Bachelor's thesis
Theoretical physics

Numerical study of alpha decay

Jacob Eskin

2016

Supervisor: Kimmo Tuominen

Examiner: Kimmo Tuominen

UNIVERSITY OF HELSINKI
DEPARTMENT OF PHYSICS

P.O. Box 64 (Gustaf Hällströmin katu 2)
FI-00014 University of Helsinki

Contents

1   Introduction

Historically alpha decay has played an important role in both understanding radioactive decay of heavy nuclei and experimental verification of non-relativistic quantum mechanics. One of the great puzzles in the beginning of the study of alpha decay was an observation that, although the energies of the emitted α-particles from different isotopes varied quite little, the corresponding half-lives varied by multiple orders of magnitude. One of the first examinations of alpha decay as a barrier penetration process was conducted by Georg Gamow at the end of 1920's using only the Coulomb potential [1] and it successfully demonstrated the exponential dependency between the half-life and α-particle energy. Though qualitatively this description was successful, quantitatively it was unsatisfying. Still it took the step of introducing a purely quantum mechanical process into explaining this phenomenon; tunneling has no classical analog.

In alpha decay a heavy parent nucleus decays into an α-particle, nucleus of a helium atom He^4, and a lighter daughter nucleus. The effect is driven by the repulsive Coulomb force overcoming the strong interaction inside the nucleus: as the size of the nucleus grows the Coulomb force grows faster than the binding force. Since the α-particle is very stable, tightly bound and light, it becomes the main emitted particle since this emission allows for the largest energy release [1].

Using now a more realistic potential energy model, numerical techniques become mandatory in treating the alpha decay as a barrier penetration phenomenon. The potential that creates the barrier consists of the Coulomb potential that comes from the electrostatic interaction and the Woods-Saxon potential that is often used to model the strong interaction. In calculating the tunneling probability Wenzel-Kramers-Brillouin (WKB) approximation for the α-particle wave function is used. Isotopes studied with this model are even-even nuclei of Uranium from U^{222} to U^{238}. Since the form of the potential energy is very nonlinear, calculating parameters and solutions for the tunneling probability are done with numerical methods. MATLAB is used for all numerical and graphical analysis in this paper and accuracy of the model and the numerical techniques are analyzed by comparing the calculated half-lives with measured half-lives.

2   The model

The approach and model described will be somewhat semiclassical in nature. At the moment of decay the α-particle can be thought of having been preformed in the parent nucleus, i.e. at the moment of decay the parent nucleus consists of the α-particle and the daughter nucleus. The potential in which the α-particle moves is also determined by the α-particle and the

daughter nucleus, and is for simplicity assumed to be spherically symmetric. Consequently, the tunneling process can be treated in one dimension. It is also assumed that the alpha decay happens from the first positive energy state, so called quasibound state, and that the energy of this state is no less than E_α. For simplicity only decays happening into the $l = 0$ channel, with the initial and final nuclear spins being $I_i = I_f = 0$, are considered.

As the α-particle wave function we use the WKB wave function $\psi(r_t) = \psi(r_0)e^{-\sigma}$. The α-particle hits the potential barrier at point r_1 from inside the parent nucleus with a frequency f_0 and has a probability $|\psi(r_2)|^2 = |\psi(r_1)e^{-\sigma}|^2$ of tunneling to the point r_2, points r_1 and r_2 being the beginning and end of the tunnel and σ the WKB factor. These can be combined in a form of decay probability per unit time, i.e. as the decay constant

$$\lambda = f_0 P e^{-2\sigma}, \tag{2.1}$$

where $|\psi(r_1)|^2 \equiv P$, P being the preformation probability taken for now to be 1. The WKB factor is

$$\sigma = \frac{1}{\hbar} \int_{r_1}^{r_2} \sqrt{2m(V(r) - E_\alpha)} dr, \tag{2.2}$$

where m is the mass of the α-particle. Potential energy in this model is taken to be the sum of the Coulomb potential $V_c(r)$, describing the electrostatic interaction, and the Woods-Saxon potential $V_{w-s}(r)$, describing the strong interaction. The Coulomb interaction is taken to have the form of a point particle of charge $2e$ interacting with a spherical charge distribution with radius w. Thus the potentials are

$$V_c(r) = \begin{cases} 2(Z-2)e^2 \dfrac{3w^2 - r^2}{2w^3} & (r \leq w) \\[3mm] \dfrac{2(Z-2)e^2}{r} & (r > w) \end{cases} \tag{2.3}$$

$$V_{w-s}(r) = \frac{V_0}{exp(\frac{r-w}{a}) + 1}, \tag{2.4}$$

where Z is the parent nucleus proton number, w is radius of the daughter nucleus, a is the so called "surface thickness" and V_0 is the strength of the Woods-Saxon potential. The frequency f_0 is obtained by dividing the α-particles' nonrelativistic velocity with the diameter of the potential well at hight r_1:

$$f_0 = \frac{v}{2r_1}, \tag{2.5}$$

$$f_0 = \sqrt{\frac{2\varepsilon}{m}} \frac{1}{2r_1}, \tag{2.6}$$

3

where $\varepsilon = E_\alpha - (V_0 + V_c(0))$ is the kinetic energy of the α-particle inside the nucleus. With all this, the decay constant can be written, with $P \equiv 1$, as

$$\lambda = \sqrt{\frac{2(E_\alpha - (V_0 + V_c(0)))}{m}} \frac{e^{-2\frac{1}{\hbar} \int_{r_1}^{r_2} \sqrt{2m(V(r) - E_\alpha)} dr}}{2r_1}. \tag{2.7}$$

Parameters a and w will be fixed as [2] $a = 0.8$ fm and $w = 1.3A^{1/3}$ fm, where A is mass number of the daughter nucleus. In order to calculate the decay constant, still a few parameters need to be determined: V_0, r_1, r_2 and the WKB factor σ. The case where $P < 1$ is discussed later. V_0 term will be solved numerically from the Schrödinger equation, and when this is known r_1 and r_2 can be solved from the relation $V(r_1) = V(r_2) = E_\alpha$. When these variables have been solved, the WKB factor σ will be evaluated numerically as well due to the form of the potential $V(r)$.

