Numerical Linear Algebra QS 7 *(MT 2019)*

Optional questions: Qns 5, 8

1. Derive and write down the MINRES algorithm and show that the work per iteration is $O(n)$ for a sparse real symmetric matrix with $O(1)$ entries per row.

2. Consider the recurrence

$$\gamma_{j+1}\mathbf{v}_{j+1} = A\mathbf{v}_j - \delta_j\mathbf{v}_j - \gamma_j\mathbf{v}_{j-1}, \quad 1 \le j \le k-1,$$

where $\mathbf{v}_1$ is an arbitrary vector with $\|\mathbf{v}_1\|_2 = 1$, $\mathbf{v}_0 = 0$, $\delta_j = \mathbf{v}_j^T A\mathbf{v}_j$, and γ_j is chosen so that $\|\mathbf{v}_j\|_2 = 1$. Prove that for a real symmetric matrix A this procedure generates an orthonormal basis for the Krylov subspace $\mathcal{K}_k(A, \mathbf{v}_1)$. (Hint: use induction and note that for a symmetric matrix A

$$\langle A\mathbf{v}_j, \mathbf{v}_{j-1}\rangle = \langle \mathbf{v}_j, A\mathbf{v}_{j-1}\rangle = \mathbf{v}_j^T A\mathbf{v}_{j-1},$$

and also that $A\mathbf{v}_{j-1} = \gamma_j\mathbf{v}_j + \mathbf{w}$, with $\mathbf{w} \in \mathtt{span}\{\mathbf{v}_1, \dots, \mathbf{v}_{j-1}\}$.)

3. For a chosen $\mathbf{x}_0$, if $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and $\mathbf{p}_0 = \mathbf{r}_0$ and for $k = 0, 1, \dots$

$$\begin{aligned} & \alpha_k = \langle \mathbf{p}_k, \mathbf{r}_k\rangle / \langle \mathbf{p}_k, A\mathbf{p}_k\rangle \\ (1) \quad & \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k \\ (2) \quad & \mathbf{r}_{k+1} = \mathbf{b} - A\mathbf{x}_{k+1} \\ & \beta_k = -\langle \mathbf{p}_k, A\mathbf{r}_{k+1}\rangle / \langle \mathbf{p}_k, A\mathbf{p}_k\rangle \\ (3) \quad & \mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k \end{aligned}$$

show that (2) and (1) imply

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k.$$

Prove that the definition of α_k implies $\langle \mathbf{r}_{k+1}, \mathbf{p}_j\rangle = 0$ for $j = k$ and that the definition of β_k implies $\langle \mathbf{p}_{k+1}, A\mathbf{p}_j\rangle = 0$ for $j = k$. Prove also that $\langle \mathbf{r}_{k+1}, \mathbf{r}_j\rangle = 0$ for $j = k$. Now by employing induction in k for $k = 1, 2, \dots$, prove these three assertions for $j = 1, 2, \dots, k-1$. (The inductive assumption will be that

$$\langle \mathbf{r}_k, \mathbf{p}_j\rangle = 0, \quad \langle \mathbf{r}_k, \mathbf{r}_j\rangle = 0, \quad \langle \mathbf{p}_k, A\mathbf{p}_j\rangle = 0, \quad j = 0, 1, \dots, k-1$$

and you may wish to tackle the assertions in this order.)

4. By expanding $\|\mathbf{x} - (\mathbf{x}_k + \alpha \mathbf{p}_k)\|_A^2$ and using simple calculus, show that the value $\alpha = \mathbf{p}_k^T \mathbf{r}_k / \mathbf{p}_k^T A \mathbf{p}_k$ is minimising. Use the result of the question above to further show that $\mathbf{p}_k^T \mathbf{r}_k = \mathbf{r}_k^T \mathbf{r}_k$ and that and alternative formula for β_k is

$$\beta_k = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k}.$$

These equivalent formulae give the form of the Conjugate Gradient Algorithm usually used for computation:

For a chosen $\mathbf{x}_0$, if $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and $\mathbf{p}_0 = \mathbf{r}_0$ and for $k = 0, 1, \ldots$

$$\begin{aligned}
& \alpha_k = \langle \mathbf{r}_k, \mathbf{r}_k \rangle / \langle \mathbf{p}_k, A\mathbf{p}_k \rangle \\
(1) \quad & \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \\
(2) \quad & \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A \mathbf{p}_k \\
& \beta_k = \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle / \langle \mathbf{r}_k, \mathbf{r}_k \rangle \\
(3) \quad & \mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k
\end{aligned}$$

5. Based on the form of the Conjugate Gradient algorithm given in the question above, write an efficient implementation (in pseudocode or matlab notation) which requires only one matrix$\times$vector product at each iteration and minimises the number of vector operations.

