# AMATH 341 / CS 371 Fall 2004: Assignment 1

Instructor: Hans De Sterck Office: MC 5016 e-mail: *hdesterck@uwaterloo.ca*
TA: Jenny Lee Office: MC 5133 e-mail: *j39lee@math.uwaterloo.ca*

Web Site: http://www.math.uwaterloo.ca/~hdesterc/websiteW/courses/amath341.html
Newsgroup: uw.cs.cs371

**Due Date: Friday October 1st, beginning of class**

1. (5 marks)

   Consider the floating point system $F(b = 2, m = 9, e = 5)$.

   (a) what is $\epsilon_{mach}$?

   (b) what is the largest number that can be represented? (give binary and decimal representation)

   (c) what is the smallest, positive number that can be represented? (give binary and decimal representation)

   (d) how many ways are there to represent the number 0 ?

   (e) what is the decimal representation of 1|101110110|0|01011?

   (f) with $x = 1031$, what is $fl(x)$ (in binary and decimal representation; assume chopping)? Calculate $\delta x$ and verify that $|\delta x| \leq \epsilon_{mach}$.

2. (5 marks)

   Download the matlab function `determine_b.m` from the course homepage. Verify in matlab that this function correctly determines the base of the floating point number system on the computer you use (what is the correct base?). Try to understand how the algorithm works. Write down the steps that the algorithm would take to determine the base of floating point number system $F(b = 10, m = 3, e = 3)$. Give the successive values that $a$ takes on in the first phase of the algorithm, the values that $i$ and $a + i$ take on in the second phase, and the final value of $b$ and how it is obtained. Assume that your 'decimal' computer uses rounding.

3. (5 marks)

   Given $p_0 = 1/3$ and $p_1 = 2/3$, a recurrence relation for calculating $p_n$ is given by $p_n = 2/3\ p_{n-1} - 4/9\ p_{n-2}$ for $n \geq 2$. Analyze the stability of the recursion w.r.t. the absolute error. (You can ignore rounding errors that occur after the assignment of the initial values.) (hint: derive a difference equation for the error, find the general solution, and use inequalities)

4. (5 marks) Download the matlab function `qroots.m` from the course homepage. It calculates the roots of the quadratic polynomial $ax^2 + bx + c = 0$ using the well-known formula $x_{1,2} = (-b \pm \sqrt{b^2 - 4ac})/(2a)$.

   (a) For $(a, b, c) = (1, -4e7, 2)$, the roots are, with high accuracy, $x_1^* = 3.999999999999996e7$ and $x_2^* = 5.000000000000006e-8$. Calculate $x_{1,2}$ using `qroots.m`. How many correct digits do you get for $x_1$, and how many for $x_2$? What is the relative error of $x_2$ w.r.t. $x_2^*$? How much bigger than $\epsilon_{mach}$ is this? Find the reason why the algorithm in `qroots.m` is unstable w.r.t. the relative error for calculating $x_2$ in this case. Is there a way to verify that the $x_{1,2}^*$ are more accurate than the $x_{1,2}$ calculated by `qroots.m`?

(b) Find a different algorithm to calculate $x_2$ that is stable w.r.t. $\delta$ (hint: what do you know about the product $x_1\ x_2$?). Modify `qroots.m` accordingly, and call your new routine `qroots_modif1.m`. How many correct digits do you obtain for $x_2$ now? What is the relative error, and compare with $\epsilon_{mach}$. Why do you get a better result than before?

(c) For $(a, b, c) = (1, 2e5, 1.23456789)$, the roots are, with high accuracy, $x_1^* = -6.172839450190519e-6$ and $x_2^* = -1.999999999938272e5$. Which roots do you obtain with `qroots_modif1.m`? Does `qroots.m` give better results? Explain. Find a good algorithm for this case and implement it in matlab. Call your new routine `qroots_modif2.m`.

For (b) and (c), hand in the listing of your modified algorithms.

5. (5 marks)

Define mathematical problem $P$ as follows: compute $z(x) = 1/(x + a)$, with $a$ a fixed, real constant. Derive estimates for the absolute and relative condition numbers $\kappa_A$ and $\kappa_R$ for problem $P$. Discuss the condition of problem $P$ w.r.t. absolute and relative errors. For which values of $x$ is problem $P$ ill-conditioned?

6. (5 marks)

Consider the Ordinary Differential Equation (ODE) $x'(t) = a\ x(t)$ on interval $t \in [0, 4]$, with $a = -5$ and $y(0) = 1$. The exact solution is $x(t) = \exp(a\ x)$.

(a) Discretizing the interval $[0, 4]$ in equal subintervals with length $\Delta t$, we can use Taylor's formula $x(t_i + \Delta t) = x(t_i) + x'(t_i)\ \Delta t + O(\Delta t^2)$ to derive the following numerical method (Euler's method) for solving the ODE:

$$w_{i+1} = w_i + a\ \Delta t\ w_i, \tag{1}$$

with $w_i$ an approximation of $x(t_i)$ (verify the derivation!). Download the skeleton file `myEuler.m` from the course homepage. Complete the skeleton such that Euler's method is implemented. Hand in the listing of your implementation.

(b) Hand in a plot with the program output for timesteps $\Delta t = 0.01,\ 0.1,\ 0.3,\ 0.5$ (all on the same page, use subwindows). What do you observe? Explain. Determine a stability bound on $\Delta t$. In this case, would exact arithmetic remedy the instability?