3 Numerical schemes for V_0 and the WKB factor

For a spherically symmetric system that satisfies $I_i = I_f = 0$ and $l = 0$, the radial part of the Schrödinger equation can be written in a form

$$\frac{-\hbar}{2m}\left(\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dR(r)}{dr}\right)\right) + V(r)R(r) = ER(r). \tag{3.1}$$

Making the substitution $u(r) = R(r)/r$, the equation simplifies further into the form

$$\frac{-\hbar}{2m}\frac{d^2u(r)}{dr^2} + V(r)u(r) = Eu(r), \tag{3.2}$$

where $V(r) = V_c(r) + V_{w-s}(r)$. Bound state solutions of equation (3.2) have to satisfy the boundary conditions $u(r) = 0$ when $r = 0$ and $u(r) \to 0$ when $r \to \infty$ coming from the requirement on $R(r)$ to be square-integrable. Writing the Taylor expansions of $u(r + \delta r)$ and $u(r - \delta r)$ around r

$$u(r + \delta r) = u(r) + \delta r \frac{du(r)}{dr} + \frac{\delta r^2}{2}\frac{d^2u(r)}{dr^2} + \frac{\delta r^3}{6}\frac{d^3u(r)}{dr^3} + \frac{\delta r^4}{24}\frac{d^4u(r)}{dr^4} + ..., \tag{3.3}$$

$$u(r - \delta r) = u(r) - \delta r \frac{du(r)}{dr} + \frac{\delta r^2}{2}\frac{d^2u(r)}{dr^2} - \frac{\delta r^3}{6}\frac{d^3u(r)}{dr^3} + \frac{\delta r^4}{24}\frac{d^4u(r)}{dr^4} - ..., \tag{3.4}$$

and adding (3.4) to (3.3), a finite difference approximation for the second derivate of $u(r)$ is obtained:

$$\frac{d^2u(r)}{dr^2} = \frac{u(r + \delta r) - 2u(r) + u(r - \delta r)}{\delta r^2} - \frac{\delta r^4}{12}\frac{d^4u(r)}{dr^4} - O(\delta r^6), \tag{3.5}$$

$$\frac{d^2u(r)}{dr^2} \approx \frac{u(r + \delta r) - 2u(r) + u(r - \delta r)}{\delta r^2}, \tag{3.6}$$

where the leading term of the truncation error is

$$err_{fdm} = \frac{\delta r^4}{12} \frac{d^4 u(r)}{dr^4}.$$ (3.7)

Substituting (3.6) into (3.2) the following form is obtained:

$$u(r + \delta r) = 2u(r) - u(r - \delta r) + \delta r^2 \frac{2m}{\hbar}(V(r) - E)u(r).$$ (3.8)

When this is discretized by dividing the domain of $u(r)$ into intervals of length δr and setting $\delta r \equiv \Delta$, $r \equiv r[i] = \Delta i$, $u(r) \equiv u(r[i]) = u[i]$ and $V(r) \equiv V(r[i]) = V[i]$, where $i \in \mathbb{N}$, the recursion relation

$$u[i+1] = (2 + \Delta^2 \frac{2m}{\hbar}(V[i] - E))u[i] - u[i-1],$$ (3.9)

with

$$V_c[i] = \begin{cases} 2(Z-2)e^2 \dfrac{3w^2 - (\Delta i)^2}{2w^3} & (\Delta i \leq w) \\[3mm] \dfrac{2(Z-2)e^2}{\Delta i} & (\Delta i > w) \end{cases}$$ (3.10)

$$V_{w-s}[i] = \frac{V_0}{exp(\frac{\Delta i - w}{a}) + 1}.$$ (3.11)

is obtained. Making use of (3.9) and the boundary conditions for $u(r)$, bound states of the α-particle can be solved using the so called bisection method. Using a boundary condition that for a quasibound state $u(r_2)$ is a minimum [2] of $u(r)$, the same bisection method can be used without modification to solve that state as well. This way V_0 may be found by imposing a condition for the quasibound state as will be shown below.

The WKB factor is calculated by means of numerical integration by dividing the integration interval $[r_1, r_2]$ into N very small subintervals of length $\delta x = (r_2 - r_1)/N$ and approximating the integrand $f(r)$ with the Taylor series in a point r_i for a subinterval $[r_i, r_i + \delta r]$:

$$\int_{r_i}^{r_i + \delta r} f(r')dr' = \int_{r_i}^{r_i + \delta r} (f(r_i) + (r' - r_i)\frac{df(r_i)}{dr} + \frac{(r' - r_i)^2}{2}\frac{d^2 f(r_i)}{dr^2} + O((r' - r_i)^3))dr'.$$ (3.12)

Writing $f(r_i) = f_i$, $\dfrac{df(r_i)}{dr} = f_i'$, $\dfrac{d^2 f(r_i)}{dr^2} = f_i''$ and doing the substitution

$$r' = y\delta r + r_i \longrightarrow y = \frac{r - r_i}{\delta r} = \frac{r - r_i}{(r_i + \delta r) - r_i},$$

the integral becomes

$$\delta r \int_0^1 [f_i + y\delta r f_i' + \frac{(y\delta r)^2}{2}f_i'' + O((y\delta r)^3)dy] = \delta r f_i + \frac{\delta r^2}{2}f_i' + \frac{\delta r^3}{6}f_i'' + O(\delta r^4). \quad (3.13)$$

The first derivative may be written according to equation (3.3) in the form

$$\frac{df(r_i)}{dr} = \frac{f(r_i + \delta r) - f(r_i)}{\delta r} - \frac{\delta r}{2}\frac{d^2 f(r_i)}{dr^2} - O(\delta r^2), \quad (3.14)$$

and substituting this on the right hand side of equation (3.13) the integral for one subinterval becomes

$$\int_{r_i}^{r_i + \delta r} f(r')dr' = \frac{\delta r}{2}(f(r_i + \delta r) + f(r_i)) - \frac{\delta r^3}{12}\frac{d^2 f(r_i)}{dr^2} - O(\delta r^4). \quad (3.15)$$

For the whole interval $[r_1, r_2]$ all subintervals are summed together, and dropping higher order terms

$$\int_a^b f(r)dr = \sum_{i=0}^{N-1} \int_{r_i}^{r_i + \delta r} f(r')dr' = \sum_{i=0}^{N-1} \frac{\delta r}{2}(f(r_i + \delta r) + f(r_i)). \quad (3.16)$$

This is called the trapezoid method, and the leading term in truncation error is

$$err_{tpz} = -\frac{\delta r^3}{12}\sum_{i=0}^{N-1}\frac{d^2 f(r_i)}{dr^2} = -\frac{\delta r^3 N}{12}\sum_{i=0}^{N-1}\frac{\dfrac{d^2 f(r_i)}{dr^2}}{N} = -\frac{\delta r^2 (r_2 - r_1)}{12}\frac{d^2 f(\xi)}{dr^2}, \quad (3.17)$$

where the last step comes from the continuity of $f(r)$ in the integration interval, ξ being a point in that interval.

