nag®

CSC / NAG Autumn School on

Core Algorithms in High-Performance Scientific Computing

Maths III

Dwight Barkley

Eigenvalue Problems I

Maths III: Eigenvalue Problems

A key problem in numerical linear algebra is the eigenvalue problem: Given an $n \times n$ matrix A, find v and λ such that

$$Av = \lambda v \tag{1.1}$$

The scalar λ is an eigenvalue and the vector v is an eigenvector.

In this lecture we address direct methods for solving the eigenvalue problem problem and in Maths lecture V we consider iterative methods. The methods considered here are most appropriate for dense matrices and where all eigenvalues are desired. Iterative methods are better suited for sparse problems and possibly cases where only a few of many eigenvalues are required.

The linear systems of equations we considered previously occur very frequently as a sub-problem of a larger calculation. While the same may be true for the eigenvalue problem, quite often the solution to the eigenvalue problem is the answer sought. In fact, often it is just the eigenvalue themselves that are desired.

1.1 Two Examples

Chemical Reaction

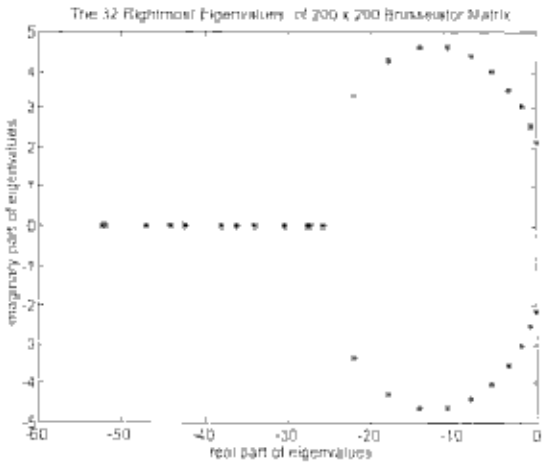

Figure 1.1: Eigenvalue spectrum for the Brusselator wave model in a chemical reaction. From Matrix Market http://math.nist.gov/MatrixMarket.

The stability of chemical reactions illustrates a typical situation in which one needs to know the eigenvalues

of a system. The problem considered here is a model for the concentration waves arising from reaction and transport of chemical solutions in a tubular reactor. The details are not important. The issue is that one needs to know the set of all eigenvalues of the system: the **eigenvalue spectrum**. If all eigenvalues lie in the left half of the complex plane, then the reaction is stable, otherwise it is unstable. Figure 1.1 shows the 32 right-most eigenvalues (those of most interest) for a 200×200 matrix.

Eigenfaces

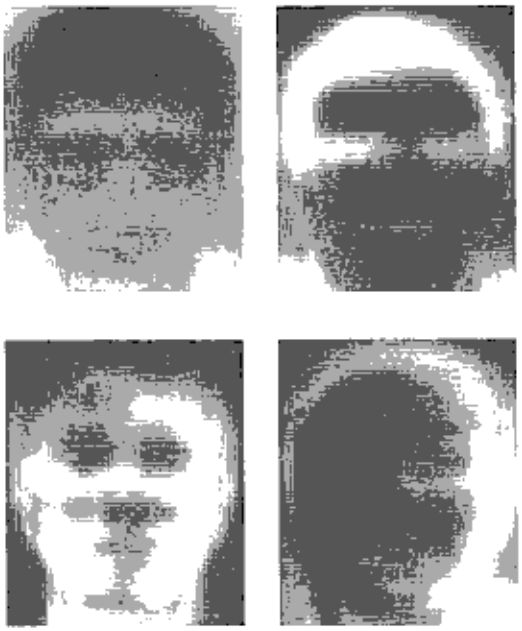

Figure 1.2: Eigenfaces.

From Wikipedia:

> *Eigenfaces are a set of eigenvectors used in the computer vision problem of human face recognition. The approach of using eigenfaces for recognition was developed by Sirovich and Kirby (1987) and used by Matthew Turk and Alex Pentland in face classification. It is considered the first successful example of facial recognition technology. These eigenvectors are derived from the covariance matrix of the probability distribution of the high-dimensional vector space of possible faces of human beings.* [Emphasis added.]

