Random processes and Monte Carlo Simulation

- Applications of Monte Carlo simulations.
- Random number generators.

Random or Stochastic processes are those in which you cannot predict the outcome of the upcoming event from the outcome of the current event. For example:

Random or Stochastic Processes

Random or Stochastic processes are those in which you cannot predict the outcome of the upcoming event from the outcome of the current event. For example:

- Coin toss: the only prediction about the outcome: 50% of the events will end up as tail being up.

Random or Stochastic Processes

Random or Stochastic processes are those in which you cannot predict the outcome of the upcoming event from the outcome of the current event. For example:

- Coin toss: the only prediction about the outcome: 50% of the events will end up as tail being up.
- Dice: In a large number of throws, the probability of getting a given face is $\frac{1}{6}$.

Applications of Monte Carlo simulations

- Stochastic processes.

Applications of Monte Carlo simulations

- Stochastic processes.
- Complex systems (science).

Applications of Monte Carlo simulations

- Stochastic processes.
- Complex systems (science).
- Numerical integration.

Applications of Monte Carlo simulations

- Stochastic processes.
- Complex systems (science).
- Numerical integration.
- Risk management.

Applications of Monte Carlo simulations

- Stochastic processes.
- Complex systems (science).
- Numerical integration.
- Risk management.
- Financial planning.

Applications of Monte Carlo simulations

- Stochastic processes.
- Complex systems (science).
- Numerical integration.
- Risk management.
- Financial planning.
- Cryptography

Applications of Monte Carlo simulations

- Stochastic processes.
- Complex systems (science).
- Numerical integration.
- Risk management.
- Financial planning.
- Cryptography
- ...

How does one do Monte Carlo simulations

- Let the computer throw "the coin" and record the outcome.

How does one do Monte Carlo simulations

- Let the computer throw "the coin" and record the outcome.
- Need a program that generates a variable with random value.

How does one do Monte Carlo simulations

- Let the computer throw "the coin" and record the outcome.
- Need a program that generates a variable with random value.
- Often need a program that generates a random variable with a given probability distribution.

Sources of random numbers:

Random numbers

Sources of random numbers:

- Tables – "A million random digits with 100,000 normal deviates" by RAND.

Random numbers

Sources of random numbers:

- Tables – "A million random digits with 100,000 normal deviates" by RAND.
- Hardware – external sources of random numbers which are generated by from a physical process.

Random numbers

Sources of random numbers:

- Tables – "A million random digits with 100,000 normal deviates" by RAND.
- Hardware – external sources of random numbers which are generated by from a physical process.
- Software – source of pseudo random numbers.

Software – random number generators

- There are NO true random number generators – only pseudo random number generators!!!

Software – random number generators

- There are NO true random number generators – only pseudo random number generators!!!
- This is because computers have only a limited number of bits to represent a number.

Software – random number generators

- There are NO true random number generators – only pseudo random number generators!!!
- This is because computers have only a limited number of bits to represent a number.
- It implies that no matter which pseudo random number generator you use – it will always repeat itself (period of the generator).

Good random number generators

Important issues:

- Randomness.

Good random number generators

Important issues:

- Randomness.
- Distribution of the numbers.

Good random number generators

Important issues:

- Randomness.
- Distribution of the numbers.
- Long period.

Good random number generators

Important issues:

- Randomness.
- Distribution of the numbers.
- Long period.
- Produce the same sequence if started with the same seed.

Good random number generators

Important issues:

- Randomness.
- Distribution of the numbers.
- Long period.
- Produce the same sequence if started with the same seed.
- Fast.

The standard method of generating pseudorandom numbers use modular reduction in congruential relationships:

- Congruential methods.

