rewritten for another system/energy function, then all these functions could be safely removed because their functionality was already lost. Several other functions may need to be slightly changed since PySCF has several requirements which caused changes that cascaded through the code.

These changes are mainly in the energy and gradient functions. The reason for this is that the RHF method needs to know both the basis and what kinds of atoms are present in the molecule and this information is given via passable arguments instead of hard-coded values to ensure a larger flexibility in this specific application of the model. If these values were hard-coded, then the system may have been more flexible in general with respect to transitions to different situations and energy functions but then the different situations in the current system would take more effort. It was a trade-off were both choices had some merit. Then, let us discuss the 4 major functions that add functionality aside from conversions between data formats.

3.1.1 *sim_complete*

The main function itself works in 3 stages: initialization, iteration, and termination. In the initialization phase are all the inputs checked for errors and necessary variables created. Then, the iteration phase starts where the iteration from the NEB method is applied. This begins with collecting, for each image separately, the data it will need to calculate the current acting forces and its next velocity and location. All this information is then put into tuples. The reason for this is that all the calculations done with these are done on separate cores for multiprocessing and tuples are immutable so there are no accidental race conditions since each image gets its own little 'information package' with which it needs to work. Then, when this is complete can all the data be updated when every image is done with its calculations.

User functions	Back-end functions
sim_complete	_energy
plotter	_inserter
reaction_path	_init_guess
<i>StringToXYZ</i>	_normal_parallel
<i>Optimize</i>	_project_perpendicular
<i>XYZToString</i>	_switch
<i>StringToPosAtoms</i>	_spring_force
<i>PosAtomsToXYZ</i>	_spring_final
<i>FinalPosAtomsFromLog</i>	_energy_gradient
<i>FinalVeloAtomsFromLog</i>	_force_final
	_single_image
	_create_canvas
	<i>_SinglePosToString</i>
	<i>_SingleAtomToString</i>

Table 3.1: *The functions in the code and their categorization. More information regarding every function can be found in the documentation of each function in the code itself. Some information regarding specifically the functions for PySCF can be found in Subsection 3.1.6*

This specific application is done using the concept of Pools which contain a certain number of worker processes. All the information packages are put into a single list on which the calculation function is then mapped in a special manner. It designates each element and the function to map it to a specific worker process in the Pool. When this calculation is done will the worker receive another information package and the function until every calculation is done. Setting up the Pool and distributing the tasks does take time, which results in this method only being faster whenever the calculations take long enough, which is the case for the RHF method.

After these calculations, all the forces, velocities, and locations of the images in the central location variable are then updated such that the information packages for the next iteration can be prepared again. However, before this happens can the convergence check take place, if it is enabled. This is done entirely as discussed in Subsubsection 2.2.6. If the system has determined that either the simulation has converged or that the maximum amount of iterations is reached, then the iteration phase stops. It closes the Pool of workers and returns the final positions of the images and the atom symbols.

This then makes the entire simulation a function which can be defined as

$$f : D \times N \times \mathbb{R} \cup \mathbb{R}^{(N-1)} \times \mathbb{R} \times \mathbb{R}^D \times \mathbb{R}^D \times \mathbb{A}^N \times \mathbb{B} \left(\times \mathbb{N} \times \{0, 1\} \times \mathbb{R} \times \mathbb{R} \times \mathbb{N}_{\leq 7} \times (\mathbb{R}^D)^N \right) \longrightarrow (\mathbb{R}^D)^N \quad (3.1)$$

where

$$D \in \mathbb{N} \quad \& \quad N \in \mathbb{N} \quad \& \quad \mathbb{A} \equiv \text{The set of all atoms} \quad \& \quad \mathbb{B} \equiv \text{The set of all bases.} \quad (3.2)$$

The first variable is the number of dimensions in the system, the second is the number of images, and the third is the spring force. This can be either a number that is applied to every spring or a vector that can contain different values for each spring. Then is the time step size, the beginning position, the ending position, the atoms vector, and lastly the basis. Every variable after this, in the brackets, is optional and has default values if not specified. First of these is the number of iterations, whether or not to try to determine whether the system converged, the upper tolerance of the convergence and then the lower tolerance, the level of verbose-ness (the higher the more information will be given in the log file which will be discussed later) and lastly a possible initial position for each image to start the simulation from in the form of a vector containing vectors. The output, the final location of each image, is also a vector of vectors.

Actual results of these functions will be shown later in Section 3.3

3.1.2 *plotter*

This function is meant as a simple graphical tool to view the results. It creates a 2D contour plot where the contours are based on the energy function supplied. Furthermore, the images are all displayed as points connected with lines to their neighbor. The advantage of this function is that it gives a clear overview of the situation where it can be easily seen what the current situation is. However, it has 2 very clear disadvantages: Sufficient resolution in the contour plot requires a lot of energy calculations. This can take an incredible amount of time, especially considering the fact that the energy calculations via the RHF method already take a long time. Next, the plot can only show 2 dimensions, so if more dimensions are present in the system itself then information is lost. This is not a problem for a simple system which can indeed be described by that amount of atoms, but it is extremely unwieldy for larger systems such as molecules with 3 or more atoms. An example of a plot this function creates can be seen in Figure 3.1. The function can be considered to be defined by

$$f : (\mathbb{R}^D)^N \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{A}^N \times R \times \mathbb{B} \longrightarrow \mathbb{R}^{(R \times R)} \times (\mathbb{R}^D)^N \quad (3.3)$$

where $R \in \mathbb{R}$ the number of samples for the contour plot in each direction. The first term is the vector containing the location vectors of all images. The next two terms give the limits for

the variable on the horizontal axis and the next two for the vertical axis. Then the atoms are given, followed by the amount R and the basis for the energy calculation. The result is then best described as a matrix as it is the energy value for each point in the contour plot together with the locations of the images as this is all the information contained in the plot.

