1 Iterative Methods for Solving $Ax = b$

1.1 The Arnoldi Iteration

Given some $m \times m$ matrix $A$, suppose we reduce $A$ to Hessenberg form $H$ by an orthogonal matrix $Q$. Then, $AQ = QH$. Let $Q_n$ be the matrix composed of the first $n$ columns of $Q$.

The Arnoldi iteration is based on the idea that we find $Q_n$’s for $n = 1, \cdots, m$ iteratively by projecting this $m$-dimensional problem into lower-dimensional Krylov subspaces, $K_n = \langle b, Ab, \cdots, A^{n-1}b \rangle$, whose basis vectors can be calculated by computers systematically and should be of no worry to algorithm designers. Let $\tilde{H}_n$ be the $(n + 1) \times n$ upper-left block of $H$ and $\hat{H}_n$ be the $m \times n$ leftmost block of $H$. Because $AQ_n$ contains the first $n$ columns of $AQ$ and the entries in $H$, as a Hessenberg matrix, right below the block $\hat{H}_n$ are all zero, we have

$$AQ_n = Q\hat{H}_n = Q_{n+1}\tilde{H}_n. \quad (1)$$

Suppose $q_1, q_2, \cdots, q_n$ are columns of $Q_n$ and we know they are orthogonal vectors. We can rewrite the $n$th column of equation 1 above as

$$Aq_n = \sum_{i=0}^{n+1} h_{in} q_i, \quad (2)$$

which is an iterative formula for $q_n$’s.

We then obtain the following algorithm, formally known as the Arnoldi iteration, for $q_n$’s. Let $b$ be some arbitrary $m$-dimensional vector.

---

**Algorithm 1: Arnoldi Iteration**

\[ q_1 = \frac{b}{\|b\|} \]

for $n = 1, 2, \cdots$ do

\[ v = Aq_n \]

for $i = 1$ to $n$ do

\[ h_{in} = q_i^* v \]

\[ v = v - h_{in} q_i \]

end

\[ h_{n+1,n} = \|v\| \]

\[ (h_{n+1,n} = \|h_{n+1,n}q_{n+1}\| = \|Aq_{n} - \sum_{i=0}^{n} h_{in} q_i\| = \|v\|) \]

\[ q_{n+1} = \frac{v}{h_{n+1,n}} \]

end

---
By the algorithm above, each \( q_n \) can be written as a linear combination of the vectors \( b, Ab, \cdots, A^{n-1}b \). So, we may conclude that
\[
\mathcal{K}_n = \langle b, Ab, \cdots, A^{n-1}b \rangle = \langle q_1, q_2, \cdots, q_n \rangle. \tag{3}
\]

### 1.2 GMRES

#### 1.2.1 The Least Squares Problem

Consider the system of linear equations \( Ax = b \), where \( A \) is nonsingular. GMRES, known as “generalized minimal residuals”, is an algorithm approximating solutions of such systems. As the name implies, its idea is to approximate the solution as “generalized minimal residuals”, which minimizes the norm \( \| A_r \|_2 \). GMRES converges monotonically, i.e.,

\[
\| r_n \| \leq \| r_{n+1} \|,
\]

where \( \mathcal{K}_n \subseteq \mathbb{R}^n \), once \( n = m \), \( r_n = 0 \). So, GMRES has to converge at step \( n \leq m \). Also, because \( \mathcal{K}_n \subseteq \mathcal{K}_{n+1} \), GMRES converges monotonically, i.e., \( \| r_{n+1} \| \leq \| r_n \| \).

To further explore the convergence of GMRES, we reduce GMRES to a polynomial approximation problem. Because \( x_n \in \mathcal{K}_n = \langle b, Ab, \cdots, A^{n-1}b \rangle \), \( x_n = \sum_{i=0}^{n-1} c_i A^i b = g_n(A)b \), where \( q_n \) is a polynomial of degree \( n - 1 \). Then,\n
\[
r_n = b - Ax_n = (I - Aq_n(A))b = p_n(A)b, \text{ where } p_n(z) = 1 - zq_n(z). \tag{4}
\]

Let \( P_n = \{ 1 + \sum_{i=1}^{n} c_i z^i \mid c_i \in \mathbb{C} \} \). The GMRES Approximation Problem can be phrased as

\[
\text{finding } p_n \in P_n \text{ which minimizes } \| p_n(A)b \|. \tag{5}
\]
With the preparation work ready, we can now show the following convergence theorem. We define, for some \( S \subseteq \mathbb{C} \) and some polynomial \( p \), the scalar \( \| p \|_S = \sup_{z \in S} |p(z)| \). Also, we assume that \( A \) is diagonalizable, meaning that \( A = V \Lambda V^{-1} \) for some nonsingular matrix \( V \) and diagonal matrix \( \Lambda \).

