Numerical integration methods

Henrik Lind Petlund | *github.com/henrikx2*

oday

Abstract

Key results, why is this work worthwhile? Credibility for my claims.

1 Introduction

The ground state corrolation energy between two electrons in a helium atom can be determined by solving a 6-dimensional integral. This integral is derived by modelling the wave function of each electron as an single-particle wave function of the electron in the hydrogen atom. For an electron i in the 1s state, the dimensionless and unnormalized single-particle wave function can be expressed as

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

where α is a parameter, and

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

with

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

The parameter $\alpha = 2$ gives the charge of the helium atom ($Z = 2$). Further, the wave function of the two 1s electrons are given by

$$\Psi(r_1 + r_2) = e^{-\alpha(r_2+r_2)}$$

The integral to solve in this report is the expectation value for the corrolation energy between the two electrons in the helium atom. The corrolation energy depends on the classical Coluomb interactions of the two electrons, and is given by

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

This (unnormalized) integral can be solved on closed form to be $5\pi^2/16^2 \approx 0.19276571$. This can be showed to be correct by ####

2 Methods

2.1 Gauss Quadrature

Gauss Quadrature is a method that uses orthogonal polynomials with weight functions to estimate integrals and are referenced in [@gaussQuad]. However, the topic is quite extensively to cover for this report and is therefore just explaned in short and otherwise sited.

2.2.1 Gauss-Legendre Quadrature (GauLeg) First off is using the Gaussian Quadrature with Legendre polinomials. These polinomials are defined at the interval $x \in [-1, 1]$ with the weight function $W(x) = 1$. The integral in Eq. (1) can be rewritten in terms of dx_i, dy_i and dz_i as

$$\langle \frac{1}{|\mathbf{r}_1 - \mathbf{r_2}|} \rangle =$$

$$\int\int\int\int\int\int_{-\infty}^{\infty} \frac{dx_1 dx_2 dy_1 dy_2 dz_1 dz_2 e^{-2\alpha(\sqrt{x_1^2+y_1^2+z_1^2}+\sqrt{x_2^2+y_2^2+z_2^2})}}{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}} \tag{2}$$

Now, every variable is defined on the interval $[-\infty, \infty]$, but since infinity cannot be represented exactly from a numerical point of view, it is here necessery to define infinity as a number. This is done to get small enough mesh points, so that the integral becomes more "continous". Figure 1 shows how the function e^{-2r} is approximately zero (< 0.01) when the $r \approx \lambda = 3$. Here, λ is the eigenvalue of the ground state single particle system. This gives that the interval $[-3, 3]$ should be sufficient to have three correct leading digits.

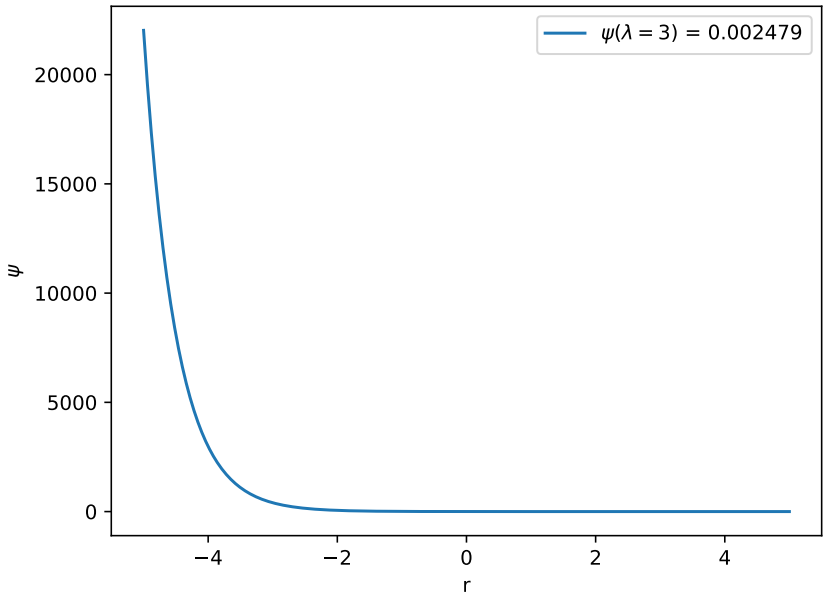

Figure 1: Plot of the wavefunction $\psi = e^{-2r}$ of a single particle in ground state. It's easy to see how the function converges to zero when r increases.

