CAAM 454 Lecture Notes:
Iterative Methods for Linear Systems

D. C. Sorensen
Rice University

Draft of 21 January 2015

— NOT FOR DISTRIBUTION —
# Contents

## 0 A Motivating Example

## 1 Basic Iterative Methods

1.1 Splitting Methods ........................................... 3
1.2 The Jacobi Iteration ......................................... 4
1.3 The Gauss-Seidel Iteration ................................. 5
1.4 Successive Over Relaxation (SOR): ......................... 6
1.5 Convergence of Splitting Methods: ....................... 7
1.6 Convergence of SOR ........................................ 9
1.7 The Effect of Non-Normality ................................. 9
1.8 The Lyapunov Equation and ADI ........................... 9
1.9 Iterative solution methods ................................ 12
   1.9.1 Smith–ADI for Lyapunov equations ............... 14
   1.9.2 Convergence of the Smith-ADI method .......... 15
   1.9.3 Shift selection, multiple shifts ................. 16

## 2 Theory of Eigenvalue Decompositions

2.1 Eigenvalues and Eigen-Decompositions .................. 19
   2.1.1 Exercises ........................................... 20
2.2 Schur Decomposition .................................... 21
   2.2.1 Exercises ........................................... 23
2.3 Eigen-decompositions of $\mathbb{C}^n$ ...................... 24
   2.3.1 Exercise ............................................ 24
2.4 Invariant Subspaces ..................................... 25
   2.4.1 Exercises ........................................... 25
   2.4.2 Exercise ............................................ 28
   2.4.3 Exercises ........................................... 29
2.5 The Cayley-Hamilton Theorem ............................ 29
2.5.1 Exercises ........................................... 30
2.6 The Jordan Canonical Form ......................... 31
  2.6.1 Exercise ........................................... 31
  2.6.2 Exercise ........................................... 35
  2.6.3 Exercise ........................................... 37

3 Krylov Methods ....................................... 41
  3.1 Richardson’s Iteration .............................. 41
  3.2 The Krylov Subspace: .............................. 42
    3.2.1 The Arnoldi process ......................... 43
    3.2.2 Iterative Methods via Arnoldi Factorization 45
    3.2.3 An Alternate Derivation ..................... 47
    3.2.4 The GMRES and FOM Polynomials ........... 48
    3.2.5 Implementation Issues ...................... 51
    3.2.6 Symmetry and the Lanczos process .......... 55
    3.2.7 Quadratic Functions and Symmetric Linear Systems ......................... 57
    3.2.8 The Conjugate Gradient Method: Lanczos + FOM = CG ................. 58
    3.2.9 The bi-Lanczos process ...................... 62
0 · A Motivating Example
1 · Basic Iterative Methods

This chapter will briefly review a small selected set of classic algorithms for the iterative solution of linear systems

\[ Ax = b, \text{ with } A \in \mathbb{R}^{n \times n}, \ b, x \in \mathbb{R}^n, \]

where \( A \) is assumed to be nonsingular unless stated otherwise. The goal is to obtain a numerical approximation to the solution vector \( x \) iteratively without computing a matrix factorization.

1.1 Splitting Methods

To begin, consider a *splitting* of the matrix \( A \) of the form \( A = M - N \). This should be thought of as choosing or constructing a matrix \( M \) that somehow approximates \( A \) with the property that \( Mz = y \) is “easy” to solve for arbitrary \( y \in \mathbb{R}^n \). The matrix \( N \) is then simply defined as whatever is left over: \( N = M - A \). An example might be to take \( M \) to be the main diagonal of \( A \) if \( A \) is somehow “diagonally dominant”. Notice that

\[ Ax = b \iff Mx = Nx + b. \]

This trivial rearrangement leads immediately to an iterative method. A given iterate \( x^{(k)} \) will provide

\[ Mx^{(k+1)} = Nx^{(k)} + b. \]

The iteration is initiated with a guess \( x^{(0)} \) (the initial guess).
Chapter 1. Basic Iterative Methods

The choice of $M$ is of course key to the performance of such an iteration. There are numerous possibilities, but three of these stand out as classic and fundamental. To study and compare these approaches it is convenient to decompose $A$ into

$$A = D - E - F,$$

where $D$ is the matrix consisting of just the diagonal of $A$ with $-E$ consisting of the strict lower triangle of $A$ and $-F$ consisting of the strict upper triangle of $A$.

A Glimpse at Convergence:

$$Mx^{(k+1)} = Nx^{(k)} + b$$

Thus

$$M(x^{k+1} - x) = N(x^{(k)} - x).$$

The cumulative effect of this observation is

$$e^{(k)} = M^{-1}Ne^{(k-1)} = (M^{-1}N)^2e^{(k-2)} = (M^{-1}N)^3e^{(k-3)} = \ldots = (M^{-1}N)^ke^{(0)},$$

where $e^{(k)} \equiv x^{(k)} - x$. It follows that

$$\|x^{k} - x\| \leq \|(M^{-1}N)^k\|\|x^{(0)} - x\|. \leq \|M^{-1}N\|^k\|x^{(0)} - x\|. \ (1.1)$$

Hence a splitting method will converge to the solution $x$ from any initial guess $x^{(0)}$ at a linear rate if there is a norm $\|\|$ such that $\|M^{-1}N\| < 1$. Unfortunately, this condition is generally difficult to satisfy. A full analysis of convergence requires a detailed understanding of eigenvalue decomposition of the iteration matrix $M^{-1}N$.

1.2 The Jacobi Iteration

The simplest such method is attributed to Jacobi. The choices defining this method are

$$M = D, \quad N = E + F$$

so that $x^{(k+1)} = D^{-1}[(E + F)x^{(k)} + b]$.

Denoting the $i$-th component of $x^{(k)}$ as $x_i^{(k)}$ gives the following scalar form of the iteration

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[ -\sum_{j<i} a_{ij} * x_j^{(k)} - \sum_{j>i} a_{ij} x_j^{(k)} + b_i \right], \quad \text{for } i = 1, 2, \ldots, n.$$
1.3 The Gauss-Seidel Iteration

A Matlab style code for this iteration would be

```matlab
x = input(' initial guess ')
k = 0;
while ('not converged' & k < Kmax),
    for i = 1:n,
        xp(i) = (b(i) - A(i,1:i-1)*x(1:i-1) - A(i,i+1:n)*x(i+1:n))/A(i,i);
    end
    x = xp;
k = k+1;
end
```

Note that two vectors of storage for $x, \ xp$ are required for this iteration.

1.3 The Gauss-Seidel Iteration

An immediate criticism of the Jacobi iteration is that the diagonal is usually not a very good approximation to $A$. More of the matrix $A$ can be represented by $M$ if the lower triangle of $A$ is included. The “easy to solve” requirement is fulfilled since $M = D - E$ is lower triangular. The Gauss-Seidel (G-S) iteration is thus defined by

$$M = D - E, \ N = F \text{ so that } x^{(k+1)} = (D - E)^{-1} \left[ Fx^{(k)} + b \right].$$

However, rather than explicitly solving a triangular system note that

$$ (D - E)x^{(k+1)} = Fx^{(k)} + b$$

and an examination of the $i$-th component gives

$$a_{ii}x_i^{(k+1)} + \sum_{j<i} a_{ij}x_j^{(k+1)} = -\sum_{j>i} a_{ij}x_j^{(k)} + b_i.$$

A Matlab style code for this iteration would be

```matlab
x = input(' initial guess ')
k = 0;
while ('not converged' & k < Kmax),
    for i = 1:n,
        xp(i) = (b(i) - A(i,1:i-1)*x(1:i-1) - A(i,i+1:n)*x(i+1:n))/A(i,i);
    end
    x = xp;
k = k+1;
end
```
for i = 1:n,
    x(i) = (b(i) - A(i,1:i-1)*x(1:i-1) - A(i,i+1:n)*x(i+1:n))/A(i,i);
end
k = k+1;
end

Notice that at the i-th step of the inner i-loop, the leading $i - 1$ components of $x^{(k+1)}$ occupy $x(1:i-1)$ while the trailing $n - i$ components of $x^{(k)}$ occupy $x(i+1:n)$. Hence, not only is the G-S iteration better than Jacobi (in a sense that will be explained later), it takes less storage. Only one vector of storage is required for the G-S iteration.

If $A$ is strictly row-diagonally dominant (meaning $\sum_{j \neq i} |a_{ij}| < |a_{ii}|$ for $1 \leq i \leq n$) then $\|D^{-1}(E + F)\|_{\infty} < 1$. Hence, the Jacobi iteration is convergent for diagonally dominant $A$. It turns out that the G-S iteration is convergent whenever Jacobi is convergent so G-S is convergent on diagonally dominant matrices as well and the rate of convergence is faster. Moreover, the more diagonally dominant the matrix, the faster the Jacobi and hence the G-S iterations will converge.

1.4 Successive Over Relaxation (SOR):

Since “diagonally dominant is good” perhaps a method based upon artificially enhancing the diagonal of the matrix would lead to a very good or even an optimal splitting method. So, to this end, consider the following:

$$Ax = b \iff (D + \omega A)x = Dx + \omega b.$$  

Using $A = D - E - F$, rearrange the equation on the right as

$$(D - \omega E)x = [(1 - \omega)D + \omega F]x + \omega b$$

to obtain the Successive Overrelaxation Method (SOR) as

$$x^{(k+1)} = M_{\omega}^{-1} \left[ N_{\omega} x^{(k)} + \omega b \right],$$

where

$$M_{\omega} = D - \omega E \quad \text{and} \quad N_{\omega} = (1 - \omega)D + \omega F.$$  

Now, the parameter $\omega$ might be adjusted so that the iteration matrix $M_{\omega}^{-1}N_{\omega}$ has “optimal” convergence properties.
1.5 Convergence of Splitting Methods:

The scalar iteration may be derived in a manner similar to the derivation of the scalar G-S iteration:

\[ a_{ii}x_i^{(k+1)} + \omega \sum_{j<i} a_{ij}x_j^{(k+1)} = (1 - \omega)a_{ii}x_i^{(k)} - \omega \sum_{j>i} a_{ij}x_j^{(k)} + \omega b_i. \]

The resulting Matlab style code would be

```matlab
x = input(' initial guess ')
omega = input(' value of omega ')
k = 0;
while ('not converged' & k < Kmax),
    for i = 1:n,
        x(i) = (1 - omega)*x(i) + omega*(b(i) - A(i,1:i-1)*x(1:i-1) ...
            - A(i,i+1:n)*x(i+1:n))/A(i,i)
    end
    k = k+1;
end
```

Again, notice that at the i-th step of the inner i-loop, the leading i − 1 components of \( x^{(k+1)} \) occupy \( x(1:i-1) \) while the trailing \( n-i \) components of \( x^{(k)} \) occupy \( x(i+1:n) \). Thus, the SOR iteration also only requires one vector of storage.

1.5 Convergence of Splitting Methods:

The key to analyzing the convergence of splitting methods lies with inequality (1.1). It has already been noted that if one can find a consistent norm \( \| \cdot \| \) with \( \| M^{-1}N \| < 1 \) then the method is convergent from any starting point \( x^{(0)} \).

To analyze the general case, consider the following lemma.

**Lemma 1.1.** Let \( R \in \mathbb{C}^{n \times n} \) be upper triangular and let \( \epsilon > 0 \) be given. Then there is a consistent norm \( \| \cdot \|_\epsilon \) such that

\[ \|R\|_\epsilon \leq \rho(R) + \epsilon. \]

**Proof.** First, let \( 0 < \delta < 1 \) be arbitrary and put \( D = \text{diag}(\delta, \delta^2, \ldots, \delta^n) \),
and consider the scaled matrix \( D^{-1}RD \). Note

\[
\left\{ \begin{array}{ll}
\rho_{ij} \delta^{j-i}, & j > i \\
\rho_{ii}, & j = i \\
0, & j < i,
\end{array} \right.
\]

where \( \rho_{ij} = e_i^T R e_j \). Hence,

\[
\| e_i^T D^{-1} R e_j \|_1 = |\rho_{ii}| + \sum_{j>i} |\rho_{ij} \delta^{j-i}| \leq |\rho_{ii}| + \sum_{j>i} |\rho_{ij}| \delta,
\]

since \( 0 < \delta < 1 \). From this inequality it readily follows that

\[
\| D^{-1} RD \|_\infty \leq \rho(G) + \delta \| R \|_\infty.
\]

Now, just choose \( \delta < \epsilon / (\| R \|_\infty) \) and define

\[
\| G \|_\epsilon = \| D^{-1} GD \|_\infty
\]

for any matrix \( G \in \mathbb{C}^{n \times n} \). It is easily shown that \( \| \cdot \|_\epsilon \) is a consistent norm and the stated result is established.

**Corollary 1.2.** Let \( R \in \mathbb{C}^{n \times n} \) be upper triangular with \( \rho(R) < 1 \). Then \( R^k \to 0 \) as \( k \to \infty \).

**Proof.** Let \( \epsilon \) satisfy \( \rho(R) < \rho(R) + \epsilon < 1 \). From Lemma (1.1) there is a consistent norm \( \| \cdot \| \) such that \( \| R \| < \rho(R) + \epsilon < 1 \). Since this norm is consistent,

\[
\| R^k \| \leq \| R \|^k < (\rho(R) + \epsilon)^k \to 0 \quad \text{as} \quad k \to \infty.
\]

Hence, \( R^k \to 0 \) as claimed.

The general convergence result is obtained with the following theorem.

**Theorem 1.3.** Let \( G \in \mathbb{C}^{n \times n} \). Then \( \lim_{k \to \infty} G^k = 0 \) if and only if the spectral radius \( \rho(G) < 1 \).

**Proof.** Suppose \( G^k \to 0 \) as \( k \to \infty \). Let \( \| \cdot \| \) be a consistent norm and let \( \lambda \in \sigma(G) \) such that \( |\lambda| = \rho(G) \) with corresponding eigenvector \( v \) such that \( \| v \| = 1 \). Then

\[
\rho(G)^k = |\lambda|^k = \| G^k v \| \leq \| G^k \| \to 0 \quad \text{as} \quad k \to \infty.
\]

Hence \( \rho(G) < 1 \) must hold.

Now, suppose \( \rho(G) < 1 \). Let \( G = QR \) be a Schur decomposition of \( G \). Then \( \rho(R) = \rho(G) < 1 \) and due to Corollary (1.2), \( G^k Q = QR^k \to 0 \) which in turn implies \( G^k \to 0 \) as \( k \to \infty \).
1.6 Convergence of SOR

In addition to providing a necessary and sufficient condition for convergence of a splitting method, the previous discussion provides a rate of convergence. The inequality (1.1) together with Theorem (1.3) infers the existence of a consistent norm such that

\[ \|(x^k - x)\| \leq \|(M^{-1}N)^k\|\|(x^{(0)} - x)\| \leq (\rho(M^{-1}N) + \epsilon)^k\|(x^{(0)} - x)\| \]

given any \( \epsilon > 0 \). Hence convergence is linear with an asymptotic rate of convergence equal to \( \rho(M^{-1}N) \). This notion can be somewhat misleading in the case \( M^{-1}N \) is highly non-normal, but it holds asymptotically in any case. It is apparent that a smaller spectral radius will provide more rapid convergence. It is not generally possible to provide an optimal splitting in the sense of minimal spectral radius. However, it is possible to select an optimal value of the parameter \( \omega \) in the SOR iteration to minimize the spectral radius \( \rho(M^{-1}N_\omega) \).

More needs to be added to SOR convergence

1.7 The Effect of Non-Normality

Recall that a matrix \( G \) is non-normal if and only if it has a Schur decomposition \( GQ = QR \) with at least one nonzero element the strict upper triangle of \( R \).

1.8 The Lyapunov Equation and ADI

An important subclass of Sylvester’s equations is the Lyapunov equation

\[ A\mathcal{P} + \mathcal{P}A^* + BB^* = 0. \tag{1.2} \]

The Bartels-Stewart algorithm may be applied to this equation but this algorithm does not preserve the structure of the solution, so the solution \( \mathcal{P} \) computed by Bartels-Stewart algorithm might not be Hermitian positive-semidefinite, a property the solution must obey.