**Theorem 1.1.** At step \( n \) of the GMRES iteration, the residual \( r_n \) satisfies

\[
\frac{\| r_n \|}{\| b \|} \leq \inf_{p_n \in P_n} \| p_n(A) \| \leq \kappa(V) \inf_{p_n \in P_n} \| p_n \|_{\Lambda(A)},
\]

where \( \Lambda(A) \) is the set of eigenvalues of \( A \) and \( \kappa(V) = \| V \| \| V^{-1} \| \) is the condition number of \( V \).

**Proof.** By problem statement 5, for any \( p_n \in P_n \),

\[
\frac{\| r_n \|}{\| b \|} \leq \frac{\| p_n(A)b \|}{\| p_n \|}\|
\]

\[
\Rightarrow \frac{\| r_n \|}{\| b \|} \leq \inf_{p_n \in P_n} \| p_n(A) \|
\]

\[
\leq \inf_{p_n \in P_n} \| V p_n(\Lambda) V^{-1} \|
\]

\[
\leq \kappa(V) \inf_{p_n \in P_n} \| p_n(\Lambda) \|
\]

\[
= \kappa(V) \inf_{p_n \in P_n} \left( \sum_{i=1}^{m} p_n^2(\lambda_i) \right)^{\frac{1}{2}} = \kappa(V) \inf_{p_n \in P_n} \| p_n \|_{\Lambda(A)}.
\]

\[ \square \]

### 1.3 Conjugate Gradient (CG)

#### 1.3.1 The CG Iteration

Again we want to solve some nonsingular system of linear equations \( Ax = b \), but this time we assume \( A \) is a real, symmetric and positive definite \( m \times m \) matrix. Again let \( x_*=A^{-1}b \) be the solution and \( \mathcal{K}_n \) be the corresponding nth Krylov subspace. Because of the nature of the matrix \( A \), we can define the \( A \)-norm on \( \mathbb{R}^m \) by \( \| x \|_A = \sqrt{x^T A x} \). The CG iteration aims to find \( x_n \in \mathcal{K}_n \) at step \( n \) which minimizes \( \| e_n \|_A = \| x_* - x_n \|_A \). The algorithm is as follows,

---

**Algorithm 2: CG Iteration**

\[
x_0 = 0, r_0 = b, p_0 = r_0
\]

**for** \( n = 1, 2, \ldots \) **do**

\[
\alpha_n = \frac{(r_{n-1}^T r_{n-1})(p_{n-1}^T A p_{n-1})}{(r_{n-1}^T A p_{n-1})^2} \quad (1)
\]

\[
x_n = x_{n-1} + \alpha_n p_{n-1} \quad (2)
\]

\[
r_n = r_{n-1} - \alpha_n A p_{n-1} \quad (3)
\]

\[
\beta_n = \frac{(r_n^T r_n)(r_{n-1}^T r_{n-1})}{(r_{n-1}^T r_{n-1})(r_n^T r_{n-1})} \quad (4)
\]

\[
p_n = r_n + \beta_n p_{n-1} \quad (5)
\]

**end**

---

3
1.3.2 Optimality of CG

In this section we are going to show that the algorithm given above does the job, i.e., it does minimize $\|e_n\|_A = \|x_r - x_n\|_A$. We first prove some orthogonality properties of this algorithm and then use them to derive the optimality of CG.