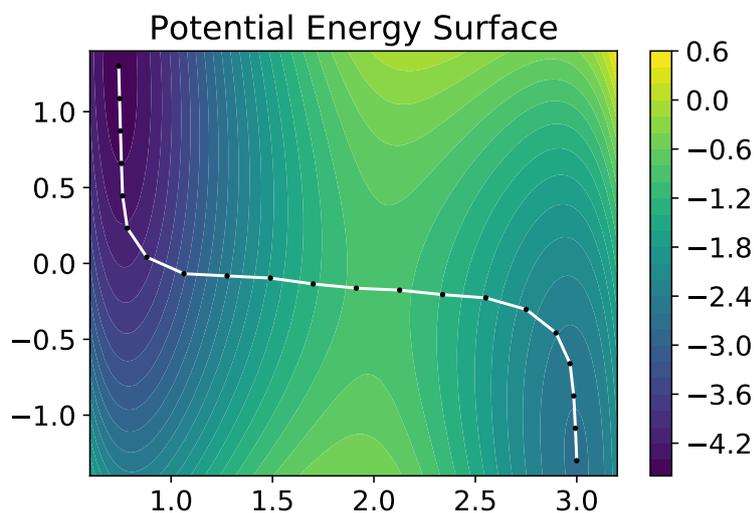


Figure 3.1: The reaction pathway for some system found using the NEB method. The main canvas and the color bar are the output of 'plotter'. The labels and title were added later.

3.1.3 *reaction_path*

This function gives perhaps the most important information on the outcome of a system. It receives a vector of location vectors of the images, the atoms, and a basis and it then plots the energy along the reaction path. It is able to do this because it plots this energy against the Euclidean distance between neighbors. It therefore gives a great picture of the energy levels in the reaction and it can thus also nicely give meaning to activation energy because that is the height difference between the beginning point and the highest point of the plot. It can be defined as

$$f : (\mathbb{R}^D)^N \times \mathbb{A}^N \times \mathbb{B} \longrightarrow \mathbb{R}^N \quad (3.4)$$

where the first term is the vector of the location vectors of the images, the second term is the atoms, and the third term is the basis. This results in then in one vector with the energy values displayed in the plot. The result will then look like Figure 3.2

3.1.4 *Optimize*

This function in itself is not very useful, however, it helps with setting the stage such that the simulation can do its job. The reason for this is that this function uses the gradient to find a least-energy state near the given initial state. The use case for this can be found in one method in determining the initial locations for a simulation. For this, www.molview.org was used to create the molecule structures. However, this site does not always give correct structures so a tool like the 'Optimize' function can take these mediocre structures and make them adequate to use in the simulation. It can be defined as

$$f : (\mathbb{R}^D)^N \times \mathbb{A}^N \times \mathbb{B} \longrightarrow (\mathbb{R}^D)^N \quad (3.5)$$

where the first term is again the vector the location vectors of the images, the second term the atoms, and the third term the basis. The result is again a vector of location vectors.

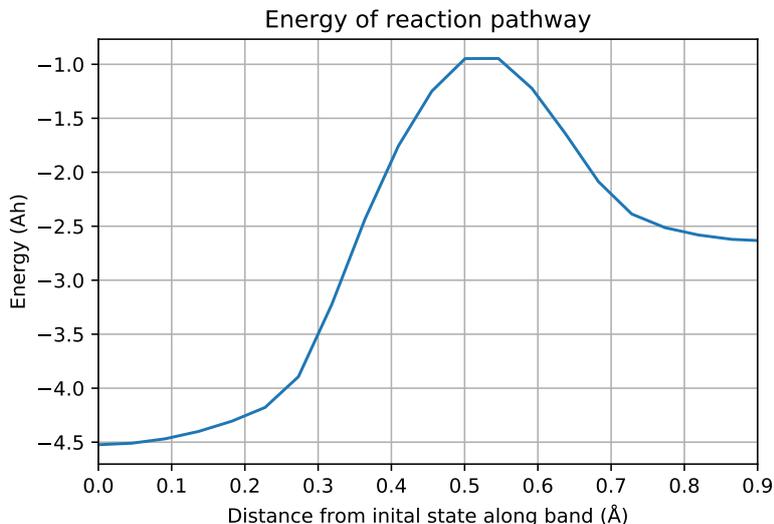


Figure 3.2: The figure shows an example of the output of the function *reaction_path*. The labels and the title have been added afterward.

3.1.5 Jmol

While it is an outside piece of software, its use in this report cannot be understated. This software plots the molecules, more specifically the '.xyz' files, as a proper molecule with bonds in a 3D manner. This allows any image, no matter the 24 dimensions for an 8-atom molecule, to be perfectly visualized without loss of information. It even is able to create animations of the images along the reaction pathway as will be later shown in this Chapter. It can be found on <https://jmol.sourceforge.net/> [development team, 2023].

