Numerical methods in Fortran: Lec. III

Devendra Singh Bhakuni

dbhakuni@ictp.it

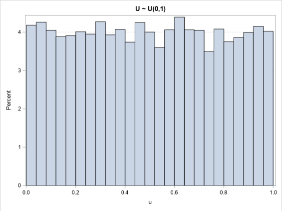
/afs/ictp/public/d/dbhakuni/NM_course/NM_lec3.pdf

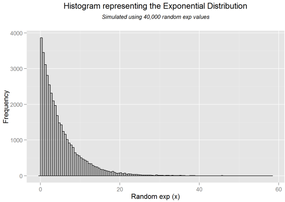
October 25, 2023

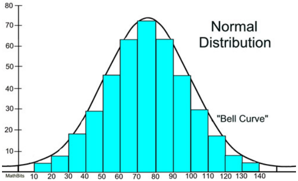
Recap..

- Pseudo Random numbers generators
  - Middle square method - “Negative numbers - Change integer type”
  - Linear congruent method
  - In-built FORTRAN functions: **rand()**, **random_number()**.

- Non-uniform random number generators: Transformation method - Calculate cumulative density function, invert it, pass uniform random numbers $u \in [0,1]$ to it
  - Exponential distribution
  - Gaussian distribution (Box-Muller method)







Topics to be covered

- Random numbers generators, middle square method, linear congruent method, in-built FORTRAN random number generators.

- Non-uniform random number generators: Transformation method, exponential and Gaussian random number generator.

- *Rejection method and Monte-Carlo methods.*

- Non parametric density estimation.

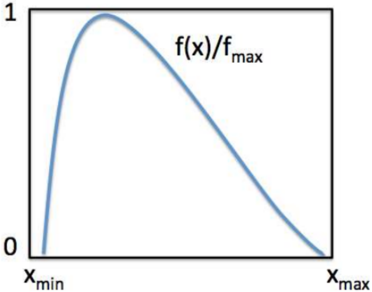
Rejection method

- If the inverse of the integral cannot be found.
- Less efficient (because of rejections).
- Works only if the function has a finite bound.

1) generate a random number x uniformly distributed in $[x_{\min}, x_{\max}]$
2) generate another r.n., call it r, in $[0, 1]$
3) If: $r < f(x)/f_{\max}$; accept x, otherwise: reject and go to step 1.

$f_{\max} = \max(f(x))$

- It works because the probability of acceptance of the generated random numbers is proportional to the PDF.
- We need the interval and the maximum value of the PDF in this interval.



Algorithm

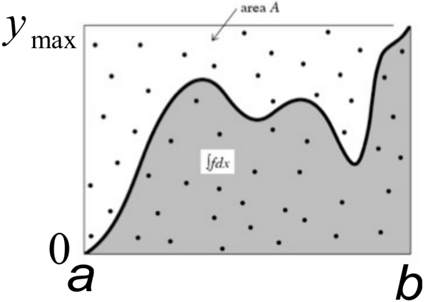
Data: Uniform random number x in the range $[0, 1]$
Data: Uniform random number r in the range $[0, 1]$
Data: Target distribution with PDF $f(x)$
Result: Random number X from the target distribution

repeat
  Generate a random sample x from the uniform distrinution $[0, 1]$;
  Generate a uniform random number r in the range $[0, 1]$;
  Calculate the acceptance probability $A = \frac{f(x)}{f_{\max}}$;
  if $r \leq A$ **then**
    Accept $X = x$;
  end
until *acceptance*;

• Rejected points do not count when generating random numbers, so if you need N random numbers you need N ACCEPTED points (use DO loop)

Integrating using random numbers: "Hit-or-Miss"

- We are interested in solving the integration: $I_{\text{curve}} = \int_a^b f(x)\, dx$



- Consider a square box around the curve.
- Area of the box: $A_{\text{box}} = y_{\text{max}}(b-a)$.
- Generate N uniformly distributed points $\{x_i, y_i\}$ in $[a, b]$ and $[0, y_{\text{max}}]$.
- $x_i = a + u_1(b-a)$, $y_i = u_2 y_{\text{max}}$ (u_1 and u_2 are uniform random numbers in $[0,1]$).
- Count N_0, the number of points which fall below the curve: $y_i < f(x_i)$.

$$\frac{I_{\text{curve}}}{A_{\text{box}}} \approx \frac{N_0}{N}$$

$$I_{\text{curve}} \approx \frac{N_0}{N} y_{\text{max}}(b-a)$$

Integrating using random numbers: "Hit-or-Miss"

- We are interested in solving the integration: $I_{\text{curve}} = \int_a^b \, f(x) \, dx$

Algorithm

Data: Integration limits a, b, function $f(x)$, number of samples N
Result: Approximate integral value A

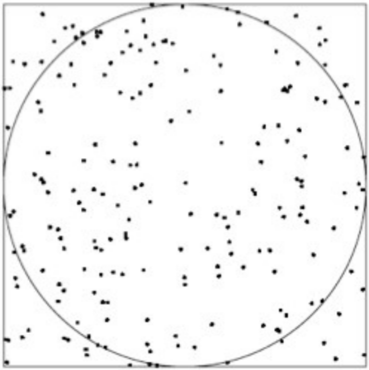
$A \leftarrow 0$;
for $i = 1$ *to* N **do**
    Generate a random number x between a and b;
    Generate a random number y between 0 and M, where M is an upper bound of $f(x)$;
    if $y < f(x)$ **then**
        $A \leftarrow A + 1$;
    end
end
$A \leftarrow \frac{A}{N} \cdot (b - a) \cdot M$;
return A;

Estimating the value of π

We can use the "Hit-or-Miss" Monte-Carlo method to estimate the value of π.

