Draft version October 17, 2019
Typeset using LaTeX **twocolumn** style in AASTeX62

Solving Eigenvalue Problems by Means of the Jacobi Algorithm

Håkon Tansem, Nils-Ole Stutzer, and Bernhard Nornes Lotsberg

Abstract

1. INTRODUCTION

When solving problems in science and mathematics an ever recurring problem is to solve integrals. Integrals are found in all sorts of manners both directely and indirectely though e.g. differential equations. In this paper we will consider ways of solving an example of a six dimensional integral in several different ways. We will consider a brute force Gauss-Legendre quadrature, an improved Gauss-Laguerre quadrature, a brute force monte carlo integration and a monte carlo integration with importance sampling. The integral to solve is an expectation value problem from quantum mechanics. The resulting integrals will be compared to the analytical solution and the run times of each mothod are compared, in order to find which method is most efficient.

In the theory section we present needed theory wich we discuss how to implement in the method section. The results are presented in the results section and discussed in the discussion section.

2. THEORY

2.1. *The integral*

Before stating the needed integration methods we used we present the integral to integrate.

First we assume that the wave function of two electrons can be modelled like a the single-particle wave function of an electron in the hydrogen atom. The wave function of the ith electon in the 1 s state is given in terms of

$$\vec{r}_i = x_i\hat{e}_x + y_i\hat{e}_y + z_i\hat{e}_z \tag{1}$$

the dimensionless posission, where $\hat{e}_i$ are orthonormal unit vectors, so that the wave function

$$\psi_{1,s}(\vec{r}_i) = e^{-\alpha r_i}. \tag{2}$$

The distance $r_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ and we let the parameter $\alpha = 2$ corresponding to the charge of a helium $Z = 2$. Then the ansats for the wave function for two electrons is given by the product of the two 1 s wave functions

$$\Psi(\vec{r}_1, \vec{r}_2) = e^{-\alpha(r_1+r_2)}. \tag{3}$$

We now want to find the expectation value of the correlation energy between electrons which repel each other by means of the Coulomb interaction as

$$\langle\frac{1}{\vec{r}_1 - \vec{r}_2}\rangle = \int d\vec{r}_1 d\vec{r}_2 e^{-2\alpha(r_1+r_2)}\frac{1}{|\vec{r}_1 - \vec{r}_2|}. \tag{4}$$

This integral has an analytical solution $5\pi^2/16^2$, which we can later compare with.

2.2. *Brute Force Gauss-Legendre Quadrature*

The following theory presented follows closely (**?**, Ch. 5.3).

The essence of Gaussian Quadrature (GQ) is to approximate an integral

$$I = \int f(x)dx \approx \sum_{i=1}^{N}\omega_i f(x_i), \tag{5}$$

for some weights ω_i and grid points x_i. The grid points and wights are obtained through the zeros of othogonal polynomials. These polynomials are orthogonal on some intervale, for instance $[-1, 1]$ for Legendre polynomials. Since we must find N grid points and weigths, we must fit $2N$ parameters. Therefore we must approximate the integrand $f(x)$ by a polynomial of degree $2N-1$, i.e. $f(x) \approx P_{2N-1}(x)$. Then the integral

$$I \approx \int P_{2N-1}(x)dx = \sum_{i=0}^{N-1} P_{2N-1}(x_i)\omega_i. \tag{6}$$

GQ can integrate all polynomials of degree up to $2N-1$ exactelly, we thus get an equallity when approximating the integral over P_{2N-1}.

If we choose to expande the polynomial P_{2N-1} in terms of the Legendre polynomials L_N, we can through polynomial devision write

$$P_{2N-1} = L_N(x)P_{N-1}(x) + Q_{N-1}(x), \tag{7}$$

where P_{N-1} and Q_{N-1} are polynomials of degree $N-1$. We can thus write

$$I \approx \int_{-1}^{1} P_{2N-1}(x)dx = \int_{-1}^{1}(L_N(x)P_{N-1}(x) + Q_{N-1}(x))dx \tag{8}$$

$$= \int_{-1}^{1} Q_{N-1}(x)dx, \tag{9}$$

where the last equallity is due to the orthogonallity between $L_N(x)$ and P_{N-1}. Furthermore if one lets x_k be the zeros of L_N we have $P_{2N-1} = Q_{N-1}$, for $k = 0, 1, 2, \ldots, N-1$, we can fully define the polynomial $Q_{N-1}(x)$ and thus the integral. The polynomial $Q_{N-1}(x)$ can further be expanded in terms of the Legendre basis

$$Q_{N-1}(x) = \sum_{i=0}^{N-1} \alpha L_i(x). \tag{10}$$

Integrating this we get

$$\int_{-1}^{1} Q_{N-1}(x)dx = \sum_{i=0}^{N-1} \alpha_i \int_{-1}^{1} L_0(x)L_i(x)dx = 2\alpha, \tag{11}$$

where we insert that the first Legendre polynomial is normalized $L_0 = 1$ and utilize the orthogonality relation of the basis, i.e. $\int_{-1}^{1} L_i(x)L_j(x)dx = \frac{2}{2_i+1}\delta_{ij}$.