3.1.6 Custom functions for PySCF compatibility

The main purpose of the functions specifically used for PySCF are as conversions between the following:

$$\text{XYZ} \longleftrightarrow \text{String} \longrightarrow (\text{Pos,Atoms}) \longrightarrow \text{XYZ} \quad (3.6)$$

Here, 'XYZ' implies the file format called '.xyz', 'String' is the format in which PySCF accepts the locations and types of atoms, and '(Pos,Atoms)' the manner in which the simulation itself accepts position and atom information. Other functions linked to the PySCF library in this code are 'Optimize', which optimizes a given molecule structure until a least-energy point is found, and 'Final...AtomsFromLog' which extracts information from the final position stated in a log file from a finished simulation. This can, however, easily be changed such that it would work with different versions of the simulation.

3.1.7 Log file

The main function 'sim_complete' is always creates a log file when run. It will be created in the current working directory and will be named 'sim_complete-YYYY-MM-DD-HH:MM:SS.log' where the time will be of the time the function started. This log file is done for the reason of storing the output of the calculations outside of the current memory space. The amount of details is defined by the value of the 'verbose' variable and it can be changed. The options can be seen in Table 3.2

Verbose level	Stored details
0	Every variable given to the function, when the loop will stop, how long the function took to run, and the final positions of each image.
1	Everything above and the final velocities of each image.
2	Everything above and the initial guess of each image's location.
3	Everything above and the new location of each image after every step of the NEB method.
4	Everything above and the new velocity of each image after every step of the NEB method.
5	Everything above and the forces that acted on each image during each step of the NEB method.
6	Everything above and the information package each image uses to compute its new parameters during the NEB method.
7	Everything above and the force magnitude and velocity magnitude of each image after every step of the NEB method.

Table 3.2: *The functions in the code and their categorization. More information regarding every function can be found in the documentation of each function in the code itself. Some information regarding specifically the functions for PySCF can be found in Subsection 3.1.6*

Two functions that have interactions with the log file are 'FinalPosAtomsFromLog' and 'FinalVeloAtomsFromLog'. These functions both look in the log file to extract the vector with the vectors of their corresponding quantity. Do note that the 'FinalVeloAtomsFromLog' function will give incorrect values if the level of verbose-ness is not at least 1. These functions will not work if the simulation is no longer used for the current system but they should be fixable with relatively few changes. Both these functions can be defined as

$$f : \mathbb{S} \longrightarrow (\mathbb{R}^D)^N \times \mathbb{A}^N \quad (3.7)$$

where \mathbb{S} is the set containing every possible string.

3.2 Pitfalls

Even though the code itself contains quite a large number of type tests and a smaller number of unit tests, it is still important to take note of the input variables of the functions and not blindly trust them if there is no error. Several pitfalls can occur during the running of the functions which can create wildly inaccurate results or things which can excessively slow down the process.

3.2.1 Order in coordinates

The order of the coordinates and atoms is extremely important. Suppose we have a correctly defined beginning and ending position as

$$X_{begin} = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N), \quad (3.8)$$

$$X_{end} = (\overline{x_1}, \overline{y_1}, \overline{z_1}, \overline{x_2}, \overline{y_2}, \overline{z_2}, \dots, \overline{x_N}, \overline{y_N}, \overline{z_N}). \quad (3.9)$$

where $(x_i, y_i, z_i) \in \mathbb{R}^3$ is the beginning location of the i -th atom. This is very different compared to

$$X_{begin} = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_N, y_N, z_N), \quad (3.10)$$

$$X_{end} = (\overline{x_2}, \overline{y_2}, \overline{z_2}, \overline{x_1}, \overline{y_1}, \overline{z_1}, \dots, \overline{x_N}, \overline{y_N}, \overline{z_N}). \quad (3.11)$$

In the second case, the first and second atoms will want to switch places somewhere during the reaction path. It is therefore important to know beforehand which atom will end up at what place after the reaction. This is usually something that can be imaged intuitively. It can also often be seen in Jmol when an incorrect reaction pathway is animated. However, in case of uncertainty then all options can be tried and the option with the lowest activation energy is the correct option. In the worst case does this take

$$\prod_{a \in \overline{A}} (N(a))! \quad (3.12)$$

possibilities where \overline{A} is the set of all atoms present in the molecule and $N(a)$ how much atom a is present in the molecule. However, each of these can again be ran parallel to speed up the process. There is always a possibility that such a mistake will 'work itself out' by going to what should have been the correct beginning/end position before moving to the incorrect position. This does however both take a long time and increase the length of the reaction pathway and lower the resolution on the pathway itself.

3.2.2 Ridge walking

As mentioned earlier in Subsection 2.2.4, it is possible to have the entire band of images on top of a ridge in some dimensions, an example of this can also be seen in the earlier mentioned Subsection. It is therefore important that the final result is checked in programs such as Jmol such that these

things can be spotted. This is sometimes possible to do this intuitively. Think of it as a pencil that stands perfectly straight. It does not want to fall to any side because gravity is a force only going downward. Options to prevent this behavior are to first examine the Potential Energy Surface to see if any such ridges are present or to add noise or a perturbation to the initial guess. If these last options are chosen and multiple new different solutions are found then it should be noted that the one with the lowest activation energy is the correct solution.

