Numerieke Natuurkunde 1: Monte Carlo Method

Q1: Random number generation

1. Write a function which generates random numbers according to a flat distribution between -5 and 5. Try a few times and calculate the mean and standard deviation. Plot a histogram of the random numbers.

2. Write a function which generates random numbers according to a Gaussian distribution with a mean value of 0 and a standard deviation of 1. Use the hit&miss method. Check that the function really peaks at around 0 and has a width of 1. Plot a histogram of the random numbers.

Hint: `srand` in the `stdlib.h` is a random number generator which takes a seed value as argument for the initialization. Random numbers between 0 and `RAND_MAX` can be obtained by calling `rand()`. Here is a simple example how to generate random numbers between 0 and 1.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main()
{
  // initialize random number generator with system time
  srand(time(NULL));

  // generate a random number between 0 and 1
  double r = (double) rand() / (double) RAND_MAX;

  // print random number
  printf("my random number: %f\n", r);

  // return no error
  return 0;
}
```

Q2: Numerical integration

1. Calculate π by generating pairs of random numbers (x, y) inside a 2-D box with $-1 \leq x\,y \leq 1$. Count the number of times a point lies within a circle with radius 1. How does that relate to the ratio of areas of the circle and the box? Hint: be careful with factors of 2.

2. Calculate the integral of a Gaussian distribution with a mean value of $\mu = 0$ and a standard deviation of $\sigma = 1$ in the limits of -1 and 1, i.e.,

$$\int_{-1}^{1} dx\,\mathrm{Gauss}(x|\mu,\sigma)\,.$$

   What would you expect knowing the answer from your favorit statistics book?

3. Draw a graph of the intergral as a function of the number of points, N. Verify that the relative uncertainty drops as $1/\sqrt{N}$.

Q3: Random walk

1. Write a function which randomly moves a particle from one point to another (in 2-D space) using

   - a fixed step size of $r = 1$, or
   - a random step size between 0 and 1.

   Visualize the trajectory of one particle.

2. Let $n = 100$ particles move for $m = 10$ iteration. Calculate the average distance to the starting point of (0, 0). Calculate the standard deviation of distance. Increase the number of steps to $m = 1000$.

3. Generate a simulation for particles inside a box. Modify the function from 1 in such a way that particles cannot leave the box but are deflected. Chose a random position inside the box. Visualize the trajectory of one particle in a box. Repeat study 2.

4. Repeat 3 but with the box being split into two part (left and right). Start with all particles in the left part. Calculate the number of particles in the left and right side as a function of iteration. Calculate the relaxation time of the system. Play with the number of particles and the number of steps.

5. Repeat the simulation and calculate the time it takes for the first particle to arrive at the right side. (BONUS)

6. Think about a strategy on how to include interactions between the particles. You can find one solution in the 'Maxwell' exercise in your script. (BONUS)

Q4: Error propagation (BONUS)

1. Assume you are measuring an angle θ with an uncertainty of 1°. Calculate the uncertainty on $\cos\theta$ for $\theta = 90^\circ, 45^\circ, 10^\circ$, and 0° using

   (a) Gaussian error propagation, and

   (b) a Monte Carlo method, i.e., generate Gaussian numbers and calculate $\cos\theta$ for each angle. The standard deviation is a measure of the uncertainty.

   Compare the two results. What is going wrong?

2. Assume you are measuring two quantities, x and y, with (Gaussian) uncertainties of $\sigma_x = 4$ and $\sigma_y = 2$, respectively. Calculate the uncertainty on the ratio $r = x/y$, σ_r, using

   (a) Gaussian error propagation, and

   (b) a Monte Carlo method, i.e., generate Gaussian numbers for x and y and calculate $r = x/y$ for each pair. The standard deviation of r is a measure of the uncertainty.

   Compare the two results. What is the shape of the distribution of r? Compare it to your assumptions when performing error propagation.