Scientific Computing (COMP3407)

Programming Assignment 1

Start: Feb. 13, 2019 Due: 22:00, Feb. 27, 2019

Please upload your source code file (*.m) or livescript file (*.mlx) to Moodle.

For problem 1–3, a livescript file is suggested, because you can add more demonstration on it.

1 Catastrophic cancellation (15 marks)

When two nearly equal machine numbers are subtracted, the number of significant digits in the result will be reduced significantly. This is called *catastrophic cancellation.*

(a) (5 marks) Please write a MATLAB program to compute

$$f(x) = \sqrt{x^2+1} - 1$$

$$g(x) = \frac{x^2}{\sqrt{x^2+1}+1}$$

for a succession of values $\{x_n\} = \{8^{-1}, 8^{-2}, \cdots, 8^{-5}\}$. Theoretically, these two functions $f(x)$ and $g(x)$ are the same, but in different expressions. Compute their relative differences $|f(x) - g(x)|/|g(x)|$ and list them in a table.

x	8^{-1}	8^{-2}	8^{-3}	8^{-4}	8^{-5}
$\lvert f(x) - g(x)\rvert/\lvert g(x)\rvert$					

Which of the two expressions is more reliable?

(b) (10 marks) Suggest an alternative expression for each of the following functions to avoid loss of significance when evaluating them in the same process as in a):

$$(i)\; f(x) = \frac{1-x}{1+x} - \frac{1}{3x+1}$$

$$(ii)\; g(x) = \sin x - \tan x$$

Please give in your file the alternative expressions used to replace $f(x)$ and $g(x)$, respectively.

2 Harmonic series (25 marks)

In this problem we investigate the process of summing up the terms of the partial harmonic series

$$h_k = \sum_{i=1}^{k} \frac{1}{i},$$

in the interest of studying the effect of rounding errors of a floating-point number system. It is known that $\lim_{k\to\infty} h_k = \infty$. Please see page 5 of lecture notes #1 for more discussion. Suppose that the IEEE standard **single-precision** floating-point number system is used.

(a) (10 marks) Determine analyticlaly the approximate value of k_0 at which the computed partial harmonic series h_k stops increasing. Explain why this happens. (**Hint**: $\lim_{k\to\infty} h_k/\log(k) = 1$.)

(b) (15 marks) Write a MATLAB program to verify your estimated value k_0 in (a).

(**Remark**: In MATLAB, floating-point numbers are stored as double precision by default. Thus, you should convert variables or numbers in your program to single precision. For this purpose, you may use the "**single**" command in MATLAB. Please find more details about this command at MATLAB on-line help "Creating Floating-Point Data".)

3 Numerical derivative (15 marks)

Let h be a small positive number. The derivative of a function f at x_0 can be approximated by the *forward divided difference*

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0)}{h}$$

or by the *central divided difference*

$$f'(x_0) \approx \frac{f(x_0+h) - f(x_0-h)}{2h}.$$

Use these two schemes respectively to compute approximations to the derivative of $f(x) = \sin(x)$ at $x_0 = 1$, with $h = 2^{-1}, 2^{-2}, \cdots, 2^{-32}$. Since $f'(x) = \cos(x)$, use $\cos(1)$ as the ground truth to compute the relative errors of these approximations. Use the **loglog** command in MATLAB to plot the relative error of forward and central method in one figure. **loglog** is similar to **plot** but has both axis scaled by log. Which of the two schemes do you think is more accurate according to the error plots? Explain your answer. (**Hint**: Use the Taylor expansion to study the order of approximation in h.)

4 Gaussian elimination with partial pivoting (25 marks)

Please implement Gaussian elimination with partial pivoting without storing permutation matrix explicitly. Instead, two integer arrays should be used to store the interchange of rows and variables. For example, with row permutation, we transform the matrix

$$\begin{pmatrix} 2 & 4 & 6 \\ 1 & 8 & 9 \\ 3 & 2 & 5 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 8 & 9 \\ 3 & 2 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

.

The integer array will be (2 3 1). It means that, the first row after permutation is the second row in the original matrix, the second row after permutation is the third row in the original matrix, and so on.

Please submit your program in the form of a function file. For a linear system $\mathbf{Ax} = \mathbf{b}$, the function should have the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ as input, and return $\mathbf{x}$ as output. It is assumed here that $\mathbf{A}$ is square and non-singular. Your program should print proper error messages for rectangular or singular $\mathbf{A}$.

5 Hilbert matrix (20 marks)

The Hilbert matrix of order n is a square matrix $\mathbf{H} = \{h_{ij}\}_{i,j=1}^{n}$ with the entries being

$$h_{ij} = \frac{1}{i+j-1}, \quad 1 \leq i, j \leq n.$$

This is an ill-conditioned matrix. We will see that a linear system of equations is difficult to solve accurately when its coefficient matrix is such an ill-conditioned matrix.

Generate the Hilbert matrix $\mathbf{H}$ of order n, and also generate the n-vector $\mathbf{b} = \mathbf{H}\mathbf{x_0}$, where $\mathbf{x_0} = (1, 1, \ldots, 1)^T$, for $n = 2, 3, \ldots, 12$. Use your program for Gaussian elimination implemented in question 4 to solve the resulting linear system $\mathbf{Hx} = \mathbf{b}$, to obtain an approximate solution $\hat{\mathbf{x}}$. Compute the ∞-norm of the residual $\mathbf{r} = \mathbf{b} - \mathbf{H}\hat{\mathbf{x}}$ and that of the error $\Delta\mathbf{x} = \hat{\mathbf{x}} - \mathbf{x_0}$, where $\mathbf{x_0} = (1, 1, \ldots, 1)^T$. Also use the MATLAB routine *cond*() to compute the condition number cond($\mathbf{H}$) for each value of n.

Submit a script to draw the following graphs and display them all in the same window via MATLAB function **subplot**.

1) The plot of *cond*($\mathbf{H}$) for $n = 2, 3, \ldots, 12$. Please use $\log_{10}$ to change the scale of the condition numbers in the plot;
2) The plot of $\|\Delta\mathbf{x}\|_\infty$ for $n = 2, 3, \ldots, 12$.