AIMS Lecture Notes 2006

Peter J. Olver

8. Numerical Computation of Eigenvalues

In this part, we discuss some practical methods for computing eigenvalues and eigenvectors of matrices. Needless to say, we completely avoid trying to solve (or even write down) the characteristic polynomial equation. The very basic power method and its variants, which is based on linear iteration, is used to effectively approximate selected eigenvalues. To determine the complete system of eigenvalues and eigenvectors, the remarkable $Q\,R$ algorithm, which relies on the Gram–Schmidt orthogonalization procedure, is the method of choice, and we shall close with a new proof of its convergence.

8.1. The Power Method.

We have already noted the role played by the eigenvalues and eigenvectors in the solution to linear iterative systems. Now we are going to turn the tables, and use the iterative system as a mechanism for approximating the eigenvalues, or, more correctly, selected eigenvalues of the coefficient matrix. The simplest of the resulting computational procedures is known as the *power method*.

We assume, for simplicity, that A is a complete[†] $n \times n$ matrix. Let $\mathbf{v}_1, \dots, \mathbf{v}_n$ denote its eigenvector basis, and $\lambda_1, \dots, \lambda_n$ the corresponding eigenvalues. As we have learned, the solution to the linear iterative system

$$\mathbf{v}^{(k+1)} = A\,\mathbf{v}^{(k)}, \qquad \mathbf{v}^{(0)} = \mathbf{v}, \tag{8.1}$$

is obtained by multiplying the initial vector $\mathbf{v}$ by the successive powers of the coefficient matrix: $\mathbf{v}^{(k)} = A^k\,\mathbf{v}$. If we write the initial vector in terms of the eigenvector basis

$$\mathbf{v} = c_1\,\mathbf{v}_1 + \,\cdots\, + c_n\,\mathbf{v}_n, \tag{8.2}$$

then the solution takes the explicit form given in Theorem 7.2, namely

$$\mathbf{v}^{(k)} = A^k\,\mathbf{v} = c_1\,\lambda_1^k\,\mathbf{v}_1 + \,\cdots\, + c_n\,\lambda_n^k\,\mathbf{v}_n. \tag{8.3}$$

[†] This is not a very severe restriction. Most matrices are complete. Moreover, perturbations caused by round off and/or numerical inaccuracies will almost inevitably make an incomplete matrix complete.

Suppose further that A has a single dominant *real* eigenvalue, λ_1, that is larger than all others in magnitude, so

$$|\lambda_1| > |\lambda_j| \qquad \text{for all} \qquad j > 1. \tag{8.4}$$

As its name implies, this eigenvalue will eventually dominate the iteration (8.3). Indeed, since

$$|\lambda_1|^k \gg |\lambda_j|^k \qquad \text{for all} \quad j > 1 \quad \text{and all} \quad k \gg 0,$$

the first term in the iterative formula (8.3) will eventually be much larger than the rest, and so, provided $c_1 \neq 0$,

$$\mathbf{v}^{(k)} \;\approx\; c_1 \lambda_1^k \mathbf{v}_1 \qquad \text{for} \qquad k \gg 0.$$

Therefore, the solution to the iterative system (8.1) will, almost always, end up being a multiple of the dominant eigenvector of the coefficient matrix.

To compute the corresponding eigenvalue, we note that the i^{th} entry of the iterate $\mathbf{v}^{(k)}$ is approximated by $v_i^{(k)} \approx c_1 \lambda_1^k v_{1,i}$, where $v_{1,i}$ is the i^{th} entry of the eigenvector $\mathbf{v}_1$. Thus, as long as $v_{1,i} \neq 0$, we can recover the dominant eigenvalue by taking a ratio between selected components of successive iterates:

$$\lambda_1 \;\approx\; \frac{v_i^{(k)}}{v_i^{(k-1)}}, \qquad \text{provided that} \qquad v_i^{(k-1)} \neq 0. \tag{8.5}$$

