Solving multidimensional Integrals using Monte Carlo Integration

Ove Haugvaldstad, Eirik Gallefoss

November 16, 2019

Abstract

This work aims to evaluate Monte Carlo Integration (MCI) for solving physical problems with many degrees of freedom. The physical problem in question is the ground state correlation energy between two electrons in a helium atom, which we solved using four different approaches, "brute force" MCI, MCI with importance sampling, Gaussian Quadrature (GQ) with Legendre polynomials and GQ combining Legendre- and Laguerre polynomials. We also parallelized the code. We did an extensive analysis of the error, CPU time and scalability of the different approaches. The result of our analysis showed that we achieved a close to optimal speed up for MCI with importance sampling. For run times larger than 10 seconds MCI with importance sampling had an error that was between 10 and 383 times smaller than the GQ Laguerre. Importance sampling MCI had also on average 57 times lower error compared to brute force MCI.

1 Introduction

Physical systems are often too complex to be solved analytically due to having too many physical parameters (degrees of freedom) which can all vary independently. To circumvent having to solve for each degree of freedom, we can instead randomly sample the behaviour of each parameter and take the average. This kind of statistical approach is known as Monte Carlo Integration (MCI).

Our aim is to demonstrate MCI, which we believe also yields a rewarding insight into probability and statistics. Then we will demonstrate how to improve the accuracy of MCI, by using a technique called importance sampling. We will also briefly describe Gaussian Quadrature (GQ) which we will compare against MCI. The physical problem we will look at is the ground state correlation energy between two electrons in a helium atom. We compute the correlation energy both using GQ and MCI. To be clear this work will not go into any quantum mechanical details and we will treat this purely as a mathematical problem.

2 Methods

To be able to calculate the correlation energy, we need an ansatz eq. (1) for the wave function for the two electrons.

$$\Psi(\mathbf{r}_2, \mathbf{r}_2) = e^{-\alpha(r_1+r_2)} \tag{1}$$

The integral we will solve is the quantum mechanical expectation value for the correlation energy eq. (2), where r$_i$ is the distance from the origin given by eq. (3). Here we have also set $\alpha = 2$. The integral has an analytical solution equal to $5\pi^2/16^2$ that we will use to evaluate the errors.

$$I = \int_{-\infty}^{\infty} d\mathbf{r}_1 d\mathbf{r}_2 e^{-4(r_1+r_2)} \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \tag{2}$$

$$r_i = \sqrt{x_i^2 + y_i^2 + z_i^2} \tag{3}$$

2.1 Transformation to spherical coordinates

Before we move onto describing MCI it is useful to transform our integral to spherical coordinates. Why this is useful will be come clear when we discuss MCI using importance sampling and GQ using Laguerre polynomials. To give some idea as why, notice that our integral eq. (2) runs from $-\infty \to \infty$, this integral is not feasible to integrate without making an approximation of infinity.

Transforming $d\mathbf{r}_1 = dx_1dy_1dz_1$ to spherical coordinates gives us $r_1^2dr_1 \sin\theta_1 d\theta_1 d\phi$ with the new limits $r_1 \in [0, \infty)$, $\theta_1 \in [0, 2\pi]$ and $\phi_1 \in [0, \pi]$. The same holds for $d\mathbf{r}_2$. In Hjort-Jensen 2015 it is shown that the distance between $\mathbf{r}_1$ and $\mathbf{r}_2$ is given by eq. (5), with $\cos\beta$ defined in eq. (6). Then the integral in spherical coordinates becomes eq. (4).

$$I_s = \int_0^\infty \int_0^\infty \int_0^\pi \int_0^\pi \int_0^{2\pi} \int_0^{2\pi} dr_1dr_2d\theta_1d\theta_2d\phi_1d\phi_2r_1^2r_2^2 \sin\theta_1 \sin\theta_2 \frac{e^{-4(r_1+r_2)}}{r_{12}} \tag{4}$$

$$\frac{1}{r_{12}} = \frac{1}{\sqrt{r_1^2 + r_2^2 - 2r_1r_2\cos\beta}} \tag{5}$$

$$\cos\beta = \cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2\cos\phi_1 - \phi_2 \tag{6}$$

2.2 Monte-Carlo Integration

The idea behind Monte Carlo Integration (MCI) is to approximate an integral by evaluating the integrand at randomly selected points in the integration domain. Constructing estimators that have expectation value equal to the integral ensures that the variance of our estimations decreases as the sample size increases.