Two basic techniques for RNG

The standard method of generating pseudorandom numbers use modular reduction in congruential relationships:

- Congruential methods.
- Feedback shift register methods

Linear congruent method for RNG

Generates a pseudo random sequence of numbers $\{x_1, x_2, \dots, x_k\}$ of length M over the interval $[0, M-1]$:

$$x_i = \quad \mod (ax_{i-1} + c, M) = \text{remainder}\left(\frac{ax_{i-1} + c}{M}\right)$$

Linear congruent method for RNG

Generates a pseudo random sequence of numbers $\{x_1, x_2, \ldots, x_k\}$ of length M over the interval $[0, M-1]$:

$$x_i = \mod (ax_{i-1} + c, M) = \text{remainder}\left(\frac{ax_{i-1} + c}{M}\right)$$

- Starting value of x_0 is called "seed"

Linear congruent method for RNG

Generates a pseudo random sequence of numbers $\{x_1, x_2, \ldots, x_k\}$ of length M over the interval $[0, M-1]$:

$$x_i = \mod (ax_{i-1} + c, M) = \text{remainder}\left(\frac{ax_{i-1} + c}{M}\right)$$

- Starting value of x_0 is called "seed"
- Coefficients a and c should be chosen very carefully.

Linear congruent method for RNG

Generates a pseudo random sequence of numbers $\{x_1, x_2, \ldots, x_k\}$ of length M over the interval $[0, M-1]$:

$$x_i = \mod (ax_{i-1} + c, M) = \text{remainder}\left(\frac{ax_{i-1} + c}{M}\right)$$

- Starting value of x_0 is called "seed"
- Coefficients a and c should be chosen very carefully.

Note that

$$\mod (b, M) = b - \text{int}(b/M) * M$$

Example of linear congruent method

$$
\begin{aligned}
x_i =& \mod (a x_{i-1} + c, M) \\
\mod (b, M) =& b - \mathrm{int}(b/M) * M
\end{aligned}
$$

Example of linear congruent method

$$
\begin{aligned}
x_i =& \mod (a x_{i-1} + c, M) \\
\mod (b, M) =& b - \mathrm{int}(b/M) * M
\end{aligned}
$$

$a = 4, c = 1, M = 9, x_1 = 3$ leads to the following sequence:

$$
\begin{aligned}
x_2 =& 4 \\
x_3 =& 8 \\
x_4 =& 6 \\
x_{5-10} =& 7, 2, 0, 1, 5, 3
\end{aligned}
$$

Example of linear congruent method

$$x_i = \mod (ax_{i-1} + c, M)$$
$$\mod (b, M) = b - \text{int}(b/M) * M$$

$a = 4, c = 1, M = 9, x_1 = 3$ leads to the following sequence:

$$x_2 = 4$$
$$x_3 = 8$$
$$x_4 = 6$$
$$x_{5-10} = 7, 2, 0, 1, 5, 3$$

interval:$0 - 8$ i.e. $[0, M - 1]$
Period: 9 i.e. M numbers (then repeat).

Magic numbers for linear congruent method

- M (length of the sequence) is quite large.

Magic numbers for linear congruent method

- M (length of the sequence) is quite large.
- No overflow. (For 32 bit machines $M \leq 2^{32}$).

Magic numbers for linear congruent method

- M (length of the sequence) is quite large.
- No overflow. (For 32 bit machines $M \leq 2^{32}$).
- Good magic numbers for linear congruent method:

$$a = 16,807 \quad c = 0 \quad M = 2,147,483,647$$

$$a = 1,664,525 \quad c = 1,013,904,223 \quad M = 2,147,483,648$$

Magic numbers for linear congruent method

- M (length of the sequence) is quite large.
- No overflow. (For 32 bit machines $M \leq 2^{32}$).
- Good magic numbers for linear congruent method:

$$a = 16,807 \quad c = 0 \quad M = 2,147,483,647$$

$$a = 1,664,525 \quad c = 1,013,904,223 \quad M = 2,147,483,648$$

- For $c = 0$ called "Multiplicative congruential generator".

Random numbers on interval $[A, B]$

- Scale results from x_i on $[0, M-1]$ to y_i on $[0, 1]$.

$$y_i = \frac{x_i}{M-1}$$

Random numbers on interval $[A, B]$

- Scale results from x_i on $[0, M-1]$ to y_i on $[0, 1]$.

$$y_i = \frac{x_i}{M-1}$$

- Scale results from y_i on $[0, 1]$ to z_i on $[A, B]$

$$z_i = A + (B-A) * y_i$$

Feedback shift register generator

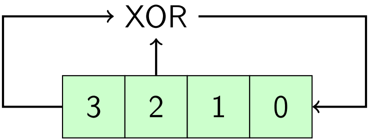
Simple shift register where the vacated bit is filled with the exclusive-or of two other bits in the shift register.