6. Use matlab (`[x,flag,relres,iter,resvec]=pcg(A,b,1.e-6,size(A,1))`) with suitably chosen matrices A and b as below to investigate the behaviour of Conjugate Gradients.

Note in the form above matlab will use unpreconditioned Conjugate Gradients, `flag=0` will indicate successful convergence (the relative residual norm - `relres` - less than 10^{-6} in less than dimension(A) `=size(A,1)` iterations), `iter` is the number of iterations taken and `resvec` is the vector of residual norms at each iteration (hence `semilogy(resvec)` will plot the convergence curve). See `help pcg` if you want to read more or change any of the defaults.

(i) `A=randn(n); A=A*A'; b=ones(n,1);` for `n=7,47,...` as you choose (and have patience for! (note `ctrl C` will interrupt a computation). These are dense matrices!

(ii) `A=sprandsym(100,0.1,invkappa,1); b=ones(100,1);`. This is a sparse 100×100 symmetric and positive definite matrix with approximately 10 non-zero entries per row (ie. `0.1` of the 10,000 entries non-zero) and with $\|\cdot\|_2$-norm condition number `1/invkappa`. Try `pcg` with well-conditioned matrices (small κ or `invkappa` just less than 1) and badly conditioned matrices (large κ or `invkappa` nearly zero).

(iii) a symmetric and positive definite matrix that has few distinct eigenvalues
eg. `X=randn(9,9); X=orth(X); A=X*diag([1,1,4,3,3,4,4,4,3])*X'`
(note that it is possible that an X generated with random entries is singular, but is rarely so!)

(iv) any of the above with a preconditioner of your choice.

The remaining questions are really on the work of week 8: please leave them to attempt after the last lectures on this course,

7. Derive the preconditioned Congugate Gradient Algorithm with preconditioner P:

For a chosen $\mathbf{x}_0$, if $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ and $\mathbf{z}_0$ solves $P\mathbf{z}_0 = \mathbf{r}_0$ with $\mathbf{p}_0 = \mathbf{z}_0$ and for $k = 0, 1, \ldots$

$$\begin{aligned}
& \alpha_k = \langle \mathbf{z}_k, \mathbf{r}_k \rangle / \langle \mathbf{p}_k, A\mathbf{p}_k \rangle \\
(1) \quad & \mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \\
(2) \quad & \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{p}_k \\
(3) \quad & \text{Solve } P\mathbf{z}_{k+1} = \mathbf{r}_{k+1} \\
& \beta_k = \langle \mathbf{z}_{k+1}, \mathbf{r}_{k+1} \rangle / \langle \mathbf{z}_k, \mathbf{r}_k \rangle \\
(4) \quad & \mathbf{p}_{k+1} = \mathbf{z}_{k+1} + \beta_k \mathbf{p}_k
\end{aligned}$$

by considering the unpreconditioned Conjugate Gradient Algorithm as in Question 4 above applied to

$$H^{-1}AH^{-T}\mathbf{v} = H^{-1}\mathbf{b}, \quad \mathbf{v} = H^T\mathbf{x}$$

where $P = HH^T$.

(Hint you may wish to write $\widehat{A} = H^{-1}AH^{-T}$, $\widehat{\mathbf{x}} = H^T\mathbf{x}$, $\widehat{\mathbf{b}} = H^{-1}\mathbf{b}$ and write down the Conjugate Gradient algorithm for $\widehat{A}\widehat{\mathbf{x}} = \widehat{\mathbf{b}}$ to generate $\{\widehat{\mathbf{x}}_k\}$, $\{\widehat{\mathbf{p}}_k = H^T\mathbf{p}_k\}$ etc.)

8. Consider a symmetric coefficient matrix A, show that if the splitting matrix M is also symmetric, then the iteration matrix $S = I - M^{-1}A$ is symmetric with respect to the A inner product; that is

$$\langle S\mathbf{x}, \mathbf{y} \rangle_A = \langle \mathbf{x}, S\mathbf{y} \rangle_A.$$

This means that S (and indeed S^k, where k is the number of iteration steps) may be used as a preconditioner with Conjugate Gradients.