The random points can be selected in many different ways. Lazzarini 1901 threw roughly 30000 needles to approximate π and Dumoulin and Thouin 2014 used a shotgun. We will use a random number generator (RNG) to select these points. For both brute force and importance sampling in several dimensions we will assume that $p(\mathbf{x}) = \prod_{i=1}^{d} p(\mathbf{x}_i)$, with d the number of dimensions, meaning that the sampling of points are independent events.

We first explain the theory for an integral in one dimension, on the interval 0 to 1, before generalizing it to other intervals and multi dimensional integrals.

2.2.1 Brute Force Monte-Carlo

In the brute force variant of MCI we use the estimator eq. (7), where the random variable X contains N points sampled from the uniform distribution.

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} f(X_i) \tag{7}$$

The expectation value of an estimator is $\langle \hat{I} \rangle = \int_0^1 \hat{I} p(x) dx$. $p(x)$ is the probability density function (pdf) of the random variable associated with the estimator. Since $p(x) = 1$ for $x \in [0, 1]$ $\langle \hat{I} \rangle = \int_0^1 f(x)dx$, showing that eq. (7) is an unbiased estimator of the $\int_0^1 f(x)dx$. The fact that our estimator is unbiased ensures that our estimator will converge to our integral.

To estimate integrals over other intervals we note that the uniform distribution on $[a, b]$ has pdf $p(x; a, b) = \frac{1}{b-a}$. To keep the estimator unbiased we have to add a factor $(b - a)$, leading to $\hat{I}_N = \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$. For an integral of dimension d we get the estimator shown in eq. (8).

$$\hat{I}_N = \frac{\prod_{j=1}^{d} b_j - a_j}{N}\sum_{i=1}^{N} f(\mathbf{x}_i) \tag{8}$$

with b_j and a_j being the upper and lower limits of the ith integral.

The standard deviation or the error for any MC estimator goes according to eq. (9)

$$\sigma \sim \frac{1}{\sqrt{N}} \tag{9}$$

In evaluating eq. (2) we will use the same approximation to infinity ($\infty \approx \lambda$) as when integrating using Legendre polynomials. Our estimator is then given by eq. (10).

$$\hat{I}_N = \frac{(2\lambda)^6}{N}\sum_{i=1}^{N} e^{-4(r_{1,i}+r_{2,i})} \tag{10}$$

2.2.2 Importance sampling

To obtain better estimates of the integral we can reduce the variance and standard deviation of our results by sampling from other distributions and using another estimator. Instead of sampling X from the uniform distribution we will sample from a distribution with pdf proportional to the integrand. We would then evaluate the integrand more often where it is large than where it is small. To account for this bias we also change our estimator to eq. (11), which also have expectation value equal to the integral of $f(x)$. By scaling with the pdf we weigh the range with high density X and large function values less.

$$\frac{1}{N}\sum_{i=1}^{N}\frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} \tag{11}$$

Since our integral in spherical coordinates eq. (4) have an exponential component in the radial domain we will sample r_1 and r_2 from the exponential distribution ($p(r) = ae^{-ar}$). Setting a = 4 gives $p(r_1, r_2) = p(r_1) \cdot p(r_2) = p(r_1, r_2) = 4e^{-4r_1}4e^{-4r_2}$. This leads to cancellation of the exponential part of the integral. For the rest of the integral we will still sample from the uniform distribution, but now with $\theta_1, \theta_2 \in [0, \pi]$ and $\phi_1, \phi_2 \in [0, 2\pi]$. The new estimator is then eq. (12). The factor $4\pi^4$ is a normalization factor due to sampling from the uniform distribution on other intervals than $[0, 1]$.