The integral to solve with Gauss-Legendre Quadrature is then given by

$$\int\int\int\int\int\int_{-3}^{3} \frac{dx_1 dx_2 dy_1 dy_2 dz_1 dz_2 e^{-2\alpha(\sqrt{x_1^2+y_1^2+z_1^2}+\sqrt{x_2^2+y_2^2+z_2^2})}}{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}} \tag{3}$$

This is solved in the program `gaussLegendre.cpp`.

2.2.2 Gauss-Laguerre Quadrature (Improved Gauss Quadrature/GauLag) The Gaussian Quadrature with Laguerre polinomials is defined at the interval $x \in [0, \infty]$ and has the corresponding weight function $W(x) = x^{\alpha'} e^{-x}$ $(\alpha' \neq \alpha)$. By changing to spherical coordinates

$$d\mathbf{r}_1 d\mathbf{r}_2 = r_1^2 dr_1 r_2^2 dr_2 dcos(\theta_1) dcos(\theta_2) d\phi_1 d\phi_2$$

with

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

and

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

it is possible to rewrite the integral with different integration limits ($\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$). This reads

$$\langle \frac{1}{|\mathbf{r}_1 - \mathbf{r_2}|} \rangle =$$
$$\int_0^\infty r_1^2 dr_1 \int_0^\infty r_2^2 dr_2 \int_0^\pi dcos(\theta_1) \int_0^\pi dcos(\theta_2) \int_0^{2\pi} d\phi_1 \int_0^{2\pi} d\phi_2 \frac{e^{-2\alpha(r_1+r_2)}}{r_{12}}$$

where

$$dcos(\theta_1) = -\sin(\theta_1)d\theta$$

such that

$$\langle \frac{1}{|\mathbf{r}_1 - \mathbf{r_2}|} \rangle =$$
$$\int_0^\infty r_1^2 dr_1 \int_0^\infty r_2^2 dr_2 \int_0^\pi sin(\theta_1)d\theta \int_0^\pi sin(\theta_2)d\theta \int_0^{2\pi} d\phi_1 \int_0^{2\pi} d\phi_2 \frac{e^{-2\alpha(r_1+r_2)}}{r_{12}}$$

Among these integrals, it is easiest to map ϕ_1, ϕ_2, θ_1 and θ_2 using Legandre polynomials and r_1 and r_2 using Laguerre polynomials. This is because $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$ is easily transformed to $[-1, 1]$ and r is already defined at the interval $[0, \infty]$. Taking the weight function, $W(x) = x^{\alpha'} e^{-x}$, with $\alpha' = 0$ (but $\alpha = 2$), into account for the $[0, \infty]$ integrals, the total integrand becomes

$$f(r_1, r_2) = \frac{e^{-3(r_1+r_2)} r_1^2 r_2^2}{r_{12}}$$

With the Jacobi determinant given by

$$\prod_{i=1}^{6} (b_i - a_i) = 4\pi^4 \cdot \frac{1}{(2\alpha)^2}$$

This integral is solved in the program `gaussLaguerre.cpp`.

2.2 Monte Carlo Integration

When using Monte Carlo integration, the integration points are defined using a probability distribution. As long as a sufficient number of psudo-random integration points are chosen; this is supposed to make the numerical approximation of the integral have less error. It is the choice of the probability distribution function (PDF) that determines the presicion of the Monte Carlo integration. A thorough explanation of the Monte Carlo methods can found in the lecture notes [@monteCarlo] of FYS3150.