Since we know the value of $Q_{N-1}(x)$ at the zeros of L_N, we can rewrite the expansion of Q_{N-1} as

$$Q_{N-1}(x_k) = \sum_{i=0}^{N-1} \alpha L_i(x_k). \tag{12}$$

The resulting matrix $L_i(x_k) = L_{ik}$ has linearly independent columns due to the Legendre polynomials being linearly independent as well. Therefore the matrix L_{ik} is orthogonal, i.e.

$$L^{-1}L = I. \tag{13}$$

We thus multiply both sides of (**??**) by $\sum_{i=0}^{N-1} L_{ij}^{-1}$, so that

$$\alpha_k = \sum_{i=0}^{N-1} (L^{-1})_{ki} Q_{N-1}(x_i). \tag{14}$$

This result in addition to the approximation of the integral then gives

$$\begin{aligned} I &\approx \int_{-1}^{1} P_{2N-1}(x)dx = \int_{-1}^{1} Q_{N-1}(x)dx = 2\alpha_0 && (15)\\ &= 2\sum_{i=0}^{N-1} (L^{-1})_{0i} P_{2N-1}(x_i). && (16) \end{aligned}$$

Here we can clearly see that the weights $\omega_i = 2(L^{-1})_{0i}$ and the meshpoints x_i are the zeros of $L_N(x)$. Finding the weights is now simply a matrix inversion problem. We now have an integral approximation

$$\begin{aligned} I &= \int_{-1}^{1} f(x)dx \approx \int_{-1}^{1} P_{2N-1}(x)dx && (17)\\ &= \sum_{i=0}^{N-1} \omega_i P_{2N-1}(x_i), && (18) \end{aligned}$$

if $f(x)$ can be approximated by a poluynomial of degree $2N-1$ $P_{2N-1}(x)$. When performing an integral with more general limits $[a, b]$, one can now perform a simple change of variables $\tilde{x} = \frac{b-a}{2}x + \frac{b+a}{2}$, to accomodate for this.

When integrating a six dimensional integral like the quantum mechanical problem stated earlier, we can simply use Gauss-Legendre quadrature on all six integrals in a brute force way. Then the integral simply becomes

$$\begin{aligned} I &= \int_{-\infty}^{\infty} d\vec{r}_1 d\vec{r}_2 e^{-2\alpha(r_1+r_2)} \frac{1}{|\vec{r}_1 - \vec{r}_2|} && (19)\\ &= \sum_{i,j,k,l,m,n=0}^{N-1} \omega_i^{x_1}\omega_j^{y_1}\omega_k^{z_1}\omega_l^{x_2}\omega_m^{y_2}\omega_n^{z_2} f(x_1, y_1, z_1, x_2, y_2, z_2), && (20) \end{aligned}$$

where each cartesian variable x_i results in its own weight, and the integrand is denoted by the function f. Note that also a change of variables was performed to accomodate for the different integration limits.

2.3. *Improved Gaussian Quadrature*

The above meansioned Gauss Laguerre quadrature is in fact quite inacurate in the case of our integral. In order to improve on the method, one can use a different orthogonal polynomial basis. In our case since we have an integral on the form

$$I = \int_0^{\infty} f(x)dx = \int_0^{\infty} x^2 e^{-x} g(x)dx \tag{21}$$

it is in fact way more efficient to use Laguerre polynomials as oppose to Legendre polynomials. Then when finding an approximation to the integral the x^2e^{-x} part of the integrand is absorbed into the weights ω_i. The derivation of the weights is however not shown here as it is completely analogous to the derivation shown for the Gauss-Legendre quadrature.

