Problem Set 1

Exercises for the lecture Fundamentals of Simulation Methods

Prof. Dr. Friedrich Röpke, Prof. Dr. Cornelis Dullemond
Tutors: Xudong Gao (gao@mpia.de), Theodoros Soultanis (theodoros.soultanis@h-its.org), Peter Rodenkirch (Rodenkirch@stud.uni-heidelberg.de)
Tutorial on Thursday/Friday 25.10./26.10. 2018

1. Pitfalls of integer and floating point arithmetic

1. Consider the following C/C++ code:

```
int      i =  7;
float    y = 2*(i/2);
float    z = 2*(i/2.);
printf("%e  %e \n", y,z);
```

   which prints out two float numbers. Explain why the numbers are not all equal.

2. Consider the following numbers:

```
double  a =  1.0e17;
double  b = -1.0e17;
double  c = 1.0;
double  x = (a +  b) + c;
double  y =  a + (b  + c);
```

   Calculate the results for x and y. Which one is correct, if any? Explain, why the law of associativity is here broken.

3. Consider the following C/C++ code:

```
float    x =  1e20;
float    y;
y = x*x;
printf("%e %e\n", x,y/x);
```

   Explain what you see.

2. Near-cancellation of numbers

Consider the following function:

$$f(x) = \frac{x + \exp(-x) - 1}{x^2} \tag{1}$$

Clearly for $x = 0$ this function is ill determined. However, for the limit $x \downarrow 0$ the function goes to a non-zero and non-infinite value.

1. Determine $\lim_{x \downarrow 0} f(x)$
2. Write a computer program that asks for a value of x from the user and then prints $f(x)$
3. For small (but positive non-zero) values of x this evaluation goes wrong. Determine experimentally at which values of x the formula goes wrong.
4. Explain why this happens.
5. Add an if-clause to the program such that for small values the function is evaluated in another way that does not break down, so that for all positive values of x the program produces a reasonable result.