Numerical Linear Algebra and Matrix Analysis

Higham, Nicholas J.

2015

MIMS EPrint: **2015.103**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

Numerical Linear Algebra and Matrix Analysis[†]

Nicholas J. Higham

Matrices are ubiquitous in applied mathematics. Ordinary differential equations (ODEs) and partial differential equations (PDEs) are solved numerically by finite difference or finite element methods, which lead to systems of linear equations or matrix eigenvalue problems. Nonlinear equations and optimization problems are typically solved using linear or quadratic models, which again lead to linear systems.

Solving linear systems of equations is an ancient task, undertaken by the Chinese around 1AD, but the study of matrices per se is relatively recent, originating with Arthur Cayley's 1858 "A Memoir on the Theory of Matrices". Early research on matrices was largely theoretical, with much attention focused on the development of canonical forms, but in the 20th century the practical value of matrices started to be appreciated. Heisenberg used matrix theory as a tool in the development of quantum mechanics in the 1920s. Early proponents of the systematic use of matrices in applied mathematics included Frazer, Duncan, and Collar, whose 1938 book *Elementary Matrices and Some Applications to Dynamics and Differential Equations* emphasized the important role of matrices in differential equations and mechanics. The continued growth of matrices in applications, together with the advent of mechanical and then digital computing devices, allowing ever larger problems to be solved, created the need for greater understanding of all aspects of matrices from theory to computation.

This article treats two closely related topics: matrix analysis, which is the theory of matrices with a focus on aspects relevant to other areas of mathematics, and numerical linear algebra (also called matrix computations), which is concerned with the construction and analysis of algorithms for solving matrix problems as well as related topics such as problem sensitivity and rounding error analysis.

Important themes that are discussed in this article include the matrix factorization paradigm, the use of unitary transformations for their numerical stability, exploitation of matrix structure (such as sparsity, symmetry, and definiteness), and the design of algorithms to exploit evolving computer architectures.

Throughout the article, uppercase letters are used for matrices and lower case letters for vectors and scalars. Matrices and vectors are assumed to be complex, unless otherwise stated, and $A^* = (\overline{a_{ji}})$ denotes the conjugate transpose of $A = (a_{ij})$. An unsubscripted norm $\|\cdot\|$ denotes a general vector norm and the corresponding subordinate matrix norm. Particular norms used here are the 2-norm $\|\cdot\|_2$ and the Frobenius norm $\|\cdot\|_F$. The notation "$i = 1\colon n$" means that the integer variable i takes on the values $1, 2, \ldots, n$.

1 Nonsingularity and Conditioning

Nonsingularity of a matrix is a key requirement in many problems, such as in the solution of n linear equations in n unknowns. For some classes of matrices, nonsingularity is guaranteed. A good example is the diagonally dominant matrices. The matrix $A \in \mathbb{C}^{n\times n}$ is *strictly diagonally dominant by rows* if

$$\sum_{j\neq i} |a_{ij}| < |a_{ii}|, \quad i = 1\colon n$$

and *strictly diagonally dominant by columns* if A^* is strictly diagonally dominant by rows. Any matrix that is strictly diagonally dominant by rows or columns is nonsingular (a proof can be obtained by applying Gershgorin's theorem in section 5.1).

Since data is often subject to uncertainty we wish to gauge the sensitivity of problems to perturbations, which is done using condition numbers. An appropriate condition number for the matrix inverse is

$$\lim_{\varepsilon\to 0} \sup_{\|\Delta A\|\leqslant\varepsilon\|A\|} \frac{\|(A+\Delta A)^{-1} - A^{-1}\|}{\varepsilon\|A^{-1}\|}.$$

This expression turns out to equal $\kappa(A) = \|A\|\|A^{-1}\|$, which is called *the condition number of A with respect to inversion*. This condition number occurs in many contexts. For example, suppose A is contaminated by errors and we perform a similarity transformation $X^{-1}(A+E)X = X^{-1}AX + F$. Then $\|F\| = \|X^{-1}EX\| \leqslant \kappa(X)\|E\|$ and this bound is attainable for some E. Hence the errors can be multiplied by a factor as large as $\kappa(X)$. We therefore prefer to carry out similarity and other transformations with matrices that are *well conditioned*, that is, ones for which $\kappa(X)$ is close to its lower bound of 1. By contrast, a matrix for which κ is large is called *ill conditioned*. For any unitary matrix X,

†. Author's final version, before copy editing and cross-referencing, of: N. J. Higham. Numerical linear algebra and matrix analysis. In N. J. Higham, M. R. Dennis, P. Glendinning, P. A. Martin, F. Santosa, and J. Tanner, editors, *The Princeton Companion to Applied Mathematics*, pages 263–281. Princeton University Press, Princeton, NJ, USA, 2015.

$\kappa_2(X) = 1$, so in numerical linear algebra transformations by unitary or orthogonal matrices are preferred and usually lead to numerically stable algorithms.

In practice we often need an estimate of the matrix condition number number $\kappa(A)$ but do not wish to go to the expense of computing A^{-1} in order to obtain it. Fortunately, there are algorithms that can cheaply produce a reliable estimate of $\kappa(A)$ once a factorization of A has been computed.

Note that the determinant, $\det(A)$, is rarely computed in numerical linear algebra. Its magnitude gives no useful information about the conditioning of A, not least because of its extreme behavior under scaling: $\det(\alpha A) = \alpha^n \det(A)$.

