This question paper consists
of 10 printed pages, each of
which is identified by the
Code Number COMP5930M01

©UNIVERSITY OF LEEDS

School of Computing

January 2017

COMP5930M

Scientific Computation

Time allowed: 2 hours

Answer ALL THREE Questions.

This is a closed book examination.

This means that you are not allowed to bring any material into the examination.

Calculators which conform to the regulations of the University of Leeds are permitted but all working must be shown in order to gain full marks.

Turn over for question 1

Question 1

Given functions $\mathbf{F}(\mathbf{x})$ and $\mathbf{J}(\mathbf{x})$ that return, respectively, a system of nonlinear equations and the Jacobian matrix evaluated at point $\mathbf{x}$, a Newton algorithm can be defined as follows to approximate the point $\mathbf{x}^*$ such that $\mathbf{F}(\mathbf{x}^*) = 0$. The algorithm requires a starting point $\mathbf{x}_0$ and a convergence measure Tol.

Algorithm A

[$\mathbf{x}$,$\mathbf{f}$] = Newton($\mathbf{x}_0, Tol$)

A1. $\mathbf{x} = \mathbf{x}_0$, $\mathbf{f} = \mathbf{F}(\mathbf{x}_0)$

A2. While $|\mathbf{f}| > Tol$

(i). Compute $\mathbf{A} = \mathbf{J}(\mathbf{x})$

(ii). Solve $\mathbf{A}\delta = -\mathbf{f}$

(iii). Update $\mathbf{x} = \mathbf{x} + \delta$, $\mathbf{f} = \mathbf{F}(\mathbf{x})$

A3. End

a Describe the purpose of the following two modifications to Algorithm A. In each case state the effect this modification could have on the final computed solution $\mathbf{x}$ and on the performance of the overall algorithm.

i A maximum number of Newton iterations, $maxit$, is imposed.

[3 marks]

ii A scalar parameter λ, with $0 < \lambda \leq 1$, is introduced at step A2.(iii) such that the solution is updated as $\mathbf{x} = \mathbf{x} + \lambda\delta$.

[3 marks]

Answer:

+——————

(a) (i) This modification prevents the algorithm from looping indefinitely, particularly in cases where convergence has stalled. [1 mark]

The final solution $\mathbf{x}$ may not be a root and there should be notification that the algorithm has terminated unsuccessfully. [1 mark]

It would have no effect on the behaviour of the algorithm, although a small value may prevent the convergence tolerance being reached [1 mark]

(ii) This modification can control the convergence of the algorithm by requiring that the function norm is reduced at each iteration. [1 mark]

It should prevent divergence of the solution $\mathbf{x}$ away from the root. [1 mark]

It will not affect the final solution computed but should increase the overall robustness of the performance. [1 mark]

——————+

b Describe an algorithm for computing an appropriate value for the scalar parameter λ during the normal execution of the Newton algorithm. This description should include criteria for completion of this step in every case.

[6 marks]

Answer:

+——————

(b) One possible algorithm is step-halving.

Replace step A2.(iii) with:

1. Set $\lambda = 1$, counter $k = 0$, and parameter $kmax$ [1 mark]

2. Compute $x1 = x + \lambda\delta, f1 = F(x1)$ and set $k = k + 1$ [1 mark]

3.

a. If $|f1| > |f|$ and $k < kmax$

Set $\lambda = \lambda/2$ and return to 2. [1 mark]

b. Else

$x = x1, f = f1$ [1 mark]

c. End

During normal execution the algorithm will terminate when the function norm has been reduced and the computed step is taken. [1 mark]

If the maximum number of steps $kmax$ is taken the algorithm will terminate unsuccessfully and a Newton iteration is attempted from the new point. This prevents a possible infinite loop and also ensures that a finite update is made to the solution in every case. [1 mark]

——————+

c A *homotopy continuation* approach can be defined for the nonlinear system $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ in the form

$$\mathbf{G}(\alpha, \mathbf{x}) \;=\; \mathbf{F}(\mathbf{x}) + (\alpha - 1)\mathbf{F}(\mathbf{x}_0) \tag{1}$$

with scalar parameter $\alpha \in [0, 1]$.

A practical algorithm can be constructed from Equation (1) that uses M steps of size $\Delta\alpha = 1/M$ with

$$\mathbf{x}^k \;=\; \mathbf{x}^{k-1} - \Delta\alpha\, \mathbf{J}^{-1}(\mathbf{x}^{k-1})\mathbf{F}(\mathbf{x}^0) \tag{2}$$

where $k = 1, ..., M$ and $\mathbf{x}^0 = \mathbf{x}_0$.

i Explain the use of this algorithm as an extension to Algorithm A. Your answer should include the purpose of this algorithm and the benefit of this approach.