Example 8.1. Consider the matrix $A = \begin{pmatrix} -1 & 2 & 2 \\ -1 & -4 & -2 \\ -3 & 9 & 7 \end{pmatrix}$. As you can check, its eigenvalues and eigenvectors are

$$\lambda_1 = 3, \qquad \mathbf{v}_1 = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}, \qquad \lambda_2 = -2, \qquad \mathbf{v}_2 = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}, \qquad \lambda_3 = 1, \qquad \mathbf{v}_3 = \begin{pmatrix} -1 \\ 1 \\ -2 \end{pmatrix}.$$

Repeatedly multiplying an initial vector $\mathbf{v} = (\,1, 0, 0\,)^T$, say, by A results in the iterates $\mathbf{v}^{(k)} = A^k \mathbf{v}$ listed in the accompanying table. The last column indicates the ratio $\lambda^{(k)} = v_1^{(k)} / v_1^{(k-1)}$ between the first components of successive iterates. (One could equally well use the second or third components.) The ratios are converging to the dominant eigenvalue $\lambda_1 = 3$, while the vectors $\mathbf{v}^{(k)}$ are converging to a very large multiple of the corresponding eigenvector $\mathbf{v}_1 = (\,1, -1, 3\,)^T$.

The success of the power method lies in the assumption that A has a unique dominant eigenvalue of maximal modulus, which, by definition, equals its spectral radius: $|\lambda_1| = \rho(A)$. The rate of convergence of the method is governed by the ratio $|\lambda_2/\lambda_1|$ between the subdominant and dominant eigenvalues. Thus, the farther the dominant eigenvalue lies away from the rest, the faster the power method converges. We also assumed that the initial vector $\mathbf{v}^{(0)}$ includes a nonzero multiple of the dominant eigenvector, i.e., $c_1 \neq 0$. As we do not know the eigenvectors, it is not so easy to guarantee this in advance, although one must be quite unlucky to make such a poor choice of initial vector. (Of course, the stupid choice $\mathbf{v}^{(0)} = \mathbf{0}$ is not counted.) Moreover, even if c_1 happens to be 0 initially,

k	$\mathbf{v}^{(k)}$			$\lambda^{(k)}$
0	1	0	0	
1	−1	−1	−3	−1.
2	−7	11	−27	7.
3	−25	17	−69	3.5714
4	−79	95	−255	3.1600
5	−241	209	−693	3.0506
6	−727	791	−2247	3.0166
7	−2185	2057	−6429	3.0055
8	−6559	6815	−19935	3.0018
9	−19681	19169	−58533	3.0006
10	−59047	60071	−178167	3.0002
11	−177145	175097	−529389	3.0001
12	−531439	535535	−1598415	3.0000

numerical round-off error will typically come to one's rescue, since it will almost inevitably introduce a tiny component of the eigenvector $\mathbf{v}_1$ into some iterate, and this component will eventually dominate the computation. The trick is to wait long enough for it to show up!

Since the iterates of A are, typically, getting either very large — when $\rho(A) > 1$ — or very small — when $\rho(A) < 1$ — the iterated vectors will be increasingly subject to numerical over- or under-flow, and the method may break down before a reasonable approximation is achieved. One way to avoid this outcome is to restrict our attention to unit vectors relative to a given norm, e.g., the Euclidean norm or the ∞ norm, since their entries cannot be too large, and so are less likely to cause numerical errors in the computations. As usual, the unit vector $\mathbf{u}^{(k)} = \| \mathbf{v}^{(k)} \|^{-1} \mathbf{v}^{(k)}$ is obtained by dividing the iterate by its norm; it can be computed directly by the modified iterative scheme

$$\mathbf{u}^{(0)} = \frac{\mathbf{v}^{(0)}}{\| \mathbf{v}^{(0)} \|}, \qquad \text{and} \qquad \mathbf{u}^{(k+1)} = \frac{A\,\mathbf{u}^{(k)}}{\| A\,\mathbf{u}^{(k)} \|}. \tag{8.6}$$

If the dominant eigenvalue $\lambda_1 > 0$ is positive, then $\mathbf{u}^{(k)} \to \mathbf{u}_1$ will converge to one of the two dominant unit eigenvectors (the other is $-\mathbf{u}_1$). If $\lambda_1 < 0$, then the iterates will switch back and forth between the two eigenvectors, so $\mathbf{u}^{(k)} \approx \pm \mathbf{u}_1$. In either case, the dominant eigenvalue λ_1 is obtained as a limiting ratio between nonzero entries of $A\,\mathbf{u}^{(k)}$ and $\mathbf{u}^{(k)}$. If some other sort of behavior is observed, it means that one of our assumptions is not valid; either A has more than one dominant eigenvalue of maximum modulus, e.g., it has a complex conjugate pair of eigenvalues of largest modulus, or it is not complete.