3.2.3 Scaling

One last pitfall is relatively easy to prevent but may go unnoticed at first. It is the connection between the number of images, the spring force, and the gradient. Suppose a better resolution of the reaction pathway is needed, so the number of images would be doubled. However, now note that the average distance between each image is now a bit less than half of what it was. Therefore, the spring force needs to be doubled as well. Suppose on the other hand that the energy function is multiplied by three to fit better to a different unit, which causes the gradient to be multiplied by three everywhere. Then the spring force needs to be multiplied by three as well to maintain the same balance as before. Thus

$$\text{Number of images} \propto \text{Spring force} \quad \& \quad \text{Spring force} \propto \text{Gradient}. \quad (3.13)$$

The key point of this is that none of these three values can be changed without some change to the balance of the system. However, these changes can be easily circumvented by adjusting other parameters.

3.2.4 Time step

One important part of the simulation is the size of the time step. If it is too small, then numerical instability may appear due to the finite precision of computers and the simulation will run exceedingly slow. If the time step is too large then inaccuracies will appear as many features of the PES will not be properly taken into account. It has been found that this last phenomenon mostly materializes in severe errors due to far too large values, while the region where the time step causes numerical instability is so exceedingly small that it hardly appears. Therefore the focus should be on choosing a time step just not large enough to cause any errors in that end. The range which is usually acceptable is $[0.1, 0.01]$.

3.3 Results

Using this code, several simulations were run and values were determined. In this section will these be discussed. It should be noted however that in general the Hartree-Fock methods are not perfect and exhibit flaws [Denis and Ventura, 2000]. Firstly, the use of basis automatically implies that there are answers you are not looking at, namely wave functions which are not linear combinations of the basis functions. Secondly, the assumptions made to allow the method to work also hinder the program in its accuracy as they are limitations on reality. Third, bigger bases will almost always yield better results at higher computation costs but due to time constraints where the best bases are not chosen. Lastly, just like the higher-level Nudged Elastic Band method used in the simulation, it is an iterative converging method. One can always compute more iterations to get a 'better' result. This all results in the numerical results not being representative of reality in most cases sadly.

However, we will still show 3 examples to showcase the possibilities of the entire model. The first will be a reaction where one atom travels from one molecule to another, with an added degree of freedom which can be interpreted as an atom coupled to B in a harmonic way. Secondly, will the structure change of Ethen-1,2-diol between its cis- and trans-variant be discussed, and lastly will the transition between Cyclopropane and Propylene be shown.

3.3.1 Tri-atomic reaction

This system did not depend on the PySCF library and was an initial test of the system. It mimics a reaction where a diatomic molecule loses one atom which goes to another atom to form a diatomic molecule again, while under the influence of a fourth atom via the manner of a harmonic manner. [JÓNSSON et al., 1998]. In equation form would the reaction itself be



An interesting technique displayed in this system is the reduction of dimensions via either symmetries or by fixing certain things. A, B, and C are all put onto one line where both A and C are fixed to reduce dimensions. Furthermore, only relative coordinates are used which further decreases the amount of dimensions needed. In the end, this system is described via 2 dimensions using the following potential:

$$V(r_{AB}, x) = V^{non-harmonic}(r_{AB}, r_{AC} - r_{AB}) + 2k_c(r_{AB} - (r_{AC}/2 - x/cc))^2 \quad (3.15)$$

where

$$\begin{aligned} V^{non-harmonic}(r_{AB}, R_{BC}) = & \frac{Q_{AB}}{1+a} + \frac{Q_{BC}}{1+b} + \frac{Q_{AC}}{1+c} \\ & - \left(\frac{J_{ZB}^2}{(1+a)^2} + \frac{J_{BC}^2}{(1+b)^2} + \frac{J_{AC}^2}{(1+c)^2} \right. \\ & \left. - \frac{J_{AB}J_{BC}}{(1+a)(1+b)} + \frac{J_{BC}J_{AC}}{(1+b)(1+c)} + \frac{J_{AB}J_{AC}}{(1+a)(1+c)} \right)^{\frac{1}{2}} \end{aligned} \quad (3.16)$$

which is a potential that could also be used for the case of only 3 atoms which are all allowed to move on the same line. In this equation, it is defined that

$$Q_{XY} = \frac{d_{XY}}{2} \left(\frac{3}{2} e^{-2\alpha(r_{XY}-r_0)} - e^{-\alpha(r_{XY}-r_0)} \right), \quad (3.17)$$

$$J_{XY} = \frac{d_{XY}}{4} \left(e^{-2\alpha(r_{XY}-r_0)} - 6e^{-\alpha(r_{XY}-r_0)} \right) \quad (3.18)$$

with $a = 0.05$, $b = 0.8$, $c = 0.05$, $d_{AB} = 4.746$, $d_{BC} = 4.747$, $d_{AC} = 3.445$, $r_0 = 0.742$ and $\alpha = 1.942$, $r_{AC} = 3.742$, $k_c = 0.2025$, $cc = 1.154$. The simulation results and some intermediate frames are shown in Figure 3.3. An animated version of the simulation can be found in Appendix A.

An interesting behavior that can be spotted from the final frame is that the least-energy reaction pathway is not perfectly smooth, it has small kinks. The reason for this is simple: the spring force is only firstly small because both neighbors pull nearly parallel to each other resulting in no resulting perpendicular force, and the angle is so small that Equation 2.4 makes it even smaller. These kinks may disappear when waiting long enough.

