Exercise 9: MATLAB's GMRES and MINRES

The goal of this assignment is to think about and experiment with basic Krylov space solvers.

1. Give the arithmetic complexities and the memory consumptions of the three algorithms GMRES, GMRES(m), and MINRES, all with preconditioning.

   Hint: The GMRES(m) algorithm is in lecture 9, but you need to write down the algorithms of GMRES and MINRES by yourself.

   Hint: The most expensive portions of the GMRES/MINRES solver without preconditioning are the matrix-vector multiplication and the Gram–Schmidt Orthogonalization. If there is preconditioning, then solving with the preconditioner, i.e., $M\boldsymbol{z}_k = \boldsymbol{r}_k$, should also be added to get the total arithmetic complexity.

2. Construct a nonsingular matrix $A \in \mathbb{R}^{n\times n}$ and an initial vector $\boldsymbol{b} \in \mathbb{R}^n$ such that the Arnoldi algorithm breaks down after 3 steps. n must be bigger than 3.

3. We experiment with MATLAB's `gmres` solver which can be invoked by

   ```
   [x,flag,relres,iter,resvec] = gmres(A,b,restart,tol,maxit);
   ```

   We use a matrix from the Matrix Market, a repository of test data for use in comparative studies of algorithms for numerical linear algebra.

   Download the matrix `fs_680_3.mtx` from `http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/facsimile/fs_680_3.html` and read it into MATLAB:

   ```
   A = mmread('fs_183_6.mtx');
   ```

   (`mmread` is available from `math.nist.gov/MatrixMarket/mmio/matlab/mmread.m`.)

   First solve the system without preconditioner with `b=ones(n,1)`, `tol=1e-6`, `maxit=10` (what does this mean?). Check `flag` to see if the system was solved correctly. `restart` has to be sufficiently big.

   Then solve the system with the Gauss–Seidel preconditioner.

   Plot the convergence histories (in `resvec`) of the two solutions on top of each other.

4. MATLAB provides a large number of interesting test matrices that are worth exploring. An overview can be obtained by typing help gallery in the command window. Today, let us play with the `TOEPPEN` matrix, see the excerpt below.

   ```
   >> help private/toeppen
    toeppen Pentadiagonal Toeplitz matrix (sparse).
       P = GALLERY('toeppen',N,A,B,C,D,E) takes integer N and
       scalar A,B,C,D,E. P is the N-by-N sparse pentadiagonal Toeplitz
       matrix with the diagonals: P(3,1)=A, P(2,1)=B, P(1,1)=C, P(1,2)=D,
   ```

Please submit your solution via e-mail to Peter Arbenz (`arbenz@inf.ethz.ch`) by November 23, 2017. (12:00). Please specify the tag **FEM17** in the subject field.

```
P(1,3)=E.

Default: (A,B,C,D,E) = (1,-10,0,10,1) (a matrix of Rutishauser).
This matrix has eigenvalues lying approximately on the line segment
2*cos(2*t) + 20*i*sin(t).
```

(a) Generate a sample of Toeppen's matrix with

```
A = gallery('toeppen',30000, 2, 3, 8, 3.5, 4.5);
```

Check that A is a sparse nonsymmetric matrix.

(b) Use the syntax

```
[x,flag,relres,iter]=gmres(A,b,restart,tol,maxit,M);
```

Compare the performance (by means of the total number of iterations and execution time) of GMRES when there is no preconditioner, with Jacobi preconditioner, and with Gauss–Seidel preconditioner, respectively. Set `tol = 1e-9`.

We explore the impacts of the restarting technique on the performance of MATLAB's `gmres` solver. To this end, let the parameter `restart` be 1, 2, 4, 6, 8, 16, and 32, respectively, and calculate the total number of iterations for each case by `(iter(1) - 1)`×`restart + iter(2)`. (Why?). Also measure the execution time of the GMRES method for each case with the commands `tic` and `toc`. Build a table to show how the total number of iterations and the execution time change with respect to the parameter restart and the three preconditioning approaches. What do you observe? How does the memory consumption vary with respect to restart? And how does the execution time vary with respect to restart? Can you determine the optimal restart for the different preconditioning approaches for this matrix?

(c) Note that MATLAB provides the source code for its iterative solvers. One can look at it using the type command. Try `type gmres` and try to understand the implementation as much as possible. Pay attention to implementation details that might not have been treated in class.

5. Now, try to solve $-\Delta u - 60u = f$, on a square with homogeneous Dirichlet boundary condition. The stiffness matrix is symmetric but not positive definite, so that the CG method cannot be applied. (How do you check if A is indefinite? What happens if you call `pcg`?) In order to solve the system we use the Minimum Residual Method, MINRES, preconditioned by Jacobi and (incomplete) Cholesky factorization. For the latter compute the (incomplete) Cholesky factorization of the operator corresponding to $-\Delta u$. If the (incomplete) Cholesky factor is L, the preconditioner is $M = LL^T$. Read the documentation of the following commands: `minres`, `chol`, and `ichol`. Again, provide and discuss iteration counts and execution times.

Please submit your solution via e-mail to Peter Arbenz (`arbenz@inf.ethz.ch`) by November 23, 2017. (12:00). Please specify the tag **FEM17** in the subject field.