For the rest of this section, we assume that \( A \) is stable. We shall develop a method, originally due to Hammarling [?], that uses ideas similar to those of the Bartels-Stewart algorithm for the Sylvester equation, but with a key distinction: Hammarling’s algorithm constructs the solution in the factored form \( \mathcal{P} = UU^* \), thus ensuring by construction that the solution is Hermitian positive-definite. (For any \( x \in \mathbb{C}^n \), \( x^*\mathcal{P}x = x^*UU^*x = \|U^*x\|^2 \).) As
with the dense Sylvester algorithm, there is an initial transformation of the
original problem to triangular form using the Schur decomposition $A = QRQ^*$ (with $Q^*Q = I$ and $R$ upper triangular with the diagonal elements
of $R$ holding the eigenvalues of $A$). Keep in mind that $Q$ and $R$ usually
take complex values, even when $A$ is real.\(^1\)

Equation (1.2) can be pre- and post-multiplied by $Q^*$ and $Q$ to obtain
the new Lyapunov equation

$$(Q^*AQ)(Q^*PQ) + (Q^*PQ)(Q^*AQ)^* + (Q^*B)(Q^*B)^* = 0,$$

which involves the upper triangular coefficient matrix

$$R := Q^*AQ.$$ 

Since $P$ is Hermitian positive semi-definite, we shall seek solutions in the
Cholesky factored form:

$$Q^*PQ =: UU^*.$$ 

For simplicity of notation, we shall replace the original matrix $B \in \mathbb{C}^{n \times m}$
in (1.2) with the transformed matrix:

$$B \leftarrow Q^*B \in \mathbb{C}^{n \times m}.$$ 

With this notation, problem (1.2) becomes

$$R(UU^*) + (UU^*)R^* + BB^* = 0; \quad (1.3)$$

Note that $U$ is upper triangular with nonnegative diagonal elements. As in
the Bartels–Stewart algorithm, we shall compute the solution one column
at a time, from right to left. Toward this end, partition the matrices $R,$ $U,$
and $B$ into

$$R = \begin{bmatrix} \hat{R} & r \\ 0 & \rho \end{bmatrix}, \quad U = \begin{bmatrix} \hat{U} & u \\ 0 & \mu \end{bmatrix}, \quad B = \begin{bmatrix} \hat{B} \\ b^* \end{bmatrix}.$$ 

An equation for the last column of $U$ may be obtained via

$$[R(UU^*) + (UU^*)R^* + BB^*]e_n = 0, \quad (1.4)$$

where $e_n$ is the $n$th column of the identity matrix. Since $\mu$ is a positive real
number,

$$(RUU^*)e_n = \begin{bmatrix} \hat{R} & r \\ 0 & \rho \end{bmatrix} \begin{bmatrix} u \\ \mu \end{bmatrix} \mu,$$

\(^1\)To compute the complex-valued Schur decomposition $A = QRQ^*$ in MATLAB, use
$[Q,R] = \text{schur}(A,'complex').$
1.8. The Lyapunov Equation and ADI

while
\[(UU^*)e_n = \begin{bmatrix} u \\ \mu \end{bmatrix} \mu \rho \quad \text{and} \quad (BB^*)e_n = \begin{bmatrix} \hat{B}b \\ \|b\|^2 \end{bmatrix}. \quad (1.5)\]

Thus, equation (1.4) can be written as
\[
\begin{bmatrix} \hat{R} + \rho I & r \\ 0 & \rho + \bar{\rho} \end{bmatrix} \begin{bmatrix} u \\ \mu \end{bmatrix} \mu = -\begin{bmatrix} \hat{B}b \\ \|b\|^2 \end{bmatrix}.
\]

The last equation gives
\[
\mu = \frac{\|b\|}{\sqrt{\rho + \bar{\rho}}} = \frac{\|b\|}{\sqrt{2|\text{Re} \rho|}}.
\]

If the system is stable and controllable, then Re \(\rho < 0\) and \(\|b\| > 0\). (If \(b = 0\) the system is not controllable; the algorithm can continue by setting \(u = 0\) and \(\mu = 0\). Notice that \(e_n\) is a left eigenvector of any upper triangular matrix, \(e_n^*R = \rho e_n^*\); if \(b = 0\), then equation (1.5) shows that this left eigenvector is orthogonal to all columns of \(B\). By condition 5 of Theorem ?? on page ??, the system is not controllable.)

Now that \(u\) and \(\mu\) are known, the leading \(n - 1\) equations become
\[
(\hat{R} + \rho I)u = -(\hat{B}b/\mu + r \mu).
\]

Note that \((\hat{R} + \rho I)\) is nonsingular since Re \((\lambda + \rho) < 0\) for all \(\lambda \in \sigma(A)\).

To continue the algorithm, it is necessary to apply a nonsingular transformation in order to arrive and an \(n - 1\) dimensional problem of the same form as the original one. Toward this end, put
\[
M = \begin{bmatrix} I & -u/\mu \\ 0 & 1 \end{bmatrix},
\]
so that
\[
MU = \begin{bmatrix} I & -u/\mu \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{U} & u \\ 0 & \mu \end{bmatrix} = \begin{bmatrix} \hat{U} & 0 \\ 0 & \mu \end{bmatrix},
\]
and
\[
MR = \begin{bmatrix} I & -u/\mu \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{R} & r \\ 0 & \rho \end{bmatrix} = \begin{bmatrix} \hat{R} & z \\ 0 & \rho \end{bmatrix},
\]
with \(z := r - u(\rho/\mu)\). Apply \(M\) as a congruence transformation to equation (1.3) to obtain
\[
0 = M [R(UU^*) + (UU^*)R^* + BB^*] M^* = (MR)U(MU)^* + (MU)U^*(MR)^* + (MB)(MB)^*.
\]
Chapter 1. Basic Iterative Methods

The leading $(n - 1) \times (n - 1)$ submatrix of the product

$$(MR)U(MU)^* = \begin{bmatrix} \hat{R} & z \\ 0 & \rho \end{bmatrix} \begin{bmatrix} \hat{U} \\ 0 \end{bmatrix} \begin{bmatrix} \hat{U}^* & 0 \end{bmatrix}$$

is $\hat{R}(\hat{U}\hat{U}^*)$ and the leading $(n - 1) \times m$ block of $MB$ is $\hat{B} := \hat{B} - ub^*/\mu$. Hence the leading $(n - 1) \times (n - 1)$ portion of the equation (1.6) yields a new Lyapunov equation of order $n - 1$:

$$0 = \hat{R}(\hat{U}\hat{U}^*) + (\hat{U}\hat{U}^*)\hat{R}^* + \hat{B}\hat{B}^*.$$

Apply the above procedure to this new equation to uncover the last column of $\hat{U}$, hence the penultimate column of $U$ in the original equation (1.2). Continue this process to find all the columns of $U$, from right to left. Once the entire $U$ matrix has been found, compute the solution $P$ to equation (1.2) as

$$P = (QU)(QU)^*.$$

This upper triangular Lyapunov solver may be implemented as shown in Figure 1.1. There is a corresponding algorithm for the observability Gramian in factored form. Under the transformation

$$R = Q^*AQ, \quad C \leftarrow CQ, \quad Q^*Q = LL^*,$$

problem (??) becomes

$$R^*(LL^*) + (LL^*)R + C^*C = 0, \quad (1.7)$$

where $L$ is lower triangular with positive diagonal elements. Essentially the same derivation may be made for computing $L$ directly in lower triangular form by beginning with the first rather than the last column: we leave the details for an exercise.

1.9 Iterative solution methods

In the common case where $A$ is a sparse matrix of large dimension, the cost of computer arithmetic and storage prohibits the use of the dense matrix factorizations so fundamental to the algorithms described in the last section. Iterative methods provide an alternative means to the solution. Given the discussion in Section 1.8, one could apply standard Krylov subspace iterative methods for the solution of linear systems (see, e.g., [2]) to the order $n^2$ standard linear system (??) equivalent to the Lyapunov equation. However,
1.9. Iterative solution methods

\[ U = \text{LyapU}(R, B) \]

**Input:**  
- \( R \) an \( n \times n \) upper triangular matrix whose diagonal entries have negative real part  
- \( B \) an \( n \times p \) matrix

**Output:**  
- \( U \) an \( n \times n \) upper triangular matrix such that  
  \[ R(UU^*) + (UU^*)R^* + BB^* = 0 \]

\[
\begin{align*}
U &\leftarrow 0 \\
\text{for } j &= n, n-1, \ldots, 2 \\
\rho &\leftarrow R(j,j) \\
b &\leftarrow B(j,:) \\
\mu &\leftarrow \|b\|/\sqrt{2|\text{Re}\rho|} \\
\text{if } \mu &> 0 \\
c &\leftarrow B(1:j-1,:)b^*/\mu + R(1:j-1,j)\mu \\
\text{Solve } (R(1:j-1,1:j-1) + \rho I)u = -c \text{ for } u \\
U(j,j) &\leftarrow \mu, \quad U(1:j-1,j) \leftarrow u \\
B(1:j-1,:) &\leftarrow B(1:j-1,:) - ub\mu \\
\text{else} \\
U(1:j-1,j) &\leftarrow 0 \\
\end{align*}
\]

\[ U(1,1) \leftarrow \|B(1,:)/\sqrt{2|\text{Re}R(1,1)|} \]

---

Figure 1.1. Algorithm for solving the Lyapunov equation with an upper triangular coefficient matrix. To solve the standard Lyapunov equation (1.2), compute a (complex) Schur factorization \( A = QRQ^* \) to obtain \( R \) and update \( B \leftarrow Q^*B \). The solution of (1.2) is then \( P = (QU)(QU)^* \).

---

this would have many drawbacks. Such an approach would be unwieldy due to sheer size; perhaps more important, there would be a complete loss of structure in the equations and solutions. Here we develop a far more effective approach, originally due to Smith [?], that can be viewed as an application of the famous ADI (Alternating Directions Implicit) method of Peaceman and Rachford for the solution of partial differential equations in oil reservoir modeling [?]; see [?, ?] for details about this connection.

D.C. Sorensen – draft – 21 January 2015
1.9.1 Smith–ADI for Lyapunov equations

The Smith–ADI method begins with the selection of a shift $\mu \in \mathbb{C}$ whose value, as we shall see, can be tuned to optimize convergence. Start with the original Lyapunov equation,

$$A\mathcal{P} + \mathcal{P}A^* + BB^* = 0,$$  \hfill (1.8)

and assume for now that $\mu > 0$ is real. Equation (1.8) is equivalent to

$$(A - \mu I)\mathcal{P} = -[\mathcal{P}(A + \mu I)^* + BB^*],$$

which gives an expression for $\mathcal{P}$ with the shift applied via inversion on the left-hand side:

$$\mathcal{P} = -(A - \mu I)^{-1}[\mathcal{P}(A + \mu I)^* + BB^*].$$ \hfill (1.9)

Similarly, we can manipulate (1.8) to apply the same shift $\mu$ via inversion from right-hand side, the *alternate direction*. This yields

$$\mathcal{P} = -[(A + \mu I)\mathcal{P} + BB^*](A - \mu I)^{-1}.$$ \hfill (1.10)

Now substitute the formula (1.9) for $\mathcal{P}$ on the right-hand side of (1.10) to obtain a *Stein equation*

$$\mathcal{P} = A_\mu \mathcal{P}A^*_\mu + B_\mu B^*_\mu,$$ \hfill (1.11)

where

$$A_\mu := (A - \mu I)^{-1}(A + \mu I), \quad B_\mu := \sqrt{2|\mu|}(A - \mu I)^{-1}B.$$

(Note that (1.11) is known as a *discrete time Lyapunov equation*, arising in the control of discrete-time dynamical systems.) If we permit a complex shift, $\mu \in \mathbb{C}$, having real part $\Re \mu =: \rho > 0$, the Stein equation (1.11) still holds, provided we generalize the definitions of $A_\mu$ and $B_\mu$ to

$$A_\mu := (A - \bar{\rho}I)^{-1}(A + \mu I), \quad B_\mu := \sqrt{2\rho}(A - \bar{\rho}I)^{-1}B.$$

The form of the Stein equation (1.11) immediately suggests a method for its solution: attempt *direct iteration*. Starting from $\mathcal{P}_{-1} = 0$, use the right-hand side of (1.11) to produce a refined estimate for the left-hand side:

$$\mathcal{P}_j := A_\mu \mathcal{P}_{j-1}A^*_\mu + B_\mu B^*_\mu.$$ \hfill (1.12)
for \( j = 0, 1, \ldots \). At the \( k \)th iteration this recurrence produces the partial sum
\[
P_k = \sum_{j=0}^{k} A_j^\mu B_\mu B^*_\mu (A_j^\mu)^*.
\]
This method of constructing approximate solutions is called the Smith–ADI method. Of course, one does not explicitly form 
\( A_\mu = (A - \mu I)^{-1}(A + \mu I) \), but instead would compute an LU factorization of \( A - \mu I \) that will expedite the solution of linear systems at each iteration. For large and sparse \( A \), sparse direct methods should be used. (Further implementation details will be addressed after we make a few refinements to this algorithm.)

When the Smith–ADI method converges as \( k \to \infty \), the recursion (1.12) suggests that the solution to (1.11) can be written as
\[
P = \sum_{j=0}^{\infty} A_j^\mu B_\mu B^*_\mu (A_j^\mu)^*.
\]
Indeed, one can verify that, formally, this infinite series satisfies the Stein equation (1.11). Under what circumstances does this series form converge?

### 1.9.2 Convergence of the Smith-ADI method

To analyze the quality of the Smith–ADI iterates, define the error
\[
E_k := P - P_{k-1}.
\]
Subtract (1.12) from the Stein equation (1.11) to obtain the formula
\[
E_k = A_\mu E_{k-1} A_\mu^* = \cdots = A_\mu^k E_0 (A_\mu^k)^* = A_\mu^k P (A_\mu^k)^*.
\]
Convergence thus depends on the behavior of \( A_\mu^k \) as \( k \to \infty \), and hence, on the spectrum of \( A_\mu \). The eigenvalues of \( A_\mu \) can be readily obtained from the eigenvalues of \( A \): an eigenvalue \( \lambda \in \sigma(A) \) is transformed to an eigenvalue
\[
\theta(\lambda) = \frac{\lambda + \mu}{\lambda - \mu} \in \sigma(A_\mu)
\]
of the transformed matrix \( A_\mu \). This linear fractional transformation, illustrated in Figure 1.2, maps the imaginary axis to the unit circle and \( \theta(-\mu) = 0 \), so that the entire open left half-plane is mapped strictly inside the unit circle. Since \( A \) is stable, the eigenvalues of \( A_\mu \) must thus all lie strictly inside the unit disk. This fact implies that the spectral radius
\[ \rho(A_\mu) < 1, \text{ so } A^k_\mu \to 0. \text{ It follows that } E_k \to 0 \text{ and } P_k \to P, \text{ thus justifying formula (1.13) for } P. \]

Figure 1.3 shows the spectra of \(A\) and \(A_\mu\) for a particular model problem. As \(A\) has many eigenvalues with large imaginary part, it is evident that from the map (1.16) that eigenvalues of \(A_\mu\) will be close to one for any choice of \(\mu\). Indeed, the choice \(\mu = 400\) used in the figure gives \(A_\mu\) with spectral radius around 0.99988. (This is not far from the optimal value over all real choices of \(\mu\).)

### 1.9.3 Shift selection, multiple shifts

How should one best choose a shift to optimize convergence? Suppose \(A\) is a diagonalizable matrix, \(A = VAV^{-1}\), so that

\[ A_\mu = V\Lambda_\mu V^{-1}, \]

where \(\Lambda_\mu := (\Lambda - \mu I)^{-1}(\Lambda + \mu I)\) is a diagonal matrix whose \((j,j)\) entry is simply

\[ (\Lambda_\mu)_{j,j} = \frac{\lambda_j + \mu}{\lambda_j - \mu}. \]

The coarsest bound on the error (1.15) gives

\[
\frac{\|E_k\|}{\|P\|} \leq \|A_\mu^k\| \|\Lambda_\mu^k\| \|(\Lambda_\mu^k)^*\| \\
\leq \|V\|^2 \|V^{-1}\|^2 \left( \max_{\lambda \in \sigma(A)} \frac{|\lambda + \mu|}{|\lambda - \mu|} \right)^k.
\]

Figure 1.2. Example of the linear fractional transformation (1.16) for parameter \(\mu = 0.25\) \((-\mu\) is marked by \(\ast\)): on the left, five vertical lines in the \(\lambda\) domain; on the right, the images \(\theta(\lambda)\) of these vertical lines form circles all passing through 1. For any real \(\mu\), the imaginary axis is mapped to the unit circle; the image of \(-\mu\) is \(\theta(-\mu) = 0\).
1.9. Iterative solution methods

Figure 1.3. Spectrum of $A$ for the CD player example with $-\mu$ marked by +, for $\mu = 400$ (left); spectrum of $A_{\mu}$ with $\theta(-\mu) = 0$ marked by + (right). All eigenvalues of $A_{\mu}$ are in the unit disk, but many have magnitude very close to one, implying slow convergence for the Smith–ADI algorithm.

The nature of the linear fractional transformation ensures that $|\lambda + \mu|/|\lambda - \mu| < 1$, so we have convergence. How should one pick $\mu$ to make this convergence as fast as possible?

We can answer this question in a few special cases, if we first bound the spectrum by a larger set in the complex plane. First, suppose the eigenvalues are all real (for example, when $A$ is Hermitian): $\sigma(A) \in [\alpha, \beta]$ for $\alpha < \beta < 0$.