2 Matrix Factorizations

The method of *Gaussian elimination* (GE) for solving a nonsingular linear system $Ax = b$ of n equations in n unknowns reduces the matrix A to upper triangular form and then solves for x by substitution. GE is typically described by writing down the equations $a_{ij}^{(k+1)} = a_{ij}^{(k)} - a_{ik}^{(k)} a_{kj}^{(k)} / a_{kk}^{(k)}$ (and similarly for b) that describe how the starting matrix $A = A^{(1)} = (a_{ij}^{(1)})$ changes on each of the $n-1$ steps of the elimination in its progress towards upper triangular form U. Working at the element level in this way leads to a profusion of symbols, superscripts, and subscripts that tend to obscure the mathematical structure and hinder insights being drawn into the underlying process. One of the key developments in the last century was the recognition that it is much more profitable to work at the matrix level. Thus the basic equation above is written as $A^{(k+1)} = M_k A^{(k)}$, where M_k agrees with the identity matrix except below the diagonal in the kth column, where its (i,k) element is $m_{ik} = -a_{ik}^{(k)} / a_{kk}^{(k)}$, $i = k+1\colon n$. Recurring the matrix equation gives $U := A^{(n)} = M_{n-1} \dots M_1 A$. Taking the M_k matrices over to the left-hand side leads, after some calculations, to the equation $A = LU$, where L is unit lower triangular, with (i,k) element m_{ik}. The prefix "unit" means that L has ones on the diagonal.

GE is therefore equivalent to factorizing the matrix A as the product of a lower triangular matrix and an upper triangular matrix—something that is not at all obvious from the element-level equations. Solving the linear system $Ax = b$ now reduces to the task of solving the two triangular systems $Ly = b$ and $Ux = y$.

Interpreting GE as LU factorization separates the computation of the factors from the solution of the triangular systems. It is then clear how to solve efficiently several systems $Ax_i = b_i$, $i = 1\colon r$, with different right-hand sides but the same coefficient matrix A: compute the LU factors once and then re-use them to solve for each x_i in turn.

This matrix factorization[1] viewpoint dates from around the 1940s and has been extremely successful in matrix computations. In general, a factorization is a representation of a matrix as a product of "simpler" matrices. Factorization is a tool that can be used to solve a variety of problems, as we will see below.

Two particular benefits of factorizations are unity and modularity. GE, for example, can be organized in several different ways, corresponding to different orderings of the three nested loops that it comprises, as well as the use of different blockings of the matrix elements. Yet all of them compute the same LU factorization, carrying out the same mathematical operations in a different order. Without the unifying concept of a factorization, reasoning about these GE variants would be difficult.

Modularity refers to the way that a factorization breaks a problem down into separate tasks, which can be analyzed or programmed independently. To carry out a rounding error analysis of GE we can analyze the LU factorization and the solution of the triangular systems by substitution separately and then put the analyses together. The rounding error analysis of substitution can be re-used in the many other contexts in which triangular systems arise.

An important example of the use of LU factorization is in *iterative refinement*. Suppose we have used GE to obtain a computed solution $\hat{x}$ to $Ax = b$ in floating-point arithmetic. If we form $r = b - A\hat{x}$ and solve $Ae = r$, then in exact arithmetic $y = \hat{x} + e$ is the true solution. In computing e we can reuse the LU factors of A, so obtaining y from $\hat{x}$ is inexpensive. In practice, the computation of r, e, and y is subject to rounding errors so the computed $\hat{y}$ is not equal to x. But under suitable assumptions $\hat{y}$ will be an improved approximation and we can iterate this refinement process. Iterative refinement is particularly effective if r can be computed using extra precision.

Two other key factorizations are:

- *Cholesky factorization*: for Hermitian positive definite $A \in \mathbb{C}^{n\times n}$, $A = R^*R$, where R is upper triangular with positive diagonal elements, and this factorization is unique.

1. Or decomposition—the two terms are essentially synonymous.

- *QR factorization*: for $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$, $A = QR$ where $Q \in \mathbb{C}^{m\times m}$ is unitary ($Q^*Q = I_m$) and $R \in \mathbb{C}^{m\times n}$ is upper trapezoidal, that is, $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ with $R_1 \in \mathbb{C}^{n\times n}$ upper triangular.

These two factorizations are related: if $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$ has full rank and $A = QR$ is a QR factorization, in which without loss of generality we can assume that R has positive diagonal, then $A^*A = R^*R$, so R is the Cholesky factor of A^*A.

The Cholesky factorization can be computed by what is essentially a symmetric and scaled version of GE. The QR factorization can be computed in three main ways, one of which is the classical Gram–Schmidt orthogonalization. The most widely used method constructs Q as a product of *Householder reflector*s, which are unitary matrices of the form $H = I - 2vv^*/(v^*v)$, where v is a nonzero vector. Note that H is a rank 1 perturbation of the identity and since it is Hermitian and unitary it is its own inverse, that is, it is *involutory*. The third approach builds Q as a product of *Givens rotation*s, each of which is a 2×2 matrix $\left[\begin{smallmatrix} c & s \\ -s & c \end{smallmatrix}\right]$ embedded into two rows and columns of an $m \times m$ identity matrix, where (in the real case) $c^2 + s^2 = 1$.

The Cholesky factorization helps us to make the most of the very desirable property of positive definiteness. For example, suppose A is Hermitian positive definite and we wish to evaluate the scalar $\alpha = x^*A^{-1}x$. We can rewrite it as $x^*(R^*R)^{-1}x = (x^*R^{-1})(R^{-*}x) = z^*z$, where $z = R^{-*}x$. So once the Cholesky factorization has been computed we need just one triangular solve to compute α, and of course there is no need to explicitly invert the matrix A.

A matrix factorization might involve a larger number of factors: $A = N_1N_2\dots N_k$, say. It is immediate that $A^T = N_k^T N_{k-1}^T \dots N_1^T$. This factorization of the transpose may have deep consequences in a particular application. For example, the discrete Fourier transform is the matrix–vector product $y = F_n x$, where the $n \times n$ matrix F_n has (p,q) element $\exp(-2\pi \mathrm{i}(p-1)(q-1)/n)$; F_n is a complex, symmetric matrix. The fast Fourier transform (FFT) is a way of evaluating y in $O(n \log_2 n)$ operations, as opposed to the $O(n^2)$ operations that are required by a standard matrix–vector multiplication. Many variants of the FFT have been proposed since the original 1965 paper by Cooley and Tukey. It turns out that different FFT variants correspond to different factorizations of F_n with $k = \log_2 n$ sparse factors. Some of these methods correspond simply to transposing the factorization in another method (recall that $F_n^T = F_n$), though this was not realized when the methods were developed. Transposition also plays an important role in automatic differentiation: the so-called reverse or adjoint mode can be obtained by transposing a matrix factorization representation of the forward mode.