**Theorem 1.2.** In algorithm 2, as long as the iteration has not yet converged, i.e., $r_{n-1} \neq 0$, the algorithm proceeds without divisions by zero, and we have the following identities of subspaces:

$$K_n = \langle x_1, x_2, \ldots, x_n \rangle = \langle p_0, p_1, \ldots, p_{n-1} \rangle = \langle r_0, r_1, \ldots, r_{n-1} \rangle.$$  

(6)

Moreover, the residuals are orthogonal,

$$r_n^T r_j = 0, \hspace{1em} j < n,$$

(7)

and the search directions are $A$-conjugate,

$$p_n^T Ap_j = 0, \hspace{1em} j < n.$$  

(8)

**Proof.** If $r_{n-1} \neq 0$, then $\alpha_n \neq 0$. By $x_0 = 0$ and equation 2 in algorithm 2, we have $\langle x_1, x_2, \ldots, x_n \rangle = \langle p_0, p_1, \ldots, p_{n-1} \rangle$. By $p_0 = r_0$ and equation 5 in algorithm 2, we have $\langle p_0, p_1, \ldots, p_{n-1} \rangle = \langle r_0, r_1, \ldots, r_{n-1} \rangle$. Finally, by $r_0 = b$ and equation 3 in algorithm 2, we have $\langle r_0, r_1, \ldots, r_{n-1} \rangle = \langle b, Ab, \ldots, A^{n-1}b \rangle$. We accordingly obtain equation 6.

We want to show property 7 by induction. When $n = 1$, $j = 0$, by $p_0 = r_0$ and equation 1,3 in algorithm 2,

$$r_1^T r_0 = r_0^T r_0 - \alpha_1 p_0^T Ar_0 = r_0^T r_0 - \frac{r_0^T r_0}{p_0^T Ap_0} = 0.$$  

Assume this property holds for some $n < m$. Then, we look at the case $n + 1$:

$$r_{n+1}^T r_j = r_n^T r_j - \alpha_{n+1} p_n^T Ar_j.$$  

If $j < n$, $r_n^T r_j = 0$ by the induction hypothesis and $\alpha_{n+1} p_n^T Ar_j = 0$ by property 8; if $j = n$,

$$r_{n+1}^T r_n = r_n^T r_n - \alpha_{n+1} p_n^T Ar_n$$

$$= r_n^T r_n - \frac{r_n^T r_n}{p_n^T Ap_n} p_n^T Ar_n \quad \text{(equation 1 in algorithm 2)}$$

$$= r_n^T r_n - \frac{r_n^T r_n}{\beta_n p_n^T Ap_n} p_n^T Ar_n \quad \text{(equation 5 in algorithm 2)}$$

$$= 0 \quad \text{(} \beta_n p_n^T Ap_{n-1} = 0 \text{ by equation 8)}$$

4
Property 8 is also shown by induction. If \( n = 1, j = 0 \), by equation 5 in algorithm 2,
\[
p_1^T A p_0 = r_1^T A p_0 + \beta_1 r_0^T A p_0
\]
\[
= r_1^T A p_0 + \frac{r_1^T r_1}{r_0^T r_0} p_0^T A p_0 \quad \text{(equation 4 in algorithm 2)}
\]
\[
= r_1^T A p_0 + \frac{r_1^T (r_0 - \alpha_1 A p_0)}{p_0^T A p_0} p_0^T A p_0 \quad \text{(equation 3 in algorithm 2)}
\]
\[
= r_1^T A p_0 + \frac{-r_1^T \alpha_1 A p_0}{\alpha_1 p_0^T A p_0} p_0^T A p_0 \quad \text{(equation 1 in algorithm 2, equation 7)}
\]
\[
= 0.
\]
Assume the property holds for some \( n < m \). Consider the case \( n + 1 \):
\[
p_{n+1}^T A p_j = r_{n+1}^T A p_j + \beta_{n+1} r_0^T A p_0.
\]
If \( j < n \), by equation 3 in algorithm 2 and property 7, rewrite the first term on the right hand side as \( r_{n+1}^T A p_j = r_{n+1}^T r_{j+1} - \frac{r_{j+1}^T A e}{\alpha_j} = 0 \). Also, the second term is zero due to the induction hypothesis. If \( j = n \), we can show that \( p_{n+1}^T A p_n = 0 \) by the same logic as the base case.

Then, the optimality theorem below follows directly from the properties in theorem 1.2.

