APiE Exercise – Random Numbers

The basic questions are equivalent to 0.5 EC, while the voluntary questions sum up to another 0.5 EC so that you can earn 1 EC in total.

Exercise 1 – RNG (2pts.)

Implement a small Random Number Generator (RNG) yourself.

$$x_{i+1} = (c * x_i)_{modp} \tag{1}$$

with start "Seed" $x_0 \neq 0$.

Advice: Define a function `rng_init(seed)` that initializes the RNG and a second function `rng_irand()` that returns the integer random number(s), and another function `rng_rand()` that returns the random real number(s) between 0 and 1. Adapt your rng to the question used and solve problems related to speed (optimize the creation) and memory (in case there are too many numbers necessary). How many random numbers can you keep in memory (M=?); but how many random numbers do you really need to keep in memory?

While doing this, check for the smallest and the largest (integer) number the RNG returns and also check the period for:
a) $p = 2^{15} - 1$ and $c = 171$, and 65539
b) $p = 2^{31} - 1$ and $c = 171$, and 65539
What are the values?

From now on, in all following exercises, use your self-made simple RNG and a matlab built-in. For the test-exercises also use a really "bad" RNG in order to see and understand what is going wrong. Comment for all exercises on the results/performance of either, in case there is something to say.

Exercise 2 – Tests (3pts.)

Choose a seed number.

Perform the square test with your RNG – and with the built-in MATLAB random number generator (which one did you choose?). Perform the following tests for your RNG and the MATLAB RNG.

Compute the average of the random numbers for total numbers $M = 1$, 10, ... 10^7. How high can you go?

Compute the standard-deviaton of the random numbers for $M = 1$, 10, ... 10^7 and scale them by the mean.

What is the expected mean and standard deviation and how close did you get?

(Voluntary) Compute the standard deviation of the random numbers for $M = 1, 10, \ldots. 10^7$, e.g. using the matlab function, and evaluate shape and quality of the RNG. Do the same for a bad RNG and comment on the result.

(Voluntary (2pt.): (self) study and use the χ^2-test.)

Use the MATLAB Fourier-transform in order to analyse a series of 1024 random numbers (using a bad, short period RNG). Explain, discuss the result.

Exercise 3 – Random Walk (3pts.)

Implement a simple 1-dimensional random walk (RW) starting at $x_0 = 0$, with $\Delta t = 1$ and $\Delta x = 1$. Generalize it to a vector (ensemble) of many RWs and compute $\langle x \rangle$ and $\langle x^2 \rangle$. Plot both as function of time for two RNGs used for at least 2^{20} (time) steps.

What is the diffusion coefficient of this simple random walk expressed in Δt and Δx?

Plot the probability distribution of positions after 2^{20} steps and compare it (fit it) with a Gaussian function. Discuss the relation between the width of the Gaussian and the distribution.

(Voluntary 3pts.) Write down (find) the partial differential equation that is solved by this Gaussian function. Solve the partial differential equation (pde) with another numerical method (pde-solver) and with the random walk and compare the models, parameters, performance.

Exercise 4 – Integration (2pts.)

Compute π using your RNG and the MATLAB RNG. (From integration of a quarter circle, by choosing random points and counting them if they are inside the circle-section.)

How many random numbers do you need to get π with 2,3,4,5,6 digits accurate?

Do the same using a threee-dimensional integral of a sphere. Is there a difference in performance?

Exercise 5 – Fractal (Voluntary 5pts.)

Implement the Random Midpoint Displacement Fractal, based on a square-lattice (or triangle if you like) and iterate until level 8. Display the result with the MATLAB surface function and choose a nice color-scale.