1.2 Basics

To start with we shall make things simple and let A be a complex matrix, and assume that there are n distinct eigenvalue-eigenvector pairs satisfying Eq. (1.1):

$$v_1, v_2, \cdots, v_j, \cdots, v_n$$
$$\lambda_1, \lambda_2, \cdots, \lambda_j, \cdots, \lambda_n$$

Note: throughout our discussion of eigenvectors the index will label different vectors, not components of a vector. The particular assignment of the index to the vectors is immaterial, e.g. we could just as well swap v_1

and v_3 in the list as long as we also swap λ_1 and λ_3. In practice a particular ordering of eigenvalues may be produced.

We shall refer to a pair (v_j, λ_j) as an eigenpair. Our assumption that A has distinct eigenpairs means that none of the v_j can be written as a linear combination of the others.

If we then form a diagonal matrix Λ of eigenvalues and a matrix V, whose columns are the v_j:

$$\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \quad V = \left(\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array} \right)$$

We then can write

$$AV = V\Lambda$$

Each column of this equation is an instance of Eq. (1.1), e.g. the second column is $Av_2 = \lambda_2 v_2$.

By assumption of distinct eigenpairs, V will be non-singular and invertible. Hence we can write

$$A = V\Lambda V^{-1} \tag{1.2}$$

This is the **eigenvalue decomposition of** A. Equivalently we may write

$$V^{-1}AV = \Lambda \tag{1.3}$$

This form is called the **the diagonalization of** A, since Λ is diagonal.

This decomposition/diagonalization will be the focus of the remainder of this lecture.

The complications

When we considered linear systems of equations, we did not need to worry about whether A was real or complex. Furthermore, we assumed without any further discussion, that A was invertible so that we could solve $Ax = b$. We deferred the issues to Maths Lecture VI. The reason is that there is a lot to say about when $Ax = b$ fails because of its connection to least squares problems and the SVD.

For the eigenvalue problem, however, we choose to have the discussion about the restrictions and potential complications right from the start. This is both because it is more essential and because there is nothing about it that is worth putting off to Maths Lecture VI.

The two main issues are easily demonstrated with a few simple matrices. Consider:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 0 & 4 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & -1 & 4 \end{pmatrix}$$

The eigenvalues of matrix A and matrix B are: $\lambda_1 = 1$ and $\lambda_2 = \lambda_3 = 4$. (The diagonal elements of an upper triangular matrix are its eigenvalues.)

Neither A nor B satisfy the assumption of distinct eigenvalues we made in arriving at the eigenvalue decomposition (1.2). However, A can be diagonalized, it is after all already diagonal. This is because it possesses

distinct eigenvectors. The corresponding matrix of eigenvectors is:

$$V = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \Rightarrow \quad v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

There are 3 distinct eigenvectors.

Matrix B, however, cannot be diagonalized. It has only 2 distinct eigenvectors

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Such a matrix is said to be **defective**. One can only diagonalize **non-defective** matrices. The lower right 2×2 corner of B is called a **Jordan block**.

Defective matrices are in fact rare (a set of measure zero). Arbitrarily small perturbations to the matrix B exist such that the perturbed matrix is non-defective. In fact almost any perturbation to B will produce a non-defective matrix. For this reason defective matrices are not a serious concern in general and we may assume that matrices are not defective.

