September 13, 2017

Matlab Assignments – Lecture 10, Fall 2016

In these assignments we will investigate how we can parallelise ILU and SSOR.

Download the file `precon.m` from the course webpage. The matrix $\mathbf{A}$ in this code is from the Poisson equation discretised on a uniform $n \times n$ grid (type `help private/poisson` in Matlab for more information).

Assignment 10.1.

(a) Add to the script `precon.m` code to create an SSOR preconditioner (without relaxation parameter. Your preconditioner must have the form $\mathbf{M} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$. Use this preconditioner to solve the system with the MATLAB routine `pcg`. Make sure your implementation is efficient. (How does MATLAB handle the preconditioner: left-explicit, right-explicit, ..., implicit? Note that the MATLAB documentation explains that effectively left-explicit preconditioning is used but does not explain what is meant by "effectively". Inspect the code. What are the advantages of 'MATLAB's approach'? MATLAB offers the possibility to include one preconditioner $\mathbf{M}$ in `pcg`, but it is also possible to include two preconditioners $\mathbf{M}_1$ and $\mathbf{M}_2$. What is the advantage of this second possibility?) Investigate how the number of iterations to solve the system $\mathbf{A}\mathbf{x} = \mathbf{b}$ depends on n, the number of gridpoints in each direction.

(b) Same assignment, but now use the incomplete Cholesky preconditioner without fill-in. Use the MATLAB routine `ichol` (replaces `cholinc`) to compute the preconditioner.

(c) The forming of the preconditioners costs time as well. How does that affect the performance? (Is it important to have sparse $\mathbf{A}$ and $\mathbf{L}$?)

Assignment 10.2.

(a) Make a red-black ordering (or checker board ordering) of the unknowns, that is, of the gridpoints. This ordering splits the unknowns into two groups such that none of the unknowns within a group has a direct neighbour in the group. The resulting matrix has the block form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

in which $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$ are diagonal matrices.

(b) Make an SSOR preconditioner on basis of the reordered matrix. Compare its performance with the SSOR preconditioner that you used in the first assignment. Why is this preconditioner better suited for parallel computing than the one you used in assignment 1?

(c) Same questions, now for the incomplete Cholesky preconditioner. Can you relate incomplete Cholesky to SSOR both for the checker board ordering?