**Theorem 1.3.** In algorithm 2, if the iteration has not yet converged, i.e., \( r_{n-1} \neq 0 \), then \( x_n \) is the unique point in \( \mathcal{K}_n \) that minimizes \( \|e_n\|_A \). The convergence is monotonic,
\[
\|e_n\|_A \leq \|e_{n-1}\|_A,
\]
and \( e_n = 0 \) is achieved for some \( n \leq m \).

**Proof.** Consider some \( x' \in \mathcal{K}_n \) such that \( x'_n = x_n - \Delta x, \Delta x \in \mathcal{K}_n \). The corresponding error has, by symmetry of \( A \),
\[
\|e'_n\|_A^2 = \|x'_n - (x_n - \Delta x)\|_A^2 = \|e_n + \Delta x\|_A^2 = (e_n + \Delta x)^T A (e_n + \Delta x) = e_n^T A e_n + \Delta x^T A (\Delta x) + 2 e_n^T A (\Delta x)
\]
\[
= e_n^T A e_n + \Delta x^T A (\Delta x) + 2 \sum_{j=0}^{n-1} c_j r_j \quad \text{(property 6)}
\]
\[
= e_n^T A e_n + \Delta x^T A (\Delta x). \quad \text{(property 7)}
\]
Because \( A \) is positive definite, \( (\Delta x)^T A (\Delta x) \geq 0 \forall \Delta x \in \mathcal{K}_n \) and \( (\Delta x)^T A (\Delta x) = 0 \) if and only if \( \Delta x = 0 \). So, \( \|e'_n\|_A^2 \) is minimized when \( x'_n = x_n \), which implies that \( x_n \) is the unique point in \( \mathcal{K}_n \) that minimizes \( \|e_n\|_A \).

By the same logic as the first paragraph of section 1.2.2, the CG iteration converges monotonically and at some step \( n \leq m \).
1.3.3 Convergence of CG

In this section, we discuss the rate of convergence of the CG iteration in more depth. As what we’ve done for GMRES, we reduce the CG iteration to an approximation problem. Because \( x_n \in \mathcal{K}_n \), write \( x_n = \sum_{i=0}^{n-1} c_i A^i b = q_n(A)b \), where \( q_n(x) = \sum_{i=0}^{n-1} c_i x^i \). Then,

\[
e_n = x_s - x_n = A^{-1} b - q_n(A)b = (I - q_n(A))A^{-1} b = p_n(A)e_0,
\]

where \( p_n(x) = 1 - x q_n(x) \in P_n = \{1 + \sum_{i=1}^{n} c_i x^i \mid c_i \in \mathbb{R}\} \) and \( e_0 = x_s - x_0 = x_s \). So, the CG approximation problem is to

\[
\text{find } p_n \in P_n \text{ such that } \|p_n(A)e_0\|_A \text{ is minimal.} \quad (11)
\]

**Theorem 1.4.** If the CG iteration has not yet converged before step \( n \), i.e., \( r_{n-1} \neq 0 \), then problem 11 has a unique solution \( p_n \in P_n \) and \( x_n \) has error \( e_n = p_n(A)e_0 \) for this same polynomial \( p_n \). Consequently we have

\[
\frac{\|e_n\|_A}{\|e_0\|_A} = \inf_{p \in P_n} \frac{\|p(A)e_0\|_A}{\|e_0\|_A} \leq \inf_{p \in P_n} \max_{\lambda \in \Lambda(A)} |p(\lambda)|, \quad (12)
\]

where \( \Lambda(A) \) denotes the spectrum of \( A \).

**Proof.** The uniqueness of \( p_n \) follows directly from the uniqueness of \( x_n \) in theorem 1.3. Besides that,

\[
\frac{\|e_n\|_A}{\|e_0\|_A} = \inf_{p \in P_n} \frac{\|p(A)e_0\|_A}{\|e_0\|_A} \quad \text{(the CG approximation problem 11)}
\]

\[
= \inf_{p \in P_n} \left( \frac{\sum_{j=1}^{m} a_j p(\lambda_j) v_j^T A (\sum_{j=1}^{m} a_j p(\lambda_j) v_j)}{\sum_{j=1}^{m} a_j v_j^T A (\sum_{j=1}^{m} a_j v_j)} \right)^{1/2}
\]

\[
= \inf_{p \in P_n} \left( \sum_{j=1}^{m} a_j^2 p^2(\lambda_j) \lambda_j \right)^{1/2}
\]

\[
\leq \inf_{p \in P_n} \max_{\lambda \in \Lambda(A)} |p(\lambda)|.
\]

\[\square\]

Theorem 1.4 gives us an estimate of the rate of convergence for the CG iteration depending on the spectrum of \( A \). Accordingly, some special features of the spectrum can result in more specific corollaries. Below are two important examples.