Matrix C is an example of the more troubling case in practice – annoying, but not fundamentally problematic. C has only real entries but two of the eigenvalues and eigenvectors are complex. The eigenvalues of C are $\lambda_1 = 1$, $\lambda_2 = 4 + i$, $\lambda_3 = 4 - i$, with corresponding eigenvectors:

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ i \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ -i \end{pmatrix}$$

This case is entirely generic and is not removed by small perturbations to matrix C. We cannot, nor should not, assume this case away for simplicity of discussion. The annoyance is of two types. First, as a practical matter, when one actually finds the eigenvalues of a general real matrix, one must accommodate the return of a complex result from real input. The second annoyance is that we wish to make statements such as any non-defective matrix A has an eigenvalue decomposition (1.2). For real A this must be interpreted in one of two ways: either V and Λ are complex or Λ is block diagonal with 1×1 blocks for real eigenvalues and 2×2 blocks for complex eigenvalues. We shall call such matrices **quasi-diagonal**. Quasi refers to 2×2 blocks along the diagonal for each complex eigenvalue. For the same reason, when we come to Schur factorization, we will have **quasi-upper triangular** matrices.

These particular annoyances do not arise for complex matrices or real symmetric matrices.

1.3 A few formal statements

The following is a brief summary of some basic properties of eigenvalue problems. The reader may find proofs in standard linear algebra texts.

- Two $n \times n$ matrices A and B are **similar** if there exists an $n \times n$ invertible matrix Z such that

$$Z^{-1} A Z = B$$

A and B have the same eigenvalues. The transformation from A to B is called a **similarity transformation**. The eigenvectors w of B are related to eigenvectors v of A via:

$$Z^{-1}v = w$$

- As a special case of the above, if the invertible matrix Z is unitary or orthogonal, we denote it instead by Q. By definition of unitary/orthogonal we have $Q^* = Q^{-1}$ and hence

$$Q^*AQ = B \qquad Q^*v = w$$

- All non-defective matrices A are diagonalizable and have an eigenvalue decomposition

$$A = V\Lambda V^{-1}$$

  Λ is a (quasi-)diagonal matrix of eigenvalues. The columns of V are the eigenvectors of A.

- All Hermitian (symmetric if real) matrices A are diagonalizable by unitary (orthogonal if real) matrices

$$A = Q\Lambda Q^*$$

  The eigenvalues of Hermitian and symmetric matrices are real. The eigenvectors, the columns of Q, are orthogonal.

- The eigenvalues of a general real matrix A come in complex conjugate pairs, i.e. if λ is a complex eigenvalue of A, then $\bar{\lambda}$ is also an eigenvalue of A.

1.4 Basics of Eigenvalue Computations

Need for iterative methods

Fundamentally all eigenvalue computations must be iterative. The reason is simple and interesting. From an $n \times n$ matrix A one can define the **characteristic polynomial** $p(z)$ of degree n by

$$p(z) = \quad det(A \quad zI) = z^n \; + \cdots$$

where z is any complex number, I is the identity matrix, and det denotes determinant. The root of this polynomial are precisely the eigenvalues of A.

$$p(\lambda_j) = 0 \quad \text{for all eigenvalues } \lambda_j$$

This may have been the way you learned to compute eigenvalues and this can be taken to be the definition of the eigenvalues of A. The point here is that once $n > 4$, the eigenvalues are the roots of a polynomial of degree greater than 4 and it has been mathematically proved to be impossible to find the roots of such polynomials in closed form (Abel in 1824). There is no equivalent to the quadratic formula for $n > 4$. This is very different from solving linear equations $Ax = b$ where in principle one can work out the exact solution with a finite number of calculations.

All eigenvalue computations must have a component in which the eigenvalues and eigenvectors are obtained through some iterative process. While in principle only a finite number of digits of accuracy can ever be obtained, in practice the precision is very high.

Eigenvalues of upper triangular matrices: the Schur Decomposition

The following simple result is central to numerical eigenvalue computation: the eigenvalues of an upper triangular matrix are its diagonal entries. This result follows immediately from the characteristic polynomial $p(z)$. For upper triangular A,

$$A - zI = \begin{pmatrix} a_{11} - z & \times & \times & & \times \\ & a_{22} - z & \times & \cdots & \times \\ & & a_{33} - z & & \times \\ & & & \ddots & \times \\ & & & & a_{nn} - z \end{pmatrix}$$

and

$$p(z) = -(a_{11} - z)(a_{22} - z)(a_{33} - z) \cdots (a_{nn} - z)$$

and the roots $p(\lambda) = 0$ are clearly a_{11}, a_{22}, etc.