3.1 Bisection method

Bound states of the α-particle can be solved in the following way: Set some initial value v_0 for V_0 and set an initial value for $E_\alpha = E$ that is slightly larger than v_0. Set also the following for initial values of $u[i]$: $u[0] = 0$ and $u[1] = 1$ and choose $\Delta = 0.01$ fm. The goal is to find, by iterating (3.9) over and over again to some distance $R \gg r_1$, an energy level E such that the boundary condition $u(r) \to 0$ when $r \to \infty$ is satisfied in the form $|u[R]| \ll 1$. After the first iteration, the value $u[R] = u_E$ is saved. The energy is then updated by setting

$$E = E + \delta E,$$

after which the iteration of (3.9) is carried out again with this new value of E in order to find a new value $u[R] = u_{E+\delta E}$. These values for the radial wave function are then compared, and if they have the same sign, i.e. $u_E u_{E+\delta E} > 0$, the following update is made:

6

$$u_E = u_{E+\delta E},$$

$$E = E + \delta E.$$

If the values for the radial wave function have different signs, i.e $u_E u_{E+\delta E} < 0$, the energy level sought is between E and $E + \delta E$, and the following update is made:

$$\delta E = -\delta E/2,$$

$$u_E = u_{E+\delta E},$$

$$E = E + \delta E.$$

Continuing this iteration and updating scheme the value E converges to the bound state energy value E_1 for which $|u[R]|$ has a minimum. This iteration is continued until the desired accuracy is reached: the accuracy is determined by the order of magnitude of δE and when this is small enough, iteration is stopped when a value of E for which $u_E u_{E+\delta E} < 0$ is reached. For finding the next bound energy level, a new E is set to a value just above the previous bound energy level, and the whole above scheme is repeated in order to find the second bound state energy level E_2. This is then continued to find all the bound state energy levels.

As stated above, this same method can be used for finding the first quasibound energy level. Even though there is a difference in the boundary condition, it does not affect the practical calculations and the above process is thus extended to finding the first positive energy level. The way this allows for the determination of the parameter V_0 is as follows: when the first quasibound energy level is known, V_o can be found by the same bisection method described above, only instead of trying to find the energy level E by adjusting it to meet the boundary condition for $u(r)$, V_0 is adjusted to meet the condition for the first quasibound energy level.

With the above method, the first quasibound energy level $E = E_p$ is calculated. If $E_p > E_\alpha$,

$$v_0 = v_0 + \Delta v_0.$$

This is continued as long as $E_p < E_\alpha$. Now the iterative part begins. A new quasibound energy level E is calculated, and analogously to the above, if both the previous and the new energy levels are on the same side of E_α, i.e. $(E_\alpha - E_p)(E_\alpha - E) > 0$ then

$$v_0 = v_0 + \delta v_0,$$

$$E_p = E,$$

and again of course if they are on different sides of E_α, i.e. $(E_\alpha - E_p)(E_\alpha - E) < 0$, the desired value for V_0 is between v_0 and $v_0 + \delta v_0$. In this case set

$$\delta v_0 = -\delta v_0/2,$$

$$v_0 = v_0 + \delta v_0,$$

$$E_p = E,$$

and repeat until the desired accuracy for v_0 is reached. This way v_0 converges to the value for which the first quasibound state is no less and as close as possible to E_α. The process is terminated when the accuracy is correct, $E_p < E_\alpha$ and $(E_\alpha - E_p)(E_\alpha - E) < 0$. In practice the iteration of the Schrödinger equation does not have to be done all the way up to r_2. Also suitable initial values for v_0, δE, δv_0 and Δv_0 should be chosen in order to keep the amount of iterations as small as possible while avoiding jumping over energy levels.

4 Calculating the half-life

As mentioned above, due to the highly nonlinear nature of the potential energy term, calculating the half-life of an isotope with this model has to be done numerically. For this purpose four m-files were written with MATLAB consisting of six functions, each one performing a specific task. In practice there is only need to call one main function that in turn takes care of calling the others when needed. The main function is written in the file `lambda.m`, but the presentation will proceed in the order the actual calculations take place. All the m-files can be found in the Appendix A.

First value to be calculated that is needed for solving the half-life is the V_0 term of the Woods-Saxon potential. This calculation is done by three functions written in the m-file `Vo.m`. Initial values $v_0 = -100$ MeV, $E = v_0 + 1$ MeV, $\delta E = 0.5$ MeV, $\delta v_0 = -1$ MeV and $\Delta v_0 = \pm 2$ MeV are chosen. The structure of this file is as follows: the main function is `Vo()`, and this function calls the function `Elevel()` which in turn calls the function `schr()`. The function `schr()` performs the iteration of equation (3.9) to a distance of $R = 20$ fm. After this the function `Elevel()` will evaluate the terms u_E, $u_{E+\delta E}$ and δE in order to determine if the bound or quasibound energy level has been found, and if not it does the appropriate updates and calls for another iteration of `schr()`. If a quasibound level has been found then the main function of the m-file, `Vo()`, does a similar evaluation in order to determine if the correct values δv_0, E and E_p have been found. If so, the value of $V_0 = v_0$ is returned for rest of the calculations, and if not then the proper updates are done and `Elevel()` is called again.

An important note on a big difference between how the model was presented in Section 3 and how the practical calculations for specifically even-even Uranium isotopes were done: It was found by Duarte and Siegel [2] and confirmed by test running `Vo()`, that for even-even

Uranium isotopes the radial quantum number n of the quasibound state is always 10. Thus in order to simplify the function `Vo()` the actual evaluation of the above terms is done only if the function `Elevel()` finds that the quasibound state is the 10th energy level. This is why the initial value of Δv_0 is ± 2 MeV: since V_0 determines the density of the energy levels, if $n < 10$ then $\Delta v_0 = -2$ MeV and if $n > 10$ then $\Delta v_0 = +2$ MeV, and in both cases $v_0 = v_0 + \Delta v_0$.

After V_0 has been calculated it is now possible to calculate values for $V(r)$. Thus the next step is to find r_1 and r_2. This is done by the function `radii()` that is written in the m-file with the same name. This function calculates the value of $V(r)$ from 0.01 fm to 100 fm, and searches for the values of r for which the relation $V(r_1) = V(r_2) = E_\alpha$ holds as accurately as possible. It also returns, among other parameters, the values of $V(r)$ in the interval $[r_1, r_2]$ which are needed for the calculation of the WKB factor.

The m-file responsible for calculating the WKB factor is `sigma.m`. An existing MATLAB function was used in performing the integration with a step length of 0.01 fm. The WKB factor is then returned to the main function `lambda()` which in turn returns the value of the decay constant λ. With this it is straightforward to calculate the half-life: $T_{1/2}^c = \frac{\ln(2)}{\lambda}$. The superscript c denotes this being the calculated half-life.

5 Results and analysis

All experimental data for the this section is gathered from the National Nuclear Data Center of Brookhaven National Laboratory [3].

Due to the half-lives varying in such a large range, from 10^{17} seconds to 10^{-7} seconds, it is most convenient in any graphical analysis to use the base 10 logarithm of the half-lives. Figure 1 shows the general trend of how the half-lives of Uranium isotopes change with the α-particle energy. Isotopes U^{237}, U^{231} and U^{229} were excluded since U^{231} does not decay via α-emission and U^{229} and U^{237} decay through mostly other means. This trend however breaks down for the isotopes around U^{218}, probably because the neutron number for this isotope is 126, a magic number, and thus is more stable. As stated, the model above describes even-even isotopes for the range U^{238} to U^{222}, which fall well inside this trend.