**Theorem 1.5.** If \( A \) has only \( n \) distinct eigenvalues, then the CG iteration converges in at most \( n \) steps.

**Proof.** Consider \( p(z) = \prod_{j=1}^{n} (1 - \frac{z}{\lambda_j}) \in P_n \). Then, \( \max_{\lambda \in \Lambda(A)} |p(\lambda)| = 0 \). So, by inequality 12,

\[
\frac{\|e_n\|_A}{\|e_0\|_A} \leq \inf_{p \in P_n} \max_{\lambda \in \Lambda(A)} |p(\lambda)| = 0 \Rightarrow \|e_n\|_A = 0 \text{ which means the CG iteration converges within } n \text{ steps.} \quad \square
\]
Theorem 1.6. The $A$-norms of the errors in the CG iteration satisfy

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq \frac{2}{(\sqrt{\kappa+1}/\sqrt{\kappa-1})^n} \leq 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n,$$

(13)

where $\kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$ and $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$ are respectively the largest and smallest eigenvalues of $A$.

Proof. Consider $p(x) = \frac{T_n(\gamma - \frac{2x}{\lambda_{\text{max}} - \lambda_{\text{min}}})}{T_n(\gamma)}$, where $T_n(x) = \frac{1}{2}((x - \sqrt{x^2 - 1})^n + (x + \sqrt{x^2 - 1})^n)$ is the $n$th Chebyshev polynomial of the first kind and $\gamma = \frac{\lambda_{\text{max}} + \lambda_{\text{min}}}{\lambda_{\text{max}} - \lambda_{\text{min}}} = \frac{\kappa+1}{\kappa-1}$. We are going to show that $\max_{\lambda \in \Lambda(A)} |p(\lambda)| \leq \frac{2}{(\sqrt{\kappa+1}/\sqrt{\kappa-1})^n}$. Let $x = \frac{1}{2}(z + z^{-1})$. Then, $T_n(x) = \frac{1}{2}(z^n + z^{-n})$. We want to write $T_n(\gamma)$ in terms of $\kappa$. If $x = \gamma$, then

$$\frac{1}{2}(z + z^{-1}) = \frac{\kappa+1}{\kappa-1} \Rightarrow \frac{1}{2}z^2 - \frac{\kappa+1}{\kappa-1}z + \frac{1}{2} = 0 \Rightarrow z = \frac{\kappa+1}{\kappa-1} \pm \left(\frac{\kappa+1}{\kappa-1}\right)^2 - 1 \right)^{1/2} = \left(\frac{\sqrt{\kappa+1}}{\sqrt{\kappa-1}}\right)^{\pm 1}.$$

So, $T_n(\gamma) = \frac{1}{2}\left((\frac{\sqrt{\kappa+1}}{\sqrt{\kappa-1}})^n + (\frac{\sqrt{\kappa+1}}{\sqrt{\kappa-1}})^{-n}\right)$. Because $|\gamma - \frac{2x}{\lambda_{\text{max}} - \lambda_{\text{min}}}| \leq 1$, we have $|T_n(\gamma - \frac{2x}{\lambda_{\text{max}} - \lambda_{\text{min}}})| \leq 1$. So, by theorem 1.4,

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq \inf_{p \in P_n} \max_{\lambda \in \Lambda(A)} |p(\lambda)| \leq \max_{\lambda \in \Lambda(A)} |p(\lambda)| \leq \frac{2}{(\sqrt{\kappa+1}/\sqrt{\kappa-1})^n} \leq \frac{2}{1 + (\sqrt{\kappa+1}/\sqrt{\kappa-1})} \leq 2 \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n.$$

By the result above, as $\kappa \to \infty$, $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \approx 1 - \frac{2}{\sqrt{\kappa}}$, which is a relatively fast rate of convergence compared with other iterative methods.