This example shows that the system also works with general potentials given in formula form and not just with the PySCF library. It furthermore nicely showcases options to decrease the number of dimensions to be dealt with. Lastly, in the system itself is it interesting to see that x is like a gate value. As soon as it is near zero, then B starts to move swiftly to the other atom.

3.3.2 C3H6

C3H6 has 2 molecules of similar molecular structure: Propylene and Cyclopropane. The difference in their structures can be seen in Figure 3.4 where in the caption it is explained which frame is what. The main difference is that the bottom carbon-carbon bond of Cyclopropane is broken open and one hydrogen atom from the middle carbon atom moves to the right. This relatively

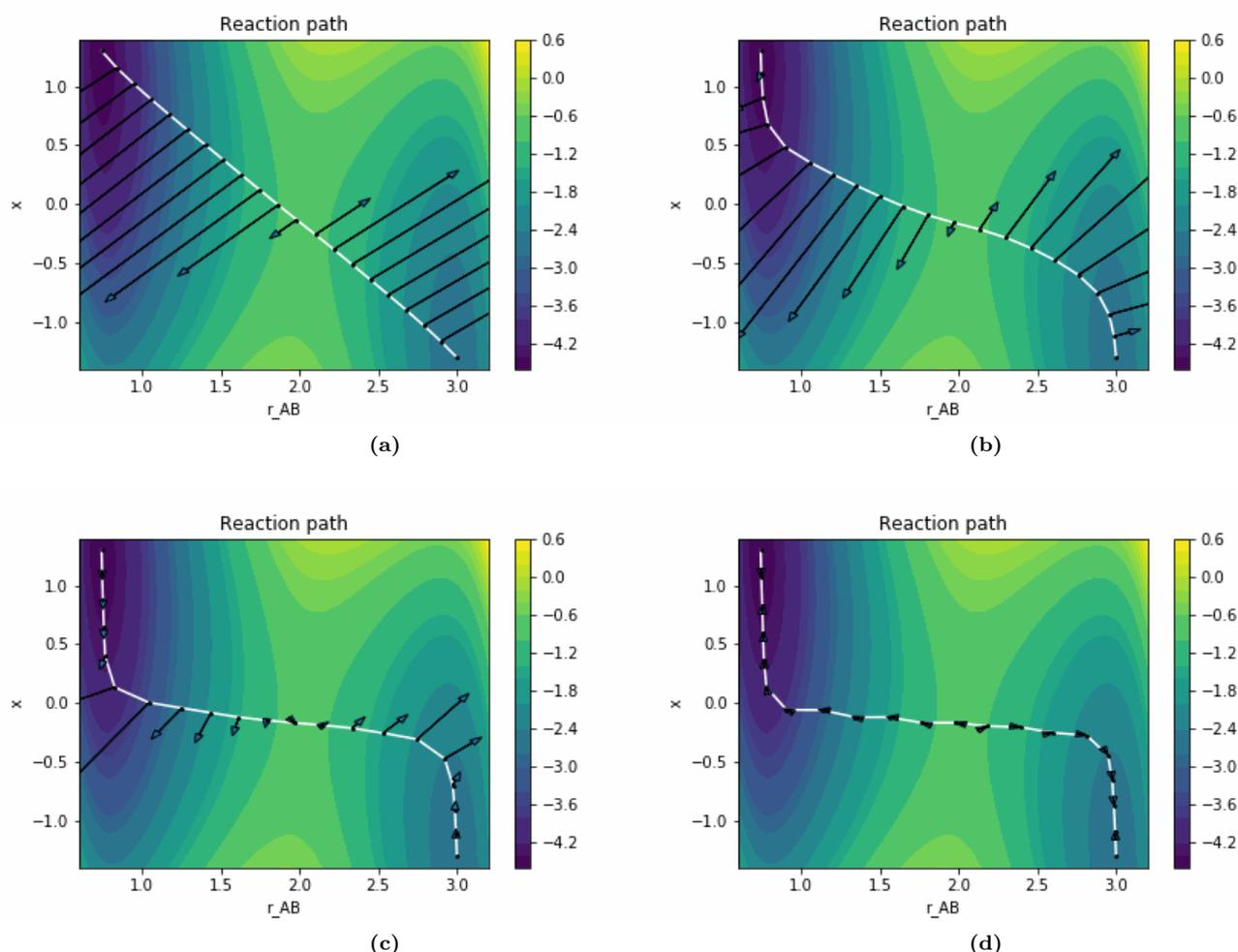


Figure 3.3: The reaction pathway after 0, 0.15, 0.40 and 1.99 seconds with a time step of 0.01 and a spring constant of 5. The arrows are the currently acting forces on the images. The extremely low forces after at 1.99 seconds imply that it is as good as converged and this is thus the least-energy reaction path.

simple model was used because it could be a good validation test for this application of the Nudged Elastic Band model. An animation of the transition, calculated using the simulation, can be found in Figure A.2.

The energy levels along the reaction pathway can be seen in Figure 3.5. One thing to notice immediately is that the value is highly negative. The reason for this is that energy is set to zero when every atom is separated from each other with their electron clouds at rest, all in the ground state. The state where several of these atoms have bonded then has less energy.

When the activation energy is extracted from the least-energy reaction pathway then the activation energy to go from Cyclopropene to Propylene is 271.96 kilojoule per mole [Jug, 1976; Lewis et al., 2006]. This in contrast to the calculated value of 448.90 kilojoule per mole which was found. This is an error of 65%. These calculations were done with the basis 'def2-TZVP' as this basis should give adequate results while still being relatively fast.

