UNIVERSITY OF HELSINKI

Numerical calculations of nuclear stopping power of an ion

M. De Meulder
University of Helsinki, Institute of Physics, Helsinki, Finland
(Dated: May 5, 2022)

Nuclear stopping power of ions were studied using numerical methods and the results were compared to values from universal stopping power formula. Numerical methods were used for root solving and numerical integration. Brent's method was used to calculate the root, when solving the minimum approach distance r_{min}, which itself is a combination of inverse quadratic interpolation, secant and bisection methods. Polynomial integration method of Gauss-Legendre quadrature was used in numerical integration, due to it's relatively simplicity. The calculation program was divided into four parts for organising purposes. The program consist of math file, where the cumbersome maths is done, file which contains the equations for numerical result of the stopping power, file which contains the equations for the universal stopping power formula and main which combines the former three file to apply the functions for representable results. The results from both numerical stopping power and universal formula were almost identical. The results showed that higher energies doesn't always mean higher stopping power, but a unique energy can be found where the stopping power is the greatest. This final project showed that numerical methods can be applied to complex physical problems and accurate results can be obtained and the results can be applied in the future.

I. INTRODUCTION

Ever since the reveal of atoms nuclear structure in early 20th century, the field of solid matter physics and structures has been a big field among physicists. A lot of different ways to study the material structure has been invented since then. From the studies we know that usually the solid materials are structured in some sort of periodic lattice, crystals structures. These crystal structures are rarely perfect and contains defects. Knowing how these different kind of defects affects the material is important. [7, 9, 11]

When building nuclear reactors (fission and fusion) it is important to know how the high energy ions affects the reactor walls, when they penetrate the material. The ion energies are quite high in these cases, but also lower energy ions penetration studied and even applied when making semiconductors.

In order to study the defects, we need to reproduce these defects accurately and in a controllable fashion. Ion implementation is a popular way to create defects because its good control of the implementation dose and depth.[4]

It is quite cumbersome to study experimentally all the different ion implementations with different materials and energies, so simulations are quite handy. With simulations we don't need to damage the material being studied. Molecular dynamics simulations are a common tool to study microscopic scale events, but the large amount of atoms prevents the use of MD simulations to some extend when studying the stopping power quantitatively, due to computation limits. In this final project we study how to calculate numerically the stopping power of ions and compare it to the universal stopping power formula. This is possible because the interactions between the atoms has been calculated before with *ab initio* methods or by some other means. [8]

II. METHODS

A. Brent's method

In order to calculate the stopping power, we need to know the minimum approach distance of the two atoms. This is calculated by solving the root from:

$$g(r_{min}) = \sqrt{1 - \left(\frac{b}{r_{min}}\right)^2 - \frac{V(r_{min})}{E_{com}}} = 0, \quad (1)$$

where b is the collision parameter, $V(r_{min})$ is the universal screened Coulomb potential and E_{com} is the ion's initial energy in center-of-mass coordinates. [10] The root of the function is calculated from $g(r_{min})^2 = 0$ with Brent's method. [5]

Brent's method is a hybrid root-finding algorithm which combines Secant method, Bisection method and inverse quadratic interpolation in order to obtain fast but reliable converging solutions. It uses the second degree Lagrange interpolating polynomial. In principle the Brent's method first tries to interpolate with parabola. If that doesn't work, it switches to Secant method and if the Secant method is ill behaving, it uses Bisection method, which is guaranteed to converge.

The Brent's method starts with a iteration of Secant method or Bisection method if the formed method is ill behaving. The interval $a < b$, must contain the root so $sign(f(a)) \neq sign(f(b))$. T values c and midpoint $m =$

$\frac{a+b}{2}$ are obtained from the first iteration. The secant intercept s is calculated using b and c and if dividing by zero occurs, the secant interception is $s = m$.