The factorizations described in this section are in "plain vanilla" form, but all have variants that incorporate pivoting. Pivoting refers to row or column interchanges carried out at each step of the factorization as it is computed, introduced either to ensure that the factorization succeeds and is numerically stable or to produce a factorization with certain desirable properties usually associated with rank deficiency. For GE, *partial pivoting* is normally used: at the start of the kth stage of the elimination an element $a_{rk}^{(k)}$ of largest modulus in the kth column below the diagonal is brought into the (k,k) (pivot) position by interchanging rows k and r. Partial pivoting avoids dividing by zero (if $a_{kk}^{(k)} = 0$ after the interchange then the pivot column is zero below the diagonal and the elimination step can be skipped). More importantly, partial pivoting ensures numerical stability; see section 8. The overall effect of GE with partial pivoting is to produce an LU factorization $PA = LU$, where P is a permutation matrix.

Pivoted variants of Cholesky factorization and QR factorization take the form $P^TAP = R^*R$ and $AP = Q\left[\begin{smallmatrix} R \\ 0 \end{smallmatrix}\right]$, where P is a permutation matrix and R satisfies the inequalities

$$|r_{kk}|^2 \geqslant \sum_{i=k}^{j} |r_{ij}|^2, \qquad j = k+1\colon n, \quad k = 1\colon n.$$

If A is rank deficient then R has the form $R = \left[\begin{smallmatrix} R_{11} & R_{12} \\ 0 & 0 \end{smallmatrix}\right]$ with R_{11} nonsingular, and the rank of A is the dimension of R_{11}. Equally importantly, when A is nearly rank deficient this tends to be revealed by a small trailing diagonal block of R.

A factorization of great importance in a wide variety of applications is the singular value decomposition (SVD) of $A \in \mathbb{C}^{m\times n}$:

$$A = U\Sigma V^*, \quad \Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \dots, \sigma_p) \in \mathbb{R}^{m\times n}, \quad (1)$$

where $p = \min(m,n)$, $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary, and the *singular values* σ_i satisfy $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_p \geqslant 0$. For a square A ($m = n$), the 2-norm condition number is given by $\kappa_2(A) = \sigma_1/\sigma_n$.

The *polar decomposition* of $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$ is a factorization $A = UH$ in which $U \in \mathbb{C}^{m\times n}$ has orthonormal columns and $H \in \mathbb{C}^{n\times n}$ is Hermitian positive semidefinite. The matrix H is unique and is given by

$(A^*A)^{1/2}$, where the exponent $1/2$ denotes the principal square root, while U is unique if A has full rank. The polar decomposition generalizes to matrices the polar representation $z = r\,\mathrm{e}^{\mathrm{i}\theta}$ of a complex number. The Hermitian polar factor H is also known as the *matrix absolute value*, $|A|$, and is much studied in matrix analysis and functional analysis.

One reason for the importance of the polar decomposition is that it provides an optimal way to orthogonalize a matrix: a result of Fan and Hoffman (1955) says that U is the nearest matrix with orthonormal columns to A in any unitarily invariant norm (a unitarily invariant norm is one with the property that $\|UAV\| = \|A\|$ for any unitary U and V; the 2-norm and the Frobenius norm are particular examples). In various applications a matrix $A \in \mathbb{R}^{n\times n}$ that should be orthogonal drifts from orthogonality because of rounding or other errors; replacing it by the orthogonal polar factor U is then a good strategy.

The polar decomposition also solves the *orthogonal Procrustes problem*, for $A, B \in \mathbb{C}^{m\times n}$,

$$\min\{\, \|A - BQ\|_F : Q \in \mathbb{C}^{n\times n},\ Q^*Q = I \,\},$$

for which any solution Q is a unitary polar factor of B^*A. This problem comes from factor analysis and multidimensional scaling in statistics, where the aim is to see whether two data sets A and B are the same up to an orthogonal transformation.

Either of the SVD and the polar decomposition can be derived, or computed, from the other. Historically, the SVD came first (Beltrami, in 1873), with the polar decomposition three decades behind (Autonne, in 1902).

3 Distance to Singularity and Low-Rank Perturbations

The question commonly arises of whether a given perturbation of a nonsingular matrix A preserves nonsingularity. In a sense, this question is trivial. Recalling that a square matrix is nonsingular when all its eigenvalues are nonzero, and that the product of two matrices is nonsingular unless one of them is singular, from $A + \Delta A = A(I + A^{-1}\Delta A)$ we see that $A + \Delta A$ is nonsingular as long as $A^{-1}\Delta A$ has no eigenvalue equal to -1. However, this is not an easy condition to check, and in practice we may not know ΔA but only a bound for its norm. Since any norm of a matrix exceeds the modulus of every eigenvalue, a sufficient condition for $A + \Delta A$ to be nonsingular is that $\|A^{-1}\Delta A\| < 1$, which is certainly true if $\|A^{-1}\|\|\Delta A\| < 1$. This condition can be rewritten as the inequality $\|\Delta A\|/\|A\| < \kappa(A)^{-1}$, where $\kappa(A) = \|A\|\|A^{-1}\| \geqslant 1$ is the condition number introduced in section 1. It turns out that we can always find a perturbation ΔA such that $A + \Delta A$ is singular and $\|\Delta A\|/\|A\| = \kappa(A)^{-1}$. It follows that the *relative distance to singularity*

$$d(A) = \min\{\, \|\Delta A\|/\|A\| : A + \Delta A \text{ is singular} \,\} \qquad (2)$$

is given by $d(A) = \kappa(A)^{-1}$. This reciprocal relation between problem conditioning and the distance to a singular problem (one with an infinite condition number) is common to a variety of problems in linear algebra and control theory, as shown by James Demmel in the 1980s.

