abstract missing

your name

FYS4150 - Project 3
Numerical integration

[redacted]

October 19, 2015

1 Introduction

This project will look into different techniques for solving a multi-dimensional integral. The different methods will be compared with respect to speed and accuracy. The specific integral put under the microscope in this report is of quantum mechanical origin and is the following:

$$\langle \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \rangle = \int_{-\infty}^{\infty} d\mathbf{r}_1 d\mathbf{r}_2 e^{-2\alpha(r_1+r_2)} \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}. \tag{1.1}$$

$$\mathbf{r}_i = x_i \mathbf{e}_x + y_i \mathbf{e}_y + z_i \mathbf{e}_z,$$
$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}.$$

My background in quantum mechanics being non-existing this report will focus only on the mathematical and numerical aspects. The methods for numerical integration to be compared are *Gaussian Quadrature*, using both *Legendre polynomials* and *Laguerre polynomials* and *Monte Carlo integration-*

2 Method

2.1 Gaussian Quadrature

A widely used method to solve integrals numerically is Gaussian quadrature. As opposed to the more basic methods for numerical integration like, *the midpoint method*, *the trapezoidal method* and *Simpson's method*, Gaussian Quadrature does not use equidistantly spaced mesh points. However, how these mesh points are decided will be mentioned at a later point. First the general idea of Gaussian quadrature will be outlined.

The goal is to approximate the integral using a set of carefully chosen *weights*, ω_i and corresponding mesh points, x_i. In the methods referred to as basic in the previous paragraph all the weights were already chosen, and all the mesh points given for a choice of step-length and interval.

The weights used in Gaussian quadrature is obtained by using orthogonal polynomials for a specific interval. Commonly used polynomials are *Legendre polynomials*, *Laguerre polynomials*, *Hermite polynomials* and *Chebyshev polynomials*. This report will make use of the first two.

The integral is then approximated as follows,

$$\int_a^b W(x)f(x) = \sum_{i=1}^{N} \omega_i f(x_i)$$

and if $f(x)$ is a polynomial of degree less or equal to $2N-1$ this is exact and not an approximation.

For a more complete lay out of Gaussian Quadrature see (Hjort-Jensen, 2015).

2.1.1 Gauss-Legendre

To solve the integral (1.1) we will apply Gaussian quadrature using Legendre polynomials. To do that we must pick a set of finite integration limits, $[a, b]$, compute the weights and the mesh points corresponding to our interval. Since all the limits are the same we end up with six equal sets of integration points and weights.

2.1.2 Gauss-Laguerre

We will now solve the same integral using a different approach to see if the results become more accurate. A change to spherical coordinates and Laguerre polynomials will now be applied. The integral can now be written in terms of:

$$\int_0^\infty \int_0^\infty \int_0^\pi \int_0^\pi \int_0^{2\pi} \int_0^{2\pi} r_1^2 r_2^2 \frac{e^{-2\alpha r_1} e^{-2\alpha r_2}}{r_{12}} dr_1 dr_2 dcos(\theta_1) dcos(\theta_2) d\phi_1 d\phi_2$$

$$r_{12} = \sqrt{r_1^2 r_2^2 - 2r_1 r_2 cos(\beta)}$$

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

Gauss-Laguerre is applied to integrals on the form:

$$\int_0^{\infty} f(x)dx = \int_0^{\infty} x^{\tilde{\alpha}} e^{-x} g(x)dx$$

To fit our integral according to the form above we do coordinate transformations $x_1 = 2\alpha r_1$, $x_2 = 2\alpha r_2$ and $dx_1 = 2\alpha dr_1$, $dx_1 = 2\alpha dr_1$. The integral can then be written as:

$$\frac{1}{(2\alpha)^5} \int_0^{\infty} \int_0^{\infty} \int_0^{\pi} \int_0^{\pi} \int_0^{2\pi} \int_0^{2\pi} \frac{e^{-x_1} e^{-x_2}}{x_{12}} dx_1 dx_2 dcos(\theta_1) dcos(\theta_2) d\phi_1 d\phi_2$$

And for the angles the Legendre polynomials is used.

