Universität Duisburg-Essen
Computational Mechanics
Campus Essen

Dr. S. Vanis

Fall Term 2010/11

14. October 2010

Introduction to Numerical Methods
Tutorial 1

This first tutorial wants to introduce you to the matlab/octave programming language. These exercises are not counted as programming exercises. The programming exercises will be explicitly tagged out as such.

Exercise 1:

(i) Give the representation of the real numbers $\frac{4}{3}$, 0.0000012345 and $1.2345e{-}06$ in the following formats

- **(1)** short ; short e ; short g
- **(2)** long ; long e ; long g
- **(3)** rat

and explain the results. What is the meaning of "e" in this context?

(ii) What happens if you use the following commands in matlab/octave

- **(1)** "$f1 = 3 * 3$; "
- **(2)** "$f2 = 3\hat{}3$; " "$g1 = [3\ ,\ 1; 1\ ,\ 3]\hat{}3$"
- **(3)** "$f3 = 3.\hat{}2$ " "$g2 = [3\ ,\ 1; 1\ ,\ 3].\hat{}3$"
- **(4)** "$3 + 3$ "
- **(5)** " disp('Hallo world!'); disp('Introduction to Numerical Methods'); "

What is the difference between $g1$ and $g2$, cf., (2) and (3) ?

Exercise 2:
Let $A = (a_{ij})_{i,j=1\ldots n}$ be the matrix with entries $a_{ij} = i/j$ and x the vector $x_j = j$ for $j = 1 \ldots n$, i.e.,

$$A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ 2 & 1 & \frac{2}{3} & \ddots & \frac{2}{n} \\ 3 & \frac{3}{2} & \ddots & \ddots & \frac{3}{n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ n & \frac{n}{2} & \cdots & \cdots & 1 \end{bmatrix} \qquad x = \begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ n \end{bmatrix}.$$

(i) Try three different ways to build the matrix A, i.e.,

(1) using 2 nested for-loops
(for k=1... for l=1... A(k,l)=k/l; end end)

(2) using one vector and one for-loop
(k=1:n; for l=1... A(:,l) = k'/l; end)

(3) using the product of two vectors
*(k=(1:n)'; l=ones(1,n)./k'; A = k * l;)*

With the commands *tic* and *toc* the time needed for the operations in between these commands can be determined (*tic; operations; toc;*). Use these commands to determine the time matlab/octave needs to build the matrix A in the ways described above for $n = 100, 500, 1000, 1500, 2500(, 5000)$.
What do you recognize?
Hint: Write an m-file which takes the size of the matrix n as input and in which all three ways are included one after another. Determine the separate times by enclosing each part in *tic ... toc*.

(ii) Build the matrix A and the vector x in matlab/octave and compute the matrix-vector product $y = Ax$ for $n = 3, 5, 10, 100, 1000$.
Verify the solution by your theoretical knowledge on matrix-vector multiplication.

(∗)-Exercise 3: $(2 + 5 + 3 = 10$ points)

(i) Build a vector x which starts with 0 and ends at 2π and use it to plot with matlab $\sin(x)$ and $\cos(x)$ on the interval $[0, 2\pi] \subset \mathbb{R}$. Plot the functions once into two separate diagrams and once into only one diagram.
Use different step sizes for the vector x, i.e., step sizes $1; 0.5; 0.1$, and compare the results. Just plot the sin-curves for all three step sizes into a single diagram.

(ii) Use the possibility to plot different graphs into one diagram to obtain the intersection points of the following three functions on the interval $[-2, 2] \subset \mathbb{R}$

- $\exp(-x^2)$,
- $\sin(x^2)$,
- $\frac{1}{x^3+9}$.

Just use the "Data Cursor" to select the intersection points and read off the values.
Try different step sizes in this case as well. Use the command *legend* (in "matlab", when you use "octave" you have to find an alternative to this command), to label the different functions in the diagram.

(iii) Use the command *mesh(x,y,z)* to make a 3 dimensional diagramm for $x \in [-2, 2]$, $y \in [-2, 2]$ and $z(x, y) = x^2 - y^2$. Choose a step size of 0.1 in both directions, i.e., in x-direction as well as in y-direction.

Exercise 4:
Create a file "*Name.txt*" (where you replace name by your first name). Write an arbitrary sentence in the file. Leave two lines free and write a value table for the function $\exp(x^3)$ on the interval $[-1.5, 3.5] \subset \mathbb{R}$ using the step size 0.25.

Delivery: The exercises have to be handed in before the tutorial on **Thursday, 21 October 2010**. Exercises which are handed in later may be corrected but will give no points for the $(*)$- and programming exercises. The source codes have to arrive by E-Mail not later than 12 o'clock on the date of delivery.