FYS4150 Project 3:
Numerical integration

Marie Foss, Maria Hammerstrøm

Abstract

We study the precision and efficiency of four different integration methods for a 6-dimensional integral with a known analytical solution: Gaussian quadrature using Legendre and Laguerre polynomials, and Monte Carlo integration with and without importance sampling. The Monte Carlo method *with* importance sampling proved to be the superior method, both when it came to precision and speed.

1 Introduction

In this project we will determine the ground state correlation energy between two electrons in an helium atom by calculating a six-dimensional integral that appears in many quantum mechanical applications. The methods we will use are Gauss-Legendre and Gauss-Laguerre quadrature and Monte-Carlo integration.

We assume that the wave function of each electron can be modeled like the single-particle wave function of an electron in the hydrogen atom. The single-particle wave function for an electron i in the $1s$ state is given in terms of a dimensionless variable (the wave function is not properly normalized)

$$\mathbf{r}_i = x_i\mathbf{e}_x + y_i\mathbf{e}_y + z_i\mathbf{e}_z,$$

as

$$\psi_{1s}(\mathbf{r}_i) = e^{-\alpha r_i},$$

where α is a parameter and

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}.$$

We will fix $\alpha = 2$, which should correspond to the charge of the helium atom $Z = 2$.

The ansatz for the wave function for two electrons is then given by the product of two so-called $1s$ wave functions as

$$\Psi(\mathbf{r}_1, \mathbf{r}_2) = e^{-\alpha(r_1+r_2)}.$$

The integral we need to solve is the quantum mechanical expectation value of the correlation energy between two electrons which repel each other via the classical Coulomb interaction, namely

$$\langle\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\rangle = \int_{-\infty}^{\infty} \mathrm{d}\mathbf{r}_1\mathrm{d}\mathbf{r}_2 e^{-2\alpha(r_1+r_2)} \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}. \quad (1)$$

This integral can be solved in closed form, which gives an answer of $5\pi^2/16^2 \approx 0.192765710958777$.

2 Methods

2.1 Gaussian quadrature

Gaussian quadrature is a method that solves integrals with excellent results, giving high precision for few integration points compared to simpler integration methods such as Newton-Cotes quadrature. The basic idea behind the method is to approximate the given function by a polynomial of degree $2N - 1$ so that

$$f(x) \approx P_{2N-1}(x),$$

where $P_{2N-1}(x)$ is a polynomial of degree $2N - 1$ with only N mesh points x. Gaussian quadrature says that we can calculate our integral in the following way:

$$I = \int_a^b f(x)\mathrm{d}x = \int_a^b W(x)g(x)\mathrm{d}x \approx \sum_{i=0}^{N-1} \omega_i g(x_i), \quad (2)$$

where ω_i are the weights determined through the use of orthogonal polynomials, $W(x)$ is the weight function and x_i are the chosen mesh points. x_i will then be the zeros of these polynomials.

In this project we make use of Legendre and Laguerre polynomials to solve the integral in Eq. (1). To know which polynomial to use we need to look at the weight function and interval for these polynomials to see which polynomial resembles our integral the most. For Legendre and Laguerre the weight functions and intervals are:

Polynomial	Interval	Weight function
Legendre	$x \in [-1, 1]$	$W(x) = 1$
Laguerre	$0 \le x \le \infty$	$W(x) = x^\alpha e^{-x}$

We will solve the integral in both linear and spherical coordinates, which means we will make use of both of these polynomials.

2.1.1 Gauss-Legendre quadrature

First we want to compute the six-dimensional integral in Eq. (1) with respect to the coordinates $x_1, x_2, y_1, y_2, z_1, z_2$ with limits $[-\infty, \infty]$. The limits will be adjusted to finite limits when doing the numerical calculation, as computers cannot deal with infinite. Because the weight function of the Legendre polynomial is $W(x) = 1$, we can use our integral as is, without any rewriting.

To find out how we can substitute the infinite limits with finite limits without losing too much precision, we plot the single-particle wave function $e^{-\alpha r_i}$ and inspect when it reaches a value close to zero. This occurs at approximately $r_i \approx 3$ as shown in Fig. 1. We therefore used the limits [-3,3].
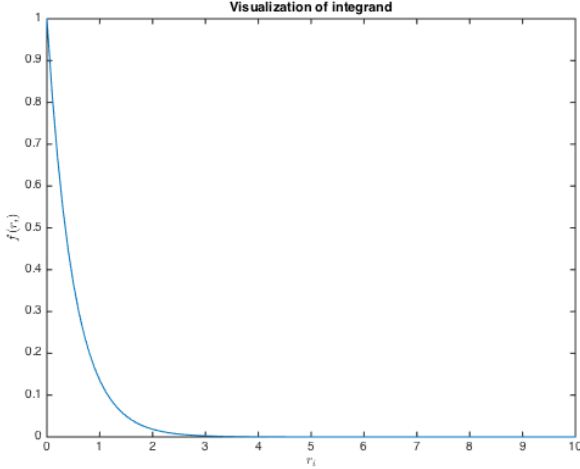

Figure 1: The single-particle wave function $e^{-\alpha r_i}$.

The function `gauleg` to calculate the weights and mesh points using Legendre polynomials has been provided. All we have to do is to call this function and go through a 6-dimensional nested `for`-loop to calculate the sum which approximates the integral.