Example 8.2. For the matrix considered in Example 8.1, starting the iterative scheme (8.6) with $\mathbf{u}^{(k)} = (\,1, 0, 0\,)^T$ by A, the resulting unit vectors are tabulated below.

k	$\mathbf{u}^{(k)}$			λ
0	1	0	0	
1	−.3015	−.3015	−.9045	−1.0000
2	−.2335	.3669	−.9005	7.0000
3	−.3319	.2257	−.9159	3.5714
4	−.2788	.3353	−.8999	3.1600
5	−.3159	.2740	−.9084	3.0506
6	−.2919	.3176	−.9022	3.0166
7	−.3080	.2899	−.9061	3.0055
8	−.2973	.3089	−.9035	3.0018
9	−.3044	.2965	−.9052	3.0006
10	−.2996	.3048	−.9041	3.0002
11	−.3028	.2993	−.9048	3.0001
12	−.3007	.3030	−.9043	3.0000

The last column, being the ratio between the first components of $A\,\mathbf{u}^{(k-1)}$ and $\mathbf{u}^{(k-1)}$, again converges to the dominant eigenvalue $\lambda_1 = 3$.

Variants of the power method for computing the other eigenvalues of the matrix are explored in the exercises.

8.2. The QR Algorithm.

The most popular scheme for simultaneously approximating all the eigenvalues of a matrix A is the remarkable QR algorithm, first proposed in 1961 by Francis, [**18**], and Kublanovskaya, [**31**]. The underlying idea is simple, but surprising. The first step is to factor the matrix

$$A = A_0 = Q_0\, R_0$$

into a product of an orthogonal matrix Q_0 and a positive (i.e., with all positive entries along the diagonal) upper triangular matrix R_0 by using the Gram–Schmidt orthogonalization procedure. Next, multiply the two factors together *in the wrong order*! The result is the new matrix

$$A_1 = R_0\, Q_0.$$

We then repeat these two steps. Thus, we next factor

$$A_1 = Q_1\, R_1$$

using the Gram–Schmidt process, and then multiply the factors in the reverse order to produce

$$A_2 = R_1\, Q_1.$$

The complete algorithm can be written as

$$A = Q_0 R_0, \qquad A_{k+1} = R_k Q_k = Q_{k+1} R_{k+1}, \qquad k = 0, 1, 2, \dots, \tag{8.7}$$

where Q_k, R_k come from the previous step, and the subsequent orthogonal matrix Q_{k+1} and positive upper triangular matrix R_{k+1} are computed by using the numerically stable form of the Gram–Schmidt algorithm.

The astonishing fact is that, for many matrices A, the iterates $A_k \longrightarrow V$ converge to an upper triangular matrix V whose diagonal entries are the eigenvalues of A. Thus, after a sufficient number of iterations, say $k^\star$, the matrix $A_{k^\star}$ will have very small entries below the diagonal, and one can read off a complete system of (approximate) eigenvalues along its diagonal. For each eigenvalue, the computation of the corresponding eigenvector can be done by solving the appropriate homogeneous linear system, or by applying the shifted inverse power method.

Example 8.3. Consider the matrix $A = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$. The initial Gram–Schmidt factorization $A = Q_0 R_0$ yields

$$Q_0 = \begin{pmatrix} .7071 & -.7071 \\ .7071 & .7071 \end{pmatrix}, \qquad R_0 = \begin{pmatrix} 2.8284 & 2.8284 \\ 0 & 1.4142 \end{pmatrix}.$$

These are multiplied in the reverse order to give

$$A_1 = R_0 Q_0 = \begin{pmatrix} 4 & 0 \\ 1 & 1 \end{pmatrix}.$$

We refactor $A_1 = Q_1 R_1$ via Gram–Schmidt, and then reverse multiply to produce

$$Q_1 = \begin{pmatrix} .9701 & -.2425 \\ .2425 & .9701 \end{pmatrix}, \qquad R_1 = \begin{pmatrix} 4.1231 & .2425 \\ 0 & .9701 \end{pmatrix},$$

$$A_2 = R_1 Q_1 = \begin{pmatrix} 4.0588 & -.7647 \\ .2353 & .9412 \end{pmatrix}.$$