After the first iteration a Lagrange polynomial is used to fit a inverse quadratic function to values a,b and c, where $y = 0$, in order to find the x-intercept of $L_2(y)$. When the x-intercept is found, the new Lagrange polynomial ($L_2(y)$) is computed to find the new intercept and this is repeated until $|f(x_n)| < \epsilon$. The simplified polynomial is shown in equation 2.

$$\begin{aligned} x_{n+1} = x_{n-2}\frac{y_{n-1}y_n}{[y_{n-2}-y_{n-1}][y_{n-2}-y_n]} \\ + x_{n-1}\frac{y_{n-2}y_n}{[y_{n-1}-y_{n-2}][y_{n-1}-y_n]} \\ + x_n\frac{y_{n-2}y_{n-1}}{[y_n-y_{n-2}][y_n-y_{n-1}]}. \end{aligned} \tag{2}$$

This method is though not very reliable. It fails if two y values matches and it has to start quite close to the solution in order to keep the iterations low. If the inverse quadrature doesn't work it tries to compute the secant intercept as mentioned earlier. If the secant method fails, the values of a, b and c are updated with Bisection method and s is tested between b and $\frac{3a+b}{4}$. This process is repeated until $|b-a| < \epsilon$. [6] Being quite trivial, the Secant method and Bisection method will not be covered here.

B. Gauss-Legendre quadrature

Integration is needed in two equations when calculating the stopping power: Calculating the scattering angle Θ and the stopping power S_n itself. These integrals were calculated numerically with Gauss-Legendre quadrature. It is a numerical approximation of a definite integral defined as

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \sum_{i=1}^{N} w_i f(x_i). \tag{3}$$

The points x_i are zeros of the Legendre polynomial of order N. The calculation of weights uses Legendre polynomials

$$P_N(x_i) = 0 \tag{4}$$

where x_i lies between

$$-1 < x_1 < x_2 < ... < x_N < 1. \tag{5}$$

The weights w_i are calculated using the points x_i and using first derivative of the Legendre polynomial:

$$w_i = \frac{2}{(1-x_i^2)(P'_N(x_i))^2} \tag{6}$$

Our integration limits for θ and S_n are not from -1 to 1. Scattering angle Θ is integrated from 0 to 1 and stopping power S_n is integrated from 0 to b_{max} after the change of variables. So in order to use the Gauss-Legendre quadrature, a change of interval has to be performed. Changing the interval from a to b is done as follows:

$$\int_a^b f(x)\mathrm{d}x = \int_{-1}^{1} f\Big(\frac{b-a}{2}u + \frac{a+b}{2}\Big)\frac{b-a}{2}\mathrm{d}u. \tag{7}$$

So when applying the Gauss-Legendre quadrature, we get

$$\int_a^b f(x)\mathrm{d}x \approx \frac{b-a}{2}\sum_{i=1}^{n} w_i f\Big(\frac{b-a}{2}u_i + \frac{a+b}{2}\Big). \tag{8}$$

This equation 8 is implemented in the code in order to calculate the integrals of scattering angle Θ and the stopping power S_n.

III. IMPLEMENTATION

The program is simply run with command `python3 main.py` from `\src`. Short description and instructions is found from README.

A. Program structure

The program is divided into four different files in order to keep the program organised. The equations used to numerically calculate the nuclear stopping power are organized in file `nuclear_stopping_equations.py`. Equations used to calculate the universal nuclear stopping power are organized in file `nuclear_stopping_formula.py`. The more cumbersome math used is located in file `math_library.py`. This file is in a sense quite short, but it helps to organize the program and it is easy to extend if other integration, root finding methods or some other methods are in interest to be used. These files are then used in the `main.py` file where the plotting and commands to calculate are executed. Files `nuclear_stopping_formula` and

`nuclear_stopping_equations` are imported in main as `nsf` and `nse`.

The weights w_i and points x_i used in the Gauss-Legendre quadrature 8 are saved in file `weights100.txt` and it is located in `\run`. It contains the first 100 points and weights for the quadrature. The weights and points are read into memory when the `math_library` is imported to save computation time. This way the weights and points are read only once and not every time the integration function is called.

B. Main

The constants and units used in the program are shown in table I. The used order of magnitude for energy in the program is [keV] and some conversion coefficients where used to convert the values to correct units.

Atom	Z	M
H	1	1.007825u
Si	14	27.976927u
Au	79	196.966543u

Table I: Used atomic numbers and masses.