- N_c: Number of points inside the circle.
- N_{tot}: Total number of points.
- A_{circ}: Area of the circle (with diameter $d = 2$)
- A: Area of the square (side $a = 2$).

$$\frac{\pi}{4} = \frac{A_{\text{circ}}}{A} \approx \frac{N_c}{N_{\text{tot}}}$$



Integration using "Crude Monte-Carlo method"

- From the lectures on integration methods (for e.g. Left Riemann)

$$I = \int_0^1 f(x)\, dx \approx \sum_{i=1}^{N} f(x_i) \cdot h$$

Here $h = 1/N$, $x_i = a + (i-1)h$. **Note: x_i are deterministic!**

- Let's use random numbers to solve this integration. The average of a function $f(x)$ for a probability density function $p(x)$ is

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

For a uniform PDF, i.e. $p(x) = 1$ for $x \in [0, 1]$,

$$\langle f \rangle = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \implies \langle f \rangle = \int_0^1 f(x)\, dx$$

x_i's are random numbers from uniform distribution!

Integration using "Crude Monte-Carlo method"

$$I = \int_a^b f(x)\, dx = \frac{1}{N}\sum_{i=1}^{N} f(x_i)\cdot(b-a)$$

Algorithm

Data: Number of samples N, function $f(x)$, integration limits a and b

Result: Approximate integral value A

$A \leftarrow 0$;

for $i \leftarrow 1$ *to* N **do**

- Generate a random number X_i uniformly distributed in $[a, b]$;
- Calculate $f(X_i)$;
- $A \leftarrow A + f(X_i)$;

end

$A \leftarrow \frac{1}{N}\sum_{i=1}^{N} f(X_i)\cdot(b-a)$;

return A;

Can be used in higher dimensions as well. $X_i \rightarrow \vec{\mathbf{X}}$, and $(b-a) \rightarrow V$

Estimating Error in Crude Monte Carlo

- Let's calculate the variance σ^2:

$$\sigma_f^2 = \langle (f - \langle f \rangle)^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - \langle f \rangle)^2 \, p(x_i) = \left(\langle f^2 \rangle - \langle f \rangle^2 \right)$$

"measure to which extent f deviates from its average"

- Consider the previous result for a fixed N as one measurement. Suppose we recalculate the average/variance for M different measurements. Then, we can write the integral as the average of M such averages: $I_M = \frac{1}{M} \sum_{l=1}^{M} \langle f \rangle_l$
- The variance of these measurements will be

$$\sigma_M^2 = \left(\langle I_M^2 \rangle - \langle I_M \rangle^2 \right) \qquad \text{and} \qquad \sigma_M \sim \frac{1}{\sqrt{N}}$$

This is true for any dimension d!

Estimating Error in Crude Monte Carlo

• Let's compare it to any numerical integration based on Taylor expansion, e.g., trapezoidal rule. The error goes as $\mathcal{O}(h^k)$, $k = 2$ for trapezoidal rule and $h = (b - a)/N$. That means, the error goes as $\sim N^{-k}$.

• Consider integration in higher dimension: Suppose integration volume is a hypercube with side L and dimension d. The number of points in the cube: $N = (L/h)^d \implies$ the error goes as $\sim N^{-k/d}$

• If we perform the same integration using MC method, the error $\sigma \sim 1/\sqrt{N}$. Thus, for dimension $d > 2k$, MC method has better accuracy than the traditional numerical integration methods.

Assignment-14

1) Generates 10^4 points in the interval $[0, 1]$ from the distribution $f(x) = 3x^2$ using the rejection method and plot the histogram.

2) Estimate the value of π using "Hit-or-Miss" method. Plot it as a function of the total number of points $N \in [10^2 - 10^4]$. Use the dimensions for square $[-1, 1] \times [-1, 1]$.

3) Estimate the integral: $\int_0^1 e^{-u^2} du$ using crude Monte-Carlo method using the number of samples $N = 10^5$.

4) Estimate the value of π using "crude Monte-Carlo" method by integrating the function: $F(x, y) = \begin{cases} 1 & \text{if } x^2 + y^2 \leq 1 \\ 0 & \text{otherwise} \end{cases}$, in the domain $\Omega = [-1, 1] \times [-1, 1]$ and with $N = 10^4$.

Hint: $\int_\Omega F(x, y) dx dy = \pi$. Generate $2N$ uniform random numbers $u, v \in [0, 1]$.

Submit your code as ***surname_assignment_14.f90*** to ***dbhakuni@ictp.it*** before the next lesson.