KFU Graz
Dénes Sexty (denes.sexty@uni-graz.at)
CP1: Monte Carlo Methods WS 2023/24

4. Exercise: Testing your random numbers 1

For a uniform distribution between 0 and 1 one should have:

$$\langle x^k \rangle = \int_0^1 x^k dx = \frac{1}{k+1} \tag{1}$$

For two independent uniform random numbers x and x' we should have

$$C_{k,k'} = \langle x^k x'^{k'} \rangle = \int_0^1 dx \int_0^1 dx' x^k x'^{k'} = \frac{1}{k+1}\frac{1}{k'+1} \tag{2}$$

Test whether the random numbers supplied by your language satisfy these formulas within errors.
Hint: Write a subroutine that returns the mean and standard error of the mean (assuming the numbers are independent), given a list of numbers. Now make a list of x^k from random numbers x, and use the above subroutine. For the second part try it such that x' is always given directly after x by your PRNG.

5. Exercise: Test also the following random generators:

- $X_{n+1} = 1277 X_n \bmod 2^{17}$
- $X_{n+1} = 16807 X_n \bmod 2^{31} - 1$
- $X_{n+1} = 48271 X_n \bmod 2^{31} - 1$
- **Middle-square method:** take an n digit number (with even n), square it and take the middle digits from the resulting $2n$ digit number (add zeroes at the beginning if needed) as the new number.

  ```
  newnumber=int(str(number*number).zfill(2*n)[n/2:2*n-n/2])
  ```

  or in binary:

  ```
  newnumber= int((bin(number*number)[2:].zfill(2*n)[n/2:2*n-n/2]),2)
  ```

- **Linear-feedback shift register:** The state is an n-bit binary number: $b_1 b_2 \cdots b_n$. To get a new random bit, take b_n. The state is modified by this procedure: first calculate a new random bit $b_{\text{new}} = b_{t_1}$ XOR b_{t_2} XOR $\cdots b_{t_k}$, where the t_i are some given integers. Than $b'_{i+1} = b_i$ (this is the shift), and $b'_0 = b_{\text{new}}$.

  To get a random floating point number between [0:1] you take e.g. a 32 bit integer using random bits and divide it by 2^{32}.

  This is implemented in `LFSR.py` which you can download from the course's moodle page.

6. Exercise: Generating random numbers with Maxwell-Boltzmann Distribution

The probability density of the Maxwell-Boltzmann distribution is given by

$$P(x) = \sqrt{\frac{2}{\pi}} \frac{x^2 e^{-x^2/(2a^2)}}{a^3}, \qquad x > 0 \tag{3}$$

Generate random numbers $x_i \in [0, 10]$ with this distribution (take $a = 1$) using the rejection method starting from a uniform distribution. What C is needed? What is the rejection rate? Calculate the mean,variance and skewness with errors and verify they agree with the exact results.

$$\mu = 2a\sqrt{\frac{2}{\pi}}, \qquad \sigma^2 = \frac{a^2(3\pi - 8)}{\pi}, \qquad \gamma_1 = \frac{2\sqrt{2}(16 - 5\pi)}{(3\pi - 8)^{3/2}} \tag{4}$$

The skewness if defined as

$$\gamma_1 = \frac{\langle (x - \mu)^3 \rangle}{\langle (x - \mu)^2 \rangle^{3/2}}, \qquad \text{with } \mu = \langle x \rangle. \tag{5}$$

To calculate errors for variance (skeweness) generate e.g. 50 samples with a sample size of 10^6, and calculate the variance (skewness) for all of them, giving you a 50 element sample of the variance (skewness). Now you can calculate the errors as usual (for example using your subroutine from Exercise 4).
Try again by starting from Gauss distributed random numbers with mean=2, sigma=1, $C = 2.1$. What is the new rejection rate?

7. Exercise: Bonus 1: Testing your random numbers 2

Study the Marsaglia effect: take a dimension d and an integer L and fill a lattice of L^d integers with zeroes. Take random points on the lattice by drawing d random integers between 1 and L. Tag the points which have already appeared by writing 1 into the lattice at the coordinates given by your random integers.
Plot the number of points still haveing 0 as a function of the number of random points used. What function does it follow? (This is easier if one only looks at a corner of the L^d space: e.g. for $d = 2$ only keep track of points if $x, y < 0.025 * L$, and take the LCG: $X_{n+1} = 16807 X_n \bmod 2^{31} - 1$.)