We may want a more refined test for whether $A + \Delta A$ is nonsingular. To obtain one we will need to make some assumptions about the perturbation. Suppose that ΔA has rank 1: $\Delta A = xy^*$, for some vectors x and y. From the analysis above we know that $A + \Delta A$ will be nonsingular if $A^{-1}\Delta A = A^{-1}xy^*$ has no eigenvalue equal to -1. Using the fact that the nonzero eigenvalues of AB are the same as those of BA for any conformable matrices A and B, we see that the nonzero eigenvalues of $(A^{-1}x)y^*$ are the same as those of $y^*A^{-1}x$. Hence $A + xy^*$ is nonsingular as long as $y^*A^{-1}x \neq -1$.

Now that we know when $A + xy^*$ is nonsingular we might ask if there is an explicit formula for the inverse. Since $A + xy^* = A(I + A^{-1}xy^*)$ we can take $A = I$ without loss of generality. So we are looking for the inverse of $B = I + xy^*$. One way to find it is to guess that $B^{-1} = I + \theta xy^*$ for some scalar θ and equate the product with B to I, to obtain $\theta(1 + y^*x) + 1 = 0$. Thus $(I + xy^*)^{-1} = I - xy^*/(1 + y^*x)$. The corresponding formula for $(A + xy^*)^{-1}$ is

$$(A + xy^*)^{-1} = A^{-1} - A^{-1}xy^*A^{-1}/(1 + y^*A^{-1}x),$$

which is known as the *Sherman–Morrison formula*. This formula and its generalizations originate in the 1940s and have been rediscovered many times. The corresponding formula for a rank p perturbation is the *Sherman–Morrison–Woodbury formula*: for $U, V \in \mathbb{C}^{n\times p}$,

$$(A + UV^*)^{-1} = A^{-1} - A^{-1}U(I + V^*A^{-1}U)^{-1}V^*A^{-1}.$$

Important applications of these formulae are in optimization, where rank-1 or rank-2 updates are made to Hessian approximations in quasi-Newton methods and to basis matrices in the simplex method. More generally, the task of updating the solution to a problem after a coefficient matrix has undergone a low-rank change, or has had a row or column added or removed, arises in

many applications, including signal processing, where new data is continually being received and old data is discarded.

The minimal distance in the definition (2) of the distance to singularity $d(A)$ can be shown to be attained for a rank-1 matrix ΔA. Rank-1 matrices often feature in the solutions of matrix optimization problems.

4 Computational Cost

In order to compare competing methods and predict their practical efficiency we need to know their computational cost. Traditionally, computational cost has been measured by counting the number of scalar arithmetic operations and retaining only the highest order terms in the total. For example, using GE we can solve a system of n linear equations in n unknowns with $n^3/3 + O(n^2)$ additions, $n^3/3 + O(n^2)$ multiplications, and $O(n)$ divisions. This is typically summarized as $2n^3/3$ flops, where a *flop* denotes any of the scalar operations $+, -, *, /$. Most standard problems involving $n \times n$ matrices can be solved with a cost of order n^3 flops or less, so the interest is in the exponent (1, 2, or 3) and the constant of the dominant term. However, the costs of moving data around a computer's hierarchical memory and the costs of communicating between different processors on a multiprocessor system can be equally important. Simply counting flops does not therefore necessarily give a good guide to performance in practice.

Seemingly trivial problems can offer interesting challenges as regards minimizing arithmetic costs. For matrices A, B, and C of any dimensions such that the product ABC is defined, how should we compute the product? The associative law for matrix multiplication tells us that $(AB)C = A(BC)$, but this mathematical equivalence is not a computational one. To see why, note that for three vectors $a, b, c \in \mathbb{R}^n$ we can write

$$\underbrace{(ab^*)}_{n\times n} c = a \underbrace{(b^*c)}_{1\times 1}.$$

Evaluation of the left-hand side requires $O(n^2)$ flops, as there is an outer product ab^* and then a matrix–vector product to evaluate, while evaluation of the right-hand side requires just $O(n)$ flops, as it involves only vector operations: an inner product and a vector scaling. One should always be alert for opportunities to use the associative law to save computational effort.

5 Eigenvalue Problems

The eigenvalue problem $Ax = \lambda x$ for a square matrix $A \in \mathbb{C}^{n\times n}$, which seeks an eigenvalue $\lambda \in \mathbb{C}$ and an eigenvector $x \neq 0$, arises in many forms. Depending on the application we may want all the eigenvalues or just a subset, such as the 10 that have the largest real part, and eigenvectors may or may not be required as well. Whether the problem is Hermitian or non-Hermitian changes its character greatly. In particular, while a Hermitian matrix has real eigenvalues and a linearly independent set of n eigenvectors that can be taken to be orthonormal, the eigenvalues of a non-Hermitian matrix can be anywhere in the complex plane and there may not be a set of eigenvectors that spans $\mathbb{C}^n$.

5.1 Bounds and Localization

One of the first questions to ask is whether we can find a finite region containing the eigenvalues. The answer is yes, because $Ax = \lambda x$ implies $|\lambda|\|x\| = \|Ax\| \leqslant \|A\|\|x\|$, and hence $|\lambda| \leqslant \|A\|$. So all the eigenvalues lie in a disc of radius $\|A\|$ about the origin. More refined bounds are provided by Gershgorin's theorem.