For a quasi-upper triangular matrix, the eigenvalues are the diagonal entries and the eigenvalues of any 2×2 blocks. The algebra is similar case but $p(z)$ will contain factors like $((a - z)(b - z) - cd))$ which can give rise to complex λ for real a, b, c, d.

This lead us to our penultimate matrix decomposition: **the Schur decomposition**. Any $n \times n$ matrix A, real, complex, defective or not, has a Schur decomposition:

$$A = Q\,T\,Q^*$$

where Q is unitary (orthogonal if A is real) and T is upper triangular (quasi-upper triangular if A is real). The columns of Q are known as **Schur vectors**. Since the similarity transformation does not change the eigenvalues, the eigenvalues of T are the same as those of A. The Schur decomposition is not unique, basically because the eigenvalues along the diagonal of T can occur in any order and each ordering results in a different Schur decomposition. This will not be important to us.

Overview of the Guts of Eigenvalue Computation

The guts of eigenvalue computation can now be stated very simply: **Perform a similarity transformation on A to transform it to Schur form, then read the eigenvalues off the diagonal of upper triangular matrix T. If eigenvectors for A are desired, they can be obtain from those of T by working backward through the similarity transformation.**

We know that we cannot expect to do this without some iteration. The overall procedure takes place in two steps (three steps if eigenvectors are also desired):

1. In step one, the matrix A is transformed to an **upper Hessenberg matrix**

   $$Q^* A Q = H$$

   Pictorially

$$Q^* \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} Q = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix}$$

where Q is unitary (orthogonal if A is real).

This part of the process is not iterative but consumes most of the actual computation time of the eigenvalue computation.

2. In step two, the matrix H is transformed to **Schur form**

$$S^* H S = T$$

where T is upper (quasi-)triangular. Pictorially

$$S^* \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix} S = \begin{pmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{pmatrix}$$

S is unitary (orthogonal if A is real). S is not a symmetric matrix. The notation for S here comes from the LAPACK manual pages.

This is the iterative step of the process. From here the eigenvalues can be read off the diagonal of T.

3. In step three, only needed if eigenvectors are required, the eigenvectors of H or T are computed and from these, and the orthogonal matrices Q and S, the eigevectors of A are obtained.

It is worth noting what the process looks like for Hermitian/symmetric matrices. The reductions in steps one and two preserve these symmetries. Hence H and T must be Hermitian/symmetric. This mean in particular that H must be tridiagonal and T must be diagonal:

$$S^* Q^* \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} Q S = S^* \begin{pmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix} S^* = \begin{pmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \times \end{pmatrix}$$

In this case the Schur vectors of H are the eigenvectors of H and as a result, step 3 is not needed for Hermitian/symmetric problems.

1.5 Further details

The Householder reflection

As just explained, the expensive part of the eigenvalue computations is the transformation of A to upper Hessenberg form H by a unitary (orthogonal) similarity transformation. In practice, this transformation is accomplished by repeated **Householder reflections**. Householder reflection is one of the most fundamental techniques in numerical linear algebra. It is sufficiently general that it could be discussed as a separate topic elsewhere, but this is where we first need to consider it.

Consider the following question. Given a vector $q = (q_1, q_2, \cdots, q_n)^T$, find a unitary (orthogonal) matrix P such that

$$Pq = P \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} = \begin{pmatrix} \pm ||q|| \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \hat{q} \tag{1.4}$$

i.e. find a matrix P that zeros all components of q except the first. We know the first component of $\tilde{q}$ must be $\pm||q||$, since multiplication by a unitary (orthogonal) matrix preserves length: $||\tilde{q}|| = ||Pq|| = ||q||$. Hence we know immediately, to within a sign, what vector $\tilde{q}$ we are transforming to, we just need the components of the matrix P which accomplishes this.

