Computational Physics Project 2

Department of Physics, Imperial College London

***Abstract*—We use numerical integration methods to compute the probability of locating a particle in a finite region of space for a quantum harmonic oscillator system. The integral is computed to a specified degree of accuracy using various methods, including Newton-Cotes summations and Monte Carlo simulations. We introduce a novel method for Monte Carlo integration which combines adaptive importance sampling with recursive stratified sampling to efficiently compute the required integral. We show that the new method is computationally superior to importance sampling and recursive stratified sampling applied separately. This project was completed using Python 3.7 with the Numpy, Scipy and Matplotlib libraries.**

I. Introduction

NUMERICAL integration methods, such as Newton-Cotes (NC) summations and Monte Carlo (MC) simulations, are widely used for solving ordinary and partial differential equations in physics and engineering. For example, the heat diffusion equation [2] is, in general, analytically intractable but can be solved using numerical integration techniques.

We define the *relative efficiency* of a numerical integration algorithm as the reciprocal of the number of integrand evaluations required for the integral to converge to a given degree of accuracy. For integrals in lower dimensions, NC integration schemes, such as the trapezium rule and Simpson's rule, are often the most efficient methods. However, as the dimensionality of the integral increases, the number of integrand evaluations for a given accuracy grows exponentially, colloquially known as the curse of dimensionality [3]. In these instances, MC simulations are the favoured method as the computation requirements do not significantly scale with the number of dimensions [4].

In this report, we will consider the wave-function, Ψ, of a particle in the ground state of a one-dimensional quantum harmonic oscillator. The probability of finding the particle in a finite region of space is given by the integral of $|\Psi|^2$ over space. We will investigate the efficiency of the trapezium rule and Simpson's rule in computing the required integral to a specified accuracy. Then, the integral will be performed using MC simulations with additional variance-reduction methods such as importance-sampling (IS) and recursive stratified sampling (RSS). Finally, we will introduce a novel method for computing the required integral with an algorithm that exploits adaptive IS and RSS. Subsequently, we will compare the efficiency of the new algorithm to other techniques used in this report.

We opt to use object-oriented programming (OOP) as the programming paradigm. The numerical methods we will use have characteristics that naturally lend themselves to OOP. For example, the trapezium rule and Simpson's rule classes can inherit from a Newton-Cotes parent class, which can perform similar error validation and attribute management for both of these routines. Similarly, we can define a generic Monte Carlo integration parent class and more specific routines such as importance sampling can inherit the main integration method from this class. This also makes code-management and debugging easier since each class can be treated as an individual unit and tested independently of other classes.

II. Problem Description

The wave-function of the particle is

$$\Psi(x,t) = \frac{1}{\sqrt[4]{\pi}} e^{ia(t)} \exp\left(\frac{x^2}{2}\right), \tag{1}$$

where $a(t)$ is a real-valued phase function corresponding to the time-dependent energy term $a(t) = E_0 t/\hbar$ for the ground state. Then, the probability of finding the particle in an infinitesimal region $x \to x + \mathrm{d}x$ is

$$\begin{aligned} \mathrm{P}(x)\,\mathrm{d}x &= |\Psi|^2\,\mathrm{d}x \\ &= \frac{1}{\sqrt{\pi}} e^{-x^2}\,\mathrm{d}x\,. \end{aligned} \tag{2}$$

Hence, the probability of locating the particle at a point x in an arbitrary region $x \in (a,b)$ is

$$\begin{aligned} \mathrm{P}_a^b &= \int_a^b P(x)\,\mathrm{d}x \\ &= \frac{1}{\sqrt{\pi}} \int_a^b e^{-x^2}\,\mathrm{d}x\,. \end{aligned} \tag{3}$$

The integral in (3) does not have analytic solutions but P_a^b can be computed for a given region using numerical integration techniques. Note that $\mathrm{P}_{-\infty}^{\infty} = 1$ as the integrand is a normalised probability density (i.e. the particle must be found at some $x \in \mathbb{R}$).

In this report, we will compute P_0^2, which is trivially less than 0.5 since the distribution in (2) is symmetric. We can write P_0^2 in terms of the error function (see Appendix A):

$$\mathrm{P}_0^2 = \frac{\operatorname{erf}(2)}{2}\,. \tag{4}$$

For validation purposes, we use Wolfram Mathematica to compute the true value $\mathrm{P}_0^2 = 0.497\,661\,132\,509$ accurate to 12 decimal places.

III. NC Integration

In NC integration, the function is evaluated at a finite set of points, not necessarily uniformly distributed, and the integral is approximated as a weighted sum of the evaluations where the weight of each coordinate depends on the NC scheme.

