Dhilan Bekah - 20512356

NE 232 Quantum Mechanics Project 2

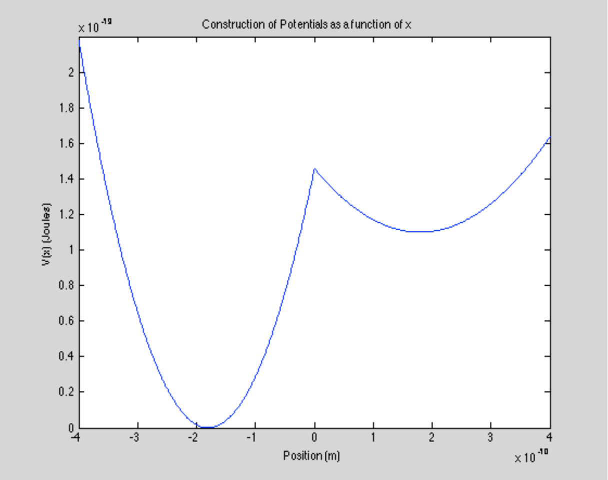
[redacted]

Analyzing DNA mutation, by method of proton tunneling between DNA nucleotide base pairs.

The potential that is used can be described as a single nucleotide base pair, containing two sites, one corresponding to an electron lone pair on either an oxygen or a nitrogen atom in the nucleotide base's structure, and the other corresponding to a proton covalently bonded to another oxygen or nitrogen atom. This proton-'electron lone pair' pair is essential to the hydrogen bonding in DNA base pairs and, consequentially, for replication. The two sites are modeled as an asymmetric double potential well, in order to address the fact that it must be difficult to reach the other site to reduce the chances of mutation. The analysis will focus on a few time-independent measurements to compare the effects of varying degrees of radiation on the likelihood of mutation, as well as the probability of the proton tunneling as time evolves.

3. Statement of the thesis:

Does increasing the radiation energy to the proton increase the tunneling probability, as well as decrease effective tunneling time, and thereby increase the chance of permanent mutations in DNA occurring? My hypothesis is yes, and in order to answer this question, we can model the problem using the following potentials:



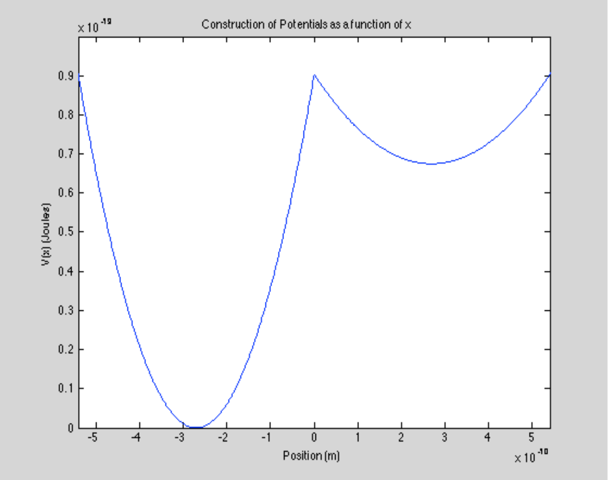


Figure 1. MATLAB Potential Plot: The average hydrogen bond distance in a G-C base pair is 3.6 Angstroms [1] as represented by the two well centers in the left figure, while the average hydrogen bond distance in an A-T base pair is 5.4 Angstroms [1] as represented by the two well centers in the right figure. The equations of both wells were calculated in order to have the energy barrier at x = 0 be 1.4595e-19 J for G-C, and 9.0352e-20 J for A-T, which are equivalent to the interaction energies 21 and 13kcal/mol, respectively [2].

4. Significance of the answer:

The exploration of quantum mechanics in biological phenomena has been rapidly increasing over the years, with quantum tunneling being one of the concepts of particularly high interest. As previously studied, proton and electron tunneling occur a great deal between molecules, which accounts for a vast majority of biochemical reactions that clearly involve quantum mechanics through changes in bonding [3]. Within the events of proton or electron tunneling, we have, arguably, one of the most important biological events, which is DNA replication and mutation.

DNA is a long polymer of repeating nucleotide units, and forms a double helix structure, with each helical chain having a pitch of 34 Angstroms and a radius of 10 Angstroms [7]. As well, one nucleotide unit is about 3.4 Angstroms [1]. More specifically, however, we have the measured hydrogen bond lengths between nucleotide base pairs, as shown by Figure 2.

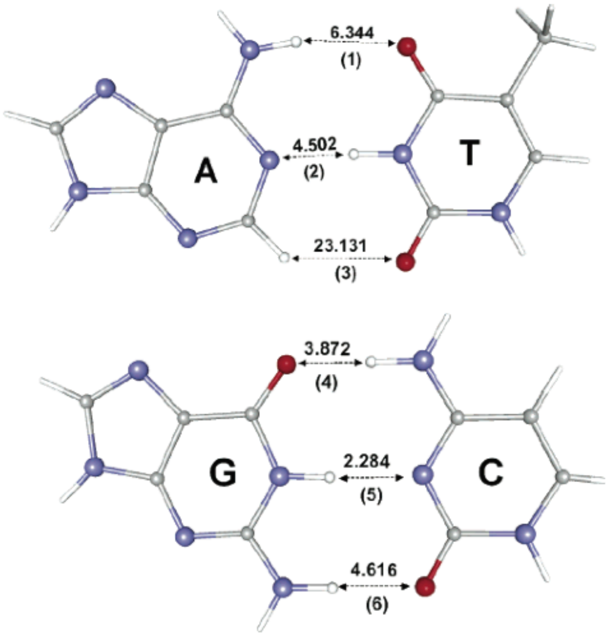


Figure 2. DNA Base Pair Structure

This image shows the hydrogen bond lengths between base pairs. The re are oxygen, while the blu are nitrogen, and the gre are carbon. Taken from http://pubs.acs.org/doi/pdf/10.1021/ja046282a [1]

One possible mechanism for DNA mutation that has been fairly well explored is proton tunneling occurring between these covalent bond-hydrogen bond base pairs, involving the proton from the original covalent bond tunneling across the potential barrier of the bond, to the electron pair on the atom involved in the original hydrogen bond [(4)]. DNA mutation can and has been attributed to radiation, which would increase the energy of the proton, resulting in a higher chance of tunneling. As a result, the structure of the original nucleotide bases can be altered, which in turn would affect ensuing replication (since replication is done through complementary strands). Figure 3 helps demonstrate this concept of altered base structure.

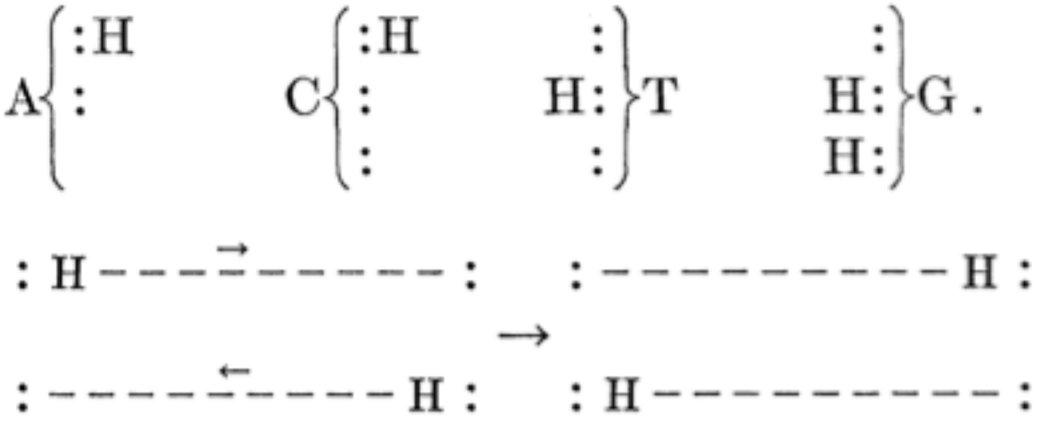

A
:H
:
C
:H
:
:
:
H:
:
T
:
H:
H:
G.
: H - - - - - - - - - - :
: - - - - - - - - - - H :
: - - - - - - - - - - H :
: H - - - - - - - - - - :

Figure 3. Electron pair/ proton locations in hydrogen bonds for each nucleotide base. Taken from http://users.df.uba.ar/giribet/f4/adn.pdf [(5)]

In a simpler form, the tunneling can be modeled with a highly asymmetric double potential well [(6)] with two sites, corresponding to either the hydrogen site within the covalent bond, or the electron lone pair on the atom involved in the hydrogen bond. Since the A-T and G-C base pairs have different interaction energies and different hydrogen bond distances, two sets of asymmetric potentials can be created to account for the different values.