$$\hat{I}_N^p = \frac{4\pi^4}{N}\sum_{i=1}^{N}\frac{f(\mathbf{x}_i)}{p(r_{1,i}, r_{2,i})} = \frac{\pi^4}{4N}\sum_{i=1}^{N}\frac{r_{1,i}^2 r_{2,i}^2 \sin\theta_{1,i}\sin\theta_{2,i}}{|r_{12,i}|} \tag{12}$$

2.3 Gaussian Quadrature

Any quadrature rule can be seen as the as sum of the function we are integrating evaluated at specified sample points or integration points x_i multiplied with a weight w_i eq. (13).

$$\int_a^b f(x)dx \approx \sum_{j=1}^{N} f(x_i)w_i \tag{13}$$

For Gaussian Quadrature (GQ) the integration points are given by the roots of an orthogonal polynomial. The integration weights are calculated by finding the inverse of a matrix defined by the orthogonal polynomial eq. (14).

$$L = \begin{bmatrix} L_0(x_0) & L_1(x_0) & \dots & L_{N-1}(x_0) \\ L_0(x_1) & L_1(x_1) & \dots & L_{N-1}(x_1) \\ & & & \\ L_0(x_{n-1}) & L_1(x_{n-1}) & \dots & L_{N-1}(x_{n-1}) \end{bmatrix} \tag{14}$$

The different orthogonal polynomial are defined over specific intervals, for instance Legendre polynomials are defined for $x \in [-1, 1]$. Legendre polynomials can still be used to for solving integrals over other intervals than $[-1, 1]$ by using transformation of variables eq. (15).

$$t = \frac{b-a}{2}x + \frac{b+a}{2} \tag{15}$$

Then rewrite integral for an interval $[a, b]$,

$$\int_a^b f(t)dt = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{b+a}{2}\right) \tag{16}$$

An issue with using Legendre polynomials to generate our sample points and weight, is that the physical problem eq. (2) we want solve involves an integral from $-\infty \to \infty$. To decide on a suitable approximation we need to determine region the integrand has the largest contribution. For our integrand eq. (2) the exponential terms goes quickly towards zero as we move away from the origin. We decided our integrand to be sufficiently close to zero for $r \pm 2$.

To avoid having to approximate infinity we can instead solve the transformed version of the integral eq. (4), by using Laguerre polynomials to generate integration points and weights. The weight function when using Laguerre polynomials is defined in the following way eq. (17).

$$x^{\alpha}e^{-x}, \quad 0 \leq x \leq \infty \tag{17}$$

Something that is important to note about Laguerre polynomials is that (for $\alpha = 0$) a factor e^{-x} is baked into weights and we have to divide eq. (4) by $e^{-(r_1+r_2)}$. Then we will use Laguerre polynomials when integrating in the radial domain and use Legendre polynomials when integrating θ and ϕ.

2.4 Experimental setup

The code was run on a laptop with the Intel Core i5-8250U, with eight cores (4 physical cores with hyper threading). For MCI the number of samples for each experiment was taken on logarithmic intervals, with 50 different values of sample points between 10 and 10^{10}. The GQ experiments were run with the number of integration points between 1 and 50. The source code for our implementation of MCI and GQ can be found at our GitHub [1].

[1] https://github.com/Ovewh/Computilus/master/Project3

3 Results

3.1 Monte-Carlo integration

3.1.1 Error and accuracy

Figure 1 shows the absolute value of the difference between the calculated value of the integral and the analytical solution with brute force and importance sampling. N is the number of samples taken. It is worth to notice the sometimes large differences in error from one run to the next, due to the stochastic nature of MCI. Still the improvement in accuracy when using importance sampling is significant especially for larger samples sizes when the standard deviation is smaller. The mean of the error ratios between brute force and importance sampling was 57.