A simple calculus exercise shows that the $\mu$ that

$$
\min_{\mu > 0} \max_{\lambda \in [\alpha, \beta]} \frac{|\lambda + \mu|}{|\lambda - \mu|} = \frac{-\mu_* - \alpha}{\mu_* - \alpha} = \frac{\mu_* + \beta}{\mu_* - \beta}, \quad \mu_* := \sqrt{\alpha \beta}.
$$

For non-Hermitian $A$, it may be possible to bound the spectrum of $A$ in a disk $D_{c,r} := \{z \in \mathbb{C} : |z - c| \leq r\}$ for some $r < |c|$ (so that the disk is contained in the left half-plane). The optimal shift is then given by

$$
\min_{\mu > 0} \max_{\lambda \in D} \frac{|\lambda + \mu|}{|\lambda - \mu|} = \frac{|c + re^{i\theta} + \mu_*|}{|c + re^{i\theta} - \mu_*|} \quad \text{for all } \theta,
$$

where (see, for example, [?, ?])

$$
\mu_* := \sqrt{(c + r)(c - r)}.
$$

Note that this is not generally the center of the containment disk!
2. Theory of Eigenvalue Decompositions

Understanding the convergence behavior of iterative methods relies heavily on complete understanding of the eigen-structure of the iteration matrix $A$. Where the right hand side $b$ sits with respect to this structure can influence convergence greatly but in general it is the eigen-structure of $A$ and the resulting iteration matrix $G$ that determines convergence. For Hermitian and more generally for normal matrices $G$, convergence is determined by the spectral radius $\rho(G)$. However, as will be seen, the convergence history of an iteration with a non-normal $G$ typically has a transient phase dictated by aspects of non-normality followed by an asymptotic convergence phase that is governed by the spectral radius.

2.1 Eigenvalues and Eigen-Decompositions

Let $A \in \mathbb{C}^{n \times n}$. The set of numbers $\sigma(A) \equiv \{ \lambda \in \mathbb{C} : \text{rank}(A - \lambda I) < n\}$ is called the spectrum of $A$. The elements of this discrete set are the eigenvalues of $A$. It is easily shown that

$$p(\lambda) \equiv \det(\lambda I - A).$$

is a polynomial of degree $n$ and that the spectrum of $A$ is the set of roots of this characteristic polynomial,

$$\sigma(A) \equiv \{ \lambda : \det(\lambda I - A) = 0 \}.$$

Thus, $A$ has precisely $n$ eigenvalues counting multiplicities. From the definition, an eigenvalue is a number $\lambda$ such that the matrix $\lambda I - A$ is singular.
and hence, for each \( \lambda \in \sigma(A) \), there is a non-zero vector \( x \) such that

\[
(\lambda I - A)x = 0,
\]

which implies

\[
Ax = x\lambda.
\]

The vector \( x \) is called a (right) eigenvector of \( A \) corresponding to the eigenvalue \( \lambda \), and \((x, \lambda)\) is an eigenpair. The multiplicity \( n_a(\lambda) \) considered as a root of the characteristic polynomial is called the algebraic multiplicity of \( \lambda \). Define

\[
S_\lambda = \{ x : (\lambda I - A)x = 0 \} = \text{null}(\lambda I - A).
\]

\( S_\lambda \) is called the (right) eigenspace of \( A \) corresponding to \( \lambda \). We denote the dimension of \( S_\lambda \) by \( n_g(\lambda) \equiv \text{dim}(S_\lambda) \), and this number is called the geometric multiplicity of \( \lambda \). Definition: If \( n_a(\lambda) < n_g(\lambda) \) then \( A \) is called a defective matrix. If \( n_a(\lambda) = 1 \) then \( \lambda \) is simple.

### 2.1.1 Exercises

1. Show that \( \det(A) = \lambda_1\lambda_2 \cdots \lambda_n \) where \( \{\lambda_1, \lambda_2, \cdots, \lambda_n\} = \sigma(A) \).

2. Show that \( \text{trace}(A) = \lambda_1 + \lambda_2 + \cdots + \lambda_n \) where \( \text{trace}(A) \equiv \alpha_{1,1} + \alpha_{2,2} + \cdots + \alpha_{n,n} \).

3. Consider the matrix

\[
A = \begin{bmatrix}
\mu & 1 & 0 \\
0 & \mu & 1 \\
0 & 0 & \mu \\
\end{bmatrix}.
\]

Show that \( n_a(\mu) = 3 \) and \( n_g(\mu) = 1 \). Give a basis for \( S_\mu \)?

4. Generalize (3) to a matrix with this structure of arbitrary order.

5. Prove for \( A \in \mathbb{C}^{n \times n} \) that \( n_g(\lambda) \leq n_a(\lambda) \) for all \( \lambda \in \sigma(A) \).

6. Show that the diagonal elements of a triangular matrix are its eigenvalues.

D.C. Sorensen – draft – 21 January 2015
7. Let the matrix

\[
T = \begin{bmatrix}
  T_{1,1} & T_{1,2} & \cdots & T_{1,m-1} & T_{1,m} \\
  0 & T_{2,2} & & T_{2,m} \\
  \vdots & \ddots & \ddots & \ddots & \vdots \\
  0 & \cdots & 0 & T_{m-1,m-1} & T_{m-1,m} \\
  0 & \cdots & 0 & 0 & T_{m,m}
\end{bmatrix}
\]

be block upper triangular with diagonal blocks \( T_{j,j} \). Prove that \( \sigma(T) = \bigcup_{j=1}^{m} \sigma(T_{j,j}) \).

8. Prove that for every \( \lambda \in \sigma(A) \) there is a vector \( y \in \mathbb{C}^{n \times n} \) such that \( y^*(\lambda I - A) = 0 \). Such a vector is called a left eigenvector of \( A \) corresponding to \( \lambda \). Construct a left eigenvector corresponding to \( \mu \) for the matrix in exercise (3). What is the relationship of this vector to the right eigenvector? Develop all of the notions concerning multiplicity, eigenspaces etc. for left eigenvectors. Is the geometric multiplicity with respect to left eigenvectors the same as for right eigenvectors corresponding to the same eigenvalue?

### 2.2 Schur Decomposition

The central theorem in the study of numerical algorithms for eigenproblems is the Schur decomposition. It states that there is a unitary basis in which the representation of a matrix \( A \in \mathbb{C}^{n \times n} \) is upper triangular.

**Theorem 2.1. (Schur Decomposition).** Let \( A \in \mathbb{C}^{n \times n} \). Then there is a unitary matrix \( Q \) and an upper triangular matrix \( R \) such that

\[
AQ = QR. \tag{2.1}
\]

The diagonal elements of \( R \) are the eigenvalues of \( A \).

**Proof.** We shall prove (2.1) by induction on \( n \). The remainder of the proof is left as an exercise. The result is clearly true for matrices of order 1. Assume that (2.1) is true for matrices of order less than \( n \). Let \( \lambda \) be any eigenvalue of \( A \) and \( q \) a corresponding normalized eigenvector so that \( Aq = q\lambda \) with \( \|q\| = 1 \). Construct a unitary matrix \( Q_1 \) such that \( Q_1q = e_1 \) (e.g. via Householder transformation, see exercise ??) Then \( Q_1 = (q, \tilde{Q}_1) \) with \( \tilde{Q}_1^*q = 0 \). Now

\[
AQ_1 = (Aq, A\tilde{Q}_1) = (q\lambda, A\tilde{Q}_1)
\]

D.C. Sorensen – draft – 21 January 2015
22 Chapter 2. Theory of Eigenvalue Decompositions

and thus

\[ Q_1^* A Q_1 = \begin{bmatrix} \lambda & q^* A Q_1 \\ 0 & Q_1^* A Q_1 \end{bmatrix} \equiv \begin{bmatrix} \lambda & r_1^* \\ 0 & A \end{bmatrix}, \]

with \( \hat{A} \in \mathbb{C}^{(n-1)\times(n-1)} \). By the induction hypothesis there exist an unitary matrix \( \hat{Q} \) and an upper triangular matrix \( \hat{R} \) such that \( \hat{A} \hat{Q} = \hat{Q} \hat{R} \). Thus

\[ Q_1^* A Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q} \end{bmatrix} \begin{bmatrix} \lambda & r^* \\ 0 & \hat{R} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}^* \end{bmatrix} \equiv Q_2 R Q_2^*, \]

where \( r^* = r_1^* \hat{Q} \) and

\[ Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q} \end{bmatrix}. \]

Hence \( A = Q R \) where \( Q = Q_1 Q_2 \) and the induction is complete. \( \square \)

The Schur decomposition provides a means to develop additional fundamental decompositions. The fundamental structure of Hermitian and Normal matrices follows readily from the Schur decomposition and is expressed in the following two Lemmas.

Lemma 2.2. A matrix \( A \in \mathbb{C}^{n\times n} \) is normal ( \( AA^* = A^* A \) ) if and only if \( A = Q \Lambda Q^* \) with \( Q \in \mathbb{C}^{n\times n} \) unitary and \( \Lambda \in \mathbb{C}^{n\times n} \) diagonal. The diagonal entries of \( \Lambda \) are the eigenvalues of \( A \) and the columns of \( Q \) are the corresponding eigenvectors.

Proof. From the Schur decomposition \( A = Q R \) it follows that \( A \) is normal if and only if

\[ RR^* = R^* R, \]

for if this property holds it is easily shown that \( A \) is normal and if \( A \) is normal then \( RR^* = Q^* AA^* Q = Q^* A^* AQ = Q^* A^* QA = R^* R \). However, an upper triangular matrix is normal if and only if it is diagonal (see exercise (2) below). \( \square \)

Lemma 2.3. A matrix \( A \in \mathbb{C}^{n\times n} \) is Hermitian ( \( A = A^* \) ) if and only if \( A = Q \Lambda Q^* \) with \( Q \in \mathbb{C}^{n\times n} \) unitary and \( \Lambda \in \mathbb{R}^{n\times n} \) diagonal.

Proof. If \( A = Q \Lambda Q^* \) as in the hypothesis then it is clearly Hermitian. On the other hand, a Hermitian matrix is necessarily normal and is therefore
unitarily similar to a diagonal matrix Λ from the Lemma (1.2). Therefore, this diagonal matrix must also be normal and hence real. (see exercise (3) below) □

This lemma indicates that the eigenvalues of a Hermitian matrix are real and the eigenvectors are mutually orthogonal. The eigensystems of Hermitian matrices possess a number of important properties that greatly simplify computational techniques for approximating them. These will be developed later.

2.2.1 Exercises

1. Complete the proof of the Schur Decomposition. Use determinants to show that the diagonal elements of R are the eigenvalues of A.

2. Prove that an upper triangular matrix R is normal if and only if it is diagonal.

3. Show directly from the Schur decomposition that if A is Hermitian then R is also Hermitian and thus a real diagonal matrix. If A is real and symmetric show that the eigenvectors are real and orthogonal.

4. Is a Schur decomposition unique in any sense? Show that for a given matrix A there is a Schur decomposition A = QRQ* with the eigenvalues of A appearing on the diagonal of R in any specified order.

5. Suppose A, B, X ∈ C^{n×n} satisfy AX = XB with X nonsingular. Prove (using eigenvectors) that σ(A) = σ(B).

6. If there is a nonsingular X such that AX = XB then A is said to be similar to B, i.e. A ∼ B. Show that

A ∼ A, A ∼ B ⇒ B ∼ A, A ∼ B and B ∼ C ⇒ A ∼ C.

7. Give an example where σ(A) = σ(B) but A is not similar to B. What is the weakest additional hypothesis that will imply A ∼ B?

8. Give an algorithm for computing the eigenvectors of an upper triangular matrix R which has distinct eigenvalues. Discuss the sensitivity of these eigenvectors to perturbations when two or more eigenvalues of R coalesce. Hint: Partition

\[
R = \begin{bmatrix}
\hat{R} & r \\
0 & \rho
\end{bmatrix},
\]
and \( x^T = [\hat{x}, 1]^T \) and solve \( (R - \rho I)\hat{x} = -r \).

9. Prove that if the eigenvalues of \( A \in \mathbb{C}^{n \times n} \) are distinct then \( A \) is diagonalizable, i.e. there exist a nonsingular matrix 
\[
X = [x_1, \cdots, x_n],
\]
such that \( AX = X\Lambda \) where \( \Lambda \) is a diagonal matrix of the eigenvalues of \( A \) and \( Xe_j \) is the corresponding eigenvector for eigenvalue \( \lambda_j \) the \( j \)-th diagonal element of \( \Lambda \).

2.3 Eigen-decompositions of \( \mathbb{C}^n \)

We have shown that a normal matrix \( A \) has a decomposition 
\[
A = Q\Lambda Q^* , \quad Q^* Q = I , \quad \Lambda \text{ is a diagonal matrix}.
\]
Normality of \( A \) is a necessary and sufficient condition for such a decomposition. Now suppose that a normal matrix \( A \) has \( k \) distinct eigenvalues \( \lambda_1, \lambda_2, \cdots, \lambda_k \) where \( \lambda_j \) has multiplicity \( n_j \). By reordering eigenvalues and vectors we can arrange this decomposition in the form
\[
A = \sum_{j=1}^{k} \lambda_j Q_j Q_j^* ,
\]
where 
\[
Q = (Q_1, Q_2, \ldots, Q_k) , \quad Q_i^* Q_j = 0 , \quad i \neq j , \quad Q_j^* Q_j = I_{n_j} , \quad \sum_{j=1}^{k} n_j = n.
\]

2.3.1 Exercise

1. Recall that \( S_{\lambda_j} = \{ x : (\lambda_j I - A)x = 0 \} \). Show that \( P_j = Q_j Q_j^* \) is an orthogonal projector onto \( S_{\lambda_j} \) and that \( S_{\lambda_j} = \text{range}(Q_j) \).

2. Show that \( \sum_{j=1}^{k} P_j = \sum_{j=1}^{k} Q_j Q_j^* = QQ^* = I_n \) so that any \( b \in \mathbb{C}^n \) may be written uniquely as \( b = \sum_{j=1}^{k} Q_j Q_j^* b = \sum_{j=1}^{k} q_j \) where \( q_j = Q_j Q_j^* b \in S_{\lambda_j} \). Conclude that \( \mathbb{C}^n = S_{\lambda_1} \oplus S_{\lambda_2} \oplus \cdots \oplus S_{\lambda_k} \) is the direct sum of the subspaces \( S_{\lambda_j} \).

3. Show that if \( s \in S_{\lambda_j} \) then \( A = s \lambda_j \). Conclude that \( AS_{\lambda_j} \subset S_{\lambda_j} \).

D.C. Sorensen  
– draft –  
21 January 2015
2.4 Invariant Subspaces

We have seen in the case of normal matrices that the space $\mathbb{C}^n$ is decomposed into the direct sum of the eigen-spaces $S_{\lambda_j}$ of the matrix $A$, and that these subspaces are mutually orthogonal to one another. As we shall see, this property has important consequences numerically since it limits the effects of perturbations (in particular perturbations due to round off).

It is natural therefore to try to understand as best we can the structure of the subspaces $S_{\lambda}$ of general matrices. An understanding of this structure is important to the development of the algorithms for finding a few selected eigenvalues and vectors of a general matrix. As we shall see, the general case is far more complicated than the normal case. Eigenspaces are special cases of a more general class of subspaces associated with a matrix.

**Definition:** A subspace $S$ of $\mathbb{C}^{n \times n}$ is called an invariant subspace of $A \in \mathbb{C}^{n \times n}$ if $AS \subset S$.

It is straightforward to show

**Lemma 2.4.** If $A \in \mathbb{C}^{n \times n}$, $X \in \mathbb{C}^{n \times k}$ and $B \in \mathbb{C}^{k \times k}$ satisfy

$$AX = XB,$$  \hspace{1cm} (2.2)

then $S \equiv \text{Range}(X)$ is an invariant subspace of $A$. Moreover, if $X$ has column rank $k$ then the columns of $X$ form a basis for this subspace and $\sigma(B) \subset \sigma(A)$.

**Proof.** The fact that $S$ is invariant follows immediately from (2.2) and the assumption of full rank assures $X$ forms a basis. Suppose $By = y\lambda$ is an eigenpair for $B$. Then

$$AXy = XBy = Xy\lambda$$

i.e. $Ax = x\lambda$ with $x = Xy$ so $\sigma(B) \subset \sigma(A)$. \hfill $\square$

2.4.1 Exercises

1. Why is $x \neq 0$ in the proof of Lemma (1.4) ?

2. Show that $AX = XB$ implies that $\psi(A)X = X\psi(B)$ for any polynomial $\psi$. Recall that if $\psi(\lambda) = \gamma_0 + \gamma_1\lambda + \cdots + \gamma_k\lambda^k$ then $\psi(A) = \gamma_0I + \gamma_1A + \cdots + \gamma_kA^k$.
In many cases it is desirable to have a unitary basis rather than an arbitrary one. It is straightforward to obtain a unitary basis from any given basis by obtaining a $QR$ factorization. We develop this formally in the following lemma.

**Lemma 2.5.** Assume the hypothesis of Lemma (1.4). If $X = QR$ with $Q^*Q = I_k$ and $R$ an upper triangular matrix then the columns of $Q$ from an orthonormal basis for $S$.

