We now write the vector $u(t)$ as $u(t) = Q(t)x$ for an arbitrary vector x,

$$\begin{aligned}(I-Q)Q(t)x &= S^{+}(\lambda)\left[-t(I-Q)EQ(t)x + \right. \\ &\qquad \left. (\lambda(t)-\lambda)(I-Q)Q(t)x\right].\end{aligned}$$

The above equation yields an estimate of the norm of $(I-Q)Q(t)$, which is the sine of the angle between the invariant subspaces $M = \mathrm{Ran}(Q)$ and $M(t) = \mathrm{Ran}(Q(t))$.

Proposition 3.5 *Assume that λ is a simple eigenvalue of A. When the matrix A is perturbed by the matrix tE, then the sine of the angle between the invariant subspaces M and $M(t)$ of A and $A+tE$ associated with the eigenvalues λ and $\lambda(t)$ is approximately,*

$$\sin\theta(M, M(t)) \approx |t| \|S^{+}(\lambda)(I-Q)EQ(t)\|$$

the approximation being of second order with respect to t.

Thus, we can define the condition number for invariant subspaces as being the (spectral) norm of $S^{+}(\lambda)$.

The more interesting situation is when the invariant subspace is associated with a multiple eigenvalue. What was just done for one-dimensional invariant subspaces can be generalized to multiple-dimensional invariant subspaces. The notion of condition numbers here will require some knowledge about generalized solutions to Sylvester's equations. A Sylvester equation is a matrix equation of the form

$$AX - XR = B \tag{3.46}$$

where A is $n \times n$, X and B are $n \times r$ and R is $r \times r$. The important observation which we would like to exploit is that (3.46) is nothing but a linear system of equations with $n\ r$ unknowns. It can be shown that the mapping $X \rightarrow AX - XR$ is invertible under the simple condition that the spectra of A and R have no point in common.

We now proceed in a similar manner as for simple eigenvalues and write,

$$AU = UR$$
$$(A + tE)U(t) = U(t)R(t)$$

in which U and $U(t)$ are $n \times r$ unitary matrices and R and $R(t)$ are $r \times r$ upper triangular. Subtracting $U(t)R$ from the second equation we obtain

$$AU(t) - U(t)R = -tEU(t) + U(t)(R(t) - R)$$

Multiplying both sides by $I - Q$ and using again the relation (3.45),

$$\begin{aligned}(I - Q)A(I - Q)U(t) &- (I - Q)U(t)R \\ &= (I - Q)[-tEU(t) + U(t)(R(t) - R)]\end{aligned}$$

Observe that the operator

$$X \rightarrow (I - Q)A(I - Q)X - XR$$

is invertible because the eigenvalues of $(I - Q)A(I - Q)$ and those of R form disjoint sets. Therefore, we can define its inverse which we call $S^+(\lambda)$, and we have

$$(I - Q)U(t) = S^+(\lambda)\left[t(I - Q)EU(t) + (I - Q)U(t)(R(t) - R)\right]$$

As a result, up to lower order terms, the sine of the angle between the two subspaces is $|t|\|S^+(\lambda)(I - Q)EU(t)\|$, a result that constitutes a direct generalization of the previous theorem.

4. Localization Theorems

In some situations one wishes to have a rough idea of where the eigenvalues lie in the complex plane, by directly exploiting some knowledge on the entries of the matrix A. We already know a simple localization result that uses any matrix norm, since we have

$$|\lambda_i| \ \leq \ \|A\|$$

i.e., any eigenvalue belongs to the disc centered at the origin and of radius $\|A\|$. A more precise localization result is provided by Gerschgorin's theorem.

Theorem 3.11 (Gerschgorin [58]) *Any eigenvalue λ of a matrix A is located in one of the closed discs of the complex plane centered at a_{ii} and having the radius*

$$\sum_{\substack{j=1 \\ j\neq i}}^{j=n} |a_{ij}| \ .$$

In other words,

$$\forall\ \lambda\ \in \sigma(A), \quad \exists\ i \quad \text{such that} \quad |\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{j=n} |a_{ij}| \ . \qquad (3.47)$$

Proof. The proof is by contradiction. Assume that (3.47) does not hold. Then there is an eigenvalue λ such that for $i = 1, 2, \ldots, n$ we have

$$|\lambda - a_{ii}| > \sum_{j=1, j\neq i}^{j=n} |a_{ij}| \ . \qquad (3.48)$$

We can write $A - \lambda I = D - \lambda I + H$, where $D = \text{diag}\ \{a_{ii}\}$ and H is the matrix obtained from A by replacing its diagonal elements by zeros. Since $D - \lambda$ is invertible we have

$$A - \lambda I = (D - \lambda I)(I + (D - \lambda I)^{-1} H) \ . \qquad (3.49)$$

The elements in row i of the matrix $C = (D - \lambda I)^{-1} H$ are $c_{ij} = a_{ij}/(a_{ii} - \lambda)$ for $j \neq i$ and $c_{ii} = 0$, and so the sum of their moduli are less than unity by (3.48). Hence

$$\rho((D - \lambda I)^{-1} H) \leq \|(D - \lambda I)^{-1} H\|_\infty < 1$$

and as a result the matrix $I + C = (I + (D - \lambda I)^{-1} H)$ is nonsingular. Therefore, from (3.49) $(A - \lambda I)$ would also be nonsingular which is a contradiction. ∎

Since the result also holds for the transpose of A, we can formulate a version of the theorem based on column sums instead of row sums,

$$\forall\, \lambda \in \sigma(A), \quad \exists\, j \quad \text{such that} \quad |\lambda - a_{jj}| \leq \sum_{\substack{i=1 \\ i \neq j}}^{i=n} |a_{ij}| \,. \tag{3.50}$$

The discs defined in the theorem are called Gerschgorin discs. There are n Gerschgorin discs and their union contains the spectrum of A. The above results can be especially useful when the matrix is almost diagonal, as is often the case when an algorithm is used to diagonalize a matrix and the process is nearing convergence. However, in order to better exploit the theorem, we need to show the following additional result.

Theorem 3.12 . *Suppose that there are m Gerschgorin discs whose union S is disjoint from all other discs. Then S contains exactly m eigenvalues, (counted with their multiplicities).*

Proof. Let $A(t) = D + tH$ where $0 \leq t \leq 1$, and D, H are defined in the proof of Gerschgorin's theorem. Initially when $t = 0$ all eigenvalues of $A(t)$ are at the discs of radius 0, centered at a_{ii}. By a continuity argument, as t increases to 1, the branches of eigenvalues $\lambda_i(t)$ will stay in their respective discs as long as these discs stay disjoint. This is because the image of the connected interval [0,1] by $\lambda_i(t)$ must be connected. More generally, if the union of m of the discs are disjoint from the other discs, the union $S(t)$ of the corresponding discs as t varies, will contain m eigenvalues. ∎

An important particular case is that when one disc is disjoint from the others then it must contain exactly one eigenvalue.

There are other ways of estimating the error of a_{ii} regarded as an eigenvalue of A. For example, if we take as approximate eigenvector the i-th column of the identity matrix we get the following result from a direct application of Kato-Temple's theorem in the Hermitian case.

Proposition 3.6 *Let i be any integer between 1 and n and let λ be the eigenvalue of A closest to a_{ii}, and μ the next closest eigenvalue to a_{ii}. Then if we call ϵ_i the 2-norm of the $(n-1)$-vector obtained from the $i-th$ column of A by deleting the entry a_{ii} we have*

$$|\lambda - a_{ii}| \leq \frac{\epsilon_i^2}{|\mu - a_{ii}|} .$$

Proof. The proof is a direct application of Kato-Temple's theorem. ∎

Thus, in the Hermitian case, the Gerschgorin bounds are not tight in general since the error is of the order of the square of the vector of the off-diagonal elements in a row (or column), whereas Gerschgorin's result will provide an error estimate of the same order as the 1-norm of the same vector (in the ideal situation when the discs are disjoint). However, we note that the isolated application of the above proposition in practice may not be too useful since we may not have an estimate of $|\mu - a_{ii}|$. A simpler, though less powerful, bound is $|\lambda - a_{ii}| \leq \epsilon_i$. These types of results are quite different in nature from those of Gerschgorin's theorem. They simply tell us how accurate an approximation a diagonal element can be when regarded as an approximate eigenvalue. It is an isolated result and does not tell us anything on the other eigenvalues. Gerschgorin's result on the other hand is a global result, in that it tells where *all* the eigenvalues are located, as a group. This distinction between the two types of results, namely the (local) a-posteriori error bounds on the one hand, and the global localizations results such as Gerschgorin's theorem on the other, is often misunderstood.

PROBLEMS

P-3.1 If P is a projector onto M along S then P^H is a projector onto $S^\perp$ along $M^\perp$. [Hint: see proof of Proposition 3.1].

P-3.2 Show that for two orthogonal bases V_1, V_2 of the same subspace M of $\mathbb{C}^n$ we have $V_1 V_1^H x = V_2 V_2^H x \;\; \forall x$.

P-3.3 What are the eigenvalues of a projector? What about its eigenvectors?

P-3.4 Let P be a projector and $V = [v_1, v_2, \cdots, v_m]$ a basis of $\mathrm{Ran}(P)$. Why does there always exist a basis $W = [w_1, w_2, \cdots, w_m]$ of $L = \mathrm{Ker}(P)^\perp$ such that the two sets form a biorthogonal basis? In general given two subspaces M and S of the same dimension m, is there always a biorthogonal pair V, W such that V is a basis of M and W a basis of S?

P-3.5 Let P be a projector, $V = [v_1, v_2, \cdots, v_m]$ a basis of $\mathrm{Ran}(P)$, and U a matrix the columns of which form a basis of $\mathrm{Ker}(P)$. Show that the system U, V forms a basis of $\mathbb{C}^n$. What is the matrix representation of P with respect to this basis?

P-3.6 Show that if two projectors P_1 and P_2 commute then their product $P = P_1 P_2$ is a projector. What are the range and kernel of P?

P-3.7 Consider the matrix seen in Example 3.6. We perturb the term a_{33} to -25.01. Give an estimate in the changes of the eigenvalues of the matrix. Use any FORTRAN library or interactive tool to compute the eigenvectors/ eigenvalues of the perturbed matrix.

P-3.8 Let

$$\delta(X, Y) \equiv \max_{x \,\in\, X, \|x\|_2 = 1} \mathrm{dist}(u, Y).$$

Show that

$$\omega(M_1, M_2) = \max\{\delta(M_1, M_2), \delta(M_2, M_1)\} \ .$$

P-3.9 Given two subspaces M and S with two orthogonal bases V and W show that the singular values of $V^H W$ are between zero and one. The canonical angles between M and S are defined as the acutes angles whose cosines are the singular values σ_i, i.e., $\cos\theta_i = \sigma_i(V^H W)$. The angles are labeled in descending order. Show that this definition does not depend on the order of the pair M, S (in other words that the singular values of $W^H V$ are identical with those of $V^H W$).

P-3.10 Show that the largest canonical angle between two subspaces (see previous problem) is $\pi/2$ iff the intersection of M and the orthogonal of S is not reduced to $\{0\}$.

P-3.11 Let P_1, P_2 be two orthogonal projectors with ranges M_1 and M_2 respectively of the same dimension $m \le n/2$ and let $V_i, i = 1, 2$ be an orthogonal basis of $M_i, i = 1, 2$. Assuming at first that the the columns of the system $[V_1, V_2]$ are linearly independent what is the matrix representation of the projector $P_1 - P_2$ with respect to the basis obtained by completing V_1, V_2 into a basis of $\mathbb{C}^n$? Deduce that the eigenvalues of $P_1 - P_2$ are $\pm\sin\theta_i$, where the θ_i's are the canonical angles between M_1 and M_2 as defined in the previous problems. How can one generalize this result to the case where the columns of $[V_1, V_2]$ are not linearly independent?

P-3.12 Use the previous result to show that

$$\omega(M_1, M_2) = \sin\theta_{max}$$

where θ_{max} is the largest canonical angle between the two subspaces.

P-3.13 Prove the second equality in equation (3.32) of the proof of Theorem 3.10.

P-3.14 Let $E = xp^H + yq^H$ where $x \perp y$ and $p \perp q$. What is the 2-norm of E? [Hint: Compute $E^H E$ and then find the singular values of E.]

P-3.15 Show that the condition number of an eigenvalue λ of a matrix A does not change if A is transformed by an orthogonal similarity transformation. Is this true for any similarity transformation? What can be said of the condition number of the corresponding eigenvector?

P-3.16 Consider the matrix obtained from that of example 3.7 in which the elements -1 above the diagonal are replaced by $-\alpha$, where α is a constant. Find bounds similar to those in Example 3.7 for the condition number of the eigenvalue λ_1 of this matrix.

P-3.17 Under the same assumptions as those of Theorem 3.6, establish the improved error

$$\sin\theta(\tilde{u}, u) \leq \sqrt{\frac{\|r\|_2^2 - \epsilon^2}{\delta^2 - \epsilon^2}}$$

in which $\epsilon \equiv |\lambda - \tilde{\lambda}|$. [Hint: Follow proof of theorem 3.6]

NOTES AND REFERENCES. Some of the material in this chapter is based on [85] and [14]. A broader and more detailed view of perturbation analysis for matrix problems is the recent book by Stewart and Sun [172]. The treatment of the equivalence between the projectors as defined from the Jordan canonical form and the one defined from the Dunford integral seems to be new. The results of Section 2.3 are simpler versions of those found in [82], which should be consulted for more detail. The notion of condition number for eigenvalue problems is discussed in detail in Wilkinson [183] who seems to be at the origin of the notion of condition numbers for eigenvalues and eigenvectors. ♠

Chapter IV

The Tools of Spectral Approximation

Many of the algorithms used to approximate spectra of large matrices consist of a blend of a few basic mathematical or algorithmic tools, such as projection methods, Chebyshev acceleration, deflation, shift-and-invert strategies, to name just a few. We have grouped together these tools and techniques in this chapter. We start with some background on well-known procedures based on single vector iterations. These have historically provided the starting point of many of the more powerful methods. Once an eigenvalue-eigenvector pair is computed by one of the single vector iterations, it is often desired to extract another pair. This is done with the help of a standard technique known as *deflation* which we discuss in some detail. Finally, we will present the common projection techniques which constitute perhaps the most important of the basic techniques used in approximating eigenvalues and eigenvectors.

1. Single Vector Iterations

One of the oldest techniques for solving eigenvalue problems is the so-called power method. Simply described this method consists of generating the sequence of vectors $A^k v_0$ where v_0 is some nonzero initial vector. A few variants of the power method have been developed which consist of iterating with a few simple functions of A. These methods involve a single sequence of vectors and we describe some of them in this section.

1.1. The Power Method

The simplest of the single vector iteration techniques consists of generating the sequence of vectors $A^k v_0$ where v_0 is some nonzero initial vector. This sequence of vectors when normalized appropriately, and under reasonably mild conditions, converges to a dominant eigenvector, i.e., an eigenvector associated with the eigenvalue of largest modulus. The most commonly used normalization is to ensure that the largest component of the current iterate is equal to one. This yields the following algorithm.

ALGORITHM **4.1 (The Power Method.)**

- *1. Start: Choose a nonzero initial vector v_0.*
- *2. Iterate: for $k = 1, 2, \ldots$ until convergence, compute*

$$v_k = \frac{1}{\alpha_k} A v_{k-1}$$

  where α_k is a component of the vector Av_{k-1} which has the maximum modulus.

The following theorem establishes a convergence result for the above algorithm.

Theorem 4.1 *Assume that there is one and only one eigenvalue λ_1 of A of largest modulus and that λ_1 is semi-simple. Then either the initial vector v_0 has no component in the invariant subspace associated with λ_1 or the sequence of vectors generated by Algorithm 4.1 converges to an eigenvector associated with λ_1 and α_k converges to λ_1.*

Proof. Clearly, v_k is nothing but the vector $A^k v_0$ normalized by a certain scalar $\hat{\alpha}_k$ in such a way that its largest component is unity. Let us decompose the initial vector v_0 as

$$v_0 = \sum_{i=1}^{p} P_i v_0 \tag{4.1}$$

where the P_i's are the spectral projectors associated with the distinct eigenvalues $\lambda_i, i = 1, \ldots, p$. Recall from (1.19) of Chapter 1, that $AP_i = P_i(\lambda_i P_i + D_i)$ where D_i is a nilpotent of index l_i, and more generally, by induction we have $A^k P_i = P_i(\lambda_i P_i + D_i)^k$. As a result we obtain,

$$v_k = \frac{1}{\hat{\alpha}_k} A^k \sum_{i=1}^{p} P_i v_0 = \frac{1}{\hat{\alpha}_k} \sum_{i=1}^{p} A^k P_i v_0 = \frac{1}{\hat{\alpha}_k} \sum_{i=1}^{p} P_i (\lambda_i I + D_i)^k v_0 \, .$$

Hence, noting that $D_1 = 0$ because λ_1 is semi-simple,

$$\begin{aligned} v_k &= \frac{1}{\hat{\alpha}_k} \sum_{i=1}^{p} P_i (\lambda_i P_i + D_i)^k v_0 \\ &= \frac{1}{\hat{\alpha}_k} \left(\lambda_1^k P_1 v_0 + \sum_{i=2}^{p} P_i (\lambda_i P_i + D_i)^k v_0 \right) \\ &= \frac{\lambda_1^k}{\hat{\alpha}_k} \left(P_1 v_0 + \sum_{i=2}^{p} \frac{1}{\lambda_1^k} (\lambda_i P_i + D_i)^k P_i v_0 \, . \right) \end{aligned} \tag{4.2}$$

The spectral radius of each operator $(\lambda_i P_i + D_i)/\lambda_1$ is less than one since $|\lambda_i/\lambda_1| < 1$ and therefore, its k-th power will converge to zero. If $P_1 v_0 = 0$ the theorem is true. Assume that $P_1 v_0 \neq 0$. Then it follows immediately from (4.2) that v_k converges to

$P_1 v_0$ normalized so that its largest component is one. That α_k converges to the eigenvalue λ_1 is an immediate consequence of the relation $Av_{k-1} = \alpha_k v_k$ and the fact the sequence of vectors v_k converges. ■

The proof suggests that the convergence factor of the method is given by

$$\rho_D = \frac{|\lambda_2|}{|\lambda_1|}$$

where λ_2 is the second largest eigenvalue in modulus. This ratio represents the spectral radius of the linear operator $\frac{1}{\lambda_1}A$ restricted to the subspace that excludes the invariant subspace associated with the dominant eigenvalue. It is a common situation that the eigenvalues λ_1 and λ_2 are very close from one another. As a result convergence may be extremely slow.

Example 4.1 Consider the Markov Chain matrix Mark(10) which has been described in Chapter 2. This is a matrix of size $n = 55$ which has two dominant eigenvalues of equal modulus namely $\lambda = 1$ and $\lambda = -1$. As is to be expected the power method applied directly to A does not converge. To obtain convergence we can for example consider the matrix $I + A$ whose eigenvalues are those of A shifted to the right by one. The eigenvalue $\lambda = 1$ is then transformed into the eigenvalue $\lambda = 2$ which now becomes the (only) dominant eigenvalue. The algorithm then converges and the convergence history is shown in Table 4.1. In the first column of the table we show the iteration number. The results are shown only every 20 steps and at the very last step when convergence has taken place. The second column shows the 2-norm of the difference between two successive iterates, i.e., $\|x_{i+1} - x_i\|_2$ at teration i, while the third column shows the residual norm $\|Ax - \mu(x)x\|_2$, in which $\mu(x)$ is the Rayleigh quotient of x and x is normalized to have a 2-norm unity. The algorithm is stopped as soon at the 2-norm of the difference between two successive iterates becomes less than $\epsilon = 10^{-7}$. Finally, the last column shows the corresponding eigenvalue estimates. Note that what is shown is simply the coefficient α_k, shifted by -1 to get an approximation to the eigenvalue of Mark(10) instead of Mark(10)$+I$. The initial vector in the iteration is the vector $x_0 = (1, 1, \ldots, 1)^T$.

Iteration	Norm of diff.	Res. norm	Eigenvalue
20	0.639D-01	0.276D-01	1.02591636
40	0.129D-01	0.513D-02	1.00680780
60	0.192D-02	0.808D-03	1.00102145
80	0.280D-03	0.121D-03	1.00014720
100	0.400D-04	0.174D-04	1.00002078
120	0.562D-05	0.247D-05	1.00000289
140	0.781D-06	0.344D-06	1.00000040
161	0.973D-07	0.430D-07	1.00000005

Table 4.1 Power iteration with $A = \text{Mark}(10) + I$.

If the eigenvalue is multiple, but semi-simple, then the algorithm provides only one eigenvalue and a corresponding eigenvector. A more serious difficulty is that the algorithm will not converge if the dominant eigenvalue is complex and the original matrix as well as the initial vector are real. This is because for real matrices the complex eigenvalues come in complex pairs and as result there will be (at least) two distinct eigenvalues that will have the largest modulus in the spectrum. Then the theorem will not guarantee convergence. There are remedies to all these difficulties and some of these will be examined later.

1.2. The Shifted Power Method

In Example 4.1 we have been lead to use the power method not on the original matrix but on the *shifted* matrix $A + I$. One observation is that we could also have iterated with a matrix of the form $B(\sigma) = A + \sigma I$ for any positive σ and the choice $\sigma = 1$ is a rather arbitrary choice. There are better choices of the shift as is suggested by the following example.

Example 4.2 Consider the same matrix as in the previous example, in which the shift σ is replaced by $\sigma = 0.1$. The new convergence history is shown in Table 4.2, and indicates a much faster convergence than before.

Iteration	Norm of diff.	Res. Norm	Eigenvalue
20	0.273D-01	0.794D-02	1.00524001
40	0.729D-03	0.210D-03	1.00016755
60	0.183D-04	0.509D-05	1.00000446
80	0.437D-06	0.118D-06	1.00000011
88	0.971D-07	0.261D-07	1.00000002

Table 4.2 Power iteration on $A = Mark(10) + 0.1 \times I$.

More generally, when the eigenvalues are real it is not too difficult to find the optimal value of σ, i.e., the shift that maximizes the asymptotic convergence rate, see Problem P-4.5. The scalars σ are called *shifts of origin*. The important property that is used is that shifting does not alter the eigenvectors and that it does change the eigenvalues in a simple known way, it shifts them by σ.