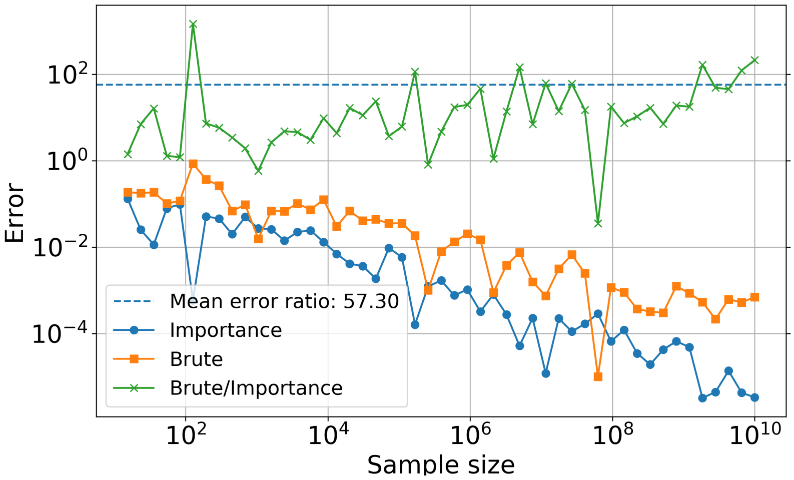

Figure 1: Absolute value of difference between Monte-Carlo results and analytical solution using brute force and importance sampling. Also shows ratio of error and the mean of the error ratio.

Figure 2 shows how the standard deviation evolves with increase in sample size. For large sample sizes we see that the standard deviation decreases according to what we would expect from the theoretical expression for the standard deviation eq. (9), to reduce the standard deviation by a factor 10 you have increase the sample size by a factor of 100. Figure 2 also show a reduction of the standard deviation by a factor of 10 for importance sampling MCI compared to brute force MCI.

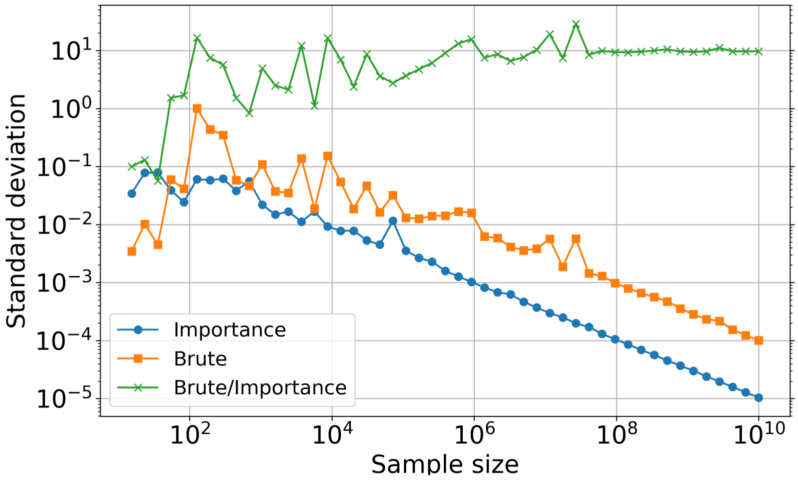

Figure 2: Standard deviation versus run time of MCI experiments.

3.1.2 Parallelization

Figure 3 highlight the benefit of parallelization, showing a reduction in run time by around a factor 4. Which is more or less what we would expect by the specifications of our laptop.