Feedback shift register generator

Simple shift register where the vacated bit is filled with the exclusive-or of two other bits in the shift register.
4 bit shift-register pseudorandom number generator:

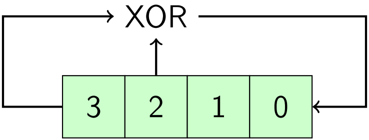
Feedback shift register generator

Simple shift register where the vacated bit is filled with the exclusive-or of two other bits in the shift register.
4 bit shift-register pseudorandom number generator:



Feedback shift register generator

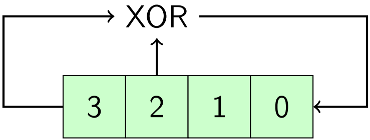
Simple shift register where the vacated bit is filled with the exclusive-or of two other bits in the shift register.
4 bit shift-register pseudorandom number generator:



- Bits 3 and 2 are combined by exclusive-or.

Feedback shift register generator

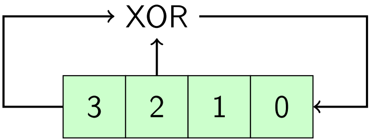
Simple shift register where the vacated bit is filled with the exclusive-or of two other bits in the shift register.
4 bit shift-register pseudorandom number generator:



- Bits 3 and 2 are combined by exclusive-or.
- The register is shifted 1 step to the left.

Feedback shift register generator

Simple shift register where the vacated bit is filled with the exclusive-or of two other bits in the shift register.
4 bit shift-register pseudorandom number generator:



- Bits 3 and 2 are combined by exclusive-or.
- The register is shifted 1 step to the left.
- The result of the exclusive-or is entered into bit 0.

4bit shift register PRNG

Here is the pattern of bits, starting with 0001:

0001
0010
0100
1001
0011
0110
1101
1010
0101
1011
0111
1111
1110
1100
1000
0001

Commonly used PRNG

- Most commonly used is Mersenne Twister (which is a generalized feedback shift register method).

Commonly used PRNG

- Most commonly used is Mersenne Twister (which is a generalized feedback shift register method).
- The commonly used version of Mersenne Twister, MT19937.

Commonly used PRNG

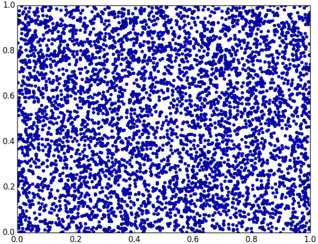
- Most commonly used is Mersenne Twister (which is a generalized feedback shift register method).
- The commonly used version of Mersenne Twister, MT19937.
- It has a period of $2^{19937} - 1$.

Commonly used PRNG

- Most commonly used is Mersenne Twister (which is a generalized feedback shift register method).
- The commonly used version of Mersenne Twister, MT19937.
- It has a period of $2^{19937} - 1$.
- Implemented in numpy.

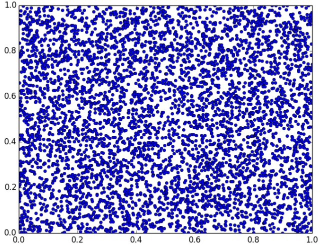
How do we check the RNG?

- 2D plot, x_i and y_i from two random sequences (parking lot test).



How do we check the RNG?

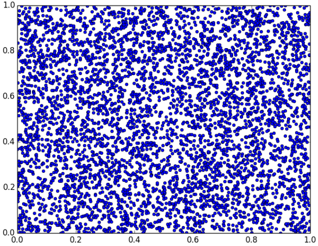
- 2D plot, x_i and y_i from two random sequences (parking lot test).



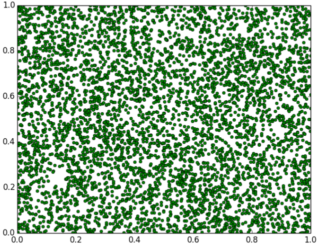
- Plot 3D figure (x_i, y_i, z_i)

How do we check the RNG?

- 2D plot, x_i and y_i from two random sequences (parking lot test).



- Plot 3D figure (x_i, y_i, z_i)
- Plot correlation (x_i, x_{i+k})



How can we check the RNG?

Examples of other assessments:

- Uniformity – A random sequence should contain numbers distributed in the unit interval with equal probability.