For example, the uniform extended trapezium rule can be expressed as [3]:

$$\int_a^b f(x)\,\mathrm{d}x \approx h\left[\frac{1}{2}f(a) + f(a+h) + ... + f(b-h) + \frac{1}{2}f(b)\right] , \tag{5}$$

where $h = (b-a)/K$ is the step size for K intervals. We can use an iterative approach to perform the trapezium rule integral (Fig. 1).

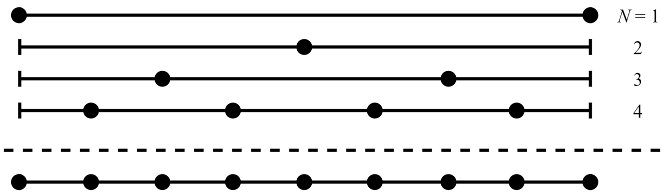

Fig. 1: The line represents the integration region and each dot corresponds to a point where the integrand is evaluated at the N-th iterative step. The final line shows the total integrand evaluations after $N = 4$ iterations (adapted from [3]).

Suppose T_k is a trapezium rule estimate of the integral from the $N = k$ step with step size h_k and let Σ_{k+1} be the sum of the integrand evaluations for the additional points at the $N = k+1$ step, then the trapezium rule approximation for the $N = k+1$ step is (see Appendix B):

$$T_{k+1} = \frac{1}{2}\left(T_k + h_k\Sigma_{k+1}\right) . \tag{6}$$

Similarly, Simpson's rule applies specific weights to the coordinates such that the integral estimate is the area under a second-order Lagrange polynomial interpolated between three successive points (x_i, f_i). It can be shown [3] that the $N = k$ step of the Simpson's rule estimate S_k is related to the trapezium rule estimates T_{k+1} and T_k by

$$S_k = \frac{1}{3}\left(4T_{k+1} - T_k\right) . \tag{7}$$

Therefore, to compute the integral using Simpson's rule, we may use the iterative trapezium rule scheme in (6).

The error in the trapezium rule and Simpson's rule integrals can be estimated by considering the absolute change in the integral value between successive iterative steps. The routine iterates until the desired accuracy is achieved or the number of iterations exceeds some predefined limit. The result of computing P_0^2 to a relative accuracy of at least 10^{-6} using the trapezium rule is shown in Table I and the corresponding result for Simpson's rule is shown in Table II. Note that Simpson's rule requires fewer integrand evaluations than the trapezium rule to achieve the same accuracy since quadratic interpolation is, in general, a better approximation of the integrand than linear interpolation.

Note that the trapezium rule requires more evaluations than Simpson's rule to achieve the same accuracy. The integrand from (3) and the corresponding integrand approximations from the trapezium rule and Simpson's rule are plotted in Fig. 2.

TABLE I: The iterative step, N, and the cumulative integrand evaluations, K, required to compute P_0^2 to a relative accuracy of at least 10^{-6} using the trapezium rule.

N	K	$\mathrm{P}_0^2 \pm$ Error
1	2	0.57
2	3	0.49 ± 0.04
3	5	0.496 ± 0.001
4	9	0.4974 ± 0.0003
5	17	$0.497\,60 \pm 0.000\,08$
6	33	$0.497\,64 \pm 0.000\,02$
7	65	$0.497\,657 \pm 0.000\,005$
8	129	$0.497\,660 \pm 0.000\,001$
9	257	$0.497\,660\,9 \pm 0.000\,000\,3$
10	513	$0.497\,661\,00 \pm 0.000\,000\,07$

TABLE II: The iterative step, N, and the cumulative integrand evaluations, K, required to compute P_0^2 to a relative accuracy of at least 10^{-6} using Simpson's rule. Note that the first step requires three total evaluations since we need to compute both T_1 and T_2 to estimate S_1 using (7).

N	K	$\mathrm{P}_0^2 \pm$ Error
1	3	0.47
2	5	0.49 ± 0.01
3	9	$0.497\,65 \pm 0.000\,07$
4	17	$0.497\,660 \pm 0.000\,004$
5	33	$0.497\,661\,0 \pm 0.000\,000\,3$
6	65	$0.497\,661\,13 \pm 0.000\,000\,02$

A comparison of the number of integrand evaluations required to compute P_0^2 to various accuracies using the trapezium rule and Simpson's rule is shown in Table III and the corresponding log-log plot is shown in Fig. 3. Note that $\log K$ scales linearly with $-\log\epsilon$ so we can write $\log K = A - k\log\epsilon$ for constants A and k. We find $k = 0.50 \pm 0.02$ for the trapezium rule and $k = 0.23 \pm 0.02$ for Simpson's rule so $\log K$ grows twice as fast for the trapezium rule.

