MAT 226B, Winter 2020
Homework 3

(due by Thursday, February 27, 11:59 pm)

General Instructions

- You are required to submit each of your assignments via file upload 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. For each assignment, upload two files: a single pdf file of your assignment and a single zip file that contains all your Matlab files. To prepare the zip file, create a directory that contains all your Matlab files and then zip that directory. For each of the subproblems that require Matlab computations include a single driver file so that I can run and check your program.
- If at all possible, use a text processing tool (such as LaTeX) for the preparation of your assignments. 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 assignment, I will not be able to grade it!
- You are required to solve the problems on the homework sets and on the final project yourself! If there are students with solutions or program codes that were obviously copied or are trivial modifications of each other, 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.

---

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

$$A = \begin{bmatrix} v & D \\ 1 & 0 \end{bmatrix}, \tag{1}$$

where $v \in \mathbb{R}^{n-1}$ and $D \in \mathbb{R}^{(n-1)\times(n-1)}$ is a nonsingular diagonal matrix, and let $r_0 = e_n$ be the n-th unit vector of length n.

(a) Determine the Krylov subspaces $K_k\big(A, r_0\big)$ for all $k = 1, 2, \ldots, d\big(A, r_0\big)$ and show that $d\big(A, r_0\big) = n$.

(b) How many iterations does the MR method with starting residual vector r_0 need to find the solution of $Ax = b$?

In the following, we assume that in (1), $D = I$ is the $(n-1) \times (n-1)$ identity matrix.

(c) Show that the matrix $A^T A$ has at most three 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) For both the CGNE method and Craig's method, give sharp upper bounds on the number of iterations that are needed to find the solution of $Ax = b$ for any right-hand side vector $b \in \mathbb{R}^n$.

2. Let $A \in \mathbb{R}^{n \times n}$ be a skew-symmetric matrix, *i.e.*, $A^T = -A$.

(a) Show that

$$x^T A^{2j+1} x = 0 \quad \text{for all} \quad j = 0, 1, 2, \ldots \quad \text{and} \quad x \in \mathbb{R}^n.$$

In the following, we assume that A is a nonsingular skew-symmetric matrix. Note that for A to be nonsingular, the matrix size n has to be even.

We employ the MR method and the CGNE algorithm with initial guess $x_0 \in \mathbb{R}^n$ to solve the linear system $Ax = b$, where $b \in \mathbb{R}^n$. We denote by x_k^{MR} and x_k^{CGNE} the k-th iterates generated by the MR method and the CGNE algorithm, respectively.

(b) Show that all eigenvalues of A are purely imaginary. Use this result to deduce that $d := d(A, r_0)$ is even.

(c) Show that

$$x_{2k}^{\mathrm{MR}} \in x_0 + \mathrm{span}\big\{\, Ar_0, A^3 r_0, A^5 r_0, \ldots, A^{2k-1} r_0 \,\big\} \quad \text{for all} \quad k = 0, 1, 2, \ldots, \frac{d}{2}.$$

and

$$x_{2k+1}^{\mathrm{MR}} = x_{2k}^{\mathrm{MR}} \quad \text{for all} \quad k = 0, 1, 2, \ldots, \frac{d}{2} - 1.$$

Hint: Write x_{2k}^{MR} in the form

$$x_{2k}^{\mathrm{MR}} = x_0 + \beta_1 r_0 + \beta_2 A r_0 + \beta_3 A^2 r_0 + \cdots + \beta_{2k} A^{2k-1} r_0, \qquad \beta_1, \beta_2, \ldots, \beta_{2k} \in \mathbb{R},$$

and use the system of linear equations derived in Problem 4(a) of Homework 2 to deduce that $\beta_{2j-1} = 0$ for $j = 1, 2, \ldots, k$.

(d) Show that
$$x_{2k}^{\mathrm{MR}} = x_k^{\mathrm{CGNE}} \quad \text{for all} \quad k = 0, 1, 2, \ldots, \frac{d}{2}.$$

3. Let $A \in \mathbb{R}^{n\times n}$ be a general nonsingular square matrix, which we write as
$$A = D_0 - F - G, \tag{2}$$
where D_0, $-F$, and $-G$ is the diagonal part, strictly lower-triangular part, and strictly upper-triangular part of A, respectively. Moreover let $D \in \mathbb{R}^{n\times n}$ be a given arbitrary nonsingular diagonal matrix, and consider the preconditioner
$$M := (D - F)D^{-1}(D - G) = M_1 M_2, \quad M_1 := D - F, \quad M_2 := D^{-1}(D - G),$$
for the matrix A. We denote by
$$A' := M_1^{-1} A M_2^{-1}$$
the corresponding preconditioned matrix.

(a) Show that
$$A' = \Big(\big(D - G\big)^{-1} + \big(D - F\big)^{-1} \Big(I + D_1 \big(D - G\big)^{-1} \Big) \Big) D, \tag{3}$$
where $D_1 := D_0 - 2D$.

(b) Employ formula (3) 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).

4. Matlab provides the function "`gmres`" for the iterative solution of linear equations
$$Ax = b, \tag{4}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsingular and $b \in \mathbb{R}^n$. The function "`gmres`" 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 $\|r_0\|_2$.

(a) Employ Matlab's "`gmres`" to write Matlab functions for each of the following algorithms:

 (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 (2);

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

 (v) GMRES with the preconditioner derived in Problem 3;

 (vi) Restarted GMRES with the preconditioner derived in Problem 3.

As output, your routines should produce the complete history of all the relative residual norms

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

produced during each run, the final approximate solution x_k of (4), 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 (4) provided in the binary Matlab files "`HW3_P4_1.mat`" and "`HW3_P4_2.mat`". The first system is of size $n = 32768$, and the second system is of size $n = 262144$. For both systems, choose the vector $x_0 = e \in \mathbb{R}^n$ of all 1's as initial guess and try to solve (4) 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, print out the total number of GMRES steps and the total number of matrix-vector products $q = Av$ or $q' = A'v'$. In addition, write your driver file such that it produces an on-screen graph that shows

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

for each of your runs.