MAT 226B, Winter 2018

Homework 4

(due by Wednesday, March 7, 11:59 pm)

General Instructions

- You are required to submit your homework by uploading a single pdf file to Canvas. Note that the due dates set in Canvas are hard deadlines. I will not accept any submissions outside of Canvas or after the deadline.

- If at all possible, use a text processing tool (such as LaTeX) for the preparation of your homework. If you submit scanned-in hand-written assignments, make sure that you write clearly and that you present your solutions in a well-organized fashion. If I cannot read your homework, I will not be able to grade it!

- Feel free to discuss the problems on the homework sets with other students, but you do need to submit your own write-up of the solutions and your own MATLAB codes. If there are students with solutions that were obviously copied, then each involved student (regardless of who copied from whom) will only get the fraction of the points corresponding to the number of involved students.

- Test cases for computational problems are often provided as binary Matlab files. For example, suppose the file "`LS.mat`" contains the coefficient matrix A and the right-hand side b of a system of linear equations. The Matlab command "`load('LS.mat')`" will load A and b into Matlab.

- When you are asked to print out numerical results, print real numbers in 15-digit floating-point format. You can use the Matlab command "`format long e`" to switch to that format from Matlab's default format. For example, the number 10π would be printed out as `3.141592653589793e+01` in 15-digit floating-point format

- When you are asked to write Matlab programs, include printouts of your codes in your homework.

---

1. Let $A \in \mathbb{R}^{n\times n}$ be a matrix of the form

$$A = \begin{bmatrix} 0 & 1 \\ I & a \end{bmatrix},$$

where $a \in \mathbb{R}^{n-1}$ and I denotes the $(n-1)\times(n-1)$ identity matrix. Let $b = e_1$ be the first unit vector of length n.

(a) Determine the Krylov subspaces $K_k(A, b)$ for all $k = 1, 2, \ldots, d(A, b)$ and show that $d(A, b) = n$.

(b) How many iterations does the MR method with initial guess $x_0 = 0$ need to find the solution of $Ax = b$?

(c) Show that the matrix $A^T A$ has at most 3 distinct eigenvalues.
Hint: The matrix $A^T A$ can be written as the sum of the identity matrix and a matrix of rank 2.

(d) Let $c \in \mathbb{R}^n$ and $x_0 \in \mathbb{R}^n$. Give a sharp upper bound for the number of iterations the CGNE method with initial guess x_0 needs to find the solution of $Ax = c$.

2. Let $A \in \mathbb{R}^{n\times n}$ with $A \succ 0$ and $E \in \mathbb{R}^{n\times n}$ with $E = E^T$ be given matrices such that E has rank r and

$$M := A + E \succ 0.$$

We use $M = M_1 M_2$, where both M_1 and M_2 are assumed to be $n \times n$ matrices, as a preconditioner to solve $Ax = b$ with the preconditioned CG method.

(a) Prove that the preconditioned matrix $A' = M_1^{-1} A M_2^{-1}$ has at most $r + 1$ distinct eigenvalues.

(b) What is the maximum number of iterations needed for the preconditioned CG method (run in exact arithmetic) to find the solution of $Ax = b$?

3. Matlab has a built-in function "`gmres`" for solving systems of linear equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$. The function can be used to run GMRES or restarted GMRES by setting the input parameter `RESTART` accordingly: `RESTART` $=$ [] runs GMRES and `RESTART` $= k_0$ runs restarted GMRES with restart parameter k_0. One of the possible outputs of the routine is the vector `RESVEC` that contains the Euclidean norms of all the residual vectors produced in the course of each run of the algorithm. Note that the values in `RESVEC` are the residual norms, $\|r_k\|_2$, and so in order to get the relative residual norms, one has to divide each entry of `RESVEC` by $\|r_0\|_2$.

Use the built-in GMRES routine to write a Matlab program that lets you run (full) GMRES or restarted GMRES (with restart parameter k_0) for any system (1), using the zero vector $x_0 = 0$ as initial guess. As output, your routine should produce the complete history of all the relative residual norms

$$\varrho_k := \frac{\|r_k\|_2}{\|r_0\|_2} \tag{2}$$

produced during each run, the final approximate solution x_k of (1), and the total number of matrix-vector products $q = Av$ computed during each run.

