PhD-course on Iterative Methods for Linear Systems of Equations

Practical assignments day 4

In the fourth assignment we will test several Bi-CG methods and IDR(s) on an ocean circulation problem.

1. Download the files `ocean.m, topo.mat` and `ocean.mat` from the course webpage. The file `ocean.mat` contains the matrix A, the right-hand side vector b and a matrix P. These files define an ocean circulation model on a grid with a resolution of 3^0. P maps every entry in the solution vector x onto a corresponding gridpoint on the map of the world.
   Run the script `ocean`. You will see the nonzero structure of the matrix A. What is the size of the matrix? Is A symmetric?

2. Use the MATLAB routines `bicg` and `qmr` to solve the ocean system. Use diagonal scaling (Jacobi preconditioning) as preconditioner and as tolerance $\|r\|/\|b\| < 1e^{-8}$. Plot the convergence of the two methods in one figure. Comment on the (difference in convergence) of the two methods. Also compare the computing time (measure with `tic` and `toc`).

3. Use the MATLAB routines `cgs` and `bicgstab` to solve the ocean system. Use diagonal scaling Use diagonal scaling (Jacobi preconditioning) as preconditioner and as tolerance $\|r\|/\|b\| < 10^{-8}$. Plot the convergence curves of the two methods in one figure. **Note that `bicgstab` gives two residual norms per iteration (one per matvec), whereas `bicg`, `qmr` and `cgs` give one residual norm per iteration (= per two matvecs).** Comment on the (difference in convergence) of the two methods. Also compare the computing times (measure with `tic` and `toc`). Are these methods faster than Bi-CG and QMR?

4. Download `idrs.m` from `http://ta.twi.tudelft.nl/nw/users/gijzen/IDR.html`. Solve the ocean system with IDR(1), IDR(2), IDR(4), and IDR(8). Use diagonal scaling (Jacobi preconditioning) as preconditioner and as tolerance $\|r\|/\|b\| < 10^{-8}$. Plot the convergence curves of the methods in one figure. **Note that `idrs` gives a residual norm per matvec (=iteration).** Also compare the computing times (measure with `tic` and `toc`).

5. Compare the performance of the different methods. Which method is fastest? Which method uses the least amount of memory?

6. Also solve the system with GMRES with Jacobi preconditioning. You can do this by either extending your GMRES code with preconditioning, or by pre-multiplying the system with the inverse of the main diagonal. Compare the performance of GMRES with that of the other methods.

7. Use the MATRIX-MARKET matrices to make a more comprehensive comparison between the different methods.