The next iteration yields

$$Q_2 = \begin{pmatrix} .9983 & -.0579 \\ .0579 & .9983 \end{pmatrix}, \qquad R_2 = \begin{pmatrix} 4.0656 & -.7090 \\ 0 & .9839 \end{pmatrix},$$

$$A_3 = R_2 Q_2 = \begin{pmatrix} 4.0178 & -.9431 \\ .0569 & .9822 \end{pmatrix}.$$

Continuing in this manner, after 9 iterations we find, to four decimal places,

$$Q_9 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad R_9 = \begin{pmatrix} 4 & -1 \\ 0 & 1 \end{pmatrix}, \qquad A_{10} = R_9 Q_9 = \begin{pmatrix} 4 & -1 \\ 0 & 1 \end{pmatrix}.$$

The eigenvalues of A, namely 4 and 1, appear along the diagonal of A_{10}. Additional iterations produce very little further change, although they can be used for increasing the accuracy of the computed eigenvalues.

If the original matrix A happens to be symmetric and positive definite, then the limiting matrix $A_k \longrightarrow V = \Lambda$ is, in fact, the diagonal matrix containing the eigenvalues of A. Moreover, if, in this case, we recursively define

$$S_k = S_{k-1} Q_k = Q_0 Q_1 \cdots Q_{k-1} Q_k, \tag{8.8}$$

then $S_k \longrightarrow S$ have, as their limit, an orthogonal matrix whose columns are the orthonormal eigenvector basis of A.

Example 8.4. Consider the symmetric matrix $A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & -1 \\ 0 & -1 & 6 \end{pmatrix}$. The initial $A = Q_0 R_0$ factorization produces

$$S_0 = Q_0 = \begin{pmatrix} .8944 & -.4082 & -.1826 \\ .4472 & .8165 & .3651 \\ 0 & -.4082 & .9129 \end{pmatrix}, \qquad R_0 = \begin{pmatrix} 2.2361 & 2.2361 & -.4472 \\ 0 & 2.4495 & -3.2660 \\ 0 & 0 & 5.1121 \end{pmatrix},$$

and so

$$A_1 = R_0 Q_0 = \begin{pmatrix} 3.0000 & 1.0954 & 0 \\ 1.0954 & 3.3333 & -2.0870 \\ 0 & -2.0870 & 4.6667 \end{pmatrix}.$$

We refactor $A_1 = Q_1 R_1$ and reverse multiply to produce

$$Q_1 = \begin{pmatrix} .9393 & -.2734 & -.2071 \\ .3430 & .7488 & .5672 \\ 0 & -.6038 & .7972 \end{pmatrix}, \qquad S_1 = S_0 Q_1 = \begin{pmatrix} .7001 & -.4400 & -.5623 \\ .7001 & .2686 & .6615 \\ -.1400 & -.8569 & .4962 \end{pmatrix},$$

$$R_1 = \begin{pmatrix} 3.1937 & 2.1723 & -.7158 \\ 0 & 3.4565 & -4.3804 \\ 0 & 0 & 2.5364 \end{pmatrix}, \qquad A_2 = R_1 Q_1 = \begin{pmatrix} 3.7451 & 1.1856 & 0 \\ 1.1856 & 5.2330 & -1.5314 \\ 0 & -1.5314 & 2.0219 \end{pmatrix}.$$

Continuing in this manner, after 10 iterations we find

$$Q_{10} = \begin{pmatrix} 1.0000 & -.0067 & 0 \\ .0067 & 1.0000 & .0001 \\ 0 & -.0001 & 1.0000 \end{pmatrix}, \qquad S_{10} = \begin{pmatrix} .0753 & -.5667 & -.8205 \\ .3128 & -.7679 & .5591 \\ -.9468 & -.2987 & .1194 \end{pmatrix},$$

$$R_{10} = \begin{pmatrix} 6.3229 & .0647 & 0 \\ 0 & 3.3582 & -.0006 \\ 0 & 0 & 1.3187 \end{pmatrix}, \qquad A_{11} = \begin{pmatrix} 6.3232 & .0224 & 0 \\ .0224 & 3.3581 & -.0002 \\ 0 & -.0002 & 1.3187 \end{pmatrix}.$$