This model works well for comparing the A-T and G-C base pairs (and their associated mutation likelihood), but also for understanding how energy from radiation can induce these mutations. Better understanding the process behind DNA mutations can help us detect them earlier on before replication has had a significant effect on the body. Similar models can provide us with information on how likely a mutation is to occur, and factoring in more information about the specific nucleotide base pairs could give us important information as to the sequence that is being affected, and potentially where to look in the genome to stop the mutation from spreading.

5. Description and implementation of your method:

The approach used for solving this problem begins with constructing a double, highly asymmetric potential well, with each of the center of the wells corresponding to two 'Sites', where Site A in the lower energy well is the proton (hydrogen atom in covalent bond), and Site B in the higher energy well is the electron lone pair on the atom involved in the hydrogen bond. The initial states are that the proton is localized to Site A for both the base pairs, but at different values (3.6/2 = 1.8 Angstroms for G-C, and 5.4/2 = 2.7 Angstroms for A-T). After there is an input of external energy in the form of radiation from a certain frequency, the proton will jump to various eigenstates depending on the amount of energy. A variety of parameters such as tunneling probability and effective tunneling time that are related to the eigenenergies (and corresponding eigenstates) will be calculated to analyze the effect of radiation. Since eigenstates are time-independent, these measurements are also time independent. The proton's wavefunction, however, will be used to see how tunneling occurs as a function of time, and the probabilities will be calculated, which are time-dependent.

Analysis

My Hamiltonian:

```
mp = 1.67e-27; % units of Kg
h1 = my_hamiltonian(mp,L(1),V) % For Part A: G-C Base Pair
h2 = my_hamiltonian(mp,L(2),V) % For Part B: A-T Base Pair
```

My Eigenenergies:

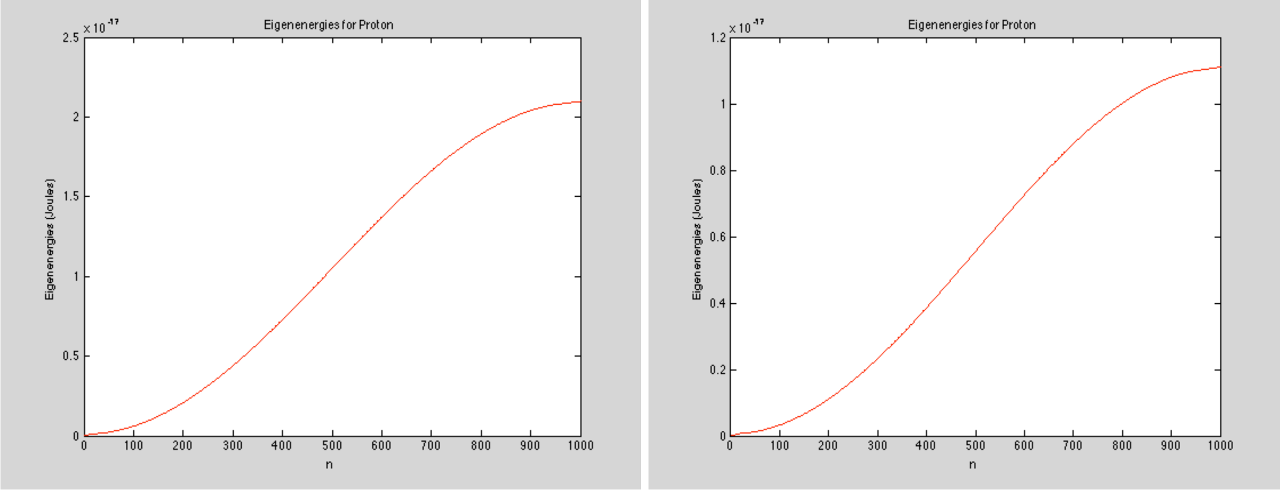


Figure 4. A) Part A Eigenenergies ; B) Part B Eigenenergies

These eigenenergy values do make sense, as they increase with increasing n, and, due to the fact that a larger 'a' value is used in Part B for the A-T base pair, we see that the energies are smaller, which is what we would expect. We also see the energies round off as n increases, which is attributed to approximation error, which also increases as n increases. The units for the eigenenergies are in Joules.

Solving the eigenvalue problem to obtain my eigenvectors:

```
%**NOTE: V is dependent on the lengths, and respective equations used
[eigenenergies, eigenstates] =
my_eigensystem(my_hamiltonian(mp,L(1),V),L(1)); % FOR PART A
[eigenenergies, eigenstates] =
my_eigensystem(my_hamiltonian(mp,L(2),V),L(2)); % FOR PART B
```

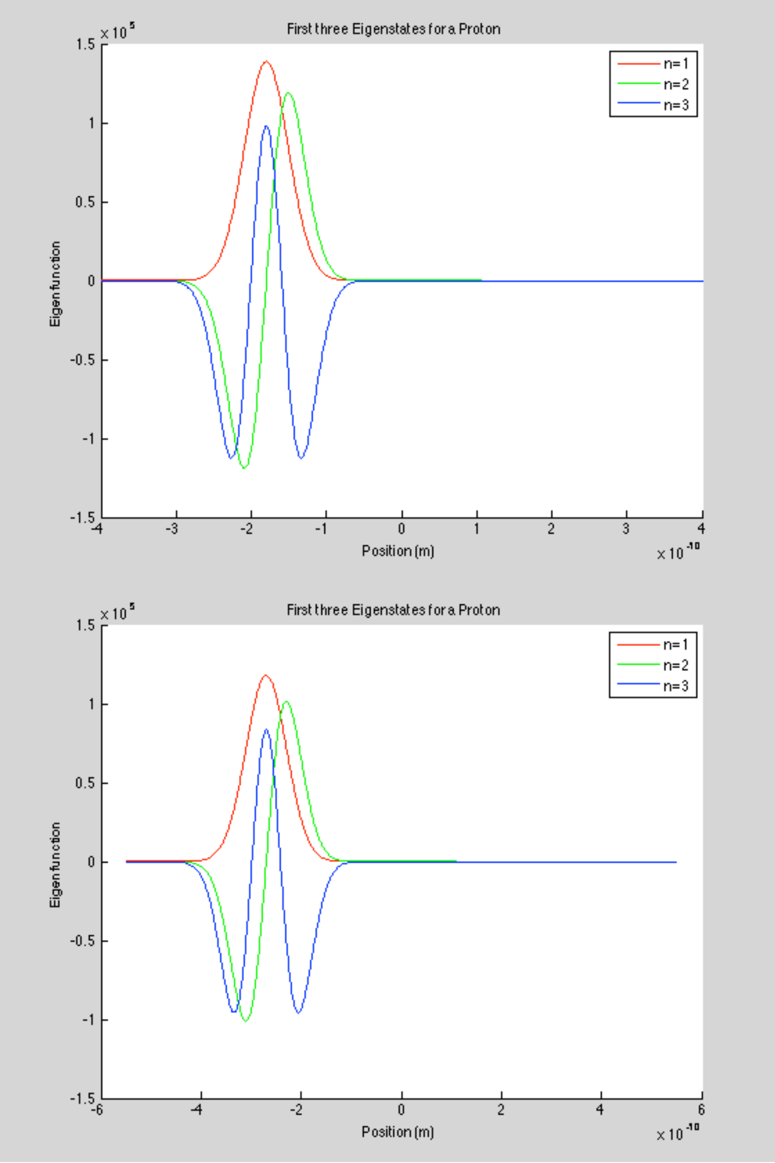


Figure 5.

A) (Top)
Part A Eigenstates
B) (Bottom)
Part B Eigenstates

We can see that these eigenvectors do make sense, as their shapes correspond with each other, and they follow the expected trends for low energy eigenvectors that we've encountered before. We also see that the eigenstates are confined to the Site A well in both, as that is where the lowest potentials are, whereas the energy barriers on both sides govern the shape of the eigenstates to approach 0, as displayed. They are not exactly the same however (and they should not be), as the lowest potential is at Site A, but Site A varies depending on whether it is Part A or Part B (G-C vs. A-T).

