Technische Universität München
Institut für Informatik
Prof. Dr. Thomas Huckle
Matous Sedlacek

WT 2010/2011

04.02.2011

Parallel Numerics

Exam

A handwritten sheet of paper (size A4, front and back page) may be used during the exam as mnemonic as well as the MPI operation reference distributed during the tutorials. No other material is allowed. The exam is to be solved within 90 minutes, and the answers are to be written in German or English. Please try to answer all parts of the questions precisely and briefly. For passing the exam you will need 17 out of 40 credits. The exam consists of six problems on three pages.

Problem 1: Flynn's Taxonomy (≈ 4 **credits)**
Describe the classifications of computer architectures defined by Michael J. Flynn in detail and give at least one example for each classification.

Problem 2: MPI Send/Recv (≈ 7 **credits)**
Consider the following incomplete code fragment (arrays are correctly initialized with values):

```
int a[10], b[10], npes, myrank;
...
MPI_Comm_size(MPI_COMM_WORLD, &npes);
MPI_Comm_rank(MPI_COMM_WORLD, &myrank);
MPI_Send(a, 10, MPI_INT, (myrank+1)%npes,1,MPI_COMM_WORLD);
MPI_Recv(b, 10, MPI_INT, (myrank-1+npes)%npes,1,MPI_COMM_WORLD);
...
```

a) What topology fits best to the given source code? (≈ **1 credit**)

b) Describe shortly what the given source code does when executed on more than 1 processor on a parallel computer? (≈ **1 credit**)

c) Describe whether the code will work correctly (is safe) if MPI uses a buffer for the send operation? (≈ **1 credit**)

d) Assume MPI uses no buffer for the send operation but the program is executed on 2 processes only. Describe whether the code fragment is safe for this setting? (≈ **1 credit**)

e) Rewrite the above code fragment such that it is safe. Use only blocking routines, i.e. `MPI_Send()` and `MPI_Recv()`. Hint: partition the processes into two groups. Modify the source code such that the program will work correctly in send mode with or without buffering (≈ **3 credits**):

Problem 3: Speedup (≈ 4 **credits)**

a) Describe in words Amdahl's law and Gustafson's law for the speedup. What are the differences between the two and what do they have in common with weak speedup and strong speedup? (≈ **2 credits**)

b) Give the formula for Amdahl's law for the speedup S and describe all variables. (≈ **1 credit**)

c) Assume you have a program which has a parallel fraction of 80%. Give the maximum attainable speedup S, according to Amdahl, for an infinite number of processors, i.e. $p \to \infty$. ($\approx$ **1 credit**)

Problem 4: Hockney/Golub Method ($\approx$ **7 credits**)

a) We consider tridiagonal systems of linear equations. Assume the system $Ax = d$ with

$$\begin{pmatrix} b_1 & c_1 & 0 \\ a_2 & b_2 & c_2 \\ 0 & a_3 & b_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \end{pmatrix}. \tag{1}$$

The system (1) can be solved by the Hockney/Golub method which combines the equations recursively to get a final system with a diagonal matrix. Therefore, a certain number of steps has to be executed. In the first combination step of the equations, the matrix A changes to matrix $A^{(1)}$. Write down the matrix $A^{(1)}$! ($\approx$ **2 credits**)

b) Explain how the Hockney/Golub method can be parallelized for (1) among $p = 3$ processors. Why and between which steps communication becomes necessary? It is sufficient to give an explanation in words of how each processor will compute which components. It is not explicitely requested to give all formulas by applying the Hockney/Golub method to (1). ($\approx$ **3 credits**)

c) Why is the Hockney/Golub method not applicable to the system $Bx = d$ with

$$B = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & -1 \\ 0 & 1 & 1 \end{pmatrix}?$$

For which class of matrices it is possible to use the Hockney/Golub method? ($\approx$ **2 credits**)

Problem 5: Sparse matrices ($\approx$ **7 credits**)
Given is the sparse matrix $C \in \mathbb{R}^{4\times 4}$ with

$$C = \begin{pmatrix} -1 & 2 & -1 & 0 \\ 3 & 0 & -1 & 0 \\ 2 & 0 & 0 & 0 \\ 3 & 1 & 0 & 4 \end{pmatrix}.$$

a) Give the Jagged Diagonal storage format for the matrix C. ($\approx$ **2 credits**)

b) Give the the Adjacency matrix and the graph of C, i.e. give $A(G(C))$ and $G(C)$. ($\approx$ **2 credits**)

c) Consider the matrix-vector multiplication $f = E \cdot d$ with $E \in \mathbb{R}^{n\times n}$ and $f, d \in \mathbb{R}^n$. The matrix E is given in compressed sparse column format (CSC). Give a pseudocode to compute the vector f using the storage format of E, i.e. use the float array `AE` and the integer arrays `JE`, and `IE` in your pseudocode. Identify and discuss possible SAXPY and GAXPY operations in your code. ($\approx$ **3 credits**)

Problem 6: Graphs and colouring (≈ 11 **credits**)

Given is the symmetric update

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \\ f_4(x) \\ f_5(x) \\ f_6(x) \end{pmatrix} = Ax = \frac{1}{5} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} x.$$

where

$$x^{(k+1)} = f(x^{(k)}) \tag{2}$$

reflects the dependency between the update $x^{(k+1)}$ and the old solution $x^{(k)}$ in an iterative algorithm. $A \in \mathbb{R}^{6\times 6}$ and $x \in \mathbb{R}^6$.

a) Give the corresponding graph of f which describes the data dependency in the update (2), i.e. give $G(f)$. (≈ **2 credits**)

b) Give and explain a minimum colouring of $G(f)$ such that the update can be performed in parallel. How many steps have to be performed in parallel for the update? (≈ **2 credits**)

c) Describe a renumbering of $G(f)$ according to the minimum coloring of $G(f)$ where the vertices of one color are numbered consecutively. Apply this permutation only to the first and second row and first and second column of A and give the upper left 2×2-block of the permuted matrix $\tilde{A}$, i.e. give $\tilde{A}(1{:}2,1{:}2)$. Explain how to read the parallelism out of the block $\tilde{A}(1{:}2,1{:}2)$. (≈ **3 credits**)

d) Give all nontrivial fixpoints of $x = Ax$ (besides $x = \vec{0}$). (≈ **4 credits**)