**Proof.** If $AX = XB, \ X = QR, \ Q^*Q = I_k$ then $X$ having rank $k$ implies that $R$ is non-singular. Hence

$$AQR = QRB,$$

so

$$AQ = Q(RBR^{-1}) \equiv QB.$$

Note that $\sigma(B) = \sigma(A) \subset \sigma(A)$ and $z \in S$ implies $z = Xw$ for some vector $w \in \mathbb{C}^k$ and $Q(Rw) \equiv Q\hat{w}$. So the columns of $Q$ are an orthonormal basis for $S$. $\square$

Examination of the proof of the Schur decomposition indicates without loss of generality that a matrix $A \in \mathbb{C}^{n \times n}$ may be decomposed into

$$AQ = QR, \ Q^*Q = I, \ R \text{ is upper triangular}, \quad (2.3)$$

with eigenvalues of $A$ appearing in any order as diagonal elements of $R$ (see exercise 1.6.4). Thus it is possible to obtain $R$ in the form

$$R = \begin{bmatrix}
R_{1,1} & R_{1,2} & \cdots & R_{1,k-1} & R_{1,k} \\
0 & R_{2,2} & & R_{2,k} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & & R_{k-1,k-1} & R_{k-1,k} \\
0 & 0 & \cdots & 0 & R_{k,k}
\end{bmatrix}, \quad (2.4)$$

where $\sigma(R_{jj}) = \{\lambda_j\}$ where $\lambda_j \in \sigma(A)$ for $1 \leq j \leq k$ are the distinct eigenvalues of $A$.

The following lemma shows that it is possible to completely decouple the diagonal blocks of $R$ at the expense of losing a unitary basis.
Lemma 2.6. Suppose
\[ R = \begin{bmatrix} R_1 & S \\ 0 & R_2 \end{bmatrix}, \]
with \( \sigma(R_1) \cap \sigma(R_2) = \emptyset \). Then
\[ R \sim \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix}. \]

Proof. We shall construct a similarity transformation. Let \( R_1 \in \mathbb{C}^{k \times k} \), \( R_2 \in \mathbb{C}^{n-k \times n-k} \), and \( S \) and \( Z \in \mathbb{C}^{k \times n-k} \) where \( 1 \leq k < n \). Note that
\[ \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \begin{bmatrix} R_1 & S \\ 0 & R_2 \end{bmatrix} = \begin{bmatrix} R_1 & 0 \\ 0 & R_2 \end{bmatrix} \begin{bmatrix} I & Z \\ 0 & I \end{bmatrix} \]
if and only if \( S + ZR_2 = R_1X \) has a unique solution \( Z \), i.e.
\[ R_1Z - ZR_2 = S. \]
This system of equations may be solved by noting that
\[ R_1Ze_j - ZR_2e_j = Se_j, \]
i.e.
\[ R_1z_j - \rho_{j,j}z_j - \left( \sum_{i=1}^{j-1} z_i\rho_{i,j} \right) = s_j, \]
where \( R_2 = (\rho_{i,j}), z_j = Ze_j, s_j = Se_j \).
Hence the columns \( z_j \) may be determined in the order \( j = 1, 2, \ldots, k \) by successively solving the triangular systems
\[ (R_1 - \rho_{j,j}I)z_j = s_j + \sum_{i=1}^{j-1} z_i\rho_{i,j}. \] (2.5)
Each of these equations has a unique solution since \( \sigma(R_1) \cap \sigma(R_2) = \emptyset \) implies that \( R_1 - \rho_{j,j}I \) is nonsingular for \( 1 \leq j \leq k \). \( \square \)
2.4.2 Exercise

1. Apply Lemma (1.6) repeatedly to \( R \) in (2.4) to obtain a nonsingular \( \hat{X} \) such that \( RX = \hat{X} \hat{R} \) where

\[
\hat{R} = \begin{bmatrix}
R_{1,1} & 0 & \cdots & 0 & 0 \\
0 & R_{2,2} & & & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & & \ddots & \ddots & 0 \\
0 & 0 & \cdots & R_{k-1,k-1} & 0 \\
0 & 0 & \cdots & 0 & R_{k,k}
\end{bmatrix},
\]

is block diagonal.

Now we have from (2.3) that

\[
AQ\hat{X} = QR\hat{X} = Q\hat{X}\hat{R}
\]

and we may express this as \( AX = X\hat{R} \) where \( X = Q\hat{X} \). Partitioning \( X = [X_1, X_2, \ldots, X_k] \) according to the block partitioning of \( R \) gives

\[
A[X_1, X_2, \ldots, X_k] = [X_1, X_2, \ldots, X_k]
\]

\[
\begin{bmatrix}
R_{1,1} & 0 & \cdots & 0 \\
0 & R_{2,2} & & & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & & \ddots & \ddots & 0 \\
0 & 0 & \cdots & R_{k-1,k-1} & 0 \\
0 & 0 & \cdots & 0 & R_{k,k}
\end{bmatrix}.
\]

Setting \( Y^* = X^{-1} \) gives \( A = X\hat{R}Y^* \). Partitioning the columns of \( Y \) the same as the partitioning of \( X \) so that \( Y = [Y_1, Y_2, \ldots, Y_k] \) gives

\[
A = X_1 R_{1,1} Y_1^* + X_2 R_{2,2} Y_2^* + \cdots + X_k R_{k,k} Y_k^*.
\]

Since \( Y^*X = I \) we have

\[
Y_i^*X_j = \begin{cases} 
I_{n_j} & i = j \\
0 & i \neq j
\end{cases}
\]

where \( n_j \) is the order of \( R_{j,j} \). Moreover,

\[
I = \sum_{j=1}^{k} X_j Y_j^* \text{ since } XY^* = I.
\]

Thus any vector \( b \in \mathbb{C}^n \) may be written uniquely as

\[
b = \sum_{j=1}^{k} X_j (Y_j^* b) = \sum_{j=1}^{k} x_j, \quad (2.6)
\]
where \( x_j \in S_j \equiv \text{Range}(X_j) \). Hence
\[
\mathbb{C}^n = S_1 \cup S_2 \cup ...S_k \quad \text{with} \quad S_i \cap S_j = \emptyset \quad \text{if} \quad i \neq j.
\]

There are two main differences between this result and the result for normal matrices. The spaces \( S_j \) are not mutually orthogonal and a polynomial of degree greater than 1 in \( A \) may be needed to annihilate a space \( S_j \).

### 2.4.3 Exercises

1. Show that \( P_j \equiv X_j Y_j^* \) is a projector of \( \mathbb{C}^n \) onto \( S_j \equiv \text{Range}(X_j) \).

2. Prove that a projector \( P_j \) is orthogonal if and only if \( P_j \) is Hermitian.

3. Assume the columns of \( X \) each have norm 1 and suppose that \( \text{dim}(S_j) = 1 \). Show that \( \|P_j\| = \frac{1}{\cos \theta_j} \) where \( \theta_j \) is the angle between \( X_j \) and \( Y_j \).

4. Show that \( P_j \) as defined in the previous exercise is an orthogonal projector if and only if \( A \) is a normal matrix.

5. Define \( U_j \equiv \text{Range}(Y_j) \). Show that \( U_i \perp S_j \) when \( i \neq j \).

6. Show that although \( S_i \cap S_j = \emptyset \) and \( U_i \cap U_j = \emptyset \) for \( i \neq j \) that \( U_i \perp U_j \) and \( S_i \perp S_j \) does not necessarily hold for every \( i \neq j \). Show that if this property does hold for every \( i \neq j \) then \( A \) must be normal.

7. From exercise (1.10.2) we know that \( \psi(A)X_j = X_j \psi(R_{j,j}) \) for any polynomial \( \psi \). Let \( \psi_j(\lambda) = (\lambda - \lambda_j)^{n_j} \). Show that \( \psi(R_{j,j}) = 0 \).

### 2.5 The Cayley-Hamilton Theorem

With the structure of \( \mathbb{C}^n \) with respect to the eigen-spaces of a given matrix \( A \) it is relatively easy to prove the following:

**Theorem 2.7. (Cayley-Hamilton)** If \( p(\lambda) \) is the characteristic polynomial of \( A \in \mathbb{C}^{n \times n} \) then \( p(A) = 0 \).

**Proof.** Since \( A \) commutes with itself and the identity matrix, it is possible to write
\[
p(A) = p_j(A)(A - \lambda_j I)^{n_j}
\]
for each distinct eigenvalue \( \lambda_j \in \sigma(A) \) where \( p_j \) is a polynomial of degree \( n - n_j \). Since
\[
(A - \lambda_j I)^{n_j}X_j = X_j(R_{j,j} - \lambda_j I)^{n_j} = 0
\]
we have \( p(A)X_j = 0 \) for \( 1 \leq j \leq k \). From (2.6) it follows that \( p(A)b = 0 \) for every \( b \in \mathbb{C}^n \) and hence that \( p(A) = 0 \). \( \square \)

This intriguing theoretical result has profound implications for algorithms to compute eigenvalues and solutions of linear systems. With respect to linear systems \( Ax = b \) we know that \( p(A)b = 0 \) and \( p(0) = (-1)^n det(A) \neq 0 \). Thus we may write the polynomial \( p(\lambda)/p(0) = 1 - \lambda \phi(\lambda) \). Hence

\[
(I - A\phi(A))b = \frac{1}{p(0)} p(A)b = 0 \quad (2.7)
\]
to give

\[
Ax = b \quad \text{with} \quad x \equiv \phi(A)b.
\]

Therefore, we are highly motivated to devise algorithms that construct approximate solutions of the form \( x = \psi(A)b \) where \( \psi \) is a polynomial. Moreover, this result hints at the possibility of an algorithm which finds a solution of this type by computing at most \( n \) matrix vector products \( w = Av \) since \( \psi(A)b \) may be constructed one term at a time from such products. Since the expense of such an algorithm would be directly proportional to the degree of the polynomial \( \psi \), the question of minimizing the degree of the polynomial that is satisfied by \( A \) or that annihilates a given vector \( b \).

This discussion motivates the following two definitions

**Definition**: A monic polynomial \( p \) of degree \( k \) is called the *minimal polynomial* of \( A \) if \( p(A) = 0 \) and no monic polynomial of lower degree has this property.

**Definition**: A monic polynomial \( \psi \) of degree \( k \) is called the *minimal polynomial of \( A \) with respect to a vector \( b \)* if \( \psi(A)b = 0 \) and no monic polynomial of lower degree has this property.

### 2.5.1 Exercises

1. Use a degree argument to prove that each of the two definitions defines a unique polynomial.

2. Give a non-trivial (\( b \neq 0 \)) example to show that these two concepts are distinct.

3. Prove that when \( A \) is normal its minimal polynomial consists of the product of \( k \) linear factors \( (A - \lambda_j I) \). One for each distinct eigenvalue \( \lambda_j \) of \( A \).
4. Prove that \( A \) is diagonalizable if and only if its minimal polynomial \( \psi(\lambda) = \prod_{j=1}^{k}(\lambda - \lambda_j) \), where \( \{\lambda_j\} \) are the \( k \) distinct eigenvalues of \( A \).

5. Prove that every eigenvector of \( A \) may be written in the form \( \psi(A)b \) where \( \psi \) is a polynomial of degree at most \( n - 1 \) and \( b \) (almost) any nonzero vector. Which vectors \( b \) will not work here?

6. Prove that the polynomial \( \phi(\lambda) \) defined as in (2.7) is the (unique) polynomial of degree \( n - 1 \) that interpolates the function \( \frac{1}{\lambda} \) at the eigenvalues of \( A \). Also show that \( A^{-1} = \phi(A) \).

### 2.6 The Jordan Canonical Form

We see that the structure of the decomposition of \( \mathbb{C}^n \) with respect to eigenspaces of a matrix \( A \) is closely related to matrix polynomials and their degree. The next question we should ask is what is the smallest degree \( m \) of a polynomial in \( A \) that will annihilate the space \( S_j \). Let’s consider a special case. Define

\[
J_m(\mu) \equiv \begin{bmatrix} \mu & 1 & 0 & \cdots & 0 \\ \mu & 1 & \cdots \\ \vdots & \ddots & 0 \\ \vdots & & 1 \\ \mu & & & & \mu \end{bmatrix}, \tag{2.8}
\]

to be an upper triangular matrix of order \( m \) with \( \mu \in \mathbb{C} \).

#### 2.6.1 Exercise

1. Show that

\[
(\mu I - J_m(\mu))^l \quad \begin{cases} = 0 & l \geq m, \\ \neq 0 & 0 < l < m. \end{cases}
\]

These observations indicate that there may be a further decomposition of the eigen-structure of \( A \) into a more fundamental form. This is indeed true as we show in the following classic result.

**Theorem 2.8.** (Jordan Canonical Form) Any matrix \( A \in \mathbb{C}^{n \times n} \) has a decomposition of the form

\[
AX = XJ,
\]

D.C. Sorensen — draft — 21 January 2015
where \( X \in \mathbb{C}^{n \times n} \) is nonsingular and

\[
J = \begin{bmatrix}
J_{n_1}(\lambda_1) & & \\
& J_{n_2}(\lambda_2) & \\
& & \ddots \\
& & & J_{n_k}(\lambda_k)
\end{bmatrix},
\]

with diagonal blocks as in (2.8).

**Proof.** The proof of this theorem begins with the Schur decomposition and through a sequence of similarity transformations, reduces \( A \) to the desired form. The steps are:

1. Obtain the Schur decomposition \( Q^*AQ = R \) with

\[
R = \begin{bmatrix}
R_{1,1} & R_{1,2} & \cdots & R_{1,m} \\
0 & R_{2,2} & & \\
& & \ddots & \\
0 & 0 & \cdots & R_{m,m}
\end{bmatrix},
\]

where \( \sigma(A) = \{\lambda_j\} \) are the distinct eigenvalues of \( A \) and where \( R_{j,j} = \lambda_j I + U_j \) with \( U_j \) a strictly upper triangular matrix.

2. Construct a nonsingular matrix \( Z \) such that

\[
Z^{-1}RZ = \begin{bmatrix}
R_{1,1} & 0 & \cdots & 0 \\
0 & R_{2,2} & & \\
& & \ddots & \\
0 & 0 & \cdots & R_{m,m}
\end{bmatrix}.
\]

3. Determine nonsingular matrices \( Y_j \) such that

\[
Y_j^{-1}R_{j,j}Y_j = \lambda_j I + J_j,
\]

where

\[
J_j = \begin{bmatrix}
J_{k_1}(0) & & \\
& J_{k_2}(0) & \\
& & \ddots \\
& & & J_{k_l}(0)
\end{bmatrix},
\]

such that \( k_1 \geq k_2 \geq \cdots \geq k_l \).
2.6. The Jordan Canonical Form

4. This sequence results in

\[(QZY)^{-1}A(QZY) = \begin{bmatrix} \lambda_1 I + J_1 & & \\ & \lambda_2 I + J_2 & \\ & & \ddots \\ & & & \lambda_m I + J_m \end{bmatrix},\]

where

\[Y = \begin{bmatrix} Y_1 & 0 & \cdots & 0 \\ 0 & Y_2 & & \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & Y_m \end{bmatrix}.

5. The stated result (JCF) follows with \(X = QZY\) and \(J\) equaling the matrix on the right hand side of 4 since \(J_k(\mu) = \mu I + J_k(0)\).

Steps (1) and (2) have been established with the previous discussion. To accomplish step (3) we need to prove the following.

Lemma 2.9. If \(U \in \mathbb{C}^{n \times n}\) is strictly upper triangular, then there is a nonsingular \(Y\) such that

\[Y^{-1}UY = \begin{bmatrix} J_{k_1}(0) & & \\ & J_{k_2}(0) & \\ & & \ddots \\ & & & J_{k_m}(0) \end{bmatrix},\]

such that \(k_1 \geq k_2 \geq \cdots \geq k_m\).

Proof. The proof is by induction on \(n\). The lemma clearly holds for \(n = 1\). Assume it is true for strictly upper triangular matrices of order less than \(n\).