How can we check the RNG?

Examples of other assessments:

- Uniformity – A random sequence should contain numbers distributed in the unit interval with equal probability.
- k-th moment:

$$< x^k >= \frac{1}{N} \sum_{i=1}^{N} x_i^k = \frac{1}{k+1}$$

How can we check the RNG?

Examples of other assessments:

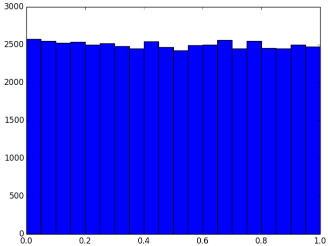
- Uniformity – A random sequence should contain numbers distributed in the unit interval with equal probability.
- k-th moment:

$$< x^k >= \frac{1}{N} \sum_{i=1}^{N} x_i^k = \frac{1}{k+1}$$

- Near neighbour correlation:

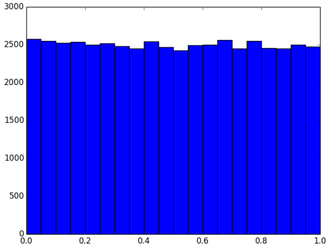
$$\frac{1}{N} \sum_{i=1}^{N} x_i x_{i+k} \approx \frac{1}{4}$$

- Uniformity (50000 random numbers)



Examples of other assessments

- Uniformity (50000 random numbers)

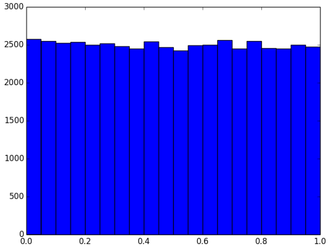


- 4th moment: (50000 random numbers)

$$< x^4 >= 0.1988$$

Examples of other assessments

- Uniformity (50000 random numbers)



- 4th moment: (50000 random numbers)

$$< x^4 >= 0.1988$$

- near neighbor correlation: (50000 random numbers) $= 0.2478$

Test suites for RNG

Good test suites exist – TestU01 – which can be used to uncover problems in random number generators.

Dont try to invent your own random number generator – unless you know what you are doing. This is very tricky business!!!

- Monte Carlo integration.

Monte Carlo integration – basic idea

- Normally when we are interested in some physical phenomenon that has some random element, we write down an exact, non-random description that gives the answer for the average behaviour.

Monte Carlo integration – basic idea

- Normally when we are interested in some physical phenomenon that has some random element, we write down an exact, non-random description that gives the answer for the average behaviour.
- In principle, we can reverse the argument: we can start with an exact problem – such as the calculation of an integral – and find an approximate solution to it by running a suitable random process on the computer!

Monte Carlo integration – basic idea

- Normally when we are interested in some physical phenomenon that has some random element, we write down an exact, non-random description that gives the answer for the average behaviour.
- In principle, we can reverse the argument: we can start with an exact problem – such as the calculation of an integral – and find an approximate solution to it by running a suitable random process on the computer!
- This leads to novel ways of performing integrals..

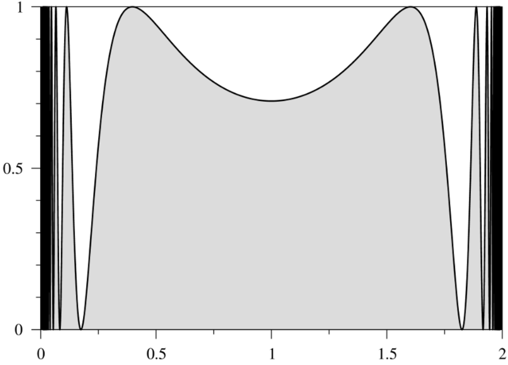
General idea

Suppose we want to evaluate the integral:

$$I = \int_0^2 \sin^2\left[\frac{1}{x(2-x)}\right]dx$$

Suppose we want to evaluate the integral:

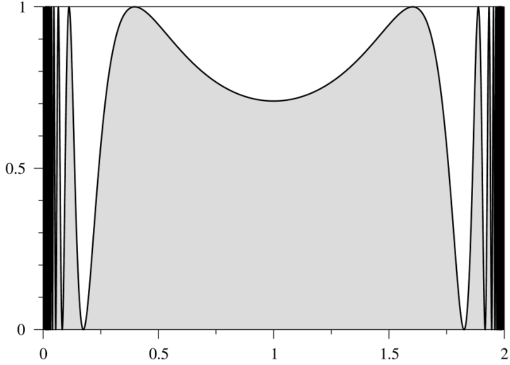
$$I = \int_0^2 \sin^2 \left[\frac{1}{x(2-x)}\right] dx$$



General idea

Suppose we want to evaluate the integral:

$$I = \int_0^2 \sin^2\left[\frac{1}{x(2-x)}\right] dx$$



It is perfectly well behaved in the middle of its range – but varies infinitely fast at the edges.

General idea

- On the other hand since the entire function fits in a rectangle of size 2×1, the integral – the shaded area under the curve – is finite and less than 2.

- On the other hand since the entire function fits in a rectangle of size 2×1, the integral – the shaded area under the curve – is finite and less than 2.
- Methods such as trapezoidal rule or Simpson's rule or Gaussian quadrature are not likely to work well as they will not capture the infinitely fast variation of the function at the edges.

- On the other hand since the entire function fits in a rectangle of size 2×1, the integral – the shaded area under the curve – is finite and less than 2.
- Methods such as trapezoidal rule or Simpson's rule or Gaussian quadrature are not likely to work well as they will not capture the infinitely fast variation of the function at the edges.
- Monte carlo integration offers a simple way to tackle this integral.

General idea

- The shaded area under the curve is I and given that the area of the rectangle is $A = 2$.

General idea

- The shaded area under the curve is I and given that the area of the rectangle is $A = 2$.
- If we choose a point randomly in the rectangle, the probability that the point falls under the curve rather than over it is $p = I/A$.

General idea

- The shaded area under the curve is I and given that the area of the rectangle is $A = 2$.
- If we choose a point randomly in the rectangle, the probability that the point falls under the curve rather than over it is $p = I/A$.
- We generate a large number, N, of random points in the bounding rectangle and check each one to see if it is below the curve and keep a count of the number that are – k.

General idea

- The shaded area under the curve is I and given that the area of the rectangle is $A = 2$.
- If we choose a point randomly in the rectangle, the probability that the point falls under the curve rather than over it is $p = I/A$.
- We generate a large number, N, of random points in the bounding rectangle and check each one to see if it is below the curve and keep a count of the number that are – k.
- Then the fraction of points below the curve is k/N. This should be equal to the probabilitly, $p = I/A$

$$I \simeq \frac{kA}{N}$$

Example

$$I = \int_0^2 \sin^2 \left[\frac{1}{x(2-x)}\right] dx$$

N	I
10^4	1.4542
10^5	1.45252
10^6	1.452492
10^7	1.4513378
10^8	1.45123546

Error Analysis

- For simple integrals – monte carlo methods are not as accurate as trapezoidal rule or Simpson's rule!

Error Analysis

- For simple integrals – monte carlo methods are not as accurate as trapezoidal rule or Simpson's rule!

Probability that a single random point falls below the curve is $p = I/A$ and that it falls above the curve is $(1 - p)$.

- For simple integrals – monte carlo methods are not as accurate as trapezoidal rule or Simpson's rule!

Probability that a single random point falls below the curve is $p = I/A$ and that it falls above the curve is $(1-p)$.

Probability that a particular k points fall below the curve and $(N-k)$ fall above the curve is $p^k(1-p)^{N-k}$.

Error Analysis

- For simple integrals – monte carlo methods are not as accurate as trapezoidal rule or Simpson's rule!

Probability that a single random point falls below the curve is $p = I/A$ and that it falls above the curve is $(1-p)$.

Probability that a particular k points fall below the curve and $(N-k)$ fall above the curve is $p^k(1-p)^{N-k}$.

But there are $\binom{N}{k}$ ways of choosing k points from a list of N.

Error Analysis

- For simple integrals – monte carlo methods are not as accurate as trapezoidal rule or Simpson's rule!

Probability that a single random point falls below the curve is $p = I/A$ and that it falls above the curve is $(1 - p)$.

Probability that a particular k points fall below the curve and $(N - k)$ fall above the curve is $p^k(1 - p)^{N-k}$.

But there are $\binom{N}{k}$ ways of choosing k points from a list of N.