After 20 iterations, the process has completely settled down, and

$$Q_{20} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad S_{20} = \begin{pmatrix} .0710 & -.5672 & -.8205 \\ .3069 & -.7702 & .5590 \\ -.9491 & -.2915 & .1194 \end{pmatrix},$$

$$R_{20} = \begin{pmatrix} 6.3234 & .0001 & 0 \\ 0 & 3.3579 & 0 \\ 0 & 0 & 1.3187 \end{pmatrix}, \qquad A_{21} = \begin{pmatrix} 6.3234 & 0 & 0 \\ 0 & 3.3579 & 0 \\ 0 & 0 & 1.3187 \end{pmatrix}.$$

The eigenvalues of A appear along the diagonal of A_{21}, while the columns of S_{20} are the corresponding orthonormal eigenvector basis, listed in the same order as the eigenvalues, both correct to 4 decimal places.

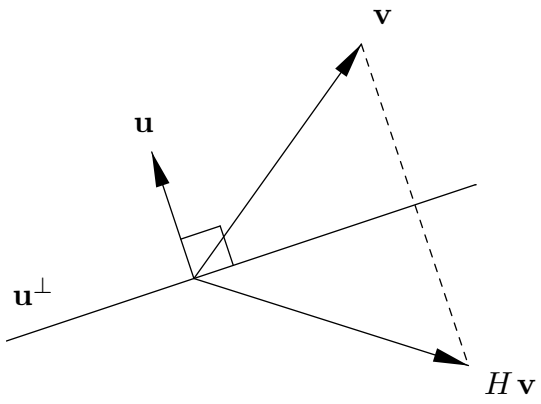

Figure 8.1. Elementary Reflection Matrix.

Tridiagonalization

In practical implementations, the direct $Q\,R$ algorithm often takes too long to provide reasonable approximations to the eigenvalues of large matrices. Fortunately, the algorithm can be made much more efficient by a simple preprocessing step. The key observation is that the $Q\,R$ algorithm preserves the class of symmetric tridiagonal matrices, and, moreover, like Gaussian Elimination, is much faster when applied to this class of matrices.

Consider the *Householder* or *elementary reflection matrix*

$$H = \mathrm{I} - 2\,\mathbf{u}\,\mathbf{u}^T \tag{8.9}$$

in which $\mathbf{u}$ is a unit vector (in the Euclidean norm). The matrix H represents a reflection of vectors through the orthogonal complement to $\mathbf{u}$, as illustrated in Figure 8.1. It is easy to shoe that H is a symmetric orthogonal matrix, and so

$$H^T = H, \qquad H^2 = \mathrm{I}\,, \qquad H^{-1} = H. \tag{8.10}$$

The proof is straightforward: symmetry is immediate, while

$$H\,H^T = H^2 = (\,\mathrm{I} - 2\,\mathbf{u}\,\mathbf{u}^T)\,(\,\mathrm{I} - 2\,\mathbf{u}\,\mathbf{u}^T) = \mathrm{I} - 4\,\mathbf{u}\,\mathbf{u}^T + 4\,\mathbf{u}\,(\mathbf{u}^T\mathbf{u})\,\mathbf{u}^T = \mathrm{I}$$

since, by assumption, $\mathbf{u}^T\mathbf{u} = \|\,\mathbf{u}\,\|^2 = 1$.

In Householder's approach to the $Q\,R$ factorization, we were able to convert the matrix A to upper triangular form R by a sequence of elementary reflection matrices. Unfortunately, this procedure does not preserve the eigenvalues of the matrix — the diagonal entries of R are *not* the eigenvalues — and so we need to be a bit more clever here.