Since \(U\) is strictly upper triangular, there is a least positive integer \(k_1\) such that \(U^{k_1} = 0\). Let \(y\) be a vector such that \(U^{k_1-1}y \neq 0\). Put

\[Y_1 = [U^{k_1-1}y, U^{k_1-2}y, \ldots, Uy, y],\]

and observe that

\[UY_1 = Y_1J_{k_1},\]

and that the columns of \(Y_1\) are linearly independent (due to the strict upper triangular form of \(U\)). Now, complete the basis, i.e., construct \(Q_2\) such that
$[Y_1, Q_2]$ is nonsingular and

$$U[Y_1, Q_2] = [Y_1, Q_2] \begin{bmatrix} J_{k_1} & U_{12} \\ 0 & U_{22} \end{bmatrix},$$

with $U_{22}$ strictly upper triangular. By the induction hypothesis, there is a nonsingular $\hat{Y}_2$ such that

$$U_{22} \hat{Y}_2 = \hat{Y}_2 \hat{J}$$

where $\hat{J}$ is the direct sum of Jordan matrices of decreasing order. Now, we have

$$U[Y_1, Y_2] = [Y_1, Y_2] \begin{bmatrix} J_{k_1} & \hat{U} \\ 0 & \hat{J} \end{bmatrix},$$

where $Y_2 = Q_2 \hat{Y}_2$ and $\hat{U} = U_{12} \hat{Y}_2$. It is straightforward to verify that

$$U^{\ell+1}[Y_1, Y_2] = [Y_1, Y_2] \begin{bmatrix} J_{k_1} & \hat{U} \\ 0 & \hat{J} \end{bmatrix}^{\ell+1} = [Y_1, Y_2] \begin{bmatrix} J_{k_1}^{\ell+1} & \sum_{j=0}^{\ell} J_{k_1}^{\ell-j} \hat{U} \hat{J}^j \\ 0 & \hat{J}^{\ell+1} \end{bmatrix},$$

for any positive integer $\ell$ and hence that

$$\sum_{j=0}^{\ell} J_{k_1}^{\ell-j} \hat{U} \hat{J}^j = 0 \quad \text{and} \quad \hat{J}^{\ell+1} = 0$$

when $\ell = k_1 - 1$. We intend to show that the Sylvester’s equation

$$J_{k_1} W = U + W \hat{J}$$

has a solution $W$ even though it is singular. In fact, we claim that

$$W \equiv \sum_{j=0}^{\ell} S^{\ell-j} \hat{U} \hat{J}^j$$

is a solution where $S \equiv J_{k_1} + e_{k_1} e_1^T$ is the right circular shift operator. We shall verify this essentially by direct substitution.

First, we observe that $e_1^T W = 0$, since

$$e_1^T W = e_1^T \sum_{j=0}^{\ell} S^{\ell-j} \hat{U} \hat{J}^j = e_1^T \sum_{j=0}^{\ell} J_{k_1}^{\ell-j} \hat{U} \hat{J}^j = 0,$$
2.6. The Jordan Canonical Form

when \( \ell = k_1 - 1 \). Thus

\[
J_{k_1} W = (J_{k_1} + E_{k_1} e_1^T) W = SW.
\]

Then we compute (assuming \( \ell = k_1 - 1 \))

\[
U + W \hat{J} = U + \sum_{j=0}^{\ell} S^{\ell-j} \hat{U} J^{j+1}
\]

\[
= S^{\ell+1} U + S \sum_{j=0}^{\ell} S^{\ell-j-1} \hat{U} J^{j+1}
\]

\[
= S[S^{\ell} U + \sum_{j=0}^{\ell} S^{\ell-j-1} \hat{U} J^{j+1}]
\]

\[
= S[S^{\ell} U + \sum_{j=1}^{\ell+1} S^{\ell-j} \hat{U} J^{j}]
\]

\[
= S[S^{\ell} U + \sum_{j=1}^{\ell} S^{\ell-j} \hat{U} J^{j}]
\]

\[
= SW,
\]

where the second to last equality follows from the fact \( \hat{J}^{\ell+1} = 0 \) and we have used \( S^{\ell+1} = I \).

This solution to the Sylvester equation implies

\[
\begin{bmatrix}
J_{k_1} & 0 \\
0 & \hat{J}
\end{bmatrix}
\begin{bmatrix}
I & W \\
0 & I
\end{bmatrix} =
\begin{bmatrix}
I & W \\
0 & I
\end{bmatrix}
\begin{bmatrix}
J_{k_1} & \hat{U} \\
0 & \hat{J}
\end{bmatrix},
\]

and hence that

\[
UY = YJ
\]

where \( J = diag(J_{k_1}, \hat{J}) \) and \( Y = [Y_1, Y_2 - Y_1 W] \). This completes the induction and concludes the proof. \(\square\)

2.6.2 Exercise

1. Explain why the matrix factors in the Jordan Canonical Form are not continuous functions of the elements of the original matrix \( A \). What are the implications of this with respect to the development of a numerical algorithm to compute the Jordan Canonical Form?
2. The proof of the Jordan Canonical Form presented here is constructive. At what point is the discontinuity mentioned in the previous exercise apparent in the proof.

Lemma 1.6 uses a special case of the following more general result.

**Theorem 2.10.** *(Sylvester's Equation)* Suppose \( A \in \mathbb{C}^{n \times n} \), \( B \in \mathbb{C}^{k \times k} \), and \( C \in \mathbb{C}^{n \times k} \). Then the equation

\[
AX - XB = C,
\]

has a unique solution \( X \in \mathbb{C}^{n \times k} \) if and only if

\[
\sigma(A) \cap \sigma(B) = \emptyset.
\]

**Proof.** Let \( A = QR_AQ^* \) and \( B = UR_BU^* \), be the Schur decompositions of \( A \) and \( B \) respectively. Note that \( Q^*Q = I_n \) and \( U^*U = I_k \). Set \( X = QSU^* \), with \( S \in \mathbb{C}^{n \times k} \) to be determined. Then

\[
AX - XB = QR_ASU^* - QSR_BU^*,
\]

and (2.9) is satisfied if and only if

\[
R_AS - SR_B = \hat{C},
\]

where \( \hat{C} = Q^*CU \). The last row, \( s^*_n = e_n^TS \) may be determined from

\[
e_n^T R_A S - e_n^T SR_B = e_n^T \hat{C},
\]

giving the equation

\[
\rho_{n,n}e_n^T S - s^*_n R_B = e_n^* \hat{c}_n,
\]

i.e.

\[
s^*_n(\rho_{n,n}I - R_B) = e_n^* \hat{c}_n.
\]

This equation has a unique solution if and only if \( \rho_{n,n} \) is not an element of \( \sigma(B) \). Moreover, if we partition

\[
R_A = \begin{bmatrix} \hat{R}_A & r \\ 0 & \rho_{n,n} \end{bmatrix}, \quad S = \begin{bmatrix} \hat{S} \\ s^*_n \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} \hat{S}^* \\ e_n^* \end{bmatrix},
\]
we obtain the equation
\[ \hat{R}_A \hat{S} - \hat{S} R_B = \hat{C} - rs_n^*. \]

The last row of \( \hat{S} \) may now be determined using the same mechanism. Repeating this process all the rows of \( \hat{S} \) may be uniquely determined if and only if \( \{ \rho_{j,j} \}_{j=1}^n \cap \sigma(B) = \emptyset \), i.e. \( \sigma(A) \cap \sigma(B) = \emptyset \). \( \square \)

2.6.3 Exercise

1. In the case \( \mu = 1 \) in the proof of lemma 1.9 we constructed a solution \( M \) to
\[ e_1 u_2^* + MJ_2 = \hat{J}_1 M, \]
even though \( \sigma(J_2) \cap \sigma(\hat{J}_1) = \{ 0 \} \neq \emptyset \). Does this contradict Sylvester’s theorem? Why not?

2. Let \( A \in \mathbb{C}^{m \times k} \) and \( B \in \mathbb{C}^{n \times l} \). Define the tensor product \( A \otimes B \) to be
\[ A \otimes B = (\alpha_{ij} B), \]
where \( A \otimes B \in \mathbb{C}^{mn \times kl} \), i.e. a block matrix with the \( i,j \)-th block equal to \( \alpha_{ij} B \). If \( X \in \mathbb{C}^{n \times k} \) and we define
\[ \hat{X} = \begin{bmatrix} X e_1 \\ X e_2 \\ \vdots \\ X e_k \end{bmatrix} \quad \text{and} \quad \hat{C} = \begin{bmatrix} C e_1 \\ C e_2 \\ \vdots \\ C e_k \end{bmatrix}, \]
show that
\[ DX -XE = C \]
if and only if
\[ (I_k \otimes D) \hat{X} - (E^T \otimes I_n) \hat{X} = \hat{C}, \]
i.e.
\[ (I_k \otimes D - E^T \otimes I_n) \hat{X} = \hat{C}, \]
where \( D \in \mathbb{C}^{n \times n} \) and \( E \in \mathbb{C}^{k \times k} \).

3. If \( AC \) and \( BC \) are defined then \( (A \otimes B)(C \otimes D) = AC \otimes BD \).

4. If \( AB \) and \( AC \) are defined then \( A \otimes (B + C) = A \otimes B + A \otimes C \).
5. If \( AB \) is defined and \( A \) and \( B \) are nonsingular then \((A \otimes B)^{-1} = A^{-1} \otimes B^{-1}\).

6. Use the Schur decompositions \( AQ = QR_A \) and \( BU = UR_B \) and the tensor product theory developed in the previous exercises to prove Theorem (1.10).

7. Suppose that
\[
AU = U\Lambda \quad \text{and} \quad BV = V\Omega
\]
where \( V^*V = I_n = U^*U \) and both \( \Lambda \) and \( \Omega \) are diagonal matrices containing the eigenvalues of \( A \) and \( B \) respectively. Show that
\[
(A \otimes B)(U \otimes V) = (U \otimes V)(\Lambda \otimes \Omega).
\]
Conclude that \((A \otimes B)q_{i,j} = q_{i,j}\lambda_i\omega_j\) where \( q_{i,j} = Ue_i \otimes Ve_j \).

8. Consider the eigenvalue problem,
\[
-\frac{d^2}{d\xi^2} u(\xi) = \nu u(\xi),
\]
with boundary conditions \( u(0) = 0 = u(1) \). The eigenvalue problem is to determine the non-zero functions \( u(\xi) \) and the corresponding \( \nu \). Using finite difference approximations on the uniform partition,
\[
0 = \xi_0 < \xi_1 < \cdots < \xi_n < \xi_{n+1} = 1,
\]
we obtain the following finite dimensional system of order \( n \),
\[
-\frac{1}{h^2}T_n u_k = \lambda_k u_k.
\]
The eigenvector \( u_k = [\mu_{1,k}, \cdots, \mu_{n,k}]^T \) represents the approximations \( \mu_{i,k} \) to the eigenfunction \( u_k(\xi) \) where \( \xi_{i+1} = \xi_i + h \) for \( i, k = 1 : n \) and \( h = \frac{1}{n+1} \). The coefficient matrix \( T_n \) is the 1-D Laplacian which has the following tridiagonal structure:
\[
T_n = \begin{bmatrix}
2 & -1 & 0 & \cdots & 0 \\
-1 & 2 & -1 & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
\vdots & -1 & 2 & -1 & \ddots \\
0 & \cdots & 0 & -1 & 2
\end{bmatrix}
\]
2.6. The Jordan Canonical Form

Use Matlab to graph the eigenvectors. Why do they look like \( \sin(k\pi \theta) \) for \( 0 \leq \theta \leq 1 \)? The 2-D Laplacian is

\[
-\Delta u = -\left( \frac{\partial}{\partial \xi} + \frac{\partial}{\partial \eta} \right) u,
\]

with boundary condition \( u = 0 \) on the unit square. If as above we take equally spaced points \( \xi_i \) and \( \eta_j \) on the unit square show that the discrete 2-D Laplacian is of the form

\[
L_n = \begin{bmatrix}
T_n + 2I_n & -I_n & 0 & \cdots & 0 \\
-I_n & T_n + 2I_n & -I_n & \vdots \\
0 & \cdots & \cdots & \cdots & 0 \\
\vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & 0 & -I_n & T_n + 2I_n & -I_n
\end{bmatrix}.
\]

Also show that \( L_n = T_n \otimes I + I \otimes T_n \). What is eigensystem of \( L_n \)? Use Matlab to graph the eigenvectors as 2-D surfaces over the unit square.
3 · Krylov Methods

3.1 Richardson’s Iteration

A splitting method may be viewed from a different perspective with a simple rearrangement of terms. Noting that $A = M - N$ implies $N = M - A$, the iteration

$$Mx^{(k+1)} = Nx^{(k)} + b \quad \text{becomes} \quad Mx^{(k+1)} = Mx^{(k)} + (b - Ax^{(k)}).$$

Hence,

$$x^{(k+1)} = x^{(k)} + M^{-1}(b - Ax^{(k)}). \quad (3.1)$$

This is suggestive of a Newton-like method for solving $f(x) = 0$ where $f(x) = Ax - b$. A natural “improvement” might be to include a line search parameter $\alpha$ and consider

$$x(\alpha) = x^{(k)} + \alpha M^{-1}(b - Ax^{(k)}).$$

One might, for example, choose $\alpha$ to minimize $\|b - Ax(\alpha)\|$ and set

$$x^{(k+1)} = x(\alpha_{k+1}) \quad \text{where} \quad \alpha_{k+1} = \arg \min_{\alpha} \|b - Ax(\alpha)\|. \quad \text{(3.2)}$$

Before continuing, notation can be simplified by considering the “pre-conditioned” linear system

$$M^{-1}Ax = M^{-1}b.$$

If $A \leftarrow M^{-1}A$ and $b \leftarrow M^{-1}b$, then iteration (3.1) becomes

$$x^{(k+1)} = x^{(k)} + \alpha_{k+1}(b - Ax^{(k)}). \quad (3.2)$$
Now, set \( r^{(k)} = b - Ax^{(k)} \) so that Richardson’s iteration becomes
\[
x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}.
\]
Hence,
\[
b - Ax^{(k+1)} = b - Ax^{(k)} - \alpha_k r^{(k)},
\]
so that
\[
r^{(k+1)} = r^{(k)} - \alpha_k r^{(k)} = (I - \alpha_k A)r^{(k)}. \tag{3.3}
\]
From this equation it is easily seen that
\[
r^{(k)} = (I - \alpha_k A)r^{(k-1)} = (I - \alpha_k A)(I - \alpha_{k-1} A)r^{(k-2)} = \ldots = (I - \alpha_1 A)b.
\]
Hence,
\[
r^{(k)} = p_k(A)r^{(0)} \quad \text{where} \quad p_k(t) = \prod_{j=1}^{k} (1 - \alpha_j t).
\]
Note that the polynomial \( p_k \) is of degree \( k \) and also that \( p_k(0) = 1 \). Hence,
\[
p_k(t) = 1 - \gamma_1 t - \gamma_2 t^2 - \ldots - \gamma_k t^k = 1 - t\phi_{k-1}(t),
\]
where \( \phi_{k-1}(t) = \gamma_1 + \gamma_2 t + \gamma_3 t^2 + \ldots + \gamma_k t^{k-1} \). From this
\[
r^{(k)} = p_k(A)r^{(0)} = [I - A\phi_{k-1}(A)]b \quad \text{if} \quad x^{(0)} = 0.
\]
Since \( A \) is non-singular, the fact that \( b - Ax^{(k)} = b - A\phi_{k-1}(A)b \) implies that \( x^{(k)} = \phi_{k-1}(A)b \), for some polynomial \( \phi_{k-1} \) of degree \( k - 1 \).

3.2 The Krylov Subspace:

The discussion of Richardson’s method has revealed some important observations about iterative methods based on splitting. First, it is now clear that since any splitting method is equivalent to some form of Richardson’s method, after \( k \)-steps the iteration results in a linear system residual that involves a matrix polynomial \( (r^{(k)} = p_k(A)r^{(0)} \) with \( p_k(0) = 1 \). Second, the approximate solution also involves a matrix polynomial \( (x^{(k)} = \phi_{k-1}(A)b) \). Finally, the Cayley-Hamilton Theorem implies that \( k \leq n \) is theoretically sufficient since \( p_n(A) = 0 \) if \( p_n(t) = p(t)/p(0) \) where \( p(t) = \det(tI - A) \).

Hence, \( x^k \) is contained in the Krylov subspace
\[
\mathcal{K}_k(A, b) = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\} \subseteq \mathbb{R}^n. \tag{3.4}
\]
3.2. The Krylov Subspace:

An equivalent definition is

\[ \mathcal{K}_k(A, b) \equiv \{ p(A)b : p \in \mathbb{P}_{k-1} \} \subseteq \mathbb{R}^n. \]  (3.5)

This alternate definition is quite useful as it allows conditions involving concrete vectors to have equivalent expressions involving polynomials.

Computation on elements from a Krylov subspace requires a set of basis vectors. One option is immediate from the definition, namely the power basis \( \{ A^j b : j = 1, 2, \ldots, k-1 \} \). In practice this is almost always a terrible choice because this basis is usually very ill-conditioned, i.e., some of the basis vectors form a small angle with the subspace spanned by the others. This observation becomes evident when one notices that these basis vectors are iterates of the power method for eigenvalue computation (see, e.g., [?], §7.3): these vectors rapidly align with the dominant eigenvector, hence become very nearly linearly dependent. An orthonormal basis for the Krylov space will provide a far more stable foundation for the development of numerical algorithms.

3.2.1 The Arnoldi process

For the remainder of this section, assume that \( \mathcal{K}_k(A, b) \) has dimension \( k \) for all \( k \leq n \) (independence assumption).