[3 marks]

ii Describe the steps required to implement the algorithm described by Equation (2). This should be written in a similar manner to Algorithm A. Estimate the computational cost of a single step of this algorithm compared to a step of the Newton Algorithm.

[3 marks]

iii What are the issues associated with using the algorithm described by Equation (2) with respect to the step size $\Delta\alpha$ and the final computed solution, $\mathbf{x}^M$? Hence explain why the algorithm should be combined with the Newton Algorithm in practice.

[2 marks]

Answer:

+——————

(c) (i) This algorithm can be used to improve convergence of the overall algorithm from a poor initial guess x_0 at step A1 in Algorithm A. [1 mark]

The modified system $G(\alpha, x)$ has a root at x_0 for $\alpha = 0$. [1 mark]

The sequence of steps in (2) leads to an approximation to the root of $F(x) = 0$ at $\alpha = 1$. [1 mark]

(ii)

Algorithm B

$[\mathbf{x}$,$\mathbf{f}]$ = Contin($\mathbf{x}_0, Tol$)

B1. $\mathbf{x} = \mathbf{x}_0$, $\mathbf{f0} = \mathbf{F}(\mathbf{x}_0)$

B2. For $k = 1$ to M

(i). Compute $\mathbf{A} = \mathbf{J}(\mathbf{x})$

(ii). Solve $\mathbf{A}\delta = -\mathbf{f0}$

(iii). Update $\mathbf{x} = \mathbf{x} + \boldsymbol{\Delta}\alpha$

B3. End

[2 marks]

The cost is same - requiring the construction of the Jacobian and the solution of a linear system of equations.

[1 mark]

(iii)

The accuracy of the final solution computed by Algorithm B depends on $\boldsymbol{\Delta}\alpha$. In the form presented here it is $\mathcal{O}(\boldsymbol{\Delta}\alpha)$ accurate. [1 mark]

This final solution should be a good initial guess for the Newton algorithm and improve the convergence properties of that part of the algorithm. [1 mark]

——————+

[20 marks total]

Question 2

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

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(u^2\right) = \epsilon \frac{\partial}{\partial x}\left(u^2 \frac{\partial u}{\partial x}\right), \tag{3}$$

for $x \in [0,1]$ with boundary conditions $u(0,t) = 0$, $u(1,t) = 0$, and $t > 0$ with initial conditions $u(x,0) = U_0(x)$. ϵ is a known, positive constant.

On a uniform grid of m nodes, with nodal spacing h, covering the domain $x \in [0,1]$, we can write a numerical approximation to the PDE (3) at a typical internal node i as,

$$\begin{aligned} \frac{u_i^{k+1} - u_i^k}{\Delta t} &= \frac{1}{h}\left(\frac{(u_{i+1}^{k+1})^2 + (u_i^{k+1})^2}{2}\left(-1 + \frac{\epsilon}{h}\left(u_{i+1}^{k+1} - u_i^{k+1}\right)\right)\right. \\ &\quad \left. - \frac{(u_i^{k+1})^2 + (u_{i-1}^{k+1})^2}{2}\left(-1 + \frac{\epsilon}{h}\left(u_i^{k+1} - u_{i-1}^{k+1}\right)\right)\right) \end{aligned} \tag{4}$$

a Describe the compact finite difference method and the steps required to approximate the PDE (3) in the discrete form (4). [2 marks]

Answer:

+————————

(a)

Considering the PDE in the form

$$\frac{\partial u}{\partial t} = \frac{\partial q}{\partial x}$$

where $q = u^2(-1 + \epsilon\frac{\partial u}{\partial x})$, we can discretise the right hand side as

$$\frac{\partial q}{\partial x} \approx \frac{1}{h}\left(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}\right)$$

[1 mark]

We can then approximate the term $q(x)$ at the half grid points to produce the final discrete form, using

$$\left(\frac{\partial u}{\partial x}\right)_{i+\frac{1}{2}} \approx \frac{1}{h}\left(u_{i+1} - u_i\right)$$

and

$$(u_{i+\frac{1}{2}})^2 \approx \frac{1}{2}\left((u_{i+1})^2 + (u_i)^2\right)$$

[1 mark]

————————+

b State the size of the nonlinear system that would be solved, and the precise form of the solution vector $\mathbf{U}$ that would be required. [2 marks]

Answer:

+————————

(b)

There are $m-2$ unknown solution values on the grid.