Lemma 8.5. *If $H = \mathrm{I} - 2\,\mathbf{u}\,\mathbf{u}^T$ is an elementary reflection matrix, with $\mathbf{u}$ a unit vector, then A and $B = HAH$ are similar matrices and hence have the same eigenvalues.*

Proof: According to (8.10), $H^{-1} = H$, and hence $B = H^{-1}AH$ is similar to A. *Q.E.D.*

Given a symmetric $n \times n$ matrix A, our goal is to devise a similar tridiagonal matrix by applying a sequence of Householder reflections. We begin by setting

$$\mathbf{x}_1 = \begin{pmatrix} 0 \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{pmatrix}, \qquad \mathbf{y}_1 = \begin{pmatrix} 0 \\ \pm r_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \text{where} \qquad r_1 = \| \mathbf{x}_1 \| = \| \mathbf{y}_1 \|,$$

so that $\mathbf{x}_1$ contains all the off-diagonal entries of the first column of A. Let

$$H_1 = \mathrm{I} - 2\,\mathbf{u}_1\,\mathbf{u}_1^T, \qquad \text{where} \qquad \mathbf{u}_1 = \frac{\mathbf{x}_1 - \mathbf{y}_1}{\| \mathbf{x}_1 - \mathbf{y}_1 \|}$$

be the corresponding elementary reflection matrix that maps $\mathbf{x}_1$ to $\mathbf{y}_1$. Either $\pm$ sign in the formula for $\mathbf{y}_1$ works in the algorithm; a good choice is to set it to be the opposite of the sign of the entry a_{21}, which helps minimize the possible effects of round-off error when computing the unit vector $\mathbf{u}_1$. By direct computation,

$$A_2 = H_1\,A\,H_1 = \begin{pmatrix} a_{11} & r_1 & 0 & \dots & 0 \\ r_1 & \widetilde{a}_{22} & \widetilde{a}_{23} & \dots & \widetilde{a}_{2n} \\ 0 & \widetilde{a}_{32} & \widetilde{a}_{33} & \dots & \widetilde{a}_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \widetilde{a}_{n2} & \widetilde{a}_{n3} & \dots & \widetilde{a}_{nn} \end{pmatrix} \tag{8.11}$$

for certain $\widetilde{a}_{ij}$; the explicit formulae are not needed. Thus, by a single Householder transformation, we convert A into a similar matrix A_2 whose first row and column are in tridiagonal form. We repeat the process on the lower right $(n-1) \times (n-1)$ submatrix of A_2. We set

$$\mathbf{x}_2 = \begin{pmatrix} 0 \\ 0 \\ \widetilde{a}_{32} \\ \widetilde{a}_{42} \\ \vdots \\ \widetilde{a}_{n2} \end{pmatrix}, \qquad \mathbf{y}_1 = \begin{pmatrix} 0 \\ 0 \\ \pm r_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \text{where} \qquad r_2 = \| \mathbf{x}_2 \| = \| \mathbf{y}_2 \|,$$

and the $\pm$ sign is chosen to be the opposite of that of $\widetilde{a}_{32}$. Setting

$$H_2 = \mathrm{I} - 2\,\mathbf{u}_2\,\mathbf{u}_2^T, \qquad \text{where} \qquad \mathbf{u}_2 = \frac{\mathbf{x}_2 - \mathbf{y}_2}{\| \mathbf{x}_2 - \mathbf{y}_2 \|},$$

we construct the similar matrix

$$A_3 = H_2\,A_2\,H_2 = \begin{pmatrix} a_{11} & r_1 & 0 & 0 & \dots & 0 \\ r_1 & \widetilde{a}_{22} & r_2 & 0 & \dots & 0 \\ 0 & r_2 & \widehat{a}_{33} & \widehat{a}_{34} & \dots & \widehat{a}_{3n} \\ 0 & 0 & \widehat{a}_{43} & \widehat{a}_{44} & \dots & \widehat{a}_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \widehat{a}_{n3} & \widehat{a}_{n4} & \dots & \widehat{a}_{nn} \end{pmatrix}.$$

whose first two rows and columns are now in tridiagonal form. The remaining steps in the algorithm should now be clear. Thus, the final result is a tridiagonal matrix $T = A_n$ that has the *same eigenvalues* as the original symmetric matrix A. Let us illustrate the method by an example.