The `screened_Coulomb()` functions in `nuclear_stopping_equations` uses coefficient 9 in order to convert the dimensions to [keV Å].

$$\text{coeff} = 0.0143996 \tag{9}$$

The result of function `Sn()` in `nuclear_stopping_equations` is also multiplied by 10^{-13} to convert [keV Å^2] to [eV cm^2] in order to get the same order of magnitude as from the universal stopping power formula.

When the program is run, it first calculates the potential curves between $r = 0.0001 - 10$ Å. This is performed in order to obtain a value for b_{max}. As result the chosen value was $b_{max} = 6$ Å. That's when the potential curves of H-Si and Si-Au crosses (figure 2), so the results may be more comparable and it is far enough for both potentials to be a big enough collision parameter.

After this a test to compare the computing time and accuracy was done to choose a good value for number of nodes in Gauss-Legendre polynomials. It measures the L_∞ between the numerically calculated stopping power and the stopping power from the universal stopping power formula. After about 40 nodes, the accuracy doesn't increase significantly anymore as seen from figure 1, but I chose 100 nodes still because the computation is performed only once per case. This part takes

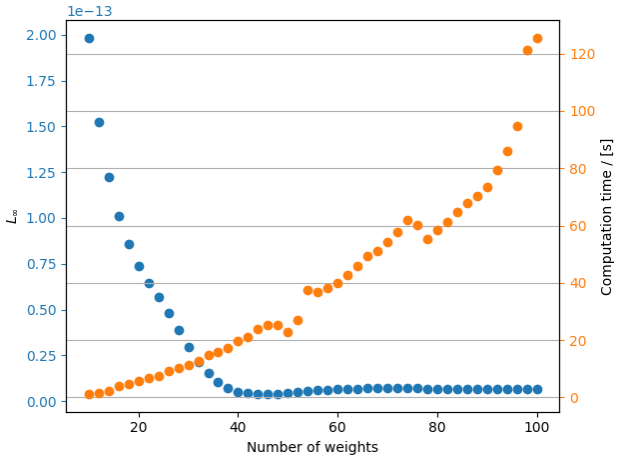


Figure 1: Comparing "accuracy" against computing time.

about 30 minutes to compute when computed the accuracy of even numbers between 10-100. The parameter n in the function is set now set to 20 so it calculates the accuracy with [10,30,50,70,90] nodes.

C. Integration and root finding parameters

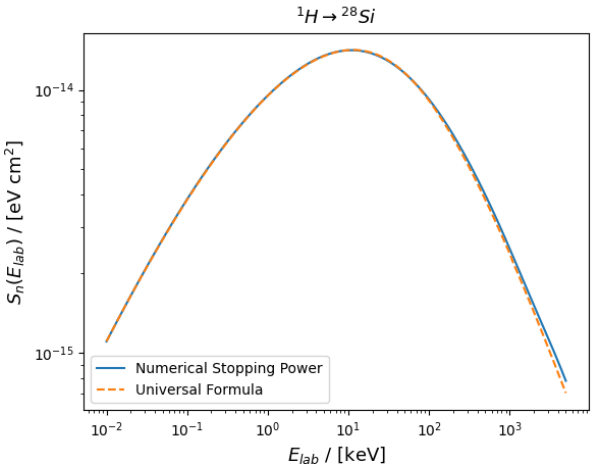
The only 3rd party libraries used in computation was `scipy`. Root solving used function `scipy.optimize.brentq` and integration used function `scipy.special.roots_legenrde`. Other libraries `numpy`, `matplotlib` and `time`, were used to some basic algebra, arrays, plotting and measuring the computing time.