TABLE III: Total number of integrand evaluations, K, required to compute P_0^2 to a given accuracy, ϵ, using the trapezium rule and Simpson's rule.

ϵ	K	
[—]	Trapezium Rule	Simpson's Rule
10^{-3}	17	9
10^{-4}	33	17
10^{-5}	129	33
10^{-6}	513	65
10^{-7}	1025	65
10^{-8}	4097	129
10^{-9}	16 385	257

IV. MC Integration

All of the subsequent analysis was completed using a fixed random seed $r = 22092009$. Monte Carlo integration uses random numbers to sample the integrand at various points to compute an expectation for the integral. Explicitly, an integral of the form

$$I = \int_a^b f(x)\,\mathrm{d}x \tag{8}$$

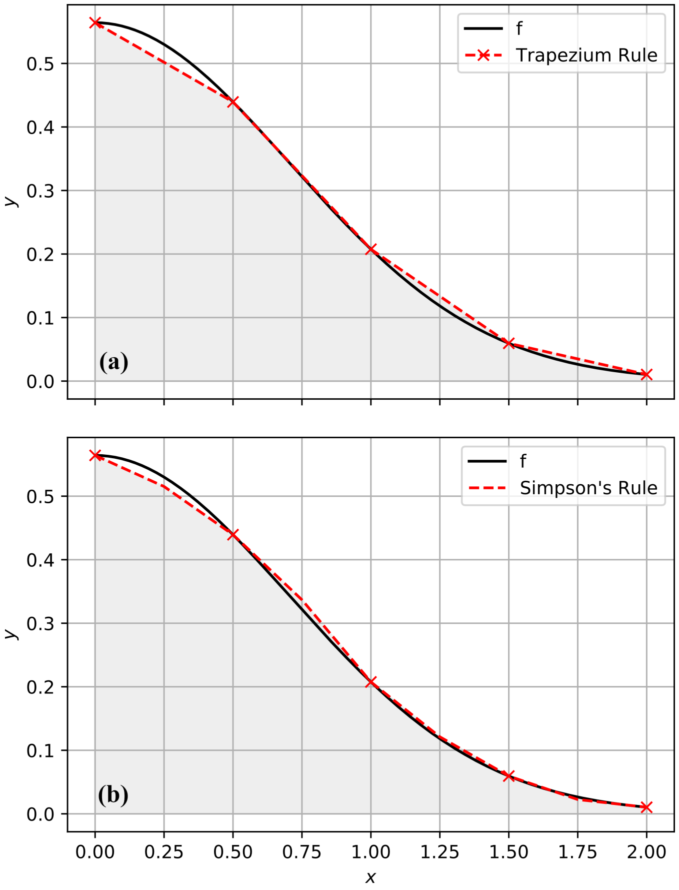

Fig. 2: The variation of the integrand over the integration region $0 \le x \le 2$ is shown as a solid black line. The area of the shaded region represents an estimate of P_0^2 to a relative accuracy of at least 10^{-1} using **(a)** the trapezium rule and **(b)** Simpson's rule. The red crosses represent the points at which the integrand is evaluated and the dashed red line is an approximation to the integrand for each scheme.

can be approximated by a random variate $\hat{I}$ given by

$$\begin{aligned} \hat{I} &= (b-a)\langle f\rangle \\ &= (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i)\,, \end{aligned} \tag{9}$$

where x_i are random variates drawn from a uniform probability distribution on the integration region [3]. In essence, we compute the expectation value $\langle f\rangle$ by taking N random samples x_i and multiply by the width of the region $W = b-a$ to approximate I.

More generally, if the integrand $f(x)$ is strongly weighted to a particular region by some normalised probability density $p(x)$, we can apply *importance sampling*:

$$\begin{aligned} I &= \int_a^b f(x)\,\mathrm{d}x \\ &= \int_a^b \frac{f(x)}{p(x)} p(x)\,\mathrm{d}x \\ &= \langle f/p\rangle_p\,, \end{aligned} \tag{10}$$