Dhilan Bekah - 20512356

Normalizing my eigenstates:

(Done in my Hamiltonian function)

```
[V,D] = eig(hamiltonian); % Eig function gives V (matrix of the
eigenvectors) and D (diagonal matrix containing the eigenenergies)
V = V'; % Transpose since V gives eigenvectors as columns
[row,col]= size(V);
delta = a/col; % Need a passed as a parameter to compute delta in order
to normalize the eigenstates
for i = 1:row total = 0;
      for j = 1:col     % Normalization process
            total = total + conj(V(i,j))*V(i,j)*delta;
      end
      V(i,:) = V(i,:)./sqrt(total);
end
eigenstates = V;
```

Orthogonality of my eigenstates:

Gathered from the code in Appendix A, we get that

Part A:

```
temp = 1.0000
```

when n = m for the general orthogonality expression (Integral from –inf to inf of φm* φn dx), and

```
temp = 7.0939e-16
```

when n != m for the general orthogonality expression (Integral from –inf to inf of φm* φn dx)

Part B:

```
temp = 1.0000
```

when n = m for the general orthogonality expression (Integral from –inf to inf of φm* φn dx), and

```
temp = -3.8706e-17
```

when n != m for the general orthogonality expression (Integral from –inf to inf of φm* φn dx)

Therefore these values show that the eigenstates in both parts are orthogonal, as we should expect a value of 1 when n=m (which we get in both cases), and a value of 0 when n!=m (which we still do get in both cases as the values printed by Matlab are extremely small and are essentially 0), according to the Kronecker-delta function.

Dhilan Bekah - 20512356

My wavefunction:

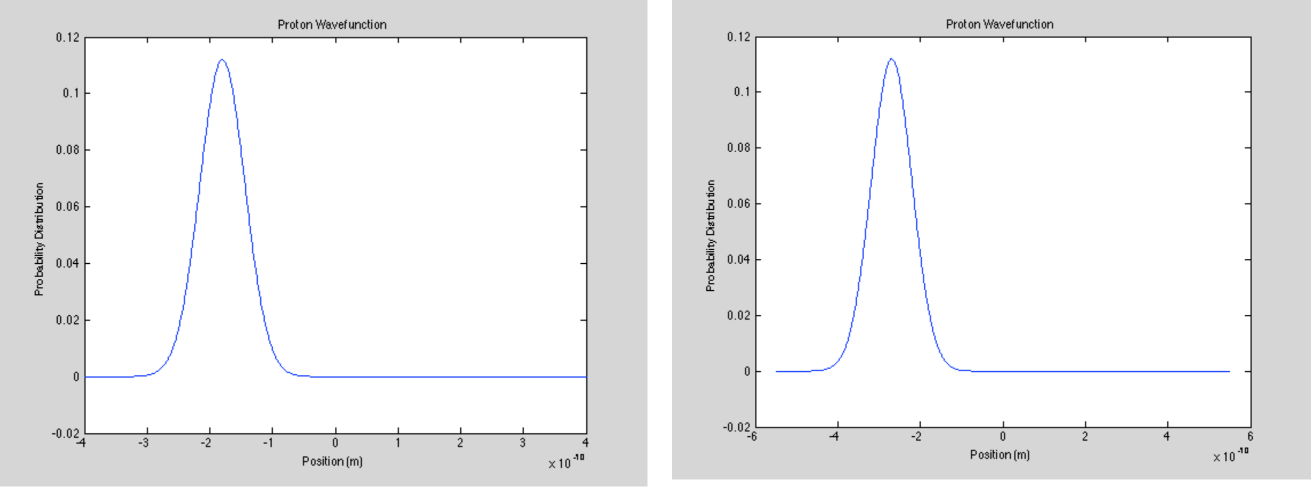


Figure 6. A) Part A Wavefunction ; B) Part B Wavefunction

Both wavefunctions are correct, as they both resemble the Gaussian (as they should), and they are centered at their respective Site A values. The wavefunctions show how the proton is initially localized to Site A.

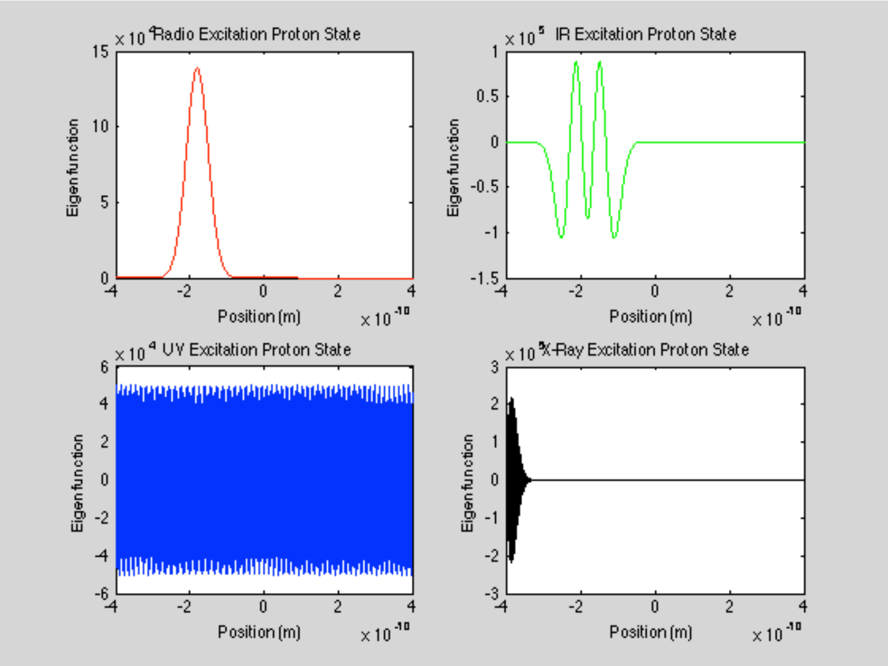
My Measurements:

The time-independent measurements are made for 4 different wave frequencies, in order to observe the effect that radiation energy would have on proton tunneling. We have the following initial values:

```
f_radio = 1e8; % Hz
f_IR = 5e13; % Hz
f_UV = 1e16; % Hz
f_xray = 1e18; % Hz
h = 6.626e-34; % units of m^2*Kg/s (J*s)
E_radio = h*f_radio; % units of J
E_IR = h*f_IR; % units of J
E_UV = h*f_UV; % units of J
E_xray = h*f_xray; % units of J
```

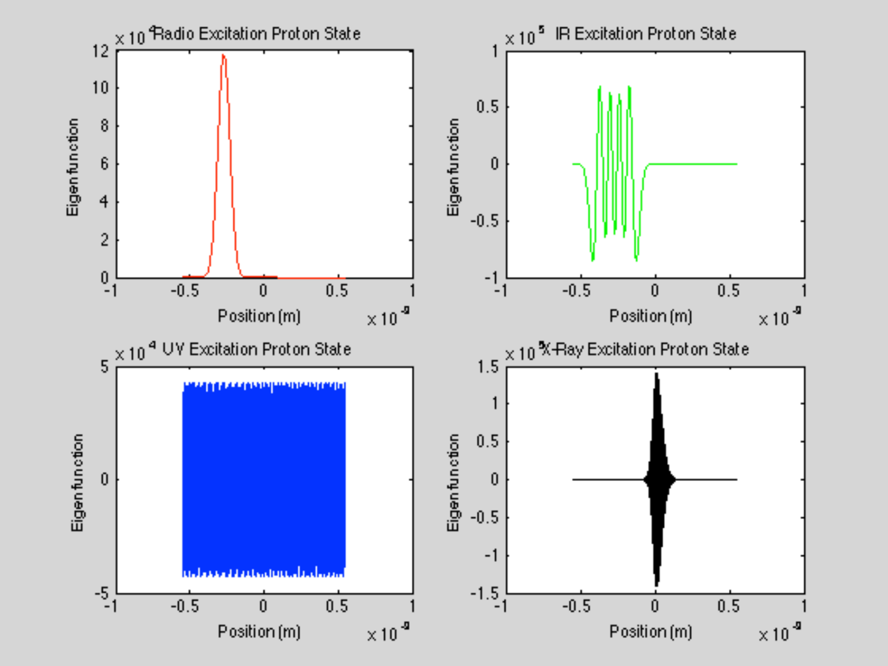
From these energies, the eigenstates achieved were found and plotted. Gathered from the code in Appendix A, we have the following:

Part A:



```
error_radio =
0.2538%

error_IR =
52.7596%

error_UV =
159.2363%

error_xray =
170.1181%

omega =
7.3706e+13
```

Part B:



```
error_radio =
0.0040%

error_IR =
0.0308%

error_UV =
189.7147%

error_xray =
172.9827%

omega =
3.8562e+13
```

Figure 7. Excitation states for 4 different radiation energies.