The `brentq` function takes as arguments the function, left range and right range which were set to $a = 0.0001$ and $b = r = 10$. The tolerance and max iterations were left untouched and their default values are `xtol=2e-12`, `rtol=8.881784197001252e-16` and `maxiter=100`. [2]

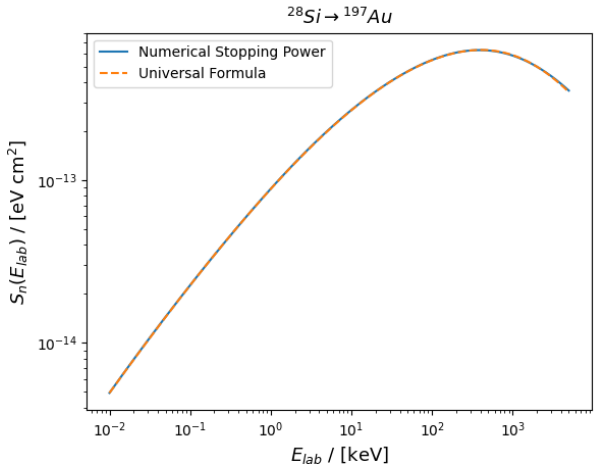
In integration `roots_legendre` function was used to calculate the Legendre polynomial weights and points (w_i,x_i). If the number of nodes is set to 100, the weights and points are read from the file `weights100.txt`, which has the values of the weights and points up to 22 digits. [1, 3]

IV. RESULTS

As mentioned in section III, the screened Coulomb potential was calculated between $r = 0.0001 - 10$ Å in order to deduce a large enough collision parameter b. From figure 2 we can see that the potential overlap at ≈ 6 Å. This value was the chosen to be the collision pa-



(a)



(b)

rameter b. As mentioned also in section III, 100 nodes was chosen to use in Gauss-Legendre quadrature, even though nodes over 40 would have been enough according to plot 1. These values were then used to numerically calculate the nuclear stopping powers which are shown in figure (a) and (b). The calculated stopping power is plotted against the stopping power obtained from the universal stopping formula and the values doesn't differ significantly at all. From figure 1 we see that with 100 nodes the L_∞ between these two is in order of magnitude of 10^{-14}.

The stopping power for hydrogen in silicon seems to have its peak around 10 keV. Si has much bigger stopping power in Au and it start to decrease around 1000 keV. From these results we can calculate the energy needed to implement ions into defined depths as discussed in section I.

Although the accuracy between the models is very good, some divergence is observed in higher energies in figure (a). The divergence is clearer when the stopping power is calculated with fewer nodes. The stopping power was calculated using 30 nodes in Gauss-Legendre quadrature in figure 3 and it is clearly seen that with smaller energies, the results are more accurate than with higher energies. I'm not sure why this affects like this, but it has something to do with the E_{lab} energy. In universal stopping power formula $s_n(\epsilon)$ is calculated differently depending if $\epsilon \leq 30$ or $\epsilon > 30$. The grey dashed line in figure 3 shows, when $\epsilon > 30$. With the stopping power of Si-Au, ϵ never exceeds 30 and it didn't show any signs of divergence at higher energies. So this is not a proof of anything, but something that is could affect the results in high enough energies.

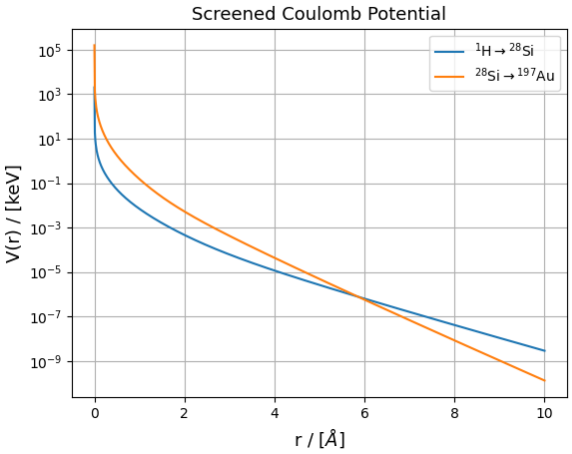


Figure 2: Screened Coulomb potential of H-Si and Si-Au.

V. CONCLUSIONS

The properties of atoms nucleus has been used ever since the discovery of atoms nucleus structure. It has been used to study material structures, enhance material properties, particle colliders etc. Important application for the nucleus properties has been the ion implementation in semiconductors. In order to get the ions deployed reliable to correct depths and correct amounts, the behavior of materials stopping power is crucial.

