MATH 8610 (SPRING 2023) HOMEWORK 9

Assigned 04/12/2023, due 04/22/2023 (Saturday) by 11:59pm.

Instructor: Dr. Fei Xue, Martin O-203, fxue@clemson.edu.

1. **[Q1]** (10 pts) (a) For a generic Krylov subspace method that takes the initial approximation x_0, gets the initial residual $r_0 = b - Ax_0$, develops the sequence of Krylov subspaces $\mathcal{K}_k(A, r_0)$ and constructs the approximate solution $x_k = x_0 + z_k$ where $z_k \in \mathcal{K}_k(A, r_0)$, the residual $r_k = b - Ax_k$ can be written as $r_k = p_{k+1}(A)r_0$, where p_{k+1} is a polynomial of degree no greater than $k+1$ with $p_{k+1}(0) = 1$.

   (b) Let A be SPD, and x_0 and $r_0 = b - Ax_0$ be the initial approximation and residual, respectively. Consider the Lanczos relation $AU_k = U_kT_k + \beta_k u_{k+1}e_k^T$ (Arnoldi's method applied to a symmetric A), where $u_1 = \frac{r_0}{\|r_0\|_2}$. Show that the k-th iterate of CG can be written as $x_k = x_0 + U_ky_k$, where y_k satisfies $T_ky_k = \|r_0\|_2 e_1$.
   (Hint: use the fact that $r_k = b - Ax_k = r_0 - AU_ky_k \perp \mathcal{K}_k(A, r_0) = \mathrm{col}(U_k)$)

   (c) Show that the k-th residual of GMRES $r_k = b - Ax_k$ satisfies $r_k \in \mathcal{K}_{k+1}(A, r_0)$, $r_k \perp A\mathcal{K}_k(A, r_0)$, $(r_k, r_k) = (r_j, r_k)$ for all $0 \le j \le k-1$, and therefore $\|r_k\|_2 \le \|r_j\|_2$.

2. **[Q2]** (10 pts for (a); 5 pts for (b); 5 pts for (c)) (a) Trefethen's book, Prob. 35.2.

   (b) Let $A \in \mathbb{R}^{n\times n}$ be nonsymmetric and diagonalizable. Assume that all eigenvalues of A lie in the disk centered at $c \in \mathbb{C} \setminus \{0\}$ with radius $r < |c|$. Consider using GMRES to solve the linear system $Ax = b$ iteratively. Show that the k-th relative residual satisfies $\frac{\|r_k\|_2}{\|r_0\|_2} \le C\left(\frac{r}{|c|}\right)^k$ for some constant C independent of k. What if A has a small number, say, $m \ll n$ eigenvalues outside such a disk?

   (c) If A is an SPD matrix with the smallest eigenvalue λ_1 and the largest eigenvalue λ_n, what is the convergence factor obtained in part (b)? Compare this factor with that of CG we learned in class. Which one is better?

3. **[Q3]** (10 pts) Let x^* be the true solution of $Ax = b$ with SPD A, x_k be the k-th iterate of CG, and $\varphi(x) = \frac{1}{2}x^TAx - b^Tx$ for CG minimization.

   (a) Note that $r_k \perp r_j$ for $0 \le j \le k-1$, and hence $r_k \perp U_k = \mathrm{span}\{p_0, p_1, \ldots, p_{k-1}\}$. Also note that $r_k = -\nabla\varphi(x_k)$, and any vector $x \in W_k = x_0 + U_k$. Explain from the optimization point of view, why $x_k = \mathrm{argmin}_{x\in W_k}\varphi(x)$.

   Hint: one possible (and easier) solution is to show that W_k is a convex set, and $\varphi(x)$ is a convex function defined on W_k; then local minimizer of $\varphi(x)$ is necessarily a global minimizer. Please do a little search on convex set/functions yourselves. The condition $r_k \perp U_k$ is crucial to show the optimality here.

   (b) Show directly that $x_k = \mathrm{argmin}_{x\in W_k}\|x - x^*\|_A$, without referring to the connection between $\varphi(x)$ and $\|e_k\|_A$. (Hint: consider a different $\tilde{x}_k \in W_k$, with $d_k = \tilde{x}_k - x_k \ne 0$. Show that $\|\tilde{x}_k - x^*\|_A = \|d_k + x_k - x^*\|_A \ge \|x_k - x^*\|_A$)

4. **[Q4]** (10 pts for (a)+(b); 10 **extra pts** for (c)) (a) A common misconception is that Krylov subspace methods solving $Ax = b$ converge rapidly if the condition number, say, $\kappa_2(A)$ is small. This is largely true if A is SPD, but in general not true otherwise.