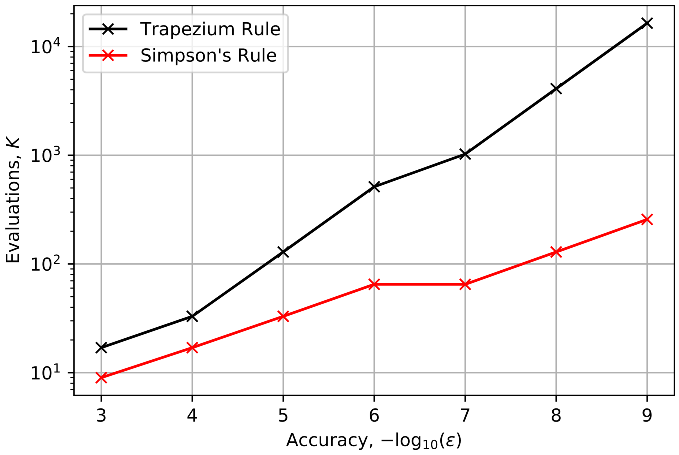

Fig. 3: For a given accuracy, ϵ, the plot shows that the number of integrand evaluations, K, required to compute P_0^2 grows with increasing accuracy.

where the subscript denotes that the expectation is calculated with respective to $p(x)$. Thus, if the random variates x_i are instead drawn with probability $p(x)\,\mathrm{d}x$, the integral estimator becomes

$$\begin{aligned} \hat{I} &= \langle f/p\rangle \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \end{aligned} \tag{11}$$

and the corresponding variance estimate [3] is

$$\begin{aligned} \hat{\sigma}_I &= \frac{\langle f^2/p^2\rangle - \langle f/p\rangle^2}{N} \\ &= \frac{1}{N}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)^2}{p(x_i)^2} - \left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right)^2\right] \end{aligned} \tag{12}$$

For a uniform distribution $p(x) = 1/(b-a)$ defined on $[a,b]$, (11) readily simplifies to (9) as expected.

Importance sampling with an appropriate distribution can significantly reduce the computation time. The ideal choice for $p(x)$ that minimises $\hat{\sigma}_I$ is $p(x) \propto |f(x)|$, but it is usually impractical to sample directly from $f(x)$ [3]. Instead, $p(x)$ is chosen such that it can be easily sampled and so that it reduces the variance in $f(x)/p(x)$, which consequently reduces the error. To compute P_0^2 with importance sampling, we use the linear probability distribution $p(x) = 0.98 - 0.48x$, which is normalised on $[0,2]$, and draw random, independent samples using the transformation method [3].

We consider how the transformation method can be applied to an arbitrary linear distribution. Suppose we wish to draw random samples from a linear probability distribution $p(x) = Cx + D$ which is normalised on the region (a,b). By the transformation method, a random variate drawn from a uniform distribution $u \in [0,1]$ can be mapped to a random variate on the subregion $x \in [a,b]$ by the inverse cumulative distribution function $F^{-1}(u)$ where $F(u)$ is defined as

$$\begin{aligned} F(u) &= \int_{-\infty}^{u} p(x)\,\mathrm{d}x \\ &= \int_{a}^{u} Cx + D\,\mathrm{d}x \\ &= \frac{C}{2}u^2 + Du - \frac{C}{2}a^2 - Da\,. \end{aligned} \tag{13}$$

Hence, the corresponding inverse cumulative distribution function is

$$\begin{aligned} F^{-1}(u) &= -\frac{D}{C} \pm \sqrt{\left(\frac{D}{C}\right)^2 + 2\frac{Da+u}{C} + a^2} \\ &= -\frac{D}{C} + \operatorname{sgn}(C)\sqrt{\left(\frac{D}{C}\right)^2 + 2\frac{Da+u}{C} + a^2}\,, \end{aligned} \tag{14}$$

where we have replaced the plus-minus with the sign operator $\operatorname{sgn}(C) = C/|C|$ since we require $a \le F^{-1}(u) \le b$. Fig. 4 shows a histogram of 10^5 random samples drawn from a uniform and a linear distribution.