The % errors were calculated using the analytical energies, which were calculated using the respective 'omega's for parts A and B (which were calculated from the difference between the two lowest level eigenenergies, and confirmed by checking consecutive differences). The eigenstates for radio and IR excitation make sense for both Parts A and B, as they are both low energy eigenstates, and so the approximation error has not had a significant impact. With UV and X-ray excitation, however, n is much larger, and so the eigenenergies approximation becomes very bad, resulting in skewed displayed eigenstates, as well as very large % errors. Nonetheless, the time-independent quantum tunneling calculations can still be done, as the energy values still correspond to the excitation levels (they just do not match up with the corresponding analytical eigenenergies and eigenstates at higher n). In fact, it would be incorrect to use the analytical solutions at those n found, as they would not represent the energy that the proton is excited to.

The first measurement is the probability of a 'quantum jump' occurring. Bohr assumed that an atom could only exist in certain 'stationary states', and that transitions between them occurred as discontinuous 'quantum jumps', associated with radiation [(4)]. The equation says that the probability of a quantum jump occurring, q, is roughly proportional to exp(-($E_{threshold}$ – E_{state})/k*T), where $E_{threshold}$ – E_{state}, or ΔE, is the activation energy from the state under consideration to the energy threshold leading over to the other state [(4)].

For Part A:

```
qj_radio = 1.1574e-04

qj_IR = 0.2066

qj_UV = Inf

qj_xray = Inf
```

For Part B:

```
qj_radio = 7.4164e-05

qj_IR = 0.1909

qj_UV = Inf

qj_xray = Inf
```

An important thing to note is that, if the state energy is already greater than the threshold energy, the values of the 'probability' q will exceed 1 as the expression is meant to consider states under the threshold energy. Nonetheless, a value of q > 1 simply indicates that the energy required to overcome the threshold is already present, and so there is no 'jump' to consider the possibility of.

The next measurement made was the tunneling probability, g, which is dependent on the quantity s. The quantity s has the following expression:

$$s = \frac{2\pi}{h} \int_{x1}^{x2} (2m[V(x) - E])^{1/2} dx$$

E and m are the energy and mass of the proton, V(x) is the potential energy at x, and x1 and x2 are the limits of the barrier region for when V(x) >= E [(6)].

For Part A: `s_radio = 68.6963; g_radio = 2.1435e-60`

```
            s_IR = 47.2905; g_IR = 8.3951e-42
```

```
s_UV = 0; g_UV = 1

s_xray = 0; g_xray = 1
```

For Part B:

```
s_radio = 81.8575; g_radio = 7.9337e-72

s_IR = 44.0531; g_IR = 5.4450e-39

s_UV = 0; g_UV = 1

s_xray = 0; g_xray = 1
```

The forward and reverse tunneling rates per second were measured next, following the method in [6] to calculate the oscillation frequencies (see code in Appendix A).

For Part A:

```
c1_radio = 0; c2_radio = -3.5585e-46

c1_IR = 3.9301e-28; c2_IR = -1.0007e-27

c1_UV = 9.9576e+15; c2_UV = 9.7916e+15

c1_xray = 3.1678e+16; c2_xray = 3.1512e+16
```

For Part B:

```
c1_radio = 0; c2_radio = -8.0822e-58

c1_IR = 2.6736e-25; c2_IR = -2.8732e-25

c1_UV = 9.9850e+15; c2_UV = 9.8831e+15

c1_xray = 1.6749e+16; c2_xray = 1.6648e+16
```

And finally, effective tunneling time was measured, $\tau = 1/(c1 + c2)$ [6].

For Part A:

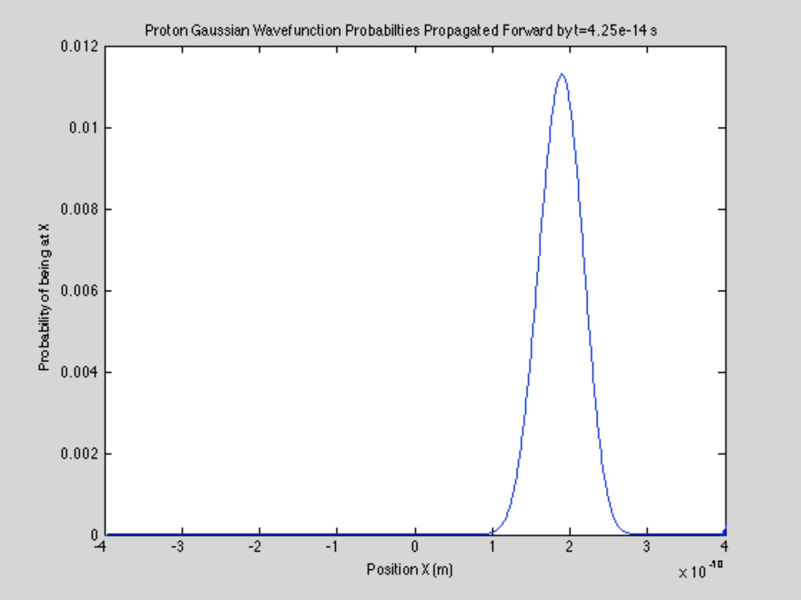
```
tau_radio = -2.8101e+45; tau_IR = -1.6456e+27

tau_UV = 5.0635e-17; tau_xray = 1.5825e-17
```

For Part B:

```
tau_radio = -1.2373e+57; tau_IR = -5.0103e+25

tau_UV = 5.0332e-17; tau_xray = 2.9943e-17
```

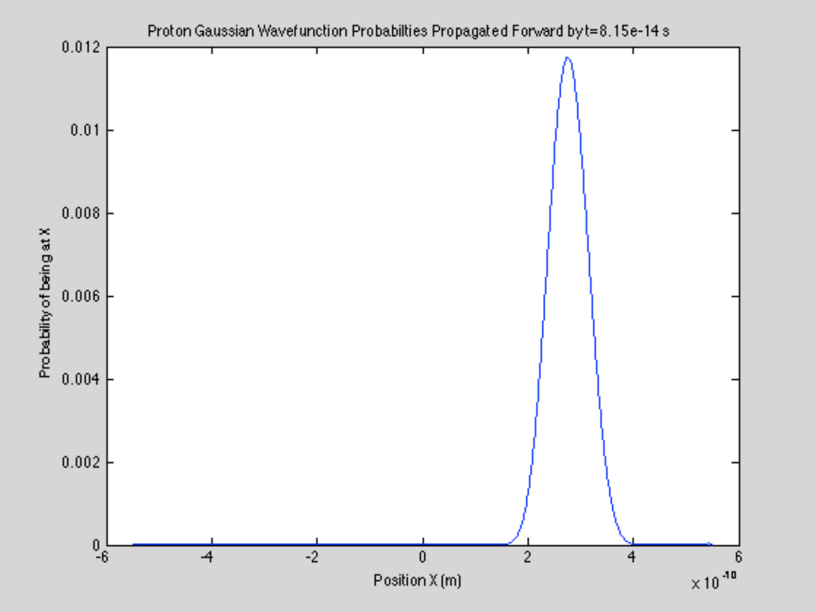
Now, let us introduce time dependence and analyze the proton's wavefunction. To display the probabilities of being in the second well (i.e. tunneling), we can choose a time point at which there is the largest probability of the proton being at Site B. From this, we obtain the following graphs:

Part A:



```
h_tunneling_probability = 0.6251
```

Part B:



```
h_tunneling_probability = 0.5841
```

The parameters used in obtaining these measurements are referenced throughout the results and associated descriptions. They are obtained from a few sources, but the main ones are the different wave frequencies listed above for the radiation energies, as well as the hydrogen bond distances from (1). In addition, the energy barriers for both G-C and A-T base pairs were taken to be the interaction energy values from (2).