2.2 Monte Carlo integration

Monte Carlo integration is based on the *mean value theorem* from calculus stating:

> If $f(x)$ is a continuous function in $[a, b]$, then there exists a $c \in [a, b]$ such that $\int_a^b f(x) = (b - a)f(c) = (b - a)\mu$. μ being the *true mean value* of $f(x) \in [a, b]$.

The goal is to make a good approximation(*sample mean*), of the true mean value of $f(x)$ by simply using random numbers for $x \in [a, b]$. If we could do infinitely many samples *the law of large numbers* ensures us that the sample mean will converge towards to true mean value. However, not being able to run infinitely many cycles we must introduce an error for this method.

For a large number of samples N the *standard deviation*, σ_I of the integral becomes:

$$\sigma_I = \frac{1}{\sqrt{N}}$$

2.2.1 Brute force: uniform distribution

Using brute force-, or *"blind peoples shooting range"*- Monte carlo integration as some might call it, we will not make any considerations concerning the specific target integral, simply just draw random numbers for the different variables.

Our integral is on the form:

$$I = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(x_1, y_1, z_1, x_2, y_2, z_2) dx_1 dy_1 dz_1 dx_2 dy_2 dz_2$$

Since we are stuck with finite integration limits, $[-\infty, \infty] \mapsto [a, b]$ (this notation would probably make a mathematician cry, since this is an strictly illegal mapping), it can be approximated as:

$$I \simeq (b-a)^6 \frac{1}{N} \sum_{i=1}^{N} f(x_{1(i)}, y_{1(i)}, z_{1(i)}, x_{2(i)}, y_{2(i)}, z_{2(i)})$$

$x_{1(i)}, y_{1(i)}, ..$ being random numbers uniformly distributed in $[a, b]$.

2.2.2 Importance sampling and exponential distribution

A more clever way to solve such an integral using Monte carlo integration is to take a closer look at the integrand and the integration variables. The integrand involves an exponential function, so using an exponential distribution is a good choice.

Let's say we want to use an exponential distribution on the form, $p(y) = \frac{1}{\lambda} e^{-y/\lambda}$, then we have $x(y) = \int_0^y \frac{1}{\lambda} e^{-\tilde{y}/\lambda} d\tilde{y} = 1 - e^{-y/\lambda}$. This means that $y = -\lambda ln(1-x)$, for a random number $x \in [0, 1]$ (normal distribution).

The integrand can be written as a function of $cos(\theta)$, instead of θ be using trigonometric identities:

$$cos(\beta) = cos(\theta_1) cos(\theta_2) + \sqrt{1 - cos^2(\theta_1)} \sqrt{1 - cos^2(\theta_2)} cos(\phi_1 - \phi_2)$$

Now we can look for something in the integrand that looks like $e^{-y/\lambda}$ and let it be absorbed in the "exponential weights". By choosing $\lambda = 1/2\alpha$ the function to integrate reduces to:

$$f(r_1, r_2, cos(\theta_1), cos(\theta_2), \phi_1, \phi_2) = \frac{r_1^2 r_2^2}{r_{12}}$$

This is almost too good to be true. Now it's time to get ready to draw a few thousand numbers.

Both r_1 and r_1 should be from the exponential distribution, $[0, \infty)$, $cos(\theta_1)$ and $cos(\theta_2)$ should be uniformly distributed random numbers from its natural habitat, $[-1, 1]$, and finally the ϕ-s uniformly in $[0, 2\pi]$.

Now the integral can be approximated as follows:

$$I \simeq (2\pi)^2 2^2 \frac{\lambda}{N} \sum_{i=1}^{N} f(r_{1(i)}, r_{2(i)}, cos(\theta_1)_{(i)}, cos(\theta_2)_{(i)}, \phi_{1(i)}, \phi_{2(i)})$$

For the impressive results using importance sampling, see the *results* section.