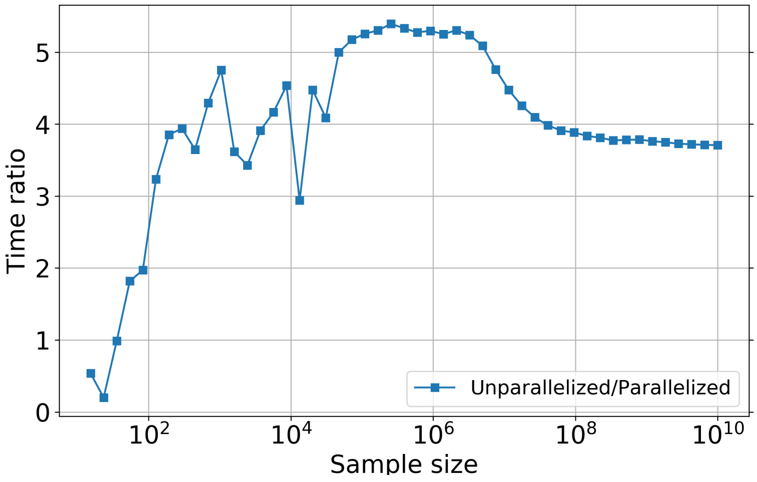

Figure 3: Run time of MCI using importance sampling. Timing of unparallelized version divided by run time of the parallelized version.

3.2 Gaussian Quadrature

The error of the GQ methods fig. 4 shows that GQ Legendre gives relatively low(high) errors when the number of integration points are odd(even). For GQ Legendre the error seems to converge around roughly 3×10^{-3}. The GQ Laguerre shows a huge decrease in the error with between 1 and 15 integration points, with the smallest errors at N = 14 and 15. For higher Ns the error increases and seem to converge to 2×10^{-3}, and does not show the oscillatory nature of the GQ Legendre.

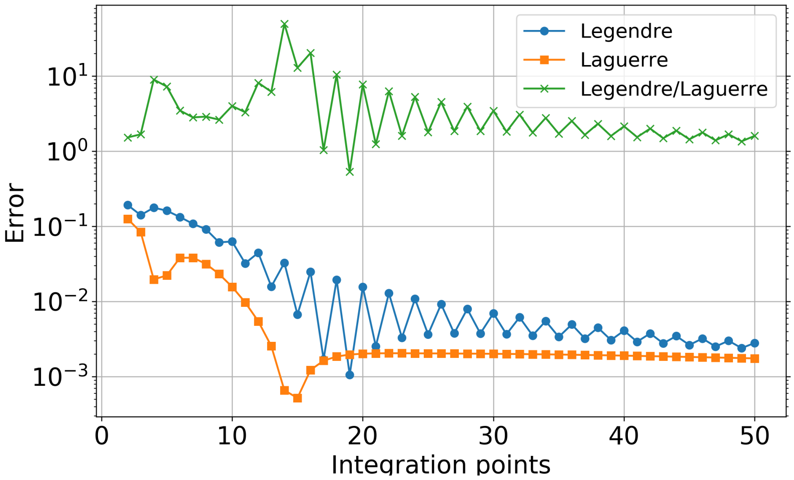

Figure 4: Absolute value of difference between GQ results and analytical solution using Legendre and Laguerre polynomials.

3.3 Comparison of Gaussian Quadrature and Monte-Carlo Integration

To decide whether running Gaussian Quadrature or Monte-Carlo Integration would give the best results for a given run time we made error run time plots for all of the methods we have used fig. 5. For MCI low values of run time corresponds to few samples, and for GQ it corresponds to a small number of integration points. For the shortest run times the results of MCI were a bit all over the place, but looking at longer run times we see a clear trend of the error decreasing, especially for the importance sampling.

We also did a linear interpolation of the error run time results in the range where we had results for both importance sampling and GQ Laguerre. Taking the rolling mean of three and three values for the error ratio of GQ Laguerre and importance sampling gave, for run times larger than 10 seconds, ratios between 20 and 383.