1.3. Inverse Iteration

The inverse power method, or inverse iteration, consists simply of iterating with the matrix A^{-1} instead of the original matrix A. In other words, the general iterate v_k is defined by

$$v_k = \frac{1}{\alpha_k} A^{-1} v_{k-1} \ . \tag{4.3}$$

Fortunately it is not necessary to compute the matrix A^{-1} explicitly as this could be rather expensive for large problems. Instead, all that is needed is to carry out the LU factorization of A prior to starting the vector iteration itself. Subsequently, one must solve an upper and lower triangular system at each step. The vector v_k will now converge to the eigenvector associated with the dominant eigenvalue of A^{-1}. Since the eigenvalues of A and A^{-1} are the inverses of each other while their eigenvectors are identical, the iterates will converges to the eigenvector of A associated with the eigenvalue of smallest modulus. This may or may not be what is desired but in practice the method is often combined with shifts of origin. Indeed, a more common problem in practice is to compute the eigenvalue of A that is closest to a certain scalar σ and the corresponding eigenvector. This is achieved by iterating with the matrix $(A - \sigma I)^{-1}$. Often, σ is referred to as the *shift*. The corresponding algorithm is as follows.

ALGORITHM **4.2 : Inverse Power Method**

1. **Start:** *Compute the LU decomposition $A - \sigma I = LU$ and choose an initial vector v_0.*

2. **Iterate:** *for $k = 1, 2, \ldots,$ until convergence compute*

$$v_k = \frac{1}{\alpha_k}(A - \sigma I)^{-1} v_{k-1} = \frac{1}{\alpha_k} U^{-1} L^{-1} v_{k-1} \tag{4.4}$$

where α_k is a component of the vector $(A-\sigma I)^{-1} v_{k-1}$ which has the maximum modulus.

Note that each of the computations of $y = L^{-1} v_{k-1}$ and then $v = U^{-1} y$ can be performed by a forward and a backward triangular system solve, each of which costs only $O(n^2/2)$ operations when the matrix is dense. The factorization in step 1 is much more expensive whether the matrix is dense or sparse.

If λ_1 is the eigenvalue closest to σ then the eigenvalue of largest modulus of $(A-\sigma I)^{-1}$ will be $1/(\lambda_1 - \sigma)$ and so α_k will converge to this value. An important consideration that makes Algorithm 4.2 quite attractive is its potentially high convergence rate. If λ_1 is the eigenvalue of A closest to the shift σ and λ_2 is the next closet one then the convergence factor is given by

$$\rho_I = \frac{|\lambda_1 - \sigma|}{|\lambda_2 - \sigma|} \tag{4.5}$$

which indicates that the convergence can be very fast if σ is much closer to the desired eigenvalue λ_1 than it is to λ_2.

From the above observations, one can think of changing the shift σ occasionally into a value that is known to be a better approximation of λ_1 than the previous σ. For example, one can replace occasionally σ by the estimated eigenvalue of A that is derived from the information that α_k converges to $1/(\lambda_1 - \sigma)$, i.e., we can take

$$\sigma_{new} = \sigma_{old} + \frac{1}{\alpha_k}.$$

Strategies of this sort are often referred to as shift-and-invert techniques.

Another possibility, which may be very efficient in the Hermitian case, is to take the new shift to be the Rayleigh quotient of the latest approximate eigenvector v_k. One must remember however, that the LU factorization is expensive so it is desirable to keep such shift changes to a minimum. At one extreme where the shift is never changed, we obtain the simple inverse power method represented by Algorithm 4.2. At the other extreme, one can also change the shift at every step. The algorithm corresponding to this case is called Rayleigh Quotient Iteration (RQI) and has been extensively studied for Hermitian matrices.

ALGORITHM **4.3 Rayleigh Quotient Iteration**

1. **Start:** *Choose an initial vector* v_0 *such that* $\|v_0\|_2 = 1$.

2. **Iterate:** *for* $k = 1, 2, \ldots,$ *until convergence compute*

$$\begin{aligned} \sigma_k &= (Av_{k-1}, v_{k-1}) \ , \\ v_k &= \frac{1}{\alpha_k}(A - \sigma_k I)^{-1} v_{k-1}, \end{aligned}$$

where α_k *is chosen so that the 2-norm of the vector* v_k *is one.*

It is known that this process is globally convergent for Hermitian matrices, in the sense that α_k converges and the vector v_k either converges to an eigenvector or alternates between two eigenvectors. Moreover, in the first case α_k converges cubically towards an eigenvalue, see Parlett [118]. In the case where v_k oscillates, between two eigenvectors, then α_k converges towards the mid-point of the corresponding eigenvalues. In the non-Hermitian case, the convergence can be at most quadratic and there are no known global convergence results except in the normal case. This algorithm is not much used in practice despite these nice properties, because of the high cost of the frequent factorizations.

2. Deflation Techniques

Suppose that we have computed the eigenvalue λ_1 of largest modulus and its corresponding eigenvector u_1 by some simple algorithm, say algorithm (A), which always delivers the eigenvalue of largest modulus of the input matrix, along with an eigenvector. For example, algorithm (A) can simply be one of the single vector iterations described in the previous section. It is assumed that the vector u_1 is normalized so that $\|u_1\|_2 = 1$. The problem is to compute the next eigenvalue λ_2 of A. An old technique for achieving this is what is commonly called a deflation procedure. Typically, a rank one modification is applied to the original matrix so as to displace the eigenvalue λ_1, while keeping all other eigenvalues unchanged. The rank one modification is chosen so that the eigenvalue λ_2 becomes the one with largest modulus of the modified matrix and therefore, algorithm (A) can now be applied to the new matrix to extract the pair λ_2, u_2.

2.1. Wielandt Deflation with One Vector

In the general procedure known as Wielandt's deflation only the knowledge of the right eigenvector is required. The deflated matrix is of the form

$$A_1 = A - \sigma u_1 v^H \tag{4.6}$$

where v is an arbitrary vector such that $v^H u_1 = 1$, and σ is an appropriate shift. It can be shown that the eigenvalues of A_1

are the same as those of A except for the eigenvalue λ_1 which is transformed into the eigenvalue $\lambda_1 - \sigma$.

Theorem 4.2 (Wielandt) *The spectrum of A_1 as defined by (4.6) is given by*

$$\sigma(A_1) = \{\lambda_1 - \sigma, \lambda_2, \lambda_3, \ldots, \lambda_p\} .$$

Proof. For $i \neq 1$ the left eigenvectors of A satisfy

$$(A^H - \bar{\sigma} v u_1^H) w_i = \lambda_i w_i$$

because w_i is orthogonal to u_1. On the other hand for $i = 1$, we have $A_1 u_1 = (\lambda_1 - \sigma) u_1$. ■

The above proof reveals that the left eigenvectors $w_2, \ldots, w_p$ are preserved by the deflation process. Similarly, the right eigenvector u_1 is preserved. It is also important to see what becomes of the other right eigenvectors. For each i, we seek a right eigenvector of A_1 in the form of $\hat{u}_i = u_i - \gamma_i u_1$. We have,

$$\begin{aligned} A_1 \hat{u}_i &= (A - \sigma u_1 v^H)(u_i - \gamma_i u_1) \\ &= \lambda_i u_i - [\gamma_i \lambda_1 + \sigma v^H u_i - \sigma \gamma_i] u_1. \end{aligned} \tag{4.7}$$

Taking $\gamma_1 = 0$ shows, as is already indicated by the proposition, that any eigenvector associated with the eigenvalue λ_1 remains an eigenvector of A_1, associated with the eigenvalue $\lambda_1 - \sigma$. For $i \neq 1$, it is possible to select γ_i so that the vector $\hat{u}_i$ is an eigenvector of A_1 associated with the eigenvalue λ_i,

$$\gamma_i(v) \equiv \frac{v^H u_i}{1 - (\lambda_1 - \lambda_i)/\sigma} . \tag{4.8}$$

Observe that the above expression is not defined when the denominator vanishes. However, it is known in this case that the eigenvalue $\lambda_i = \lambda_1 - \sigma$ is already an eigenvalue of A_1, i.e., the eigenvalue $\lambda_1 - \sigma$ becomes multiple, and we only know one eigenvector namely u_1.

There are infinitely many different ways of choosing the vector v. One of the most common choices is to take $v = w_1$ the left eigenvector. This is referred to as Hotelling's deflation. It has the advantage of preserving both the left and right eigenvectors of A as is seen from the fact that $\gamma_i = 0$ in this situation. Another simple choice is to take $v = u_1$. In the next section we will consider these different possibilities and try to make a rational choice between them.

Example 4.3 As a test we consider again the matrix Mark(10) seen is Example 4.1. For u_1 we use the vector computed from the shifted power method with shift 0.1. If we take v to be a random vector and x_0 to be a random vector, then the algorithm converges in 135 steps and yields $\lambda_2 \approx 0.93715016$. The stopping criterion is identical with the one used in Example 4.1. If we take $v = u_1$ or $v = (1, 1, \ldots, 1)^T$, then the algorithm converges in 127 steps.

2.2. Optimality in Wieldant's Deflation

An interesting question that we wish to answer is: among all the possible choices of v, which one is likely to yield the best possible condition number for the next eigenvalue λ_2 to be computed? This is certainly a desirable goal in practice. We will distinguish the eigenvalues and eigenvectors associated with the matrix A_1 from those of A by denoting them with a tilde. The condition number of the next eigenvalue $\tilde{\lambda}_2$ to be computed is, by definition,

$$\mathrm{Cond}(\tilde{\lambda}_2) = \frac{\|\tilde{u}_2\|_2 \|\tilde{w}_2\|_2}{|(\tilde{u}_2, \tilde{w}_2)|}$$

where $\tilde{u}_2, \tilde{w}_2$ are the right and left eigenvectors of A_1 associated with the eigenvalue $\tilde{\lambda}_2$. From what we have seen before, we know that $\tilde{w}_2 = w_2$ while $\tilde{u}_2 = u_2 - \gamma_2(v) u_1$ where $\gamma_2(v)$ is given by (4.8). Assuming that $\|w_2\|_2 = 1$ we get,

$$\mathrm{Cond}(\tilde{\lambda}_2) = \frac{\|u_2 - \gamma_2(v) u_1\|_2}{|(u_2, w_2)|} \tag{4.9}$$

where we have used the fact that $(u_1, w_2) = 0$. It is then clear from (4.9) that the condition number of λ_2 is minimized whenever

$$\gamma_2(v) = u_1^H u_2 \equiv \cos\theta(u_1, u_2) \ . \tag{4.10}$$

Substituting this result in (4.8) we obtain the equivalent condition

$$v^H u_2 \ = \left(1 - \frac{\lambda_1 - \lambda_2}{\sigma}\right) \ u_1^H u_2 \ , \tag{4.11}$$

to which we add the normalization condition,

$$v^H u_1 \ = \ 1. \tag{4.12}$$

There are still infinitely many vectors v that satisfy the above two conditions. However, we can seek a vector v which is spanned by two specific vectors. There are two natural possibilities; we can either take v in the span of (u_1, w_1) or in the span of (u_1, u_2). The second choice does not seem natural since the eigenvector u_2 is not assumed to be known; it is precisely what we are trying to compute. However, it will illustrate an interesting point, namely that the choice $v = u_1$ may be nearly optimal in realistic situations. Thus, we will now consider the case $v \in \text{span}\{u_1, u_2\}$. The other interesting case, namely $v \in \text{span}\{u_1, w_1\}$, is left as an exercise, see Exercise P-4.3.

We can write v as $v = \alpha u_1 + \beta z$ in which z is obtained by orthonormalizing u_2 against u_1, i.e., $z = \hat{z}/\|\hat{z}\|_2$, $\hat{z} = u_2 - u_1^H u_2 u_1$. From (4.12) we immediately get $\alpha = 1$ and from (4.11) we obtain

$$\beta = -\frac{\lambda_1 - \lambda_2}{\sigma} \ \frac{u_1^H u_2}{z^H u_2} \ ,$$

which leads to the expression for the optimal v,

$$v_{opt} = u_1 - \frac{\lambda_1 - \lambda_2}{\sigma} \ \text{cotan} \ \theta(u_1, u_2) z \ . \tag{4.13}$$

We also get that

$$\text{Cond}(\tilde{\lambda}_2) = \text{Cond}(\lambda_2) \sin\theta(u_1, u_2) \ . \tag{4.14}$$

Interestingly enough, when $(\lambda_2 - \lambda_1)$ is small with respect to σ or when θ is close to $\pi/2$, the choice $v = u_1$ is nearly optimal.

This particular choice has an interesting additional property: *it preserves the Schur vectors.*

Proposition 4.1 *Let u_1 be an eigenvector of A of norm 1, associated with the eigenvalue λ_1 and let $A_1 \equiv A - \sigma u_1 u_1^H$. Then the eigenvalues of A_1 are $\tilde{\lambda}_1 = \lambda_1 - \sigma$ and $\tilde{\lambda}_j = \lambda_j, j = 2, 3 \ldots, n$. Moreover, the Schur vectors associated with $\tilde{\lambda}_j, j = 1, 2, 3 \ldots, n$ are identical with those of A.*

Proof. Let $AU = UR$ be the Schur factorization of A, where R is upper triangular and U is orthonormal. Then we have

$$A_1 U = [A - \sigma u_1 u_1^H] U = UR - \sigma u_1 e_1^H = U[R - \sigma e_1 e_1^H] \ .$$

The result follows immediately. ∎

Example 4.4 We take again as a test example the matrix Mark(10) seen is Example 4.1 and Example 4.3. We use the approximate eigenvectors u_1 and u_2 as computed from Example 4.3. We then compute the left eigenvector $\hat{w}_2$ using again the power method on the deflated and transposed matrix $A^H - \sigma u_1^H v$. This is done fpur times: first with $v = w_1 = (1, 1, ..., 1)^T$, then $v = u_1$,

$$v = (1, -1, 1, -1, 1, \ldots, (-1)^n)^T,$$

and finally $v =$ a random vector. The condition numbers obtained for the second eigenvalue for each of these choices are shown in Table 4.3. See Problem P-4.7 for additional facts concerning this example.

v	Cond(λ_2)
w_1	1.85153958
u_1	1.85153958
$(1, -1, \ldots)^T$	9.87049400
Random	2.27251031

Table 4.3 Condition numbers of the second eigenvalue for different v's.

As is observed here the best condition numbers are obtained for the first two choices. Note that the vector $(1, 1, \ldots, 1)$ is a left eigenvector associated with the eigenvalue λ_1. Surprisingly, these best two condition numbers are equal. In fact computing the inner product of u_1 and u_2 we find that it is zero, a result that is probably due to the symmetries in the physical problem. The relation (4.14) indicates that in this situation the two condition numbers are equal to the condition number for the undeflated matrix.

2.3. Deflation with Several Vectors.

Let $q_1, q_2, \ldots q_j$ be a set of Schur vectors associated with the eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_j$. We denote by Q_j the matrix of column vectors $q_1, q_2, \ldots q_j$. Thus,

$$Q_j \equiv [q_1, q_2, \ldots, q_j]$$

is an orthonormal matrix whose columns form a basis of the eigenspace associated with the eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_j$. We do not assume here that these eigenvalues are real, so the matrix Q_j may be complex. An immediate generalization of Proposition 4.1 is the following.

Proposition 4.2 *Let Σ_j be the $j \times j$ diagonal matrix*

$$\Sigma_j = \text{diag}\ (\sigma_1, \sigma_2, \ldots \sigma_j),$$

and Q_j an $n \times j$ orthogonal matrix consisting of the Schur vectors of A associated with $\lambda_1, \ldots, \lambda_j$. Then the eigenvalues of the matrix

$$A_j \equiv A - Q_j \Sigma_j Q_j^H,$$

are $\tilde{\lambda}_i = \lambda_i - \sigma_i$ for $i \leq j$ and $\tilde{\lambda}_i = \lambda_i$ for $i > j$. Moreover, its associated Schur vectors are identical with those of A.

Proof. Let $AU = UR$ be the Schur factorization of A. We have

$$A_j U = [A - Q_j \Sigma_j Q_j^H] U = UR - Q_j \Sigma_j E_j^H,$$

where $E_j = [e_1, e_2, \ldots e_j]$. Hence

$$A_j U = U[R - E_j \Sigma_j E_j^H]$$

and the result follows. ∎

Clearly, it is not necessary that Σ_j be a diagonal matrix. We can for example select it to be a triangular matrix. However, it is not clear how to select the nondiagonal entries in such a situation. An alternative technique for deflating with several Schur vectors is described in Exercise P-4.6.

2.4. Partial Schur Decomposition.

It is interesting to observe that the preservation of the Schur vectors is analogous to the preservation of the eigenvectors under Hotelling's deflation in the Hermitian case. The previous proposition suggests a simple incremental deflation procedure consisting of building the matrix Q_j one column at a time. Thus, at the j-th step, once the eigenvector $\tilde{u}_{j+1}$ of A_j is computed by the appropriate algorithm (A) we can orthonormalize it against all previous q_i's to get the next Schur vector q_{j+1} which will be appended to q_j to form the new deflation matrix Q_{j+1}. It is a simple exercise to show that the vector q_{j+1} thus computed is a Schur vector associated with the eigenvalue λ_{j+1} and therefore at every stage of the process we have the desired decomposition

$$AQ_j = Q_j R_j, \tag{4.15}$$

where R_j is some $j \times j$ upper triangular matrix.

More precisely we may consider the following algorithm, in which the successive shifts σ_i are chosen so that for example $\sigma_i = \lambda_i$.

ALGORITHM **4.4 Schur Wielandt Deflation**
For $i = 0, 1, 2, \ldots, j-1$ *do:*

1. *Define* $A_i \equiv A_{i-1} - \sigma_{i-1} q_{i-1} q_{i-1}^H$ *(initially define* $A_0 \equiv A$*) and compute the dominant eigenvalue* λ_i *of* A_i *and the corresponding eigenvector* $\tilde{u}_i$.
2. *Orthonormalize* $\tilde{u}_i$ *against* $q_1, q_2, \ldots, q_{i-1}$ *to get the vector* q_i.

With the above implementation, we may have to perform most of the computation in complex arithmetic even when A is real. Fortunately, when the matrix A is real, this can be avoided. In this case the Schur form is traditionally replaced by the quasi-Schur form, in which one still seeks for the factorization (4.2) but simply requires that the matrix R_j, be quasi-triangular, i.e. one allows for 2×2 diagonal blocks. In practice, if λ_{j+1} is complex, most algorithms do not compute the complex eigenvector y_{j+1} directly but rather deliver its real and imaginary parts y_R, y_I separately. Thus, the two eigenvectors $y_R \pm i y_I$ associated with the complex pair of conjugate eigenvalues $\lambda_{j+1}, \lambda_{j+2} = \bar{\lambda}_{j+1}$ are obtained at once.

Thinking in terms of bases of the invariant subspace instead of eigenvectors, we observe that the real and imaginary parts of the eigenvector generate the same subspace as the two conjugate eigenvectors and therefore we can work with these two real vectors instead of the (complex) eigenvectors. Hence if a complex pair occurs, all we have to do is orthogonalize the two vectors y_R, y_I against all previous q_i's and pursue the algorithm in the same way. The only difference is that the size of Q_j increases by two instead of just one in these instances.

2.5. Practical Deflation Procedures

To summarize, among all the possible deflation procedures we can use to compute the next pair λ_2, u_2, the following ones are the most useful in practice.

1. $v = w_1$ the left eigenvector. This has the disadvantage of requiring the left and right eigenvector. On the other hand both right and left eigenvectors of A_1 are preserved.

2. $v = u_1$ which is often nearly optimal and preserves the Schur vectors.

3. Use a block of Schur vectors instead of a single vector.

From the point of view of the implementation an important consideration is that we never need to form the matrix A_1 explicitly. This is important because in general A_1 will be a full matrix. In many algorithms for eigenvalue calculations, the only operation that is required is an operation of the form $y := A_1 x$. This operation can be performed as follows:

(a) Compute the vector $y := Ax$;

(b) Compute the scalar $t = \sigma\ v^H x$;

(c) Compute $y := y - t\ u_1$.

The above procedure requires only that the vectors u_1, and v be kept in memory along with the matrix A. It is possible to deflate A_1 again into A_2 , and then into A_3 etc. At each step of the process we have

$$A_i = A_{i-1} - \sigma \tilde{u}_i v_i^H \ .$$

Here one only needs to save the vectors $\tilde{u}_i$ and v_i along with the matrix A. However, one should be careful about the usage of deflation in general. It should not be used to compute more than a few eigenvalues and eigenvcectors. This is especially true in the non Hermitian case because of the fact that the matrix A_i will accumulate errors from all previous computations and this could be disastrous if the currently computed eigenvalue is poorly conditioned.

3. General Projection Methods

Most eigenvalue algorithms employ in one way or another a projection technique. The projection process can be the body of the method itself or it might simply be used within a more complex algorithm to enhance its efficiency. A simple illustration of the necessity of resorting to a projection technique is when one uses the power method in the situation when the dominant eigenvalue is complex but the matrix A is real. Although the usual sequence $x_{j+1} = \alpha_j A x_j$ where α_j is a normalizing factor, does not converge a simple analysis shows that the subspace spanned by the last two iterates x_{j+1}, x_j will contain converging approximations to the complex pair of eigenvectors. A simple projection technique onto those vectors will extract the desired eigenvalues and eigenvectors, see Exercise P-4.2 for details.

A projection method consists of approximating the exact eigenvector u, by a vector $\tilde{u}$ belonging to some subspace $\mathcal{K}$ referred to as the subspace of approximants or the right subspace, by imposing the so-called Petrov-Galerkin method that the residual vector of $\tilde{u}$ be orthogonal to some subspace $\mathcal{L}$, referred to as the left subspace. There are two broad classes of projection methods: orthogonal projection methods and oblique projection methods. In an orthogonal projection technique the subspace $\mathcal{L}$ is the same as $\mathcal{K}$. In an oblique projection method $\mathcal{L}$ is different from $\mathcal{K}$ and can be totally unrelated to it.

Not surprisingly, if no vector of the subspace $\mathcal{K}$ comes close to the exact eigenvector u, then it is impossible to get a good approximation $\tilde{u}$ to u from $\mathcal{K}$ and therefore the approximation obtained by any projection process based on $\mathcal{K}$ will be poor. If, on the other hand, there is some vector in $\mathcal{K}$ which is at a small distance ϵ from u then the question is: what accuracy can we expect to obtain? The purpose of this section is to try to answer this question.