Total probability that we get k points below:

$$P(k) = \binom{N}{k} p^k (1 - p)^{N-k}$$

Binomial distribution!

Mean of this distribution:

$$
\begin{aligned}
< k > &= \sum_{k=0}^{N} kP(k) \\
&= \sum_{k=1}^{N} k \binom{N}{k} p^k (1-p)^{N-k} \\
&= Np \sum_{k=1}^{N} \binom{N-1}{k-1} p^{k-1} (1-p)^{(N-1)-(k-1)}
\end{aligned}
$$

$$\text{Subsitute} \quad j = k-1 \quad \& \quad M = N-1$$

$$
\begin{aligned}
&= Np \sum_{j=0}^{M} \binom{M}{j} p^j (1-p)^{M-j} \\
&= Np
\end{aligned}
$$

$< k^2 >$ of this distribution:

$$< k(k-1) > = \sum_{k=0}^{N} k(k-1)P(k)$$

$$= \sum_{k=2}^{N} k(k-1)\binom{N}{k} p^k (1-p)^{N-k}$$

$$= N(N-1)p^2 \sum_{k=2}^{N} \binom{N-2}{k-2} p^{k-2}(1-p)^{(N-2)-(k-2)}$$

$$\text{Subsitute} \quad j = k-2 \quad \& \quad M = N-2$$

$$= N(N-1)p^2 \sum_{j=0}^{M} \binom{M}{j} p^j (1-p)^{M-j}$$

$$= N(N-1)p^2$$

$$< k^2 > = < k(k-1) > + < k > = N(N-1)p^2 + Np$$

Variance of this distribution:

$$\mathrm{var} k = Np(1-p) = N\frac{I}{A}\left(1 - \frac{I}{A}\right)$$

Variance of this distribution:

$$\mathrm{var} k = Np(1-p) = N\frac{I}{A}\left(1 - \frac{I}{A}\right)$$

Expected error in the integral:

$$\sigma = \sqrt{\mathrm{var} k}\frac{A}{N} = \frac{\sqrt{I(A-I)}}{\sqrt{N}}$$

Variance of this distribution:

$$\text{var}k = Np(1-p) = N\frac{I}{A}\left(1 - \frac{I}{A}\right)$$

Expected error in the integral:

$$\sigma = \sqrt{\text{var}k}\frac{A}{N} = \frac{\sqrt{I(A-I)}}{\sqrt{N}}$$

The error varies with N as $N^{-1/2}$ which means the accuracy improves as we increase N.

Variance of this distribution:

$$\text{var}k = Np(1-p) = N\frac{I}{A}\left(1 - \frac{I}{A}\right)$$

Expected error in the integral:

$$\sigma = \sqrt{\text{var}k}\frac{A}{N} = \frac{\sqrt{I(A-I)}}{\sqrt{N}}$$

The error varies with N as $N^{-1/2}$ which means the accuracy improves as we increase N.

Error in Trapezoidal rule went as $\mathcal{O}(h^2) \sim \frac{1}{N^2}$ and in Simpson's rule as $\mathcal{O}(h^4) \sim \frac{1}{N^4}$ – clearly showing that when we can use the regular methods – we should use them. This method is only good for pathological integrands.

Mean value method

- There are better ways of evalutating this integral:

$$I = \int_a^b f(x)dx$$

Mean value method

- There are better ways of evalutating this integral:

$$I = \int_a^b f(x)dx$$

- Average value $\langle f \rangle$ in range a to b is:

$$\langle f \rangle = \frac{1}{b-a}\int_a^b f(x)dx = \frac{I}{b-a}$$

Mean value method

- There are better ways of evalutating this integral:

$$I = \int_a^b f(x)dx$$

- Average value $\langle f \rangle$ in range a to b is:

$$\langle f \rangle = \frac{1}{b-a}\int_a^b f(x)dx = \frac{I}{b-a}$$

- A simple way to estimate $\langle f \rangle$ is to just measure $f(x)$ at N points, $x_1, x_2, \ldots, x_N$ chosen uniformly at random between a and b:

$$\langle f \rangle = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$I \simeq \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

Error Analysis

- Variance of the sum on N independent random numbers is equal to N times the variance of a single one.