2.2.1 Brute force Monte Carlo Integration (MCBF) The brute force Monte Carlo integration uses the uniform PDF given by

$$p(x) = \frac{1}{b-a}\Theta(x-a)\Theta(b-x)$$

where Θ is the Heaviside function and which at the interval $[a,b] = [0,1]$ gives the function $p(x) = 1$. In the case of Eq. (3) the interval is not $[0,1]$, but a change of variables such that

$$y(x) = a + (b-a)x$$

where $x \in [0,1]$ would make it possible to generate random numbers on the general interval $[a,b]$. In a multidimensional integral the change of variable is expressed using the indices i

$$x_i = a_i + (b_i - a_i)t_i$$

Using the integral from Eq. (3), the brute force integrand is given by

$$g(r_1, r_2) = \frac{e^{-2\alpha(\sqrt{x_1^2+y_1^2+z_1^2}+\sqrt{x_2^2+y_2^2+z_2^2})}}{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2}} \tag{4}$$

And the Jacobi-determinant given by

$$\prod_{i=1}^{d}(b_i - a_i) = (b-a)^6$$

which must be multiplied in the integral in the end.

This integral is solved in the program `monteCarloBruteForce.cpp`

2.2.2 Improved Monte Carlo Integration (MCIS) The improved Monte Carlo method introduces two new aspects to improve the results, namely; *change of variable* and *importance sampling*. In general, when doing importance sampling, one uses a PDF that has similarities with the integrand itself. In this case (when transforming to spherical coordinates) the integrals with r-dependance would satisfy the exponential distribution given by

$$p(y) = e^{-y}$$

From [@monteCarlo] this function gives the change of variable as

$$y(x) = -ln(1 - x)$$

where x is a random number generated by i.e. the `ran0`-function. Since the exponential expression in the integrand is raised to the power of $-4r$, it would also be needed to alter the "change of variable"-expression such that $y = 2\alpha y'$ and

$$y'(x) = -\frac{1}{2\alpha} ln(x - 1)$$

As for the other integrands with θ and ϕ dependance, the change of variable follows the uniform distribution with $x \in [0, 1]$ as follows

$$y(x) = a - (b - a)x = bx$$

After applying this, the integrand will have the form

$$\frac{r_1^2 r_2^2 sin(\theta_1) sin(\theta_2)}{r_{12}}$$

and in the end it's important to multiply this with the Jacobi determinant which reads

$$\prod_{i=1}^{6} (b_i - a_i) = 4\pi^4 \cdot \frac{1}{(2\alpha)^2}$$

This integration is solved in `monteCarloImportanceSampling.cpp`.

2.2.3 Improved Monte Carlo Integration with Parallization Parallization of the program `monteCarloImportanceSampling.cpp` is done with the use of openMP/MPI to see if this gives a considerable speed up.

This program is found in `monteCarloImportanceSamplingParallization.cpp`.

3 Resulsts

3.1 GauLeg and GauLag Table 1 and 2 shows the integral approximation, error and time usage for the gaussian legendre and laguerre methods respectively as functions of n.

Steps	Integral	Error	Time
n=5	0.264249	0.071483	0.0085196 s
n=7	0.329525	0.136759	0.0707594 s
n=9	0.321518	0.128753	0.408223 s
n=11	0.291261	0.098495	1.01692 s
n=13	0.261821	0.069055	2.82199 s
n=15	0.239088	0.046323	12.228 s
n=17	0.222933	0.030167	18.0855 s
n=19	0.211832	0.019066	28.9484 s
n=21	0.204307	0.011541	92.0323 s
n=25	0.195817	0.003051	123.012 s
n=27	0.193524	0.000759	263.981 s

3.2 MCBF and MBIS Table ## shows the the experimental approximation, error, σ, σ^2 and time usage of the two different (MCBR and MCIS) algorithms with varying n's.

4 Discusson

5 Conclusion

6 Appendix

Rererences