3.1. Orthogonal Projection Methods

Let A be an $n \times n$ complex matrix and $\mathcal{K}$ be an m-dimensional subspace of $\mathbb{C}^n$. As a notational convention we will denote by the same symbol A the matrix and the linear application in $\mathbb{C}^n$ that it represents. We consider the eigenvalue problem: find u belonging to $\mathbb{C}^n$ and λ belonging to $\mathbb{C}$ such that

$$Au = \lambda u. \tag{4.16}$$

An orthogonal projection technique onto the subspace $\mathcal{K}$ seeks an approximate eigenpair $\tilde{\lambda}, \tilde{u}$ to the above problem, with $\tilde{\lambda}$ in $\mathbb{C}$ and $\tilde{u}$ in $\mathcal{K}$, such that the following Galerkin condition is satisfied:

$$A\tilde{u} - \tilde{\lambda}\tilde{u} \perp \mathcal{K} , \tag{4.17}$$

or, equivalently,

$$(A\tilde{u} - \tilde{\lambda}\tilde{u}, v) = 0 , \quad \forall \, v \in \mathcal{K}. \tag{4.18}$$

Assume that some orthonormal basis $\{v_1, v_2, \ldots, v_m\}$ of $\mathcal{K}$ is available and denote by V the matrix with column vectors $v_1, v_2, \ldots, v_m$. Then we can solve the approximate problem numerically by translating it into this basis. Letting

$$\tilde{u} = Vy, \tag{4.19}$$

equation (4.19) becomes

$$(AVy - \tilde{\lambda}Vy, v_j) = 0, \quad j = 1, \ldots, m.$$

Therefore, y and $\tilde{\lambda}$ must satisfy

$$B_m y = \tilde{\lambda} y \tag{4.20}$$

with

$$B_m = V^H AV.$$

If we denote by A_m the linear transformation of rank m defined by $A_m = \mathcal{P}_{\mathcal{K}} A \mathcal{P}_{\mathcal{K}}$ then we observe that the restriction of this operator to the subspace $\mathcal{K}$ is represented by the matrix B_m with respect to the basis V. The following is a procedure for computing numerically the Galerkin approximations to the eigenvalues/eigenvectors of A known as the Rayleigh-Ritz procedure.

ALGORITHM **4.5 Rayleigh-Ritz Procedure:**

1. *Compute an orthonormal basis* $\{v_i\}_{i=1,\ldots,m}$ *of the subspace* $\mathcal{K}$. *Let* $V = [v_1, v_2, \ldots, v_m]$.
2. *Compute* $B_m = V^H AV$;
3. *Compute the eigenvalues of* B_m *and select the* k *desired ones* $\tilde{\lambda}_i, i = 1, 2, \ldots, k$, *where* $k \le m$.
4. *Compute the eigenvectors* $y_i, i = 1, \ldots, k$, *of* B_m *associated with* $\tilde{\lambda}_i, i = 1, \ldots, k$, *and the corresponding approximate eigenvectors of* A, $\tilde{u}_i = V y_i, i = 1, \ldots, k$.

The above process only requires basic linear algebra computations. The numerical solution of the $m \times m$ eigenvalue problem in steps 3 and 4 can be treated by standard library subroutines such as those in EISPACK. Another important note is that in step 4 one can replace eigenvectors by Schur vectors to get approximate Schur vectors $\tilde{u}_i$ instead of approximate eigenvectors. Schur vectors y_i can be obtained in a numerically stable way and, in general, eigenvectors are more sensitive to rounding errors than are Schur vectors.

We can reformulate orthogonal projection methods in terms of projection operators as follows. Defining $\mathcal{P}_{\mathcal{K}}$ to be the orthogonal projector onto the subspace $\mathcal{K}$, then the Galerkin condition (4.17) can be rewritten as

$$\mathcal{P}_{\mathcal{K}}(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0 \ , \ \tilde{\lambda} \in \mathbb{C} \ , \ \tilde{u} \in \mathcal{K}$$

or,

$$\mathcal{P}_{\mathcal{K}} A\tilde{u} = \tilde{\lambda}\tilde{u} \ , \quad \tilde{\lambda} \in \mathbb{C} \ , \ \tilde{u} \in \mathcal{K} \ . \tag{4.21}$$

Note that we have replaced the original problem (4.16) by an eigenvalue problem for the linear transformation $\mathcal{P}_{\kappa} A_{|\mathcal{K}}$ which is from $\mathcal{K}$ to $\mathcal{K}$. Another formulation of the above equation is

$$\mathcal{P}_{\kappa} A \mathcal{P}_{\kappa} \tilde{u} = \tilde{\lambda} \tilde{u} \ , \ \tilde{\lambda} \in \mathbb{C} \ , \ \tilde{u} \in \mathbb{C}^n \tag{4.22}$$

which involves the natural extension

$$A_m = \mathcal{P}_{\kappa} A \mathcal{P}_{\kappa}$$

of the linear operator $A'_m = \mathcal{P}_{\kappa} A_{|\mathcal{K}}$ to the whole space. In addition to the eigenvalues and eigenvectors of A'_m, A_m has zero as a trivial eigenvalue with every vector of the orthogonal complement of $\mathcal{K}$, being an eigenvector. Equation (4.21) will be referred to as the Galerkin approximate problem.

The following proposition examines what happens in the particular case when the subspace $\mathcal{K}$ is invariant under A.

Proposition 4.3 *If $\mathcal{K}$ is invariant under A then every approximate eigenvalue / (right) eigenvector pair obtained from the orthogonal projection method onto $\mathcal{K}$ is exact.*

Proof. An approximate eigenpair $\tilde{\lambda}, \tilde{u}$ is defined by

$$\mathcal{P}_{\kappa}(A\tilde{u} - \tilde{\lambda}\tilde{u}) = 0 \ ,$$

where $\tilde{u}$ is a nonzero vector in $\mathcal{K}$ and $\tilde{\lambda} \in \mathbb{C}$. If $\mathcal{K}$ is invariant under A then $A\tilde{u}$ belongs to $\mathcal{K}$ and therefore $\mathcal{P}_{\kappa} A\tilde{u} = A\tilde{u}$. Then the above equation becomes

$$A\tilde{u} - \tilde{\lambda}\tilde{u} = 0 \ ,$$

showing that the pair $\tilde{\lambda}, \tilde{u}$ is exact. ∎

An important quantity for the convergence properties of projection methods is the distance $\|(I - \mathcal{P}_{\kappa})u\|_2$ of the exact eigenvector u, supposed of norm 1, from the subspace $\mathcal{K}$. This quantity plays a key role in the analysis of projection methods. First, it is clear that the eigenvector u cannot be well approximated from $\mathcal{K}$ if $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ is not small because we have

$$\|\tilde{u} - u\|_2 \geq \|(I - \mathcal{P}_{\mathcal{K}})u\|_2.$$

The fundamental quantity $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ can also be interpreted as the sine of the acute angle between the eigenvector u and the subspace $\mathcal{K}$. It is also the gap between the space $\mathcal{K}$ and the linear span of u. The following theorem establishes an upper bound for the residual norm of the *exact* eigenpair with respect to the approximate operator A_m, using this angle.

Theorem 4.3 *Let $\gamma = \|\mathcal{P}_{\mathcal{K}} A(I - \mathcal{P}_{\mathcal{K}})\|_2$. Then the residual norms of the pairs $\lambda, \mathcal{P}_{\mathcal{K}} u$ and λ, u for the linear operator A_m satisfy respectively*

$$\|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2 \leq \gamma \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \tag{4.23}$$

$$\|(A_m - \lambda I)u\|_2 \leq \sqrt{\lambda^2 + \gamma^2}\ \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \ . \tag{4.24}$$

Proof. For the first inequality we use the definition of A_m to get

$$\begin{aligned}
\|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2 &= \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)(u - (I - \mathcal{P}_{\mathcal{K}})u)\|_2 \\
&= \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u\|_2 \\
&= \|\mathcal{P}_{\mathcal{K}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})(I - \mathcal{P}_{\mathcal{K}})u\|_2 \\
&\leq \gamma \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \ .
\end{aligned}$$

As for the second inequality we simply notice that

$$\begin{aligned}
(A_m - \lambda I)u &= (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u + (A_m - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u \\
&= (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u - \lambda(I - \mathcal{P}_{\mathcal{K}})u \ .
\end{aligned}$$

Using the previous inequality and the fact that the two vectors on the right hand side are orthogonal to each other we get

$$\begin{aligned}
\|(A_m - \lambda I)u\|_2^2 &= \|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2^2 + |\lambda|^2 \|(I - \mathcal{P}_{\mathcal{K}})u\|_2^2 \\
&\leq (\gamma^2 + |\lambda|^2)\|(I - \mathcal{P}_{\mathcal{K}})u\|_2^2
\end{aligned}$$

which completes the proof. ∎

Note that γ is bounded from above by $\|A\|_2$. A good approximation can therefore be achieved by the projection method in case the distance $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ is small, provided the approximate eigenproblem is well conditioned. Unfortunately, in contrast with the Hermitian case the fact that the residual norm is small does not in any way guarantee that the eigenpair is accurate, because of potential difficulties related to the conditioning of the eigenvalue.

If we translate the inequality (4.23) into matrix form by expressing everything in an orthonormal basis V of $\mathcal{K}$, we would write $\mathcal{P}_{\mathcal{K}} = VV^H$ and immediately obtain

$$\|(V^H AV - \lambda I)V^H u\|_2 \leq \gamma \|(I - VV^H)u\|_2,$$

which shows that λ can be considered as an approximate eigenvalue for $B_m = V^H AV$ with residual of the order of $(I - \mathcal{P}_{\mathcal{K}})u$. If we scale the vector $V^H u$ to make it of 2-norm unity, and denote the result by y_u we can rewrite the above equality as

$$\|(V^H AV - \lambda I)y_u\|_2 \leq \gamma \frac{\|(I - \mathcal{P}_{\mathcal{K}})u\|_2}{\|\mathcal{P}_{\mathcal{K}} u\|_2} \equiv \gamma \tan \theta(u, \mathcal{K}).$$

The above inequality gives a more explicit relation between the residual norm and the angle between u and the subspace $\mathcal{K}$.

3.2. The Hermitian Case

The approximate eigenvalues computed from orthogonal projection methods in the particular case where the matrix A is Hermitian, satisfy strong optimality properties which follow from the Min-Max principle and the Courant characterization seen in Chapter 1. These properties follow by observing that $(A_m x, x)$ is the same as (Ax, x) when x runs in the subspace $\mathcal{K}$. Thus, if we label the eigenvalues decreasingly, i.e., $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, we have

$$\tilde{\lambda}_1 \;=\; \max_{x \in \mathcal{K}, x \neq 0} \frac{(\mathcal{P}_{\mathcal{K}} A \mathcal{P}_{\mathcal{K}} x, x)}{(x, x)} = \max_{x \in \mathcal{K}, x \neq 0} \frac{(\mathcal{P}_{\mathcal{K}} Ax, \mathcal{P}_{\mathcal{K}} x)}{(x, x)}$$

$$= \max_{x \in \mathcal{K},\ x \neq 0} \frac{(Ax, x)}{(x, x)} \tag{4.25}$$

This is because $\mathcal{P}_{\kappa} x = x$ for any element in $\mathcal{K}$. Similarly, we can show that

$$\tilde{\lambda}_m = \min_{x \in \mathcal{K}, x \neq 0} \frac{(Ax, x)}{(x, x)}.$$

More generally, we have the following result.

Proposition 4.4 *The $i-th$ largest approximate eigenvalue of a Hermitian matrix A, obtained from an orthogonal projection method onto a subspace $\mathcal{K}$, satisfies,*

$$\tilde{\lambda}_i = \max_{\substack{S \subset \mathcal{K} \\ dim(S)=i}} \min_{x \in S, x \neq 0} \frac{(Ax, x)}{(x, x)} \ . \tag{4.26}$$

As an immediate consequence we obtain the following corollary.

Corollary 4.1 *For $i = 1, 2, \ldots, m$ the following inequality holds*

$$\lambda_i \ \geq \tilde{\lambda}_i \ . \tag{4.27}$$

Proof. This is because,

$$\tilde{\lambda}_i = \max_{\substack{S \subset \mathcal{K} \\ \dim(S)=i}} \min_{x \in S, x \neq 0} \frac{(Ax, x)}{(x, x)} \leq \max_{\substack{S \subset \mathbb{C}^n \\ dim(S)=i}} \min_{x \in S, x \neq 0} \frac{(Ax, x)}{(x, x)} = \lambda_i \ .$$

■

A similar argument based on the Courant characterization results in the following theorem.

Theorem 4.4 *The approximate eigenvalue $\tilde{\lambda}_i$ and the corresponding eigenvector $\tilde{u}_i$ are such that*

$$\tilde{\lambda}_1 = \frac{(A\tilde{u}_1, \tilde{u}_1)}{(\tilde{u}_1, \tilde{u}_1)} = \max_{x \in \mathcal{K}, x \neq 0} \frac{(Ax, x)}{(x, x)} \ .$$

and for $i > 1$:

$$\tilde{\lambda}_i = \frac{(A\tilde{u}_i, \tilde{u}_i)}{(\tilde{u}_i, \tilde{u}_i)} = \max_{\substack{x \in \mathcal{K}, x \neq 0, \\ \tilde{u}_1^H x = \ldots = \tilde{u}_{i-1}^H x = 0}} \frac{(Ax, x)}{(x, x)} \tag{4.28}$$

One may suspect that the general bounds seen earlier for non-Hermitian matrices may be improved for the Hermitian case. This is indeed the case. We begin by proving the following lemma.

Lemma 4.1 *Let A be a Hermitian matrix and u an eigenvector of A associated with the eigenvalue λ. Then the Rayleigh quotient $\mu \equiv \mu_A(\mathcal{P}_{\mathcal{K}} u)$ satisfies the inequality*

$$|\lambda - \mu| \leq \|A - \lambda I\| \frac{\|(I - \mathcal{P}_{\mathcal{K}})u\|_2^2}{\|\mathcal{P}_{\mathcal{K}} u\|_2^2} \ . \tag{4.29}$$

Proof. From the equality

$$(A - \lambda I)\mathcal{P}_{\mathcal{K}} u = (A - \lambda I)(u - (I - \mathcal{P}_{\mathcal{K}})u) = -(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u$$

and the fact that A is Hermitian we get,

$$\begin{aligned} |\lambda - \mu| &= \left|\frac{((A - \lambda I)\mathcal{P}_{\mathcal{K}} u, \mathcal{P}_{\mathcal{K}} u)}{(\mathcal{P}_{\mathcal{K}} u, \mathcal{P}_{\mathcal{K}} u)}\right| \\ &= \left|\frac{((A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u, (I - \mathcal{P}_{\mathcal{K}})u)}{(\mathcal{P}_{\mathcal{K}} u, \mathcal{P}_{\mathcal{K}} u)}\right| \ . \end{aligned}$$

The result follows from a direct application of the Cauchy-Schwartz inequality. ∎

Assuming as usual that the eigenvalues are labeled decreasingly, and letting $\mu_1 = \mu_A(\mathcal{P}_{\mathcal{K}} u_1)$, we can get from (4.25) that

$$0 \leq \lambda_1 - \tilde{\lambda}_1 \leq \lambda_1 - \mu_1 \leq \|A - \lambda_1 I\|_2 \frac{\|(I - \mathcal{P}_{\mathcal{K}})u_1\|_2^2}{\|\mathcal{P}_{\mathcal{K}} u_1\|_2^2} \ .$$

A similar result can be shown for the smallest eigenvalue. We can extend this inequality to the other eigenvalues at the price of a little complication in the equations. In what follows we will denote by $\tilde{Q}_i$ the sum of the spectral projectors associated with the approximate eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{i-1}$. For any given vector x, $(I - \tilde{Q}_i)x$ will be the vector obtained by orthogonalizing x against the first $i-1$ approximate eigenvectors. We consider a candidate vector of the form $(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i$ in an attempt to use an argument similar to the one for the largest eigenvalue. This is a vector obtained by projecting u_i onto the subspace $\mathcal{K}$ and then stripping it off its components in the first $i-1$ approximate eigenvectors.

Lemma 4.2 *Let $\tilde{Q}_i$ be the sum of the spectral projectors associated with the approximate eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{i-1}$ and define $\mu_i = \mu_A(x_i)$, where*

$$x_i = \frac{(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i}{\|(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i\|_2} \quad .$$

Then

$$|\lambda_i - \mu_i| \le \|A - \lambda_i I\|_2 \; \frac{\|\tilde{Q}_i u_i\|_2^2 + \|(I - \mathcal{P}_{\mathcal{K}})u_i\|_2^2}{\|(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i\|_2^2} \quad . \qquad (4.30)$$

Proof. To simplify notation we set $\alpha = 1/\|(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i\|_2$. Then we write,

$$(A - \lambda_i I)x_i = (A - \lambda_i I)(x_i - \alpha u_i) \quad ,$$

and proceed as in the previous case to get,

$$|\lambda_i - \mu_i| = |((A - \lambda_i I)x_i, x_i))| = |((A - \lambda_i I)(x_i - \alpha u_i), (x_i - \alpha u_i))| \quad .$$

Applying Cauchy-Schwartz inequality to the above equation, we get

$$|\lambda_i - \mu_i| = \|A - \lambda_i I\|_2 \|x_i - \alpha u_i\|_2^2 \quad .$$

We can rewrite $\|x_i - \alpha u_i\|_2^2$ as

$$\begin{aligned} \|x_i - \alpha u_i\|_2^2 &= \alpha^2 \|(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i - u_i\|_2^2 \\ &= \alpha^2 \|(I - \tilde{Q}_i)(\mathcal{P}_{\mathcal{K}} u_i - u_i) - \tilde{Q}_i u_i\|_2^2 \ . \end{aligned}$$

Using the orthogonality of the two vectors inside the norm bars, this equality becomes

$$\begin{aligned} \|x_i - \alpha u_i\|_2^2 &= \alpha^2 \left(\|(I - \tilde{Q}_i)(\mathcal{P}_{\mathcal{K}} u_i - u_i)\|_2^2 + \|\tilde{Q}_i u_i\|_2^2 \right) \\ &\leq \alpha^2 \left(\|(I - \mathcal{P}_{\mathcal{K}})u_i\|_2^2 + \|\tilde{Q}_i u_i\|_2^2 \right) \ . \end{aligned}$$

This establishes the desired result. ∎

The vector x_i has been constructed in such a way that it is orthogonal to all previous approximate eigenvectors $\tilde{u}_1, \ldots, \tilde{u}_{i-1}$. We can therefore exploit the Courant characterization (4.28) to prove the following result.

Theorem 4.5 *Let $\tilde{Q}_i$ be the sum of the spectral projectors associated with the approximate eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{i-1}$. Then the error between the i-th exact and approximate eigenvalues λ_i and $\tilde{\lambda}_i$ is such that*

$$0 \leq \lambda_i - \tilde{\lambda}_i \leq \|A - \lambda_i I\|_2 \frac{\|\tilde{Q}_i u_i\|_2^2 + \|(I - \mathcal{P}_{\mathcal{K}})u_i\|_2^2}{\|(I - \tilde{Q}_i)\mathcal{P}_{\mathcal{K}} u_i\|_2^2} \ . \tag{4.31}$$

Proof. By (4.28) and the fact that x_i belongs to $\mathcal{K}$ and is orthogonal to the first $i-1$ approximate eigenvectors we immediately get

$$0 \leq \lambda_i - \tilde{\lambda}_i \leq \lambda_i - \mu_i.$$

The result follows from the previous lemma. ∎

We point out that the above result is valid for $i = 1$, provided we define $\tilde{Q}_1 = 0$. The quantities $\|\tilde{Q}_i u_i\|_2$ represent the cosines of the acute angle between u_i and the span of the previous approximate eigenvectors. In the ideal situation this should be zero. In addition, we should mention that the error bound is semi-a-priori, since it will require the knowledge of previous eigenvectors in order to get an idea of the quantity $\|\tilde{Q}_i u_i\|_2$.

We now turn our attention to the eigenvectors.

Theorem 4.6 *Let $\gamma = \|\mathcal{P}_{\mathcal{K}} A(I - \mathcal{P}_{\mathcal{K}})\|_2$, and consider any eigenvalue λ of A with associated eigenvector u. Let $\tilde{\lambda}$ be the approximate eigenvalue closest to λ and δ the distance between λ and the set of approximate eigenvalues other than $\tilde{\lambda}$. Then there exists an approximate eigenvector $\tilde{u}$ associated with $\tilde{\lambda}$ such that*

$$\sin[\theta(u, \tilde{u})] \leq \sqrt{1 + \frac{\gamma^2}{\delta^2}} \; \sin[\theta(u, \mathcal{K})] \tag{4.32}$$

Proof.

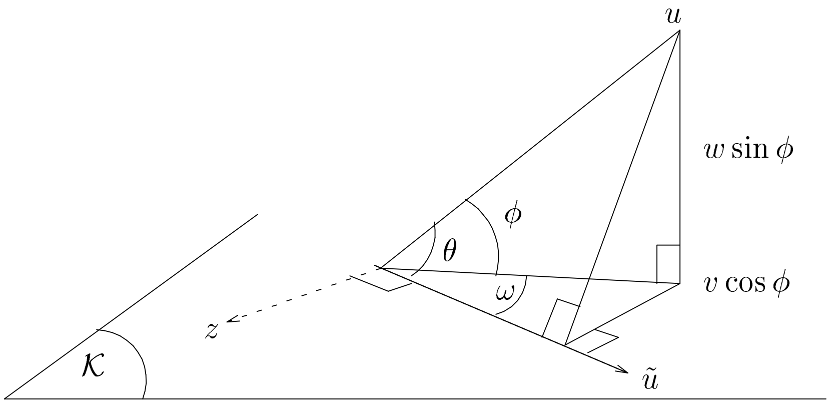

Figure 4.1 Projections of the eigenvector u onto $\mathcal{K}$ and then onto $\tilde{u}$.