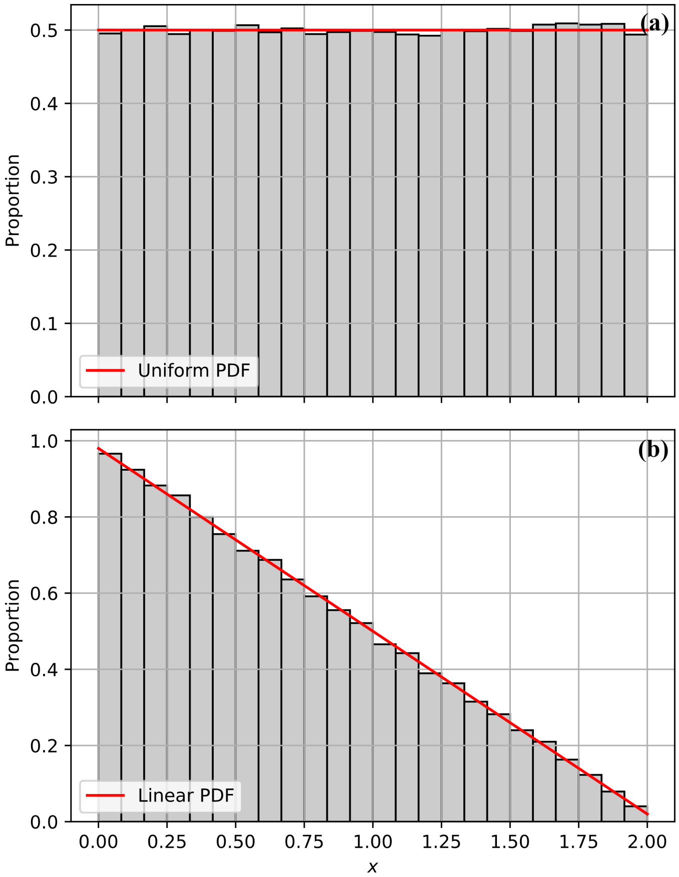

Fig. 4: The transformation method is used to draw 10^5 random samples from **(a)** a uniform distribution, $p(x) = 0.5$, and **(b)** a linear distribution, $p(x) = 0.98 - 0.48x$. Both distributions are normalised on $[0, 2]$ and the histograms serve as validation that the transformation method is yielding the expected results.

For a desired accuracy ϵ, let us compare the number of evaluations required to compute P_0^2 using the two distributions from Fig. 4. The results of this comparison are tabulated in Table IV.

TABLE IV: Total number of integrand evaluations, K, required to compute P_0^2 to a given relative accuracy, ϵ, using Monte Carlo integration sampling from a uniform distribution $p(x) = 0.5$ and a linear distribution $p(x) = 0.98 - 0.48x$.

ϵ	K	
[—]	Uniform PDF	Linear PDF
10^{-1}	78	5
10^{-2}	6091	726
10^{-3}	610 119	73 031
10^{-4}	61 075 761	7 341 744

Similar to Fig. 3, $\log K$ scales linearly with $-\log\epsilon$ so we may write $\log K = A - k\log\epsilon$ for constants A and k. We find $(A, k) = (-0.11 \pm 0.05, 1.97 \pm 0.02)$ for the uniform distribution and $(A, k) = (-1.30 \pm 0.08, 2.05 \pm 0.03)$ for the linear distribution. Based on these values, we can extrapolate the number of integrand evaluations required for higher accuracies, as tabulated in Table V.

TABLE V: Expected number of integrand evaluations, K, required to compute P_0^2 to a relative accuracy, ϵ, using Monte Carlo integration sampling from a uniform distribution $p(x) = 0.5$ and a linear distribution $p(x) = 0.98 - 0.48x$. These values could not be computed due to computation time restrictions, but we can extrapolate from the values in Table IV.

ϵ	K	
[—]	Uniform PDF	Linear PDF
10^{-5}	5.4×10^{9}	8.9×10^{8}
10^{-6}	5.0×10^{11}	1.0×10^{11}
10^{-7}	4.7×10^{13}	1.1×10^{13}
10^{-8}	4.3×10^{15}	1.3×10^{15}

The results in Table V explain why we were not able to achieve a relative accuracy greater than $\epsilon = 10^{-4}$ using Monte Carlo sampling. On a typical modern computer, a sequence of 10^6 random numbers can be generated and the integrand evaluated in $\delta t = (2.96 \pm 0.01)\,\mathrm{s}$. This means we would expect a total computation time of around $T = 40\,\mathrm{min}$ to achieve an accuracy of $\epsilon = 10^{-5}$ and $T = 70\,\mathrm{h}$ for $\epsilon = 10^{-6}$.

Note that the scale parameter k is approximately the same for both distributions so the effect of importance sampling is not related to the number of samples. In fact, the computational advantage of importance sampling becomes less noticeable for larger K, as demonstrated in Fig. 5.