A few assumptions were made in order to model this problem. Firstly, it is assumed that the system is at room temperature, 300 K. Secondly, the G-C base pair has three hydrogen bonds with different lengths, and so to make it easier, and average was taken and used as a general distance. The A-T base pair has two hydrogen bonds with different lengths, and so the same thing was done. Thirdly, for the Site B wells, the amount that they are raised from the 0 potential bottom is somewhat arbitrary, as it is loosely based off of how it looks and some values, in (8). However, that estimate is controlled, as it was 2/3 higher for both base pair wells.

Firstly, the probabilities of a quantum jump occurring for the 4 different radiation energies (radio, IR, UV, X-ray) were found to be 1.1574e-04, 0.2099, infinity and infinity, respectively, for part A, and 7.4164e-05, 0.1909, infinity, and infinity for part B. As previously described in the results, a value of $q > 1$ indicates that the energy required to overcome the threshold is already present, and so there is no ‘jump’ to consider the possibility of. This occurs with the UV and X-ray waves in both parts A and B. This makes sense, as, with higher radiation energies, the probability of the proton being able to ‘jump’ to the other Site should increase. We can also see that the quantum jump probabilities for the A-T base pair are very similar to the G-C base pair.

Secondly, the tunneling probabilities for the 4 different radiation energies were found to be 2.1435e-60, 8.3951e-42, 1, and 1 for part A, and 7.9337e-72, 5.4450e-39, 1, and 1 for part B. We can see that the tunneling probabilities for radio and IR radiation, for both G-C and A-T base pairs, are essentially 0, as they are very small, while the tunneling probabilities for UV and X-ray radiation are 1, as the energy that the proton is excited to is greater than the barrier, meaning that the proton will definitely move across to the adjacent Site (i.e. essentially ‘tunnel’ through a 0 J barrier).

Thirdly, both the forward and reverse tunneling rates increase as the radiation energy increases, for both G-C and A-T base pairs. In part A, the forward tunneling rates for radio and IR excitation are both essentially 0, which correspond well with our tunneling probabilities. The reverse tunneling rates for radio and IR excitation are also both essentially 0, and the negatives attached can be explained by the fact the since the forward rates indicate no tunneling, there will be a lack of the proton in the higher potential well to even tunnel back. For the UV and X-ray excitations, the forward and reverse tunneling rates are very similar, with the reverse being slightly lower. This makes sense as, in correspondence with our tunneling probabilities of 1, we expect high tunneling rates. These same trends are observed in part B as well.

For the effective tunneling times, it is important to note that the reason the radio and IR times are negative is due to the negative reverse tunneling values. Ignoring the negatives and simply comparing magnitudes (as the orders of magnitude will remain essentially the same). Taking this into account, the tunneling times decrease as the radiation energy increases, for both G-C and A-T base pairs. This makes sense as the addition of radiation to the system excited the proton to the higher energy levels, and from there it is much easier to tunnel through the barrier, reducing effective tunneling time.

Finally, after introducing time dependence for the proton wavefunction (as a superposition of all the eigenstates), we can see that, as time evolves, there is significant probability of tunneling in both G-C and A-T base pairs. Specifically, at the chosen time points of 4.25e-18 s and 8.15e-18 s, the probabilities of being at Site B were the greatest, and the tunneling probabilities were found to be 0.6251 and 0.5841 respectively.

6. Conclusion:

In summary, a few measurements were obtained in order to give us tunneling information in response to various degrees of energy excitation. These measurements clearly help address the problem that was setup as were able to analyze how different amounts of radiation energy affect the proton tunneling process. For instance, the quantum jump probabilities helped show that increasing radiation energy also increased the likelihood of the proton jumping to a higher state, which would be able to traverse the barrier. The tunneling probabilities based on the quantity s (defined n the results section) also showed that increasing radiation energy increased the tunneling probabilities, and that, essentially, tunneling would not occur for the lower two radiations, and would definitely occur for the higher two radiations. The tunneling rates further support these results, as the radio and IR rates are essentially 0, whereas the rates for UV and X-ray are very high. Finally, from our findings of the time dependence measurement, we can see that there is a significant chance of tunneling occurring as time evolves. From all of these results, we can infer that increasing radiation energy would increase the probability of the proton being in a higher energy state, thus changing the wavefunction and potentially increasing the likelihood of tunneling, and thus potential mutation.

Given the results obtained, my hypothesis was correct. We can see from our analysis of the results that increasing the radiation energy to the proton increases the tunneling probability, and also decreases effective tunneling time. As a result, this would consequentially alter the original nucleotide base’s structure, as the increased proton tunneling would increase the likelihood of the proton being at Site B (which is not its original and non-mutational position), and if replication were to occur during any such time, a permanent mutation in the base sequence would occur. This would result in further mutations, as the complementary strands produced would not contain the same genetic information as the original, non-mutated sequence. One thing that I did not address in my hypothesis was whether there would be any significant differences in the measurements between the G-C and A-T base pairs. From our results, it turns out that, even with different energy barrier and separation distance values (taken from literature), the tunneling measurements were essentially the same in both parts of the analysis. This makes sense as there should not be a significant difference between the two base pairs, as that would indicate that one of them is more likely to experience mutations than the other, which is something that has not been explicitly proven. It is important to note, however, that DNA replication and mutation are extremely complicated processes (even in classical terms), and are much more complex than this simplified model used to analyze them. Nonetheless, it is a good approach to begin understanding how quantum mechanics plays such an important role in such an important biological event, as DNA replication is key to our survival. From here, different models can be extrapolated to other possible mechanisms of mutation i.e. through proton shifting between electron lone pairs on a nucleotide base [(8)]. These models could go even further and also incorporate more information about the specific nucleotide bases (A, T, G, C) and their interactions with each other, thus giving us important information as to the sequence that is being affected.

7. Appendix A:

My Script:
(**Please see attached 11 pages)

Basis_projection:

```
function [ amplitudes ] = basis_projection( wavefunction,
eigenfunctions )
% Takes a wavefunction and projects it onto a set of eigenfunctions,
% returning a set of amplitudes, corresponding to each eigenfunction
%
%--- Parameters ----
% wavefunction: Vector of complex amplitudes
% eigenfunctions: List of eigenfunctions to project onto
%
%--- Returns ----
% amplitudes: Vector of amplitudes

amplitudes = wavefunction*eigenfunctions;
end
```

Gaussian_wavefunction:

```
function [ wavefunction ] = gaussian_wavefunction( N,mu,sigma)
% Generates a normalized gaussian wavefunction
% Default "window" that values will appear from are between 0 and 1.
% if your mu and sigma generate a gaussian either too far away, or too
% wide, it will look weird.
%--- Parameters ----
% N: Number of points
% mu: Center of gaussian
% sigma: Standard deviation of gaussian

%--- Returns ---
% wavefunction: Normalized gaussian wavefunction vector

x = linspace(0,1,N);
wavefunction = normalize_wavefunction(normpdf(x,mu,sigma));
end
```

Dhilan Bekah - 20512356

My_eigensystem:

```
function [ eigenenergies,eigenstates ] = my_eigensystem( hamiltonian,
a )
% This function accepts a Hamiltonina Matrix and outputs a vector of
% eigenenergies, as well as a matrix containing the normalized
eigenstates.
% Each row vector corresponds to an eigenstate
%

[V,D] = eig(hamiltonian); % Eig function gives V (matrix of the
eigenvectors) and D (diagonal matrix containing the eigenenergies)
V = V'; % Transpose since V gives eigenvectors as columns
[row,col] = size(V);
delta = a/col; % Need a passed as a paramter to compute delta in order
to normalize the eigenstates

for i = 1:row
    total = 0;
    for j = 1:col
        total = total + conj(V(i,j))*V(i,j)*delta; % Normalization
process
    end
    V(i,:) = V(i,:)./sqrt(total);
end

eigenenergies = diag(D)';
eigenstates = V;
end
```

My_hamiltonian:

```
function [ hamiltonian_matrix ] = my_hamiltonian( m,a,V )
% This function computes the hamiltonian matrix, given a mass, the
length
% of the well, and a vector of potentials.
% The Hamiltonian is based off of the finite diffence method.

hbar = 1.055e-34; % units of m^2*Kg/s (J*s)
s = size(V);
n = s(2);
w = a/n;

H1 = diag(ones(n,1));
H2 = diag(ones(n-1,1),1) + diag(ones(n-1,1),-1);

hamiltonian_matrix = ((hbar^2)/(m*(w^2)))*H1 -
((hbar^2)/(2*m*(w^2)))*H2 + diag(V);
end
```