This final project calculated numerically the nuclear stopping power of H-Si and Si-Au. Numerical algorithms of Brent's method and Gauss-Legendre quadrature were used to calculate the roots and integrals. The results were identical compared to the results calculated

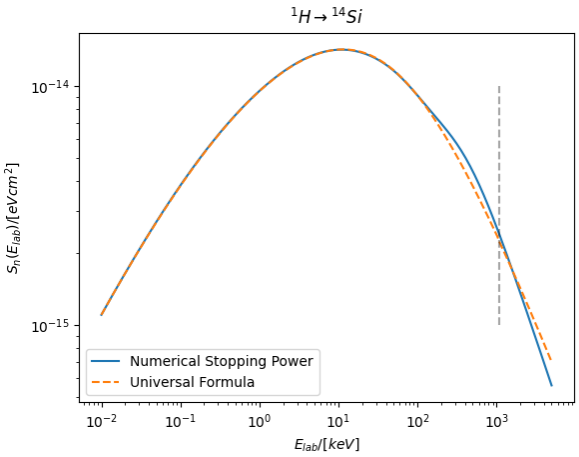


Figure 3: Stopping power calculated with 30 nodes. The grey lines shows when $\epsilon > 30$.

with the universal nuclear stopping power formula. In case of H-Si the numerically calculated values started to diverge a bit compared to the universal formula with higher energies, but stopping power of Si-Au didn't suffer from this problem. The effect was clearer when less nodes were used integration.

Though the results were relatively accurate, the accuracy could be enhanced by using double precision floats. Now only normal 24 bits float were used. By using double precision, the weights and points saved in `weights100.txt` could have had a bigger impact because they are saved up to 22 digits. This is still too much for double precision (16 digits), but it would have had some impact already. Bigger problem than the accuracy would be the computing time. Using 100 nodes in the Gauss-Legendre quadrature results in a computation time of ≈ 70 seconds to obtain plot of figure (a) or (b). This was reduced by about 10 seconds, because with 100 nodes the weights and points are read from the file `weights100.txt`. To improve the computation time further, the weights and points could be read from the file when using less nodes also. As implied before, the number of nodes in integration can be reduced to around 40 nodes without significant accuracy loss. Because the `scipy` library, what's used to make the heaviest calculations, is already quite well optimized, I don't think that other very impactful optimizations can be performed.

From this final project it can be deduced that numerical methods are well suited to calculate more complex problems in physics and obtain accurate results to apply them is future.

VI. ACKNOWLEDGEMENTS

Thank for the course lecturers to organise an interesting course with various compelling topics. Thanks for the TA's to help us through the exercise problems and answering to our endless questions. Huge thanks also for the other students on the course. Struggling and learning with you through the course has been fun and without your peer support and help, the course would have been so much harder.

[1] Nodes and weights of gauss-legendre calculatorl. url. Accessed: 2021-20-05.
[2] Scipy brentq. url. Accessed: 2021-20-05.
[3] Scipy legendre polynomial. url. Accessed: 2021-20-05.
[4] Jim Asher. Mev ion processing applications for industry. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 89(1):315–321, 1994.
[5] Richard P. Brent. *Algorithms for minimization without derivatives*. Dover Publications, Inc., Mineola, NY, 2002. Reprint of the 1973 original [Prentice-Hall, Inc., Englewood Cliffs, NJ; MR0339493 (49 #4251)].
[6] R.P. Brent. *Algorithms for minimization without derivatives*. Prentice-Hall, 1973.
[7] P. J. Brown and J. Bruce. Forsyth. *The crystal structure of solids / P.J. Brown, J.B. Forsyth*. Edward Arnold London, 1973.
[8] R. Car and M. Parrinello. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.*, 55:2471–2474, Nov 1985.
[9] E. N. da C. Andrade and K. Lonsdale. William henry bragg. 1862-1942. *Obituary Notices of Fellows of the Royal Society*, 4(12):277–300, 1943.
[10] J. F. Ziegler, J. P. Biersack, and M. D. Ziegler. the stopping and range of ions in matter.
[11] A. R. Lang. *X-ray methods in the study of defects in single crystals* edited by J. Avleytner. *Acta Crystallographica*, 23(1):183–184, Jul 1967.