CORK INSTITUTE OF TECHNOLOGY

INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Autumn Examinations 2017/2018

Module Title: Numerical Methods 1

Module Code: MATH7015

School: School of Mathematics

Programme Title: Bachelor of Engineering (Hons) in Mechanical Engineering – Year 2
Bachelor of Engineering (Hons) in Structural Engineering – Year 2

Programme Code: CR_EMECH_8 CR_CSTRU_8

External Examiners(s): Dr. Ann O'Shea

Internal Examiners(s): Dr. Robert Heffernan

Instructions: Answer all four questions. Each question carries 25 marks.
The exam has two parts.
The first part should be answered on the computer. You should save your work as a **macro enabled** workbook.
The second part should be answered by hand in an exam booklet.
Do not write, draw or underline in RED.
Show all calculations and workings in full.

Duration: 2 Hours

Sitting: Autumn 2017

Requirements for this examination:

Note to Candidates:

Please check the Programme Title and the Module Title to ensure that you have received the correct examination paper. If in doubt please contact an Invigilator.

Part 1: Computer-based

Question 1.

Consider the function

$$f(x) = x\sin(x).$$

(a) Use Excel to plot the graph of $f(x)$ over the interval $[2,4]$ using a step size of 0.1.

(b) Write VBA code that implements the **Bisection method** to find roots of the function $f(x)$. Your code should take the following into account:

- Your code should ask the user for an initial approximation and should display the final approximation either using a message box or by writing it to the spreadsheet.
- The code should halt when two successive iterates are within $\varepsilon = 10^{-5}$ of each other, i.e. when $|x_{k+1} - x_k| \leq \varepsilon$.
- Your code should also halt when a maximum number of 1000 iterations have been reached. In this case, the user should be notified that the method did not converge.

[25 Marks]

Question 2.

Consider the linear system

$$A\vec{x} = \vec{b}$$

where

$$A = \begin{bmatrix} 5 & -2 & 3 \\ -3 & 9 & 1 \\ 2 & -1 & -7 \end{bmatrix}, \vec{b} = \begin{bmatrix} -1 \\ 2 \\ 3 \end{bmatrix} \text{ and } \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Write VBA code that implements the **Gauss-Seidel** method to approximate the solution of this system.

Your code should take the following into account:

- If $\vec{x_{k+1}}$ is the $(k+1)$th iterate, then your code should halt if

  $$||\vec{x_{k+1}} - \vec{x_k}|| \leq \varepsilon ||\vec{b}||$$

  where the norm is the **Euclidean norm** and $\varepsilon = 0.001$.
- Your code should otherwise halt if 100 iterations have been computed. In this case the user should be informed that the method not not converge.
- If the method does converge, then the answer should be written to the spreadsheet.

[25 Marks]

Part 2: Written

Question 3.

(a) Use the method of **Gaussian elimination** to solve the system

$$A\vec{x} = \vec{b}$$

where

$$A = \begin{bmatrix} 1 & -1 & 2 \\ 3 & -1 & 2 \\ 6 & -3 & 8 \end{bmatrix} \text{ and } \vec{b} = \begin{bmatrix} 5 \\ -1 \\ 9 \end{bmatrix}$$

(b) On the basis of part (a) above, write down the LU-decomposition of the matrix A.

[18 Marks]

Question 4.

In this question you are given the following data about a function $y = f(x)$.

i	0	1	2
x_i	1	2	3
y_i	3	2	9

(a) Construct the Lagrange interpolating polynomial for this data.

(b) Create a divided-difference table and use this to construct the Newton interpolating polynomial for this data.

[16 Marks]

Question 5.

(a) Consider the data

x	1.2	1.4	1.6	1.8	2
$f(x)$	0.182	0.336	0.470	0.588	0.693

i. Approximate $f'(1.6)$ using the three-point formula.

ii. Approximate $f''(1.6)$ using the three-point centred formula.

(b) Approximate the integral

$$\int_1^2 e^{-x^2}\, dx$$

using Simpson's rule.

[16 Marks]

Formulæ

Bisection method: $x_k = \frac{a_k+b_k}{2}$

Gauss-Seidel: If $x_i^{(k+1)}$ is the ith component of the $(k+1)$th iterate, then

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)$$

Euclidean norm: $\|\vec{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x_1^2 + x_2^2 + \cdots x_n^2}$.

Lagrange interpolating polynomial:

$$p(x) = \sum_{j=0}^{n} y_j L_j(x) \text{ where } L_j(x) = \prod_{\substack{i=0\\ i\neq j}}^{n} \frac{x-x_i}{x_j-x_i}.$$

Divided differences:

$$f[x_i] = y_i \text{ and } f[x_i,\ldots,x_j] = \frac{f[x_{i+1},\ldots,x_j] - f[x_i,\ldots,x_{j-1}]}{x_j - x_i}.$$

Newton interpolating polynomial:

$$p(x) = \sum_{j=0}^{n} c_j\phi_j(x) \text{ where } \phi_j(x) = \prod_{i=0}^{j-1}(x-x_j) \text{ and } c_j = f[x_0,\ldots,x_j].$$

Three-point formula for $f'(x)$:

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

Three-point centred formula for $f''(x)$**:**

$$f''(x_0) \approx \frac{1}{h^2}\left(f(x_0-h) - 2f(x_0) + f(x_0+h)\right)$$

Simpson's rule:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left(f(a) + 4f\left(\frac{b+a}{2}\right) + f(b)\right).$$