2.1.2 Gauss-Laguerre quadrature

In using Legendre polynomials our originally infinite limits had to be approximated by finite ones. Our integral in Eq. (1) can be rewritten with spherical coordinates to better deal with the infinite limits. Then the integral in becomes

$$I = \int r_1^2 r_2^2 \frac{e^{-2\alpha(r_1+r_2)}}{r_{12}} \sin\theta_1 \sin\theta_2 dr_1 dr_2 d\theta_1 d\theta_2 d\phi_1 d\phi_2, \quad (3)$$

where $r_i \in [0, \infty]$, $\theta_i \in [0, \pi]$ and $\phi_i \in [0, 2\pi]$, with

$$\frac{1}{r_{12}} = \frac{1}{\sqrt{r_1^2 + r_2^2 - 2r_1 r_2 \cos(\beta)}}, \quad (4)$$

where

$$\cos(\beta) = \cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2)\cos(\phi_1 - \phi_2)).$$

The angular parts of the integral has finite limits and can be calculated using Legendre polynomials without having to have different limits in the analytical and numerical case. Great! The radial part of the integral however has limits $[0,\infty)$, just as the Laguerre polynomials has. Thus we can use Laguerre polynomials with the weight function $W(x) = r^\alpha e^{-r}$ to calculate the radial parts of the integral. But first we need to do a variable change so that our integral is on the same form as the weight function.

We substitute $r_i = u_i/2\alpha$ and $dr_i = du_i/2\alpha$. In our case $\alpha = 2$, and the integral becomes

$$I = \frac{1}{1024} \int_0^\pi \sin\theta_1 \sin\theta_2 d\theta_1 d\theta_2 \int_0^{2\pi} d\phi_1 d\phi_2 \int_0^\infty du_1 du_2 u_1^2 u_2^2 \frac{e^{-(u_1+u_2)}}{r_{12}}.$$

Locating the part of the integral that represents the weight function $W(x)$ we see that the integral becomes

$$I = \frac{1}{1024} \int_0^\pi \sin\theta_1 \sin\theta_2 d\theta_1 d\theta_2 \int_0^{2\pi} d\phi_1 d\phi_2 \int_0^\infty du_1 du_2 \frac{W(u_1)W(u_2)}{r_{12}}.$$

Thus

$$g(u_1, u_2, \theta_1, \theta_2, \phi_1, \phi_2) = \frac{\sin\theta_1 \sin\theta_2}{r_{12}}. \quad (5)$$

The function `gauss_laguerre` to calculate the weights and mesh points of the radial parts of the integral has been provided. Again all we have to do is to call `gauss_laguerre` for the radial part, and `gauleg` for the angular parts, and go through a 6-dimensional nested `for`-loop to calculate the sum which approximates the integral.

2.2 Monte Carlo

The Monte Carlo method is a statistical simulation method which can be used for systems that are described by their probability distribution functions (PDFs). The Monte Carlo method proceeds by random samplings from the PDF. The final result is taken as an average over the number of simulations, and multiplied with the Jacobi determinant of the change of variables.

$$r_{i,j} = -l + 2lx_{i,j}; \ i = 1, 2; \ j = 1, 2, 3. \quad (6)$$

$x_{i,j}$ is a uniformly distributed random number between 0 and 1. Thus $r_{i,j}$ goes from $-l$ to l. The Jacobi determinant for this is $(2l)^6$, as we have 6 variables. We solve the integral as it appears with $x_1, x_2, y_1, y_2, z_1, z_2$.

To get better precision we also run the Monte Carlo method by using importance sampling. For this case we return to spherical coordinates,

$$I = \int_0^\infty dr_1 dr_2 \int_0^\pi d\theta_1 d\theta_2 \int_0^{2\pi} d\phi_1 d\phi_2 \frac{r_1^2 r_2^2 \sin\theta_1 \sin\theta_2}{r_{12}}, \quad (7)$$

where r_{12} is given in eq. (4). The PDF of r_i is given by

$$p(r_i) = 4e^{-4r_i}, \quad (8)$$

2

so we want to use exponentially distributed random numbers. This is done by changing variables as follows

$$r_i \rightarrow -\frac{1}{4} \ln(1 - x_i)$$

$$\theta_i \rightarrow \pi x_i$$

$$\phi_i \rightarrow 2\pi x_i,$$

where x_i is a random number between 0 and 1. The Jacobi determinant is then given by $(1/4)^2 \times \pi^2 \times (2\pi)^2 = \pi^4/4$.

Lastly, we calculate the variance, which is given by

$$\sigma^2 = \int_{-\infty}^{\infty} dx P(x)(x - \mu)^2 = \langle x^2 \rangle - \langle x \rangle^2. \tag{9}$$

3   Results

The results for the various methods are collected here with three leading digits:

Method	N	I	σ	t (s)
Gauss-Legendre	25	0.196	-	50
Gauss-Laguerre	25	0.192	-	38
Monte Carlo	$5 \cdot 10^7$	0.197	$7.00147 \cdot 10^{-3}$	23
Monte Carlo*	$1 \cdot 10^6$	0.193	$1.73337 \cdot 10^{-2}$	0

where * denotes Monte Carlo method with importance sampling. The N values given here are the minimal values we found to give satisfying results. The exact solution with three leading digits is 0.193.

The result from a run of the code can be seen in the file `selected_results.txt`.

4   Conclusions

From the results it is obvious that Monte Carlo with importance sampling is the preferred method. The result is excellent and the time spent to run the code is incredibly low.

The result from the regular Monte Carlo method is not very impressive. For lower N the result becomes too small, and for larger N the code takes too long to give a result.

The time it takes to run Gauss-Legendre and Gauss-Laguerre increases rapidly with N, making these methods unbearably slow. Testing out different values of N showed that the result from Gauss-Legendre was very unstable with changing N, while Gauss-Laguerre improved for larger N. But the improvement in precision was low compared to the increase in time, so we limited the calculation to $N = 25$.

5   List of codes

The codes developed for this project are:

`main.cpp` – main program, C++
`plotting.m` – plotting program, MATLAB