Error Analysis

- Variance of the sum on N independent random numbers is equal to N times the variance of a single one.
- Random numbers in this case are the values $f(x_i)$ and we can estimate the variance of a single one of them $\mathrm{var} f = \langle f^2 \rangle - \langle f \rangle^2$ with:

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \quad \langle f^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} [f(x_i)]^2$$

- Variance of the sum on N independent random numbers is equal to N times the variance of a single one.
- Random numbers in this case are the values $f(x_i)$ and we can estimate the variance of a single one of them $\mathrm{var} f = \langle f^2 \rangle - \langle f \rangle^2$ with:

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \quad \langle f^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} [f(x_i)]^2$$

- Variance on the sum is N times the variance on a single term or $N \mathrm{var} f$.

- Variance of the sum on N independent random numbers is equal to N times the variance of a single one.
- Random numbers in this case are the values $f(x_i)$ and we can estimate the variance of a single one of them $\mathrm{var} f = \langle f^2 \rangle - \langle f \rangle^2$ with:

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \quad \langle f^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} [f(x_i)]^2$$

- Variance on the sum is N times the variance on a single term or $N \mathrm{var} f$.
- Error/standard deviation on the integral:

$$\sigma = \frac{b-a}{N} \sqrt{N \mathrm{var} f} = (b-a) \frac{\sqrt{\mathrm{var} f}}{\sqrt{N}}$$

which goes as $1/\sqrt{N}$ but the variance is smaller!

Multidimensional integrals

- The error analysis we did for Monte Carlo integration remains the same irrespective of the number of dimensions!!!

Multidimensional integrals

- The error analysis we did for Monte Carlo integration remains the same irrespective of the number of dimensions!!!
- If we have N points then deterministic methods will get $N^{1/d}$ points in each dimension. As a result the overall error in midpoint rule would be $N^{-1/d}$ where as in trapeziodal $N^{-2/d}$.

Multidimensional integrals

- The error analysis we did for Monte Carlo integration remains the same irrespective of the number of dimensions!!!
- If we have N points then deterministic methods will get $N^{1/d}$ points in each dimension. As a result the overall error in midpoint rule would be $N^{-1/d}$ where as in trapeziodal $N^{-2/d}$.
- For higher dimensions – more than 4/5, Monte carlo method becomes faster than any of the deterministic methods!

Importance Sampling

- Monte carlo integration is good for integrating pathological functions but sometimes it does not work very well.

- Monte carlo integration is good for integrating pathological functions but sometimes it does not work very well.
- In particular, if the function to be integrated contains a divergence!

Importance Sampling

- Monte carlo integration is good for integrating pathological functions but sometimes it does not work very well.
- In particular, if the function to be integrated contains a divergence!
- This is because occasionally when the random point is near the divergence, you will get a big change in the sum..

Importance Sampling

- Monte carlo integration is good for integrating pathological functions but sometimes it does not work very well.
- In particular, if the function to be integrated contains a divergence!
- This is because occasionally when the random point is near the divergence, you will get a big change in the sum..
- The variance, σ in such cases is very high.

- Monte carlo integration is good for integrating pathological functions but sometimes it does not work very well.
- In particular, if the function to be integrated contains a divergence!
- This is because occasionally when the random point is near the divergence, you will get a big change in the sum..
- The variance, σ in such cases is very high.
- Importance sampling is a way to get around this problem.

Importance Sampling

- For any general function, $g(x)$, we can define a weighted average over the interval from a to b:

$$\langle g \rangle_w = \frac{\int_a^b w(x) g(x) dx}{\int_a^b w(x) dx}$$

where $w(x)$ is any function we choose.

Importance Sampling

- For any general function, $g(x)$, we can define a weighted average over the interval from a to b:

$$\langle g \rangle_w = \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx}$$

  where $w(x)$ is any function we choose.
- Consider the integral:

$$I = \int_a^b f(x)dx$$

Importance Sampling

- For any general function, $g(x)$, we can define a weighted average over the interval from a to b:

$$\langle g \rangle_w = \frac{\int_a^b w(x) g(x) dx}{\int_a^b w(x) dx}$$

where $w(x)$ is any function we choose.