2 Preconditioning

As the previous convergence theorems imply, the rate of convergence is dependent on the condition number of $A$. If we significantly reduce the condition number, the algorithm will converge in a much faster speed. This can actually be achieved by preconditioning. Given an $m \times m$ nonsingular system $Ax = b$, we can form an equivalent system $M^{-1}Ax = M^{-1}b$, where $M$ is chosen to be an $m \times m$ nonsingular matrix which makes $\kappa(M^{-1}A)$ small. This matrix $M$ is called a (left) preconditioner. Specially, if $A$ is hermitian positive definite, we usually transform the original system in a slightly different way to retain this property in the new system. Assume $M$ is the corresponding preconditioner, which is also hermitian positive...
Because $M^{-1}A$ may not be hermitian positive definite, we (Cholesky) decompose $M = LL^*$ and consider the new system

$$(L^{-1}AL^{-*})(L^*x) = L^{-1}b,$$

(14)

where $L^{-1}AL^{-*}$ satisfies the required property. This transformation works because $L^{-1}AL^{-*}$ is similar to $L^{-*}L^{-1}A = M^{-1}A$ so they share the same eigenvalues. With the new systems now, we can then apply our original iterative methods to find the solutions, some of which are exactly the same as before and some of which can be transformed back to the previous solutions. Although the idea of preconditioning is straightforward, knowing which preconditioners to use is a complicated subject and I will mention a few examples in section 2.2.

2.1 Preconditioned Conjugate Gradient (PCG)

We now look at how the idea of preconditioning is applied to the conjugate gradient method. The system of concern is $Ax = b$, where $A$ is a symmetric and positive definite $m \times m$ matrix. Suppose $M$ is the preconditioner we want use, which is also symmetric and positive definite. Following the idea of equation 14, we rewrite $M = LL^T$ and then construct the new system below:

$$\tilde{A}\tilde{x} = \tilde{b},$$

where $\tilde{A} = L^{-1}AL^{-*}$, $\tilde{x} = L^T x$, $\tilde{b} = L^{-1}b$.

(15)

It’s not hard to see that $\tilde{A}$ is still symmetric and positive definite. So, PCG is actually just CG applied to system 15 above, which results in the following PCG algorithm:

**Algorithm 3: PCG**

$x_0 = 0, r_0 = b, z_0 = M^{-1}r_0, p_0 = z_0$

for $n = 1, 2, \ldots$ do

$$\alpha_n = \frac{(r_{n-1}^T z_{n-1})}{(p_{n-1}^T A p_{n-1})}$$

(1)

$$x_n = x_{n-1} + \alpha_n p_{n-1}$$

(2)

$$r_n = r_{n-1} - \alpha_n A p_{n-1}$$

(3)

$$z_n = M^{-1}r_n$$

(4)

$$\beta_n = \frac{(r_n^T z_n)}{(r_{n-1}^T z_{n-1})}$$

(5)

$$p_n = z_n + \beta_n p_{n-1}$$

(6)

end

2.2 Examples of Preconditioners

2.2.1 Diagonal Scaling

Again we consider the $m \times m$ nonsingular system $Ax = b$. A preconditioner of this system can be $M = \text{diag}(A)$, which is a simple but very important example. More generally, we can also use the matrix $M = \text{diag}(c)$, $c \in \mathbb{C}^m$, as a preconditioner.

2.2.2 Incomplete Cholesky Factorization

We obtain the matrix $L$ in equation 14 through Cholesky factorization of the matrix $M$. During this process, the sparsity of the matrix $M$ is usually destroyed. To preserve this
property, we instead use incomplete Cholesky factorization, denoted with \( M = \tilde{L}\tilde{L}^* \), where the matrix \( \tilde{L} \) is at least as sparse as the matrix \( M \).

