Computational Methods Assignment 6 - Monte Carlo Integration

Martik Aghajanian

1 Question 1 - Discussion of the Method

In this program, the *Mcint* class takes a specific, potentially multivalued function of N variables to integrate between upper and lower limits, each individually specified by N-dimensional vectors. To integrate the function between the limits using importance sampling the user must supply the function to be integrated, the probability distribution to weight the samples, and an extra function supplied to either (a) generate the distribution of random numbers from a uniform deviate or (b) a function which produces numbers distributed according to the user's desired probability distribution. The uniform deviate generator used for this was the same as for assignment 3.

In this method, the function $f(\vec{x})$ (or $f(\vec{x})/p(\vec{x})$ if importance sampling) is sampled over volume V at M_{init} values, to calculate the average $V\langle f\rangle$, or $\langle f/p\rangle$ for importance sampling (no factor of V. At each step, an additional amount of samples, equal to the number currently included in the average M, are taken, which doubles the number of samples. The old evaluation of the integral is multiplied by M and then the further M samples are added to it, to later divide by $2M = M_{\text{new}}$. As a measure of error, the standard error is used as a statistical metric to determine the point to stop iterating the method. At each step, as well as calculating $\langle f\rangle$ (or $\langle f/p\rangle$), the value of $\langle f^2\rangle$ (or $\langle f^2/p^2\rangle$) is also calculated to obtain the standard error according to:

$$\sigma = \sqrt{\frac{\langle f^2\rangle - \langle f\rangle^2}{M}}$$

This means that the error, as expected, should decrease as $M^{-1/2}$ as the number of samples M is increased. This means that by doubling the samples it is expected that the error should decrease by a factor of $\sqrt{2}$ each time.

2 Question 2 - Uniform vs Importance sampling

The error function was supplied as the integrand and firstly the evaluation of erf(2) was performed using the uniform sampling of points in an integral, later to be calculated using a distribution $\text{pdf}_1(y) = 0.98 - 0.48y$ with a corresponding cummulative distribution of:

$$\text{pdf}_1(y) = -0.24^y + 0.98y = x \implies y = \text{cdf}_1^{-1}(x) = \frac{0.98 - \sqrt{0.98^2 - 0.96x}}{0.48}$$

Also to test the effect of different probability distributions on the convergence of the method, a second probability distribution function was used which bared a similar functional form to the integrand.

$$\text{pdf}_2(y) = \frac{3e^{-1/5y}}{2(1-e^{-3})} \implies y(x) = \text{cdf}_2^{-1}(x) = -\frac{2(1-e^{-1.5x})}{3(1-e^{-3})}$$

The function $f(\vec{x})$ was passed to the generalised integrator class along with the probability distribution function and cummulative distribution. A step constitutes the doubling of the number of samples and calculation of the average and the average square of the integrand. After 22 steps, it became too computationally cumbersome to iterate the method.

The value of erf(2) after 22 steps, the standard error after 22 steps, and the predicted iterations required to obtain an accuracy of $\epsilon = 10^{-6}$ are shown in Table 1. This shows that implementing importance sampling results in a higher convergence of the integral, with a smaller standard error than the value obtained by

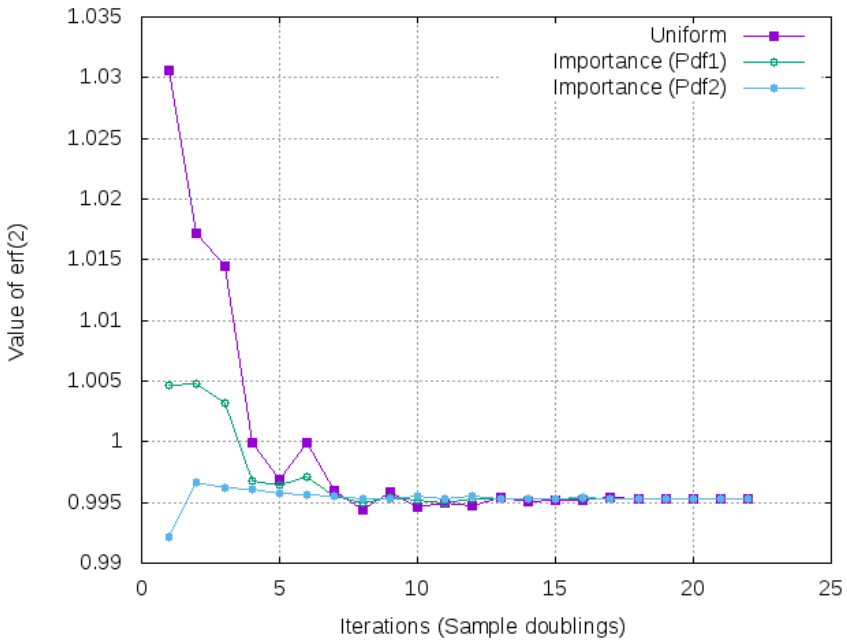

Figure 1: Convergence of values for erf(2) with repeated doubling of the sample size for uniform sampling and importance sampling with both $\text{pdf}_1(x)$ and $\text{pdf}_2(x)$.