Normalize_wavefunction:

```
function [ normalized_wavefunction ] =
normalize_wavefunction( wavefunction )
% Normalizes the given wavefunction
%--- Parameters ----
```

```
% wavefunction: Vector of complex amplitudes

%
%--- Returns ---
% normalized_wavefunction: Returns a normalized wavefunction

normalized_wavefunction = wavefunction /
sqrt(sum(conj(wavefunction).*wavefunction));
end
```

Probabilities:

```
function [ probs ] = probabilities( wavefunction )
% Calculates the probability amplitude of position for a wavefunction
%--- Parameters ----
% wavefunction: Vector of complex amplitudes

%
%--- Returns ---
% probs: Returns a vector of probabilities

probs = conj(wavefunction).*wavefunction;
end
```

Propagator:

```
function [ prop_wavefunction ] = propagator( wavefunction,
eigenfunctions, eigenenergies, time )
% Propagates a wavefunction forward through time
%--- Parameters ----
% wavefunction: Vector of complex amplitudes
% eigenfunctions: 2d Array of Hamiltonian eigenfunctions
% eigenenergies: Array of Hamiltonian eigenenergies
% time: Time to propagate too
%
%--- Returns ---
% prop_wavefunction: New wavefunction that has been propagated forward
in
% time
% Summary of your code here
hbar = 1.055e-34;
amplitudes = basis_projection(wavefunction,eigenfunctions);
propagators = exp(-1j*eigenenergies*time/hbar);
prop_wavefunction =
normalize_wavefunction(transpose(sum(eigenfunctions*diag(amplitudes.* .
..
                    propagators),2)));
end
```

8. References:

1) J. Grunenberg, 'Direct Assessment of Interresidue Forces in Watson−Crick Base Pairs Using Theoretical Compliance Constants', *J. Am. Chem. Soc.*, vol. 126, no. 50, pp. 16310-16311, 2004.

2) J. Leszczynski, *Computational Molecular Biology (Theoretical and computational chemistry ; 8)*. Elsevier Science Limited, 1999.

3) T. Nordlund, *Quantitative Understanding of Biosystems*. Boca Raton: Taylor & Francis, 2011.

4) P. Löwdin, 'Quantum Genetics and the Aperiodic Solid', *Advances in Quantum Chemistry Volume 2*, pp. 213-360, 1966.

5) P. Löwdin, 'Proton Tunneling in DNA and its Biological Implications', *Reviews of Modern Physics*, vol. 35, no. 3, pp. 724-732, 1963.

6) M. Wolfe, 'Quantum Tunneling in DNA', 2013.
http://www.physics.drexel.edu/~bob/Term_Reports/Megan_Wolfe.pdf

7) J. D. Watson and F. H. C. Crick, A Structure for Deoxyribose Nucleic Acid. Nature, 171, 737 (1953).

8) E. Rieper, 'Quantum Coherence in Biological Systems', 2011.
https://www.quantumlah.org/media/thesis/CQT_120507_Elisabeth.pdf

