Fall 2015: Numerical Methods I
Assignment 1 (due Sep. 21, 2015)

- Please attempt all homework problems and hand them in by the due date (or earlier!), preferably using \LaTeX{} typesetting (try LyX if you are not yet familiar with \LaTeX{}). Preferably, hand in a printout of your homework (details on where and how to follow). If you need to email your homework, please email it as a single PDF file. Late submissions will only be accepted if you send me an email with a justification before the due date.

- The problems are meant to be challenging and to leave room for discussion. You are not expected to get all of them 100% right. Please make sure you discuss your results whenever that is required; a meaningful discussion that illustrates that you understand the point is usually more important than obtaining the correct result. If you believe that you have to spend an unreasonable amount of time per week on the assignments, please talk to me.

- NYU's integrity policies will be enforced. You are encouraged to discuss the problem with other students, but I expect you to write every line of code yourself and also to write up the solutions by yourself. Copying of any portion of someone else's solution or allowing others to copy your solution is considered cheating.

1. [1+1+2+2pt] Let $\| \cdot \|$ denote a norm in $\mathbb{R}^n$, and $A, B \in \mathbb{R}^{n \times n}$ square matrices. The induced matrix norm is defined by

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}.$$  

(a) Show that

$$\|A\| = \sup_{\|x\|=1} \|Ax\|.$$ 

(b) Show that the induced matrix norm satisfies

$$\|AB\| \leq \|A\| \|B\|$$ for all $A, B \in \mathbb{R}^{n \times n}$.

(c) Show that the matrix norm induced by the vector 1-norm $\|x\|_1 = \sum_{i=1}^n |x_i|$ is the maximum 1-norm of the matrix columns.

*Hint:* Denote the columns of $A$ by $a_j$, i.e., $A = [a_1 \cdots a_n]$ and take a vector $x$ with $\|x\|_1 = 1$. Show that

$$Ax = \sum_j x_j a_j, \text{ and } \|Ax\|_1 \leq \max_j \|a_j\|_1.$$ 

Then, by choosing $x$ appropriately, show that one can satisfy $\|Ax\|_1 = \max_j \|a_j\|_1$ and use that to argue that the matrix 1-norm is equal to the maximum 1-norm of the matrix columns.
(d) Now, let's compute the matrix norm induced by the $\infty$-vector norm defined by $\|x\|_\infty := \max_{1 \leq i \leq n} |x_i|$. Hint: Denote the rows of $B$ by $b_j$, i.e.,

$$B = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix},$$

and consider a vector $x$ with $\|x\|_\infty = 1$. Show first that $\|Bx\|_\infty \leq \max_i \|b_i\|_1$.

Then, show that by choosing $x$ appropriately, one can satisfy $\|Bx\|_\infty = \max_i \|b_i\|_1$. Use that to argue that the matrix norm induced by the $\infty$-norm is the maximum 1-norm of the matrix rows.

2. [1+1+1+2pt] Compute absolute and relative condition numbers (for infinitesimal perturbations) for the following problems

(a) $G(x) = \sqrt{x}$ for $x \in \mathbb{R}, x \geq 0$.
(b) $G(x) = x^2$ for $x \in \mathbb{R}$.
(c) $G(x) = 1/x$ for $x \in \mathbb{R}, x \neq 0$.
(d) $G : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ defined by

$$G(x) = \begin{pmatrix} x_1x_2 \\ x_2^2 \end{pmatrix},$$

where $x = (x_1, x_2)^T \in \mathbb{R}^2$. Compute the condition numbers when using the $\| \cdot \|_1$ and the $\| \cdot \|_\infty$ norms in $\mathbb{R}^2$.

3. [4pt] Due to the finite precision in floating point number representation, there are gaps between consecutive numbers. The size of these gaps depends on the size of the number and on the precision (e.g., double or single precision). Matlab provides the function `eps(·)` (the analogue in NumPy is the function `spacing`), which returns, for a number, the distance to the next floating point number in the same precision. Using the form of double and single floating point number representation, explain the values you find for

- `eps(1),` 
- `eps(single(1)),` 
- `eps(2^{40}),` 
- `eps(single(2^{40})).`

4. [2pt] Try out the following two experiments and explain the behavior:

```
> a = 0.8;
> b = 0.7;
```

---

1This value, $\text{eps}(1) = \text{eps} \approx 2.22 \times 10^{-16}$, is usually referred to as machine epsilon, and it gives an indication of the rounding error when dealing with double precision floating point numbers.

2Note that the Matlab command `single` switches to single precision.
5. **[2+3pt]** Often, derivatives of functions can only be computed numerically. For a three times differentiable function \( f : \mathbb{R} \to \mathbb{R} \) one can use Taylor expansions around the point \( x_0 \) to show the following approximations:

\[
    f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + O(|h|) \quad \text{(one-sided finite differences)}.
\]

\[
    f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} + O(|h|^2) \quad \text{(centered finite differences)}.
\]

In these approximations, one subtracts very similar numbers from each other, which is badly conditioned and can lead to significant cancellation errors. Here, we discuss an alternative.

(a) Assuming the function \( f \) can also be defined for complex arguments, use a Taylor expansion of \( f(x_0 + ih) \) to show that

\[
    f'(x_0) = \frac{\text{Im}(f(x_0 + ih))}{h} + O(|h|^2),
\]

where “\( \text{Im} \)” denotes the imaginary part of a complex number. This is called complex step differentiation, and it obviously avoids numerical cancellation.

(b) Compare these three approximations to compute the derivative of the function

\[
    f(x) = \frac{\exp(x)}{\cos(x)^3 + \sin(x)^3}
\]

at \( x_0 = \pi/4 \). In order to do so, use progressively smaller perturbations \( h = 10^{-k} \) for \( k = 1, \ldots, 16 \) and present the resulting approximations in a table. The exact value is \( f'(x_0) = 3.101766393836051 \). Discuss your result.

---

3 The double equal sign checks if the numbers to its left and right are the same (i.e., every bit coincides!). An answer of 0 means false (not the same), and 1 means true. Note that you should never do this for floating point numbers! You should always test if the difference between two numbers is small (e.g., \( 10^{*}\text{eps}(1) \)) rather than if they are equal.
6. **[4pt]** Write a program that approximates $\exp(x)$ with its Taylor series:

$$\exp(x) \approx \sum_{i=0}^{n} \frac{x^i}{i!}$$  \hspace{1cm} (1)

and plot, for $x = -5.5$ the difference to the exact value obtained with $n = 1, \ldots, 30$, where you use the (truncated) Taylor expansion in three different ways:

- Use the formula (1) directly to compute $\exp(-5.5)$.
- First use that $\exp(-5.5) = 1/\exp(5.5)$, and then use (1).
- Use that $\exp(-5.5) = (1/\exp(0.5))^{11}$ and then use (1).

Discuss the results, which show that a different mathematical formulation can significantly influence the accuracy of a computation.