Analysis of the results are divided into three parts. In the first part the calculations were made assuming the preformation factor P was 1. In the second part the results were corrected by searching for a fit for P from the data. In the third, some analysis is done on the errors coming from above numerical analysis.
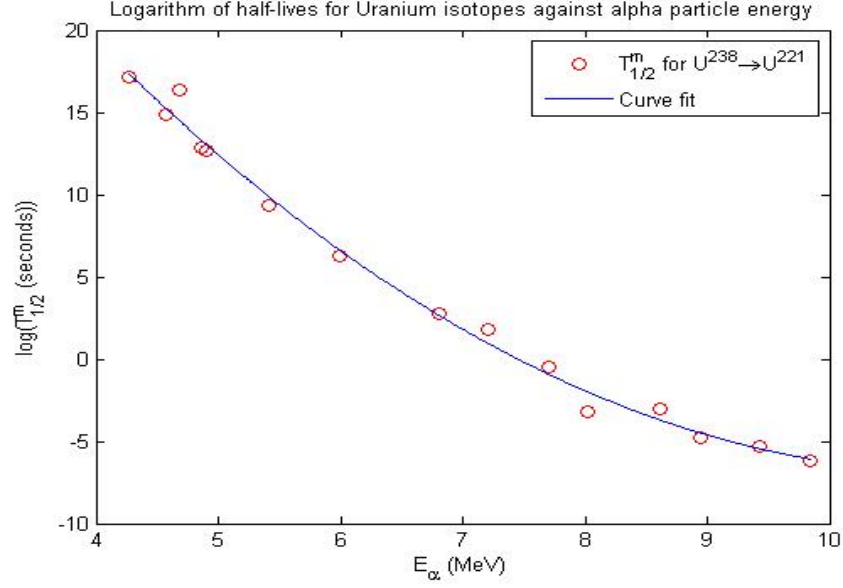
Figure 1: Half-life of Uranium isotopes from U^{238} to U^{221} plotted against E_α, excluding isotopes U^{237}, U^{231} and U^{229}. At this range E_α increases as the mass number decreases.

5.1 $P = 1$
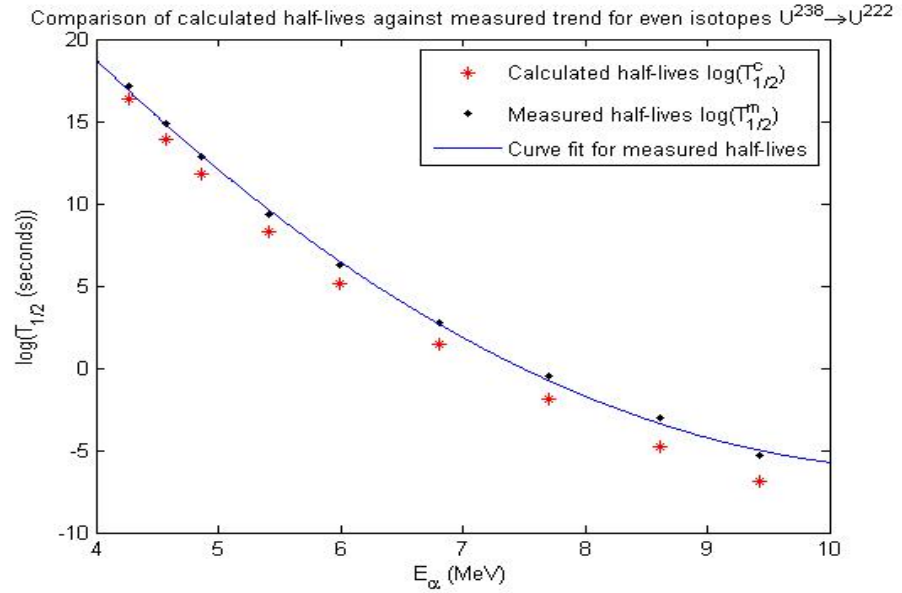
Figure 2: Comparison of calculated half-lives against the measured half-lives with a curve fitted for the measured half-lives.

As can be seen from Figure 2, the calculated half-lives follow the shape of the measured trend quite well, especially for heavier isotopes. Even though the above model is able to reproduce qualitatively somewhat the behavior observed in nature, the actual quantitative results are not so realistic. From Table 1 below it can be seen that the calculated half-

Isotope	E_α MeV	$T_{1/2}^m$ s	$T_{1/2}^c$ s	V_0 MeV
U^{238}	4.2697	$1.409 \cdot 10^{17}$	$2.121 \cdot 10^{16}$	-115.71
U^{236}	4.5732	$7.386 \cdot 10^{14}$	$7.807 \cdot 10^{13}$	-115.84
U^{234}	4.8577	$7.742 \cdot 10^{12}$	$6.794 \cdot 10^{11}$	-116
U^{232}	5.4136	$2.173 \cdot 10^{9}$	$1.872 \cdot 10^{8}$	-115.78
U^{230}	5.9928	$1.797 \cdot 10^{6}$	$1.331 \cdot 10^{5}$	-115.54
U^{228}	6.8041	$5.46 \cdot 10^{2}$	$2.685 \cdot 10^{1}$	-114.96
U^{226}	7.7014	$3.5 \cdot 10^{-1}$	$1.2 \cdot 10^{-2}$	-114.27
U^{224}	8.620	$9 \cdot 10^{-4}$	$1.732 \cdot 10^{-5}$	-113.53
U^{222}	9.430	$4.7 \cdot 10^{-6}$	$1.317 \cdot 10^{-7}$	-112.94
U^{218}	8.776	$5.1 \cdot 10^{-4}$	$7.924 \cdot 10^{-6}$	-115.94

Table 1: Measured and calculated values for half-lives: T^m is for measured and T^c is for calculated. In the bottom is data for U^{218} which has a magic number of neutrons and the break in above mentioned trend is seen.

lives are consistently lower than the experimental values by around one order of magnitude. The calculations indicate that the modeled isotopes decay faster than is observed in nature and this behavior increases as the mass number decreases. This might be the result of the assumption $|\psi(r_1)|^2 \equiv P = 1$. The preformation factor P is the probability that an α-particle has preformed in the parent nucleus, i.e. 2 protons and 2 neutrons have clustered together inside the parent nucleus to form an α-particle at r_1 at a certain point in time and having this value at 1 means that there constantly exists a preformed α-particle. Also simplifying the description of the decays considered to only situations where the α-particle does not carry away any angular momentum surely has an appreciable effect on the calculated results.