A. *AIS+RSS*

We now introduce an improved MC integration scheme to compute P_0^2 which combines adaptive importance sampling (AIS) with recursive stratified sampling (RSS). Briefly, RSS reduces the variance of the integral estimator by iteratively bisecting the region with the largest variance until each subregion has approximately equal variance. Simple MC integration is performed on each subregion to yield an estimate with

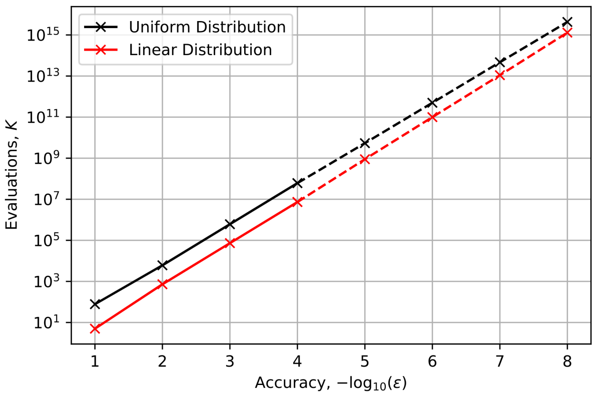

Fig. 5: The variation of the integrand evaluations, K, required to compute P_0^2 to a given relative accuracy, ϵ, using Monte Carlo integration sampling from a uniform distribution $p(x) = 0.5$ (solid black) and a linear distribution $p(x) = 0.98 - 0.48x$ (solid red). The dashed lines represent projected number of evaluations where the integral could not be performed empirically due to computation time restrictions.

a reduced variance. Fig. 6 shows how RSS may split the integration region after 15 iterations.

This method is adaptive (since the subregions are bisected based on the variance of the integrand) and requires fewer integrand evaluations than MC with IS. RSS is also more general and can be extended to many integrands unlike IS with a fixed PDF. It has the disadvantage that all previous evaluations in a subregion are lost when the subregion is bisected, but these losses can be minimised by only sampling a small number of points, say $N = 10$ which is usually sufficient to accurately gauge the variance of the subregion. At the start of each iteration, the list of subregions is searched to find the subregion with the smallest variance, which introduces an additional computation that scales with the number of subregions. In practice, it is unfeasible to have more than 100 000 subregions and reasonable accuracies (i.e. $\epsilon \approx 10^{-6}$ can be easily achieved with a subregion limit of 2000, even without AIS.

To implement AIS in conjunction with RSS, recall that the aim of importance sampling is to define a probability distribution $p(x)$ such that $f(x)/p(x)$ is as close to constant as possible. Thus, we can define an appropriate linear probability distribution on each of the subregions to minimise the variance of the overall estimator. Suppose $p_i(x)$ is the probability distribution for the i-th subregion which has limits $x \in [a_i, b_i]$. We introduce three conditions that $p_i(x)$ should satisfy:

1) $p_i(x)$ should be normalised on the subregion (i.e. $\int_{a_i}^{b_i} p_i(x)\,\mathrm{d}x = 1$).
2) The gradient of $p_i(x)$ should be related to the gradient of the integrand so that sampling in each subregion is weighted by the magnitude of the integrand.
3) $f(x)/p_i(x)$ should be as close to constant as possible over the subregion.

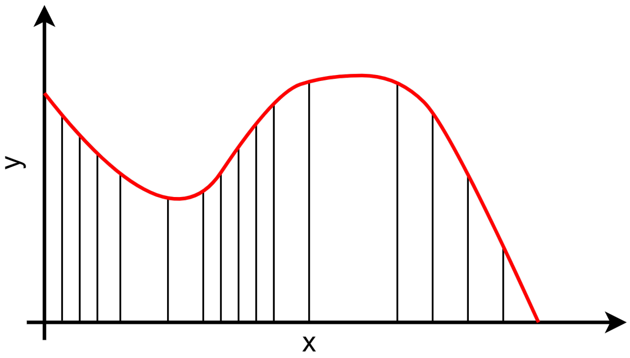

Fig. 6: RSS operates by bisecting subregions based on the variance of the integrand [1]. For some arbitrary integrand $f(x)$ shown in solid red, RSS may output the subregions (bounded by solid black lines) after 15 iterations. Note that the width of the subregions is smaller where $f(x)$ varies rapidly so $f(x)$ can be approximated by a linear function in each subregion.

Let us define the probability distribution on the i-th subregion as $p_i(x) = k_i A_i x + B_i$ where k_i, A_i and B_i are functions of the subregion limits a_i and b_i. Then, we can apply the above conditions to define the unknowns. Specifically, we apply condition 2) and define A_i to be an estimate of the gradient of the integrand:

$$A_i = \frac{f(b_i) - f(a_i)}{b_i - a_i}\,. \tag{15}$$

Similarly, condition 1) gives an equation for B_i in terms of k_i.

