Technical University of Munich
Department of Informatics
Prof. Dr. Thomas Huckle
Dr. Benjamin Uekermann

Winter term 2017/18

Parallel Numerics

Exercise 7: Sparse Matrix–Vector Multiplication & Red-Black Colouring

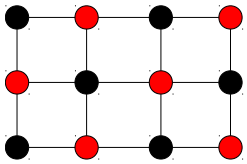
1 Red-Black Colouring

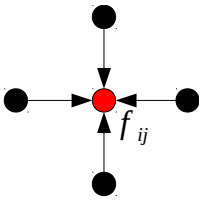
The idea of any colouring approach for an iterative algorithm is to assign evaluations to different computing steps (colours). Afterwards, these steps are executed sequentially.

a) What are typical properties of elements of one colour?
All elements of one colour work with "same old" data. The operations of one colour should be evaluated in parallel.

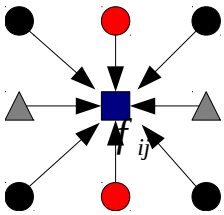
b) Derive a colouring scheme for a 5-point stencil on a cartesian two-dimensional grid.
Unknowns are arranged in cartesian grid u_{ij} and are updated via function f_{ij}. Function f_{ij} depends on left (f_{ij-1}), right (f_{ij+1}), upper (f_{i+1j}), and lower (f_{i-1j}) neighbour. Using red-black ordering:

- red phase: update all unknowns (evaluate all f_{ij}) belonging to red points. The evaluation can be done completely in parallel.
- black phase: use new values of red phase.
- $\rightarrow$ alternate phases until convergence.





c) Why is bicolouring for a 9–point stencil on a cartesian two-dimensional grid not well-suited? Derive an improved colouring.
No parallel evaluation possible with two colours $\rightarrow$ sequential algorithm. Use four colours instead.



2 Sparse Matrix-Vector Multiplication

The solution of the matrix-vector product $A \cdot b$ with a sparse matrix $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ should be calculated. Write different procedures for the calculation of the matrix-vector multiplication. Implement the storage schemes for matrix A that have been discussed in the lecture:

1. storage in coordinate form
2. compressed sparse row format (CSR)
3. compressed sparse column format (CSC)
4. CSR with extraction of the main diagonal entries
5. diagonal–wise storage (for band matrices)
6. rectangular, row–wise storage scheme
7. jagged diagonal form

Test your routines for large sparse matrices whose elements are randomly distributed as zeros or as non-zero values. Compare the computation time to the time needed by applying the full matrix storage scheme.
See sourcecode to corresponding tutorial on webpage.