Let us define the two vectors

$$v = \frac{\mathcal{P}_{\mathcal{K}} u}{\|\mathcal{P}_{\mathcal{K}} u\|_2} \quad \text{and} \quad w = \frac{(I - \mathcal{P}_{\mathcal{K}})u}{\|(I - \mathcal{P}_{\mathcal{K}})u\|_2} \tag{4.33}$$

and denote by ϕ the angle between u and $\mathcal{P}_{\mathcal{K}} u$, as defined by $\cos\phi = \|\mathcal{P}_{\mathcal{K}} u\|_2$. Then, clearly

$$u = v\cos\phi + w\sin\phi,$$

which, upon multiplying both sides by $(A-\lambda I)$ leads to

$$(A-\lambda I)v \ \cos\phi + (A-\lambda I)w \ \sin\phi = 0 \ .$$

We now project both sides onto $\mathcal{K}$, and take the norms of the resulting vector to obtain

$$\|\mathcal{P}_{\mathcal{K}}(A-\lambda I)v\|_2 \ \cos\phi = \|\mathcal{P}_{\mathcal{K}}(A-\lambda I)w\|_2 \ \sin\phi \ . \tag{4.34}$$

For the-right-hand side note that

$$\begin{aligned} \|\mathcal{P}_{\mathcal{K}}(A-\lambda I)w\|_2 &= \|\mathcal{P}_{\mathcal{K}}(A-\lambda I)(I-\mathcal{P}_{\mathcal{K}})w\|_2 \\ &= \|\mathcal{P}_{\mathcal{K}}A(I-\mathcal{P}_{\mathcal{K}})w\|_2 \le \gamma \ . \end{aligned} \tag{4.35}$$

For the left-hand-side, we decompose v further as

$$v = \tilde{u} \ \cos\omega + z \ \sin\omega,$$

in which $\tilde{u}$ is a unit vector from the eigenspace associated with $\tilde{\lambda}$, z is a unit vector in $\mathcal{K}$ that is orthogonal to $\tilde{u}$, and ω is the acute angle between v and $\tilde{u}$. We then obtain,

$$\begin{aligned} \mathcal{P}_{\mathcal{K}}(A-\lambda I)v &= \mathcal{P}_{\mathcal{K}}(A-\lambda I)[\cos\omega\tilde{u} + \sin\omega z] \\ &= \tilde{u}(\tilde{\lambda}-\lambda)\cos\omega + \mathcal{P}_{\mathcal{K}}(A-\lambda I)z\sin\omega. \end{aligned} \tag{4.36}$$

The eigenvalues of the restriction of $\mathcal{P}_{\mathcal{K}}(A-\lambda I)$ to the orthogonal of $\tilde{u}$ are $\tilde{\lambda}_j - \lambda$, for $j = 1, 2, \ldots m$, and $\tilde{\lambda}_j \neq \tilde{\lambda}$. Therefore, since z is orthogonal to $\tilde{u}$, we have

$$\|\mathcal{P}_{\mathcal{K}}(A-\lambda I)z\|_2 \ge \delta > 0. \tag{4.37}$$

The two vectors in the right hand side of (4.36) are orthogonal and by (4.37),

$$\begin{aligned} \|\mathcal{P}_{\mathcal{K}}(A-\lambda I)v\|_2^2 &= |\tilde{\lambda}-\lambda|^2\cos^2\omega + \sin^2\omega\|\mathcal{P}_{\mathcal{K}}(A-\lambda I)z\|_2^2 \\ &\ge \delta^2 \ \sin^2\omega \end{aligned} \tag{4.38}$$

To complete the proof we refer to Figure 4.1. The projection of u onto $\tilde{u}$ is the projection onto $\tilde{u}$ of the projection of u onto $\mathcal{K}$. Its length is $\cos\phi\cos\omega$ and as a result the sine of the angle θ between u and $\tilde{u}$ is given by

$$\begin{aligned} \sin^2\theta &= 1-\cos^2\phi\ \cos^2\omega \\ &= 1-\cos^2\phi\ \ (1-\sin^2\omega) \\ &= \sin^2\phi+\sin^2\omega\ \cos^2\phi\ . \end{aligned} \tag{4.39}$$

Combining (4.34), (4.35), (4.38) we obtain that

$$\sin\omega\ \cos\phi \leq \frac{\gamma}{\delta}\ \sin\phi$$

which together with (4.39) yields the desired result. ∎

This is a rather remarkable result given that it is so general. It tells us among other things that the only condition we need in order to guarantee that a projection method will deliver good approximation in the Hermitian case is that the angle between the exact eigenvector and the subspace $\mathcal{K}$ be sufficiently small.

As a consequence of the above result we can establish bounds on eigenvalues that are somewhat simpler than those of Proposition 4.5. This results from the following proposition.

Proposition 4.5 *The eigenvalues λ and $\tilde{\lambda}$ in Theorem 4.6 are such that*

$$|\lambda-\tilde{\lambda}| \leq \|A-\lambda I\|_2 \sin^2\theta(u,\tilde{u})\ . \tag{4.40}$$

Proof. We start with the simple observation that $\tilde{\lambda}-\lambda = ((A-\lambda I)\tilde{u},\tilde{u})$. Letting $\alpha=(u,\tilde{u})=\cos\theta(u,\tilde{u})$ we can write

$$\tilde{\lambda}-\lambda = ((A-\lambda I)(\tilde{u}-\alpha u),\tilde{u}) = ((A-\lambda I)(\tilde{u}-\alpha u),\tilde{u}-\alpha u)$$

The result follows immediatly by taking absolute values, exploiting the Cauchy-Schwartz inequality, and observing that $\|\tilde{u}-\alpha u\|_2=\sin\theta(u,\tilde{u})$. ∎

3.3. Oblique Projection Methods

In an oblique projection method we are given two subspaces $\mathcal{L}$ and $\mathcal{K}$ and seek an approximation $\tilde{u} \in \mathcal{K}$ and an element $\tilde{\lambda}$ of $\mathbb{C}$ that satisfy the Petrov-Galerkin condition,

$$((A - \tilde{\lambda} I)\tilde{u}, v) = 0 \quad \forall\, v \in \mathcal{L}\ . \tag{4.41}$$

The subspace $\mathcal{K}$ will be referred to as the right subspace and $\mathcal{L}$ as the left subspace. A procedure similar to the Rayleigh-Ritz procedure can be devised by again translating in matrix form the approximate eigenvector $\tilde{u}$ in some basis and expressing the Petrov-Galerkin condition (4.41). This time we will need two bases, one which we denote by V for the subspace $\mathcal{K}$ and the other, denoted by W, for the subspace $\mathcal{L}$. We assume that these two bases are biorthogonal, i.e., that $(v_i, w_j) = \delta_{ij}$, or

$$W^H V \;=\; I$$

where I is the identity matrix. Then, writing $\tilde{u} = Vy$ as before, the above Petrov-Galerkin condition yields the same approximate problem as (4.20) except that the matrix B_m is now defined by

$$B_m \;=\; W^H A V.$$

We should however emphasize that in order for a biorthogonal pair V, W to exist the following additional assumption for $\mathcal{L}$ and $\mathcal{K}$ must hold.

For any two bases V and W of $\mathcal{K}$ and $\mathcal{L}$ respectively,

$$\det(W^H V) \neq 0 \quad . \tag{4.42}$$

In order to interpret the above condition in terms of operators we will define the oblique projector $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$ onto $\mathcal{K}$ and orthogonal to $\mathcal{L}$. For any given vector x in $\mathbb{C}^n$, the vector $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x$ is defined by

$$\begin{cases} \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x \in \mathcal{K} \\ x - \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x \perp \mathcal{L}. \end{cases}$$

Note that the vector $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} x$ is uniquely defined under the assumption that no vector of the subspace $\mathcal{L}$ is orthogonal to $\mathcal{K}$. This fundamental assumption can be seen to be equivalent to assumption (4.42). When it holds the Petrov-Galerin condition (4.18) can be rewritten as

$$\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A\tilde{u} - \tilde{\lambda}\tilde{u}) \;=\; 0 \tag{4.43}$$

or

$$\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A\tilde{u} \;=\; \tilde{\lambda}\tilde{u} \;.$$

Thus, the eigenvalues of the matrix A are approximated by those of $A' = \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A_{|\mathcal{K}}$. We can define an extension A_m of A'_m analogous to the one defined in the previous section, in many different ways. For example introducing $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$ before the occurrences of $\tilde{u}$ in the above equation would lead to $A_m = \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$. In order to be able to utilize the distance $\|(I - \mathcal{P}_{\mathcal{K}})u\|_2$ in a-priori error bounds a more useful extension is

$$A_m \;=\; \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} A \mathcal{P}_{\mathcal{K}} \;.$$

With this notation, it is trivial to extend the proof of Proposition 4.3 to the oblique projection case. In other words, when $\mathcal{K}$ is invariant, then no matter which left subspace $\mathcal{L}$ we choose, the oblique projection method will always extract exact eigenpairs.

We can establish the following theorem which generalizes Theorem 4.3 seen for the orthogonal projection case.

Theorem 4.7 *Let $\gamma = \|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}(A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})\|_2$. Then the following two inequalities hold:*

$$\|(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u\|_2 \leq \gamma \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \tag{4.44}$$

$$\|(A_m - \lambda I)u\|_2 \leq \sqrt{|\lambda|^2 + \gamma^2}\; \|(I - \mathcal{P}_{\mathcal{K}})u\|_2 \;. \tag{4.45}$$

Proof. For the first inequality, since the vector $\mathcal{P}_{\mathcal{K}} y$ belongs to $\mathcal{K}$ we have $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} \mathcal{P}_{\mathcal{K}} = \mathcal{P}_{\mathcal{K}}$ and therefore

$$\begin{aligned}(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u &= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} (A - \lambda I)\mathcal{P}_{\mathcal{K}} u \\ &= \mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} (A - \lambda I)(\mathcal{P}_{\mathcal{K}} u - u) \\ &= -\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} (A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u \ .\end{aligned}$$

Since $(I - \mathcal{P}_{\mathcal{K}})$ is a projector we now have

$$(A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u = -\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}} (A - \lambda I)(I - \mathcal{P}_{\mathcal{K}})(I - \mathcal{P}_{\mathcal{K}})u.$$

Taking Euclidean norms of both sides and using the Cauchy-Schwartz inequality we immediately obtain the first result.

For the second inequality, we write

$$\begin{aligned}(A_m - \lambda I)u &= (A_m - \lambda I)\left[\mathcal{P}_{\mathcal{K}} u + (I - \mathcal{P}_{\mathcal{K}})u\right] \\ &= (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u + (A_m - \lambda I)(I - \mathcal{P}_{\mathcal{K}})u \ .\end{aligned}$$

Noticing that $A_m(I - \mathcal{P}_{\mathcal{K}}) = 0$ this becomes

$$(A_m - \lambda I)u = (A_m - \lambda I)\mathcal{P}_{\mathcal{K}} u - \lambda(I - \mathcal{P}_{\mathcal{K}})u \ .$$

Using the orthogonality of the two terms in the right hand side, and taking the Euclidean norms we get the second result. ∎

In the particular case of orthogonal projection methods, $\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}$ is identical with $\mathcal{P}_{\mathcal{K}}$, and we have $\|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}\|_2 = 1$. Moreover, the term γ can then be bounded from above by $\|A\|_2$. It may seem that since we obtain very similar error bounds for both the orthogonal and the oblique projection methods, we are likely to obtain similar errors when we use the same subspace. This is not the case in general. One reason is that the scalar γ can no longer be bounded by $\|A\|_2$ since we have $\|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}\|_2 \geq 1$ and $\|\mathcal{Q}_{\mathcal{K}}^{\mathcal{L}}\|_2$ is unknown in general. In fact the constant γ can be quite large. Another reason which was pointed out earlier is that residual norm does not provide enough information. The approximate problem can have a much worse condition number if non-orthogonal transformations are used, which may lead to poorer results. This however is only based on intuition as there are no rigorous results in this direction.

The question arises as to whether there is any need for oblique projection methods since dealing with oblique projectors may be numerically unsafe. Methods based on oblique projectors can offer some advantages. In particular they may allow to compute approximations to left as well as right eigenvectors simultaneously. There are methods based on oblique projection techniques that require also far less storage than similar orthogonal projections methods. This will be illustrated in Chapter VI.

4. Chebyshev Polynomials

Chebyshev polynomials are crucial in the study of the Lanczos algorithm and more generally of iterative methods in numerical linear algebra, such as the conjugate gradient method. They are useful both in theory, when studying convergence, and in practice, as a means of accelerating single vector iterations or projection processes.

4.1. Real Chebyshev Polynomials

The Chebyshev polynomial of the first kind of degree k is defined by

$$C_k(t) = \cos[k \cos^{-1}(t)] \quad \text{for} \quad -1 \leq t \leq 1 \ . \tag{4.46}$$

That this is a polynomial with respect to t can be easily shown by induction from the trigonometric relation

$$\cos[(k+1)\theta] + \cos[(k-1)\theta] = 2\cos\theta\cos k\theta,$$

and the fact that $C_1(t) = t, C_0(t) = 1$. Incidentally, this also shows the important three-term recurrence relation

$$C_{k+1}(t) = 2\,tC_k(t) - C_{k-1}(t) \ .$$

It is important to extend the definition (4.46) to cases where $|t| > 1$ which is done with the following formula,

$$C_k(t) = \cosh\left[k \cosh^{-1}(t)\right], \quad |t| \geq 1 \ . \tag{4.47}$$

This is readily seen by passing to complex variables and using the definition $\cos\theta = (e^{i\theta} + e^{-i\theta})/2$. As a result of (4.47) we can derive the expression,

$$C_k(t) = \frac{1}{2}\left[\left(t + \sqrt{t^2-1}\right)^k + \left(t + \sqrt{t^2-1}\right)^{-k}\right] \ , \tag{4.48}$$

which is valid for $|t| \geq 1$ but can also be extended to the case $|t|<1$. As a result, one may use the following approximation for large values of k

$$C_k(t) \quad \gtrsim \quad \frac{1}{2}\left(t + \sqrt{t^2-1}\right)^k \quad \text{for} \quad |t| \geq 1 \ . \tag{4.49}$$

In what follows we denote by $\mathbb{P}_k$ the set of all polynomials of degree k. An important result from approximation theory, which we state without proof, is the following theorem.

Theorem 4.8 *Let $[\alpha, \beta]$ be a non-empty interval in $\mathbb{R}$ and let γ be any real scalar such with $\gamma \geq \beta$. Then the minimum*

$$\min_{p \in \mathbb{P}_k, p(\gamma)=1} \ \max_{t\in[\alpha,\beta]} |p(t)|$$

is reached by the polynomial

$$\hat{C}_k(t) \equiv \frac{C_k\left(1 + 2\frac{t-\beta}{\beta-\alpha}\right)}{C_k\left(1 + 2\frac{\gamma-\beta}{\beta-\alpha}\right)} \ .$$

For a proof see [16]. The maximum of C_k for t in $[-1, 1]$ is 1 and as a corollary we have

$$\min_{p \in \mathbb{P}_k, \ p(\gamma)=1} \ \max_{t\in[\alpha,\beta]} |p(t)| = \frac{1}{|C_k(1 + 2\frac{\gamma-\beta}{\beta-\alpha})|} = \frac{1}{|C_k(2\frac{\gamma-\mu}{\beta-\alpha})|} \ .$$

in which $\mu \equiv (\alpha + \beta)/2$ is the middle of the interval. Clearly, the results can be slightly modified to hold for the case where $\gamma \leq \alpha$, i.e., when γ is to the left of the interval.

4.2. Complex Chebyshev Polynomials

The standard definition given in the previous section for Chebyshev polynomials of the first kind, see equation (4.46), extends without difficulty to complex variables. First, as was seen before, when t is real and $|t| > 1$ we can use the alternative definition, $C_k(t) = \cosh[k \cosh^{-1}(t)], \quad 1 \le |t|$. More generally, one can unify these definitions by switching to complex variables and writing

$$C_k(z) = \cosh(k\zeta), \quad \text{where} \quad \cosh(\zeta) = z .$$

Defining the variable $w = e^{\zeta}$, the above formula is equivalent to

$$C_k(z) = \frac{1}{2}[w^k + w^{-k}] \quad \text{where} \quad z = \frac{1}{2}[w + w^{-1}]. \qquad (4.50)$$

We will use the above definition for Chebyshev polynomials in $\mathbb{C}$. Note that the equation $\frac{1}{2}(w + w^{-1}) = z$ has two solutions w which are inverses of each other, and as a result the value of $C_k(z)$ does not depend on which of these solutions is chosen. It can be verified directly that the C_k's defined by the above equations are indeed polynomials in the z variable and that they satisfy the three term recurrence

$$C_{k+1}(z) = 2\,zC_k(z) - C_{k-1}(z), \qquad (4.51)$$

with $C_0(z) \equiv 1$ and $C_1(z) \equiv z$.

As is now explained, Chebyshev polynomials are intimately related to ellipses in the complex plane. Let C_ρ be the circle of center the origin and radius ρ. Then the so-called Joukowski mapping

$$J(w) = \frac{1}{2}[w + w^{-1}]$$

transforms C_ρ into an ellipse of center the origin, foci $-1, 1$ and major semi-axis $\frac{1}{2}[\rho + \rho^{-1}]$ and minor semi-axis $\frac{1}{2}|\rho - \rho^{-1}|$. This is illustrated in Figure 4.2.

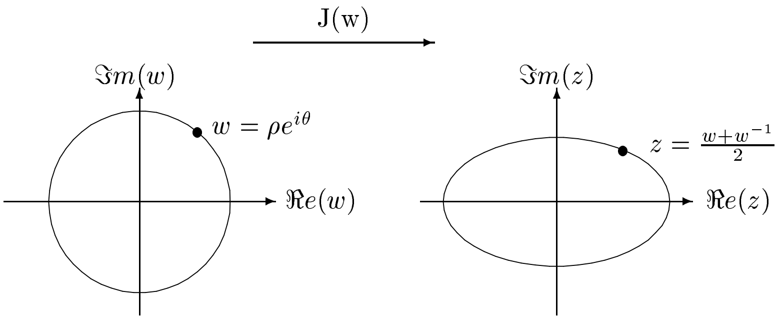

Figure 4.2 The Joukowski mapping transforms a circle into an ellipse in the complex plane.

There are two circles which have the same image by the mapping $J(w)$, one with the radius ρ and the other with the radius ρ^{-1}. So it suffices to consider those circles with $\rho \geq 1$. Note that the case $\rho = 1$ is a degenerate case in which the ellipse $E(0, 1, -1)$ reduces the interval $[-1, 1]$ traveled through twice.

One important question we now ask is whether or not a min-max result similar to the one of Theorem 4.8 holds for the complex case. Here the maximum of $|p(z)|$ is taken over the ellipse boundary and γ is some point not enclosed by the ellipse. A 1963 paper by Clayton [19] was generally believed for quite some time to have established the result, at least for the special case where the ellipse has real foci and γ is real. It was recently shown by Fischer and Freund that in fact Clayton's result was incorrect in general [46]. On the other hand, Chebyshev polynomials are asymptotically optimal and in practice that is all that is needed.

To show the asymptotic optimality, we start by stating a lemma due to Zarantonello, which deals with the particular case where the ellipse reduces to a circle. This particular case is important in itself.

Lemma 4.3 (Zarantonello) *Let $C(0, \rho)$ be a circle of center the origin and radius ρ and let γ a point of $\mathbb{C}$ not enclosed by $C(0, \rho)$.*

Then,

$$\min_{p\in \mathbb{P}_k,\ \ p(\gamma)=1} \ \max_{z\ \in\ C(0,\rho)} |p(z)| \ = \ \left(\frac{\rho}{|\gamma|}\right)^k , \tag{4.52}$$

the minimum being achieved for the polynomial $(z/\gamma)^k$*.*

Proof. See reference [132] for a proof. ∎

Note that by changing variables, shifting and rescaling the polynomial, we also get for any circle centered at c and for any scalar γ such that $|\gamma| > \rho$,

$$\min_{p\in \mathbb{P}_k\ \ p(\gamma)=1} \ \max_{z\ \in\ C(c,\rho)} |p(z)| \ = \ \left(\frac{\rho}{|\gamma - c|}\right)^k$$

We now consider the general case of an ellipse centered at the origin, with foci $1, -1$ and semi-major axis a, which can be considered as mapped by J from the circle $C(0,\rho)$, with the convention that $\rho \geq 1$. We denote by E_ρ such an ellipse.

Theorem 4.9 *Consider the ellipse* E_ρ *mapped from* $C(0,\rho)$ *by the mapping* J *and let* γ *any point in the complex plane not enclosed by it. Then*

$$\frac{\rho^k}{|w_\gamma|^k} \leq \min_{p\in \mathbb{P}_k\ p(\gamma)=1} \ \max_{z\ \in\ E_\rho} |p(z)| \leq \frac{\rho^k + \rho^{-k}}{|w_\gamma^k + w_\gamma^{-k}|} \tag{4.53}$$

in which w_γ *is the dominant root of the equation* $J(w) = \gamma$*.*

Proof. We start by showing the second inequality. Any polynomial p of degree k satisfying the constraint $p(\gamma) = 1$ can be written as,

$$p(z) = \frac{\sum_{j=0}^k \xi_j z^j}{\sum_{j=0}^k \xi_j \gamma^j} \quad .$$

A point z on the ellipse is transformed by J from a certain w in $C(0,\rho)$. Similarly, let w_γ be one of the two inverse transforms of γ by the mapping, namely the one with largest modulus. Then, p can be rewritten as

$$p(z) = \frac{\sum_{j=0}^{k} \xi_j (w^j + w^{-j})}{\sum_{j=0}^{k} \xi_j (w_\gamma^j + w_\gamma^{-j})} \; . \tag{4.54}$$

Consider the particular polynomial obtained by setting $\xi_k = 1$ and $\xi_j = 0$ for $j \neq k$,

$$p^*(z) = \frac{w^k + w^{-k}}{w_\gamma^k + w_\gamma^{-k}}$$

which is a scaled Chebyshev polynomial of the first kind of degree k in the variable z. It is not too difficult to see that the maximum modulus of this polynomial is reached in particular when $w = \rho e^{i\theta}$ is real, i.e., when $w = \rho$. Thus,

$$\max_{z \in E_\rho} |p^*(z)| = \frac{\rho^k + \rho^{-k}}{|w_\gamma^k + w_\gamma^{-k}|}$$

which proves the second inequality.