In order to transorm our integrand to a form resembeling (**??**) we need to transform from cartesian to spherical, i.e. $(x, y, z) \to (r, \theta, \phi)$. In spherical coordinates we get that

$$d\vec{r}_1 d\vec{r}_2 = r_1^2 r_2^2 dr_1 dr_2 \sin(\theta_1)\sin(\theta_2) d\theta_1 d\theta_2 d\phi_1 d\phi_2 \tag{22}$$

$$|\vec{r}_1 - \vec{r}_2| = \sqrt{r_1^2 + r_2^2 - 2r_1 r_2 cos(\beta)}, \tag{23}$$

where $cos(\beta) = cos(\theta_1)\cos(\theta_2) + \sin(\theta_1)\sin(\theta_2)\cos(\phi_1 - \phi_2)$. We introduce the change of variables $u = \alpha r$ so that the integral is on the form

$$I = \frac{1}{32\alpha^5} \int_0^{\pi}\int_0^{\pi}\int_0^{2\pi}\int_0^{2\pi}\int_0^{\infty}\int_0^{\infty} f du_1 du_2 d\theta_1 d\theta_2 d\phi_1 d\phi_2, \tag{24}$$

where $f = f(u_1, u_2, \theta_1, \theta_2, \phi_1, \phi_2) = \frac{\sin(\theta_1)\sin(\theta_2)u_1^2 u_2^2 e^{-(u_1+u_2)}}{\sqrt{u_1^2+u_2^2-2u_1u_2cos(\beta)}}$ is the new integrand. Now for the radial part (u_1 and u_2) a Gauss-Laguerre approach is used, while we use Gauss-Legendre quadrature for the angular part. (HUSK AT HVORFOR VI KASTER r12=0).

2.4. *Brute Force Monte Carlo Integration*

An integration method frequently used, espessially when computing multi-dimensional integrals as its error remains constant for any higher dimension, is the Monte Carlo integration. In this integration method one approximates the integral to solve by an expectation value. Consider for instance an integral

$$I = \int_a^b f(x)dx = (b-a)\int_a^b \frac{f(x)}{b-a}dx \tag{25}$$

$$= (b-a)\int_a^b f(x)p(x)dx, \tag{26}$$

where we let $p(x) = \frac{1}{b-a}$ be the uniform probability density function PDF for stochastic variables $x \in [a, b]$. Since we know that an expectation value

$$\langle f(x)\rangle = \int_a^b f(x)p(x)dx \approx \frac{1}{N}\sum_{i=0}^{N-1} f(x_i), \tag{27}$$

where we used that the expectation value of $f(x)$ is approximately the average of the $f(x_i)$'s where the x_i's are drawn from the distribution $p(x)$, for large enough N.

Note that we implisitly performed a mapping in this case. Because uniform distribution by default only returns values $y \in [0, 1]$, we change the variable to $x = a + (b-a)y$ so as to draw values y_i from a uniform distribution between a and b.

The integral can thus be approximated by

$$I \approx (b-a)\langle f(x)\rangle \approx \frac{b-a}{N}\sum_{i=0}^{N-1} f(x_i). \tag{28}$$

In order to get an estimate for the accuracy of the integration we can calculate the variance defined as

$$\sigma^2 = \frac{1}{N}\sum_{i=0}^{N-1} f^2(x_i) - \left(\frac{1}{N}\sum_{i=0}^{N-1} f(x_i)\right)^2 = \langle f^2\rangle - \langle f\rangle^2. \tag{29}$$

The variance is a measure of how far the integral I is from the mean.

Considering the quantum mechanical integral in cartesian we want to compute, we let $a = -\lambda$ and $b = \lambda$ for some λ approximating infinity. The we simply get a six-dimensional expectation value so that

$$I = (b-a)^6\langle f\rangle \approx \frac{(b-a)^6}{N}\sum_{i=0}^{N-1} f(x_1^i, y_1^i, z_1^i, x_2^i, y_2^i, z_2^i), \tag{30}$$

where the cartesian coordiantes x^i are all drawn from a uniform distribution for $x^i \in [-\lambda, \lambda]$. The function f represents the integrand function (in this case as a function of cartesian samples). The variance is calculated completely analogous to the one-dimensional case.

2.5. *Improved Monte Carlo Integration*

One way to improve the Monte Carlo Integration shown in the previous subsection, i.e. to reduce its variance, is to use a different PDE to draw the stochastic variables from.