### 2.2.3 Preconditioning for Spectral Discretizations

Consider the one-dimensional problem,

\[
-\frac{d^2u}{dx^2} = f \quad \text{in } (0, 2\pi)
\]

\[
u \text{ 2}\pi\text{-periodic.}
\]

Let \( L \) denote the operator \(-\frac{d^2}{dx^2}\) with \(2\pi\)-periodic boundary conditions in \((0, 2\pi)\), so equation 16 can be rewritten as

\[
Lu = f.
\]

Suppose \( L \) is the spectral collocation discretization of \( L \). The high accuracy of the spectral methods sacrifices the sparsity of the matrices involved and thus creates the demand of preconditioners. There are several choices for the preconditioners which I will introduce below.

One can use finite-difference method to approximate the preconditioner. Let \( H^{(fd)} \) denote the second-order finite-difference discretization of \( L \), given by

\[
-\frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2} = f_j, \quad j = 0, 1, \cdots, N - 1,
\]

where \( \Delta x = \frac{2\pi}{N}, \) \( x_j = j\Delta x, \) \( u_j \approx u(x_j) \) and \( f_j = f(x_j) \). Convert equation 18 to the preconditioned one below:

\[
(H^{(fd)})^{-1}Lu = (H^{(fd)})^{-1}f.
\]

Because the eigenfunctions of \( H^{(fd)} \) and \( L \) are the same, the eigenvalues of the operator \((H^{(fd)})^{-1}L\) are accordingly

\[
\Lambda_p^{(fd)} = p^2(\lambda_p^{(fd)})^{-1} = \frac{(p\Delta x/2)^2}{\sin^2(p\Delta x/2)},
\]

where \( p = -\frac{N}{2\pi}, \cdots, \frac{N}{2\pi} \).

The preconditioner can also be found through finite-difference method. First, we define the stiffness and mass matrices and the spectral matrices on the grid \( \{x_j\}, j = 0, \cdots, N \) by:

\[
(K_{FE})_{ij} = \int_0^{2\pi} \varphi_{j,x}\varphi_{i,x} \, dx, \quad (M_{FE})_{ij} = \int_0^{2\pi} \varphi_j\varphi_i \, dx,
\]

\[
K_{ij} = \int_0^{2\pi} \psi_{j,x}\psi_{i,x} \, dx, \quad M_{ij} = \int_0^{2\pi} \psi_j\psi_i \, dx.
\]

where \( \varphi_j \)'s and \( \psi_j \)'s are respectively the periodic and trigonometric characteristic Lagrange functions at \( x_j \)'s.
This preconditioning method relies on a simple iterative method solving problems of the form 18, called Richardson’s method. The idea is the same as those introduced in section 1: given any initial guess \( v_0 \) of \( u \), we recursively approximate the actual solution \( u \) via

\[
\begin{align*}
  r_n &= f - \mathcal{L}v_n, \quad v_{n+1} = v_n + \omega r_n, \\
  r_{n}^s &= f - M^{-1}Kv_n, \quad K_{FE}(v_{n+1} - v_n) = \omega M_{FE}r_{n}^s;
\end{align*}
\]

where \( \omega \) is a relaxation parameter. Preconditioning this method in the strong form results in

\[
\begin{align*}
  r_{n}^s &= f - M^{-1}Kv_n, \quad K_{FE}(v_{n+1} - v_n) = \omega M_{FE}r_{n}^s;
\end{align*}
\]

while the weak form looks like

\[
\begin{align*}
  r_{n}^w &= Mf - Kv_n, \quad K_{FE}(v_{n+1} - v_n) = \omega r_{n}^w.
\end{align*}
\]

If we apply the method above to problem 16, we have the eigenvalues of the matrix \( K_{(FE)}^{-1}M_{(FE)}M^{-1}K \) and the matrix \( K_{(FE)}^{-1}K \), which are the matrices governing convergence in the strong and weak form, respectively as:

\[
\Lambda^{(fes)}_p = \frac{(p\Delta x/2)^2}{\sin^2(p\Delta x/2)} \frac{2 + \cos(p\Delta x)}{3}, \quad \Lambda^{(few)}_p = \frac{(p\Delta x/2)^2}{\sin^2(p\Delta x/2)} \frac{2 + \cos(p\Delta x)}{3},
\]

where \( p = -\frac{N}{2+1}, \ldots, \frac{N}{2-1} \).
References

[1] Numerical Linear Algebra, Volume 50 of Other Titles in Applied Mathematics, Lloyd N. Trefethen, David Bau, III