Exercise sheet 2

07/11/2017

Exercise 1
In this exercise we study the method of direct sampling as it is applied by children in the Monte Carlo beach. So while on the beach kids play the following game. They draw a square and a circle inside the square. Then they throw pebbles randomly in the square. Each pebble inside the circle is considered as a hit. Compute the ratio of the area of the circle with respect to the area of the square with edges (-1, 1), (1, -1), (1, 1) (-1, 1).

- To what is this ratio equal?
- Create a program that performs this calculation.
  Hint: your program should compute $4N_h/N_t$ where N_h is the number of hits and N_t is the number of trials
  Hint2: you might want to draw uniform random numbers (in the correct interval) and set a criterion that is inspired from your background in analytical geometry. You might also find the following command handy: (float)rand()/RAND_MAX$*2-1$ and check here for more details on how to initialize and seed a pseudo-random number generator
  `http://en.cppreference.com/w/c/numeric/random/srand`
- Repeat this experiment to reach an accuracy of your satisfaction

Exercise 2
Integrate a function $f(x)$ in the interval $[a, b]$ employing N random numbers.
Hint: in your program $f(x)$ should be coded up as a separate function

- Integrate a simple polynomial like $x^3 + 5x - 3$ from $[0, 1]$ and compare to the exact result
  Hint: the following transformation might be handy for you $x = a + (b - a)u$, how would you approximate the new integral as a sum?
  Hint2: as a general strategy you should use double precision numbers in order to avoid running into trouble with rounding errors
  Hint3: the use of the ranlux random number generator by M. Lüscher is recommended. You can download it from `http://luscher.web.cern.ch/luscher/ranlux/index.html`. Then don't forget that you should have the following line in your code #include "ranlux-3.3/ranlxd.h". And you should link it appropriately when compiling cc -O2 ../ranlxd.c myintegral.c -lm -o myintegral
- Change the amount of random numbers appropriately to study the convergence of the algorithm
  Hint: you might want to use the standard deviation as a convergence criterion
- Study the scaling with respect to N

Exercise 3
In this exercise we will understand the need and advantages of importance sampling in practice. Use your code of the previous problem in order to integrate, from 0 to ∞ the following function

$$g(x) = (c\sqrt{x} + d)\frac{a}{x^2 + a^2}$$

Hint: you can approximate the upper integration limit by 10^4

- Use $c = 1.5$ and $d = 0.5$
- The main goal is to study the $a \to 0$ limit of the previous integral. In order to do so you can start from $a = 1$ and monitor the number of random numbers N needed in order to obtain a given accuracy (we discussed how to quantify "accuracy" in the previous problem)

- Use $p(x) = \frac{a}{a^2+x^2}$ as a probability distribution for the new random variable u and cast the integral in the following form

$$\int_0^\infty dx g(x) = \int_0^\infty dx c\sqrt{x} + d)p(x) \propto \int_0^1 du(c\sqrt{x(u)} + d) \propto \frac{1}{N}\sum_{i=1}^{N}(c\sqrt{x_i} + d)$$

  Hint: u should be a random number in the interval $[0, 1]$

- Compare the amount of random numbers needed for a sought accuracy (and given value of a) with respect to the previous algorithm. What do you observe?

Exercise 4

In this problem we want to study the Ising model. The exercise will revolve around the simulation of the $2-$dim Ising model but one can always use the $1-$ dim Ising model as a warm-up exercise.

The $2-$dim Ising model has a very interesting and rich structure. The zero magnetic field model has a second order phase transition for a finite value of the temperature and one can compute critical exponents and various observables and study the continuum limit of the model. Despite its complicated nature, the model is simple enough to possess an analytical solution.

Consider a lattice with N sites labeled by the index i and N_l links among the nearest neighbors. Since two sites are connected by one link and 4 links intersect at one site we have $4N_l = 4N$. The Hamiltonian has the usual form of

$$\mathcal{H} = -J\sum_{<ij>} s_i s_j$$

and for $J > 0$ the system is ferromagnetic (we will only focus on this case for simplicity). You are invited to search the relevant literature in order to see what happens when $J < 0$. A link connecting same spins has energy $-J$. The difference between the two spin configurations is $2J$. The minimum energy of the system is naturally the energy of the ground system, $E_0 = -JN_l - BN = -2(2J + B)N$.

- How many terms do we have in the sum of the partition function? What is the number of configurations?

- When $B = 0$ the system has some extra symmetry. Which one is this?

- What are the consequences of the above symmetry for the ground state of the Hamiltonian?

- Define properly the data structure of your program. Define the appropriate boundary conditions and map correctly your lattice to the memory of the computer. Usually one uses periodic boundary conditions. Each spin is stored in an array $s[L][L]$. The site $s[i][j]$ has four neighbors $s[i \pm 1][j]$ and $s[i][j \pm 1]$. Periodic boundary conditions are easily implemented by adding $\pm L$ to i, j when they become $> L$ or < 1. The elements of this array s are stored linearly in the memory of the computer. This means that $s[i][j]$ is $i * L + j$ array positions away from $[0][0]$. Notice the multiplication involved. One can use helical boundary conditions to avoid that. Search for this option in the literature if you want to do so.

- What about the initial conditions? The choice of initial conditions can affect how fast thermalization will be achieved. Some standard options are the case of cold start with $\beta = \infty$ and all spins aligned, or the case of hot start $\beta = 0$ and spins equal to $\pm 1/2$ with probability $1/2$. Remember, to always check for thermalization and for ergodicity.

- Can you envision a choice of wrong initial conditions that will never lead to thermalization?

- Measure the energy $E = -\sum_{<ij>} s_i s_j$ and the magnetization $M = |\sum_i s_i|$. Why do we use an absolute value symbol here?
  The energy difference between two states μ and ν which differ by a single spin flip $s_k^\nu \to -s_k^\mu$ with all other spins the same is given by

$$E_\nu - E_\mu = 2s_k^\mu\left(\sum_{<ik>} s_i^\mu\right)$$

(show it).
Locality is a crucial property of this relation, since it depends only on nearest neighbors. Remember the Metropolis algorithm when do we accept and when do we reject?
The possible values of $\sum_{ik} s_i^\mu = -4, -2, 0, 2, 4$. These are the only values entering the term $A(\mu \to \nu)$. How many of these values do we need to consider? Store them into an array and do not compute repeatedly $e^{\beta(E_\mu - E_\nu)}$.
Hint: Remember to test your code at each stage of its development.