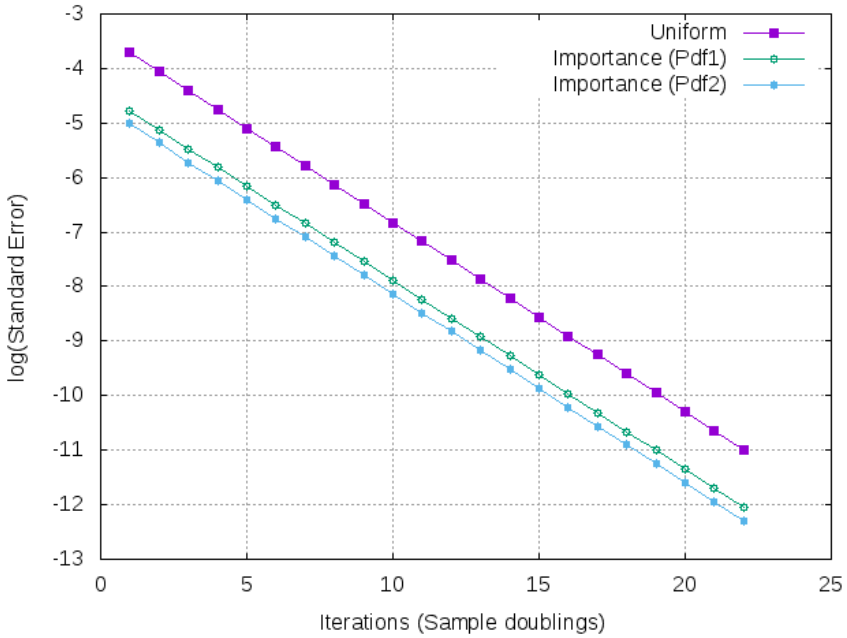

Figure 2: The logarithm of the Standard error of the evaluation of erf(2) using Monte Carlo integration with increasing number of steps in which the sample size doubles each time.

uniform sampling, after 22 steps. The estimated value of the integral and the associated standar error is shown in Fig. 1 and Fig. 2 respectively, against the number of steps/iterations in which the number of samples required is doubled, from an initial number of $M_{\text{init}} = 500$. Firstly, Fig. 1 shows that this method of integration does converge to a value that is in the vicinity of the true value 0.995322 to $\epsilon = 10^{-6}$. It also

	Uniform	Importance (pdf_1)	Importance (pdf_2)
$\text{erf}(x = 2)$	0.995311	0.99532	0.995325
Std. Error	1.7×10^{-5}	5.9×10^{-6}	4.9×10^{-6}
Predicted steps for $\epsilon = 10^{-6}$	31	28	27
Corresponding samples	1.074×10^{12}	1.342×10^{11}	6.71×10^{10}
Value for c	0.7785	0.2701	0.2244
Predicted min. samples for $\epsilon = 10^{-6}$	6.061×10^{11}	7.295×10^{10}	5.036×10^{10}

Table 1: Results from calculation of erf(2) using both uniform sampling and importance sampling, for to different probability distribution functions.

	Samples N	Scaling of ϵ with N
Monte Carlo (Importance sampling pdf_1)	7.295×10^{10}	$\epsilon \sim N^{-1/2}$
Monte Carlo (Importance sampling pdf_2)	5.036×10^{10}	$\epsilon \sim N^{-1/2}$
Runge-Kutta 45 (Adaptive step size)	12	$\epsilon \sim N^{-6}$
Simpson's Rule (Fixed step)	65	$\epsilon \sim N^{-4}$
Trapezium Rule	513	$\epsilon \sim N^{-2}$

Table 2: Comparison of the number of samples required to obtain an accuracy of $\epsilon = 10^{-6}$ for the evaluation of erf(2), additionally showing the expected scaling of ϵ with sample size.

can be seen that as the number of steps increased, the factor that the standard error decreased by converged to $\sqrt{2}$, and this can be seen from the data tables outputted by the program. This meant that, assuming that the error decreases by a factor of $\sqrt{2}$ each step, the number of steps required to reach the convergence criterion of $\epsilon = 10^{-6}$ could be predicted. This was found by taking the current standard error value at 22 steps, finding how many divisions by $\sqrt{2}$ were needed to obtain $\sigma < \epsilon$, and adding that to the number of steps already carried out (22). The number of samples this corresponds to is $M_{\text{start}} \times 2^{n_r}$ where $M_{\text{start}} = 500$ is the initial number of iterations carried out in the first step, and n_r is the number of required iterations for convergence. However, this number of samples required is not necessarily the minimum number, since this method overshoots due to the fact that the number of samples doubles each step. If the error after including N samples is ϵ_N, then the constant of proportionality c (assuming that $\epsilon_N = c \times N^{0.5}$) is $\epsilon_N\sqrt{N}$. To calculate this, the highest number of iterations (22) is used to get c for each method and for this, by converting the number of steps M into number of samples N. Then this c is used to estimate the number of samples needed for eachsampling method to acquire an accuracy of $\epsilon = 10^{-6}$.