If we consider the quantum mechanical integral in spherical coordinates, as shown previously, we recognize the e^{-u} part as an exponential distribution. Thus we let $p(y) = e^{-y}$ denote the exponential distribution. Using conservation of probability under change of variable, we set $p(y)dy = exp(-y)dy = p(x)dx = dx$ for the uniform PDF $p(x)$ with $x \in [0, 1]$. If we integrate this cumulative distribution we find

$$x(y) = \int_0^y \exp(-\xi)d\xi = 1 - \exp(-y), \tag{31}$$

wich we can invert to get the mapping $y(x) = -\ln(1-x)$ from the uniform to the exponential PDF. Now $y \in [0\infty)$ is quite convenient to represent the radial distance u. We can now absorbe the exponential part of the integral into the expectation value so that

$$I = \int_0^\infty f(u)du = \int_0^\infty \frac{f(u)}{p(u)}p(u)du \tag{32}$$

$$= \int_0^1 g(u(x))dx \approx \frac{1}{N}\sum_{i=0}^{N-1} g(u(x_i)), \tag{33}$$

where we used that $p(u) = \exp(-u)$ and that $p(u)du = dx$ to change variables. The samples $u(x_i)$ are now drawn from the exponential distribution $p(u)$. This is what is called importance sampling, which results in a lower variance than using the brute force Monte Carlo integration, because we simply sample from a distribution which fits the integrand better.

However, we must note that the integral we want to solve in spherical coordinates only has limits $[0, \infty)$ for the two radial coordiantes u_1 and u_2. We must thus sample from the uniform distribution for the angular integrals, as the angles exist within the finit limits $[0, \pi]$ and $[0, 2\pi]$, while the radial integral should utilize importance sampling from the exponential distribution.

The final integral then looks as follows

$$I \approx \frac{1}{N}\frac{\pi^4}{8\alpha^5}\sum_{i=0}^{N-1}\frac{\sin\left(\theta_1^i\right)\sin\left(\theta_2^i\right)(u_1^i)^2(u_2^i)^2}{\sqrt{(u_1^i)^2+(u_2^i)^2-2u_1^iu_2^icos(\beta^i)}}, \tag{34}$$

where the index i simply represents that the corresponding sample is drawn from its respective PDF.

3. METHOD

4. RESULTS

The results were produced running on a MacBook Pro (macOS Mojave 10.14.6) with 8gb RAM using a dual coreIntel Core i5-7360U 2.3GHz CPU with four threads. The program was compiled using openMP for the parallelization with the gcc compiler version (Homebrew GCC 9.2.0_1) 9.2.0.

When calculating the integral, the relative error was calculated for the integral given by (**??**) using the brute force method, as described in section(REFERER TIL BRUTE FORCE QUAD SECTION). The relative error was also calculated for the integral in spherical coordinates given by (REFERER TIL SFÆRISK INTEGRAL) using the improved method, as described in section (REFERER TIL IMPROVED GAUSSIAN QUAD SECTION). These results are shown in figure **??**. In the figure a line is drawn when the relative error surpasses 10^{-2} to illustrate when the method achieves three digit precision.

The relative error was also compared for both the brute force Monte Carlo method and the improved Monte Carlo method described in section (REFERER TIL BRUTE FORCE) and (REFERER TIL IMPROVED) respectively. This is shown in figure **??**. For the same calculations using both Monte Carlo methods, the variance was also plotted. This is result is illustrated in figure **??**.

Figures/exercise_a_b.pdf

Figure 1. Figure showing the relative error for the brute force gaussian quadrature and the improved gaussian quadrature methods described in section (REFERER TIL SEKSJON). A line is drawn when the relative error is 10^{-}2 to illustrate when the integral reaches three digit precision.

The CPU time for both the brute force and the improved Monte Carlo methods was calculated using no parallelization and parallelization with two threads. This is shown in figure **??**. The three different compiler flags -O1, -O2 and -O3 was also compared to study their impact on the CPU time, which is shown in figure **??**. This result was produced using parallelization with four threads.

Figures/error_monte_carlo.pdf

Figure 2. Figure showing the relative error for the brute force and the improved Monte Carlo methods described in section (REFERER TIL SEKSJON).

Figures/variance_monte_carlo.pdf

Figure 3. Figure showing the variance for the brute force and the improved Monte Carlo methods as described in sections (REFERER TIL SEKSJONER) and respectively.

Figures/cpu_time_monte_carlo.pdf

Figure 4. Figure showing the CPU time the brute force and the improved Monte Carlo methods described in section (REFERER TIL SEKSJON). For both methods the CPU time was compared both unparallelized and parallelized with two threads.

Figures/cpu_time_compilerflag.pdf

Figure 5. Figure showing showing the CPU time for the improved Monte Carlo method, as described in section (REFERER TIL SEKSJON), using the different compiler flags -O1, -O2 and -O3. These results were produced using parallelization with four threads.

5. DISCUSSION

6. CONCLUSION

nocitejensen:2019