   To explore this point, construct three matrices as follows

```
rng(‘default’);  n = 1024;  A = randn(n,n);  [A,R] = qr(A);
Ahat = A+1.2*eye(n);  E = randn(n,n);  E = E+E’;
B = (A+A’)/2;  B = B+1e-4*E;  Bhat = B+1.01*eye(n);
```

Check that A and $\widehat{A}$ are unsymmetric, B is symmetric and indefinite, and $\widehat{B}$ is SPD, and find $\kappa_2(A)$, $\kappa_2(\widehat{A})$, $\kappa_2(B)$ and $\kappa_2(\widehat{B})$. Are these condition numbers really large at all? Use `eig` to compute all eigenvalues of A, $\widehat{A}$, B and $\widehat{B}$, and plot them on the complex plane. How are these eigenvalues distributed around the origin?

Let us try GMRES, MINRES and CG, unpreconditioned, to solve $Ax = f$, $\widehat{A}x = f$, $Bx = f$ and $\widehat{B}x = f$, respectively, where $f = [1, 1, \ldots, 1]^T$, as follows.

```
f = ones(n,1);  m = n-1;  restart = 1;  tol = 1e-12;
[x1,flag1,relres1,iter1,resvec1] = gmres(A,f,m,tol,1);
semilogy(resvec1/norm(f),‘ro’); hold on;
[x2,flag2,relres2,iter2,resvec2] = gmres(Ahat,f,m,tol,1);
semilogy(resvec2/norm(f),‘go’); hold on;
[x3,flag3,relres3,iter3,resvec3] = minres(B,f,tol,m);
semilogy(resvec3/norm(f),‘bo’); hold on;
[x4,flag4,relres4,iter4,resvec4] = pcg(Bhat,f,tol,m);
semilogy(resvec4/norm(f),‘ko’); hold on;
legend(‘A by gmres’,‘Ahat by gmres’,‘B by minres’,‘Bhat by cg’);
```

Do you see any obvious relation between $\frac{\|r_k\|_2}{\|r_0\|_2}$ (convergence rate) and the condition number of the coefficient matrix? What about the eigenvalue distribution?

(b) Run the code `HW10_linsolvecomp.m` (need a machine with 32GB memory for the LU factorization of the nonsymmetric matrix B), read the output and make comments on the performance of iterative solvers using incomplete factorization preconditioners, compared to direct solvers based on sparse exact factorizations, for this problem.

(c*) Choose two methods from preconditioned CG, MINRES and GMRES(m) and implement them using the pseudocode below. Replace MATLAB's `pcg`, `minres`, and `gmres` in `HW10_linsolvecomp.m` with your codes. Check if they work.

Algorithm 1 Preconditioned conjugate gradient (PCG) for SPD linear system $Ax = b$

Input: Symmetric and positive definite $A \in \mathbb{R}^{n\times n}$, right-hand side $b \in \mathbb{R}^n$, initial approximation x_0 (typically zero), a tolerance $\delta > 0$, and a SPD preconditioner M
Output: An approximate solution x_k to $Ax = b$

Compute $r_0 = b - Ax_0$; solve $Mz_0 = r_0$ for z_0 (action of preconditioning); set $p_0 = z_0$;
for $k = 0, 1, 2, \ldots$ **do**
  $\alpha_k = \frac{(z_k, r_k)}{(Ap_k, p_k)}$;
  $x_{k+1} = x_k + \alpha_k p_k$;
  $r_{k+1} = r_k - \alpha_k A p_k$;
  if $\frac{\|r_{k+1}\|_2}{\|b\|_2} \le \delta$ **then**
    exit;
  end if
  Solve $Mz_{k+1} = r_{k+1}$ for z_{k+1}; (action of preconditioning)
  $\beta_k = \frac{(z_{k+1}, r_{k+1})}{(z_k, r_k)}$;
  $p_{k+1} = z_{k+1} + \beta_k p_k$;
end for

Algorithm 2 Preconditioned MINRES for symmetric linear system $Ax = b$

Input: Symmetric and possibly indefinite $A \in \mathbb{R}^{n\times n}$, right-hand side $b \in \mathbb{R}^n$, initial approximation x_0 (typically zero), a tolerance $\delta > 0$, and an SPD preconditioner M
Output: An approximate solution x_k to $Ax = b$

