Dr. Ole Klein
M.Sc. Robin Görmer, M.Sc. Philipp Thiele

Lima, Oct. 24, 2019

Spring School on the Introduction on Numerical Modelling of Differential Equations – Programming Exercise 2

Exercise 2.1

The exponential function $\exp(x)$ can be seen as a power series for $x \in \mathbb{R}$. The recursive formula for the computation reads

$$
\begin{aligned}
y_1 &:= x, & f_1 &:= 1 + y_1, \\
y_n &:= \frac{x}{n} y_{n-1}, & f_n &:= f_{n-1} + y_n.
\end{aligned}
$$

Write a program `powerseries.cc` which computes the approximation of $\exp(x)$ for given point x and iteration count n. The datatype for x should be variable (i.e. `double` or `float`). User input shall be over the command line. Please ask for x, n and the datatype.

(a) Test your program with $x = 5$ and $x = -10$ for 100 iterations and different datatypes. Print the difference between the exact value of `std::exp` and your approximation f_n

$$e_n := |\exp(x) - f_n|$$

(b) especially for values $x \ll 0$ the result is off by many orders of magnitude. Use the properties of the exponential function and change the algorithm to get a smaller error. Test this with $x = -20$ and `float`.

Hints:

- The exact value of $\exp(x)$ can be approximated by `long double`:
  `long double exact = std::exp(x);`
- $\exp(-x) = \exp(x)^{-1}$ might be helpful

Exercise 2.2

Consider the tent map

$$f\colon [0,1] \to [0,1], \quad x \mapsto \begin{cases} 2x, & \text{for } x \in [0, 0.5) \\ 2 - 2x, & \text{for } x \in [0.5, 1] \end{cases}.$$

This is a simple example for a non-linear mapping for a dynamical system. For $x_0 \in [0,1]$ define $(x_i)_i$ by

$$x_i := f(x_{i-1}), \quad i \in \mathbb{N}. \tag{1}$$

The tent map is a chaotic system, i.e. little changes in the initial value lead to large effects to later values and the values reached seem unpredictable. However, this is invalid if x_0 can be represented as finite binary number. Thus, we can not reproduce this behavior on the computer.
During this exercise we examine the tent map for finite binary numbers and how to implement the desired chaotic behavior.

(a) Write a program `tent_map.cc`, that computes the sequence $(x_i)_i$ for $x_0 = 0.01401$ and $i = 1, \dots, 100$ and print the results in the terminal. It should look like the following:

```
$ g++ -o tent_map tent_map.cc
$ ./tent_map
0.01401
0.02802
...
```

Write the values in a file `data.dat` by passing the terminal output to this file:

```
$ ./tent_map > data.dat
```

Visualize the data using `gnuplot`. Therefore, start `gnuplot` in the same directory of the data `data.dat` in the terminal with the command `gnuplot`. Afterwards visualize the data using the command

```
plot 'data.dat'
```

(b) The results are chaotic in the beginning, however, we can see that the values form a pattern in the end. If $x_0 = (0.m_1 \dots m_r)_2 \in [0,1]$ is a fixed-point number in binary representation with at most r non-zero decimals and (x_i) defined by (1) then it is possible to show that $x_{r+1} = 0$.

The proof illustrates that for irrational initial values $x_0 \in [0,1]$ the sequence $(x_i)_i$ will be non-periodic. However, this will not help us with our programming task. To achieve non-periodically sequences with finite binary representations, we change the tent map by

$$\widetilde{f}\colon [0,1] \to [0,1], \quad x \mapsto \begin{cases} 1.999999x, & \text{for } x \in [0,0.5) \\ 1.999999 \cdot (1-x), & \text{for } x \in [0.5,1] \end{cases}.$$

Use the new function $\widetilde{f}$ in your code from (a) and visualize your results using `gnuplot`.