```
            for (i=m-1;i>=l;i--) {              A plane rotation as in the origi-
                f=s*e[i];                           nal QL, followed by Givens
                b=c*e[i];                           rotations to restore tridiag-
                e[i+1]=(r=pythag(f,g));             onal form.
                if (r == 0.0) {                 Recover from underflow.
                    d[i+1] -= p;
                    e[m]=0.0;
                    break;
                }
                s=f/r;
                c=g/r;
                g=d[i+1]-p;
                r=(d[i]-g)*s+2.0*c*b;
                d[i+1]=g+(p=s*r);
                g=c*r-b;
                /* Next loop can be omitted if eigenvectors not wanted*/
                for (k=1;k<=n;k++) {            Form eigenvectors.
                    f=z[k][i+1];
                    z[k][i+1]=s*z[k][i]+c*f;
                    z[k][i]=c*z[k][i]-s*f;
                }
            }
            if (r == 0.0 && i >= l) continue;
            d[l] -= p;
            e[l]=g;
            e[m]=0.0;
        }
    } while (m != l);
  }
}
```


CITED REFERENCES AND FURTHER READING:

Acton, F.S. 1970, *Numerical Methods That Work*; 1990, corrected edition (Washington: Mathematical Association of America), pp. 331–335. [1]

Wilkinson, J.H., and Reinsch, C. 1971, *Linear Algebra*, vol. II of *Handbook for Automatic Computation* (New York: Springer-Verlag). [2]

Smith, B.T., et al. 1976, *Matrix Eigensystem Routines — EISPACK Guide*, 2nd ed., vol. 6 of Lecture Notes in Computer Science (New York: Springer-Verlag). [3]

Stoer, J., and Bulirsch, R. 1980, *Introduction to Numerical Analysis* (New York: Springer-Verlag), §6.6.6. [4]

11.4 Hermitian Matrices

The complex analog of a real, symmetric matrix is a Hermitian matrix, satisfying equation (11.0.4). Jacobi transformations can be used to find eigenvalues and eigenvectors, as also can Householder reduction to tridiagonal form followed by QL iteration. Complex versions of the previous routines `jacobi`, `tred2`, and `tqli` are quite analogous to their real counterparts. For working routines, consult [1,2].

An alternative, using the routines in this book, is to convert the Hermitian problem to a real, symmetric one: If $\mathbf{C} = \mathbf{A} + i\mathbf{B}$ is a Hermitian matrix, then the $n \times n$ complex eigenvalue problem

$$(\mathbf{A} + i\mathbf{B}) \cdot (\mathbf{u} + i\mathbf{v}) = \lambda(\mathbf{u} + i\mathbf{v}) \tag{11.4.1}$$

is equivalent to the $2n \times 2n$ real problem

$$\begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \tag{11.4.2}$$

Note that the $2n \times 2n$ matrix in (11.4.2) is symmetric: $\mathbf{A}^T = \mathbf{A}$ and $\mathbf{B}^T = -\mathbf{B}$ if $\mathbf{C}$ is Hermitian.

Corresponding to a given eigenvalue λ, the vector

$$\begin{bmatrix} -\mathbf{v} \\ \mathbf{u} \end{bmatrix} \tag{11.4.3}$$

is also an eigenvector, as you can verify by writing out the two matrix equations implied by (11.4.2). Thus if $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of $\mathbf{C}$, then the $2n$ eigenvalues of the augmented problem (11.4.2) are $\lambda_1, \lambda_1, \lambda_2, \lambda_2, \ldots, \lambda_n, \lambda_n$; each, in other words, is repeated twice. The eigenvectors are pairs of the form $\mathbf{u} + i\mathbf{v}$ and $i(\mathbf{u} + i\mathbf{v})$; that is, they are the same up to an inessential phase. Thus we solve the augmented problem (11.4.2), and choose one eigenvalue and eigenvector from each pair. These give the eigenvalues and eigenvectors of the original matrix $\mathbf{C}$.

Working with the augmented matrix requires a factor of 2 more storage than the original complex matrix. In principle, a complex algorithm is also a factor of 2 more efficient in computer time than is the solution of the augmented problem.

CITED REFERENCES AND FURTHER READING:

Wilkinson, J.H., and Reinsch, C. 1971, *Linear Algebra*, vol. II of *Handbook for Automatic Computation* (New York: Springer-Verlag). [1]

Smith, B.T., et al. 1976, *Matrix Eigensystem Routines — EISPACK Guide*, 2nd ed., vol. 6 of Lecture Notes in Computer Science (New York: Springer-Verlag). [2]

11.5 Reduction of a General Matrix to Hessenberg Form

The algorithms for symmetric matrices, given in the preceding sections, are highly satisfactory in practice. By contrast, it is impossible to design equally satisfactory algorithms for the nonsymmetric case. There are two reasons for this. First, the eigenvalues of a nonsymmetric matrix can be very sensitive to small changes in the matrix elements. Second, the matrix itself can be defective, so that there is no complete set of eigenvectors. We emphasize that these difficulties are intrinsic properties of certain nonsymmetric matrices, and no numerical procedure can "cure" them. The best we can hope for are procedures that don't exacerbate such problems.

The presence of rounding error can only make the situation worse. With finite-precision arithmetic, one cannot even design a foolproof algorithm to determine whether a given matrix is defective or not. Thus current algorithms generally *try* to find a *complete* set of eigenvectors, and rely on the user to inspect the results. If any eigenvectors are almost parallel, the matrix is probably defective.