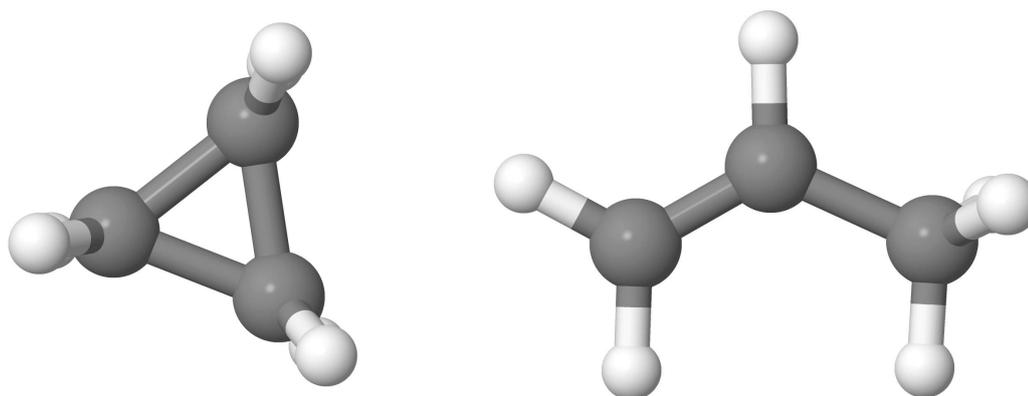


Figure 3.4: The left molecule is Cyclopropane in its base state. The right molecule is Propylene in its base state. Note that the rightmost carbon atom has three hydrogen atoms bonded, which thus implies that the bond between the leftmost carbon atoms is a double bond. An animated version of the final reaction pathway can be found in the Appendix A.

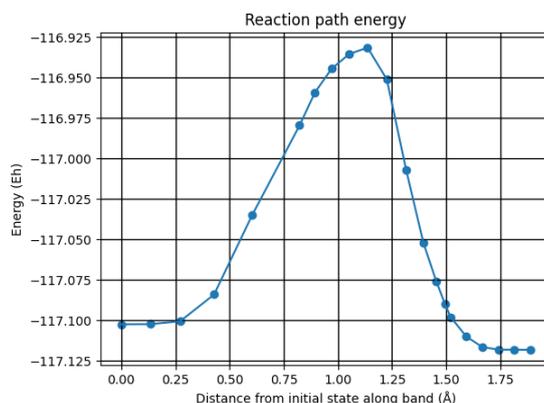


Figure 3.5: The energy over the reaction pathway of the structure transition of C_3H_6 . The images are more concentrated in the middle since the spring force was triple there to increase the resolution in the area of interest. The initial state is Cyclopropane and the final state is Propylene.

3.3.3 $C_2O_2H_4$

$C_2O_2H_4$, named Ethen-1,2-diol, has an interesting structural property in that it has 2 different structures for exactly the same set of connections. Each atom is connected to the same other atoms but the inflexibility of a double bond, which does not rotate, causes these two structures as seen in Figure 3.6 to be distinct isomers. These two variants are called the 'cis' and the 'trans' variant which comes from Latin with the meanings 'the side of' and 'the other side of'. Cis-ethen-1,2-diol thus has its oxygen atoms on the same side while the Trans-ethen-1,2-diol has them on different sides. This example was chosen because it has several different atoms while still being doable. Furthermore, no bonds were changed so the energy difference is purely from other interactions. The energy along the reaction pathway can be seen in Figure 3.7. An animation of the transition, calculated using the simulation, can be found in Figure A.3.

When the activation energy is extracted from the least-energy reaction pathway then the activation energy to go from the cis variant to the trans variant has been calculated to be 429.542 kilojoule per mole. No sources for the activation energy have been found so no comparison can be made.

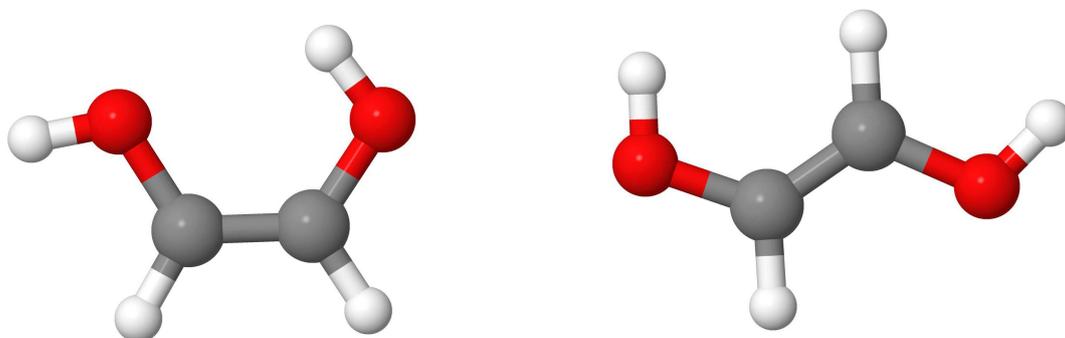


Figure 3.6: The left molecule is the cis variant while the right molecule is the trans variant. Note that the rightmost carbon atom has three hydrogen atoms bonded, which thus implies that the bond between the leftmost carbon atoms is a double bond. An animated version of the final reaction pathway can be found in the Appendix A.