[1 mark]

Assuming we number the grid from 1 to m, $\mathbf{U} = (u_2, u_3, \ldots, u_{m-1})$ is the set of $m-2$ unknown solution values.

[1 mark]

————————+

c Describe the algorithm required to advance the model in time. This should include:

- initialisation of the time stepping;
- a suitable initial state for Newton's method at each time step.

[3 marks]

Answer:

+————————

(c)

$t = 0$

Set initial solution U^0 from initial conditions $U_i^0 = U_0(x_i)$

[1 mark]

for k = 0 to maxSteps

$t = t + \Delta t$

Initial guess $U_0^{k+1} = U^k$ [1 mark]

$U^{k+1} =$ Newton(pdeModel, Jacobian, U_0^{k+1}, Tol) [1 mark]

end

————————+

d i Explain why the Jacobian for this nonlinear system has tridiagonal structure. State the precise number of non-zero entries (nz) as a function of the number of equations in your system N.

[3 marks]

ii State which steps of the Newton algorithm can be made more efficient, in terms of memory and CPU time, for a problem with a tridiagonal Jacobian. In each case give a reason for your answer.

[4 marks]

Answer:

+————————

(d)(i)

An equation F_i in this system only depends on 3 solution values: $F_i(u_{i-1}, u_i, u_{i+1}) = 0$

[1 mark]

Hence only Jacobian terms J_{ii-1},J_{ii},J_{ii+1} are non-zero and the system is tridiagonal

[1 mark]

There are $3N - 2$ non-zeros in the Jacobian

[1 mark]

(d)(ii)

Computing the Jacobian matrix at each Newton iteration [1 mark]

The Jacobian matrix can be computed analytically, requiring only the non-zero entries to be computed with $\mathcal{O}(N)$ work. The Jacobian can be stored in a sparse matrix structure requiring $\mathcal{O}(N)$ storage [1 mark]

Solving the linear system $J\delta = -f$ at each Newton iteration. [1 mark]

The system can be solved with a tridiagonal solution algorithm (Thomas algorithm) with $\mathcal{O}(N)$ work. [1 mark]

————————+

e i Describe an efficient numerical approximation to the Jacobian matrix that could be made assuming the tridiagonal sparse structure.

[4 marks]

ii State one advantage and one disadvantage of an analytical form of the Jacobian in this case.

[2 marks]

Answer:

+————————

(e)(i)

A term of the Jacobian can be approximated numerically by the difference

$$J_{ij} = \frac{\partial F_i}{\partial u_j} = \frac{F_i(\mathbf{u} + \epsilon u_j) - F_i(\mathbf{u})}{\epsilon}$$

[1 mark]

Each column of the Jacobian has only 3 non-zero entries hence we can form 3 non-overlapping sets of Jacobian columns [1 mark]

The numerical approximation can be applied to each set as one function evaluation $F(\mathbf{u} + \epsilon u_j)$ where u_j varies depending which Jacobian column we are computing. [1 mark]

The columns of the Jacobian can be extracted from the resulting vector in groups of 3. [1 mark]

(e)(ii)

Advantage: The analytical form would be at least as efficient as the numerical approximation and would be numerically exact up to round-off error. [1 mark]

Disadvantage: The analytical Jacobian terms have to calculated in a problem-specific way. [1 mark]

————————+

[20 marks total]

Question 3

A two-dimensional nonlinear PDE for $u(x,y)$ is defined as

$$\frac{\partial}{\partial x}\left(u^2\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(u^2\frac{\partial u}{\partial y}\right) \;=\; 0 \qquad (5)$$

for the spatial domain $(x,y)\in[0,1]\times[0,1]$. On the boundary of the domain, boundary conditions $u(x,y)=U_b(x,y)$ are known.

A uniform mesh of m nodes is used in each coordinate direction, with nodal spacing h.

Applying standard finite difference approximations in space a possible discretised form of this problem is given by Equation (6).

$$\begin{aligned}\frac{1}{4h^2}\Big(&\left(u_{i+1j}+u_{ij}\right)^2\left(u_{i+1j}-u_{ij}\right)-\left(u_{ij}+u_{i-1j}\right)^2\left(u_{ij}-u_{i-1j}\right)\\&+\left(u_{ij+1}+u_{ij}\right)^2\left(u_{ij+1}-u_{ij}\right)-\left(u_{ij}+u_{ij-1}\right)^2\left(u_{ij}-u_{ij-1}\right)\Big) \;=\; 0 \qquad (6)\end{aligned}$$

where $u_{ij}\equiv u(x_i,y_j)$, $i,j=2,...,m-1$.

a Describe the steps that are required to approximate the PDE (5) in the discrete form (6).