To prove the left inequality, we rewrite (4.54) as

$$p(z) = \left(\frac{w^{-k}}{w_\gamma^{-k}}\right) \; \frac{\sum_{j=0}^{k} \xi_j (w^{k+j} + w^{k-j})}{\sum_{j=0}^{k} \xi_j (w_\gamma^{k+j} + w_\gamma^{k-j})}$$

and take the modulus of $p(z)$,

$$|p(z)| = \frac{\rho^{-k}}{|w_\gamma|^{-k}} \; \left| \frac{\sum_{j=0}^{k} \xi_j (w^{k+j} + w^{k-j})}{\sum_{j=0}^{k} \xi_j (w_\gamma^{k+j} + w_\gamma^{k-j})} \right| \; .$$

The polynomial in w of degree $2k$ inside the large modulus bars in the right-hand-side is such that its value at w_γ is one. By Lemma 4.3, the modulus of this polynomial over the circle $C(0,\rho)$

is not less than $(\rho/|w_\gamma|)^{2k}$, i.e., for any polynomial, satisfying the constraint $p(\gamma) = 1$ we have,

$$\max_{z \in E_\rho} |p(z)| \geq \frac{\rho^{-k}}{|w_\gamma|^{-k}} \frac{\rho^{2k}}{|w_\gamma|^{2k}} = \frac{\rho^k}{|w_\gamma|^k}.$$

This proves that the minimum over all such polynomials of the maximum modulus on the ellipse E_ρ is $\geq (\rho/|w_\gamma|)^k$. ■

The difference between the left and right bounds in (4.53) tends to zero as k increases to infinity. Thus, the important point made by the theorem is that, for large k, the Chebyshev polynomial

$$p^*(z) = \frac{w^k + w^{-k}}{w_\gamma^k + w_\gamma^{-k}}, \quad \text{where} \quad z = \frac{w + w^{-1}}{2}$$

is close to the optimal polynomial. In other words these polynomials are *asymptotically* optimal.

For a more general ellipse centered at c, and with focal distance d, a simple change of variables shows that the near-best polynomial is given by

$$C_k\left(\frac{z-c}{d}\right).$$

We should point out that an alternative result, which is more complete, has been proven by Fischer and Freund in [45].

PROBLEMS

P-4.1 What are the eigenvalues and eigenvectors of $(A-\sigma I)^{-1}$. What are all the shifts σ that will lead to a convergence towards a given eigenvalue λ?

P-4.2 Consider a real nonsymmetric matrix A. The purpose of this exercise is to develop a generalization of the power method that can handle the case where the dominant eigenvalue is complex (i.e., we have a complex conjugate pair of dominant eigenvalues). Show that by a projection process onto two successive iterates of the power method one can achieve convergence towards the dominant pair of eigenvalues [Consider the diagonalizable case only]. Without giving a proof, state what the rate of convergence toward the pair of complex conjugate eigenvectors should be. Develop a simple version of a corresponding algorithm and then a variation of the algorithm that orthonormalizes two successive iterates at every step, i.e., starting with a vector x of 2-norm unity, the iterates are as follows,

$$x_{new} := \frac{\hat{x}}{\|\hat{x}\|_2} \quad \text{where} \quad \hat{x} := Ax_{old} - (Ax_{old}, x_{old})x_{old} \ .$$

Does the orthogonalization have to be done at every step?

P-4.3 By following a development similar to that subsection 4.2, find the v vector for Wielandt deflation, which minimizes the condition number for A_1, among all vectors in the span of u_1, w_1. Show again that the choice $v = u_1$ is nearly optimal when $\lambda_1 - \lambda_2$ is small relative to σ.

P-4.4 Consider the generalized eigenvalue problem $Ax = \lambda Bx$. How can one generalize the power method? The shifted power method? and the shift-and-invert power method?

P-4.5 Assume that all the eigenvalues of a matrix A are real and that one uses the shifted power method for computing the largest, i.e., the rightmost eigenvalue of a given matrix. What are all the admissible shifts, i.e., those that will lead to convergence toward the rightmost eigenvalue? Among all the admissible choices which one leads to the best convergence rate?

P-4.6 Consider a deflation technique which would compute the eigenvalues of the matrix

$$A_1 = (I - Q_j Q_j^H)A$$

in which $Q_j = [q_1, q_2, \ldots, q_j]$ are previously computed Schur vectors. What are the eigenvalues of the deflated matrix A_1? Show that an eigenvector of A_1 is a Schur vector for A. The advantage of this technique is that there is no need to select shifts σ_j. What are the disadvantages if any?

P-4.7 Show that in example 4.4 any linear combination of the vectors u_1 and w_1 is in fact optimal.

P-4.8 Nothing was said about the left eigenvector $\tilde{w}_1$ of the deflated matrix A_1 in Section 4.2. Assuming that the matrix A is diagonalizable find an eigenvector $\tilde{w}_1$ of A_1 associated with the eigenvalue $\lambda_1 - \sigma$. [*Hint*: Express the eigenvector in the basis of the left eigenvectors of A.] How can this be generalized to the situation where A is not diagonalizable?

P-4.9 Assume that the basis V of the subspace $\mathcal{K}$ used in an orthogonal projection process is not orthogonal. What matrix problem do we obtain if we translate the Galerkin conditions using this basis. Same question for the oblique projection technique, i.e., assuming that V, W does not form a bi-orthogonal pair. Ignoring the cost of the small m-dimensional problems, how do the computational costs compare? What if we include the cost of the orthonormalization (by modified Gram-Schmidt) for the approach which uses orthogonal bases (Assuming that the basis V is obtained from orthonormalizing a set of m basis vectors).

P-4.10 Let A be Hermitian and let $\tilde{u}_i, \tilde{u}_j$ two Ritz eigenvectors associated with two different eigenvalues $\tilde{\lambda}_i, \tilde{\lambda}_j$ respectively. Show that $(A\tilde{u}_i, \tilde{u}_j) = \tilde{\lambda}_j \delta_{ij}$.

P-4.11 Prove from the definition (4.50) that the C_k's are indeed polynomials in z and that they satisfy the three-term recurrence (4.51).

NOTES AND REFERENCES. Much of the material on projection methods presented in this chapter is based on the papers [141, 138] and the section on deflation procedures is from [147] and some well-known results in Wilkinson [183]. Suggested additional reading on projection methods are Chatelin [14] and Krasnoselskii et al. [87]. A good discussion of Chebyshev polynomials in the complex plane is given in the book by Rivlin [132]. Deflation for non Hermitian eigenvalue problems is not that much used in the literature. I found Schur-Wielandt and related deflation procedures (based on Schur vectors rather than eigenvectors) to be essential in the design of robust eigenvalue algorithms. ♠

Chapter V

Subspace Iteration

Among the best known methods for solving large sparse eigenvalue problems, the subspace iteration algorithm is undoubtedly the simplest. This method can be viewed as a block generalization of the power method. Although the method is not competitive with other projections methods to be covered in later chapters, it still is one of the most important methods used in structural engineering. It also constitutes a good illustration of the material covered in the previous chapter.

1. Simple Subspace Iteration

The original version of subspace iteration was introduced by Bauer under the name of *Treppeniteration* (staircase iteration). *Bauer's Treppeniteration* Bauer's method consists of starting with an initial system of m vectors forming an $n \times m$ matrix $X_0 = [x_1, \ldots, x_m]$ and computing the matrix

$$X_k = A^k X_0. \tag{5.1}$$

for a certain power k. If we normalized the column vectors separately in the same manner as for the power method, then in typical cases each of these vectors will converge to the same eigenvector associated with the dominant eigenvalue. Thus the system X_k will progressively loose its linear independence. The idea of Bauer's method is to reestablish linear independence for these vectors by a process such as the LR or the QR factorization. Thus, if we use the more common QR option, we get the following algorithm.

ALGORITHM **5.1 Simple Subspace Iteration**

1. **Start:** *Choose an initial system of vectors* $X_0 = [x_1, \ldots, x_m]$.
2. **Iterate:** *Until convergence do,*
   (a) *Compute* $X_k := AX_{k-1}$
   (b) *Compute the QR factorization* $X_k = QR$ *of* X_k*, and set* $X_k := Q$.

This algorithm can be viewed as a direct generalization of the power method seen in the previous Chapter. Step 2-(b) is a normalization process that is much similar to the normalization used in the power method, and just as for the power method there are many possible normalizations that can be used. An important observation is that the subspace spanned by the vectors X_k is the same as that spanned by $A^k X_0$. Since the cost of 2-(b) can be high, it is natural to orthonormalize as infrequently as possible, i.e. to perform several steps at once before performing an orthogonalization. This leads to the following modification.

ALGORITHM **5.2 Multiple Step Subspace Iteration**

1. **Start:** *Choose an initial system of vectors* $X = [x_1, \ldots, x_m]$. *Choose an iteration parameter* $iter$.
2. **Iterate:** *Until convergence do:*
   (a) *Compute* $Z := A^{iter} X$.
   (b) *Orthonormalize* Z. *Copy resulting matrix onto* X.
   (c) *Select a new* $iter$.

We would like to make a few comments concerning the choice of the parameter $iter$. The best $iter$ will depend on the convergence rate. If $iter$ is too large then the vectors of Z in 2-(a) may become nearly linear dependent and the orthogonalization in 2-(b) may cause some difficulties. Typically an estimation on the speed of convergence is used to determine $iter$. Then $iter$ is defined in such a way that, for example, the fastest converging vector, which is the first one, will have converged to within a certain factor, e.g., the square root of the machine epsilon, i.e., the largest number ϵ that causes rounding to yield $1 + \epsilon == 1$ on a given computer.

Under a few assumptions the column vectors of X_k will converge "in direction" to the Schur vectors associated with the m dominant eigenvalues $\lambda_1, \ldots, \lambda_m$. To formalize this peculiar notion of convergence, a form of which was seen in the context of the power method, we will say that a sequence of vectors x_k converges *essentially* to a vector x if there exists a sequence of signs $e^{i\theta_k}$ such that the sequence $e^{i\theta_k} x_k$ converges to x.

Theorem 5.1 *Let* $\lambda_1, \ldots, \lambda_m$ *be the* m *dominant eigenvalues of* A *labeled in decreasing order of magnitude and assume that* $|\lambda_i| > |\lambda_{i+1}|, 1 \leq i \leq m$. *Let* $Q = [q_1, q_2, \ldots, q_m]$ *be the Schur vectors* *associated with $\lambda_j, j = 1, \ldots, m$ and P_i be the spectral projector associated with the eigenvalues $\lambda_1, \ldots, \lambda_i$. Assume that*

$$\text{rank}\,(P_i[x_1, x_2, \ldots, x_i]) = i, \quad \textit{for} \quad i = 1, 2, \ldots, m \ .$$

Then the i-th column of X_k converges essentially to q_i, for $i = 1, 2, \cdots, m$.

Proof. Let the initial system X_0 be decomposed as

$$X_0 = P_m X_0 + (I - P_m) X_0 = Q G_1 + W G_2 \tag{5.2}$$

where W is an $n \times (n-m)$ matrix whose column vectors form some basis of the invariant basis $(I - P_m)\mathbb{C}^n$ and G_2 is a certain $(n-m) \times m$ matrix. We know that there exists an $m \times m$ upper triangular matrix R_1 and an $(n-m) \times (n-m)$ matrix R_2 such that

$$AQ = QR_1 \ , \quad AW = WR_2 \ . \tag{5.3}$$

The column vectors of X_k are obtained by orthonormalizing the system $Z_k = A^k X_0$. By assumption, the system of column vectors $P_m X_0$ is nonsingular and therefore G_1 is nonsingular. Applying (5.3) we get

$$\begin{aligned} A^k X_0 &= A^k [QG_1 + WG_2] \\ &= QR_1^k G_1 + WR_2^k G_2 \\ &= [Q + WR_2^k G_2 G_1^{-1} R_1^{-k}] R_1^k G_1 \end{aligned}$$

The term $E_k \equiv WR_2^k G_2 G_1^{-1} R^{-k}$ tends to zero because the spectral radius of R_1^{-1} is equal to $1/|\lambda_m|$ while that of R_2 is $|\lambda_{m+1}|$. Hence,

$$A^k X_0 G_1^{-1} = [Q + E_k] R_1^k$$

with $\lim_{k \to \infty} E_k = 0$. Using the QR decomposition of the matrix $Q + E_k$,

$$Q + E_k = Q^{(k)} R^{(k)},$$

we obtain

$$A^k X_0 G_1^{-1} = Q^{(k)} R^{(k)} R_1^k.$$

Since E_k converges to zero, it is clear that $R^{(k)}$ converges to the identity matrix while $Q^{(k)}$ converges to Q, and because the QR decomposition of a matrix is unique up to scaling constants, we have established that the Q matrix in the QR decomposition of the matrix $A^k X_0 G_1^{-1}$ *converges essentially* to Q. Notice that the span of $A^k X_0 G_1^{-1}$ is identical with that of X_k. As a result the orthogonal projector $\mathcal{P}_m^{(k)}$ onto $span\{X_k\}$ will converge to the orthogonal projector $\mathcal{P}_m$ onto $span\{Q\}$.

In what follows we denote by $[X]_j$ the matrix of the first j vector columns of X. To complete the proof, we need to show that each column converges to the corresponding column vector of Q. To this end we observe that the above proof extends to the case where we consider only the first j columns of X_k, i.e., the j first columns of X_k converge to a matrix that spans the same subspace as $[Q]_j$. In other words, if we let $\mathcal{P}_j$ be the orthogonal projector on $span\{[Q]_j\}$ and $\mathcal{P}_j^{(k)}$ the orthogonal projector on $span\{[X_k]_j\}$ then we have $\mathcal{P}_j^{(k)} \rightarrow \mathcal{P}_j$ for $j = 1, 2, \ldots, m$. The proof is now by induction. When $j = 1$, we have the obvious result that the first column of X_k converges essentially to q_1. Assume that the columns 1 through i of X_k converge essentially to $q_1, \ldots, q_i$. Consider the last column $x_{i+1}^{(k)}$ of $[X_k]_{i+1}$, which we express as

$$x_{i+1}^{(k)} = \mathcal{P}_{i+1}^{(k)} x_{i+1}^{(k)} = \mathcal{P}_i^{(k)} x_{i+1}^{(k)} + (\mathcal{P}_{i+1}^{(k)} - \mathcal{P}_i^{(k)}) x_{i+1}^{(k)} \quad .$$

The first term in the right hand side is equal to zero because by construction $x_{i+1}^{(k)}$ is orthogonal to the first i columns of $[X_k]_{i+1}$. Hence,

$$x_{i+1}^{(k)} = (\mathcal{P}_{i+1}^{(k)} - \mathcal{P}_i^{(k)}) x_{i+1}^{(k)}$$

and by the above convergence results on the projectors $\mathcal{P}_j^{(k)}$ we see that $\mathcal{P}_{i+1}^{(k)} - \mathcal{P}_i^{(k)}$ converges to the orthogonal projector onto the span of the single vector q_{i+1}. This is because

$$\mathcal{P}_{i+1} - \mathcal{P}_i = Q_{i+1} Q_{i+1}^H - Q_i Q_i^H = q_{i+1} q_{i+1}^H \ .$$

Therefore we may write $x_{i+1}^{(k)} = q_{i+1}q_{i+1}^H x_{i+1}^{(k)} + \epsilon_k$ where ϵ_k converges to zero. Since the vector $x_{i+1}^{(k)}$ is of norm unity, its orthogonal projection onto q_{i+1} will essentially converge to q_{i+1}. ■

The proof indicates that the convergence of each column vector to the corresponding Schur vector is governed by the convergence factor $|\lambda_{i+1}/\lambda_i|$. In addition, we have also proved that each orthogonal projector $\mathcal{P}_i^{(k)}$ onto the first i columns of X_k converges under the assumptions of the theorem.

2. Subspace Iteration with Projection

In the subspace iteration with projection method the column vectors obtained from the previous algorithm are not directly used as approximations to the Schur vectors. Instead they are employed in a Rayleigh-Ritz process to get better approximations. In fact as was seen before, the Rayleigh-Ritz approximations are optimal in some sense in the Hermitian case and as a result it is sensible to use a projection process whenever possible. This algorithm with projection is as follows.

ALGORITHM **5.3 Subspace Iteration with Projection**

1. **Start:** *Choose an initial system of vectors $X = [x_0, \ldots, x_m]$ and an initial iteration parameter iter.*
2. **Iterate:** *Until convergence do:*
   (a) *Compute $\hat{Z} = A^{iter} X_{old}$.*
   (b) *Orthonormalize $\hat{Z}$ into Z.*
   (c) *Compute $B = Z^H A Z$ and use the QR algorithm to compute the Schur vectors $Y = [y_1, \ldots, y_m]$ of B.*
   (d) *Compute $X_{new} = ZY$.*
   (e) *Test for convergence and select a new iteration parameter iter.*

There are many implementation details which are omitted for the sake of clarity. Note that there is another version of the algorithm which uses eigenvectors instead of Schur vectors (in Step 2-(c)). These two versions are obviously equivalent when A is Hermitian.

Let S_k be the subspace spanned by X_k and let us denote by $\mathcal{P}_k$ the *orthogonal projector* onto the subspace S_k. Assume that the eigenvalues are ordered in decreasing order of magnitude and that,

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \cdots \geq |\lambda_m| > |\lambda_{m+1}| \geq \cdots \geq |\lambda_n| \quad .$$

Again u_i denotes an eigenvector of A of norm unity associated with λ_i. The spectral projector associated with the invariant subspace associated with $\lambda_1, \ldots, \lambda_m$ will be denoted by P. We will now prove the following theorem.

Theorem 5.2 *Let $S_0 = span\{x_1, x_2, \ldots, x_m\}$ and assume that S_0 is such that the vectors $\{Px_i\}_{i=1,\ldots,m}$ are linearly independent. Then for each eigenvector u_i of A, $i = 1, \ldots, m$, there exists a unique vector s_i in the subspace S_0 such that $Ps_i = u_i$. Moreover, the following inequality is satisfied*

$$\|(I - \mathcal{P}_k)u_i\|_2 \leq \|u_i - s_i\|_2 \left(\left| \frac{\lambda_{m+1}}{\lambda_i} \right| + \epsilon_k \right)^k , \tag{5.4}$$

where ϵ_k tends to zero as k tends to infinity.

Proof. By their assumed linear independence, the vectors Px_j, form a basis of the invariant subspace $P\mathbb{C}^n$ and so the vector u_i, which is a member of this subspace, can be written as

$$u_i = \sum_{j=1}^{m} \eta_j P x_j = P \sum_{j=1}^{m} \eta_j x_j \equiv P s_i .$$

The vector s_i is such that

$$s_i = u_i + w, \tag{5.5}$$

where $w = (I - P)s_i$. Next consider the vector y of S_k defined by $y = (\frac{1}{\lambda_i})^k A^k s_i$. We have from (5.5) that

$$y - u_i = \left(\frac{1}{\lambda_i}\right)^k A^k w \ . \tag{5.6}$$

Denoting by W the invariant subspace corresponding to the eigenvalues $\lambda_{m+1}, \ldots, \lambda_n$, and noticing that w is in W, we clearly have

$$y - u_i = \left(\frac{1}{\lambda_i}\right)^k [A_{|W}]^k w.$$

Hence,

$$\|u_i - y\|_2 \leq \left\| \left[\frac{1}{\lambda_i} A_{|W}\right]^k \right\|_2 \|w\|_2 \ . \tag{5.7}$$

Since the eigenvalues of $A_{|W}$ are $\lambda_{m+1}, \lambda_{m+2}, \ldots, \lambda_n$ the spectral radius of $[\frac{1}{\lambda_i} A_{|W}]$ is simply $|\lambda_{m+1}/\lambda_i|$ and from Corollary 1.1 of Chapter I, we have,

$$\left\| \left[\frac{1}{\lambda_i} A_{|W}\right]^k \right\|_2 = \left[\left|\frac{\lambda_{m+1}}{\lambda_i}\right| + \epsilon_k\right]^k , \tag{5.8}$$

where ϵ_k tends to zero as $k \to \infty$. Using the fact that

$$\|(I - \mathcal{P}_k)u_i\|_2 = \min_{y \in S_k} \|y - u_i\|_2$$

together with inequality (5.7) and equality (5.8) yields the desired result (5.4). ■

We can be a little more specific about the sequence ϵ_k of the theorem by using the inequality

$$\|B^k\|_2 \leq \alpha \rho^k k^{\eta - 1}, \tag{5.9}$$

where B is any matrix, ρ its spectral radius, η the dimension of its largest Jordan block, and α some constant *independent* on k, see Exercise P-5.6 as well as Householder's book [73]. Without loss of generality we assume that $\alpha \geq 1$.

Initially, consider the case where A is diagonalizable. Then $\eta = 1$, and by replacing (5.9) in (5.8) we observe that (5.4) simplifies into

$$\|(I - \mathcal{P}_k)u_i\|_2 \leq \alpha \|u_i - s_i\|_2 \left| \frac{\lambda_{m+1}}{\lambda_i} \right|^k . \tag{5.10}$$

Still in the diagonalizable case, it is possible to get a more explicit result by expanding the vector s_i in the eigenbasis of A as

$$s_i = u_i + \sum_{i=m+1}^{n} \xi_j u_j .$$

Letting $\beta = \sum_{i=m+1}^{n} |\xi_j|$, we can reproduce the proof of the above theorem to obtain

$$\|(I - \mathcal{P}_k)u_i\|_2 \leq \alpha\beta \left| \frac{\lambda_{m+1}}{\lambda_i} \right|^k . \tag{5.11}$$

When A is not diagonalizable, then from comparing (5.9) and (5.8) we can bound ϵ_k from above as follows:

$$\epsilon_k \leq \left| \frac{\lambda_{m+1}}{\lambda_i} \right| (\alpha^{1/k} k^{(\eta-1)/k} - 1)$$

which confirms that ϵ_k tends to zero as k tends to infinity.

Finally, concerning the assumptions of the theorem, it can be easily seen that the condition that $\{Px_j\}_{j=1,\ldots,r}$ form an independent system of vectors is equivalent to the condition that

$$\det[U^H S_0] \neq 0,$$

in which U is any basis of the invariant subspace $P\mathbb{C}^n$. This condition constitutes a generalization of a similar condition required for the convergence of the power method.

3. Practical Implementations

There are a number of implementation details that enhance the performance of the simple methods described above. The first of these is the use of locking, a form of deflation, which exploits the inequal convergence rates of the different eigenvectors. In addition, the method is rarely used without some form of acceleration. Similarly to the power method the simplest form of acceleration, is to shift the matrix to optimize the convergence rate for the eigenvalue being computed. However, there are more elaborate techniques which will be briefly discussed later.

