AMSC 660 Homework 7
Due Oct 18
By Marvyn Bailly

Problem 1

Prove that for a sequence of iterates of the conjugate gradient algorithm, the preliminary version, the residuals are orthogonal, i.e.,

$$r_k^\top r_i = 0, \quad i = 0, 1, \ldots, k-1.$$

You can use the facts proven in class:

$$\text{span}\{p_0, \ldots, p_k\} = \text{span}\{r_0, \ldots, r_k\} = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\} = \mathcal{K}(r_0, k),$$

and

$$p_k^\top A p_i = 0, \quad i = 0, 1, \ldots, k-1,$$

and Theorem 5.2 from NW.

Solution

Proof. We wish to prove that for a sequence of iterates of the conjugate gradient algorithm, the preliminary version, the residuals are orthogonal. From Theorem 5.2 we have that,

$$r_k^\top p_i = 0, \quad i = 0, 1, \ldots, k-1. \tag{1}$$

From Algorithm 5.1, notice that

$$p_i = -r_i + \beta_i p_{i-1} \implies r_i = \beta_i p_{i-1} - p_i. \tag{2}$$

Plugging this into $r_k^\top r_i$ yields

$$r_k^\top r_i = r_k^\top(\beta_i p_{i-1} - p_i) = \beta_i r_k^\top p_{i-1} - r_k^\top p_i = 0,$$

as $r_k^\top p_{i-1} = r_k^\top p_i = 0$ by Eq. (1). Note that $r_k^\top r_0 = -r_k^\top p_0 = 0$ by definition of p_0. Therefore $r_k^\top r_i = 0$ for $i = 0, 1, \ldots, k-1$.

$\square$

Problem 2

Prove that the conjugate gradient algorithm, the preliminary version (Algorithm 5.1, page 108 in [NW]), is equivalent to Algorithm 5.2 (CG) (page 112 in [NW]), i.e., that

$$\alpha_k = \frac{r_k^\top r_k}{p_k^\top A p_k},$$

and

$$\beta_{k+1} = \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k}.$$

Solution

Proof. We wish to show that α_k and β_{k+1} from Algorithm 5.1 and Algorithm 5.2 are equivalent. In Algorithm 5.1, we define α_k as

$$\alpha_k = -\frac{r_k^\top p_k}{p_k^\top A p_k}.$$

Notice that we can rewrite the numerator using Eq. (2) as

$$r_k^\top p_k = r_k^\top(-r_k + \beta_k p_{k-1}) = -r_k^\top r_k + \beta_k r_k^\top p_{k-1},$$

and from Eq. (1), we can reduce the expression to

$$r_k^\top p_k = -r_k^\top r_k + 0 = -r_k^\top r_k,$$

for all $i = 0, 1, \ldots, k-1$. Plugging this result into α_k gives

$$\alpha_k = \frac{r_k^\top r_k}{p_k^\top A p_k},$$

as defined in Algorithm 2. In Algorithm 5.1, we define β_{k+1} as

$$\beta_{k+1} = \frac{r_{k+1}^\top A p_k}{p_k^\top A p_k}.$$

Recall that we have

$$r_{k+1} = r_k + \alpha_k A p_k \implies \alpha_k A p_k = r_{k+1} - r_k,$$

and

$$p_k^\top A p_k = \frac{r_k^\top r_k}{\alpha_k}.$$

Plugging this into β_{k+1} yields

$$\beta_{k+1} = \frac{r_{k+1}^\top A p_k}{p_k^\top A p_k}$$

$$
\begin{aligned}
&= \frac{r_{k+1}^\top \alpha_k A p_k}{r_k^\top r_k} \\
&= \frac{r_{k+1}^\top (r_{k+1} - r_k)}{r_k^\top r_k} \\
&= \frac{r_{k+1}^\top r_{k+1} - r_{k+1}^\top r_k}{r_k^\top r_k} \\
&= \frac{r_{k+1}^\top r_{k+1}}{r_k^\top r_k},
\end{aligned}
$$

where $r_{k+1}^\top r_k = 0$ from question 1. Thus gives β_{k+1} from Algorithm 2.