An orthonormal basis \( \{ v_1, v_2, \ldots, v_k \} \) for \( \mathcal{K}_k(A, b) \) can be constructed by building up bases for the earlier Krylov subspaces one at a time. This is possible because the successive subspaces are nested:

\[ \mathcal{K}_1(A, b) \subset \mathcal{K}_2(A, b) \subset \mathcal{K}_3(A, b) \subset \cdots \subset \mathcal{K}_n(A, b). \]

Suppose, inductively, that an orthonormal basis for \( \mathcal{K}_j(A, b) \) has been computed and is stored in the \( j \) columns of the matrix \( V_j \in \mathbb{R}^{n \times j} \), and that \( V_i \in \mathbb{R}^{n \times i} \), the first \( i \) columns of \( V_j \), form an orthonormal basis for \( \mathcal{K}_i(A, b) \), \( i = 1, 2, \ldots, j \). Set \( v_1 = b/\|b\| \) and for each \( i \), note that \( v_i \in \mathcal{K}_i(A, b) \) implies that \( v_i = \phi_{i-1}(A)v_1 \) for some polynomial \( \phi_{i-1} \) of exact degree \( i-1 \) (A polynomial of lower degree would violate the independence assumption). Hence, of all the vectors \( w_i = Av_i = A\phi_{i-1}(A)v_1 \in \mathcal{K}_{i+1}(A, v_1) \), only \( w_j \) has a component in \( \mathcal{K}_{j+1}(A, b) \). To get the next orthonormal basis vector \( v_{k+1} \), we will extract the component of \( w_j \) that is in \( \mathcal{K}_{j+1}(A, b) \) but orthogonal to \( \mathcal{K}_j(A, b) \). To expose this component, break \( w_j \) into the sum of its part in \( \mathcal{K}_j(A, b) \) and its component outside \( \mathcal{K}_j(A, b) \):

\[ w_j = V_j h_j + f_j, \quad \text{with} \quad h_j = V_j^* w_j. \]
Since $V^TV_j = I_j$ it follows that $V_j^*f_j = 0$. Now, just put $v_{j+1} = f_j/\beta_j$, where $\beta_j = \|f_j\|$. The independence assumption ensures that $\beta_j > 0$ for $j = 1, \ldots, n - 1$. This algorithm is summarized in Figure 3.1.

\[ [V, H, f] = \text{Arnoldi}(A, b, k) \]

**Input:**
- $A$ an $n \times n$ matrix
- $b$ a vector of length $n$
- $k$ a positive integer ($k \leq n$)

**Output:**
- $V$ an $n \times k$ matrix with orthonormal columns
- $H$ a $k \times k$ upper Hessenberg matrix
- $f$ a vector of length $n$

with $AV = VH + fe_k^T$ and $V^Tf = 0$

\[ v_1 \leftarrow b/\|b\| \]
\[ w \leftarrow Av_1, \quad \alpha_1 \leftarrow v_1^Tw \]
\[ f_1 \leftarrow w - v_1\alpha_1 \]
\[ V_1 \leftarrow [v_1]; \quad H_1 \leftarrow [\alpha_1] \]

for $j = 1, 2, \ldots, k - 1$

\[ \beta_j \leftarrow \|f_j\|, \quad v_{j+1} \leftarrow f_j/\beta_j \]
\[ V_{j+1} \leftarrow [V_j, v_{j+1}] \]
\[ \hat{H}_j \leftarrow \begin{bmatrix} H_j & \beta_j e_j^T \\ \beta_j e_j & \beta_j \end{bmatrix} \]
\[ w \leftarrow Av_{j+1} \]
\[ h \leftarrow V_{j+1}^Tw \]
\[ f_{j+1} \leftarrow w - V_{j+1}h \]
\[ H_{j+1} \leftarrow [\hat{H}_j, h] \]

end

Figure 3.1. $k$-step Arnoldi factorization.

This derivation gives the Arnoldi process for constructing an orthonormal basis for the Krylov space. The construction can be expressed in matrix form as

\[ AV_k = V_k H_k + fe_k^T \]  \hspace{1cm} (3.6)
where the $j$th column of $H_k$ is
\[
\begin{bmatrix}
h_j \\ \beta_j \\ 0
\end{bmatrix} \in \mathbb{R}^k,
\]
so $H_k$ is upper Hessenberg; the subdiagonal elements $\beta_j = \|f_j\|$ that normalize $v_j$ are always nonnegative (and positive, under the independence assumption). The leading $j$ columns of $V_k$ give the matrix $V_j$ for $1 \leq j \leq k$. The orthogonality conditions imply that $H_k = V^*_k A V_k$.

Noting that $v_{k+1} \beta_k = f_k$ and that $V_{k+1} = [V_k, v_{k+1}]$, it is often useful to express (3.6) as
\[
AV_k = V_{k+1} \tilde{H}_k,
\] (3.7)
where
\[
\tilde{H}_k = \begin{bmatrix} H_k \\ \beta_k e_k^T \end{bmatrix}.
\]

### 3.2.2 Iterative Methods via Arnoldi Factorization

Numerous iterative methods for solving $Ax = b$ can be derived from the Arnoldi k-step factorization. Two of the most important variants are the Full Orthogonalization Method (FOM) and the Generalized Minimum Residual Method (GMRES). In the derivations that follow, it is always assumed that the starting guess $x(0) = 0$ and that $\|b\| = 1$ (hence $v_1 = b$).

There is no loss of generality with these two assumptions. If $x(0) \neq 0$ simply replace $b \leftarrow b - Ax(0)$. Then reset $x^0 = 0$ and proceed. Since
\[
b - Ax = (b - Ax(0)) - A(x - x(0)),
\]
on convergence (to $x^{\text{conv.}}$), simply replace $x^{\text{conv.}} \leftarrow x^{\text{conv.}} + x(0)$ using the original $x(0)$. Moreover, if $\|b\| = \beta_0 \neq 0$ replace $b \leftarrow b / \beta_0$ and on convergence replace $x^{\text{conv.}} \leftarrow x^{\text{conv.}} \beta_0$.

**Full Orthogonalization, FOM:** Perhaps the first thing that comes to mind is to apply a Galerkin procedure. To this end, expand the unknown iterate $x^{(k)} = V_k y \in \mathcal{K}_k(A, b)$ where $y \in \mathbb{R}^k$ is to be determined. Next, force the residual to be orthogonal to all vectors $w \in \mathcal{K}_k(A, b)$. This yields
\[
0 = w^T (b - AV_k y) = z^T V_k^T (b - AV_k y) \quad \text{for all} \quad z \in \mathbb{R}^k.
\] (3.8)

Now,
\[
z^T V_k^T (b - AV_k y) = z^T (V_k^T b - V_k^T [V_k H_k y + f e_k^T y]) = z^T (e_1 - H_k y),
\]
since $V_k^T b = e_1$, $V_k^T V_k = I_k$ and $V_k^T f = 0$.

Hence the Galerkin condition of (3.8) is satisfied by the following:

\begin{enumerate}
  \item Solve $H_k y = e_1$, and
  \item put $x^{(k)} = V_k y$.
\end{enumerate}

This simple procedure yields a good algorithm in many cases but it has two drawbacks. First, it is tacitly assumed that $H_k$ is nonsingular at each step $k$. Examples with $H_k$ singular are easily constructed but even more important numerically, $H_k$ can become very ill-conditioned in practice. Second, it is possible to do better; FOM is not optimal in the sense that it does not give the minimal residual possible from the constructed Krylov subspace.

**Generalized Minimum Residual (GMRES):** Ideally, we would like to make the best possible use of the stored information in $V_k$, $H_k$, $f_k$. Is it possible to solve

$$\min_{x \in \mathcal{K}_k(A, b)} \| b - Ax \|^2.$$  

Recall that with “clairvoyance” in the choice of the parameters $\alpha_j$ in Richardson’s method it would be possible to do this, i.e. to solve

$$\min \| b - Ax^{(k)} \| = \min_{p \in \mathcal{P}_k, p(0) = 1} \| p(A) b \| = \min_{\phi \in \mathcal{P}_{k-1}} \| (I - A \phi(A)b) \|.$$  

While this formulation is completely impractical, with the aid of the Arnoldi factorization, the minimum residual problem may easily be solved.

Just note that

$$\min_{\phi \in \mathcal{P}_{k-1}} \| (I - A \phi(A)b) \| = \min_{x \in \mathcal{K}_k(A, b)} \| b - Ax \|.$$  

Moreover,

$$\min_{x \in \mathcal{K}_k(A, b)} \| b - Ax \| = \min_{y \in \mathbb{R}^k} \| b - AV_k y \|.$$  

Now, $b - AV_k y = V_{k+1} e_1 - V_{k+1} \hat{H}_k y$ and thus for $x = V_k y$

$$\| b - Ax \| = \| V_{k+1} (e_1 - \hat{H}_k y) \| = \| e_1 - \hat{H}_k y \|,$$

since $V_{k+1}$ has ortho-normal columns. Thus our goal is achieved by the following:

\begin{enumerate}
  \item Solve $\min_{y} \| e_1 - \hat{H}_k y \|$ and
  \item put $x^{(k)} = V_k y$.
\end{enumerate}

This provides the GMRES method.
The normal equations for this reduced problem are
\[ \hat{H}_k^T \hat{H}_k y = \hat{H}_k^T e_1 = H_k^T e_1. \]
Observe that \( \hat{H}_k \) must be of full rank \( k \) since the main subdiagonal will
consist of \( \beta_j, \ 1 \leq j \leq k \) and all of these are positive under the independence
assumption. Hence the problem of a singular \( H_k \) is completely finessed.
Moreover, if the independence assumption is violated and \( \beta_k = 0 \) then the
FOM solution is exact since the residual will be zero! In other words, if \( \beta_k = 0 \) occurs at some step \( k \) then the iteration may be terminated with \( x^k \)
being the exact solution to the linear system.

3.2.3 An Alternate Derivation

There is an alternate derivation of the Arnoldi k-step factorization that is
of no use numerically, but lends some insight to the structure of the Arnoldi
factorization. An ill-advised but straightforward construction begins with a
matrix \( K \) composed of the power basis,
\[ K = [b, Ab, A^2b, \ldots, A^{k-1}b]. \]
Let
\[ S = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & & \\
0 & 1 & 0 & 0 & \\
\vdots & \ddots & \ddots & \ddots & \\
0 & \cdots & 0 & 1 & 0
\end{bmatrix}, \]
and observe that
\[ AK = KS + A^K b = K(S + \hat{g} e_k^T) + (A^K b - K \hat{g}) e_k^T, \]
where \( \hat{g} \in \mathbb{R}^k \) is to be determined. Clearly (under the independence
assumption) the \( QR \) Factorization of \( K \) into \( K = QR \) with \( Q^T Q = I_k \) and
\( R \) upper triangular (with positive diagonal elements) will provide an ortho-
normal basis for \( \mathcal{K}_k(A, b) \) since \( \text{Ran}(Q) = \mathcal{K}_k(A, b) \). It is now easily seen
that
\[ AQ = QR(S + \hat{g} e_k^T)R^{-1} + \rho_n^{-1}(A^K b - K \hat{g}) e_k^T = QG + \hat{g} e_k^T, \]
where \( G, g \) have the obvious definitions. It is easily seen that \( G \) must be
upper Hessenberg and also \( G \) is similar to the companion matrix \( S + \hat{g} e_k^T \).
In addition, note that $\phi(A)b = A^k b - K\hat{g}$ is a monic (leading coefficient 1) polynomial of degree $k$ and that $\phi(t)$ is the characteristic polynomial of the matrix $S + \hat{g}e_k^T$. Hence $\phi(t)$ is also the characteristic polynomial of $G$. Now, specify the vector $\hat{g}$ by solving the least squares problem

$$
\min_{\hat{g}} \|A^k b - K\hat{g}\|, \quad i.e., \quad \hat{g} = R^{-1}Q^T A^k b.
$$

With this choice of $\hat{g}$ it follows that $Q^T g = 0$.

The immediate question is: Does this give the same $k$-step factorization as Arnoldi? The answer lies within the very simple yet profound Implicit $Q$-Theorem:

**Theorem 3.1.** (Implicit $Q$) Suppose

$$
AQ = QG + ge_k^T \quad \text{and} \quad AV = VH + fe_k^T
$$

with both $G$ and $H$ upper Hessenberg with positive diagonal elements, $Q^T g = 0$, $V^T f = 0$ and with $Q^T Q = V^T V = I_k$. If $Qe_1 = Ve_1$ then

$$
V = Q, \quad H = G, \quad f = g.
$$

**Proof.** For a given $A$, the steps of the Arnoldi factorization are uniquely determined by the first column $v_1 = Ve_1$. $lacksquare$

### 3.2.4 The GMRES and FOM Polynomials

Both the FOM and GMRES methods produce an iterate $x^{(k)} \in \mathcal{X}_k(A, b)$. Hence, in either case $x^{(k)} = \phi_{k-1}(A)b$ where $\phi_{k-1}$ is a polynomial of degree $k - 1$. The linear system residual is then

$$
r^{(k)} = b - Ax^{(k)} = [I - A\phi_{k-1}(A)]b = p_k(A)b.
$$

The goal is to force $\|r^{(k)}\| \to 0$ as $k$ increases. Supposing for the moment that $AX = X\Lambda$ is diagonalizable (with $X$ nonsingular and $\Lambda$ diagonal) observe that

$$
\|r^{(k)}\| = \|p_k(A)b\| = \|p_k(A)X\hat{b}\| \leq \|X\|\|p_k(\Lambda)\hat{b}\|, \quad (3.9)
$$

where $\hat{b} = X^{-1}b$.

Since $\|p_k(\Lambda)\hat{b}\|^2 = \sum_{j=1}^n |p_k(\lambda_j)|^2 |\hat{b}(j)|^2$, it is clear that the desired goal $\|r^{(k)}\| \to 0$ can only be attained if $p_k(\lambda_j) \to 0$ for $1 \leq j \leq n$. Yet another interpretation of (3.9) is

$$
\|r^{(k)}\| = \|p_k(A)X\hat{b}\| = \|XP_k(\Lambda)X^{-1}b\| \leq \kappa_2(X) \max_j |p_k(\lambda_j)|.
$$

D.C. Sorensen - draft - 21 January 2015
3.2. The Krylov Subspace:

To study these polynomials it is helpful to characterize them in terms of concrete quantities available in the k-Step Arnoldi factorization.

**Fundamental Theorem of Krylov Spaces**

**Theorem 3.2.** Suppose \( AV = VH + fe_k^T \) with \( V \in \mathbb{C}^{n \times k} \) of full rank \( k \), and \( H \in \mathbb{C}^{k \times k} \) an unreduced upper Hessenberg matrix (i.e., all subdiagonal entries \( \beta_1, \ldots, \beta_{k-1} \) are nonzero). If \( \phi \) is a monic polynomial (leading coefficient 1), then

\[
\phi(A)v_1 = \begin{cases} 
  V\phi(H)e_1, & \text{if } \deg(\phi) < k; \\
  V\phi(H)e_1 + f\gamma, & \text{if } \deg(\phi) = k,
\end{cases}
\]

where \( \gamma = \prod_{j=1}^{k-1} \beta_j \).

**Proof.** Due to linearity, it is only necessary to establish the result for the monomials, \( \phi(z) = z^j \). If \( A^{j-1}v_1 = VH^{j-1}e_1 \), then

\[
A^jv_1 = A(A^{j-1}v_1) = AVH^{j-1}e_1 = (VH + fe_k^T)H^{j-1}e_1 = VH^j e_1,
\]

when \( j < k \) since \( e_k^T H^{j-1} e_1 = 0 \). If \( j = k \), then the same argument will give

\[
A^kv_1 = VH^k e_1 + fe_k^T H^{k-1} e_1.
\]

It is straightforward to verify that \( \gamma = e_k^T H^{k-1} e_1 \), and hence the result follows.

Numerous approximation results follow directly from this simple theorem. It may be used here to characterize the FOM and GMRES polynomials

**The FOM Polynomial:** In FOM, \( x^{(k)} \) is obtained by solving \( H_k y = e_1 \) and setting \( x^{(k)} = V_k y \). Let \( p(t) = \det(tI - H_k) \) and put \( p_k(t) = p(t)/p(0) \). The FOM residual is

\[
r^{(k)} = b - Ax^{(k)} = V_k(e_1 - H_k y) - f_k y(k) = -f_k y(k).
\]

Now according to Theorem (3.2)

\[
p_k(A)v_1 = V_k p_k(H_k)e_1 + f_k \gamma_k = f_k \gamma_k,
\]

since \( p_k(H_k) = 0 \). Hence,

\[
f_k = \gamma p_k(A)v_1,
\]

D.C. Sorensen – draft – 21 January 2015
where $\gamma p_k(t)$ is a multiple of the characteristic polynomial of $H_k$. The roots of this polynomial are the eigenvalues of $H_k$. These numbers are called Ritz values.