5.2 $P < 1$

Finding the preformation factor is done in the following way: As can be seen from Figure 3 the relationship between $\log(T_{1/2}^m)$ is $\log(T_{1/2}^c)$ linear, and can be written in the form

$$\log(T_{1/2}^m) = a \cdot \log(T_{1/2}^c) + b. \tag{5.1}$$

The preformation factor is usually defined as [2][4]

$$P = \frac{T_{1/2}^c}{T_{1/2}^m}, \tag{5.2}$$

and taking a logarithm on both sides of equation (5.2) it can be written in a more useful form

$$\log(T_{1/2}^m) = \log(T_{1/2}^c) - \log(P). \tag{5.3}$$
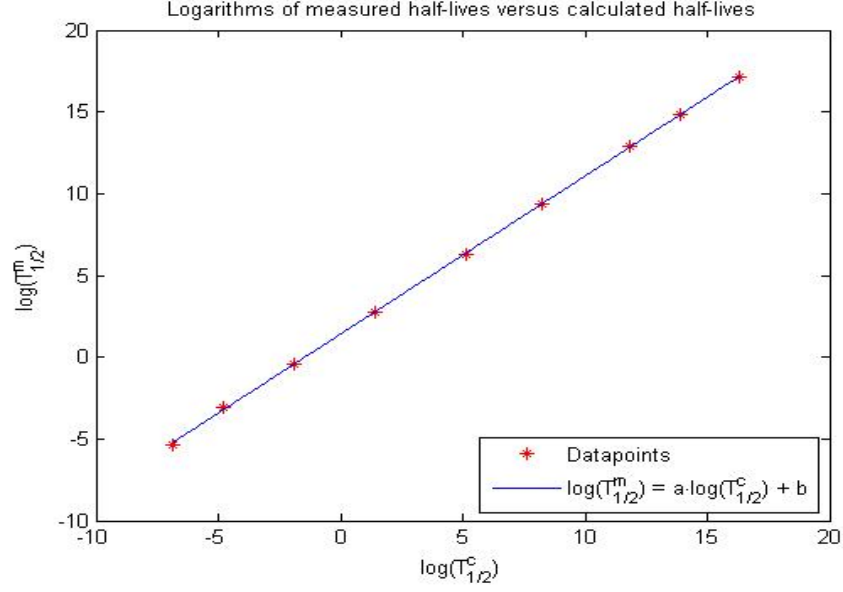
Figure 3: $\log(T_{1/2}^m)$ vs. $\log(T_{1/2}^c)$

Figure 4: Preformation corrected half-lives compared to the measured values.

Solving b from the linear fit in Figure 3 and comparing equations (5.1) and (5.3), the preformation factor can be written as

$$P = 10^{-b}, \tag{5.4}$$

and a value of $P = 0.04$ is obtained. Making the adjustment $T_{1/2}^c \rightarrow P^{-1}T_{1/2}^c$ to the calculated half-lives a clear improvement can be observed, as seen in Figure 4.

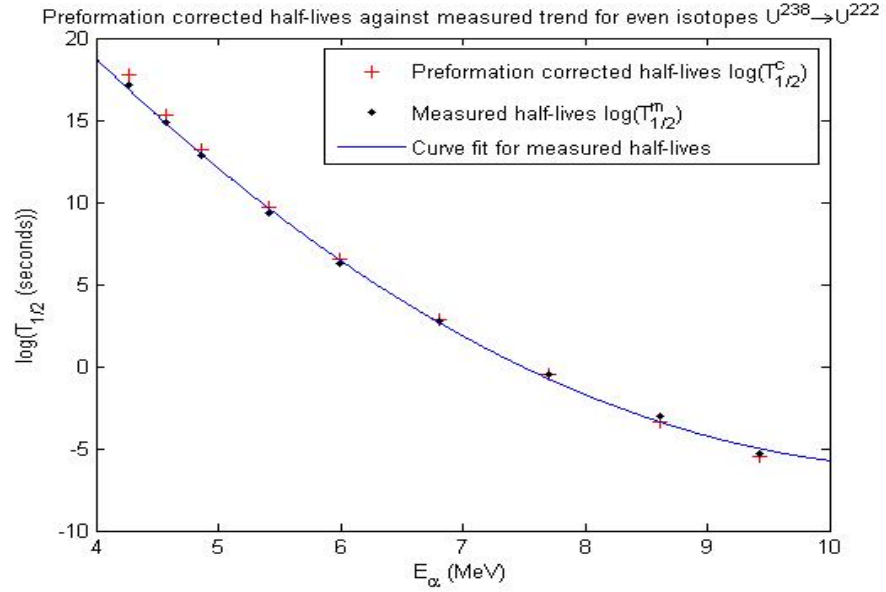| Isotope	$T_{1/2}^m$ s	$T_{1/2,P=0.04}^c$ s	$T_{1/2,GLDM-P}^c$ s	GLDM P-factor
U^{238}	$1.409 \cdot 10^{17}$	$5.302 \cdot 10^{17}$	$8.692 \cdot 10^{16}$	0.24
U^{236}	$7.386 \cdot 10^{14}$	$1.952 \cdot 10^{15}$	$4.409 \cdot 10^{14}$	0.18
U^{234}	$7.742 \cdot 10^{12}$	$1.699 \cdot 10^{13}$	$4.4552 \cdot 10^{12}$	0.15
U^{232}	$2.173 \cdot 10^{9}$	$4.681 \cdot 10^{9}$	$1.342 \cdot 10^{9}$	0.14
U^{230}	$1.797 \cdot 10^{6}$	$3.327 \cdot 10^{6}$	$1.152 \cdot 10^{6}$	0.12
U^{228}	$5.46 \cdot 10^{2}$	$6.713 \cdot 10^{2}$	$2.521 \cdot 10^{2}$	0.11
U^{226}	$3.5 \cdot 10^{-1}$	$3 \cdot 10^{-1}$	$2.186 \cdot 10^{-1}$	0.05
U^{224}	$9 \cdot 10^{-4}$	$4.329 \cdot 10^{-4}$	$3.429 \cdot 10^{-4}$	0.05
U^{222}	$4.7 \cdot 10^{-6}$	$3.293 \cdot 10^{-6}$	-	-

Table 2: Measured half-lives and calculated half-lives with both corrections.

The new calculated half-lives follow the trend and measured values a little closer than the unadjusted values. This can be seen even better when tabulating the values and comparing them as is done in Table 2: instead of having differences of one order of magnitude, now the differences range only from $T_{1/2}^c$ being at most less than 4 times larger and 2 times smaller than $T_{1/2}^m$. Even though here only one value for P was used in calculating the half-lives for all the isotopes under study, the individual preformation values having ranged from about 0.3 to 0.03, and even though 0.04 is not that close to what is obtained with more realistic potentials [4], this evaluation gives a surprisingly good fit.

The validity of using the same P for all isotopes can be checked by comparing these results with half-lives calculated using values for P computed with the generalized liquid drop model GLDM [5]. Varying P seems to lead to a more consistent fit, and the difference between it and the measured fit, even if not necessarily smaller than with constant P, clearly is more consistent in following the measured trend as can be seen from Figure 5.