3.1. Locking

Because of the different rates of convergence of each of the approximate eigenvalues computed by the subspace iteration, it is a common practice to extract them one at a time and perform a form of deflation. Thus, as soon as the first eigenvector has converged there is no need to continue to multiply it by A in the subsequent iterations. Indeed we can freeze this vector and work only with the vectors $q_2, \ldots, ...q_m$. However, we will still need to perform the subsequent orthogonalizations with respect to the frozen vector q_1 whenever such orthogonalizations are needed. The term used for this strategy is *locking* . It was introduced by Jennings and Stewart [78]. Note that acceleration techniques and other improvements to the basic subspace iteration desribed in Section 3 can easily be combined with locking.

The following algorithm describes a practical subspace iteration with deflation (locking) for computing the *nev* dominant eigenvalues.

ALGORITHM **5.4 Subspace Iteration with Projection and Deflation**

1. **Start:** *Choose an initial system of vectors* $X := [x_0, \ldots, x_m]$ *and an initial iteration parameter* $iter$. *Set* $j := 1$.

2. **Eigenvalue loop:** *While $j \leq nev$ do:*

   (a) *Compute $\hat{Z} = [q_1, q_2, \ldots, q_{j-1}, A^{iter}X]$.*

   (b) *Orthonormalize the column vectors of $\hat{Z}$ (starting at column j) into Z.*

   (c) *Update $B = Z^H AZ$ and compute the Schur vectors $Y = [y_j, \ldots, y_m]$ of B associated with the eigenvalues $\lambda_j, \ldots, \lambda_m$.*

   (d) *Test the eigenvalues $\lambda_j, \ldots, \lambda_m$ for convergence. Let i_{conv} the number of newly converged eigenvalues. Append the i_{conv} corresponding Schur vectors to $Q = [q_1, ..., q_{j-1}]$ and set $j := j + i_{conv}$.*

   (e) *Compute $X := Z[y_j, y_{j+1}, \ldots, y_m]$.*

   (f) *Compute a new iteration parameter $iter$.*

Example 5.1 Consider the matrix Mark(10) described in Chapter II and used in the test examples of Chapter IV. We tested a version of the algorithm just described to compute the three dominant eigenvalues of Mark(10). In this test we took $m = 10$ and started with an initial set of vectors obtained from orthogonalizing $v, Av, ..., A^m v$, in which v is a random vector. Table 5.1 shows the results. Each horizontal line separates an outer loop of the algorithm (corresponding to step (2) in algorithm 5.4). Thus, the algorithm starts with $iter = 5$ and in the first iteration (requiring 63 matrix-vector products) no new eigenvalue has converged. We will need three more outer iterations (requiring each 113 matrix-vector products) to achieve convergence for the two dominant eigenvalues $-1, 1$. Another outer iteration is needed to compute the third eigenvalue. Note that each projection costs 13 additional matrix by vector products, 10 for computing the C matrix and 3 for the residual vectors.

Mat-vec's	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
63	0.1000349211D+01	0.0	0.820D-02
	-0.9981891280D+00	0.0	0.953D-02
	-0.9325298611D+00	0.0	0.810D-02
176	-0.1000012613D+01	0.0	0.140D-03
	0.9999994313D+00	0.0	0.668D-04
	0.9371856730D+00	0.0	0.322D-03
289	-0.1000000294D+01	0.0	0.335D-05
	0.1000000164D+01	0.0	0.178D-05
	0.9371499768D+00	0.0	0.177D-04
402	-0.1000000001D+01	0.0	0.484D-07
	0.1000000001D+01	0.0	0.447D-07
	0.9371501017D+00	0.0	0.102D-05
495	-0.1000000001D+01	0.0	0.482D-07
	0.1000000000D+01	0.0	0.446D-07
	0.9371501543D+00	0.0	0.252D-07

Table 5.1 Convergence of subspace iteration with projection for computing the three dominant eigenvalues of $A = \text{Mark}(10)$.

3.2. Linear Shifts

Similarly to the power method, there are advantages in working with the shifted matrix $A - \sigma I$ instead of A, where σ is a carefully chosen shift. In fact since the eigenvalues are computed one at a time, the situation is very similar to that of the power method. Thus, when the spectrum is real, and the eigenvalues are ordered decreasingly, the best possible σ is

$$\sigma = \frac{1}{2}(\lambda_{m+1} + \lambda_n)$$

which will put the middle of the unwanted part of the spectrum at the origin. Note that when deflation is used this is independent of the eigenvalue being computed. In addition, we note one important difference with the power method, namely that eigenvalue estimates are now readily available. In fact, it is common practice to take $m > nev$, the number of eigenvalues to be computed, in order to be able to obtain valuable estimates dynamically. These estimates can be used in various ways to accelerate convergence, such as when selecting shifts as indicated above, or when using some of the more sophisticated preconditioning techniques mentioned in the next section.

3.3. Preconditionings

Preconditioning is especially important for subspace iteration, since the unpreconditioned iteration may be unacceptably slow in some cases. Although we will cover preconditioning in more detail in Chapter VIII, we would like to mention here the main ideas used to precondition the subspace iteration.

- Shift-and-invert. This consists of working with the matrix $(A - \sigma I)^{-1}$ instead of A. The eigenvalues near σ will converge fast.

- Polynomial acceleration. The standard method used is to replace the power A^{iter} in the usual subspace iteration algorithm by a polynomial $T_m[(A - \sigma I)/\rho]$ in which T_m is the Chebyshev polynomial of the first kind of degree m.

With either type of preconditioning subspace iteration may be a reasonably efficient method that has the advantage of being easy to code and understand. Some of the methods to be seen in the next Chapter are often preferred however, because they tend to be more economical.

PROBLEMS

P-5.1 In Bauer's original Treppeniteration, the linear independence of the vectors in $A^k X_0$ are preserved by performing its LU decomposition. Thus,

$$\hat{X} = A^k X \quad , \hat{X} = L_k U_k, \quad X := L_k,$$

in which L_k is an $n \times m$ matrix with its upper $m \times m$ corner being a unit lower triangular matrix, and U_k is an $m \times m$ upper triangular matrix. Extend the main convergence theorem of the corresponding algorithm, for this case.

P-5.2 Assume that the matrix A is real and the eigenvalues λ_m, λ_{m+1} forms a complex conjugate pair. If subspace iteration with deflation (Algorithm 5.4) is used, there will be a difficulty when computing the last eigenvalue. Provide a few possible modifications to the algorithm to cope with this case.

P-5.3 Write a modification of Algorithm 5.4 which incorporates a dynamic shifting strategy. Assume that the eigenvalues are real and consider both the case where the rightmost or the leftmost eigenvalues are wanted.

P-5.4 Let A be a matrix whose eigenvalues are real and assume that the subspace iteration algorithm (with projection) is used to compute some of the eigenvalues with largest real parts of A. The question addressed here is how to get the best possible iteration parameter *iter*. We would like to choose *iter* in such a way that in the worst case, the vectors of X will loose a factor of $\sqrt{\epsilon}$ in their linear dependence, in which ϵ is the machine accuracy. How can we estimate such an iteration parameter *iter* from quantities derived from the algorithm? You may assume that m is sufficiently large compared with *nev* (how large should it be?).

P-5.5 Generalize the result of the previous exercise to the case where the eigenvalues are not necessarily real.

P-5.6 Using the Jordan Canonical form, show that for any matrix B,

$$\|B^k\|_2 \leq \alpha \rho^k k^{\eta - 1}, \tag{5.12}$$

where ρ is the spectral radius of B, η the dimension of its largest Jordan block, and α some constant.

P-5.7 Implement a subspace iteration with projection to compute the eigenvalues with largest modulus of a large sparse matrix. Implement locking and linear shifts.

NOTES AND REFERENCES. An early reference on Bauer's Treppeniteration, in addition to the original paper by Bauer [5], is Householder's book [73]. See also the paper by Rutishauser [137] and by Clint and Jennings [21] as well as the book by Bathé and Wilson [4] which all specialize to symmetric matrices. A computer code for the symmetric real case was published in Wilkinson and Reinsch's handbook [184] but unlike most other codes in the handbook, never became part of the Eispack library. Later, some work was done to develop computer codes for the non-Hermitian case. Thus, a 'lop-sided' version of Bauer's treppeniteration based on orthogonal projection method rather than oblique projection was introduced by Jennings and Stewart [77] and a computer code was also made available [78]. However, the corresponding method did not incorporate Chebyshev acceleration, which turned out to be so useful in the Hermitian case. Chebyshev acceleration was later incorporated in the paper by Saad in [143] and some theory was proposed in [141]. G.W. Stewart [169, 170] initiated the idea of using Schur vectors as opposed to eigenvectors in subspace iteration. The motivation is that Schur vectors are easier to handle numerically but there has not been any comparisons in the literature between the two variants. A convergence theory of Subspace Iteration was proposed in [169]. The convergence results of Section 2 follow the paper [141] and a modification due to Chatelin (private communication). There are no public domain codes available as yet implementing the accelerated subspace iteration. Jenning and Stewart's LOPSI code is available in the Transactions for Mathematical Software and can be obtained from Netlib. Quite recently, a Chebyshev accelerated version of subspace iteration has been made available by Rutherford Appleton laboratories [40]. ♠

Chapter VI

Krylov Subspace Methods

In this chapter we will examine one of the most important classes of methods available for computing eigenvalues and eigenvectors of large matrices. These techniques are based on projections methods, both orthogonal and oblique, onto Krylov subpaces, i.e., subspaces spanned by the iterates of the simple power method. What may appear to be a trivial extension of a very slow algorithm turns out to be one of the most successful methods for extracting eigenvalues of large matrices, especially in the Hermitian case.

1. Krylov Subspaces

An important class of techniques known as *Krylov subspace methods* extracts approximations from a subspace of the form

$$\mathcal{K}_m \equiv \text{span } \{v, Av, A^2v, \ldots A^{m-1}v\} \tag{6.1}$$

referred to as a Krylov subspace. If there is a possibility of ambiguity, $\mathcal{K}_m$ is denoted by $\mathcal{K}_m(A, v)$. In contrast with subspace iteration, the dimension of the subspace of approximants increases by one at each step of the approximation process. A few well-known of these *Krylov subspace methods* are:

(1) The Hermitian Lanczos algorithm;

(2) Arnoldi's method and its variations;

(3) The nonhermitian Lanczos algorithm.

There are also block extensions of each of these methods termed *Block Krylov Subspace methods*, which we will discuss only briefly. Arnoldi's method and Lanczos' method are orthogonal projection methods while the nonsymmetric Lanczos algorithm is an oblique projection method. Before we pursue with the analysis of these methods, we would like to emphasize an important distinction between *implementation* of a method and *the method itself*. There are several distinct implementations of Arnoldi's method, which are all mathematically equivalent. For example the articles [42, 139, 177] all propose some different versions of the same mathematical process.

In this section we start by establishing a few elementary properties of Krylov subspaces, many of which need no proof. Recall that the minimal polynomial of a vector v is the nonzero monic polynomial p of lowest degree such that $p(A)v = 0$.

Proposition 6.1 *The Krylov subspace $\mathcal{K}_m$ is the subspace of all vectors in $\mathbb{C}^n$ which can be written as $x = p(A)v$, where p is a polynomial of degree not exceeding $m - 1$.*

Proposition 6.2 *Let μ be the degree of the minimal polynomial of v. Then $\mathcal{K}_\mu$ is invariant under A and $\mathcal{K}_m = \mathcal{K}_\mu$ for all $m \geq \mu$.*

The degree of the minimal polynomial of v is often referred to as the *grade* of v with respect to A. Clearly, the grade of v does not exceed n.

Proposition 6.3 *The Krylov subspace $\mathcal{K}_m$ is of dimension m if and only if the degree of the minimal polynomial of v with respect to A is larger than $m-1$.*

Proof. The vectors $v, Av, \ldots A^{m-1}v$ form a basis of $\mathcal{K}_m$ if and only if for any complex $m-$tuple $\alpha_i, i = 0, \ldots, m-1$, where at least one α_i is nonzero, the linear combination $\sum_{i=0}^{m-1} \alpha_i A^i v$ is nonzero. This condition is equivalent to the condition that there be no polynomial of degree $\leq m-1$ for which $p(A)v = 0$. This proves the result. ■

Proposition 6.4 *Let Q_m be any projector onto $\mathcal{K}_m$ and let A_m be the section of A to $\mathcal{K}_m$, that is, $A_m = Q_m A_{|\mathcal{K}_m}$. Then for any polynomial q of degree not exceeding $m-1$, we have $q(A)v = q(A_m)v$, and for any polynomial of degree $\leq m$, we have $Q_m q(A)v = q(A_m)v$.*

Proof. We will first prove that $q(A)v = q(A_m)v$ for any polynomial q of degree $\leq m-1$. It suffices to prove the property for the monic polynomials $q_i(t) \equiv t^i$, $i = 0, \ldots m-1$. The proof is by induction. The property is clearly true for the polynomial $q_0(t) \equiv 1$. Assume that it is true for $q_i(t) \equiv t^i$:

$$q_i(A)v = q_i(A_m)v.$$

Multiplying the above equation by A on both sides we get

$$q_{i+1}(A)v = Aq_i(A_m)v.$$

If $i+1 \leq m-1$ the vector on the left hand-side belongs to $\mathcal{K}_m$ and therefore if we multiply the above equation on both sides by Q_m we get

$$q_{i+1}(A)v = Q_m A q_i(A_m)v.$$

Looking at the right hand side we observe that $q_i(A_m)v$ belongs to $\mathcal{K}_m$. Hence

$$q_{i+1}(A)v = Q_m A_{|\mathcal{K}_m} q_i(A_m)v = q_{i+1}(A_m)v,$$

which proves that the property is true for $i+1$ provided $i+1 \leq m-1$. For the case $i+1 = m$ it remains only to show that $Q_m q_m(A)v = q_m(A_m)v$, which follows from $q_{m-1}(A)v = q_{m-1}(A_m)v$ by simply multiplying both sides by $Q_m A$. ∎

An interesting characterization of *orthogonal* Krylov projection methods can be formulated in terms of the characteristic polynomial of the approximate problem. In the orthogonal projection case, we define the characteristic polynomial of the approximate problem as that of the matrix $V_m^H A V_m$ where V_m is a matrix whose column vectors form an orthonormal basis of $\mathcal{K}_m$. It is a simple exercise to show that this definition is independent of the choice of V_m, the basis of the Krylov subspace.

Theorem 6.1 *Let $\bar{p}_m$ be the characteristic polynomial of the approximate problem resulting from an orthogonal projection method onto the Krylov subspace K_m. Then $\bar{p}_m$ minimizes the norm $\|p(A)v\|_2$ over all monic polynomials p of degree m.*

Proof. We denote by $\mathcal{P}_m$ the orthogonal projector onto $\mathcal{K}_m$ and A_m the corresponding section of A. By Cayley Hamilton's theorem we have $\bar{p}_m(A_m) = 0$ and therefore

$$(\bar{p}_m(A_m)v, w) = 0, \ \forall \ w \in \mathcal{K}_m \ . \tag{6.2}$$

By the previous proposition $\bar{p}_m(A_m)v = \mathcal{P}_m \bar{p}_m(A)v$. Hence (6.2) becomes

$$(\mathcal{P}_m \bar{p}_m(A)v, w) = 0, \ \forall \ w \in \mathcal{K}_m,$$

or, since orthogonal projectors are self adjoint,

$$(\bar{p}_m(A)v, \mathcal{P}_m w) = 0 = (\bar{p}_m(A)v, w) \quad \forall\ w \in \mathcal{K}_m,$$

which is equivalent to

$$(\bar{p}_m(A)v, A^j v) = 0 \ , \ j = 0, \dots m-1.$$

Writing $\bar{p}_m(t) = t^m - q(t)$, where q is of degree $\leq m-1$, we obtain

$$(A^m v - q(A)v, A^j v) = 0 \ , \ j = 0, \dots m-1.$$

In the above system of equations we recognize the normal equations for minimizing the Euclidean norm of $A^m v - s(A)v$ over all polynomials s of degree $\leq m-1$. The proof is complete. ∎

The above characteristic property is not intended to be used for computational purposes. It is useful for establishing mathematical equivalences between seemingly different methods. Thus, a method developed by Erdelyi in 1965 [42] is based on precisely minimizing $\|p(A)v\|_2$ over monic polynomials of some degree and is therefore mathematically equivalent to any orthogonal projection method on a Krylov subspace. Another such method was proposed by Manteuffel [99, 100] for the purpose of estimating acceleration parameters when solving linear systems by Chebyshev method. His method named the Generalized Power Method, was essentially Erdelyi's method with a special initial vector.

An important point is that this characteristic property seems to be the only known optimality property that is satisfied by the approximation process in the nonsymmetric case. Other optimality properties, such as the mini-max theorem which are fundamental both in theory and in practice for symmetric problems are no longer valid. This results in some significant difficulties in understanding and analyzing these methods for nonsymmetric eigenvalue problems.

2. Arnoldi's Method

Arnoldi's method is an orthogonal projection method onto $\mathcal{K}_m$ for general non-Hermitian matrices. The procedure was introduced in 1951 as a means of reducing a dense matrix into Hessenberg form. Arnoldi introduced this method precisely in this manner and he hinted that the process could give good approximations to some eigenvalues if stopped before completion. It was later discovered that this strategy lead to a good technique for approximating eigenvalues of large sparse matrices. We first describe the method without much regard to rounding errors, and then give a few implementation details.

2.1. The Basic Algorithm

The procedure introduced by Arnoldi in 1951 starts by building an orthogonal basis of the Krylov subspace $\mathcal{K}_m$. In exact arithmetic, one variant of the algorithm is as follows.

ALGORITHM **6.1 Arnoldi**

1. **Start:** *Choose a vector* v_1 *of norm 1.*
2. **Iterate:** *for* $j = 1, 2, \ldots, m$ *compute:*

$$h_{ij} = (Av_j, v_i), \quad i = 1, 2, \ldots, j, \tag{6.3}$$

$$w_j = Av_j - \sum_{i=1}^{j} h_{ij} v_i, \tag{6.4}$$

$$h_{j+1,j} = \|w_j\|_2 \,, \quad \textit{if} \;\; h_{j+1,j} = 0 \;\; \textit{stop} \tag{6.5}$$

$$v_{j+1} = w_j / h_{j+1,j}. \tag{6.6}$$

The algorithm will stop if the vector w_j computed in (6.4) vanishes. We will come back to this case shortly. We now prove a few simple but important properties of the algorithm.

Proposition 6.5 *The vectors* $v_1, v_2, \ldots, v_m$ *form an orthonormal basis of the subspace* $\mathcal{K}_m = \mathrm{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$.

Proof. The vectors $v_j, i = 1, 2, \ldots m$ are orthonormal by construction. That they span $\mathcal{K}_m$ follows from the fact that each vector v_j is of the form $q_{j-1}(A)v_1$ where q_{j-1} is a polynomial of degree $j-1$. This can be shown by induction on j as follows. Clearly the result is true when $j = 1$, since $v_1 = q_0(A)v_1$ with $q_0(t) \equiv 1$. Assume that the result is true for all integers $\leq j$ and consider v_{j+1}. We have

$$h_{j+1}v_{j+1} = Av_j - \sum_{i=1}^{j} h_{ij}v_i = Aq_{j-1}(A)v_1 - \sum_{i=1}^{j} h_{ij}q_{i-1}(A)v_1 \quad (6.7)$$

which shows that v_{j+1} can be expressed as $q_j(A)v_1$ where q_j is of degree j and completes the proof. ∎

Proposition 6.6 *Denote by V_m the $n \times m$ matrix with column vectors $v_1, \ldots, v_m$ and by H_m the $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations hold:*

$$AV_m = V_m H_m + h_{m+1,m}v_{m+1}e_m^H, \quad (6.8)$$
$$V_m^H A V_m = H_m \; . \quad (6.9)$$

Proof. The relation (6.8) follows from the following equality which is readily derived from (6.6) and (6.4):

$$Av_j = \sum_{i=1}^{j+1} h_{ij}v_i, \quad j = 1, 2, \ldots, m \; . \quad (6.10)$$

Relation (6.9) follows by multiplying both sides of (6.8) by V_m^H and making use of the orthonormality of $\{v_1, \ldots, v_m\}$. ∎

The situation is illustrated in Figure 6.1.

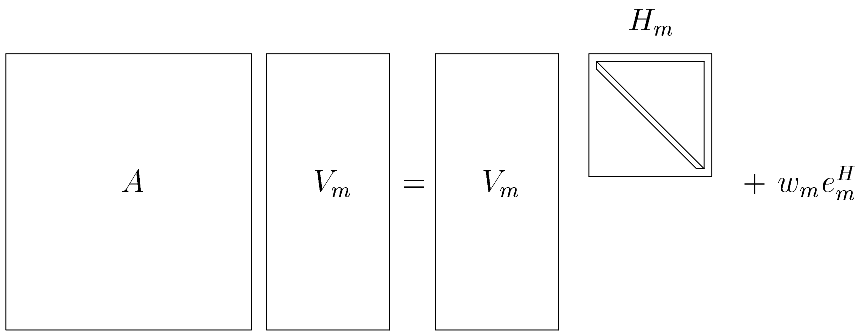

Figure 6.1 The action of A on V_m gives $V_m H_m$ plus a rank one matrix.

As was noted earlier the algorithm may break down in case the norm of w_j vanishes at a certain step j. In this situation the vector v_{j+1} cannot be computed and the algorithm stops. There remains to determine the conditions under which this situation occurs.

Proposition 6.7 *Arnoldi's algorithm breaks down at step j (i.e., $w_j = 0$ in (6.4)) if and only if the minimal polynomial of v_1 is of degree j. Moreover, in this case the subspace $\mathcal{K}_j$ is invariant and the approximate eigenvalues and eigenvectors are exact.*

Proof. If the degree of the minimal polynomial is j, then w_j must be equal to zero. Indeed, otherwise v_{j+1} can be defined and as a result $\mathcal{K}_{j+1}$ would be of dimension $j+1$, and from Proposition 6.3, this would mean that $\mu \geq j+1$, which is not true. To prove the converse, assume that $w_j = 0$. Then the degree μ of the minimal polynomial of v_1 is such that $\mu \leq j$. Moreover, we cannot have $\mu < j$ otherwise by the previous proof the vector w_μ would be zero and the algorithm would have stopped at the earlier step number μ. The rest of the result follows from Proposition 4.3 seen in Chapter IV. ∎

The approximate eigenvalues $\lambda_i^{(m)}$ provided by the projection process onto $\mathcal{K}_m$ are the eigenvalues of the Hessenberg matrix H_m. The Ritz approximate eigenvector associated with $\lambda_i^{(m)}$ is defined by $u_i^{(m)} = V_m y_i^{(m)}$ where $y_i^{(m)}$ is an eigenvector associated with the eigenvalue $\lambda_i^{(m)}$. A number of the Ritz eigenvalues, typically a small fraction of m, will usually constitute good approximations of corresponding eigenvalues λ_i of A and the quality of the approximation will usually improve as m increases. We will examine these 'convergence' properties in detail in later sections. The original algorithm consists of increasing m until all desired eigenvalues of A are found. This is costly both in terms of computation and storage. For storage, we need to keep m vectors of length n plus an $m \times m$ Hessenberg matrix, a total of approximately $nm + m^2/2$. Considering the computational cost of the j-th step, we need to multiply v_j by A, at the cost of $2 \times Nz$, where Nz is number of nonzero elements in A, and then orthogonalize the result against j vectors at the cost of $4(j+1)n$, which increases with the step number j.