For condition 3), we consider a heuristic argument to minimising the variance. Let us define $g_i(x) = f(x)/p_i(x)$ and suppose that the variance of $g_i(x)$ is sufficiently minimised if $g_i(a_i) = g_i(b_i)$, which yields a second equation for B_i in terms of k_i. Mathematically, this will not necessarily minimise the variance of $g_i(x)$, but it prevents $g_i(x)$ from diverging at the endpoints of the subregion which is sufficient for the purposes of MC integration. These final two equations can be simultaneously solved to yield expressions for B_i and k_i:

$$B_i = \frac{1}{b_i - a_i}\left[1 - \frac{b_i + a_i}{b_i - a_i}\,\frac{f(b_i) - f(a_i)}{f(b_i) + f(a_i)}\right] \tag{16}$$

$$k_i = \frac{1}{b_i - a_i}\,\frac{2}{f(b_i) + f(a_i)}\,. \tag{17}$$

Note that the above scheme cannot be applied, as-is, to integrands whose sign changes over the subregion since this would require the gradient of the probability distribution function to change sign within the subregion. This can be readily resolved by manually splitting the integral at the roots of the integrand and performing each integral separately. Alternatively, the integrand can be vertically shifted by a known constant to ensure sign consistency over the integration region and an appropriate adjustment can be applied to the resulting integral estimate. For P_0^2, this is not required since the integrand is positive for all $x \in [0, 2]$.

We verified the validity of the new algorithm by performing integrals with known analytic solutions. In the subsequent analysis, we compare the number of evaluations required to compute P_0^2 to various degrees of accuracy using the new algorithm. These results are summarised in Table VI and Fig. 7.

TABLE VI: Total number of integrand evaluations, K, required to compute P_0^2 to a relative accuracy, ϵ, using Monte Carlo integration (uniform distribution, IS with linear distribution, RSS and AIS+RSS). Integrals that did not converge to the desired accuracy within reasonable computational limits have been omitted.

ϵ	K			
[—]	Uniform	IS	RSS	AIS+RSS
10^{-1}	78	5	20	20
10^{-2}	6091	726	30	20
10^{-3}	610 119	73 031	160	40
10^{-4}	61 075 761	7 341 744	780	100
10^{-5}	[—]	[—]	3500	250
10^{-6}	[—]	[—]	15 740	620
10^{-7}	[—]	[—]	74 480	1580
10^{-8}	[—]	[—]	[—]	4050
10^{-9}	[—]	[—]	[—]	9800

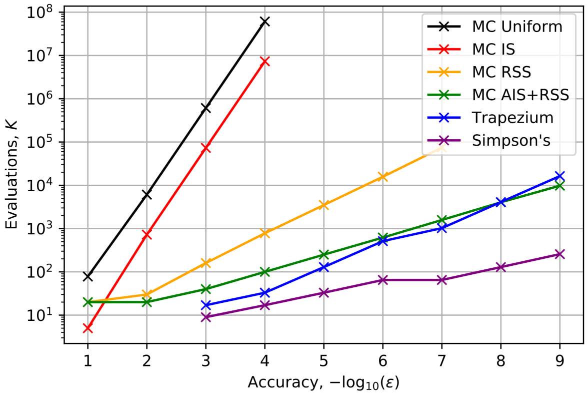

Fig. 7: The variation of the number of integrand evaluations, K, required to compute P_0^2 to a given degree of accuracy, ϵ, with various integration techniques. Methods prefixed with "MC" are Monte Carlo simulation methods, where "uniform" corresponds to Monte Carlo integration with a uniform PDF, "IS" is importance sampling with a linear PDF, "RSS" is recursive stratified sampling with a uniform PDF and "AIS+RSS" is the new adaptive importance sampling with recursive stratified sampling method proposed in this paper.

As expected, in the case of one-dimensional integrals, Newton-Cotes summations are invariably computationally superior to Monte Carlo simulations. However, variance reduction techniques such as importance sampling and recursive stratified sampling can significantly reduce the computational expenses of MC simulations. Ultimately, the use of adaptive algorithms remains the most critical factor in minimising the computational overhead of a Monte Carlo simulation. The AIS+RSS algorithm we have introduced, which combines adaptive importance sampling with recursive stratified sampling, offers results comparable to the trapezium rule for accuracies better than $\epsilon = 10^{-6}$. Indeed, a relative accuracy of $\epsilon = 10^{-9}$ is achieved with fewer integrand evaluations using AIS+RSS than with the trapezium rule, as shown in Fig. 7.