Theorem 1 (Gershgorin's theorem, 1931)**.** *The eigenvalues of $A \in \mathbb{C}^{n\times n}$ lie in the union of the n discs in the complex plane*

$$D_i = \left\{ z \in \mathbb{C} : |z - a_{ii}| \leqslant \sum_{j\neq i} |a_{ij}| \right\}, \qquad i = 1\colon n.$$

An extension of the theorem says that if k discs form a connected region that is isolated from the other discs then there are precisely k eigenvalues in this region. The Gershgorin discs for the matrix

$$\begin{bmatrix} -1 & 1/3 & 1/3 & 1/3 \\ 3/2 & -2 & 0 & 0 \\ 1/2 & 0 & 3 & 1/4 \\ 1 & 0 & -1 & 6 \end{bmatrix} \tag{3}$$

are shown in figure 1. We can conclude that there is one eigenvalue in the disc centered at 3, one in the disc centered at 6, and two in the union of the other two discs.

Gershgorin's theorem is most useful for matrices that are close to diagonal, such as those eventually produced by the Jacobi iterative method for eigenvalues of Hermitian matrices. Improved estimates can be sought by applying Gershgorin's theorem to a matrix $D^{-1}AD$ similar to A, with the diagonal matrix D chosen in an attempt to isolate and shrink the discs. Many variants

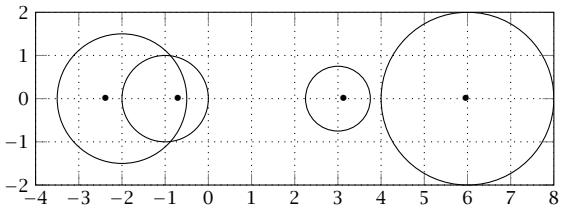

Figure 1 Gershgorin discs for the matrix in (3); the eigenvalues are marked as solid dots.

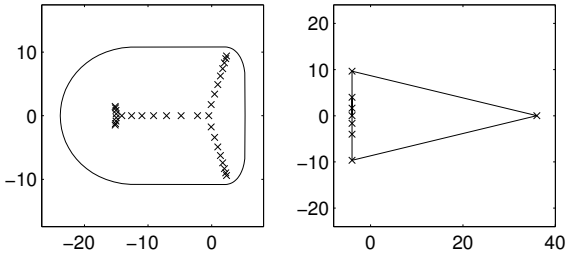

Figure 2 Fields of values for a pentadiagonal Toeplitz matrix (left) and a circulant matrix (right), both of dimension 32. The eigenvalues are denoted by crosses.

of Gershgorin's theorem exist with discs replaced by other shapes.

The spectral radius $\rho(A)$ (the largest absolute value of any eigenvalue of A) satisfies $\rho(A) \leqslant \|A\|$, as shown above, but this inequality can be arbitrarily weak, as the matrix $\left[\begin{smallmatrix} 1 & \theta \\ 0 & 1 \end{smallmatrix}\right]$ shows for $|\theta| \gg 1$. It is natural to ask whether there are any sharper relations between the spectral radius and norms. One answer is the equality

$$\rho(A) = \lim_{k\to\infty} \|A^k\|^{1/k}. \tag{4}$$

Another is the result that given any $\varepsilon > 0$ there is a norm such that $\|A\| \leqslant \rho(A) + \varepsilon$; however, the norm depends on A. This result can be used to give a proof of the fact, discussed in the article on the Jordan canonical form, that the powers of A converge to zero if $\rho(A) < 1$.

The *field of values*, also known as the *numerical range*, is a tool that can be used for localization and many other purposes. It is defined for $A \in \mathbb{C}^{n\times n}$ by

$$F(A) = \left\{ \frac{z^*Az}{z^*z} : 0 \neq z \in \mathbb{C}^n \right\}.$$

The set $F(A)$ is compact and convex (a nontrivial property proved by Toeplitz and Hausdorff) and it contains all the eigenvalues of A. For normal matrices it is the convex hull of the eigenvalues. The *normal matrices* A are those for which $AA^* = A^*A$, and they include the Hermitian, the skew-Hermitian, and the unitary matrices. For a Hermitian matrix $F(A)$ is a segment of the real axis while for a skew-Hermitian matrix it is a segment of the imaginary axis. Figure 2 illustrates two fields of values, the second of which is the convex hull of the eigenvalues because a circulant matrix is normal.

5.2 Eigenvalue Sensitivity

If A is perturbed how much do its eigenvalues change? This question is easy to answer for a simple eigenvalue λ—one that has algebraic multiplicity 1. We need the notion of a *left eigenvector* of A corresponding to λ, which is a nonzero vector y such that $y^*A = \lambda y^*$. If λ is simple with right and left eigenvectors x and y, respectively, then there is an eigenvalue $\lambda + \Delta\lambda$ of $A+\Delta A$ such that $\Delta\lambda = y^*\Delta Ax/(y^*x)+O(\|\Delta A\|^2)$ and so

$$|\Delta\lambda| \leqslant \frac{\|y\|_2\|x\|_2}{|y^*x|}\|\Delta A\| + O(\|\Delta A\|^2).$$

The term $\|y\|_2\|x\|_2/|y^*x|$ can be shown to be an (absolute) condition number for λ. It is at least 1 and tends to infinity as y and x approach orthogonality (which can never exactly be achieved for simple λ), so λ can be very ill conditioned. However if A is Hermitian then we can take $y = x$ and the bound simplifies to $|\Delta\lambda| \leqslant \|\Delta A\| + O(\|\Delta A\|^2)$, so all the eigenvalues of a Hermitian matrix are perfectly conditioned.

Much research has been done to obtain eigenvalue perturbation bounds under both weaker and stronger assumptions about the problem. Suppose we drop the requirement that λ is simple. Consider the matrix and perturbation

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \Delta A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \varepsilon & 0 & 0 \end{bmatrix}.$$

The eigenvalues of A are all zero and those of $A + \Delta A$ are the third roots of ε. The change in the eigenvalue is proportional not to ε but to a fractional power of ε. In general, the sensitivity of an eigenvalue depends on the Jordan structure for that eigenvalue.