5.3 Accuracy of the numerical methods

A computer can represent real numbers only by finite approximations, so called floating point numbers. On default MATLAB represents floating point numbers in double-precision; in MATLAB real numbers are rounded up or down to the 16th decimal place. Errors accumulating from preforming repeated arithmetic operations such as the sum in trapezoid method or iterating formula (3.9) repeatedly would be significant if the numbers and step sizes used in these calculations were of the order 10^{-16}. Even though the above calculations took place in the femtometer scale, all quantities containing the unit of length were scaled accordingly in order to avoid dealing with too small numbers and accumulating rounding off errors.
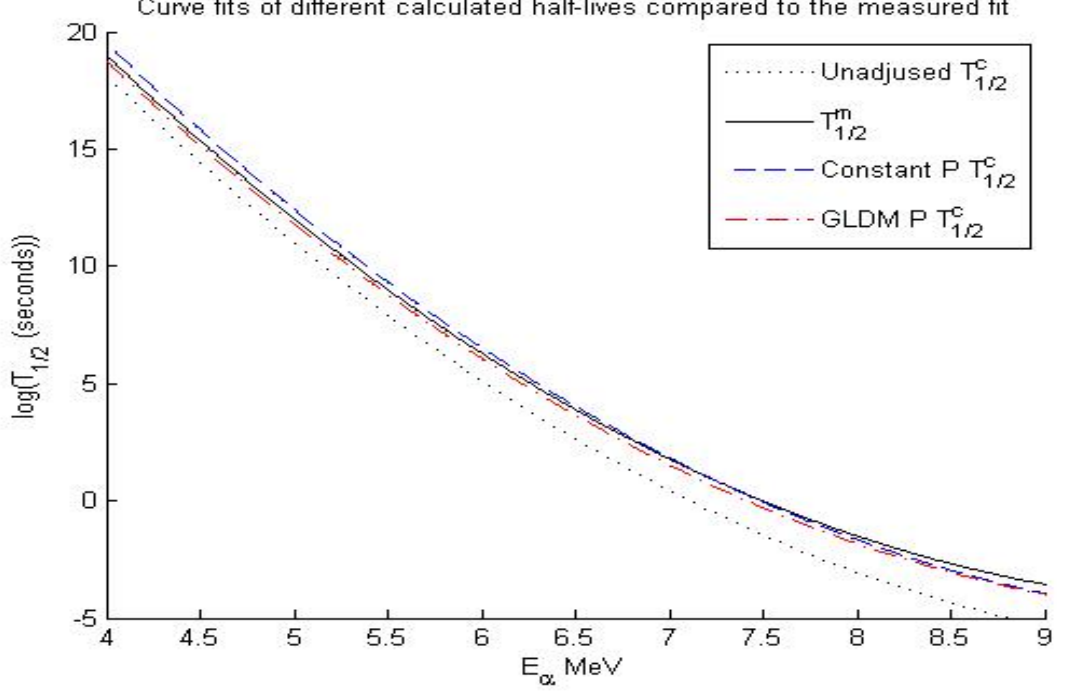
Figure 5: Half-lives calculated with $P = 1$, $P = 0.04$ and varying P plotted against E_α, with measured half-life as a comparison.

Another consequence of the finiteness of computer representation is that usually some approximations have to made. This leads to truncation errors, which can vary depending on the approximation. For the above calculations there were two main sources for truncation error, (3.7) from the finite difference approximation and (3.17) from the trapezoid method of estimating an integral

$$
\begin{aligned}
err_{fdm} &= \frac{\delta r^4}{12} \frac{d^4 u(r)}{dr^4}, \\
err_{tpz} &= -\frac{\delta r^2 (r_2 - r_1)}{12} \frac{d^2 f(\xi)}{dr^2}.
\end{aligned}
$$

Looking first at err_{tpz}, one can note that the derivative of the integrand is proportional to a sum of terms like $[V(r) - E_\alpha]^{-3/2}$, $[V(r) - E_\alpha]^{-1/2}$ and first and second derivatives of $V(r)$ which in turn are proportional to terms such as r^{-3} and r^{-2}. Based on this one can argue that the order of magnitude for the error coming from the trapezoid method is at most $\delta x^2 = 0.0001$. Values of the integrand are mainly well above zero and almost of the order 10^1, so this is not a significant source for error.

The fourth derivative in the error term err_{fdm} contains also terms such as above with even higher derivatives of the integrand $f(r)$ and $V(r)$. Thus it is reasonable to assume that the

14

error coming from the finite difference method is at most of the order $\delta x^4 = 0.000001$. The wave function has values closer to the same range as the integrand in the trapezoid case, so this is not a significant source of error either.

6   Conclusions

Success of the above model in reproducing the qualitative dependency between the α-particle energies and half-lives of the isotopes under study demonstrates again the validity of a quantum mechanical approach into α-decay phenomenon. With a more realistic representation of the potential energy term than in Gamows original work a relatively good quantitative result was also obtained. Inaccuracies stemming from the numerical methods and finite representation of real numbers by a computer was not the main cause for deviations from measured values; simplifications and approximations such as taking the potential to be spherically symmetric thus neglecting important effects of nucleus deformation via the electric quadrupole moment, always important shell effects and not considering angular momentum carried out by the α-particle no doubt are the main reason for reduced accuracy of the calculations.

Even though there does not appear to be any reason to assume that the α-particle preforms or exists in the parent nucleus separately during the decay process, the results of this and more sophisticated models [4][5] seem to suggest that this type of phenomenon actually occurs. A fit for constant P was made from half-lives calculated by the above model and the results were compared to half-lives calculated with varying P, and as a more consistent fit was achieved with the varying P, the result suggests that in nature P depends on the specific isotope and its properties with heavier isotopes having a larger preformation factor.

Numerical techniques demonstrated here, especially the bisection method, were shown to be effective in solving energy levels of an α-particle from the Schrödinger equation. These same techniques can be used with different potential energy models in solving e.g. bounded energy levels of atoms [6]. There exists more accurate methods for numerical integration, e.g. the so called Simpson's rule, but the trapezoid method was deemed accurate enough for the purposes of the above calculations.

7   References

[1] Krane KS. Introductory nuclear physics /. New York :: Wiley,; 1988. Rev. ed. of Introductory nuclear physics / David Halliday. 2nd. ed. 1955.

[2] Duarte D, Siegel PB. A potential model for alpha decay. American Journal of Physics. 2010;78(9):949–953. Available from: `http://scitation.aip.org/content/aapt/journal/ajp/78/9/10.1119/1.3432752`.

[3] National Nuclear Data Center of Brookhaven National Laboratory;. Available from: `http://www.nndc.bnl.gov/`.

[4] Ibrahim TT, Wyngaardt SM. Preformation factor and alpha decay half-lives of superheavy nuclei. Journal of Physics G: Nuclear and Particle Physics. 2014;41(5):055111. Available from: `http://stacks.iop.org/0954-3899/41/i=5/a=055111`.

[5] Zhang HF, Royer G. α particle preformation in heavy nuclei and penetration probability. Phys Rev C. 2008 May;77:054318. Available from: `http://link.aps.org/doi/10.1103/PhysRevC.77.054318`.

[6] Cheng L, Goldstein E, Siegel PB. Nice things about mean fields: Mean field potentials for the classroom. American Journal of Physics. 2000;68(9):848–853. Available from: `http://scitation.aip.org/content/aapt/journal/ajp/68/9/10.1119/1.1302822`.