The main advantage of AIS+RSS is that it can be readily generalised to achieve similar efficiencies in higher dimensions, while NC summations suffer from the curse of dimensionality. Briefly, we describe the two adjustments required for AIS+RSS to be applied to multi-dimensional integrals. Firstly, the probability density in each subregion, $p_i(x)$, should now become a separable density $p_i(x, y, ...) = p_i^x(x)p_i^y(y)...$ where each individual term is a linear probability distribution on the subregion as before. Additionally, the RSS algorithm should search subregions in all dimensions and bisect the subregion with highest variance, irrespective of the dimension. The final modification prevents the number of subregions growing exponentially.

V. Summary

We have computed the probability of finding a particle, which is in the ground state of a quantum harmonic oscillator, in the region $x \in [0, 2]$. Mathematically, this is equivalent to numerically integrating a normalised Gaussian function with $(\mu, \sigma) = (0, 1/2)$ on the same region. For integrals in one dimension, we found that Newton-Cotes summations (i.e. trapezium rule, Simpson's rule) are generally the most efficient method for computing the integral. We used Monte Carlo integration methods with several variance reduction techniques, including importance sampling (IS), adaptive IS and recursive stratified sampling (RSS). The AIS+RSS algorithm we have proposed applies IS with linear probability distributions on sufficiently small integration regions achieved from the RSS algorithm, offering a fast and efficient way to compute integrals. The algorithm is computationally superior to the other Monte Carlo techniques tested in this paper and, for accuracies better than $\epsilon = 10^{-6}$, its efficiency is comparable to that of the trapezium rule.

References

[1] U. Siddiqui, "Computational Physics Project 2", Department of Physics, Imperial College London, London, 2019.
[2] A. Fick, "Ueber Diffusion", *Annalen der Physik*, vol. 170, no. 1, pp. 59-86, 1855. [Accessed 8 December 2019].
[3] W. Press, *Numerical Recipes in C*, 2nd ed. Cambridge: Cambridge University Press, 2002.
[4] M. Newmann and G. Barkema, *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1999.
[5] M. Boas, *Mathematical Methods in the Physical Sciences*, 3rd ed. New York: John Wiley & Sons, Inc., 2005.

Appendix A
Error Function

The error function [5] is defined as

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\,\mathrm{d}t\,. \tag{18}$$

Thus, we can write the required integral in terms of the error function:

$$\begin{aligned} \mathrm{P}_0^2 &= \frac{1}{\sqrt{\pi}} \int_0^2 e^{-t^2}\,\mathrm{d}t \\ &= \frac{1}{2}\frac{2}{\sqrt{\pi}} \int_0^2 e^{-t^2}\,\mathrm{d}t \\ &= \frac{1}{2}\operatorname{erf}(2)\,, \end{aligned} \tag{19}$$

as required.

Appendix B
Iterative Trapezium Rule

Suppose we wish to compute an integral defined as

$$I = \int_a^b f(x)\,\mathrm{d}x\,. \tag{20}$$

Let T_k correspond to the trapezium rule estimate from the $N = k$ iterative step with step size h_k such that

$$T_k = \frac{h_k}{2}\left[f(a) + f(b) + 2f(a+h_k) + 2f(a+2h_k) + ... + 2f(b-2h_k) + 2f(b-h_k)\right]\,. \tag{21}$$

Now consider the next iteration of the trapezium rule estimate T_{k+1} which has step size $h_{k+1} = h_k/2$. The corresponding equation for T_{k+1} can be written as

$$\begin{aligned} T_{k+1} &= \frac{h_{k+1}}{2}\left[f(a) + f(b) + 2f(a+h_{k+1}) + 2f(a+2h_{k+1}) + ... + 2f(b-2h_{k+1}) + 2f(b-h_{k+1})\right] \\ &= \frac{h_k}{4}\left[f(a) + f(b) + 2f(a+h_k/2) + 2f(a+h_k) + ... + 2f(b-h_k) + 2f(b-h_k/2)\right]\,. \end{aligned} \tag{22}$$

Then, multiply (22) by 2 and subtract (21) to find

$$\begin{aligned} 2T_{k+1} - T_k &= \frac{h_k}{2}\left[2f(a+h_k/2) + 2f(a+3h_k/2) + ... + 2f(b-3h_k/2) + 2f(b-h_k/2)\right] \\ &= h_k \Sigma_{k+1}\,, \end{aligned} \tag{23}$$

where $\Sigma_{k+1} = f(a+h_{k+1}) + f(a+3h_{k+1}) + ... + f(b-3h_{k+1}) + f(b-h_{k+1})$ is the sum of the additional integrand evaluations at the $N = k+1$ step. We may rearrange (23) to give the desired form in (6).