Tutorial

1: Simple and important sampling

Kai Grass*

April 19, 2007
SimBio group, FIAS, Frankfurt

Contents

*grass@fias.uni-frankfurt.de

1 Introduction

Welcome to the tutorial to the course Simulation Techniques for Soft Matter Sciences. In order to get a deeper understanding of the methods described in the lecture, we provide you with some hands-on tasks and further informations on the subject matter.

1.1 Structure

Each tutorial is divided into different section that focus on aspects from corresponding lecture course. Each section is accompanied by tasks that are to be worked on during the lab. These tasks are important for future tutorials and thus should be understood. Use the opportunity to ask questions if you have problems.

Following the Task section, you usually find a section enlisting complementary tasks. These questions will give you further insight and should be worked on till next tutorial, where they will be discussed briefly.

1.2 Account distribution

We set up accounts for you that are valid till the end of the semester. If you have your own account to access the local computer system, you can use them as well.

2 GNU Scientific Library (GSL)

The GNU Scientific Library (GSL) is a collection of routines for numerical computing. The routines have been written from scratch in C, and present a modern Applications Programming Interface (API) for C programmers, allowing wrappers to be written for very high level languages.

2.1 Advantages of using GSL

The GSL is open source, i.e. the source code is distributed under the GNU public license. As such, the use of this library is free of charge.

The library covers a wide range of topics in numerical computing. Routines are available for numerous areas of mathematicsthe following areas, such as Complex Numbers, Sorting, Random Numbers, Differential Equations, Minimization, . . .

The use of these routines is described in the manual. Each chapter provides detailed definitions of the functions, followed by example programs and references to the articles on which the algorithms are based.

In addition to the broad range of functions, the most important advantage of the using GSL is that the functions are already implemented (saves you work) and checked by many people (you can rely on the correctness of the implementation).

2.2 Tasks

1. Look at the simple program `gsl-demo.c` and try to understand how to generate uniformly distributed random numbers with the aid of the GSL.

Remark: All source code of this tutorial is accompanied by a Makefile that helps you compiling it. Simply type `make` in the source folder.

2.3 Supplementary tasks

1. A nice feature of the GSL random number generator implementation is, that the method to generate random numbers can be chosen at run time. See the documentation for details on this.

Remark: You can check this by starting the demo with the following line:
`GSL_RNG_TYPE=taus GSL_RNG_SEED=123 ./gsl-demo`.

3 Calculating Π

Consider a circle of diameter d surrounded by a square of length l ($l > d$). Random coordinates within the square are generated. The value of can be calculated from the fraction of points that fall within the circle.

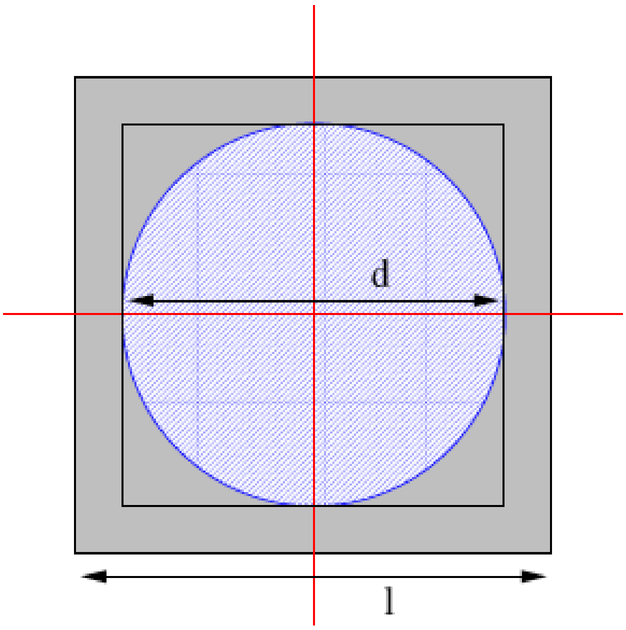

Figure 1: A circle of diameter d surrounded by a square of length l ($l > d$).

3.1 Tasks

1. How can Π be calculated from the fraction of points that fall in the circle?
2. Complete the small Monte Carlo program to calculate Π using this method.
3. How does the accuracy of the result depend on the ratio l/d and the number of generated coordinates?

3.2 Supplementary tasks

1. Derive a formula to calculate the relative standard deviation of the estimate of Π.
2. Is it a good idea to calculate many decimals of Π using this method?.

4 Numerical integration

Definition: Monte Carlo is the art of approximating an expectation by the sample mean of a function of simulated random variables.

4.1 Simple sampling

The MC method can be used to numerically integrate a function $f(x)$.

$$F(a,b) = \int\limits_a^b f(x)dx = (b-a)E(f(x)),$$

where $E(f(x))$ is the expectation value of $f(x)$ on the interval $[a,b]$. This expectation value can be calculated via Monte Carlo method.

$$E(f(x)) = \int\limits_{x \in \chi} f_\chi(x)dx$$

Here x is drawn from a uniform distribution χ. For a finite number of samples we get the *Monte Carlo estimate* $\tilde{f(x)}$

$$\tilde{f}_n(x) = \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

4.2 Task

1. How can the method of simple sampling be applied to calculate the value of an integral?

4.3 Supplementary tasks

1. Write a program, that estimates the value of the normal distribution in the interval [-50,50] using simple sampling.
2. Comment on the accuracy of this method to determine the integral. How is it depending on n (the number of random samples)?
3. Is a uniform sampling a good choice to integrate such a function?

Hints: The Normal distribution is a Gaussian with zero mean and variance of 1.

$$N(x) = \frac{1}{\sqrt{(2\pi)}} \exp(-\frac{x^2}{2})$$

The integral over the normal distribution can be calculated analytically. The value for the interval [-50,50] is very close to 1.

5 Suggested Resources

5.1 GNU Scientific Library (GSL)

- http://www.gnu.org/software/gsl/manual/html_node/

5.2 Monte Carlo techniques

- http://ib.berkeley.edu/labs/slatkin/eriq/classes/guest_lect/mc_lecture_notes.pdf