$\square$

Problem 3

Let A be an $n \times n$ matrix. A subspace spanned by the columns of an $n \times k$ matrix B is an invariant subspace of A if A maps it into itself, i.e., if $AB \subset \text{span}(B)$. This means that there is a $k \times k$ matrix C such that $AB = BC$. Prove that if a vector $r \in \mathbb{R}^n$ lies in the k-dimensional subspace spanned by the columns of B, i.e., if $r = By$ for some $y \in \mathbb{R}^k$ (r is a linear combination of columns of B with coefficients $y_1, \dots, y_k$) then the Krylov subspaces generated by r spot expanding at degree $k-1$, i.e,

$$\text{span}\{r, Ar, \dots, A^p r\} = \text{span}\big\{r, Ar, \dots, A^{k-1}r\big\}, \quad \forall p \geq k.$$

Solution

Proof. Let A be an $n \times n$ matrix. A subspace spanned by the columns of an $n \times k$ matrix B is an invariant subspace of A if A maps it into itself, i.e., if $AB \subset \text{span}(B)$. This means that there is a $k \times k$ matrix C such that $AB = BC$. We wish to show that if a vector $r \in \mathbb{R}^n$ lies in the k-dimensional subspace spanned by the columns of B, i.e., if $r = By$ for some $y \in \mathbb{R}^k$ (r is a linear combination of columns of B with coefficients $y_1, \dots, y_k$) then the Krylov subspaces generated by r spot expanding at degree $k-1$, i.e,

$$\text{span}\{r, Ar, \dots, A^p r\} = \text{span}\big\{r, Ar, \dots, A^{k-1}r\big\}, \quad \forall p \geq k.$$

Observe that for $p \leq k$, we have that

$$\begin{aligned}
\text{span}\{r, Ar, \dots, A^p r\} &= \text{span}(By, ABy, \dots, A^p By) \\
&= \text{span}\{By, BCy, \dots, BC^p y\} \\
&= B\text{span}\{y, Cy, \dots, C^{k-1}y, \dots, C^p y\} \\
&= B\text{span}\{y, Cy, \dots, y, C^{k-1}y\} \\
&= \text{span}\{By, BCy, \dots, BC^{k-1}y\} \\
&= \text{span}\{By, ABy, \dots, A^{k-1}By\} \\
&= \text{span}\{r, Ar, \dots, A^{k-1}r\}
\end{aligned}$$

where we use the fact, $A^n B = BC^n$ for $\forall n$ and that a vector $r = By \in \mathbb{R}^k$ lies in the k-dimensional subspace spanned by the columns of B so

$$B\text{span}\{y, Cy, \dots, C^{k-1}y, \dots, C^p y\} = B\text{span}\{y, Cy, \dots, y, C^{k-1}y\} \quad \forall p \geq k.$$

Therefore,

$$\text{span}\{r, Ar, \dots, A^p r\} = \text{span}\big\{r, Ar, \dots, A^{k-1}r\big\}, \quad \forall p \geq k.$$

$\square$

Problem 4

Prove Theorem 5.5 From [NW], page 115. Here are the steps that you need to work out.

(a) Construct a polynomial $Q(\lambda)$ of degree $k+1$ with roots $\lambda_n, \lambda_{n-1}, \ldots, \lambda_{n-k+1}$ and $\frac{1}{2}(\lambda_1 + \lambda_{n-k})$ such that $Q(0) = 1$.

(b) Argue that $P(\lambda)$ defined as

$$P(\lambda) = \frac{Q(\lambda) - 1}{\lambda},$$

is a polynomial, not a rational function, by referring to the theorem about factoring polynomials. Cite that theorem.

(c) Use the ansatz

$$\|x_{k+1} - x^*\|_A^2 \leq \min_{P \subset \mathcal{P}_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x^*\|_A^2.$$

Argue that

$$\|x_{k+1} - x^*\|_A^2 \leq \max_{1 \leq i \leq n} Q(\lambda_i)^2 \|x_0 - x^*\|_A^2.$$

(d) Show that

$$\max_{\lambda \in [\lambda_1, \lambda_{n-k}]} [Q(\lambda)]^2 \leq \max_{\lambda \in [\lambda_1, \lambda_{n-k}]} \left| \frac{\lambda - \frac{1}{2}(\lambda_1 + \lambda_{n-k})}{\frac{1}{2}(\lambda_1 + \lambda_{n-k})} \right|^2.$$

(e) Find the maximum of the function in the right-hand side of the last equation in the interval $[\lambda_1, \lambda_{n-k}]$.