A   MATLAB m-files

A.1   Vo.m

```
function [V, T, Et] = Vo(Ealpha, A, V0)
% Calculates the Vo term for the Woods-Saxon potential. Also gives out an
% array of all the energy levels including the quasi bound energy level,
% and the alpha particle energy (this so one can check that the calculation
% is correct). This function calls two other functions, written below as is
% done in MATLAB, called Elevel() and schr(); details of these below. This
% code is written for even isotopes of Uranium from U(238) to U(222), so
% incase one wants to use this for other isotopes, modification is
% necessary.
%
% Input arguments:
% Ealpha: Measured energy of the alpha particle
% A: Mass number of the DAUGHTER nucleus
% V0: Some beginnign value for Vo, need to start the search somewhere
%
% Output arguments:
% V: The Vo term calculated below
% T: Array of calculated energy levels
% Et: Energy of the quasi bound state, should be as close as possible to
%     the alpha particle energy
```

```matlab
% Defining variables
l = 0;    % Variable for the while-loop
dV = -1; % Starting value for the change in Vo term
Et = 0;  % Quasi bound energy level from the previous iteration, starting value set to zero
n1 = 10; % Desired quantum number based on previous code runs, cuts down on computing time

while l<1

    % Calculate the energy levels up to the first positive (quasibound)
    % level, the quantum number n and array of all energy levels
    [Ea, n, T] = Elevel(A, V0);

    % If n is too large:
    if n>n1
        V0 = V0+2; % Lift the bottom of the potential well in order to decrease n
    end

    % If n is too small:
    if n<n1
        Et = Ea;
        V0 = V0-2; % Push down the bottom of the potential well in order to increase n
    end

    % If n is exactly what we want:
    if n==n1

        % If  the new and old calculated quasi bound energy levels on the
        % same side of alpha particle energy, change V0 in order to get Ea
        % colser to Ealpha:
        if (Ealpha-Ea)*(Ealpha-Et)>0
            Et = Ea;      % Update Et, the previous calculated (quasi)bound energy level
            V0 = V0+dV; % Update V0

        % Or if the are on different sides, we crossed the desired value
        % for V0. In that case:
        elseif (Ealpha-Ea)*(Ealpha-Et)<0

            % In case the energy Et is smaller than Ealpha, meaning Ea, the
            % energylevel now calculated is slightly above Ealpha:
            if (Et<Ealpha)

                % If we have not reached the desired accuracy:
```

```
                        if abs(dV)>10^(-6)
                            Et = Ea;            % Update Et
                            dV = (dV/2)*(-1); % Update dV
                            V0 = V0+dV;         % And update V0

                        % If we have reached the desired accuracy
                        elseif ((n1==n) && (abs(dV)<10^(-6)))
                            Et = Ea;
                            V = V0;  % Value that reproduces the measured alpha particle energy
                            l = 2;    % Break the while loop

                        end

                    % Or if Et is above Ealpha, meaning Ea is below it then:
                    elseif Et>Ealpha
                        Et = Ea;
                        dV = (dV/2)*(-1);
                        V0 = V0+dV;

                    end
                end

            end
        end
end
%------------------------------------------------------------------------
%------------------------------------------------------------------------
%------------------------------------------------------------------------
% The functions that are calle by the main function Vo().
%------------------------------------------------------------------------
%------------------------------------------------------------------------
%------------------------------------------------------------------------
function r1 = schr(E, A, V0)
% This function iterates the discretized Schrödingers equation
% u[i+1]=2u[i]+2m(d/h)^2(V[i]-E)u[i]-u[i-1], where h is the reduced plancks
% constant, d is length of the iteration step and m is mass of the alpha
% particle, to the distance of 20 fm.
%
% Input arguments:
% E: Energy level of the state
% A: Mass number of the DAUGHTER nucleus
% V0: Vo term of the Woods-Saxon potential
%
```

```
% Output arguments:
% r1: Value of the radial wave function at 20 fm

q = 1.4399764;          % e^2 in units of MeV*fm
a = 0.8;                % Surface "thickness" of the parent nucleus, units fm
w = 1.3*(A^(1/3));      % "Radius" of the daughter nucleus, units fm
X = 1.9145*(10^(-5));   % The constant term "2m(d/h)^2" in units of 1/MeV
r1 = 0;                 % Value of u[i] at the distance of 20 fm, starting value arbitrary
u0 = 0;                 % u[i-1]- term
u1 = 1;                 % u[i]- term
r = 0.01:0.01:20;       % Vector of iteration steps
Z = 92;                 % Proton number of the parent nucleus (Uranium)


for i = 1:length(r)

    % First, calculate the value of the potential energy terms
    V_em1 = 2*(Z-2)*q*(3*(w^2)-(r(i)^2))/(2*(w^3)); % r<=w
    V_em2 = 2*(Z-2)*q/r(i);                         % r>w
    x = (r(i)-w)/a;
    V_ws = V0/(exp(x)+1);

    % Calculate u2=u[i+1]
    if r(i)<=w
        u2 = 2*u1+X*(V_em1+V_ws-E)*u1-u0;

    elseif r(i)>w
        u2 = 2*u1+X*(V_em2+V_ws-E)*u1-u0;

    end
    % Update variables for next iteration step
    u0 = u1;       % u[i-1] --> u[i]
    u1 = u2;       % u[i] ---> u[i+1]

    % If we are done, u[i+1]=u2==r1
    if r(i)==r(end)
        r1 = u2; % u[i+1] --> u(r1)

    end

end
end
%-----------------------------------------------------------------------------
%-----------------------------------------------------------------------------
```

```
function [Ea, n, T] = Elevel(A, V0)
% Calculates the first positive (quasibound) energy level. All the energy
% levels are calculated by means of the bisection methotd described in the
% thesis paper. Iteration of the Schrödinger equation is done by calling
% the function schr() defined above.
%
% Input arguments:
% A: Mass number of the DAUGHTER nucleus
% V0: A value for the Vo term in Woods-Saxon potential
%
% Output arguments:
% Ea: The first positive (quasibound) energy level
% n: Radial quantum number
% T: Vector of all energy levels

z = 0;          % Parameter for the while- loop
r0 = 0;         % Result of the previous run of schr(), starting value arbitrary
dE = 0.5;       % Staring value for the energy increment
E = V0+1;       % Energy level in the beginning of the iteration
E1 = V0+1;      % Last bound energy level calculated, starting value arbitrary
n = 0;          % Starting value for the radial quantum number
T = zeros(1);   % Will be a vector of energy levels

while z<1

    % Calculate r1 from schr()
    r1 = schr(E, A, V0);

    % If the last energy level calculated is negative:
    if E1<0
        % And if r0 is 0 or r0 and r1 have the same sign
        if ((r0==0) || (r0*r1>0))
            r0 = r1;  % Update r0
            E = E+dE; % Update E


        % or if ro and r1 have different signs, but desired accuracy has
        % not been acheived
        elseif ((r0*r1<0) && (abs(dE)>10^(-6)))
                r0 = r1;         % Update r0
                dE = (dE/2)*(-1); % Update dE
                E = E+dE;         % Update E
```

```
            % or if r0 and r1 have different signs and the desired accuracy has
            % been reached
            elseif ((r0*r1<0) && (abs(dE)<10^(-6)))
                    n = n+1;     % Update the radial quantum number
                    T(1,n) = E; % Update the array for energy levels
                    E1 = E;      % Update the value for most recent found bound state
                    % Reset variables for next round
                    r0 = 0;
                    E = E+1;
                    dE = 0.5;

            end

        end

        % If the most recent found energy level was positive, then we are
        % finished
        if E1>0
            Ea = T(end); % Output Ea gets the value of last element in the array T
            z=2;          % update  the while-loop parameter in order to break the loop

        end

end
end
%-------------------------------------------------------------------------
```