This means that the number of samples required for the same accuracy is around 8.3 times higher for the uniform sampling than the importance sampling for the probability distribution function $0.98 - 0.48x$, which is close to 2^3 corresponding to three extra doublings of the sample size. For the second probability distribution, the sample size is 12.12 times lower than the uniform sampling, and around 1.5 times less than for the importance sampled method with the first probability distribution. In comparison to the other methods of integration in previous assignments, including Runge-Kutta 45 Integration (with adaptive step size), the trapezium rule and Simpson's rule, this Monte Carlo method requires a higher number of samples to be taken. These values are compared in Table 2, and shows that the number of samples required for this accuracy is much higher for Monte Carlo than any others.

This is understandable considering the scaling of the error with number of steps for the different methods. If the formula for error goes as $\epsilon = cN^{-n}$ where n is some number indicating scaling whilst c is a constant, and if it is assumed that $c \approx 0.3$ similar to the Monte-Carlo method, then:

$$\begin{aligned}
\text{Runge-Kutta 45 (Adaptive):}\epsilon = cN^{-6} &\implies N = (c/\epsilon)^{1/6} = (0.3 \times 10^6)^{1/6} = 8.18 \sim 10 \\
\text{Simpson's Rule (Fixed step):}\epsilon = cN^{-4} &\implies N = (c/\epsilon)^{1/4} = (0.3 \times 10^6)^{1/4} = 23.4 \sim 25 \\
\text{Trapezium:}\epsilon = cN^{-2} &\implies N = (c/\epsilon)^{1/2} = (0.3 \times 10^6)^{1/2} = 547 \sim 550
\end{aligned}$$

This shows that the order of the number of steps required for evaluating erf(2) to an accuracy of $\epsilon = 10^{-6}$ calculated from the known scaling of the error with sample size, is of the same order as what has been found in the programs written for the corresponding assignments (Since the evaluation is very close to one, the error for the RK45 method is approximately equal to the relative error).

3 Question 3

The interpolating function used in assignment 2 for the data given, produces the image shown in Fig. 3. If the MC integration program was used to integrate this, then the a sampling probability distribution which may be useful for this is of a Gaussian distribution form. This would be centred on 0 and have a full-width-at-half-maximum of approximately 1.6 (to match the width of the function). There are several issues regarding this however, the first being that this Gaussian would now need to be normalised between the maximum and minimum indepedent variable values, requiring the integration of the error function. Another problem is that unless the user supplied a Gaussian random number generator or a set of normally distribued random numbers, the user would have to supply the inverse cummulative distribution function to the method. Luckily, methods are available for both generating normally distributed random numbers and transforming uniform deviates into such. Primarily, the user can supply a generator called the Box-Muller Transform in which, given 2 uniform deviates x_1 and x_2, two independent normally distributed random numbers z_1 and z_2 can be determined through the equations (Ref. Wolfram Mathworld: Box-Muller Transformation)::

$$z_1 = \sqrt{-2\ln x_1}\cos(2\pi x_2) \qquad x_1 = e^{(-z_1^2 - z_2^2)/2}$$

$$z_2 = \sqrt{-2\ln x_2}\sin(2\pi x_1) \qquad x_2 = \frac{1}{2\pi}\tan^{-1}\left(\frac{z_2}{z_1}\right)$$

The Jacobian used to transform between these two coordinates (which for random variables gives the pdf quotient), is the product of two *independent* random normally distributed variables. This would then allow for the interpolant function to be integrated for a faster convergence than if uniform deviates were used. Another method is to use an approximated form of the cummulative distribution function of the Gaussian distribution which is the error function. It is difficult to obtain the inverse of this, however a possibility is for the user to supply this approximation to the program as the inverse error function (Ref. S. Winitzki "A handy approximation ofr the error function", 2008):

$$\mathrm{erf}^{-1}(x) = \mathrm{sign}(x)\sqrt{\sqrt{\left(\frac{2}{a\pi} + \frac{\ln(1-x^2)}{2}\right)^2 - \frac{\ln(1-x^2)}{a}} - \left(\frac{2}{a\pi} + \frac{\ln(1-x^2)}{2}\right)}$$