MATH 4093/5093 Homework 2 Due Fri, 2/4/11

Problem 1. Find each of the following limits and determine the corresponding convergence rate. Your answers should look like this: $\frac{\cos x - 1}{x^2} = -\frac{1}{2} + O(x^2)$ for $x \to 0$; here O is the "big O" notation introduced in Sec. 1.2 of the book. Taylor series will be your main tool to solve these problems.

(a) $\displaystyle\lim_{x\to 0} \frac{\mathrm{e}^x - 1}{x}$;

(b) $\displaystyle\lim_{x\to 0} \frac{\sin x}{x}$;

(c) $\displaystyle\lim_{x\to 0} \frac{\mathrm{e}^x - \cos x - x}{x^2}$;

(d) $\displaystyle\lim_{x\to 0} \frac{\cos x - 1 + x^2/2 - x^4/24}{x^6}$.

Problem 2.

(a) Find the solution $y_0(t)$ of the linear first-order ordinary differential equation

$$y' - \frac{1}{t} y = t \sin t \tag{1}$$

satisfying the initial condition $y(\frac{\pi}{2}) = 0$. Find the solution $y_\epsilon(t)$ of (1) satisfying the "perturbed" initial condition $y(\frac{\pi}{2}) = \epsilon$.

(b) Find the solution $z_0(t)$ of the linear first-order ordinary differential equation

$$z' - z = \mathrm{e}^{-2t} \tag{2}$$

satisfying the initial condition $z(0) = -\frac{1}{3}$. Find the solution $z_\epsilon(t)$ of (2) satisfying the "perturbed" initial condition $z(0) = -\frac{1}{3} + \epsilon$.

(c) Consider $|y_\epsilon(t) - y_0(t)|$ and $|z_\epsilon(t) - z_0(t)|$. How do they behave as functions of the "time" t? Which of the following initial value problems will be more "dangerous" to solve numerically:

$$y' - \frac{1}{t} y = t \sin t\ , \qquad y(\tfrac{\pi}{2}) = 0\ ,$$

or

$$z' - z = \mathrm{e}^{-2t}\ , \qquad z(0) = -\frac{1}{3}\ ?$$

Explain your reasoning.

Problem 3. One important issue in numerical computations is the so-called *machine epsilon* – roughly speaking, this is the smallest numbers that the computer sees as different from zero (working with certain language and certain type of variables). We learned in class that the in the IEEE-1985 standard, a real number is represented by 64 bits so that the smallest positive number that can be represented is $2^{-1022} \approx 2.225 \times 10^{-308}$.

The simple MATLAB script[1] below, called `nikola_petrov_eps1.m`

```
format long
eps = 1.0;
while ( eps ~= 0.0 )
    eps = eps / 10.0
end
```

attempts to compute the machine epsilon in MATLAB; you can download this script from the class web-site. It defines the number `eps` and then divides it many times by 10 until the result is so small that the computer sees it as zero.

On the class web-site you can also download the MATLAB script `nikola_petrov_eps2.m`, which was written for the same purpose, but is slightly different.

(a) Run `nikola_petrov_eps1.m` (by just typing `nikola_petrov_eps1` in MATLAB). Describe what the output is. *Do not attach a printout with the output!*

(b) Run `nikola_petrov_eps2.m`. Explain what this script does, and describe the output of running it. (Again, there is no need to attach a printout with the output.)

(c) Explain why the results of running the two scripts were so different. What do you think is the accuracy of MATLAB? In other words, which script should you use to find out how many accurate decimal digits MATLAB can give you?

Problem 4. Even simple tasks like solving a quadratic equation numerically should be approached with care. One can just use the formula for the roots of a quadratic equation

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \ .$$

– the MATLAB function `nikola_petrov_quadr_eqn1.m` (available at the class web-site) does exactly this. MATLAB works with complex numbers – to see this, solve the quadratic equation $x^2 - 6x + 25 = 0$ by typing `nikola_petrov_quadr_eqn1(1,-6,25)`.

(a) Use `nikola_petrov_quadr_eqn1.m` to solve the quadratic equation

$$10^{-5}\,x^2 - 500000\,x + 1 = 0 \ .$$

---

[1] A *MATLAB script* is a file with MATLAB commands that can be executed by typing the name of the script (without the `.m` extension) on the MATLAB command line.

Compute the absolute and the relative error of the values this function gives for the roots of the equations; The exact roots of this equation (with 20 digits of accuracy) are

$$\begin{aligned} x_1 &= 2.0000000000000000800\ldots \times 10^{-6} \ , \\ x_2 &= 5.0000000000000002000\ldots \times 10^{10} \ . \end{aligned}$$

Remark: To get 10^{-5} in MATLAB, type `1e-5` (*not* `10e-5` , as sometimes people do).

(b) The loss of accuracy in the computations in part (a) was due to the fact that in computing

$$500000 - \sqrt{500000^2 - 4 \times 10^{-5} \times 1}$$

we subtract two nearly equal numbers, which leads to a loss of accurate digits; on the other hand, computing

$$500000 + \sqrt{500000^2 - 4 \times 10^{-5} \times 1}$$

is perfectly safe numerically. To avoid the numerical problems caused by subtracting nearly equal numbers, one can use the "standard" formula when there is no danger, while in the case the sign of the b is such that a dangerous cancellation may occur, the expression for the root can be rewritten as

$$\frac{-b - \sqrt{b^2 - 4ac}}{2a} \cdot \frac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}} = \frac{4ac}{2a(-b + \sqrt{b^2 - 4ac})} = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \ .$$

If the computation of $-b - \sqrt{b^2 - 4ac}$ is dangerous numerically, then the computation of $-b + \sqrt{b^2 - 4ac}$ will be perfectly safe.

Write a MATLAB function called `yourfirstname_yourfamilyname_quadr_eqn2.m` that uses the "standard" formula for one of the roots and the newly derived formula for the other one. You have to use the MATLAB command `if` in order to check the sign of b. Attach a copy of your program, and put your file in the Dropbox of D2L (`learn.ou.edu`).

(c) Use your function `yourfirstname_yourfamilyname_quadr_eqn2.m` to solve the quadratic equation from part (a), and attach a copy of the printout. Compute the absolute and the relative errors of the values of the root produced by your function.