5.3 Companion Matrices and the Characteristic Polynomial

The eigenvalues of a matrix A are the roots of its characteristic polynomial, $\det(\lambda I - A)$. Conversely, associated with the polynomial

$$p(\lambda) = \lambda^n - a_{n-1}\lambda^{n-1} - \cdots - a_0$$

is the companion matrix

$$C = \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & \dots & a_0 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \ddots & & 0 \\ \vdots & & \ddots & 0 & \vdots \\ 0 & \dots & \dots & 1 & 0 \end{bmatrix},$$

and the eigenvalues of C are the roots of p.

This relation means that the roots of a polynomial can be found by computing the eigenvalues of an $n \times n$ matrix, and this approach is used by some computer codes, for example the `roots` function of MATLAB. While standard eigenvalue algorithms do not exploit the structure of C, this approach has proved competitive with specialist polynomial root-finding algorithms. Another use for the relation is to obtain bounds for roots of polynomials from bounds for matrix eigenvalues, and vice versa.

Companion matrices have many interesting properties. For example, any nonderogatory $n \times n$ matrix is similar to a companion matrix. Companion matrices therefore have featured strongly in matrix analysis and also in control theory. However, similarity transformations to companion form are little used in practice because of problems with ill conditioning and numerical instability.

Returning to the characteristic polynomial, $p(\lambda) = \det(\lambda I - A) = \lambda^n - a_{n-1}\lambda^{n-1} - \cdots - a_0$, we know that $p(\lambda_i) = 0$ for every eigenvalue λ_i of A. The *Cayley–Hamilton theorem* says that $p(A) = A^n - a_{n-1}A^{n-1} - \cdots - a_0 I = 0$ (which cannot be obtained simply by putting "$\lambda = A$" in the previous expression!). Hence the nth power of A, and inductively all higher powers, are expressible as a linear combination of I, A, $\dots$, A^{n-1}. Moreover, if A is nonsingular then from $A^{-1}p(A) = 0$ it follows that A^{-1} can also be written as a polynomial in A of degree at most $n-1$. These relations are not useful for practical computation because the coefficients a_i can vary tremendously in magnitude and it is not possible to compute them to high relative accuracy.

5.4 Eigenvalue Inequalities for Hermitian Matrices

The eigenvalues of Hermitian matrices $A \in \mathbb{C}^{n\times n}$, which in this section we order $\lambda_n \leqslant \cdots \leqslant \lambda_1$, satisfy many beautiful inequalities. Among the most important are those in the *Courant–Fischer theorem* (1905), which states that every eigenvalue is the solution of a min-max problem over a suitable subspace S of $\mathbb{C}^n$:

$$\lambda_i = \min_{\dim(S)=n-i+1} \max_{0\neq x\in S} \frac{x^*Ax}{x^*x}.$$

Special cases are $\lambda_n = \min_{x\neq 0} x^*Ax/(x^*x)$ and $\lambda_1 = \max_{x\neq 0} x^*Ax/(x^*x)$.

Taking x to be a unit vector e_i in the previous formula for λ_1 gives $\lambda_1 \geqslant a_{ii}$ for all i. This inequality is just the first in a sequence of inequalities relating sums of eigenvalues to sums of diagonal elements, obtained by Schur in 1923:

$$\sum_{i=1}^{k} \lambda_i \geqslant \sum_{i=1}^{k} \widetilde{a}_{ii}, \qquad k = 1\colon n, \tag{5}$$

where $\{\widetilde{a}_{ii}\}$ is the set of diagonal elements of A arranged in decreasing order: $\widetilde{a}_{11} \geqslant \cdots \geqslant \widetilde{a}_{nn}$. There is equality for $k = n$, since both sides equal $\operatorname{trace}(A)$. These inequalities say that the vector $[\lambda_1, \dots, \lambda_n]$ of eigenvalues *majorizes* the vector $[\widetilde{a}_{11}, \dots, \widetilde{a}_{nn}]$ of diagonal elements.

In general there is no useful formula for the eigenvalues of a sum $A + B$ of Hermitian matrices. However, the Courant–Fischer theorem yields the upper and lower bounds

$$\lambda_k(A) + \lambda_n(B) \leqslant \lambda_k(A + B) \leqslant \lambda_k(A) + \lambda_1(B),$$

from which it follows that $|\lambda_k(A + B) - \lambda_k(A)| \leqslant \max(|\lambda_n(B)|, |\lambda_1(B)|) = \|B\|_2$. The latter inequality again shows that the eigenvalues of a Hermitian matrix are well conditioned under perturbation.

The *Cauchy interlace theorem* has a different flavor. It relates the eigenvalues of successive leading principal submatrices $A_k = A(1\colon k, 1\colon k)$ by

$$\begin{aligned} \lambda_{k+1}(A_{k+1}) &\leqslant \lambda_k(A_k) \leqslant \lambda_k(A_{k+1}) \\ &\leqslant \cdots \leqslant \lambda_2(A_{k+1}) \leqslant \lambda_1(A_k) \leqslant \lambda_1(A_{k+1}) \end{aligned}$$

for $k = 1\colon n-1$, showing that the eigenvalues of A_k interlace those of A_{k+1}.

In 1962 Alfred Horn made a conjecture that a certain set of linear inequalities involving real numbers α_i, β_i, and γ_i, $i = 1\colon n$, is necessary and sufficient for the existence of $n \times n$ Hermitian matrices A, B, and C with eigenvalues the α_i, β_i, and γ_i, respectively, such that $C = A + B$. The conjecture was open for many years but was finally proved to be true in papers published by Klyachko in 1998 and Knutson and Tao in 1999, which exploit deep connections with algebraic geometry, representations of Lie groups, and quantum cohomology.