- Consider the integral:

$$I = \int_a^b f(x) dx$$

- Setting $g(x) = f(x)/w(x)$ we have:

$$\left\langle \frac{f(x)}{w(x)} \right\rangle_w = \frac{\int_a^b w(x) f(x)/w(x) dx}{\int_a^b w(x) dx} = \frac{I}{\int_a^b w(x) dx}$$

$$I = \left\langle \frac{f(x)}{w(x)} \right\rangle_w \int_a^b w(x) dx$$

Importance Sampling

- This is similar to the mean value method but allows us to calculate the integral from a weighted average rather than a standard uniform average.

Importance Sampling

- This is similar to the mean value method but allows us to calculate the integral from a weighted average rather than a standard uniform average.
- Let us define a probability density function:

$$p(x) = \frac{w(x)}{\int_a^b w(x)dx}$$

Importance Sampling

- This is similar to the mean value method but allows us to calculate the integral from a weighted average rather than a standard uniform average.
- Let us define a probability density function:

$$p(x) = \frac{w(x)}{\int_a^b w(x)dx}$$

- Let us sample N random points, x_i, non-uniformly with this density. That is the probability of generating a value in the interval between x and $x + dx$ will be $p(x)dx$.

Importance Sampling

- This is similar to the mean value method but allows us to calculate the integral from a weighted average rather than a standard uniform average.
- Let us define a probability density function:

$$p(x) = \frac{w(x)}{\int_a^b w(x)dx}$$

- Let us sample N random points, x_i, non-uniformly with this density. That is the probability of generating a value in the interval between x and $x + dx$ will be $p(x)dx$.
- Then the average number of sample that fall in this interval are $Np(x)dx$ and so for any function $g(x)$:

$$\sum_{i=1}^{N} g(x_i) \simeq \int_a^b Np(x)g(x)dx$$

Importance Sampling

- So using this the general weighted average is given as:

$$\langle g \rangle_w = \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx} = \int_a^b p(x)g(x)dx \simeq \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

Importance Sampling

- So using this the general weighted average is given as:

$$\langle g \rangle_w = \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx} = \int_a^b p(x)g(x)dx \simeq \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

- Putting it all together for our integral:

$$I \simeq \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)} \int_a^b w(x)dx$$

Importance Sampling

- So using this the general weighted average is given as:

$$\langle g \rangle_w = \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx} = \int_a^b p(x)g(x)dx \simeq \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

- Putting it all together for our integral:

$$I \simeq \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{w(x_i)}\int_a^b w(x)dx$$

- The formula allows us to estimate I by calculating not the sum $\sum_{i=1}^{N} f(x_i)$, but instead the modified sum $\sum_{i=1}^{N} f(x_i)/w(x_i)$ where $w(x)$ is any function we choose.

Importance Sampling

- So using this the general weighted average is given as:

$$\langle g \rangle_w = \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx} = \int_a^b p(x)g(x)dx \simeq \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

- Putting it all together for our integral:

$$I \simeq \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w(x_i)} \int_a^b w(x)dx$$

- The formula allows us to estimate I by calculating not the sum $\sum_{i=1}^{N} f(x_i)$, but instead the modified sum $\sum_{i=1}^{N} f(x_i)/w(x_i)$ where $w(x)$ is any function we choose.
- This is useful because it allows us to choose a $w(x)$ that can get rid of the pathologies of $f(x)$.

- So using this the general weighted average is given as:

$$\langle g \rangle_w = \frac{\int_a^b w(x)g(x)dx}{\int_a^b w(x)dx} = \int_a^b p(x)g(x)dx \simeq \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

- Putting it all together for our integral:

$$I \simeq \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{w(x_i)}\int_a^b w(x)dx$$

- The formula allows us to estimate I by calculating not the sum $\sum_{i=1}^{N} f(x_i)$, but instead the modified sum $\sum_{i=1}^{N} f(x_i)/w(x_i)$ where $w(x)$ is any function we choose.
- This is useful because it allows us to choose a $w(x)$ that can get rid of the pathologies of $f(x)$.
- The price we pay is that we have to draw our samples from a non-uniform distribution rather than a uniform distribution.

Error Analysis

$$\sigma = \frac{\sqrt{\mathrm{var}_w(f/w)}}{\sqrt{N}} \int_a^b w(x)dx$$

where

$$\mathrm{var}_w g = \langle g^2 \rangle_w - \langle g \rangle_w^2$$