LARGE AND SPARSE MATRIX PROBLEMS, 2014

HOMEWORK ASSIGNMENT number 3

Well performed this homework assignment gives 1 credit point

To be handed in by February 28 at the latest

Exercise HA3 a. Confirm by pen and paper calculations:
- the equation (6.58) in the book.
- that the matrix ZMZ (page 347) has nonzeros only on its main diagonal and perdiagonal. In order to do this you may use (6.60) without proof.

Exercise HA3 b. Take a reasonable order of the matrix M and show by implementing in MATLAB that it has the eigenvalues stated in Theorem 6.11.

COMPUTER EXERCISE number 3

To be handed in by February 28 at the latest

Exercise CE3. We test the multigrid method on the model 2D Poisson problem. For this purpose several MATLAB programs are available on the course web site. The program *makemgdemo.m* makes up data for the test problems of size n*n, where n is $2^k + 1$, for different levels k of meshes, to your choice. You may choose among three different right hand sides, corresponding to a true smooth solution, corresponding to the tent-solution in CE2 or a very smooth right hand side. You may select the different cases by simply commenting away lines in the code. Here you also specify the number of weighted Jacobi-iterations and the convergence tolerance. The routine *testfmgv.m* makes a number of iterations using full multigrid. You get tables of results as well as graphics of the solution and error. The functions *fmgv.m*, *mgv.m* and *mgvrhs.m* are the routines that do the job.
For grade 3: Take n as large as the computer can master with reasonable cpu-time. Study the different right hand sides. Compare the behavior for the smooth and non-smooth cases. Also try to find out the optimal number of weighted Jacobi-iterations before and after the multigrid V-cycle. Also try to compare multigrid with the pcg methods in CE2.
Additional for grade 4: Study a less suited problem for multigrid. Replace the Poisson problem by $-\delta u''_{xx} - u''_{yy} = f$, i.e. with $\delta = 1$ it is the standard Poisson problem. Try small values of δ like 10^{-2} or 10^{-3}. You should expect an increase in the number of required Jacobi-iterations. Compare the efficiency with the pcg method with IC or MIC factorizations.
Additional for grade 5: For the case $\delta = 10^{-3}$, find out if the IC(2) factorization could be a better smoother. Compare the number of smoothing iterations required for this method and the classical weighted Jacobi method.