5.5 Solving the Non-Hermitian Eigenproblem

The simplest method for computing eigenvalues, the *power method*, computes just one: the largest in modulus. It comprises repeated multiplication of a starting

vector x by A. Since the resulting sequence is liable to overflow or underflow in floating-point arithmetic one normalizes the vector after each iteration. Therefore one step of the power method has the form $x \leftarrow Ax$, $x \leftarrow \nu^{-1}x$, where $\nu = x_j$ with $|x_j| = \max_i |x_i|$. If A has a unique eigenvalue λ of largest modulus and the starting vector has a component in the direction of the corresponding eigenvector then ν converges to λ and x converges to the corresponding eigenvector. The power method is most often applied to $(A-\mu I)^{-1}$, where μ is an approximation to an eigenvalue of interest. In this form it is known as *inverse iteration* and convergence is to the eigenvalue closest to μ. We now turn to methods that compute all the eigenvalues.

Since similarities $X^{-1}AX$ preserve the eigenvalues and change the eigenvectors in a controlled way, carrying out a sequence of similarity transformations to reduce A to a simpler form is a natural way to tackle the eigenproblem. Some early methods used nonunitary X, but such transformations are now avoided because of numerical instability when X is ill conditioned. Since the 1960s the focus has been on using unitary similarities to compute the *Schur decomposition* $A = QTQ^*$, where Q is unitary and T is upper triangular. The diagonal entries of T are the eigenvalues of A, and they can be made to appear in any order by appropriate choice of Q. The first k columns of Q span an invariant subspace corresponding to the eigenvalues $t_{11}, \ldots, t_{kk}$. Eigenvectors can be obtained by solving triangular systems involving T.

For some matrices the Schur factor T is diagonal; these are precisely the normal matrices defined in section 5.1. The *real Schur decomposition* contains only real matrices when A is real: $A = QRQ^T$, where Q is orthogonal and R is real upper quasi-triangular, which means that R is upper triangular except for 2×2 blocks on the diagonal corresponding to complex conjugate eigenvalues.

The standard algorithm for solving the non-Hermitian eigenproblem is the *QR algorithm*, which was proposed independently by John Francis and Vera Kublanovskaya in 1961. The matrix $A \in \mathbb{C}^{n\times n}$ is first unitarily reduced to upper Hessenberg form $H = U^*AU$ ($h_{ij} = 0$ for $i > j+1$), with U a product of Householder matrices. The QR iteration constructs a sequence of upper Hessenberg matrices beginning with $H_1 = H$ defined by $H_k - \mu_k I =: Q_k R_k$ (QR factorization, computed using Givens rotations), $H_{k+1} := R_k Q_k + \mu_k I$, where the μ_k are shifts chosen to accelerate the convergence of H_k to upper triangular form. It is easy to check that $H_{k+1} = Q_k^* H_k Q_k$, so the QR iteration carries out a sequence of unitary similarity transformations.

Why the QR iteration works is not obvious but can be elegantly explained by analyzing the subspaces spanned by the columns of Q_k. To produce a practical and efficient algorithm various refinements of the iteration are needed, which include

- deflation, whereby when an element on the first subdiagonal of H_k becomes small, that element is set to zero and the problem is split into two smaller problems that are solved independently,
- a double shift technique for real A that allows two QR steps with complex conjugate shifts to be carried out entirely in real arithmetic and gives convergence to the real Schur form,
- a multishift technique for including m different shifts in a single QR iteration.

A proof of convergence is lacking for all current shift strategies. Implementations introduce a random shift when convergence appears to be stagnating. The QR algorithm works very well in practice and continues to be the method of choice for the non-Hermitian eigenproblem.

5.6 Solving the Hermitian Eigenproblem

The eigenvalue problem for Hermitian matrices is easier to solve than that for non-Hermitian matrices and the range of available numerical methods is much wider.

To solve the complete Hermitian eigenproblem we need to compute the *spectral decomposition* $A = QDQ^*$, where $D = \operatorname{diag}(\lambda_i)$ contains the eigenvalues and the columns of the unitary matrix Q are the corresponding eigenvectors. Many methods begin by unitary reduction to tridiagonal form $T = U^*AU$, where $t_{ij} = 0$ for $|i-j| > 1$ and the unitary matrix U is constructed as a product of Householder matrices. The eigenvalue problem for T is much simpler, though still nontrivial. The most widely used method is the QR algorithm, which has the same form as in the non-Hermitian case but with the upper Hessenberg H_k replaced by the Hermitian tridiagonal T_k and the shifts chosen to accelerate the convergence of T_k to diagonal form. The Hermitian QR algorithm with appropriate shifts has been proved to converge at a cubic rate.

Another method for solving the Hermitian tridiagonal eigenproblem is the *divide and conquer method*.

This method decouples T in the form

$$T = \begin{bmatrix} T_{11} & 0 \\ 0 & T_{22} \end{bmatrix} + \alpha v v^*,$$

where only the trailing diagonal element of T_{11} and the leading diagonal element of T_{22} differ from the corresponding elements of T and hence the vector v has only two nonzero elements. The eigensystems of T_{11} and T_{22} are found by applying the method recursively, yielding $T_{11} = Q_1 \Lambda_1 Q_1^*$ and $T_{22} = Q_2 \Lambda_2 Q_2^*$. Then

$$\begin{aligned} T &= \begin{bmatrix} Q_1 \Lambda_1 Q_1^* & 0 \\ 0 & Q_2 \Lambda_2 Q_2^* \end{bmatrix} + \alpha v v^* \\ &= \operatorname{diag}(Q_1, Q_2)\left(\operatorname{diag}(\Lambda_1, \Lambda_2) + \alpha \tilde{v} \tilde{v}^*\right) \operatorname{diag}(Q_1, Q_2)^*, \end{aligned}$$

where $\tilde{v} = \operatorname{diag}(Q_1, Q_2)^* v$. The eigensystem of a rank-1 perturbed diagonal matrix $D + \rho z z^*$ can be found by solving the *secular equation* obtained by equating the characteristic polynomial to zero:

$$f(\lambda) = 1 + \rho \sum_{j=1}^{n} \frac{|z_j|^2}{d_{jj} - \lambda} = 0.$$

Putting the pieces together yields the overall eigendecomposition.