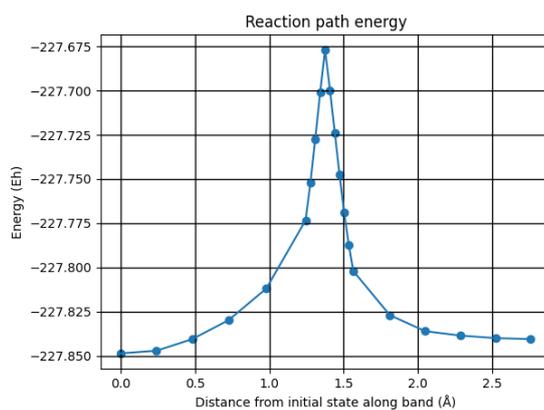


Figure 3.7: The energy over the reaction pathway of the structure transition of C₂O₂H₄. The images are more concentrated in the middle since the spring force was triple there to increase the resolution in the area of interest. The initial state is Cyclopropane and the final state is Propylene.

4. Conclusion & Discussion

This project aimed to create a piece of software that is capable of using the Nudged Elastic Band method to determine the activation energy of reactions by finding the least-energy reaction path. To do this was the first goal to create this as a model which would work in general and the second goal was to successfully integrate it with a Python library to be able to use it for quantum chemistry reactions.

The first goal was achieved following the steps of Hannes Jónsson, Greg Mills and karsten W. Jacobsen [JÓNSSON et al., 1998]. This was mirrored in the first half of Chapter 2. Here the general idea behind the method is explained and the general setup of the final loop once it is applied. Then, the second half of the chapter is dedicated to an explanation of the Hartree-Fock method. This fairly old method approximates the many-electron wave function of a set of atoms. Following several assumptions or approximations. The subject is discussed at a level where the main idea and steps are clear, but several key steps to convert several equations to the equations that the computer can solve have been left out due to complexity and being out of scope for this project.

Then the application of both methods in the code is discussed as well as several shortcomings. Several pitfalls to take into account are discussed, as these can cause seriously deviating results when not circumvented. Then, while the software is capable of finding the correct reaction paths for simple reduced 2-dimensional reactions such as the reaction given in Subsection 3.3.1, it may not be valid for more complex models, as has been seen when the calculated value of the activation energy between Cyclopropane and Propylene was 60% larger than the experimental value. The reasons for this can be many, the they most often boil down to 3 cases:

- The potential energy surface is incorrect
The Restricted Hartree-Fock method is old. It was invented in the initial years of the study of quantum mechanics in general and there are already 2 generations of newer solver methods: Post-Hartree-Fock and Density Functional Theory. Post-Hartree-Fock is a class of methods that try to improve Hartree-Fock by including effects ignored by Hartree-Fock or by doing things in a more physically motivated method. Density Functional Theory does away mostly with the electron wave function and focuses mainly on the electron density. In any case, if the Restricted Hartree-Fock method is not accurate enough, then the results would also not be accurate enough.
- The code itself is incorrect
The code for the Nudged Elastic Band method was written by hand. It is possible that some mistake is still present somewhere and that this impacts the solution value. In fact, this happened during the development of this model when two slight inaccuracies together caused the system to oscillate during certain situations. Currently, there is no other strange behavior that might suggest such things, however.
- The code was not run long/complex enough
This NEB method can always be run with more iterations and a smaller time step while the RHF can always use a more complex basis. However, both these things were not done to such an extent as many simulations already took over 12 hours with the basis 'def2-TZVP' and around 500 iterations on the NEB method with a time step of 0.01. The time step was deemed good enough as it was large enough to allow for converging results while not too large to cause crashing due to divergence.

Therefore, the code should not yet be used for any research or similar. The code does not produce good enough results compared to the experimental values in more complex cases. It does give valid results in two-dimensional cases with an analytical gradient instead of PySCF.