On the practical side it is crucial to be able to estimate the residual norm inexpensively as the algorithm progresses. This turns out to be quite easy to do for Arnoldi's method and, in fact, for all the Krylov subspace methods described in this chapter. The result is given in the next proposition.

Proposition 6.8 *Let $y_i^{(m)}$ be an eigenvector of H_m associated with the eigenvalue $\lambda_i^{(m)}$ and $u_i^{(m)}$ the Ritz approximate eigenvector $u_i^{(m)} = V_m y_i^{(m)}$. Then,*

$$(A - \lambda_i^{(m)} I) u_i^{(m)} = h_{m+1,m}\, e_m^H y_i^{(m)} v_{m+1}$$

and, therefore,

$$\|(A - \lambda_i^{(m)} I) u_i^{(m)}\|_2 = h_{m+1,m} |e_m^H y_i^{(m)}| \ .$$

Proof. This follows from multiplying both sides of (6.8) by $y_i^{(m)}$:

$$\begin{aligned} AV_m y_i^{(m)} &= V_m H_m y_i^{(m)} + h_{m+1,m}\, e_m^H y_i^{(m)} v_{m+1} \\ &= \lambda_i^{(m)} V_m y_i^{(m)} + h_{m+1,m} e_m^H y_i^{(m)} v_{m+1} \ . \end{aligned}$$

Hence,

$$AV_m y_i^{(m)} - \lambda_i^{(m)} V_m y_i^{(m)} = h_{m+1,m}\, e_m^H y_i^{(m)}\ v_{m+1} \ .$$

∎

In simpler terms, the proposition states that the residual norm is equal to the last component of the eigenvector $y_i^{(m)}$ multiplied by $h_{m+1,m}$. In practice, the residual norms, although not always indicative of actual errors, are quite helpful in deriving stopping procedures.

2.2. Practical Implementations

The description of the Arnoldi process given earlier assumed exact arithmetic. In reality, much is to be gained by using the Modified Gram-Schmidt or the Householder algorithm in place of the standard Gram-Schmidt algorithm. With the modified Gram-Schmidt alternative the algorithm takes the following form.

ALGORITHM **6.2 Arnoldi - Modified Gram-Schmidt**

1. **Start.** *Choose a vector v_1 of norm 1.*
2. **Iterate.** *For $j = 1, 2, \ldots, m$ do:*
   (a) $w := Av_j$;
   (b) For $i = 1, 2, \ldots, j$ do:

$$\begin{aligned} &h_{ij} = (w, v_i), \\ &w := w - h_{ij} v_i; \end{aligned}$$

(c) $h_{j+1,j} = \|w\|_2$;

(d) $v_{j+1} = w/h_{j+1,j}$.

There is no difference in exact arithmetic between this algorithm and Algorithm 6.1. Although this formulation is numerically superior to the standard Gram Schmidt formulation, we do not mean to imply that the above Modified Gram-Schmidt is sufficient for all cases. In fact there are two alternatives that are implemented to guard against large cancellations during the orthogonalization process.

The first alternative is to resort to double orthogonalization. Whenever the final vector obtained at the end of the second loop in the above algorithm has been computed, a test is performed to compare its norm with the norm of the initial w (which is $\|Av_j\|_2$). If the reduction falls below a certain threshold, an indication that sever cancellation might have occurred, a second orthogonalization is made. It is known from a result by Kahan that additional orthogonalizations are superfluous (see for example Parlett [118]).

The second alternative is to resort to a different technique altogether. In fact one of the most reliable orthogonalization techniques, from the numerical point of view, is the Householder algorithm. This has been implemented for the Arnoldi process by Walker [181]. We do not describe the Householder algorithm here but we would like to compare the cost of each of the three versions.

In the table shown below, GS stands for Gram-Schmidt, MGS for Modified Gram-Schmidt, MGSR for Modified Gram-Schmidt with Reorthogonalization, and HO for Householder.

	GS	MGS	MGSR	HO
Flops	m^2n	m^2n	$2m^2n$	$2m^2n - \frac{2}{3}m^3$
Storage	$(m+1)n$	$(m+1)n$	$(m+1)n$	$(m+1)n - \frac{1}{2}m^2$

A few comments are in order. First, the number of operations shown for MGSR are for the worst case situation when a second orthogonalization is needed every time. This is unlikely to take place and in practice the actual number of operations is much more likely to be close to that of the simple MGS. Concerning storage, the little gain in storage requirement in the Householder version comes from the fact that the Householder transformation requires vectors whose length diminishes by 1 at every step of the process. However, this difference is negligible relative to the whole storage requirement given that usually $m \ll n$. Moreover, the implementation to take advantage of this little gain may become rather complicated. In spite of this we do recommend implementing Householder orthogonalization for developing general purpose reliable software packages. A little additional cost in arithmetic may be more than offset by the gains in robustness in these conditions.

Example 6.1 Consider the matrix Mark(10) used in the examples in the previous two Chapters. Table 6.1 shows the convergence of the rightmost eigenvalue obtained by Arnoldi's method.

m	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
5	0.9027159373	0.0	0.316D+00
10	0.9987435899	0.0	0.246D-01
15	0.9993848488	0.0	0.689D-02
20	0.9999863880	0.0	0.160D-03
25	1.000000089	0.0	0.135D-05
30	0.9999999991	0.0	0.831D-08

Table 6.1 Convergence of rightmost eigenvalue computed from a simple Arnoldi algorithm for $A = \text{Mark}(10)$.

Comparing the results shown in Table 6.1 with those of the examples seen in Chapter IV, it is clear that the convergence is much faster than the power method or the shifted power method.

As was mentioned earlier the standard implementations of Arnoldi's method are limited by their high storage and computational requirements as m increases. Suppose that we are interested in only one eigenvalue/eigenvector of A, namely the eigenvalue of largest real part of A. Then one way to circumvent the difficulty is to *restart* the algorithm. After a run with m Arnoldi vectors, we compute the approximate eigenvector and use it as an initial vector for the next run with Arnoldi's method. This process, which is the simplest of this kind, is iterated to convergence.

ALGORITHM **6.3 Iterative Arnoldi**

1. **Start:** *Choose an initial vector* v_1 *and a dimension* m.
2. **Iterate:** *Perform* m *steps of Arnoldi's algorithm.*
3. **Restart:** *Compute the approximate eigenvector* $u_1^{(m)}$ *associated with the rightmost eigenvalue* $\lambda_1^{(m)}$. *If satisfied stop, else set* $v_1 \equiv u_1^{(m)}$ *and goto 2.*

Example 6.2 Consider the same matrix Mark(10) as above. We now use a restarted Arnoldi procedure for computing the eigenvector associated with the eigenvalue with algebraically largest real part. We use $m = 10$.

m	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
10	0.9987435899D+00	0.0	0.246D-01
20	0.9999523324D+00	0.0	0.144D-02
30	0.1000000368D+01	0.0	0.221D-04
40	0.1000000025D+01	0.0	0.508D-06
50	0.9999999996D+00	0.0	0.138D-07

Table 6.2 Convergence of rightmost eigenvalue computed from a restarted Arnoldi procedure for $A = \text{Mark}(10)$.

Comparing the results of Table 6.2 with those of the previous example indicates a loss in performance, in terms of total number of matrix-vector products. However, the number of vectors used here is 10 as opposed to 50, so the memory requirement is much more modest.

2.3. Incorporation of Implicit Deflation

We now consider the following implementation which incorporates a deflation process. The previous algorithm is valid only for the case where only one eigenvalue/eigenvector pair must be computed. In case several such pairs must be computed, then there are two possible options. The first, is to take v_1 to be a linear combination of the approximate eigenvectors when we restart. For example, if we need to compute the p rightmost eigenvectors, we may take

$$\hat{v}_1 = \sum_{i=1}^{p} \rho_i \tilde{u}_i,$$

where the eigenvalues are numbered in decreasing order of their real parts. The vector v_1 is then obtained from normalizing $\hat{v}_1$. The simplest choice for the coefficients ρ_i is to take $\rho_i = 1, i = 1, \ldots, p$. There are several drawbacks to this approach, the most important of which being that there is no easy way of choosing the coefficients ρ_i in a systematic manner. The result is that for hard problems, convergence is difficult to achieve.

An alternative is to compute one eigenpair at a time and use deflation. We can use deflation on the matrix A explicitly as was described in Chapter IV. This entails constructing progressively the first k Schur vectors. If a previous orthogonal basis $[u_1, \ldots, u_{k-1}]$ of the invariant subspace has already been computed, then, to compute the eigenvalue λ_k, we work with the matrix $A - U\Sigma U^H$, in which Σ is a diagonal matrix.

Another implementation, which we now describe, is to work with a single basis $v_1, v_2, ..., v_m$ whose first vectors are the Schur vectors that have already converged. Suppose that $k-1$ such vectors have converged and call them $v_1, v_2, ..., v_{k-1}$. Then we start by choosing a vector v_k which is orthogonal to $v_1, ..., v_{k-1}$ and of norm 1. Next we perform $m-k$ steps of an Arnoldi process in which orthogonality of the vector v_j against all previous $v_i's$, including $v_1, ..., v_{k-1}$ is enforced. This generates an orthogonal basis of the subspace

$$\text{span}\{v_1, \ldots, v_{k-1}, v_k, Av_k, \ldots, A^{m-k}v_k\} \ . \tag{6.11}$$

Thus, the dimension of this modified Krylov subspace is constant and equal to m in general. A sketch of this implicit deflation procedure combined with Arnoldi's method is the following.

ALGORITHM **6.4 Deflated Iterative Arnoldi**

A. Start: *Choose an initial vector v_1 of norm unity. Set $k := 1$.*

B. Eigenvalue loop:

1. *Arnoldi Iteration. For $j = k, k+1, ..., m$ do:*
   - *Compute $w := Av_j$.*
   - *Compute a set of j coefficients h_{ij} so that $w := w - \sum_{i=1}^{j} h_{ij} v_i$ is orthogonal to all previous v_i's, $i = 1, 2, ..., j$.*
   - *Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.*
2. *Compute approximate eigenvector of A associated with the eigenvalue $\tilde{\lambda}_k$ and its associated residual norm estimate ρ_k.*
3. *Orthonormalize this eigenvector against all previous v_j's to get the approximate Schur vector $\tilde{u}_k$ and define $v_k := \tilde{u}_k$.*
4. *If ρ_k is small enough then (accept eigenvalue):*
   - *Compute $h_{i,k} = (Av_k, v_i)$, $i = 1, .., k$,*
   - *Set $k := k + 1$,*
   - *If $k \geq nev$ then stop else goto B.*
5. *Else go to B-1.*

Note that in the B-loop, the Schur vectors associated with the eigenvalues $\lambda_1, ..., \lambda_{k-1}$ are frozen and so is the corresponding upper triangular matrix corresponding to these vectors. As a new Schur vector has converged, step B.4 computes the k-th column of R associated with this new basis vector. In the subsequent steps, the approximate eigenvalues are the eigenvalues of the $m \times m$ Hessenberg matrix H_m defined in the algorithm and whose $k \times k$ principal submatrix is upper triangular For example when $m = 6$ and after the second Schur vector, $k = 2$, has converged, the matrix H_m will have the form

$$H_m = \begin{pmatrix} * & * & * & * & * & * \\ & * & * & * & * & * \\ & & * & * & * & * \\ & & * & * & * & * \\ & & & * & * & * \\ & & & & * & * \end{pmatrix}. \tag{6.12}$$

Therefore in the subsequent steps, we will consider only the eigenvalues that are not associated with the 2×2 upper triangular matrix.

It can be shown that, in exact arithmetic, the $(n-k) \times (n-k)$ Hessenberg matrix in the lower (2×2) block is the same matrix that would be obtained from an Arnoldi run applied to the matrix $(I - P_k)A$ in which P_k is the orthogonal projector onto the (approximate) invariant subspace that has already been computed, see Exercise P-6.3. The above algorithm although not competitive with the more elaborate versions that use some form of preconditioning, will serve as a good model of a deflation process combined with Arnoldi's projection.

Example 6.3 We will use once more the test matrix Mark(10) for illustration. Here we test our restarted and deflated Arnoldi procedure for computing the three eigenvalues with algebraically largest real part. We use $m = 10$ as in the previous example. We do not show the run corresponding to the first eigenvalue since the data is already listed in Table 6.2. The first column shows the eigenvalue being computed. Thus, it takes five outer iterations to compute the first eigenvalue (see example 6.2), 4 outer iterations to compute the second one, and finally

8 outer iterations to get the third one. The convergence towards the last eigenvalue is slower than for the first two. This could be attributed to poorer separation of λ_3 from the other eigenvalues but also to the fact that m has implicitly decreased from $m = 10$ when computing the first eigenvalue to $m = 8$ when computing the third one.

Eig.	Mat-Vec's	$\Re e(\lambda)$	$\Im m(\lambda)$	Res. Norm
2	60	0.9370509474	0.0	0.870D-03
	69	0.9371549617	0.0	0.175D-04
	78	0.9371501442	0.0	0.313D-06
	87	0.9371501564	0.0	0.490D-08
3	96	0.8112247133	0.0	0.210D-02
	104	0.8097553450	0.0	0.538D-03
	112	0.8096419483	0.0	0.874D-04
	120	0.8095810281	0.0	0.181D-04
	128	0.8095746489	0.0	0.417D-05
	136	0.8095721868	0.0	0.753D-06
	144	0.8095718575	0.0	0.231D-06
	152	0.8095717167	0.0	0.444D-07

Table 6.3 Convergence of three rightmost eigenvalues computed from a deflated Arnoldi procedure for $A = \mathrm{Mark}(10)$.

3. The Hermitian Lanczos Algorithm

The Hermitian Lanczos algorithm can be viewed as a simplification of Arnoldi's method for the particular case when the matrix is Hermitian. The principle of the method is therefore the same in that it is a projection technique on a Krylov subspace. However, there are a number of interesting properties that will cause the algorithm to simplify. On the theoretical side there is also much more that can be said on the Lanczos algorithm than there is on Arnoldi's method.

3.1. The Algorithm

To introduce the algorithm we start by making the observation stated in the following theorem.

Theorem 6.2 *Assume that Arnoldi's method is applied to a Hermitian matrix A. Then the coefficients h_{ij} generated by the algorithm are real and such that*

$$h_{ij} = 0, \quad \text{for} \quad 1 \le i < j-1 \ , \tag{6.13}$$

$$h_{j,j+1} = h_{j+1,j} \ , \ j = 1, 2, \ldots, m. \tag{6.14}$$

In other words the matrix H_m obtained from the Arnoldi process is real, tridiagonal, and symmetric.

Proof. The proof is an immediate consequence of the fact that $H_m = V_m^H A V_m$ is a Hermitian matrix which is also a Hessenberg matrix by construction. Therefore, H_m must be a Hermitian tridiagonal matrix. In addition, observe that by its definition the scalar $h_{j+1,j}$ is real and that $h_{jj} = (Av_j, v_j)$ is also real if A is Hermitian. Therefore, the Hessenberg matrix H_m is a real tridiagonal and symmetric matrix. ∎

The standard notation used to describe the Lanczos algorithm, is obtained by setting

$$\begin{aligned} \alpha_j &\equiv h_{jj} \ , \\ \beta_j &\equiv h_{j-1,j} \ , \end{aligned}$$

which leads to the following form of the Modified Gram Schmidt variant of Arnoldi's method, namely Algorithm 6.2.

ALGORITHM **6.5 The Lanczos Algorithm**

1. **Start:** *Choose an initial vector v_1 of norm unity. Set $\beta_1 \equiv 0, v_0 \equiv 0$.*

2. **Iterate:** *for* $j = 1, 2, \ldots, m$ *do*

$$w_j := Av_j - \beta_j v_{j-1} \tag{6.15}$$

$$\alpha_j := (w_j, v_j) \tag{6.16}$$

$$w_j := w_j - \alpha_j v_j \tag{6.17}$$

$$\beta_{j+1} := \|w_j\|_2 \tag{6.18}$$

$$v_{j+1} := w_j / \beta_{j+1} \tag{6.19}$$

An important and rather surprising property is that the above simple algorithm guarantees, at least in exact arithmetic, that the vectors $v_i, i = 1, 2, \ldots,$ are orthogonal. In reality, exact orthogonality of these vectors is only observed at the beginning of the process. Ultimately, the v_i's start losing their global orthogonality very rapidly. There has been much research devoted to finding ways to either recover the orthogonality, or to at least diminish its effects by *partial* or *selective* orthogonalization, see Parlett [118].

The major practical differences with Arnoldi's method are that the matrix H_m is tridiagonal and, more importantly, that we only need to save three vectors, at least if we do not resort to any form of reorthogonalization.

3.2. Relation with Orthogonal Polynomials

In exact arithmetic the equation (6.17) in the algorithm takes the form

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

This three term recurrence relation is reminiscent of the standard three term recurrence relation of orthogonal polynomials. In fact as we will show in this section, there is indeed a strong relationship between the Lanczos algorithm and orthogonal polynomials. We start by recalling that if the grade of v_1 is $\geq m$ then the subspace $\mathcal{K}_m$ is of dimension m and consists of all vectors of the form $q(A)v_1$ with degree$(q) \leq m-1$. In this case there is even an isomorphism between $\mathcal{K}_m$ and $\mathbb{P}_{m-1}$, the space of polynomials of degree $\leq$

$m-1$, which is defined by

$$q \in \mathbb{P}_{m-1} \rightarrow x = q(A)v_1 \in \mathcal{K}_m$$

Moreover, we can consider that the subspace $\mathbb{P}_{m-1}$ is provided with the inner product

$$< p, q >_{v_1} = (p(A)v_1, q(A)v_1) \tag{6.20}$$

which is indeed a nondegenerate bilinear form under the assumption that m does not exceed μ, the grade of v_1. Now observe that the vectors v_i are of the form

$$v_i = q_{i-1}(A)v_1$$

and the orthogonality of the v_i's translates into the orthogonality of the polynomials with respect to the inner product (6.20). Moreover, the Lanczos procedure is nothing but the Stieltjes algorithm (see, for example, Gautschi [55]) for computing a sequence of orthogonal polynomials with respect to the inner product (6.20). From Theorem 6.1 the characteristic polynomial of the tridiagonal matrix produced by the Lanczos algorithm minimizes the norm $\|.\|_{v_1}$ over the monic polynomials. It is easy to prove by using a well-known recurrence for determinants of tridiagonal matrix, that the Lanczos recurrence computes the characteristic polynomial of H_m times the initial vector v_1. This is another way of relating the v_i's to the orthogonal polynomials.

4. Non-Hermitian Lanczos algorithm

This is an extension of the algorithm seen in the previous section to the non-Hermitian case. We already know of one such extension namely Arnoldi's procedure which is an orthogonal projection method. However, the non-Hermitian Lanczos algorithm is an oblique projection technique and is quite different in concept from Arnoldi's method.

4.1. The Algorithm

The algorithm proposed by Lanczos for non-Hermitian matrices differs from Arnoldi's method in one essential way: instead of building an orthogonal basis of $\mathcal{K}_m$, it builds a pair of biorthogonal bases for the two subspaces

$$\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, \ldots, A^{m-1}v_1\}$$

and

$$\mathcal{K}_m(A^H, w_1) = \text{span}\{w_1, A^H w_1, \ldots, (A^H)^{m-1}w_1\}.$$

The algorithm to achieve this is as follows.

ALGORITHM **6.6 The non-Hermitian Lanczos Algorithm**

1. **Start:** *Choose two vectors* v_1, w_1 *such that* $(v_1, w_1) = 1$. *Set* $\beta_1 \equiv 0, w_0 = v_0 \equiv 0$.

2. **Iterate:** *for* $j = 1, 2, \ldots, m$ *do*

$$\alpha_j = (Av_j, w_j) \tag{6.21}$$
$$\hat{v}_{j+1} = Av_j - \alpha_j v_j - \beta_j v_{j-1} \tag{6.22}$$
$$\hat{w}_{j+1} = A^H w_j - \bar{\alpha}_j w_j - \delta_j w_{j-1} \tag{6.23}$$
$$\delta_{j+1} = |(\hat{v}_{j+1}, \hat{w}_{j+1})|^{1/2} \tag{6.24}$$
$$\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1})/\delta_{j+1} \tag{6.25}$$
$$w_{j+1} = \hat{w}_{j+1}/\overline{\beta_{j+1}} \tag{6.26}$$
$$v_{j+1} = \hat{v}_{j+1}/\delta_{j+1} \tag{6.27}$$

We should point out that there is an infinity of ways of choosing the scalars $\delta_{j+1}, \beta_{j+1}$ in (6.24)–(6.25). These two parameters are scaling factors for the two vectors v_{j+1} and w_{j+1} and can be selected in any manner to ensure that $(v_{j+1}, w_{j+1}) = 1$. As a result of (6.26), (6.27) all that is needed is to choose two scalars $\beta_{j+1}, \delta_{j+1}$ that satisfy the equality

$$\delta_{j+1}\beta_{j+1} = (\hat{v}_{j+1}, \hat{w}_{j+1}) \tag{6.28}$$

The choice made in the above algorithm attempts to scale the two vectors so that they are divided by two scalars having the same modulus. Thus, if initially v_1 and w_1 have the same norm, all of the subsequent v_i's will have the same norms as the w_i's. One can scale both vectors by their 2-norms, so that the inner product of v_i and w_i is no longer equal to one. A modified algorithm can be written with these constraint. In this situation a generalized eigenvalue problem $T_m z = \lambda D_m z$ must be solved to compute the Ritz values where D_m is a diagonal matrix, whose entries are the inner products (v_i, w_i). The modified algorithm is the subject of Exercise P-6.9.