$v_0 = \mathbf{0}$, $w_0 = \mathbf{0}$, $w_1 = \mathbf{0}$;
Compute $v_1 = r_0 = b - Ax_0$; solve $Mz_1 = v_1$ for z_1 (action of preconditioning);
Set $\eta_0 = \sqrt{b^T M^{-1} b}$, $\gamma_0 = 1$, $\gamma_1 = \sqrt{(z_1, v_1)}$, $\eta = \gamma_1$, $s_0 = s_1 = 0$, $c_0 = c_1 = 1$;
for $j = 1, 2, \ldots$ **do**
  $z_j = z_j/\gamma_j$;
  $\delta_j = (Az_j, z_j)$;
  $v_{j+1} = Az_j - \frac{\delta_j}{\gamma_j} v_j - \frac{\gamma_j}{\gamma_{j-1}} v_{j-1}$;
  Solve $Mz_{j+1} = v_{j+1}$ for z_{j+1} (action of preconditioning)
  $\gamma_{j+1} = \sqrt{(z_{j+1}, v_{j+1})}$;
  $\alpha_0 = c_j\delta_j - c_{j-1}s_j\gamma_j$, $\alpha_1 = \sqrt{\alpha_0^2 + \gamma_{j+1}^2}$; $\alpha_2 = s_j\delta_j + c_{j-1}c_j\gamma_j$; $\alpha_3 = s_{j-1}\gamma_j$;
  $c_{j+1} = \frac{\alpha_0}{\alpha_1}$; $s_{j+1} = \frac{\gamma_{j+1}}{\alpha_1}$;
  $w_{j+1} = (z_j - \alpha_3 w_{j-1} - \alpha_2 w_j)/\alpha_1$;
  $x_j = x_{j-1} + \eta c_{j+1} w_{j+1}$; $\eta = -s_{j+1}\eta$;
  if $\frac{|\eta|}{\eta_0} \le \delta$ **then**
    exit;
  end if
end for

Algorithm 3 Right-preconditioned GMRES(m) for nonsymmetric linear system $Ax = b$

Input: Nonsymmetric $A \in \mathbb{R}^{n\times n}$, right-hand side $b \in \mathbb{R}^n$, maximum dimension m, initial approximation x_0 (typically zero), a tolerance $\delta > 0$, and a preconditioner M
Output: An approximate solution x_k to $Ax = b$

Compute $r_0 = b - Ax_0$, $\beta_0 = \|r_0\|_2$, $u_1 = r_0/\beta_0$;
for $\ell = 1, 2, \ldots$ **do**
  for $k = 1, 2, \ldots, m$ **do**
    Solve $Mz_k = u_k$ for z_k (action of preconditioning); update $z_k = Az_k$;
    for $j = 1, \ldots, k$ **do**
      $h_{jk} = u_j^T z_k$;
      $z_k = z_k - h_{jk}u_j$;
    end for
    for $j = 1, \ldots, k$ **do**
      $\Delta h = u_j^T z_k$;
      $z_k = z_k - \Delta h u_j$;
      $h_{jk} = h_{jk} + \Delta h$;
    end for
    $h_{k+1,k} = \|z_k\|_2$; $u_{k+1} = z_k/h_{k+1,k}$;
    Compute y_k s.t. $\beta_k = \|\beta_0 e_1 - \underline{H}_k y_k\|_2$ is minimized, where $\underline{H}_k = [h_{ij}]_{1\le i\le k+1, 1\le j\le k}$
    if $\beta_k/\|b\|_2 \le \delta$ **then**
      Solve $Mz_k = U_k y_k$ for z_k (action of preconditioning); compute $x_k = x_0 + z_k$; exit;
    end if
  end for
  Solve $Mz_k = U_k y_k$ for z_k (action of preconditioning); compute $x_k = x_0 + z_k$;
  $x_0 = x_k$; $r_0 = b - Ax_0$; $\beta_0 = \|r_0\|_2$; $u_1 = r_0/\beta_0$;
end for

Matrix A	Solver	Preconditioner	Comment
SPD	CG	SPD	symmetric split preconditioning
Symmetric indefinite	MINRES	SPD	symmetric split preconditioning
	SQMR	Symmetric	more flexible preconditioning
Unsymmetric	GMRES(m)	Any	restart needed
	BICGSTAB(ℓ)	Any	no restart needed
	IDR(s)	Any	no restart needed

TABLE 0.1
Structure of A, solvers and preconditioners (right preconditioning recommended for unsymmetric A)