[5 marks]

Answer:

+——————

(a) At grid point ij use the following approximations

$$\begin{aligned}\frac{\partial v}{\partial x} &\approx \frac{v_{i+\frac{1}{2}j}-v_{i-\frac{1}{2}j}}{h}\\ \frac{\partial v}{\partial y} &\approx \frac{v_{ij+\frac{1}{2}}-v_{ij-\frac{1}{2}}}{h}\end{aligned}$$

[1 mark]

In the PDE approximate the outer derivative

$$\frac{1}{h}\left(u^2_{i+\frac{1}{2}j}\left(\frac{\partial u}{\partial x}\right)_{i+\frac{1}{2}j}-u^2_{i-\frac{1}{2}j}\left(\frac{\partial u}{\partial x}\right)_{i-\frac{1}{2}j}+u^2_{ij+\frac{1}{2}}\left(\frac{\partial u}{\partial x}\right)_{ij+\frac{1}{2}}-u^2_{ij-\frac{1}{2}}\left(\frac{\partial u}{\partial x}\right)_{ij-\frac{1}{2}}\right) \;=\; 0$$

[1 mark]

Approximate the inner derivatives

$$\frac{1}{h}\left(u^2_{i+\frac{1}{2}j}\left(\frac{u_{i+1j}-u_{ij}}{h}\right)-u^2_{i-\frac{1}{2}j}\left(\frac{u_{ij}-u_{i-1j}}{h}\right)+u^2_{i+\frac{1}{2}j}\left(\frac{u_{ij+1}-u_{ij}}{h}\right)-u^2_{i-\frac{1}{2}j}\left(\frac{u_{ij}-u_{ij-1}}{h}\right)\right) \;=\; 0$$

[1 mark]

Approximate the terms at half grid points with averages

$$u^2_{i+\frac{1}{2}j} \;\approx\; \left(\frac{1}{2}\left(u_{i+1j}+u_{ij}\right)\right)^2$$

[1 mark]

Rearrange to the required form [1 mark]

——————+

b Deduce the sparse structure of the Jacobian matrix for this problem, stating a realistic bound on the number of non-zero entries in the matrix.

[3 marks]

Answer:

+——————

(b)

An equation F_{ij} depends on 5 solution values $F_{ij}(u_{ij-1}, u_{i-1j}, u_{ij}, u_{i+1j}, u_{ij+1})$

There are $N = (m-2)^2$ equations and a realistic bound is $nz < 5N$ [1 mark]

If a row-by-row numbering system is adopted, the structure of the Jacobian is an $(m-2) \times (m-2)$ array of blocks each of size $(m-2) \times (m-2)$ [1 mark]

The blocks on the diagonal are tridiagonal, first off-diagonal blocks are diagonal, all other blocks are zero. [1 mark]

——————+

c The Jacobian matrix is determined to be numerically symmetric and positive definite.

Describe an efficient iterative solution strategy for the linear equations system at each Newton iteration. [3 marks]

Answer:

+——————

(c)

The Conjugate Gradient method would be the most appropriate choice [1 mark]

Preconditioning should be used to increase efficiency [1 mark]

For this system an Incomplete Choleski decomposition could be used [1 mark]

——————+

d If the discrete system (6) is written in the form $\mathbf{F}(\mathbf{U}) = \mathbf{0}$ describe a pseudo-timestepping solution algorithm for this problem.

State one advantage and one disadvantage of this approach. [6 marks]

Answer:

+——————

(d) The pseudo-timestepping approach can be used to set up a time dependent nonlinear problems

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{F}(\mathbf{U}) = \mathbf{0}$$

[1 mark]

The given initial guess U_0 becomes the initial conditions for the time stepping [1 mark]

Each time step requires the solution of a nonlinear system, but we have an accurate initial state from the previous time step. [1 mark]

Time steps continue until a steady state is reached, which is the solution of the original nonlinear system [1 mark]

Advantage: Removes problems associated with the initial state for Newton's method [1 mark]

Disadvantage: Requires the solution of a series of nonlinear problems rather than just one. [1 mark]

——————+

e How would your answers to parts (b)-(d) change if the three-dimensional form of the PDE (5) was to be solved? [3 marks]

Answer:

+——————

(e)

The sparsity pattern would change. The overall structure would expand to reflect the 3D finite difference grid. [1 mark]

The matrix would still be SPD and preconditioned CG in the form described in (c) could be used. [1 mark]

The pseudo-timestepping approach (d) would still be valid and work in the same way [1 mark]

——————+

[20 marks total]