This question paper consists of 5 printed pages, each of which is identified by the Code Number COMP5930M.

This is a closed book examination. No material is permitted.

School of Computing

January 2019

COMP5930M

Scientific Computation

Answer ALL questions

Time allowed: 2 hours

STUDENT NUMBER									
SEAT NUMBER									

NOTE TO INVIGILATOR AND STUDENT
This question paper must be attached with a treasury tag to the back of the answer booklets. It is the student's responsibility to attach the question paper.

Question 1

Assume that $F(x) : \mathbb{R} \rightarrow \mathbb{R}$ is a continuously differentiable scalar function that has the derivative $F'(x)$. We want to find a solution x^* to the nonlinear equation $F(x^*) = 0$.

To find the solution, we consider two different numerical methods, *Newton's method* or the *bisection method*. The pseudocode for these two methods is provided below. The methods require either a starting point x_0 or an initial bracket $[x_L, x_R]$, a convergence tolerance Tol, and the maximum number of iterations $k_{\max}$.

Newton's method

[x,f] = Newton(F, x_0, Tol, $k_{\max}$)

A1. Set $k = 0$

A2. While $|F(x_k)| > Tol$ and $k < k_{\max}$

- (i). Calculate $F'(x_k)$
- (ii). Update $x_{k+1} = x_k - F(x_k)/F'(x_k)$
- (iii). Increment $k \rightarrow k + 1$

A3. End

A4. Set $x = x_{k+1}$ and $f = F(x_{k+1})$

Bisection method

[x,f] = Bisection(F, x_L, x_R, Tol, $k_{\max}$)

A1. Set $k = 0$

A2. While $|x_R - x_L| > Tol$ and $k < k_{\max}$

- (i). Set $x_C = (x_L + x_R)/2$
- (ii). Evaluate $F(x_C)$
- (iii). If $|F(x_C)| < Tol$, Exit While.
- (iv). If $F(x_L)F(x_C) < 0$, set $x_R = x_C$. Otherwise set $x_L = x_C$.

A3. End

A4. Set $x = x_C$ and $f = F(x_C)$

(a) Identify three key differences between Newton's method and the bisection method. **[3 marks]**

(b) Consider the case $F(x) = x^3 - x^2$.

(i) Find the exact solutions of $F(x^*) = 0$ by polynomial factorisation. **[3 marks]**

(ii) For which of these solutions x^* is the convergence of Newton's method *quadratic* (with order 2) and for which is it only *linear* (with order 1)? Justify your answer by studying the properties of $F(x)$ at x^*. **[3 marks]**

(iii) Is the initial bracket $[x_L, x_R] = [-1, 2]$ for the bisection method suitable for finding a solution $F(x^*) = 0$? Justify your answer by studying the properties of $F(x)$. **[2 marks]**

(iv) Apply one step of the bisection method starting from the initial bracket $[x_L, x_R] = [-1, 2]$. Which solution x^* does the method converge to? **[3 marks]**

(v) Can the bisection method converge to a different solution $x^* \in [-1, 2]$ if we change the initial bracket? Justify your answer by studying the properties of $F(x)$ at x^*. **[3 marks]**

(c) Explain how Newton's method can be modified to eliminate the requirement of computing the exact derivative $F'(x)$ (the *secant method*) How does this change the way the method has to be started? **[3 marks]**

[question 1 total: 20 marks]

Question 2

A scalar function $u(x, t)$ satisfies the following nonlinear partial differential equation

$$\frac{\partial u}{\partial t} = \left[\frac{\partial u}{\partial x}\right]^2 + 2u\frac{\partial^2 u}{\partial x^2}, \tag{1}$$

for $x \in [0, 1]$ with boundary conditions $u(0, t) = \alpha$, $u(1, t) = \beta$, and $t > 0$ with initial conditions $u(x, 0) = U_0(x)$.

On a uniform grid of m nodes, with nodal spacing h, covering the domain $x \in [0, 1]$, we first discretise the equation in space using the finite difference schemes

$$\frac{\partial u}{\partial x}(x_i) \approx \frac{u_{i+1} - u_i}{h}, \quad \frac{\partial^2 u}{\partial x^2}(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}.$$

The semi-discrete problem $\dot{u}_i = f(u_{i-1}, u_i, u_{i+1})$ is then discretised using backward Euler

$$\frac{u^{k+1} - u^k}{\Delta t} = f(u_{i-1}^{k+1}, u_i^{k+1}, u_{i+1}^{k+1})$$

where $\Delta t > 0$ is a constant time-step size.

(a) Show that the fully-discrete form of the problem may be written as:

$$F_i(\mathbf{U}) = \frac{u_i^{k+1} - u_i^k}{\Delta t} - \frac{[u_{i+1}^{k+1}]^2 - 3[u_i^{k+1}]^2 + 2u_i^{k+1}u_{i-1}^{k+1}}{h^2} = 0 \tag{2}$$

for $i = 1, 2, \ldots, m$, where $\mathbf{U} = (u_1^{k+1}, u_2^{k+1}, \ldots, u_m^{k+1})$ is the discrete solution vector at the next time step. **[7 marks]**

(b) The Jacobian matrix $\mathbf{J}$ for the nonlinear system $\mathbf{F}(\mathbf{U}) = \mathbf{0}$ is defined elementwise as

$$J_{ij} = \frac{\partial F_i}{\partial U_j}.$$

Show that the Jacobian for the discrete problem (2) is a tridiagonal matrix by computing the nonzero elements J_{ij} of the i'th row. **[4 marks]**

(c) Consider a numerical Jacobian approximation

$$J_{ij} \approx \frac{F_i(\mathbf{U} + \delta \mathbf{e}_j) - F_i(\mathbf{U})}{\delta}$$

where $\delta > 0$ is a small perturbation and $\mathbf{e}_j = (0, 0, \ldots, 1, 0, \ldots)^T$ is a vector with the j'th component equal to 1 and all others equal to 0.

Describe an efficient method for evaluating the numerical Jacobian of a tridiagonal matrix using only three evaluations of $\mathbf{F}$. **[2 marks]**

(d) Formulate in pseudocode an algorithm for solving the fully-discrete problem (2) using Newton method's in each time step, including:

- a time-stepping procedure that calls Newton's method at each time step;
- a choice of the initial guess for Newton's method at each iteration;
- an efficient solution of the linear system at each Newton iteration.

(Note: It is not necessary to provide pseudocode for solving the tridiagonal linear system, simply mention which algorithm you would choose.) **[7 marks]**

[question 2 total: 20 marks]

Question 3

Provide brief answers to the following questions on computational linear algebra:

(a) Define what we mean when we say that a matrix A of size $n \times n$ is *sparse*. **[1 mark]**

Identify three different data structures for efficiently storing sparse matrices. **[3 marks]**

(b) Explain how the *LU factorisation* of an $n \times n$ matrix, $A = LU$, can be used to solve a linear system $Ax = b$. **[2 marks]**

Explain why reordering the rows/columns of a sparse matrix A can increase the computational efficiency of LU factorisation. **[2 marks]**

(c) Describe the *greedy minimum degree algoritm* for reordering a sparse matrix A. **[4 marks]**

(d) Formulate in pseudocode the *Jacobi iteration* as an iterative method for solving the linear problem $Ax = b$. **[4 marks]**

(e) Identify four properties of a good *preconditioner* M for a linear problem $Ax = b$ that allow the preconditioned problem $(M^{-1}A)x = M^{-1}b$ to be solved more efficiently using an iterative method (such as the conjugate gradient method) than the original problem. **[4 marks]**

[question 3 total: 20 marks]

[grand total: 60 marks]