Contents

```
% NE 232 - Project 2 DNA MUTATION v3
% Dhilan Bekah - 20512356
```

PART A: G-C base pair

```
clear;
clc;

hbar = 1.055e-34; % units of m^2*Kg/s (J*s)
eV = 1.602e-19; % units of Joules
mp = 1.67e-27; % units of Kg
T = 300; % units of Kelvin, 'room temperature'
k = 1.381e-23; % units of Joules/Kelvin
a = [1.8e-10, 2.7e-10]; % units of m, average hydrogen bond length btwn base GC and AT base pairs, respectively
E_barrer = [1.4595e-19, 9.0352e-20]; % units of Joules, equivalent to 21Kcal/mol and 13Kcal/mol, respectively
L = [8.0e-10,11e-10]; % units of m, length encompassing both potentials

x = linspace(-L(1)/2,L(1)/2,1000);
for i = 1:length(x) % Constructing the potentials
    if (x(i) >= (-L(1)/2) && x(i) <= 0)
        V(i) = 4.5046*(x(i) + a(1))^2;
    elseif (x(i) > 0 && x(i) <= L(1)/2)
        V(i) = 1.1096*(x(i) - a(1))^2 + 1.1e-19;
    end
end

figure(1)
plot(x,V) % to see the construction of the potentials
title('Construction of Potentials as a function of x');
xlabel('Position (m)');
ylabel('V(x) (Joules)');
axis([-4e-10,4e-10,0,2.2e-19]);

[eigenenergies, eigenstates] = my_eigensystem(my_hamiltonian(mp,L(1),V),L(1));

delta = L(1)/1000;
temp = 0;
for j = 1:length(eigenstates)
    temp = temp + (conj(eigenstates(j,1))*eigenstates(j,1)*delta);
end
temp

temp = 0;
for j = 1:length(eigenstates)
    temp = temp + (conj(eigenstates(j,1))*eigenstates(j,2)*delta);
end
```

```
temp

figure(2)
plot(linspace(1,1000,1000),eigenenergies,'r')
title('Eigenenergies for Proton');
xlabel('n');
ylabel('Eigenenergies (Joules)');

figure(3)
hold on
plot(x,eigenstates(1,:),'r')
plot(x,eigenstates(2,:),'g')
plot(x,eigenstates(3,:),'b')
hold off
title('First three Eigenstates for a Proton');
xlabel('Position (m)');
ylabel('Eigenfunction');
legend('n=1','n=2','n=3')

% Make assumption of input energy is just E = hf, based on frequencies of different waves, i.e. radio, IR, UV, x-rays.
% Omega is determined from difference in eigenenergies 1 and 2 (confirmed with other differences), which should be hbar*omega.
% Since these are numerical solutions, the difference decreases as n increases (approximation error)
omega = 7.3706e+13;
f_radio = 1e8; % Hz
f_IR = 5e13; % Hz
f_UV = 1e16; % Hz
f_xray = 1e18; % Hz
h = 6.626e-34; % units of m^2*Kg/s (J*s)
E_radio = h*f_radio; % units of J
E_IR = h*f_IR; % units of J
E_UV = h*f_UV; % units of J
E_xray = h*f_xray; % units of J

for i = 1:1000
    if (eigenenergies(1) + E_radio) >= eigenenergies(i)
        n_radio = i;
    end
end
for i = 1:1000
    if (eigenenergies(1) + E_IR) >= eigenenergies(i)
        n_IR = i;
    end
end
for i = 1:1000
    if (eigenenergies(1) + E_UV) >= eigenenergies(i)
        n_UV = i;
    end
end
for i = 1:1000
    if (eigenenergies(1) + E_xray) >= eigenenergies(i)
        n_xray = i;
    end
end
```

```
figure(4)
subplot(2,2,1);
plot(x,eigenstates(n_radio,:),'r')
title('Radio Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');
subplot(2,2,2);
plot(x,eigenstates(n_IR,:),'g')
title('IR Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');
subplot(2,2,3);
plot(x,eigenstates(n_UV,:),'b')
title('UV Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');
subplot(2,2,4);
plot(x,eigenstates(n_xray,:),'k')
title('X-Ray Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');

error_radio = abs(((0+0.5)*hbar*omega - eigenenergies(n_radio))/((0+0.5)*hbar*omega) * 100)
error_IR = abs(((9+0.5)*hbar*omega - eigenenergies(n_IR))/((9+0.5)*hbar*omega) * 100)
error_UV = abs(((327+0.5)*hbar*omega - eigenenergies(n_UV))/((327+0.5)*hbar*omega) * 100)
error_xray = abs(((999+0.5)*hbar*omega - eigenenergies(n_xray))/((999+0.5)*hbar*omega) * 100)
% Error % explains the lack of accuracy with higher energies, but the
% calculations can still be done, as the energy values still correspond to
% the excitation, they just do not match up with the corresponding
% analytical eigenenergies and eigenstates at higher n. In fact, it would
% be incorrect to use the analytical solutions at those n found, as they
% would not represent the energy that the proton is excited to.

% "Probablity" of quantum jump from state under consideration to energy threshold
% leading over to other state (values hold true so long as threshold >=
% state energy. If state energy > threshold, values of "probability" exceed
% 1, indicating that the energy required to overcome threshold is already present,
% and so there is no jump to consider the possibility of.
qj_radio = exp(-(4.1431e-20 - eigenenergies(n_radio))/(k*T))
qj_IR = exp(-(4.1431e-20 - eigenenergies(n_IR))/(k*T))
qj_UV = exp(-(4.1431e-20 - eigenenergies(n_UV))/(k*T))
qj_xray = exp(-(4.1431e-20 - eigenenergies(n_xray))/(k*T))

%** TUNNELING PROBABILITY AND EFFECTIVE TIME CALCULATIONS
% By WKB Method:
syms y;
for i = 1:length(x) % Constructing V as a symbolic function in order to integrate
    if (x(i) >= (-L(1)/2) && x(i) <= 0)
        V_f = 4.5046*(y + a(1))^2;
    elseif (x(i) > 0 && x(i) <= L(1)/2)
        V_f = 1.1096*(y - a(1))^2 + 1.1e-19;
    end
end

for i = 275:500
    if V(i) >= eigenenergies(n_radio)
        x1 = i;
        break;
    end
end
for i = 725:-1:500
    if V(i) >= eigenenergies(n_radio)
        x2 = i;
        break;
    end
end
fnc_radio = ((2*mp)*(V_f - eigenenergies(n_radio)))^0.5;
s_radio = double((2*pi/h)*int(fnc_radio,x(x1),x(x2)))

for i = 275:500
    if V(i) >= eigenenergies(n_IR)
        x1 = i;
        break;
    end
end
for i = 725:-1:500
    if V(i) >= eigenenergies(n_IR)
        x2 = i;
        break;
    end
end
fnc_IR = ((2*mp)*(V_f - eigenenergies(n_IR)))^0.5;
s_IR = double((2*pi/h)*int(fnc_IR,x(x1),x(x2)))

for i = 275:500
    if V(i) >= eigenenergies(n_UV)
        x1 = i;
        break;
    else
        x1 = 500;
    end
end
for i = 725:-1:500
    if V(i) >= eigenenergies(n_UV)
        x2 = i;
        break;
    else
        x2 = 500;
    end
end
fnc_UV = ((2*mp)*(V_f - eigenenergies(n_UV)))^0.5;
s_UV = double((2*pi/h)*int(fnc_UV,x(x1),x(x2)))

for i = 275:500
    if V(i) >= eigenenergies(n_xray)
        x1 = i;
        break;
    else
        x1 = 500;
    end
end
for i = 725:-1:500
    if V(i) >= eigenenergies(n_xray)
        x2 = i;
        break;
    else
        x2 = 500;
    end
end
fnc_xray = ((2*mp)*(V_f - eigenenergies(n_xray)))^0.5;
s_xray = double((2*pi/h)*int(fnc_xray,x(x1),x(x2)))

g_radio = exp(-2*s_radio)
g_IR = exp(-2*s_IR)
g_UV = exp(-2*s_UV)
g_xray = exp(-2*s_xray)

v1_radio = (eigenenergies(n_radio) - eigenenergies(1))/h;
v1_IR = (eigenenergies(n_IR) - eigenenergies(1))/h;
v1_UV = (eigenenergies(n_UV) - eigenenergies(1))/h;
v1_xray = (eigenenergies(n_xray) - eigenenergies(1))/h;

v2_radio = (eigenenergies(n_radio) - eigenenergies(1) - 1.1e-19)/h;
v2_IR = (eigenenergies(n_IR) - eigenenergies(1) - 1.1e-19)/h;
v2_UV = (eigenenergies(n_UV) - eigenenergies(1) - 1.1e-19)/h;
v2_xray = (eigenenergies(n_xray) - eigenenergies(1) - 1.1e-19)/h;

c1_radio = v1_radio*g_radio
c1_IR = v1_IR*g_IR
c1_UV = v1_UV*g_UV
c1_xray = v1_xray*g_xray

c2_radio = v2_radio*g_radio
c2_IR = v2_IR*g_IR
c2_UV = v2_UV*g_UV
c2_xray = v2_xray*g_xray

tau_radio = (c1_radio + c2_radio)^(-1)
tau_IR = (c1_IR + c2_IR)^(-1)
tau_UV = (c1_UV + c2_UV)^(-1)
tau_xray = (c1_xray + c2_xray)^(-1)
%}

% Center wavefunction at x = -1.8e-10 (localize particle to Site A center)
gauss_wf = gaussian_wavefunction(1000,0.275,0.045);
%figure(5)
%plot(x,V,x,gauss_wf*1e-19);

% Project wavefunction onto eigenvectors of proton
wvfunction = propagator(gauss_wf,eigenstates,eigenenergies,0);
figure(5)
plot(x,wvfunction)
title('Proton Wavefunction');
xlabel('Position (m)');
ylabel('Probability Distribution');

expect_x_ti = expectation_x(gauss_wf,-L(1)/2,L(1)/2) % Initial state Position expectation
expect_p_ti = expectation_p(gauss_wf,-L(1)/2,L(1)/2) % Initial state Momentum expectation

tf = 5e-16; % time-step, in s
time = zeros(1,201); % 201 time points used
```

```
% Initializing matrices for speed
h_gauss_wf = zeros(201,1000);
h_probabilities = zeros(201,1000);
for i = 1:201
    time(i) = (i-1)*tf;
    h_gauss_wf(i,:) = propagator(gauss_wf,eigenstates,eigenenergies,time(i));
    h_probabilities(i,:) = probabilities(h_gauss_wf(i,:));
end

figure(6)
plot(x,h_probabilities(86,:)); % time(86) chosen as largest probability of proton being at Sit
e B center (1.8e-10)
xlabel('Position X (m)');
ylabel('Probability of being at X');
title(['Proton Gaussian Wavefunction Probabilties Propagated Forward by t=',num2str(time(86)),
' s']);

h_tunneling_probability = 1-sum(h_probabilities(86,1:724))
%{%}
```