A.2   radii.m

```
function [r1, r2, V, X] = radii(Ealpha, A, Vo)
% Determines the ends of the tunnel through which the alpha particle
% penetrates the potential barrier. Also gives the values of the potential
% energy from the 0.01 fm up to 100 fm and the tunneling interval. The ends
% of the tunnel satisfy the relation Ealpha=V(r1)=V(r2).
%
% Input arguments:
% Ealpha: Alpha particle energy
% A: Mass number of the DAUGHTER nucleus
% Vo: Vo term of the Woods-Saxon potential
%
% Output arguments:
% r1: Beginning of the tunnel
% r2: End of the tunnel
```

```matlab
% V: Array of values of the potential in the tunneling interval
% X: Array of the values the potential energy gets in the interval
%    0.01->100 fm

q = 1.4399764;          % e^2 in units MeV*fm
a = 0.8;                % Surface "thickness" of the nucleus
w = 1.3*(A^(1/3));      % Radius of the daughter nucleus
r = 0.01:0.01:100;      % Array of the discrete steps to which 100 fm are divided
Z = 92;                 % Proton number of the parent nucleus (Uranium)
X = zeros(1,length(r)); % Values of the potential energy between the interval 0.01->100 fm

% Calculate the value of the potenrial between 0.01->100 fm
for i = 1:length(r)

    V_em1 = 2*(Z-2)*q*(3*(w^2)-(r(i)^2))/(2*(w^3)); % r<=w
    V_em2 = 2*(Z-2)*q/r(i);                         % r>w
    x = (r(i)-w)/a;
    V_ws = Vo/(exp(x)+1);

    % Calculating the value depending if r<=w or r>w
    if r(i)<=w
        v = V_em1+V_ws;
    elseif r(i)>w
        v = V_em2+V_ws;
    end

    X(1,i) = v; % Assigning the value of the potential to the array X

end

% Declairing these variables now, avoiding a weird error....
r1 = 0;
r2 = 0;
I1 = 0;
I2 = 0;

% Determining the values of r for which the relation Ealpha=V(r1)=V(r2)
% holds as closely as possible
for j = 1:length(X)-1

    % Follow 2 consecutive values of the potenital energy
    x1 = X(j);
    x2 = X(j+1);
```

```matlab
        % If Ealpha is either of these values of falls in between for V(r1)
        if ((x1<=Ealpha) && (x2>=Ealpha))
            % We save the position and value of x2, since there may be many
            % values of x1 and x2 this might apply; we take the largest value
            r1 = r(j+1);
            I1 = j+1;

        % If Ealpha is eithr x1 or x2 or falls in between for V(r2)
        elseif ((x1>=Ealpha) && (x2<=Ealpha))
            % We save the position and value of x1; reason above
            r2 = r(j);
            I2 = j;
        end
    end


% Create the array for values of the potential in the tunneling interval
V = X(1,I1:I2);


end
```

A.3   sigma.m

```matlab
function [S, vo, r1, r2, X, T, Et]  = sigma(Ealpha, A)
% Calculates the WKB- factor, i.e the tunneling integral. This function
% calls the function Vo() in order to determine the Vo term for the
% Woods-Saxon potential needed in the function radii(), which is also
% called in order to determine the values of the potential energy in the
% interval of the tunnel, and the values r1, and r2 for the ends of the
% tunnel.
%
% Input arguments
% Ealpha: The alpha particle energy
% A: Mass number of the DAUGHTER nucleus
%
% Output arguments:
% S: The WKB- factor
% vo: The Vo term of the Woods-Saxon potential
% r1: Beginning of the tunnel
% r2: End of the tunnel
% X: Array of the values of the potential energy has between 0.01->100 fm
% T: Array of the energy levels, bound and the first quasibound
% Et: Energy of the first positive (quasibound) energy level
```

```
H = 1/(6.582119514*10^(-22)); % 1 over the reduced Planck constant, units (MeV*s)^-1
c = 299792458*10^(15);        % Speed of light in the units fm/s
M = (3.72737924*10^3)/(c^2);  % Mass of the alpha particle in units MeVs^2/(m^2)


% Determine the Vo term of Woods-Saxon potetnial by calling Vo() with the
% beginning value of -100 MeV for Vo
V0 = -100;
[vo, T, Et] = Vo(Ealpha, A, V0);


% Determine the values of V(r) in the interval between the tunnel, and the
% ends of the tunnel r1 and r2
[r1, r2, V, X] = radii(Ealpha, A, vo);


% Calculate the WKB- factor by numerical integration using trapetzoidal
% method
x = sqrt(2*M*abs(V-Ealpha)); % The integrand
z = (r1:0.01:r2);            % The integration interval
S = H*trapz(z,x);            % WKB- factor


end
```

A.4   lambda.m

```
function [L, vo, r1, r2, V, T, Et, S] = lambda(Ealpha, A, P)
% Calculates the decay constant from the formula lambda = P*fo*exp(-2S),
% where P is the preformation factor (probability that an alpha particle
% exists at a certain time), fo is the frequency with which the alpha
% particle hits the potential barrier and S is the WKB- factor. fo is
% evaluated as fo = sqrt(2*e/m)/(2*r1), where e = Ealpha-(Vo+V_em(r=0)) is
% an approximation of the kinetic energy of the alpha particle in the
% nucleus and m is the alpha particles' mass.
%
% Input arguments:
% Ealpha: Energy of the alpha particle
% A: Mass number of the DAUGHTER nucleus
% P: Preformation factor
%
% Output arguments:
% L: Decay constant
% vo: Vo term of the Woods-Saxon potential
% r1: Beginning of the tunnel
% r2: End of the tunnel
```

```
% V: Array of values for V(r) from 0.01->100 fm
% T: Array of the energy levels, bound and first quasibound
% Et: First quasibound energylevel
% S: WKB- factor

tic
Z = 92; % Proton number of the parent nucleus (Uranium)

% Calculate the WKB- factor by calling roo()
[S, vo, r1, r2, V, T, Et] = sigma(Ealpha, A);

V_em0 = 3*(Z-2)*1.4399764*(1.3*(A^(1/3))^(-1)); % Coulomb potential at r=0, units MeV
e = Ealpha-(vo+V_em0);                  % Kinetic energy of the alpha particle, units MeV
c = 299792458*10^(15);                  % Speed of light in units fm/s
m = (3.72737924*10^3)/(c^2);            % Mass of the alpha particle, units MeVs^2/(m^2)
fo = sqrt(2*e/m)/(2*r1); % Frequency by which the alpha particle hits the potential barrier

L = P*fo*exp(-2*S); % Decay constant
toc
end
```