University of Nottingham
School of Mathematical Sciences

Submission Date: Monday, 7th November 2016 **Assessed Coursework 1**

The following questions are to be used for the coursework assessment in the module G14SCC. Credit will be given for relevant working associated with solving the problems, as well as the answer.
A single zip or tar file containing your solution should be submitted through the module webpage on Moodle as per the instructions on the Coursework Submission form.
The style and efficiency of your programs is important. A barely-passing solution will include attempts to write programs which include some of the correct elements of a solution. A borderline distinction solution will include working code that neatly and efficiently implements the relevant algorithms, and that shows evidence of testing.
An indication is given of the approximate weighting of each question by means of a figure enclosed by square brackets, e.g. [12].

All calculations should be done in double precision.

1. In this question, you are required to write a C++ program to solve the linear system

$$Ax = b,$$

using Gaussian elimination with partial pivoting, where A is an $n \times n$ matrix and b is a column vector with n rows. The structure of your code should be as follows:

- In a file `matrix_allocation.cpp` write C++ functions `allocate_matrix` and `deallocate_matrix` to dynamically allocate and delete storage for an $n \times m$ matrix, cf. the lecture notes.
- In a file `solve_ax_b.cpp` write a C++ function called `solve_ax_eq_b` which performs Gaussian elimination with partial pivoting. This should use the functions `row_pivoting` and `back_substitution`, cf. below.
- In a file `row_pivoting.cpp` write a C++ function called `row_pivoting` which performs partial pivoting on the augmented matrix.
- In a file `back_substitution.cpp` write a C++ function called `back_substitution` which performs backward substitution on the augmented matrix.
- (For testing, cf. below). In a file `mat_vec_prod.cpp` write a C++ function called `mat_vec_prod` which computes the product of an $n \times n$ matrix with a column vector with n rows.
- In a file `q1.cpp` write a C++ test program which computes (and outputs) the solution to the following two matrix problems:

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 10 \\ 8 \end{pmatrix}.$$

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 10 \\ 8 \end{pmatrix}.$$

The matrix A should be dynamically allocated using the function `allocate_matrix`. Test your code by computing the ℓ_2-norm of the residual $b - A\tilde{x}$, where $\tilde{x}$ is the solution computed using the function `solve_ax_eq_b`. [60]

2. Consider the following nonlinear equation: find x such that

$$f(x) = 0, \tag{1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}^n$, $n \geq 1$.

(a) In a file `newton.cpp` write a C++ function `newton` which computes the solution x to (1) using Newton's method.

(b) In a file `q2b.cpp` write a C++ program which computes the solution $x = (x_1, x_2, x_3)^\top$ of the equations

$$\begin{aligned} x_1^2 + x_2^2 + x_3^2 - 1 &= 0, \\ 2x_1^2 + x_2^2 - 4x_3 &= 0, \\ 3x_1^2 - 4x_2 + x_3^2 &= 0, \end{aligned}$$

which satisfies $x_1 > 0, x_2 > 0, x_3 > 0$. The file `q2b.cpp` should also contain the functions `fun` and `dfun` needed by `newton` to evaluate f and the Jacobi matrix of f, respectively. A starting guess of $x = (0.5, 0.5, 0.5)^\top$ may be employed.

(c) In a file `q2c.cpp` write a C++ program which computes the solution $x = (x_1, x_2)^\top$ of the equations

$$\begin{aligned} x_1^2 + x_2^2 - 1 &= 0, \\ 5x_1^2 + 21x_2^2 - 9 &= 0. \end{aligned}$$

As above, the file `q2c.cpp` should also contain the functions `fun` and `dfun` needed by `newton` to evaluate f and the Jacobi matrix of f, respectively. A starting guess of $x = (0.5, 0.5)^\top$ may be employed to compute the first solution; can you find any others?

[40]