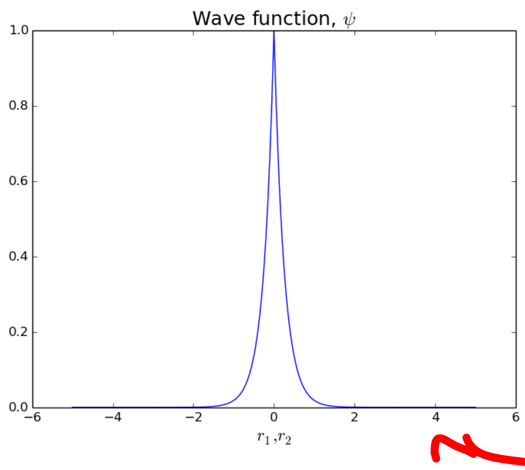
3 Implementation

For all programs, benchmark calculations and plots used see:

`https://github.com/[redacted]/FYS4150-gitrep`

A plot of the wave function, $\Psi(\mathbf{r}_1, \mathbf{r}_2) = e^{-\alpha(r_1+r_2)}$., shows that it dies off quickly and that finite integration limits around $3-5$ should give reasonably good results.

Labels



Labels

Figure 1: Plot of the wave function.

3.1 Parallelization

The monte carlo integration using importance sampling was parallelized using the software *OpenMP*.

4 Results

4.1 Gauss-Legendre

Solving integral (1.1) using Gauss-Legendre gave highly unstable results. Only for carefully chosen integration limits and lots of integration points were the results acceptable. Below is a set of results using this method.

Table 1: Results using Gaussian quadrature with Legendre polynomials. All the integrals were evaluated between the limits $[-5, 5]$ using N integration points.

N	**Approximation(I)**	**Relative error**
10	0.0111647	0.9421
15	0.3158630	0.6386
20	0.0967888	0.4979
25	0.2401350	0.2457

4.2 Gauss-Laguerre

Using Laguerre polynomials for the radial part, and Legendre polynomials for the angles have greatly improved the approximations. Now the result is stable and acceptable with just a few integration points.

Table 2: Results using Gaussian quadrature with Laguerre polynomials. All the integrals were evaluated between the limits $[-5, 5]$ using N integration points.

N	**Approximation(I)**	**Relative error**
10	0.1864573	0.0327
15	0.1897590	0.0156
20	0.1910818	0.0087
25	0.1917407	0.0053

4.3 Brute-force monte carlo

The approximations using brute-force monte carlo becomes fairly good, especially with respect to speed, as long as we do enough cycles.

Table 3: Results using brute-force Monte carlo. Integration limits $[-3, 3]$ using N Monte carlo cycles.

N	**Approximation(I)**	**Relative error**	σ_I	**CPU time [seconds]**
10e5	0.215615	0.118533	0.0712194	0.058493
10e6	0.234996	0.0352615	0.0352615	0.29887
10e7	0.194014	0.00647816	0.0115017	2.38187

4.4 Monte carlo (importance sampling and exponential distribution)

The results obtained using importance sampling and exponential distribution are simply impressive.

Table 4: Results using Monte carlo with importance sampling. Integration limits $[-3, 3]$ and using N Monte carlo cycles.

N	Approximation(I)	Relative error	σ_I	CPU time [seconds]
10e5	0.190141	0.0136135	0.00233383	0.431379
10e6	0.193754	0.00512584	0.00080656	4.05967
10e7	0.192751	7.49079e-05	0.000244273	40.668

One can also observe what we predicted earlier that it do not matter much if we set $[-3, 3]$ or $[-5, 5]$ as integration limits.

5 Concluding remarks

The results obtained using the Monte carlo integration methods, especially with importance sampling and exponential distribution, are good and accurate. Using these methods we get away with a lot less evaluations of the integrand. This "effect" gets stronger as the dimensions of the integral is increased.

Using Gauss-Laguerre the approximations are also pretty satisfactory. However, using $25 < N$, integration points was very time consuming compared to monte carlo integration.

6 References

Hjort-Jensen,M., 2015. *Computational physics*, accessible at course github repository. 551 pages.