(f) Finish the proof of the theorem.

Solution

Proof. We wish to prove that if A has eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, then

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1} \right)^2 \|x_0 - x^*\|_A^2.$$

Let's begin by constructing a polynomial $Q(\lambda)$ of the form

$$Q(\lambda) = a(\lambda - \lambda_n)(\lambda - \lambda_{n-1}) \cdots (\lambda - \lambda_{n-k+1}) \left(\lambda - \frac{\lambda_1 + \lambda_{n-k}}{2} \right),$$

then if we want to $Q(\lambda)$ to satisfy $Q(0) = 1$ we set

$$Q(0) = 1 = a(-\lambda_n)(-\lambda_{n-1}) \cdots (-\lambda_{n-k+1}) \left(-\frac{\lambda_1 + \lambda_{n-k}}{2} \right)$$

so

$$a = \frac{(-1)^{k+1}}{(\lambda_n)(\lambda_{n-1})\cdots(\lambda_{n-k+1})\left(\frac{\lambda_1+\lambda_{n-k}}{2}\right)}.$$

Thus

$$Q(\lambda) = \frac{(-1)^{k+1}(\lambda-\lambda_n)(\lambda-\lambda_{n-1})\cdots(\lambda-\lambda_{n-k+1})\left(\lambda-\frac{\lambda_1+\lambda_{n-k}}{2}\right)}{(\lambda_n)(\lambda_{n-1})\cdots(\lambda_{n-k+1})\left(\frac{\lambda_1+\lambda_{n-k}}{2}\right)}.$$

Then if we define

$$P(\lambda) = \frac{Q(\lambda)-1}{\lambda},$$

which is a rational function by the fundamental theorem of algebra (or the polynomial remainder theorem) as we have written $Q(\lambda)$ completely in terms of its factors, we can also completely factor

$$Q(\lambda) - 1 = \lambda P(\lambda),$$

and since $Q(\lambda) - 1$ is a rational function with a root at $\lambda = 0$, so $R(\lambda)$ must be a rational function of degree k by polynomial division. Now let's assume that

$$\begin{aligned}
\|x_{k+1} - x^*\|_A^2 &\le \min_{R\subset\mathcal{P}_k} \max_{1\le i\le n} [1+\lambda_i R_k(\lambda_i)]^2 \|x_0 - x^*\|_A^2 \\
&\le \max_{1\le i\le n} [1+\lambda_i P(\lambda_i)]^2 \|x_0 - x^*\|_A^2 \\
&= \max_{1\le i\le n} Q(\lambda_i)^2 \|x_0 - x^*\|_A^2. \qquad (3)
\end{aligned}$$

Now we have that by

$$\begin{aligned}
\max_{\lambda\in[\lambda_1,\lambda_{n-k}]} [Q(\lambda)]^2 &= \max_{\lambda\in[\lambda_1,\lambda_{n-k}]} \left(\frac{(-1)^{k+1}(\lambda-\lambda_n)(\lambda-\lambda_{n-1})\cdots(\lambda-\lambda_{n-k+1})\left(\lambda-\frac{\lambda_1+\lambda_{n-k}}{2}\right)}{(\lambda_n)(\lambda_{n-1})\cdots(\lambda_{n-k+1})\left(\frac{\lambda_1+\lambda_{n-k}}{2}\right)}\right)^2 \\
&\le \max_{\lambda\in[\lambda_1,\lambda_{n-k}]} \left|\frac{(-1)^{k+1}(\lambda-\lambda_n)(\lambda-\lambda_{n-1})\cdots(\lambda-\lambda_{n-k+1})\left(\lambda-\frac{\lambda_1+\lambda_{n-k}}{2}\right)}{(\lambda_n)(\lambda_{n-1})\cdots(\lambda_{n-k+1})\left(\frac{\lambda_1+\lambda_{n-k}}{2}\right)}\right|^2 \\
&= \max_{\lambda\in[\lambda_1,\lambda_{n-k}]} \left|\frac{(\lambda-\lambda_n)}{\lambda_n}\right|^2 \max_{\lambda\in[\lambda_1,\lambda_{n-k}]} \left|\frac{(\lambda-\lambda_{n-1})}{\lambda_{n-1}}\right|^2 \cdots \max_{\lambda\in[\lambda_1,\lambda_{n-k}]} \left|\frac{\lambda-\frac{1}{2}(\lambda_1+\lambda_{n-k})}{\frac{1}{2}(\lambda_1+\lambda_{n-k})}\right|^2,
\end{aligned}$$