**The GMRES Polynomial:** In GMRES, $x^{(k)}$ is obtained by solving $\min_y \|e_1 - \hat{H}_k y\|$ and setting $x^{(k)} = V_k y$. This formula was derived during the course of solving $\min_y \|b - AVy\|$. The normal equations for this formulation are

$$(AV)^T(AV)y = (AV)^Tb$$

and the optimal linear system residual $b - Ax^{(k)}$ satisfies

$$0 = (AV)^T(b - Ax^{(k)}) = V^TA^T(p(A)b)$$

for some $k$-degree polynomial $p$. Let $\lambda_j$ be any root of $p$ and define $\phi_j(t)$ via $p(t) = (t - \lambda_j)\phi_j(t)$. Then

$$0 = V^TA^T(A - \lambda_j I)^{-1}(A)e_1 = VA^TAV\phi_j(H)e_1 - V^TA^TV\phi_j(H)e_1\lambda_j.$$

Equivalently,

$$\hat{H}^T\hat{H}y_j = H^Ty_j\lambda_j, \text{ with } y_j \equiv \phi_j(H)e_1.$$

Thus, the GMRES polynomial has the generalized eigenvalues of the pair $(\hat{H}^T\hat{H}, H^T)$ as roots. These are called Harmonic Ritz values.

When $H$ is singular (a possibility) then this generalized eigenvalue problem has infinite roots. To avoid this situation, the Harmonic Ritz values are often defined as the reciprocals of the generalized eigenvalues of the pair $(H^T, \hat{H}^T\hat{H})$ which are always finite since $\hat{H}^T\hat{H}$ is positive definite (under linear independence assumption).

An alternate derivation of the GMRES polynomial follows in a straightforward but more tedious application of the FTK. The normal equations for $\min_y \|e_1 - \hat{H}_k y\|$ are

$$\hat{H}_k^T\hat{H}_k y = \hat{H}_k^T e_1 = \hat{H}_k^T e_1.$$

Since

$$\hat{H}_k^T\hat{H}_k = H_k^T H_k + \beta_k^2 e_k e_k^T,$$

it follows that

$$(H_k + \beta e_k^T)y = e_1$$
(assuming $H_k$ is nonsingular) where $g = \beta_k^2 H_k^{-T} e_k$.

Thus,

$$y = (H_k + ge_k^T)^{-1} e_1 = \phi_{k-1}(H_k + ge_k^T) e_1,$$

where $1 - t\phi_{k-1}(t) = p(t)/p(0)$ with $p(t) = \det(tI - (H_k + ge_k^T))$.

Now, it is straightforward to show

$$\phi_{k-1}(H_k + ge_k^T) e_1 = \phi_{k-1}(H_k) e_1,$$

using essentially the same argument as in the proof of Theorem (3.2). The GMRES residual is, therefore,

$$r^{(k)} = b - Ax^{(k)} = b - AV_{k-1}(H_k) e_1 = b - A\phi_{k-1}(A) b = \gamma p(A) b$$

due to Theorem (3.2). The roots of the GMRES polynomial are the eigenvalues of the matrix $H_k + ge_k^T$. These numbers are called Harmonic Ritz values.

Graphics of GMRES polynomial, convergence, roots go here

3.2.5 Implementation Issues

The main computational issue for Krylov methods is the construction of the orthogonal basis $V_k$. As discussed above, one could simply compute the QR-factorization of the matrix $K$. However, this mathematically correct procedure is bound to fail numerically. The power sequence $\{A^j b\}$ becomes evermore linear dependent as $j$ increases, primarily due to the fact that eventually $A^j b$ aligns with the dominant eigenvector of $A$ as $j$ increases.

The orthogonalization step in Arnoldi

$$w = Av$$
$$h = V^T w$$
$$f = w - Vh = (I - VV^T) w$$

is a Classical Gram-Schmidt orthogonalization and this is notoriously numerically unstable.

Re-Orthogonalization: There is a simple scheme for numerically producing orthogonal basis vectors (columns of $V$). The scheme was proposed and analyzed by Daniel, Gragg, Kaufman and Stewart (DGKS) in [?].

$$w = Av$$
$$h = V^T w$$
Mathematically, \( c = 0 \) should hold, but this is often not the case numerically. This is particularly so when the vector \( w \) is nearly in \( \text{Ran}(V) \) which is precisely the case here. The basic idea is that while \( w \) is nearly in \( \text{Ran}(V) \), the vector \( f = w - Vh \) is not. It generally makes a significant angle with \( \text{Ran}(V) \) and as a result the vector \( c \) will have enough accuracy to correct the errors in \( f \) when the operation \( f \leftarrow Vc \) is completed.

This concept is illustrated graphically in the following:

This DGKS scheme will produce numerically orthogonal vectors to working precision. However, the cost is significant. The amount of arithmetic is doubled at each Arnoldi step.

**Modified Gram-Schmidt:** An alternative is the Modified Gram-Schmidt
3.2. The Krylov Subspace:

\[
\begin{align*}
\mathbf{w} &= \mathbf{A}\mathbf{v} \\
\text{for } i = 1:k, \quad &\mathbf{h}(i) = \mathbf{v}_i^T\mathbf{w} \\
\mathbf{w} &= \mathbf{w} - \mathbf{v}_i\mathbf{h}(i) \\
\text{end} \\
\mathbf{f} &= \mathbf{w}.
\end{align*}
\]

Figure 3.3. MGS orthogonalization.

(MGS) process for computing the projection \( \mathbf{f} = (\mathbf{I} - \mathbf{V}\mathbf{V}^T)\mathbf{w} \). The MGS process results from the observation

\[
\mathbf{I} - \mathbf{V}\mathbf{V}^T = \prod_{i=1}^{k}(\mathbf{I} - \mathbf{v}_i\mathbf{v}_i^T),
\]

where \( \mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k] \) is orthonormal, \( \mathbf{V}^T\mathbf{V} = \mathbf{I}_k \). The proof is straightforward and will be left as an exercise.

With this observation it follows that \( \mathbf{f} \) may be computed as follows:

\[
\mathbf{w} \leftarrow \mathbf{w} - \mathbf{v}_i(\mathbf{v}_i^T\mathbf{w}), \quad \text{for } i = 1, 2, \ldots, n,
\]

and then set \( \mathbf{f} = \mathbf{w} \). This translates into the following Matlab style code

This MGS scheme will produce a basis that sufficient to use for the GMRES and FOM iterations, but the vectors can be far from orthogonal, particularly in the case of the Arnoldi factorization.

**Solving the GMRES Least Squares Problem:** In most cases (where \( n \gg k \)) the cost of solving the \((k+1)\times k\) reduced order least squares problem is negligible when compared to the cost of orthogonalization. However, there are some intricacies that are instructive and which also lead to robust and efficient implementation. For this discussion, iteration indices will be dropped to generically describe the QR-factorization and solution updates between steps. Assume \( \mathbf{H} = \mathbf{QR} \) is a QR-factorization of \( \mathbf{H} = \mathbf{H}_k \). Then

\[
\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H}^T \\ \beta e_k^T \end{bmatrix} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \beta \mathbf{e}_k^T \end{bmatrix} \begin{bmatrix} \mathbf{R} \\ \mathbf{e}_k^T \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{r} \\ \mathbf{0} & \hat{\beta} \end{bmatrix},
\]

with

\[
\begin{bmatrix} \mathbf{R} \\ \beta e_k^T \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{R}} & \mathbf{r} \\ \mathbf{0} & \hat{\beta} \end{bmatrix}.
\]

D.C. Sorensen – draft – 21 January 2015
Now, construct a Givens transformation $G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ such that $c^2 + s^2 = 1$ and
\[
\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} \hat{\rho} \\ \beta \end{bmatrix} = \begin{bmatrix} \rho \\ 0 \end{bmatrix} \quad \text{with} \quad \rho^2 = \hat{\rho}^2 + \beta^2.
\]
Then $\tilde{H} = Q_+ \tilde{R}_+$, with
\[
Q_+ = \begin{bmatrix} Q & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_{k-1} & 0 \\ 0 & G^T \end{bmatrix} = \begin{bmatrix} \hat{Q} c & -q s \\ 0 & c & -s s \end{bmatrix}, \quad \text{where} \; Q = [\hat{Q} q],
\]
and with
\[
\tilde{R}_+ = \begin{bmatrix} \hat{R} & r \\ 0 & \rho \\ 0 & 0 \end{bmatrix}.
\]
Now, let $g = Q^T e_1$ (here, $e_1 \in \mathbb{R}^k$) and let $g^T = (\hat{g}^T, \gamma)$. Then
\[
Q^T e_1 = \begin{bmatrix} I_{k-1} & 0 \\ 0 & G^T \end{bmatrix} \begin{bmatrix} \hat{g} \\ \gamma \end{bmatrix} = \begin{bmatrix} \hat{g} c \gamma \\ -s \gamma \end{bmatrix}
\]
(here, $e_1 \in \mathbb{R}^{k+1}$). From this, it is easily seen that
\[
\min_{x=Vy} \| b - Ax \| = \| g_+ - \hat{R}_+ y \| = |s \gamma|.
\]
It follows that
\[
\| b - A x^{(k)} \| = |\gamma_{k+1}| \quad \text{with} \quad \gamma_{k+1} = e_{k+1}^T \hat{Q}_{k+1}^T e_1
\]
and that $|\gamma_{k+1}| = |s_k| |\gamma_k| \leq |\gamma_k|$ for all $k$, i.e., the linear system residuals produced by GMRES are decreasing with $k$. On the next Arnoldi step, the Hessenberg matrix is updated to
\[
H_+ = [\hat{H}, b] = Q_+ [\hat{R}, Q_+^T b] = Q_+ R_+,
\]
and the above process is repeated to get the next iterate. Observe that there is no need to compute $y$ until the residual $\gamma_{k+1} < tol$. The desired relative error
\[
\frac{\| b - A x^{(k)} \|}{\| b \|} < tol
\]
will be achieved at this step and only then will it be necessary to construct $y$ to get $x^{(k)} = Vy$ and halt the iteration.
3.2. The Krylov Subspace:  55

Two additional notable aspects of this updating process have become apparent. First, if the matrix $Q$ is upper Hessenberg then its update $Q_{k+1}$ is also upper Hessenberg and this inductively implies $Q_k$ is upper Hessenberg throughout the process. Second, the update $\rho_k^2 = \rho_k^{2} + \beta_k^2$ gives a direct demonstration that the GMRES iterate is always well defined and unique ($R$ is nonsingular) unless $\beta_k = 0$. The case $\beta_k = 0$ signifies “good” breakdown of the GMRES iteration. It happens if and only if the exact solution is in $K_k(A, b)$.

3.2.6 Symmetry and the Lanczos process

If $A$ is Symmetric, $A = A^T$, then notice that the upper Hessenberg matrix $H_k$ is also Symmetric, for

$$H_k = V_k^T A V_k = V_k^T A^T V_k = H_k^T.$$  

Since $H_k$ is zero in all entries below the first subdiagonal, so too is $H_k^T$; but if those lower-triangular entries of $H_k^T$ are zero, the same must be true for the corresponding upper-triangular entries of $H_k$. Thus $H_k$ is not only upper Hessenberg; it is tridiagonal when $A$ is symmetric. To emphasize this, we put

$$T_k := H_k =: \begin{bmatrix} \alpha_1 & \beta_1 \\ \beta_1 & \alpha_2 & \beta_2 \\ & \beta_2 & \ddots & \ddots \\ & & \ddots & \alpha_{k-1} & \beta_{k-1} \\ & & & \beta_{k-1} & \alpha_k \end{bmatrix}.$$  

Equating the $j$th column on both sides of the relation $A V_k = V_k T_k + f_k e_k^T$ gives the famous three term recurrence

$$A v_j = v_{j-1} \beta_{j-1} + v_j \alpha_j + v_{j+1} \beta_j.$$  

Hence, if the leading $(j - 1) \times (j - 1)$ principal submatrix $T_{j-1}$, as well as the vectors $v_{j-1}$ and $v_j$ and the scalar $\beta_{j-1}$ are known, then $T_j$, $v_{j+1}$ and $\beta_j$ can be computed as follows, exploiting the orthogonality of $v_{j-1}$, $v_j$, and $v_{j+1}$:

$$f = A v_j - v_{j-1} \beta_{j-1} \quad (3.10)$$
$$\alpha_j = v_j^T f \quad (3.11)$$
$$f = f - v_j \alpha_j \quad (3.12)$$
\[ [V, T, f] = \text{Lanczos}(A, b, k) \]

**Input:**
- \( A \) an \( n \times n \) symmetric matrix
- \( b \) a vector of length \( n \)
- \( k \) a positive integer (\( k \leq n \))

**Output:**
- \( V \) an \( n \times k \) matrix with orthonormal columns
- \( T \) a \( k \times k \) symmetric tridiagonal matrix
- \( f \) a vector of length \( n \) with \( AV = VT + f_e^T \) and \( V^T f = 0 \)

\[
\begin{align*}
\mathbf{v}_1 &= b / \| b \| \\
f &= A\mathbf{v}_1, \quad \alpha_1 = \mathbf{v}_1^T f \\
f_1 &\leftarrow f - \mathbf{v}_1\alpha_1 \\
T_1 &\leftarrow [\alpha_1] \\
\text{for } j = 1, 2, 3, \ldots, k - 1, \\
\beta_j &= \| f_j \|, \quad \mathbf{v}_{j+1} \leftarrow f_j / \beta_j \\
f &\leftarrow A\mathbf{v}_{j+1} - \mathbf{v}_j\beta_j \\
\alpha_{j+1} &\leftarrow \mathbf{v}_{j+1}^T w \\
f_{j+1} &\leftarrow f - \mathbf{v}_{j+1}\alpha_{j+1} \\
T_{j+1} &\leftarrow \begin{bmatrix} T_j & \mathbf{e}_j\beta_j \\ \beta_j\mathbf{e}_j^T & \alpha_{j+1} \end{bmatrix} \\
\end{align*}
\]

Figure 3.4. \( k \)-step Lanczos factorization (Symmetric \( A \)).

\[
\begin{align*}
\beta_j &= \| f \| \\
\mathbf{v}_{j+1} &= f / \beta_j \\
T_j &= \begin{bmatrix} T_{j-1} & \mathbf{e}_{j-1}\beta_{j-1} \\
\beta_{j-1}\mathbf{e}_{j-1}^T & \alpha_j \end{bmatrix} .
\end{align*}
\] (3.13) (3.14) (3.15)

This simplified algorithm is described in Figure 3.4.

If implemented in exact arithmetic, this algorithm would construct a matrix \( V_k \) with orthogonal columns; however this orthogonality is quickly lost in finite precision arithmetic. This loss of orthogonality – its causes and implications for the computation of eigenvalues and the solution of linear systems – has motivated considerable study, starting with the pioneering work of Paige in the 1970s; see [?].

D.C. Sorensen – draft – 21 January 2015
3.2. The Krylov Subspace:

3.2.7 Quadratic Functions and Symmetric Linear Systems

It is not surprising that efficient solution methods for solving $Ax = b$ can result from the Lanczos factorization by, for example, specializing the GMRES and FOM methods to take advantage of the limited storage requirements of Lanczos. This is indeed the case and it is certainly possible to proceed formally towards this goal. However, a great deal of geometrical insight is obtained through the intricate relationship between quadratic functions and symmetric linear systems.

Given a symmetric matrix $A$, define the quadratic function $\phi : \mathbb{R}^n \to \mathbb{R}$ as

$$\phi(x) = \frac{1}{2} x^T Ax - b^T x.$$  

There is an immediate relation to the corresponding linear system $Ax = b$ via

$$Ax = b \iff \nabla \phi(x) = 0,$$

i.e., $x$ solves the linear system if and only if it is a critical point of $\phi$.

Now, suppose that $AV = VT + fe_k^T$ and assume $Ve_1 = b$, $\|b\| = 1$. Make the change of variables $x = Vy$ to see that

$$\phi(Vy) = \frac{1}{2} y^T V^T AVy - b^T Vy = \frac{1}{2} y^T Ty - e_1^T y.$$  

The FOM iterate would be obtained by solving $Ty = e_1$ and setting $x = Vy$. Observe that

$$\frac{\partial \phi}{\partial y} = V^T \nabla \phi(Vy) = Ty - e_1.$$  

When $A$ is positive definite ($x \neq 0 \Rightarrow x^T Ax > 0$) the geometry of $\phi$ becomes quite interesting. The graph of $\phi$ is $\{(x, \phi(x)) \} \subset \mathbb{R}^{n+1}$ is a paraboloid that opens up:

To ease discussion, the solution to $Ax = b$ shall be denoted as $x^*$. As indicated by the graph, the point $x^*$ is a global minimizer of the function $\phi$. In fact

**Theorem 3.3.** If $A$ is symmetric and positive definite, then

$$Ax^* = b \iff \phi(x^*) < \phi(x) \text{ for all } x \neq x^*.$$

D.C. Sorensen – draft – 21 January 2015
Proof. Let \( p \in \mathbb{R}^n \) be arbitrary but fixed and let \( \alpha > 0 \). Then

\[
\phi(x_\ast + p\alpha) = \phi(x_\ast) + (Ax_\ast - b)^T p\alpha + \frac{\alpha^2}{2} p^T A p.
\]

If \( \phi(x_\ast) < \phi(x) \) for all \( x \neq x_\ast \), then

\[
0 < \phi(x_\ast + p\alpha) - \phi(x_\ast) = (Ax_\ast - b)^T p\alpha + \frac{\alpha^2}{2} p^T A p
\]

so that

\[
(b - Ax_\ast)^T p < \frac{\alpha}{2} p^T A p, \text{ for all } \alpha > 0.
\]

It follows that

\[
(b - Ax_\ast)^T p \leq 0 \text{ for all } p \in \mathbb{R}^n.
\]

Take \( p = (b - Ax_\ast) \) to see that \( b - Ax_\ast = 0 \) must hold.