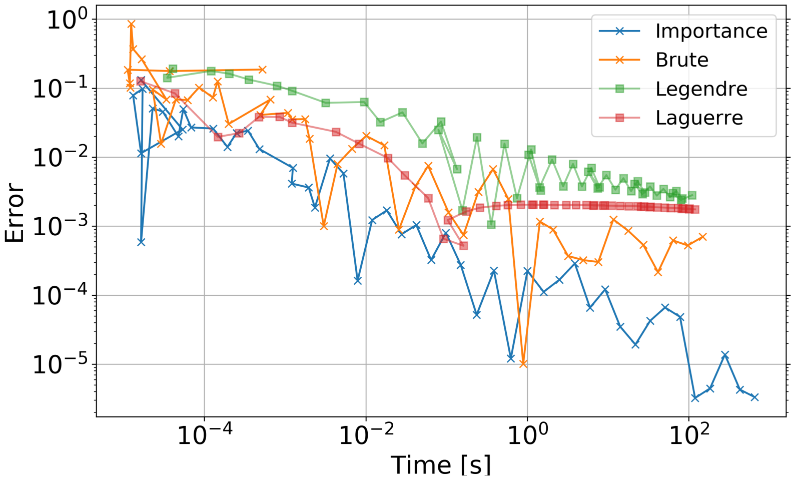

Figure 5: Absolute values of the error as a function of time for all methods we used to approximate the integral. All of the timings are taken from parallelized versions of the code.

4 Discussion

The result from our analysis clearly favours MCI, particularly when given more CPU time. The GQ approach did not benefit significantly with increased CPU time after a certain point, the sweet spot for GQ Laguerre being 14 and 15 integration points. Regardless we would not disregard using GQ, since the accuracy of the method is very much dependent on the problem being evaluated. In comparison MCI would behave more or less the same regardless of the physical problem, since the standard deviation would still follow eq. (9).

Still we think this experiment did provide some insight into the kind of situations where these two very different approaches would be best suited. For instance if the physical problem has many degrees of freedom $\nu \geq 6$, then MCI would most definitely yield better result. On the other hand, if the problem is less complex, then GQ might yield very satisfactory results with very little computational cost.

Another point to be made based on our results is the potential huge benefits from parallelization depending on how many CPU cores at your disposal. Combining this with current increase and availability of multicore processors, makes a even stronger case for MCI. Then there is also the future prospect of quantum computers, where quantum MCI can yield a quadratic improvement in error with the same sample size (Montanaro 2015).

5 Conclusion

In this study we have evaluated Monte Carlo Integration (MCI) for solving physical complex problems with many degrees of freedom. We calculated the ground state correlation energy between two electrons in a helium atom, which we implemented using two different MCI approaches, brute force MCI and importance sampling MCI. Then we also made two implementations using Gaussian Quadrature (GQ), a very different method, to compare against MCI.

The results are divided into three parts: Firstly we looked at the reduction in error for MCI with importance sampling compared to the brute force approach. We observed that importance sampling MCI on average reduced the error by a factor of around 60. Secondly we compared the two different GQ implementations (GQ Legendre and GQ Laguerre) and how the error evolved with increased CPU time. Showing that GQ implementations did not benefit significantly from more CPU time. Laguerre GQ actually yielded the lowest error for N between 14-15. Finally we did a comparison of all of our four implementations where we looked at the error versus CPU time. Where importance sampling MCI more or less made a clean sweep of the other implementations, beating GQ Laguerre by around a factor of almost 400 for longer runs. Importance sampling MCI yielded also on average 57 times lower error than brute force MCI, which in our analysis performed second best.

Lastly we want to point out that the physical problem we chose in our study was not really fair towards GQ due to the complexity of the problem, and in the case of a less complex problem, GQ might significantly outperform MCI.

References

[1] Vincent Dumoulin and Félix Thouin. "A Ballistic Monte Carlo Approximation of pi". In: *arXiv e-prints*, arXiv:1404.1499 (Apr. 2014), arXiv:1404.1499. arXiv: 1404.1499 [physics.pop-ph].

[2] O. Haugvaldstad and E. Gallefoss. "Source files for this project". 2019. URL: https://github.com/Ovewh/Computilus/tree/master/Project3.

[3] M. Hjort-Jensen. "Computational physics, lecture notes fall 2015". 2015. URL: https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Lectures/lectures2015.pdf.

[4] M. Lazzarini. "Applicazione del calcolo della probabilità alla ricerca sperimentale di un valore approssimato di pi". In: *Periodico di Matematica* (4 1901).

[5] Ashley Montanaro. "Quantum speedup of Monte Carlo methods". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2181 (2015), p. 20150301. DOI: 10.1098/rspa.2015.0301. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.2015.0301. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2015.0301.