In what follows we will place ourselves in the situation where the pair of scalars $\delta_{j+1}, \beta_{j+1}$ is *any pair that satisfies the relation* (6.28), instead of restricting ourselves to the particular case defined by (6.24) – (6.25). A consequence is that δ_j can be complex and in fact the formula defining $\hat{w}_{j+1}$ in (6.23) should then be modified to

$$\hat{w}_{j+1} = A^H w_j - \bar{\alpha}_j w_j - \bar{\delta}_j w_{j-1} \ .$$

We will denote by T_m the tridiagonal matrix

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & . & . & . & \\ & & \delta_{m-1} & \alpha_{m-1} & \beta_m \\ & & & \delta_m & \alpha_m \end{pmatrix} .$$

Note that in the particular case where A is real as well as the initial vectors v_1, w_1, and if (6.24) – (6.25) are used then the δ_j's are real positive and $\beta_j = \pm\delta_j$.

Our first observation from the algorithm is that the vectors v_i belong to $\mathcal{K}_m(A, v_1)$ while the w_j 's are in $\mathcal{K}_m(A^H, w_1)$. In fact we can show the following proposition.

Proposition 6.9 *If the algorithm does not break down before step m then the vectors $v_i, i = 1, \ldots, m$, and $w_j, j = 1, \ldots, m$, form a* *biorthogonal system, i.e.,*

$$(v_j, w_i) = \delta_{ij} \quad 1 \leq i,\ j \leq m\ .$$

Moreover, $\{v_i\}_{i=1,2,\ldots,m}$ *is a basis of* $\mathcal{K}_m(A, v_1)$ *and* $\{w_i\}_{i=1,2,\ldots,m}$ *is a basis of* $\mathcal{K}_m(A^H, w_1)$ *and we have the relations,*

$$AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^H, \tag{6.29}$$
$$A^H W_m = W_m T_m^H + \bar{\beta}_{m+1} w_{m+1} e_m^H, \tag{6.30}$$
$$W_m^H A V_m = T_m \quad . \tag{6.31}$$

Proof. The biorthogonality of the vectors v_i, w_i will be shown by induction. By assumption $(v_1, w_1) = 1$. Assume now that the vectors $v_1, \ldots v_j$ and $w_1, \ldots w_j$ are biorthogonal, and let us establish that the vectors $v_1, \ldots v_{j+1}$ and $w_1, \ldots w_{j+1}$ are biorthogonal.

We show first that $(v_{j+1}, w_i) = 0$ for $i \leq j$. When $i = j$ we have

$$(v_{j+1}, w_j) = \delta_{j+1}^{-1}[(Av_j, w_j) - \alpha_j (v_j, w_j) - \beta_j (v_{j-1}, w_j)] \quad .$$

The last inner product in the above expression vanishes by the induction hypothesis. The two other terms cancel each other by the definition of α_j and the fact that $(v_j, w_j) = 1$. Consider now

$$(v_{j+1}, w_{j-1}) = \delta_{j+1}^{-1}[(Av_j, w_{j-1}) - \alpha_j (v_j, w_{j-1}) - \beta_j (v_{j-1}, w_{j-1})]\ .$$

Again from the induction hypothesis the middle term in the right hand side vanishes. The first term can be rewritten as

$$\begin{aligned} (Av_j, w_{j-1}) &= (v_j, A^H w_{j-1}) \\ &= (v_j, \bar{\beta}_j w_j + \bar{\alpha}_{j-1} w_{j-1} + \bar{\delta}_{j-1} w_{j-2}) \\ &= \beta_j (v_j, w_j) + \alpha_{j-1} (v_j, w_{j-1}) + \delta_{j-1} (v_j, w_{j-2}) \\ &= \beta_j \end{aligned}$$

and as a result,

$$(v_{j+1}, w_{j-1}) = \delta_{j+1}^{-1}[(Av_j, w_{j-1}) - \beta_j (v_{j-1}, w_{j-1})] = 0\ .$$

More generally, consider an inner product (v_{j+1}, w_i) with $i < j-1$,

$$\begin{aligned}
(v_{j+1}, w_i) &= \delta_{j+1}^{-1}[(Av_j, w_i) - \alpha_j(v_j, w_i) - \beta_j(v_{j-1}, w_i)] \\
&= \delta_{j+1}^{-1}[(v_j, A^H w_i) - \alpha_j(v_j, w_i) - \beta_j(v_{j-1}, w_i)] \\
&= \delta_{j+1}^{-1}[(v_j, \bar{\beta}_{i+1} w_{i+1} + \bar{\alpha}_i w_i + \bar{\delta}_i w_{i-1}) - \alpha_j(v_j, w_i) \\
&\qquad - \beta_j(v_{j-1}, w_i)] \; .
\end{aligned}$$

By the induction hypothesis, all of the inner products in the above expression vanish. We can show in the same way that $(v_i, w_{j+1}) = 0$ for $i \leq j$. Finally, we have by construction $(v_{j+1}, w_{j+1}) = 1$. This completes the induction proof.

The proof of the other matrix relations is identical with the proof of the similar relations in Arnoldi's method. ∎

The relation (6.31) is key to understanding the nature of the method. From what we have seen in Chapter IV on general projection methods, the matrix T_m is exactly the projection of A obtained from an oblique projection process onto $\mathcal{K}_m(A, v_1)$ and orthogonally to $\mathcal{K}_m(A^H, w_1)$. The approximate eigenvalues $\lambda_i^{(m)}$ provided by this projection process are the eigenvalues of the tridiagonal matrix T_m. A Ritz approximate eigenvector of A associated with $\lambda_i^{(m)}$ is defined by $u_i^{(m)} = V_m y_i^{(m)}$ where $y_i^{(m)}$ is an eigenvector associated with the eigenvalue $\lambda_i^{(m)}$ of T_m. Similarly to Arnoldi's method, a number of the Ritz eigenvalues, typically a small fraction of m, will constitute good approximations of corresponding eigenvalues λ_i of A and the quality of the approximation will improve as m increases.

We should mention that the result of Proposition 6.8, which gives a simple and inexpensive way to compute residual norms can readily be extended as follows:

$$(A - \lambda_i^{(m)} I) u_i^{(m)} = \delta_{m+1} e_m^H y_i^{(m)} v_{m+1} \tag{6.32}$$

and, as a result $\|(A - \lambda_i^{(m)} I) u_i^{(m)}\|_2 = |\delta_{m+1} e_m^H y_i^{(m)}|$.

An interesting new feature here is that the operators A and A^H play a dual role in that we perform similar operations with them. We can therefore expect that if we get good approximate eigenvectors for A we should in general get as good approximations for the eigenvectors of A^H. In fact we might also view the non-Hermitian Lanczos procedure as a method for approximating eigenvalues and eigenvectors of the matrix A^H by a projection method onto $L_m = \mathrm{span}\{w_1, A^H w_1, \ldots, (A^H)^{m-1} w_1\}$ and orthogonally to $\mathcal{K}_m(A, v_1)$. As a consequence, both the left and right eigenvectors of A will be well approximated by the process. In contrast Arnoldi's method only computes approximations to the right eigenvectors. The approximations to the left eigenvectors are of the form $W_m z_i^{(m)}$ where $z_i^{(m)}$ is a left eigenvector of T_m associated with the eigenvalue $\lambda_i^{(m)}$. This constitutes one of the major differences between the two methods. There are applications where both left and right eigenvectors may be needed. In addition, when estimating errors and condition numbers of the computed eigenpair it might be crucial that both the left and the right eigenvectors be available.

From the practical point of view, another big difference between the non-Hermitian Lanczos procedure and the Arnoldi methods is that we now only need to save a few vectors in memory to execute the algorithm if no reorthogonalization is performed. More precisely, we need 6 vectors of length n plus some storage for the tridiagonal matrix, no matter how large m is. This is clearly a significant advantage.

On the other hand there are more risks of breakdown with the non-Hermitian Lanczos method. The algorithm will break down whenever $(\hat{v}_{j+1}, \hat{w}_{j+1}) = 0$ which can be shown to be equivalent to the existence of a vector in $\mathcal{K}_m(A, v_1)$ that is orthogonal to the subspace $\mathcal{K}_m(A^H, w_1)$. In fact this was seen to be a necessary and sufficient condition for the oblique projector onto $\mathcal{K}_m(A, v_1)$ orthogonally to $\mathcal{K}_m(A^H, w_1)$ not to exist. In the case of Arnoldi's method a breakdown is actually a favorable situation since we are guaranteed to obtain exact eigenvalues in this case as was seen before. The same is true in the case of the Lanczos algorithm when either $\hat{v}_{j+1} = 0$ or $\hat{w}_{j+1} = 0$. However, when $\hat{v}_{j+1} \neq 0$

and $\hat{w}_{j+1} \neq 0$ then this is non-longer true. In fact the serious problem is not as much caused by the exact occurrence of this phenomenon which Wilkinson [183] calls *serious breakdown*, as it is its near occurrence. A look at the algorithm indicates that we may have to scale the Lanczos vectors by small quantities when this happens and the consequence after a number of steps may be serious. This is further discussed in the next subsection.

Since the subspace from which the approximations are taken is identical with that of Arnoldi's method, we have the same bounds for the distance $\|(I - \mathcal{P}_m)u_i\|_2$. However, this does not mean in any way that the approximations obtained by the two methods are likely to be of similar quality. One of the weaknesses of the method is that it relies on oblique projectors which may suffer from poor numerical properties. Moreover, the theoretical bounds shown in Chapter IV do indicate that the norm of the projector may play a significant role. The method has been used successfully by Cullum and Willoughby [24, 22] to compute eigenvalues of very large matrices. We will discuss these implementations in the next section.

4.2. Practical Implementations

There are various ways of improving the standard non-Hermitian Lanczos algorithm which we now discuss briefly. A major focus of researchers in this area is to find ways of circumventing the potential breakdowns or 'near breakdowns' in the algorithm. Other approaches do not attempt to deal with the breakdown but rather try to live with it. We will weigh the pros and cons of both approaches after we describe the various existing scenarios.

4.2.1 Look-Ahead Lanczos Algorithms

As was already mentioned, a problem with the Lanczos algorithm is the potential of breakdown in the normalization steps (6.26) and (6.27). Such a break down will occur whenever

$$(\hat{v}_{j+1}, \hat{w}_{j+1}) = 0, \tag{6.33}$$

which can arise in two different situations. Either one of the two vectors $\hat{v}_{j+1}$ or $\hat{w}_{j+1}$ vanishes or they are both nonzero but their inner product is zero. In the first case, we have again the 'lucky breakdown' scenario which we have seen in the case of Hermitian matrices. Thus, if $\hat{v}_{j+1} = 0$ then span$\{V_j\}$ is invariant and all approximate eigenvalues and associated right eigenvectors will be exact, while if $\hat{w}_{j+1} = 0$ then span$\{W_j\}$ will be invariant and the approximate eigenvalues and associated left eigenvectors will be exact. The second case, when neither of the two vectors is zero but their inner product is zero is termed *serious breakdown* by Wilkinson (see [183], p. 389). Fortunately, there are some cures, that will allow one to continue the algorithm in most cases. The corresponding modifications of the algorithm are often put under the denomination *Look-Ahead Lanczos* algorithms . There are also rare cases of 'incurable' breakdowns which will not be discussed here (see [125] and [174]). The main idea of Look-Ahead variants of the Lanczos algorithm is that even though the pair v_{j+1}, w_{j+1} cannot be defined it is often the case that the pair v_{j+2}, w_{j+2} can be defined. The algorithm can then be pursued from that iterate as before until a new breakdown is encountered. If the pair v_{j+2}, w_{j+2} cannot be defined then one can try the pair v_{j+3}, w_{j+3} and so on.

To be more precise on why this is possible, we need to go back to the connection with orthogonal polynomials mentioned earlier for the Hermitian case. We can extend the relationship to the non-Hermitian case by defining the bilinear form on the subspace $\mathbb{P}_{m-1}$

$$< p, q >= (p(A)v_1, q(A^H)w_1). \tag{6.34}$$

Unfortunately, this can constitute an 'indefinite inner product' since $< p, p >$ can now be zero or even negative. We note that there is a polynomial p_j of degree j such that $\hat{v}_{j+1} = p_j(A)v_1$

and in fact the same polynomial intervenes in the equivalent expression of w_{j+1}. More precisely, there is a scalar γ_j such that $\hat{w}_{j+1} = \gamma_j p_j(A^H)v_1$. Similarly to the Hermitian case the non-Hermitian Lanczos algorithm attempts to compute a sequence of polynomials that are orthogonal with respect to the indefinite inner product defined above. If we define the moment matrix

$$M_k = \{< x^{i-1}, x^{j-1} >\}_{i,j=1...k}$$

then this process is mathematically equivalent to finding a factorization

$$M_k = L_k U_k$$

of the moment matrix M_k, in which U_k is upper triangular and L_k is lower triangular. Note that this matrix is a Hankel matrix, i.e., a_{ij} is constant for $i + j = constant$.

Because

$$< p_j, p_j >= \bar{\gamma}_j (p_j(A)v_1, p_j(A^H)w_1)$$

we observe that there is a serious breakdown at step j if and only if the indefinite norm of the polynomial p_j at step j vanishes. The main idea of the Look-Ahead Lanczos algorithms is that if we skip this polynomial it may still be possible to compute p_{j+1} and continue to generate the sequence. To explain this simply, we consider

$$q_j(x) = xp_{j-1} \quad \text{and} \quad q_{j+1}(x) = x^2 p_{j-1}(x) \ .$$

It is easy to verify that both q_j and q_{j+1} are orthogonal to the polynomials $p_1, ..., p_{j-2}$. We can, for example, define (somewhat arbitrarily) $p_j = q_j$, and get p_{j+1} by orthogonalizing q_{j+1} against p_{j-1} and p_j. It is clear that the resulting polynomial will then be orthogonal against all polynomials of degree $\leq j$, see Exercise P-6.11. Therefore we can continue the algorithm from step $j + 1$ in the same manner. Exercise P-6.11 generalizes this to the case where we need to skip k polynomials rather than just one. This simplistic description gives the main mechanism that lies behind the different versions of Look-Ahead Lanczos algorithms proposed in the literature. In the Parlett-Taylor-Liu implementation [125], it is observed that the reason for the break down of the algorithm is that the pivots encountered during the LU factorization of the moment matrix M_k vanish. Divisions by zero are then avoided by *implicitly* performing a pivot with a 2×2 matrix rather than a using a 1×1 pivot.

The drawback of Look-Ahead implementations is the nonnegligible added complexity. In addition to the difficulty of deciding when to consider that one has a near break-down situation, one must cope with the fact that the matrix T_m is no longer tridiagonal. It is easy to see that whenever a step is skipped, we introduce a 'bump', as it it termed in [125], above the superdiagonal element. This further complicates the issue of the computation of the eigenvalues of the Ritz values.

4.2.2 The Issue of Reorthogonalization

Just as in the Hermitian case, the vectors w_j and v_i will tend to loose their bi-orthogonality. Techniques that perform some form of 'partial' or 'selective' reorthogonalization can be developed for non-Hermitian Lanczos algorithm as well. One difficulty here is that selective orthogonalization, which typically requires eigenvectors, will suffer from the fact that eigenvectors may be inaccurate. Another problem is that we now have to keep two sets of vectors, typically in secondary storage, instead of only one.

An alternative to reorthogonalization is to live with the loss of orthogonality. Although the theory is not as well understood in the non-Hermitian case as it is in the Hermitian case, it has been observed that despite the loss of orthogonality, convergence is still observed in general, at the price of a few practical difficulties. More precisely, a converged eigenvalue may appear several times, and monitoring extraneous eigenvalues becomes important. Cullum and Willoughby [25] suggest precisely such a technique based on a few heuristics. The technique is based on a comparison of the eigenvalues of the successive tridiagonal matrices T_k.

5. Block Krylov Methods

In many circumstances it is desirable to work with a block of vectors instead of a single vectors. For example, in out-of core finite-element codes it is a good strategy to exploit the presence of a block of the matrix A in fast memory, as much as possible. This can easily done with a method such as the subspace iteration for example, but not the usual Arnoldi/Lanczos algorithms. In essence, the block Arnoldi method is to the Arnoldi method what the subspace iteration is to the usual power method. Thus, the block Arnoldi can be viewed as an acceleration of the subspace iteration method. There are many possible implementations of the algorithm three of which are described next.

ALGORITHM **6.7 Block Arnoldi**

1. **Start:** *Choose a unitary matrix V_1 of dimension $n \times r$.*
2. **Iterate:** *for $j = 1, 2, \ldots, m$ compute:*

$$H_{ij} = V_i^H A V_j \quad i = 1, 2, \ldots, j, \tag{6.35}$$

$$W_j = AV_j - \sum_{i=1}^{j} V_i H_{ij} \ , \tag{6.36}$$

$$W_j = V_{j+1} H_{j+1,j} \quad \textit{Q-R decomposition of } W_j. \tag{6.37}$$

The above algorithm is a straightforward block analogue of Algorithm 6.1. By construction, the blocks constructed by the algorithm will be orthogonal blocks that are orthogonal to each other. In what follows we denote by I_k the $k \times k$ identity matrix and use the following notation

$$\begin{aligned} U_m &= [V_1, V_2, \ldots, V_m] \ , \\ H_m &= (H_{ij})_{1 \le i,j \le m}, \quad H_{ij} \equiv 0, \quad i > j+1 \ , \\ E_m &= \text{matrix of the last } r \text{ columns of } I_{nr}. \end{aligned}$$

Then, the analogue of the relation (6.8) is

$$AU_m = U_m H_m + V_{m+1} H_{m+1,m} E_m^H.$$

Thus, we obtain a relation analogous to the one we had before except that the matrix H_m is no longer Hessenberg but band-Hessenberg, in that we have $r-1$ additional diagonals below the subdiagonal.

A second version of the algorithm would consist of using a modified block Gram-Schmidt procedure instead of the simple Gram-Schmidt procedure used above. This leads to a block generalization of Algorithm 6.2, the Modified Gram-Schmidt version of Arnoldi's method.

ALGORITHM **6.8 Block Arnoldi – MGS version**

1. **Start:** *Choose a unitary matrix* V_1 *of size* $n \times r$.
2. **Iterate:** *For* $j = 1, 2, \ldots, m$ *do:*
   - *Compute* $W_j := AV_j$
   - *For* $i = 1, 2, \ldots, j$ *do:*

     $$H_{ij} := V_i^H W_j$$
     $$W_j := W_j - V_j H_{ij}.$$

   - *Compute the Q-R decomposition* $W_j = V_{j+1} H_{j+1,j}$

Again, in practice the above algorithm is more viable than its predecessor. Finally, a third version, developed by A. Ruhe, see reference [134], for the symmetric case (Block Lanczos algorithm), yields an algorithm that is quite similar to the original Arnoldi algorithm.

ALGORITHM **6.9 Block Arnoldi - Ruhe's variant**

1. **Start:** *Choose* r *initial orthonormal vectors* $\{v_i\}_{i=1,\ldots,r}$.

2. **Iterate:** *for* $j = r, r+1, , \ldots, m \times r$ *do:*

   (a) *Set* $k := j - r + 1$*;*

   (b) *Compute* $w := Av_k$*;*

   (c) *For* $i = 1, 2, \ldots, j$ *do*

      - $h_{i,k} := (w, v_i)$
      - $w := w - h_{i,k} v_i$

   (d) *Compute* $h_{j+1,k} := \|w\|_2$ *and* $v_{j+1} := w / h_{j+1,k}$.

Observe that the particular case $r = 1$ coincides with the usual Arnoldi process. That the two algorithms 6.8 and 6.9 are mathematically equivalent is straightforward to show. The advantage of the above algorithm, is its simplicity. On the other hand a slight disadvantage is that we give up some potential for parallelism. In the original version the columns of the matrix AV_j can be computed in parallel whereas in the new algorithm, we must compute them in sequence.

Generally speaking, the block methods are of great practical value in some applications but they are not as well studied from the theoretical point of view. One of the reasons is possibly the lack of any convincing analogue of the relationship with orthogonal polynomials established in Subsection 3.2 for the single vector Lanczos algorithm. We have not covered the block versions of the two Lanczos algorithms (Hermitian and non-Hermitian) but these generalizations are straightforward.

6. Convergence of the Lanczos Process

In this section we examine the convergence properties of the Hermitian Lanczos algorithm, from a theoretical point of view. Well-known results from approximation theory will be used to derive a convergence analysis of the method. In particular Chebyshev polynomials play an important role and we refer the readers to the end of Chapter IV for some background on these polynomials.

6.1. Distance between $\mathcal{K}_m$ and an Eigenvector

In the following we will assume that the eigenvalues of the Hermitian matrix A are labeled in decreasing order, i.e.,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n ,$$

and that the approximate eigenvalues are labeled similarly. We will now state the main result of this section, starting with the following lemma.

Lemma 6.1 *Let P_i be the spectral projector associated with the eigenvalue λ_i. Then, if $P_i v_1 \neq 0$, we have*

$$\tan\theta(u_i, \mathcal{K}_m) = \min_{p \in \mathbb{P}_{m-1},\ p(\lambda_i)=1} \|p(A)y_i\|_2 \tan\theta(u_i, v_1) \tag{6.38}$$

in which

$$y_i = \begin{cases} \frac{(I-P_i)v_1}{\|(I-P_i)v_1\|_2} & \text{if} \quad (I-P_i)v_1 \neq 0 , \\ 0 & \text{otherwise.} \end{cases}$$

Proof. The subspace $\mathcal{K}_m$ consists of all vectors of the form $x = q(A)v_1$ where q is any polynomial of degree $\leq m-1$. We have the orthogonal decomposition

$$x = q(A)v_1 = q(A)P_i v_1 + q(A)(I - P_i)v_1$$

and the angle between x and u_i is defined by

$$\begin{aligned} \tan\theta(x, u_i) &= \frac{\|q(A)(I-P_i)v_1\|_2}{\|q(A)P_i v_1\|_2} \\ &= \frac{\|q(A)y_i\|_2}{|q(\lambda_i)|} \frac{\|(I-P_i)v_1\|_2}{\|P_i v_1\|_2} . \end{aligned}$$

If we let $p(\lambda) \equiv q(\lambda)/q(\lambda_i)$ we get

$$\tan\theta(x, u_i) = \|p(A)y_i\|_2 \tan\theta(v_1, u_i)$$

which shows the result by taking the minimum over all x's in $\mathcal{K}_m$.
∎