and since by the ordering of λ_is, we have that the max of each term is achieved by λ_1. [1] Furthermore, notice that plugging λ_1 into all of the terms expect the last term, we will have that each term is less than or equal to one, i.e,

$$\max_{\lambda\in[\lambda_1,\lambda_{n-k}]} \left|\frac{(\lambda-\lambda_{n-i})}{\lambda_{n-i}}\right|^2 = \left|\frac{(\lambda_1-\lambda_{n-i})}{\lambda_{n-i}}\right|^2 \le 1, \quad \forall 0 \le i \le k+1.$$

[1] note that λ_{n-k} also achieves the max and gives the same results for the rest of this proof.

Thus we have that

$$\max_{\lambda\in[\lambda_1,\lambda_{n-k}]}[Q(\lambda)]^2 \leq \max_{\lambda\in[\lambda_1,\lambda_{n-k}]}\left|\frac{\lambda-\frac{1}{2}(\lambda_1+\lambda_{n-k})}{\frac{1}{2}(\lambda_1+\lambda_{n-k})}\right|^2.$$

Now, as the max of the right-hand term of the above equation is also achieved λ_1, we find that

$$\max_{\lambda\in[\lambda_1,\lambda_{n-k}]}\left|\frac{\lambda-\frac{1}{2}(\lambda_1+\lambda_{n-k})}{\frac{1}{2}(\lambda_1+\lambda_{n-k})}\right|^2 = \left|\frac{\lambda_1-\frac{1}{2}(\lambda_1+\lambda_{n-k})}{\frac{1}{2}(\lambda_1+\lambda_{n-k})}\right|^2 = \left|\frac{\lambda_1-\lambda_{n-k}}{\lambda_{n-k}+\lambda_1}\right|^2 = \left(\frac{\lambda_{n-k}-\lambda_1}{\lambda_{n-k}+\lambda_1}\right)^2.$$

Plugging this back into Eq. (3) yields

$$\|x_{k+1}-x^*\|_A^2 \leq \left(\frac{\lambda_{n-k}-\lambda_1}{\lambda_{n-k}+\lambda_1}\right)^2\|x_0-x^*\|_A^2.$$

□

Problem 5

Matlab Problem

Solution

Proof. We implemented the conjugate gradient method in MatLab with the script:

```
function [res,x] = CG(A,b,tol)
    % input: A,b and tol
    % output: norm of residual at each iteration

    %compute x_0
    x = zeros(length(A),1);
    %set up
    r = A*x - b;
    p = -r;
    res = norm(r);

    while norm(r) >= tol
        alpha = (r'*r) / (p'*A*p);
        x = x + alpha*p;
        rNew = r + alpha*A*p;
        beta = (rNew'*rNew)/(r'*r);
        p = -rNew + beta*p;

        r = rNew;
        res = [res,norm(r)];
    end
end
```

and the preconditioned conjugate gradient method as:

```
function [res,x] = PCG(A,b,tol)
    % input: A,b and tol
    % output: norm of residual at each iteration

    %compute x_0
    x = ones(length(A),1);

    %compute M
    ichol_fac = ichol(sparse(A));
    M = ichol_fac*ichol_fac';

    %set up
```

```
    r = A*x - b;
    y = M\r;
    p = -y;
    res = norm(r);

    while norm(r) >= tol
        alpha = (r'*y) / (p'*A*p);
        x = x + alpha*p;
        rNew = r + alpha*A*p;
        yNew = M\rNew;
        beta = (rNew'*yNew)/(r'*y);
        p = -yNew + beta*p;

        y = yNew;
        r = rNew;
        res = [res,norm(r)];
    end
end
```