Bibliography

- Ayala, P. Y., & Schlegel, H. B. (1997). A combined method for determining reaction paths, minima, and transition state geometries. *The Journal of Chemical Physics*, *107*(2), 375–384. <https://doi.org/10.1063/1.474398>
- Bartsch, T., Revuelta, F., Benito, R. M., & Borondo, F. (2012). Reaction rate calculation with time-dependent invariant manifolds. *The Journal of Chemical Physics*, *136*(22), 224510. <https://doi.org/10.1063/1.4726125>
- Chemistry is everywhere - american chemical society. (2024). <https://www.acs.org/education/whatischemistry/everywhere.html>
- Denis, P. A., & Ventura, O. N. (2000). Density functional investigation of atmospheric sulfur chemistry. i. enthalpy of formation of hso and related molecules. *International Journal of Quantum Chemistry*, *80*(3), 439–453. [https://doi.org/https://doi.org/10.1002/1097-461X\(2000\)80:3<439::AID-QUA14>3.0.CO;2-O](https://doi.org/https://doi.org/10.1002/1097-461X(2000)80:3<439::AID-QUA14>3.0.CO;2-O)
- development team, J. (2023). Jmol: An open-source java viewer for chemical structures in 3d, <http://www.jmol.org>, 16.1.63, 2023-11-16.
- Ditchfield, R., Hehre, W. J., & Pople, J. A. (1971). Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic Molecules. *The Journal of Chemical Physics*, *54*(2), 724–728. <https://doi.org/10.1063/1.1674902>
- Dunning Jr, T. H. (1989). Gaussian basis sets for use in correlated molecular calculations. i. the atoms boron through neon and hydrogen. *The Journal of chemical physics*, *90*(2), 1007–1023.
- Elber, R., & Karplus, M. (1987). A method for determining reaction paths in large molecules: Application to myoglobin. *Chemical Physics Letters*, *139*(5), 375–380. [https://doi.org/https://doi.org/10.1016/0009-2614\(87\)80576-6](https://doi.org/https://doi.org/10.1016/0009-2614(87)80576-6)
- Hajjabadi, H. (2023, July). Hartree fock method: A simple explanation - insilicosci. <https://insilicosci.com/hartree-fock-method-a-simple-explanation/>
- Halgren, T. A., & Lipscomb, W. N. (1977). The synchronous-transit method for determining reaction pathways and locating molecular transition states. *Chemical Physics Letters*, *49*(2), 225–232. [https://doi.org/https://doi.org/10.1016/0009-2614\(77\)80574-5](https://doi.org/https://doi.org/10.1016/0009-2614(77)80574-5)
- Hehre, W. J., Stewart, R. F., & Pople, J. A. (1969). Self-consistent molecular-orbital methods. i. use of gaussian expansions of slater-type atomic orbitals.
- Hellweg, A., & Rappoport, D. (2015). Development of new auxiliary basis functions of the karlsruhe segmented contracted basis sets including diffuse basis functions (def2-svpd, def2-tzvpd, and def2-qvppd) for ri-mp2 and ri-cc calculations. *Physical Chemistry Chemical Physics*, *17*(2), 1010–1017.
- JÓNSSON, H., MILLS, G., & JACOBSEN, K. W. (1998). Classical and quantum dynamics in condensed phase simulations. https://doi.org/10.1142/9789812839664_0016
- Jug, K. (1976). Mechanism of cyclopropane-propene isomerization. *Theoretica chimica acta*, *42*, 303–310.
- Laidler, K. J. (1984). The development of the arrhenius equation. *Journal of chemical Education*, *61*(6), 494.
- Lewis, D. K., Hughes, S. V., Miller, J. D., Schlier, J., Wilkinson, K. A., Wilkinson, S. R., & Kalra, B. L. (2006). Kinetics of the thermal isomerization of 1, 1, 2-trimethylcyclopropane. *International journal of chemical kinetics*, *38*(8), 475–482.
- Malmstadt, H. V., Cordos, E. A., & Delaney, C. J. (1972). Automated reaction-rate methods of analysis. *Analytical Chemistry*, *44*(12), 26A–41a.
- Pollak, E., & Talkner, P. (2005). Reaction rate theory: What it was, where is it today, and where is it going? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *15*(2), 026116. <https://doi.org/10.1063/1.1858782>
- Sauer, T. (2012). *Numerical analysis* (Second). Pearson.
- Simm, G. N., Vaucher, A. C., & Reiher, M. (2019). Exploration of reaction pathways and chemical transformation networks [PMID: 30421924]. *The Journal of Physical Chemistry A*, *123*(2), 385–399. <https://doi.org/10.1021/acs.jpca.8b10007>

- Sun, Q. (2015). Libcint: An efficient general integral library for gaussian basis functions. *Journal of Computational Chemistry*, 36(22), 1664–1671. <https://doi.org/https://doi.org/10.1002/jcc.23981>
- Sun, Q., Berkelbach, T. C., Blunt, N. S., Booth, G. H., Guo, S., Li, Z., Liu, J., McClain, J. D., Sayfutyarova, E. R., Sharma, S., Wouters, S., & Chan, G. K.-L. (2018). Pyscf: The python-based simulations of chemistry framework. *WIREs Computational Molecular Science*, 8(1), e1340. <https://doi.org/https://doi.org/10.1002/wcms.1340>
- Swope, W. C., Andersen, H. C., Berens, P. H., & Wilson, K. R. (1982). A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of chemical physics*, 76(1), 637–649.
- Szabo, A., & Ostlund, N. S. (2012). *Modern quantum chemistry: Introduction to advanced electronic structure theory*. Courier Corporation.
- Vinu, R., & Broadbelt, L. J. (2012). Unraveling reaction pathways and specifying reaction kinetics for complex systems. *Annual Review of Chemical and Biomolecular Engineering*, 3(Volume 3, 2012), 29–54. <https://doi.org/https://doi.org/10.1146/annurev-chembioeng-062011-081108>
- Wang, L.-P., & Song, C. (2016). Geometry optimization made simple with translation and rotation coordinates. *The Journal of Chemical Physics*, 144(21), 214108. <https://doi.org/10.1063/1.4952956>

A. Animations

This Appendix contains all animations for several figures shown previously in the report. This animations will not work on some pdf viewers but will work on some others. Generally, dedicated pdf viewers are capable of this but this is no guarantee. Adobe Acrobat Reader is tested to be capable of viewing the animations, Google Chrome has been tested and is not capable of viewing the animations.

Figure A.1: The figure shows the simulation of the NEB method of Equation 3.15 as discussed in Subsection 3.3.1.

Figure A.2: The figure shows the final reaction path of the NEB method of Figure 3.5 as discussed in Subsection 3.3.2.

Figure A.3: The figure shows the final reaction path of the NEB method of Figure 3.6 as discussed in Subsection 3.3.3.