Example 8.6. To tridiagonalize $A = \begin{pmatrix} 4 & 1 & -1 & 2 \\ 1 & 4 & 1 & -1 \\ -1 & 1 & 4 & 1 \\ 2 & -1 & 1 & 4 \end{pmatrix}$, we begin with its first column. We set $\mathbf{x}_1 = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 2 \end{pmatrix}$, so that $\mathbf{y}_1 = \begin{pmatrix} 0 \\ \sqrt{6} \\ 0 \\ 0 \end{pmatrix} \approx \begin{pmatrix} 0 \\ 2.4495 \\ 0 \\ 0 \end{pmatrix}$. Therefore, the unit vector is $\mathbf{u}_1 = \dfrac{\mathbf{x}_1 - \mathbf{y}_1}{\| \mathbf{x}_1 - \mathbf{y}_1 \|} = \begin{pmatrix} 0 \\ .8391 \\ -.2433 \\ .4865 \end{pmatrix}$, with corresponding Householder matrix

$$H_1 = \mathrm{I} - 2\,\mathbf{u}_1\,\mathbf{u}_1^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -.4082 & .4082 & -.8165 \\ 0 & .4082 & .8816 & .2367 \\ 0 & -.8165 & .2367 & .5266 \end{pmatrix}.$$

Thus,

$$A_2 = H_1\,A\,H_1 = \begin{pmatrix} 4.0000 & -2.4495 & 0 & 0 \\ -2.4495 & 2.3333 & -.3865 & -.8599 \\ 0 & -.3865 & 4.9440 & -.1246 \\ 0 & -.8599 & -.1246 & 4.7227 \end{pmatrix}.$$

In the next phase, $\mathbf{x}_2 = \begin{pmatrix} 0 \\ 0 \\ -.3865 \\ -.8599 \end{pmatrix}$, $\mathbf{y}_2 = \begin{pmatrix} 0 \\ 0 \\ -.9428 \\ 0 \end{pmatrix}$, so $\mathbf{u}_2 = \begin{pmatrix} 0 \\ 0 \\ -.8396 \\ -.5431 \end{pmatrix}$, and

$$H_2 = \mathrm{I} - 2\,\mathbf{u}_2\,\mathbf{u}_2^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -.4100 & -.9121 \\ 0 & 0 & -.9121 & .4100 \end{pmatrix}.$$

The resulting matrix

$$T = A_3 = H_2\,A_2\,H_2 = \begin{pmatrix} 4.0000 & -2.4495 & 0 & 0 \\ -2.4495 & 2.3333 & .9428 & 0 \\ 0 & .9428 & 4.6667 & 0 \\ 0 & 0 & 0 & 5 \end{pmatrix}$$

is now in tridiagonal form.

Since the final tridiagonal matrix T has the same eigenvalues as A, we can apply the $Q\,R$ algorithm to T to approximate the common eigenvalues. (The eigenvectors must then be computed separately, e.g., by the shifted inverse power method.) If $A = A_1$ is

tridiagonal, so are all the iterates $A_2, A_3, \dots$. Moreover, far fewer arithmetic operations are required. For instance, in the preceding example, after we apply 20 iterations of the QR algorithm directly to T, the upper triangular factor has become

$$R_{20} = \begin{pmatrix} 6.0000 & -.0065 & 0 & 0 \\ 0 & 4.5616 & 0 & 0 \\ 0 & 0 & 5.0000 & 0 \\ 0 & 0 & 0 & .4384 \end{pmatrix}.$$

The eigenvalues of T, and hence also of A, appear along the diagonal, and are correct to 4 decimal places.

Finally, even if A is not symmetric, one can still apply the same sequence of Householder transformations to simplify it. The final result is no longer tridiagonal, but rather a similar *upper Hessenberg matrix*, which means that all entries below the subdiagonal are zero, but those above the superdiagonal are not necessarily zero. For instance, a 5×5 upper Hessenberg matrix looks like

$$\begin{pmatrix} * & * & * & * & * \\ * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{pmatrix},$$

where the starred entries can be anything. It can be proved that the QR algorithm maintains the upper Hessenberg form, and, while not as efficient as in the tridiagonal case, still yields a significant savings in computational effort required to find the common eigenvalues. Further details and analysis can be found in [**13**, **43**, **48**].