For the linear system given in the Matlab file "`HW4_Problem3.mat`", use your Matlab program to solve (1) to relative residual norm of

```
tol = 1e-9.
```

Run full GMRES, as well as restarted GMRES with restart parameters

$$k_0 = 2,\ 5,\ 10,\ 20,\ 50,\ 100,$$

all without using preconditioning. Produce a single graph that shows

$$\log \varrho_k, \quad k = 0, 1, \dots,$$

for all your runs. Also submit a list showing the total number of matrix-vector products for all your runs.

4. Let $A \in \mathbb{R}^{n\times n}$ be a general nonsingular matrix written as

$$A = D_0 - F - G, \tag{3}$$

where D_0, $-F$, and $-G$ denotes the diagonal part, strictly lower-triangular part, and strictly upper-triangular part of A, respectively. Let $D \in \mathbb{R}^{n\times n}$ be a given nonsingular diagonal matrix and consider the SSOR-type preconditioner

$$M := (D - F)D^{-1}(D - G) = M_1 M_2, \quad M_1 := (D - F)D^{-1}, \quad M_2 := D - G,$$

for the matrix A. We denote by

$$A' := M_1^{-1} A M_2^{-1} \tag{4}$$

the corresponding preconditioned matrix.

(a) Show that

$$A' = D\Big(\big(D - G\big)^{-1} + \big(D - F\big)^{-1}\Big(I + D_1\big(D - G\big)^{-1}\Big)\Big), \tag{5}$$

where $D_1 := D_0 - 2D$.

(b) Use the formula (5) to derive an algorithm that computes matrix-vector products

$$q' = A'v', \quad v' \in \mathbb{R}^n,$$

as efficiently as possible.
Hint: Your algorithm should only involve one triangular solve with $D - G$, one triangular solve with $D - F$, one multiplication with the diagonal entries of D, one multiplication with the diagonal entries of D_1, and two SAXPYs.

(c) Let m denote the number of nonzero off-diagonal entries $a_{jk} \neq 0$, $j \neq k$, of A. Assuming that $v' \in \mathbb{R}^n$ is a vector with all nonzero entries, give an exact flop count (in terms of n and m) for computing $q' = A'v'$ with your algorithm from part (b).

5. (a) Use Matlab's function "`gmres`" to write Matlab programs for each of the following algorithms for solving linear systems (1):

   (i) GMRES (without preconditioning);

   (ii) Restarted GMRES (without preconditioning);

   (iii) GMRES with diagonal preconditioning applied from the right, i.e.,

   $$M_1 = I \quad \text{and} \quad M_2 = D_0,$$

   where D_0 denotes the diagonal part of A, see (3);

   (iv) Restarted GMRES with diagonal preconditioning applied from the right;

   (v) GMRES with the SSOR-type preconditioner derived in Problem 4;

   (vi) Restarted GMRES with the SSOR-type preconditioner derived in Problem 4.

   Your algorithms for (v) and (vi) should apply GMRES directly to the preconditioned matrix (4) and use your algorithm from Problem 3(c) to compute matrix-vector products with A'.

   As output, your routines should produce the complete history of all the relative residual norms (2) produced during each run, the final approximate solution x_k of (1), and the total number of matrix-vector products $q = Av$ or $q' = A'v'$ computed during each run.

(b) Use your Matlab programs to solve the two linear systems (1) provided in the Matlab files "`HW4_Problem5b_1.mat`" and "`HW4_Problem5b_2.mat`". The first system is of size $n = 50653$, and the second system is of size $n = 389017$. For both systems, choose the vector $x_0 = e \in \mathbb{R}^n$ of all 1's as initial guess and try to solve (1) to relative residual norm of

```
tol = 1e-8.
```

For both systems, cases, run all 6 algorithms (i)–(vi). Run algorithms (v) and (vi) with these two choices of D: $D = D_0$ and $D = 10\,I$. Use the restart parameters $k_0 = 5$, $k_0 = 10$, and $k_0 = 20$ for algorithms (ii), (iv), and (vi).

For each of your runs, submit a graph that shows

$$\log \varrho_k, \quad k = 0, 1, \dots .$$

In addition, produce a single table that lists the total number of GMRES steps and the total number of matrix-vector products $q = Av$ or $q' = A'v'$ for each of your runs.