UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR11174 NUMERICAL ALGORITHMS FOR HIGH PERFORMANCE COMPUTING

Tuesday 15 th December 2020

13:00 to 16:00

INSTRUCTIONS TO CANDIDATES

1. **Note that ALL QUESTIONS ARE COMPULSORY.**
2. **EACH QUESTION IS WORTH 25 MARKS. Different sub-questions may have different numbers of total marks. Take note of this in allocating time to questions. Different sub-questions may have different numbers of total marks. Take note of this in allocating time to questions.**
3. **THIS EXAMINATION IS AN OPEN-BOOK ASSESSMENT. You may refer to material from your notes, course material, or beyond to assist you. You should not copy any text or images into your answer however as your answer must remain your own work. If you refer to material from outside the course it must be referenced properly.**
4. **THIS IS A REMOTE EXAMINATION. As stated in the Own Work Declaration, for the duration of the assessment you must not communicate with any other person about your work either electronically or by word or sign, nor let your work be seen by any other person.**
5. **Please refer to guidance in Learn under Examination Information if you have any difficulties.**

EPCC Courses

Convener: M. Mistry
External Examiners: Matt Probert

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

1. A system of N simultaneous equations can be expressed as a dense linear system

$$Ax = b$$

where A is an $N \times N$ matrix of known coefficients, b is a vector of length N of known values, and x is a vector of length N of unknown values to be determined. Such a system can be solved via LU factorisation where the main computational steps involved are: (i) the factorisation of the matrix itself, followed by (ii) forwards substitution and finally (iii) backwards substitution to find the solution, x.

(a) Explain why decomposing the matrix into L and U factors helps when solving the system. [*5 marks*]

(b) State the computational cost of each of the above 3 steps, as well as the overall cost, to find x. The cost should be stated as a function of N using "Big O" notation. [*3 marks*]

(c) How you would expect the computational costs involved would change in the following scenarios? Each change should be considered separately and you should explain your reasoning. You should assume here that N is large but can easily fit in the available memory.

i. The number of simultaneous equations is doubled.

ii. Using the same coefficients, the problem is solved for 5 right-hand sides rather than just 1.

iii. The problem is solved for 2 different sets of known coefficients rather than 1 set. [*8 marks*]

(d) When carrying out the LU factorisation using the LAPACK library, partial pivoting is usually used and the library returns an array containing information describing the pivoting performed.

i. Why would you choose to use a library rather than implement your own code? [*3 marks*]

ii. Explain what partial pivoting is and why it is used. [*4 marks*]

iii. Why is it useful that the library returns information about the pivoting carried out on the matrix? [*2 marks*]

2. The discrete Fourier transform (DFT) F_n for $0 \leq n < N$ of a set of N complex points f_k is given by

$$F_n = \sum_{k=0}^{N-1} f_k e^{2\pi i k n/N}. \tag{1}$$

(a) The DFT algorithm has a complexity of $O(N^2)$. State the complexity of the FFT algorithm and, outlining the algorithm, describe how this is achieved. [*5 marks*]

(b) FFTW is a widely-used library enabling programmers to easily implement fast Fourier Transforms in code they are writing.

i. The FFT is in general a complex-to-complex transform i.e. both the f_k and F_n above are complex. However, the most common transforms in practice are real-to-complex. Remember that a complex number is stored as two doubles representing the real and imaginary components. What approaches might an FFT library such as FFTW provide to assist you (the programmer) in performing this calculation efficiently? You don't need to give mathematical details here. [*4 marks*]

ii. The FFTW planner creates an efficient FFT algorithm tailored to the transform at hand. When creating a plan, FFTW must apply a level of rigour through a flag provided to the planner routine. The two most commonly used flags are `FFTW_ESTIMATE` and `FFTW_MEASURE`.

Describe what each of these mean in the context of a plan's creation and how you would decide on which one to use. Describe *FFTW wisdom* and in what circumstances you might use it. [*6 marks*]

(c) When designing a parallel FFT algorithm of dimensionality $d > 1$, the best approach is to parallelise over at most d–1 dimensions. Why is this? [*4 marks*]

(d) With the above in mind, describe the operations needed to efficiently calculate in parallel the Fourier transform of a 2D array. You should take care to indicate how the data is decomposed as well as any transpose and 1D FFT operations required. [*6 marks*]

3. Methods such as Conjugate Gradient (CG) and Bi-conjugate gradient with stabilisation (BiCGstab) are examples of Kylov subspace methods. Both are *iterative methods* and are often used to solve large sparse linear systems of the form

$$Ax = b$$

where A is a matrix of size $N \times N$ with a small number of known non-zero values per row, and b and x are known and unknown vectors respectively, each of size N.

(a) Explain why an iterative method might be used to solve such a sparse system rather than a direct method such as LU factorisation. [*4 marks*]

(b) The most computationally expensive step within the CG or BiCGstab methods is the matrix-vector multiplication which is carried out at least once at every iteration. For this reason it is often parallelised. The standard approach is to distribute the rows of the matrix across processors but there is a choice as to how to store the vectors.

Explain the typical approaches taken to storing the vectors and in each case explain using pseudocode and/or diagrams where necessary the steps involved to perform the matrix-vector multiplication in parallel making it clear where operations are carried out in parallel and where communications between processors are necessary. [*16 marks*]

(c) Explain the advantages and disadvantages of each of the above approaches and in which cases you would choose each one. [*5 marks*]