Now, suppose \( Ax_\ast = b \). If \( x \neq x_\ast \), let \( p = x - x_\ast \) and let \( \alpha = 1 \). Then

\[
\phi(x) = \phi(x_\ast + p\alpha) = \phi(x_\ast) + \frac{1}{2} p^T A p > \phi(x_\ast)
\]

to conclude the proof. \( \blacksquare \)

3.2.8 The Conjugate Gradient Method: Lanczos + FOM = CG

The Conjugate Gradient Method (CG) for positive definite systems is perhaps the original Krylov iterative method. It was developed independently by Hestenes and Stiefel from a purely optimization perspective. A very clear and convincing derivation stems directly from the Lanczos factorization.

Beginning with

\[
AV_k = V_k T_k + f_k e_k^T
\]

for \( k = 1, 2, \ldots, n \) with \( V_k e_1 = v_1 = b \), it is clear from the previous section that \( T_k y_k = e_1 \) provides the unique solution to

\[
\min_y \frac{1}{2} y^T T_k y - e_1^T y.
\]

Hence \( x^{(k)} = V_k y_k \) solves

\[
\min_{x \in X_k(A, b)} \phi(x) = \min_y \phi(V_k y),
\]

for each \( k = 1, 2, \ldots, n \) and from the theory already developed for FOM this process terminates with the exact solution \( x_\ast \) at some step \( k \leq n \). It is also
3.2. The Krylov Subspace: 59

It is clear that \( x^{(k)} \) is the unique minimizer of \( \phi \) restricted to the Krylov subspace and this property is independent of the choice of basis for \( \mathcal{K}_k(A, b) \).

**A-Conjugate Directions:** Since \( A \) positive definite implies \( T_k \) is positive definite, it has an \( LDL^T \) factorization

\[
T_k = L_k D_k L_k^T
\]

\[
L_k = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
\lambda_1 & 1 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
0 & 0 & \cdots & 1 \\
0 & 0 & \cdots & \lambda_{k-1} \\
\end{bmatrix}, \quad D_k = \begin{bmatrix}
\delta_1 \\
\delta_2 \\
\vdots \\
\delta_{k-1} \\
\delta_k \\
\end{bmatrix}
\]

To get the update

\[
T_{k+1} = L_{k+1} D_{k+1} L_{k+1}^T
\]

simply compute

\[
\lambda_k = \beta_k / \delta_k, \quad \delta_{k+1} = \alpha_{k+1} - \lambda_k \beta_k = \alpha_{k+1} - \lambda_k^2 \delta_k.
\]

Now consider the matrix \( P_k = V_k L_k^{-T} \) and make the change of variables \( y = L_k^{-T} z \). Then

\[
\phi(V_k y) = \phi(P_k z) = \frac{1}{2} z^T D_k z - \hat{b}_k^T z \quad \text{with} \quad \hat{b}_k^T = b^T P_k
\]

since

\[
P_k^T A P_k = L_k^{-1} V_k^T A V_k L_k^{-T} = L_k^{-1} T_k L_k^{-T} = D_k.
\]

The relation \( P_k = V_k L_k^{-T} \) implies \( V_k = P_k L_k^T \). Set \( P_k = [p_1, p_2, \ldots, p_k] \) and recall \( V_k = [v_1, v_2, \ldots, v_k] \). Equating columns gives

\[
p_1 = v_1 \quad \text{and} \quad p_j = v_j - p_{j-1} \lambda_{j-1}, \quad j = 2, 3, \ldots, k.
\]

Thus the directions \( p_j \) are readily computed sequentially from the \( v_j \) as they arrive from the Lanczos process.

**Definition 3.4.** *A set of directions \( p_j \) are called A-conjugate if*

\[
p_j^T A p_j = \begin{cases} 
\delta_j & \text{if } i = j \\
0 & \text{if } i \neq j 
\end{cases}
\]
You might think of A-conjugate directions as “cheap” substitutes for eigenvectors (which are also A-conjugate directions). Observe that in the case \( A \) positive definite \( \delta_j > 0, \ 1 \leq j \leq n \) must hold.

The most important consequence of having a set of A-conjugate directions is that

\[
\phi(P_k z) = \frac{1}{2} z^T D_k z - \hat{b}_k^T z = \sum_{j=1}^{k} \left( \frac{1}{2} \delta_j \zeta_j^2 - \hat{\beta}_j \zeta_j \right)
\]

is separable. Hence

\[
\min_z \phi(P_k z) = \sum_{j=1}^{k} \min_{\zeta_j} \left( \frac{1}{2} \delta_j \zeta_j^2 - \hat{\beta}_j \zeta_j \right)
\]

where \( \zeta_j = z(j) \) the \( j \)-th component of \( z \). Moreover, if \( z_k = \arg \min_z \{ \phi(P_k z) \} \) is known then

\[
\min_z \phi(P_{k+1} z) = \min_{\zeta} \phi(P_k z_k + p_{k+1} \zeta) = \phi(P_k z_k + p_{k+1} \zeta_k + 1).
\]

The key observation to make now is that

\[
r^{(k)} = b - Ax^{(k)} = V_k (e_1 - T_k y_k) + f_k e_k^T y_k = v_k + 1 \beta_k \eta_k,
\]

where \( \eta_k = y_k(k) \). Hence \( \nabla \phi(x^{(k)}) = -r^{(k)} = v_{k+1} \beta_k \eta_k \). Therefore the gradients \( r^{(j)} \) are mutually orthogonal (hence the name “Conjugate Gradients”).

This discussion leads to the following sequence of updates

\[
\begin{align*}
p_{k+1} &= r^{(k)} + p_k \hat{\lambda}_k \\
x^{(k+1)} &= x^{(k)} + p_{k+1} \hat{\zeta}_{k+1} \\
r^{(k+1)} &= b - Ax^{(k+1)} = r^{(k)} - Ap_{k+1} \hat{\zeta}_{k+1}.
\end{align*}
\]

The “hatted” quantities are a result of re-scaling. These numbers can all be worked out using the relations to Lanczos developed above. However, now that the principles have been worked out it is perhaps less tedious to find these scalars using the properties of the quadratic function \( \phi \) and the successive minimizations.

The parameter \( \hat{\lambda}_k \) is determined by forcing the A-conjugacy requirement

\[
0 = p_k^T Ap_{k+1} = p_k^T Ar^{(k)} + p_k^T Ap_k \hat{\lambda}_k
\]
3.2. The Krylov Subspace: 61

\[ [x, r] = \text{CG}(A, b, tol) \]

**Input:**  
- \( A \) an \( n \times n \) symmetric positive definite matrix  
- \( b \) a vector of length \( n \)  
- \( tol \) a stopping tolerance  

**Output:**  
- \( x \) an \( n \) vector such that \( \|b - Ax\| < tol \times \|b\| \)  
- \( r \) a vector of length \( n \) with \( r = b - Ax \)

\[
\begin{align*}
x &= 0, \quad p = b \\
r &= b, \quad \rho = \|r\| \\
\text{while } \rho \geq tol \|b\|, \\
\quad w &= Ap \\
\quad \alpha &= \rho^2 / p^T w \\
\quad x &= x + \rho \alpha \\
\quad r &= r - w \alpha \\
\quad \rho_+ &= \|r\| \\
\quad \beta &= (\rho_+ / \rho)^2 \\
\quad p &= r + \rho \beta \\
\quad \rho &= \rho_+
\end{align*}
\]

Figure 3.5. Conjugate Gradient Algorithm

which gives

\[
\hat{\lambda}_k = -\frac{p_k^T A r^{(k)}}{p_k^T A p_k}.
\]

Now, given \( p_{k+1} \), the formula for \( \hat{\zeta}_{k+1} \) is derived from the one-dimensional minimization step

\[
\hat{\zeta}_{k+1} = \arg \min_{\zeta} \phi(x^{(k)} + p_{k+1} \zeta)
\]

\[
= \arg \min_{\zeta} \{ \phi(x^{(k)}) - r^{(k)^T} p_{k+1} \zeta + \frac{\zeta^2}{2} p_{k+1}^T A p_{k+1} \}
\]

Hence

\[
\hat{\zeta}_{k+1} = \frac{r^{(k)^T} p_{k+1}}{p_{k+1}^T A p_{k+1}}.
\]
There are alternate expressions for these parameters that are more favorable numerically. First, $r^{(k)^T} p_k = 0$ implies that
\[ r^{(k)^T} p_{k+1} = r^{(k)^T} (r^{(k)} + p_k \hat{\lambda}_k) = r^{(k)^T} r^{(k)} \]
so that
\[ \hat{\zeta}_{k+1} = \frac{r^{(k)^T} r^{(k)}}{p_{k+1}^T A p_{k+1}}. \tag{3.16} \]
Moreover, the formula $r^{(k)} = r^{(k-1)} - A p_k \hat{\zeta}_k$ along with the orthogonality of the residual vectors implies that
\[ -\hat{\zeta}_k p_k^T A r^{(k)} = (r^{(k)} - r^{(k-1)})^T r^{(k)} = r^{(k)^T} r^{(k)}. \]
Formula (3.16) implies
\[ \hat{\zeta}_k p_k^T A p_k = r^{(k-1)^T} r^{(k-1)}, \]
and hence
\[ \hat{\lambda}_k = \frac{-\hat{\zeta}_k p_k^T A r^{(k)}}{\hat{\zeta}_k p_k^T A p_k} = \frac{r^{(k)^T} r^{(k)}}{r^{(k-1)^T} r^{(k-1)}}. \tag{3.17} \]

It is now possible to lay out the Conjugate Gradient algorithm. Unfortunately, the traditional indexing and naming of the parameters is in conflict with the development here. The indices of the $p$-vectors starts at $j = 0$ with the $\zeta$’s called $\alpha$’s and the $\lambda$’s called $\beta$’s. However, the recurrences and corresponding parameter formulas persist. The resulting CG algorithm is presented in Figure (3.5) where it is formulated in an index free manner.

### 3.2.9 The bi-Lanczos process

Historically, the Lanczos algorithm for Symmetric matrices \[?\] preceded the Arnoldi method \[?\], hence Arnoldi should properly be viewed as a generalization of Lanczos to non-Symmetric matrices. The Lanczos algorithm builds an orthonormal basis for the Krylov subspace using a three-term recurrence (hence the tridiagonal matrix, $T_k$). The Arnoldi generalization preserves the orthonormal basis, but sacrifices the short recurrence and tridiagonality of $T_k$ for a long recurrence and $H_k$ that is only upper Hessenberg. For large problems, this sacrifice can incur enormous costs in both storage and arithmetic.

An alternative generalization of Lanczos is also possible: retain the three-term recurrence and tridiagonality, but give up the orthogonality of the basis
3.2. The Krylov Subspace: 63

{v_1, \ldots, v_k}. Instead, one also develops a set of basis vectors \{w_1, \ldots, w_k\} for \mathcal{K}_k(A^T, c^T) that is bi-orthogonal to the basis for \mathcal{K}_k(A, b), i.e.,

\[ v_j^T w_\ell = \begin{cases} 0, & j \neq \ell; \\ 1, & j = \ell. \end{cases} \]

Arranging these bases in matrices,

\[ V_k := [v_1 \cdots v_k], \quad W_k := [w_1 \cdots w_k], \]

the bi-orthogonality is summarized as \[W_k^T V_k = I]. Construction of this basis through the bi-Lanczos algorithm produces as \(k\)-step factorization

\[ AV_k = V_k T_k + f_k e_k^T \]
\[ A^T W_k = W_k T_k^T + g_k e_k^T \]

with

\[ W_k^T V_k = I, \quad W_k^T f_k = 0, \quad V_k^T g_k = 0. \]

The non-Symmetric tridiagonal matrix \(T_k\) takes the form

\[
T_k := H_k =: \begin{bmatrix}
\alpha_1 & \gamma_1 & & \\
\beta_1 & \alpha_2 & \gamma_2 & \\
& \beta_2 & \ddots & \\
& & \ddots & \alpha_{k-1} & \gamma_{k-1} \\
& & & \beta_{k-1} & \alpha_k
\end{bmatrix}.
\]

Now there are two three-term recurrences:

\[
A v_j = v_{j-1} \gamma_{j-1} + v_j \alpha_j + v_{j+1} \beta_j,
\]
\[
A^T w_j = w_{j-1} \beta_{j-1} + w_j \alpha_j + w_{j+1} \gamma_j
\]

with

\[ f_j = v_{j+1} \beta_j \quad \text{and} \quad g_j = w_{j+1} \gamma_j. \]

The scalars \(\gamma_j\) and \(\beta_j\) are determined by forcing the condition \(w_{j+1}^T v_{j+1} = 1\). This will require

\[ \gamma_{j+1} \beta_{j+1} = g_j^T f_j. \]

There are, in general, infinitely many ways to choose \(\gamma_{j+1}\) and \(\beta_{j+1}\). Here we take

\[ \beta_{j+1} := \sqrt{|g_j^T f_j|}, \quad \gamma_{j+1} := g_j^T f_j / \beta_{j+1}. \]
\[ [V, W, T, f] = \text{bi-Lanczos}(A, b, k) \]

**Input:**
- \( A \) an \( n \times n \) matrix
- \( b, c \) column vectors of length \( n \)
- \( k \) a positive integer (\( k \leq n \))

**Output:**
- \( V \) an \( n \times k \) matrix
- \( W \) an \( n \times k \) matrix
- \( T \) a \( k \times k \) tridiagonal matrix
- \( f, g \) vectors of length \( n \)

with \( AV = VT + fe^T_k \)
and \( W^T V = I, W^T f = 0, V^T g = 0 \)

\[
v_1 \leftarrow b / \sqrt{|b^T c|}, \quad w_1 \leftarrow c / (\text{sign}(b^T c) \sqrt{b^T c})
\]

\[
f \leftarrow Av_1, \quad \alpha_1 \leftarrow w_1^T f
\]

\[
f_1 \leftarrow f - v_1 \alpha_1
\]

\[
g_1 \leftarrow A^T w_1 - w_1 \alpha_1
\]

\[
T_1 \leftarrow [\alpha_1]
\]

for \( j = 1, 2, \ldots, k - 1, \)

\[
\beta_j \leftarrow \sqrt{|g_j^T f_j|}, \quad \gamma_j \leftarrow g_j^T f_j / \beta_{j+1}
\]

\[
v_{j+1} \leftarrow f_j / \beta_j, \quad w_{j+1} \leftarrow g_j / \gamma_j
\]

\[
f \leftarrow Av_{j+1} - v_j \gamma_j
\]

\[
g \leftarrow A^T w_{j+1} - w_j \beta_j
\]

\[
\alpha_{j+1} \leftarrow v_{j+1}^T f
\]

\[
f_{j+1} \leftarrow f - v_{j+1} \alpha_{j+1}
\]

\[
g_{j+1} \leftarrow g - w_{j+1} \alpha_{j+1}
\]

\[
T_{j+1} \leftarrow \begin{bmatrix} T_j & e_j \gamma_j \\ \beta_j e_j^T & \alpha_{j+1} \end{bmatrix}
\]

end

**Figure 3.6.** \( k \)-step bi-Lanczos factorization

If \( f = 0 \) or \( g = 0 \), the bi-Lanczos algorithm cannot be continued.\(^2\) However, this formula suggests numerical and algorithmic difficulties are likely, for it is indeed possible that \( g_j^T f_j = 0 \) while neither \( f_j \) nor \( g_j \) are zero. Indeed,

\(^2\) If one is using the bi-Lanczos method as an iterative method to solve \( Ax = b \), the case \( f = 0 \) is no problem: it implies that \( x = Vy \) with \( Ty = e_1 \) would provide the exact solution.
3.2. The Krylov Subspace: 

Exact breakdown can occur even when \((A, b)\) is controllable and \((A, c)\) is observable. (In contrast, the Arnoldi process can breakdown for \(k < n\) only when \((A, b)\) is not controllable.) Assuming breakdown does not occur, the bi-Lanczos algorithm will terminate in \(n\) steps with \(AV_n = V_n T_n\) and \(A^T W_n = W_n T_n^*\). The algorithm is summarized in Figure 3.6.

Unfortunately, in floating point arithmetic the bi-Lanczos algorithm is completely unstable. The potential occurrence of breakdown stems from some complicated dependencies. The malady is closely tied to the occurrence of singular principal submatrices of \(A\), and is associated with the instabilities that arise when attempting to perform Gaussian elimination on a general matrix \(A\) without pivoting. Viable practical remedies for breakdown have been proposed (see, e.g., [?]) that usually work well in practice. However, for any such strategy there is a matrix \(A\) and vectors \(v_1\) and \(w_1\) that will defeat it.
References