Then using the two methods to solve

$$-L_{\text{symm}} y = -b_{\text{symm}},$$

and plotting the norm of the residual at each iteration, see Figure 1, we see that the preconditioned conjugate gradient method converges in significantly fewer iterations than the conjugate gradient. We can also visualize the solutions to see that both methods are able to correctly solve the system and find the exit, see figure 2. Code can be found at `https://github.com/MarvynBailly/AMSC660/blob/main/homework7`.

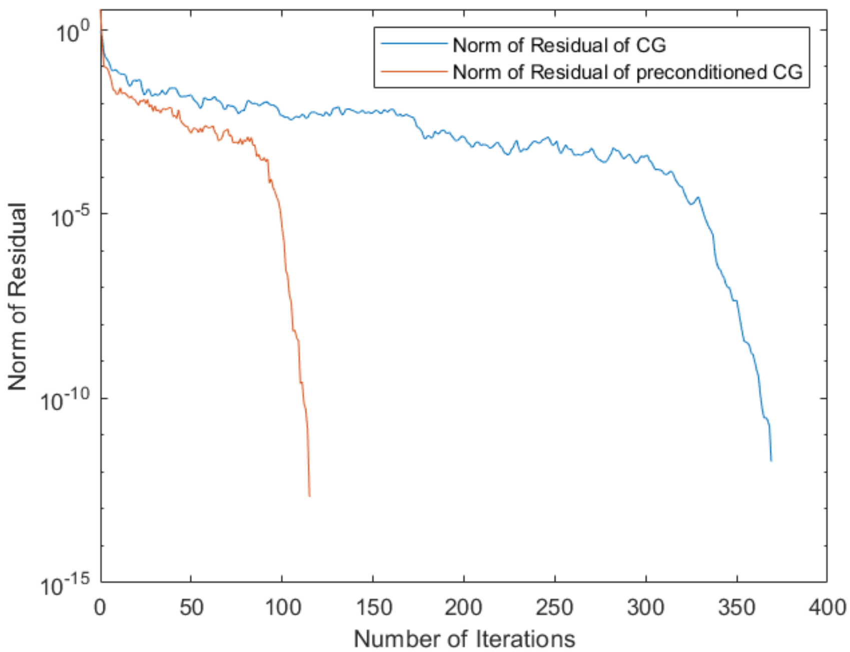
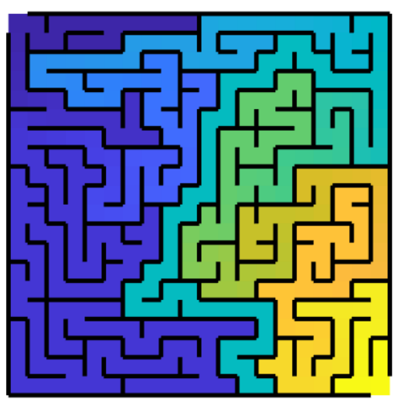
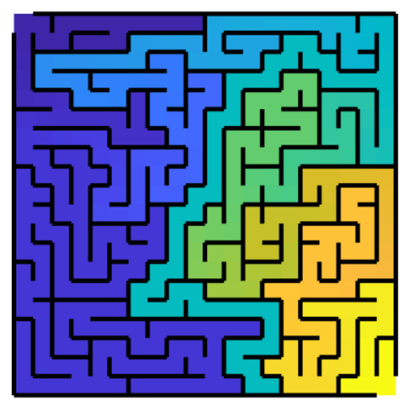


Figure 1: The 2 norm of the residuals of conjugate gradient method (seen in blue) compared to the preconditioned conjugate gradient method (seen in orange)
at each iteration.



(a)



(b)

Figure 2: Visualized solution of the conjugate gradient method (a) and preconditioned conjugate gradient method (b).

□