Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Computational Finite Element Methods

Date: Friday 20th May 2011
Time: 09:45 - 11:45

Please answer any TWO Questions from the FOUR questions provided

For full marks your answers should be concise as well as accurate.
Marks will be awarded for reasoning and method as well as being correct.

This is an OPEN book examination

The use of electronic calculators is permitted.
**Question 1.**

1. Give the formulas for the linear Lagrange interpolation of a function $f(x)$. Give mathematical formulas for the linear nodal basis functions in the finite element method in 1D. 
   [3 marks]

2. Explain how the inter-element continuity of the finite element solution is achieved in 1D case. Give the relation between the nodal and the global basis functions. 
   [3 marks]

3. Consider the following two-point boundary value problem:
   
   \[-y'' + 3y' = 2e^x - 15 \quad \text{on} \ [0,1],\]
   \[y(0) = 1, \ y(1) = e - 5.\]

   a. Derive the weak formulation of the problem and give the test and the trial (solution) spaces that are used in this context. By approximating the infinite-dimensional test and trial spaces by suitable finite-dimensional counterparts, derive the discrete weak formulation. Then, assuming general finite-dimensional test and trial spaces with the basis set $\{\phi_i\}_{i=1}^N$, derive the linear algebra problem. 
   [5 marks]

   b. Divide the interval $[0,1]$ into 3 finite elements of equal length. Adopt the global piecewise linear basis functions for the finite-dimensional basis set in the discrete weak formulation. With this choice, compute the element Galerkin matrices, and the element right-hand side vectors. Then assemble the global Galerkin matrix and the global right-hand side vector by performing local-to-global mapping. 
   [10 marks]
**Question 2.**

Consider the following transient problem: find \( u \in C^2(\Omega) \times C^1(\tau) \) satisfying:

\[
\frac{\partial u}{\partial t} - \Delta u = f(t) \quad \text{in } \Omega \times \tau = [0,1]^2 \times [0,T],
\]

\[
u(x,y) = 0 \quad \text{for } (x,y) \in \partial \Omega, \quad \forall t \in [0,T],
\]

\[
u(x,y) = u_0(x,y) \quad \text{for } (x,y) \in \Omega, \quad t = 0.
\]

1. Derive the Galerkin approximation of the problem expressed as a system of differential algebraic equations. Explain the difference between the discrete problems obtained as a result of this process in the cases of the stationary \( \frac{\partial u}{\partial t} = 0 \) and the transient version of the problem (1).

   [10 marks]

2. Present the backward Euler’s scheme for the solution of a system of differential algebraic equations obtained in part 1.

   [5 marks]

3. Explain the concept of adaptivity in a differential algebraic equations solver and give the details of the adaptive backward Euler scheme (in particular, describe the heuristics for adaptive step selection).
Question 3.

Consider a finite element procedure for the solution of a model-problem: find $u \in C^2(\Omega)$ satisfying:

$$-\Delta u = f \quad \text{in } \Omega = [0,1]^2 \subset \mathbb{R}^2,$$

$$u = u_D \quad \text{on } \partial \Omega_D, \quad \frac{\partial u}{\partial n} = u_N \quad \text{on } \partial \Omega_N,$$

where $\partial \Omega_D \cap \partial \Omega_N = \emptyset$, $\partial \Omega_D \cup \partial \Omega_N = \partial \Omega$, $\int_{\partial \Omega} ds > 0$.

1. Describe the data structures produced by a typical mesh generator and the information they contain that is used for the assembly of the Galerkin system. [3 marks]

2. Write a pseudo-code for assembling the Galerkin system in the case of a given model-problem. In your answer assume that the information from the mesh generator is given in the format you described in part 1. Assume also that the element matrices are assembled first through procedure consisting of three nested loops, followed by the assembly of the global Galerkin system from the element matrices by the application of the local-to-global mapping. Outline some ideas how this process can be parallelised. [5 marks]

3. Discuss the merits of different loop orderings in the algorithm for the assembly of Galerkin’s element matrices. Summarise the effect that different loop orderings may have on the execution performance. Restrict your answer to comparing the two loop orderings: when the loop over all elements is the outermost, and when this loop is the innermost in the three-loop nested structure. For the storage of the element Galerkin matrices consider a tensor (multi-dimensional array) data structure $A_e(n_r, n_c, E)$, where $A_e$ of size $n_r \times n_c$ is the element Galerkin matrix for $e = 1, \ldots, E$. Sketch a picture of this data structure and highlight different access patterns to its elements obtained from two different loop orderings discussed above. Note that multidimensional arrays are stored in the Fortran setting as one-dimensional arrays in a generalised column-wise fashion (the leftmost indices change the fastest). With this in mind, associate with each of the two loop orderings the optimal shape and index orderings of the tensor $A_e$, so that the assembly algorithm access the data in memory in a continuous fashion. [8 marks]

4. Discuss different approaches for including Dirichlet boundary conditions into a Galerkin matrix. Include in your answer the case when the boundary conditions are included in the right-hand side vector as well as the cases when the boundary conditions are imposed at the local or global Galerkin matrix level. [4 marks]
Question 4.

1. Explain the main problem with robustness of the fixed-point iterative methods. Include in your answer the discussion on smooth and oscillatory components of the solution error. How can this problem be rectified in cases when the solution of large, sparse linear systems obtained from the discretisations of second-order elliptic partial differential equations is needed? Assume in your answer that a nested sequence of grids, matrix representations of the problem on these grids and the means of communication between the grids are available.

   [8 marks]

2. Based on the discussion from part a), define the V-cycle of multigrid and explain all the components of the algorithm. Give a pseudo-code that implements a recursive definition of the multigrid V-cycle.

   [5 marks]

3. Explain how the transfer of information (the residual and the correction to the solution) between different levels in the multigrid hierarchy is achieved. Assuming linear interpolation, explain a general procedure, based on full weighting, for assembling the interpolation matrix. Then give an interpolation matrix for the following grid patch:

   END OF EXAMINATION