Householder realized is that there is a fast, almost trivial way to find P: view the rotation from q to $\tilde{q}$ as a reflection across a hyperplane midway between q and $\tilde{q}$. From this, P can we written

$$P = I - \frac{2}{||v||^2} vv^*, \quad \text{where} \quad \mathrm{v} = \mathrm{q} - \tilde{\mathrm{q}}.$$

This detail is not important to us. The important conceptual point is that one can easily obtain the unitary (orthogonal) matrix P which zeros all components of q except the first. Moreover, this $n \times n$ matrix can be expressed simply as the identity matrix minus the outer product of an n-vector with itself, i.e. the information about P can be stored in a single n-vector.

Repeated reflections

Using Householder reflections we can achieve the first step of eigenvalue computation – the transformation of matrix A to upper Hessenberg form. Recall, these means finding a unitary (orthogonal) similarity transformation of A that zeros all elements below the first sub-diagonal.

This is accomplished by constructing a sequence of Householder reflectors, P_1, P_2, etc. where each P_k accomplishes the necessary zeroing in one column. Pictorially,

$$A = \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} \xrightarrow{P_1} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{pmatrix} \xrightarrow{P_2} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{pmatrix} \xrightarrow{P_3} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix}$$

As one proceeds through this process, the size of the Householder reflectors decreases. Let suppose we are at step k. Let c_k denote the k^{th} column of the current matrix, the column whose elements we want to zero. Write this as:

$$c_k = \begin{pmatrix} p_k \\ q_k \end{pmatrix}$$

where p_k is of length k and q_k is of length $n - k$. The Householder reflector acts only on q_k, and hence on smaller and smaller vectors.

In effect, the matrix Q that renders A upper Hessenberg can be generated by a product of elementary matrices Q_k, each of which is composed of a $k \times k$ identity matrix and a Householder reflector of size $n - k$:

$$Q = Q_1 Q_2 Q_3 \cdots Q_{n-2} \quad \text{where} \quad \mathrm{Q_k} = \begin{pmatrix} I_{k \times k} & \\ & P_k \end{pmatrix}$$

It is, however, not necessary to construct the matrices Q_k or the final matrix Q explicitly. From the elementary reflectors one is able to compute Qx or Q^*x and one need not, unless there is a reason to, add computational work necessary to construct the matrix Q explicitly.

Complexity: the computation work of reducing an $n \times n$ matrix A to upper Hessenberg form is:

$$C(n) \sim \frac{10}{3} n^3$$

This is the most expensive computation we have yet seen.

The final steps

The computation still is not finished. From the upper Hessenberg matrix, the eigenvalues must be obtained, and if desired, the eigenvectors. We shall be brief here. The standard method for obtaining the eigenvalues is shifted QR iteration. This method converges very quickly to the desired eigenvalues and presents a minor cost compared with the Hessenberg reduction. Shifted QR iteration cannot be explained simply. However, QR iterations should not be confused with simpler QR decomposition of a matrix A. Inverse iteration, which will will discuss in Maths lecture V is another option for obtaining the eigenvectors of the upper Hessenberg matrix.

We conclude with a final point that will be important when we consider iterative methods for the eigenvalue problem. If eigenvectors are required, these are first obtained as eigenvectors of either H or T. Suppose an eigenvector w of H is produced. Then since A and H are related by $Q^* AQ = H$ the corresponding eigenvector of A is $v = Qw$. Consider a set of eigenvectors $w_1, w_2, \cdots, w_k$, where k may, but need not be as large as n. Arranging these as columns of an $n \times k$ matrix W, the corresponding matrix of k eigenvectors of A is given by:

$$V = Q\,W$$

Left multiplication by the matrix Q that rendered A upper Hessenberg produces a column matrix of eigenvectors of A from a column matrix of eigenvectors of H. You should be sure you understand this point as it will be key later on.

Similarly, from a column matrix X of eigenvectors of T, one can reconstruct the corresponding column matrix V as

$$V = Q\,S\,W$$