PART B: A-T Base Pair

```
clear;
clc;

hbar = 1.055e-34; % units of m^2*Kg/s (J*s)
eV = 1.602e-19; % units of Joules
mp = 1.67e-27; % units of Kg
T = 300; % units of Kelvin, 'room temperature'
k = 1.381e-23; % units of Joules/Kelvin
a = [1.8e-10, 2.7e-10]; % units of m, average hydrogen bond length btwn base GC and AT base pa
irs, respectively
E_barrer = [1.4595e-19, 9.0352e-20]; % units of Joules, equivalent to 21Kcal/mol and 13Kcal/mo
l, respectively
L = [8.0e-10,11e-10]; % units of m, length encompassing both potentials

x = linspace(-L(2)/2,L(2)/2,1000);
for i = 1:length(x) % Constructing the potentials
    if (x(i) >= (-L(2)/2) && x(i) <= 0)
        V(i) = 1.2394*(x(i) + a(2))^2;
    elseif (x(i) > 0 && x(i) <= L(2)/2)
        V(i) = 0.3135*(x(i) - a(2))^2 + 6.75e-20;
    end
end

figure(1)
plot(x,V) % to see the construction of the potentials
title('Construction of Potentials as a function of x');
xlabel('Position (m)');
ylabel('V(x) (Joules)');
axis([-5.4e-10,5.4e-10,0,1e-19]);

[eigenenergies, eigenstates] = my_eigensystem(my_hamiltonian(mp,L(2),V),L(2));

delta = L(2)/1000;
temp = 0;
```

```
for j = 1:length(eigenstates)
    temp = temp + (conj(eigenstates(j,1))*eigenstates(j,1)*delta);
end
temp

temp = 0;
for j = 1:length(eigenstates)
    temp = temp + (conj(eigenstates(j,1))*eigenstates(j,2)*delta);
end
temp

figure(2)
plot(linspace(1,1000,1000),eigenenergies,'r')
title('Eigenenergies for Proton');
xlabel('n');
ylabel('Eigenenergies (Joules)');

figure(3)
hold on
plot(x,eigenstates(1,:),'r')
plot(x,eigenstates(2,:),'g')
plot(x,eigenstates(3,:),'b')
hold off
title('First three Eigenstates for a Proton');
xlabel('Position (m)');
ylabel('Eigenfunction');
legend('n=1','n=2','n=3')

% Make assumption of input energy is just E = hf, based on frequencies of different waves, i.e. radio, IR, UV, x-rays.
% Omega is determined from difference in eigenenergies 1 and 2 (confirmed with other differences), which should be hbar*omega.
% Since these are numerical solutions, the difference decreases as n increases (approximation error)
omega = 3.8562e+13;
f_radio = 1e8; % Hz
f_IR = 5e13; % Hz
f_UV = 1e16; % Hz
f_xray = 1e18; % Hz
h = 6.626e-34; % units of m^2*Kg/s (J*s)
E_radio = h*f_radio; % units of J
E_IR = h*f_IR; % units of J
E_UV = h*f_UV; % units of J
E_xray = h*f_xray; % units of J

for i = 1:1000
    if (eigenenergies(1) + E_radio) >= eigenenergies(i)
        n_radio = i;
    end
end
for i = 1:1000
    if (eigenenergies(1) + E_IR) >= eigenenergies(i)
        n_IR = i;
    end
end
for i = 1:1000
    if (eigenenergies(1) + E_UV) >= eigenenergies(i)
```

```
        n_UV = i;
    end
end
for i = 1:1000
    if (eigenenergies(1) + E_xray) >= eigenenergies(i)
        n_xray = i;
    end
end

figure(4)
subplot(2,2,1);
plot(x,eigenstates(n_radio,:),'r')
title('Radio Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');
subplot(2,2,2);
plot(x,eigenstates(n_IR,:),'g')
title('IR Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');
subplot(2,2,3);
plot(x,eigenstates(n_UV,:),'b')
title('UV Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');
subplot(2,2,4);
plot(x,eigenstates(n_xray,:),'k')
title('X-Ray Excitation Proton State');
xlabel('Position (m)');
ylabel('Eigenfunction');

error_radio = abs(((0+0.5)*hbar*omega - eigenenergies(n_radio))/((0+0.5)*hbar*omega) * 100)
error_IR = abs(((8+0.5)*hbar*omega - eigenenergies(n_IR))/((8+0.5)*hbar*omega) * 100)
error_UV = abs(((561+0.5)*hbar*omega - eigenenergies(n_UV))/((561+0.5)*hbar*omega) * 100)
error_xray = abs(((999+0.5)*hbar*omega - eigenenergies(n_xray))/((999+0.5)*hbar*omega) * 100)
% Error % explains the lack of accuracy with higher energies, but the
% calculations can still be done, as the energy values still correspond to
% the excitation, they just do not match up with the corresponding
% analytical eigenenergies and eigenstates at higher n. In fact, it would
% be incorrect to use the analytical solutions at those n found, as they
% would not represent the energy that the proton is excited to.

% "Probablity" of quantum jump from state under consideration to energy threshold
% leading over to other state (values hold true so long as threshold >=
% state energy. If state energy > threshold, values of "probability" exceed
% 1, indicating that the energy required to overcome threshold is already present,
% and so there is no jump to consider the possibility of.
qj_radio = exp(-(4.1431e-20 - eigenenergies(n_radio))/(k*T))
qj_IR = exp(-(4.1431e-20 - eigenenergies(n_IR))/(k*T))
qj_UV = exp(-(4.1431e-20 - eigenenergies(n_UV))/(k*T))
qj_xray = exp(-(4.1431e-20 - eigenenergies(n_xray))/(k*T))

%** TUNNELING PROBABILITY AND EFFECTIVE TIME CALCULATIONS
% By WKB Method:
syms y;
for i = 1:length(x) % Constructing V as a symbolic function in order to integrate
    if (x(i) >= (-L(2)/2) && x(i) <= 0)
```

```
        V_f = 1.2394*(y + a(2))^2;
    elseif (x(i) > 0 && x(i) <= L(2)/2)
        V_f = 0.3135*(y - a(2))^2 + 6.75e-20;
    end
end

for i = 255:500
    if V(i) >= eigenenergies(n_radio)
        x1 = i;
        break;
    end
end
for i = 745:-1:500
    if V(i) >= eigenenergies(n_radio)
        x2 = i;
        break;
    end
end
fnc_radio = ((2*mp)*(V_f - eigenenergies(n_radio)))^0.5;
s_radio = double((2*pi/h)*int(fnc_radio,x(x1),x(x2)))

for i = 255:500
    if V(i) >= eigenenergies(n_IR)
        x1 = i;
        break;
    end
end
for i = 745:-1:500
    if V(i) >= eigenenergies(n_IR)
        x2 = i;
        break;
    end
end
fnc_IR = ((2*mp)*(V_f - eigenenergies(n_IR)))^0.5;
s_IR = double((2*pi/h)*int(fnc_IR,x(x1),x(x2)))

for i = 255:500
    if V(i) >= eigenenergies(n_UV)
        x1 = i;
        break;
    else
        x1 = 500;
    end
end
for i = 745:-1:500
    if V(i) >= eigenenergies(n_UV)
        x2 = i;
        break;
    else
        x2 = 500;
    end
end
fnc_UV = ((2*mp)*(V_f - eigenenergies(n_UV)))^0.5;
s_UV = double((2*pi/h)*int(fnc_UV,x(x1),x(x2)))

for i = 255:500
    if V(i) >= eigenenergies(n_xray)
```

```
        x1 = i;
        break;
    else
        x1 = 500;
    end
end
for i = 745:-1:500
    if V(i) >= eigenenergies(n_xray)
        x2 = i;
        break;
    else
        x2 = 500;
    end
end
fnc_xray = ((2*mp)*(V_f - eigenenergies(n_xray)))^0.5;
s_xray = double((2*pi/h)*int(fnc_xray,x(x1),x(x2)))

g_radio = exp(-2*s_radio)
g_IR = exp(-2*s_IR)
g_UV = exp(-2*s_UV)
g_xray = exp(-2*s_xray)

v1_radio = (eigenenergies(n_radio) - eigenenergies(1))/h;
v1_IR = (eigenenergies(n_IR) - eigenenergies(1))/h;
v1_UV = (eigenenergies(n_UV) - eigenenergies(1))/h;
v1_xray = (eigenenergies(n_xray) - eigenenergies(1))/h;

v2_radio = (eigenenergies(n_radio) - eigenenergies(1) - 6.75e-20)/h;
v2_IR = (eigenenergies(n_IR) - eigenenergies(1) - 6.75e-20)/h;
v2_UV = (eigenenergies(n_UV) - eigenenergies(1) - 6.75e-20)/h;
v2_xray = (eigenenergies(n_xray) - eigenenergies(1) - 6.75e-20)/h;

c1_radio = v1_radio*g_radio
c1_IR = v1_IR*g_IR
c1_UV = v1_UV*g_UV
c1_xray = v1_xray*g_xray

c2_radio = v2_radio*g_radio
c2_IR = v2_IR*g_IR
c2_UV = v2_UV*g_UV
c2_xray = v2_xray*g_xray

tau_radio = (c1_radio + c2_radio)^(-1)
tau_IR = (c1_IR + c2_IR)^(-1)
tau_UV = (c1_UV + c2_UV)^(-1)
tau_xray = (c1_xray + c2_xray)^(-1)
%}

% Center wavefunction at x = -2.7e-10 (localize particle to Site A center)
gauss_wf = gaussian_wavefunction(1000,0.255,0.045);
%figure(5)
%plot(x,V,x,gauss_wf*1e-19);

% Project wavefunction onto eigenvectors of proton
wvfunction = propagator(gauss_wf,eigenstates,eigenenergies,0);
figure(5)
plot(x,wvfunction)
```

```
title('Proton Wavefunction');
xlabel('Position (m)');
ylabel('Probability Distribution');

expect_x_ti = expectation_x(gauss_wf,-L(2)/2,L(2)/2) % Initial state Position expectation
expect_p_ti = expectation_p(gauss_wf,-L(2)/2,L(2)/2) % Initial state Momentum expectation

tf = 5e-16; % time-step, in s
time = zeros(1,201); % 201 time points used
% Initializing matrices for speed
h_gauss_wf = zeros(201,1000);
h_probabilities = zeros(201,1000);
for i = 1:201
    time(i) = (i-1)*tf;
    h_gauss_wf(i,:) = propagator(gauss_wf,eigenstates,eigenenergies,time(i));
    h_probabilities(i,:) = probabilities(h_gauss_wf(i,:));
end

figure(6)
plot(x,h_probabilities(164,:)); % time(164) chosen as largest probability of proton being at S
ite B center (2.7e-10)
xlabel('Position X (m)');
ylabel('Probability of being at X');
title(['Proton Gaussian Wavefunction Probabilties Propagated Forward by t=',num2str(time(164))
,' s']);

h_tunneling_probability = 1-sum(h_probabilities(164,1:744))
%{%}
```

Published with MATLAB® R2013a