Other methods are suitable for computing just a portion of the spectrum. Suppose we want to compute the kth smallest eigenvalue of T and that we can somehow compute the integer $N(x)$ equal to the number of eigenvalues of T that are less than or equal to x. Then we can apply the bisection method to $N(x)$ to find the point where $N(x)$ jumps from $k-1$ to k. We can compute $N(x)$ by making use of the following result about the *inertia* of a Hermitian matrix, defined by $\operatorname{inertia}(A) = (\nu, \zeta, \pi)$, where ν is the number of negative eigenvalues, ζ is the number of zero eigenvalues, and π is the number of positive eigenvalues.

Theorem 2 (Sylvester's inertia theorem)**.** *If A is Hermitian and M is nonsingular then* $\operatorname{inertia}(A) = \operatorname{inertia}(M^* A M)$.

Sylvester's inertia theorem says that the number of negative, zero, and positive eigenvalues does not change under *congruence transformation*s. By using GE we can factorize[2] $T - xI = LDL^*$, where D is diagonal and L is unit lower bidiagonal (a bidiagonal matrix is one that is both triangular and tridiagonal). Then $\operatorname{inertia}(T - xI) = \operatorname{inertia}(D)$, so the number of negative diagonal or zero elements of D equals the number of eigenvalues of $T - xI$ less than or equal to 0, which is the number of eigenvalues of T less than or equal to x, that is, $N(x)$. The LDL* factors of a tridiagonal matrix can be computed in $O(n)$ flops, so this bisection process is efficient. An alternative approach can be built by using properties of *Sturm sequences*, which are sequences comprising the characteristic polynomials of leading principal submatrices of $T - \lambda I$.

2. The factorization may not exist, but if it does not we can simply perturb T slightly and try again without any loss of numerical stability.

5.7 Computing the SVD

For a rectangular matrix $A \in \mathbb{C}^{m \times n}$ the eigenvalues of the Hermitian matrix $\left[\begin{smallmatrix} 0 & A \\ A^* & 0 \end{smallmatrix}\right]$ of dimension $m+n$ are plus and minus the nonzero singular values of A along with $m + n - 2\min(m,n)$ zeros. Hence the SVD can be computed via the eigendecomposition of this larger matrix. However, this would be inefficient, and instead one uses algorithms that work directly on A and are analogues of the algorithms for Hermitian matrices. The standard approach is to reduce A to bidiagonal form B by Householder transformations applied on the left and the right and then to apply an adaptation of the QR algorithm that works on the bidiagonal factor (and implicitly applies the QR algorithm to the tridiagonal matrix $B^* B$).

5.8 Generalized Eigenproblems

The *generalized eigenvalue problem* (GEP) $Ax = \lambda Bx$, with $A, B \in \mathbb{C}^{n \times n}$, can be converted into a standard eigenvalue problem if B (say) is nonsingular: $B^{-1}Ax = \lambda x$. However, such a transformation is inadvisable numerically unless B is very well conditioned. If A and B have a common null vector z the problem takes on a different character because then $(A - \lambda B)z = 0$ for *any* λ; such a problem is called *singular*. We will assume that the problem is *regular*, so that $\det(A - \lambda B) \not\equiv 0$. The linear polynomial $A - \lambda B$ is sometimes called a *pencil*.

It is convenient to write $\lambda = \alpha/\beta$, where α and β are not both zero, and rephrase the problem in the more symmetric form $\beta Ax = \alpha Bx$. If x is a nonzero vector such that $Bx = 0$ then, since the problem is assumed to be regular, $Ax \neq 0$ and so $\beta = 0$. This means that $\lambda = \infty$ is an eigenvalue. Infinite eigenvalues may seem a strange concept, but in fact they are no different in most respects to finite eigenvalues.

An important special case is the *definite generalized eigenvalue problem*, in which A and B are Hermitian and B (say) is positive definite. If $B = R^* R$ is a Cholesky factorization then $Ax = \lambda Bx$ can be rewritten as $R^{-*} A R^{-1} \cdot Rx = \lambda Rx$, which is a standard eigenproblem for the Hermitian matrix $C = R^{-*} A R^{-1}$. This

argument shows that the eigenvalues of a definite problem are all real. Definite generalized eigenvalue problems arise in many physical situations where an energy minimization principle is at work, such as in problems in engineering and physics.

A generalization of the QR algorithm called the *QZ algorithm* computes a generalization to two matrices of the Schur decomposition: $Q^*AZ = T$, $Q^*BZ = S$, where Q and Z are unitary and T and S are upper triangular. The generalized Schur decomposition yields the eigenvalues as the ratios t_{ii}/s_{ii} and enables eigenvectors to be computed by substitution.

The *quadratic eigenvalue problem* (QEP) $Q(\lambda)x = (\lambda^2 A_2 + \lambda A_1 + A_0)x = 0$, where $A_i \in \mathbb{C}^{n\times n}$, $i = 0\colon 2$, arises most commonly in the dynamic analysis of structures when the finite element method is used to discretize the original PDE into a system of second-order ODEs $A_2\ddot{q}(t) + A_1\dot{q}(t) + A_0 q(t) = f(t)$. Here, the A_i are usually Hermitian (though A_1 is skew-Hermitian in gyroscopic systems) and positive (semi)definite. Analogously to the GEP, the QEP is said to be regular if $\det(Q(\lambda)) \not\equiv 0$. The quadratic problem differs fundamentally from the linear GEP because a regular problem has $2n$ eigenvalues, which are the roots of $\det(Q(\lambda)) = 0$, but at most n linearly independent eigenvectors, and a vector may be an eigenvector for two different eigenvalues. For example, the QEP with

$$Q(\lambda) = \lambda^2 I + \lambda \begin{bmatrix} -1 & -6 \\ 2 & -9 \end{bmatrix} + \begin{bmatrix} 0 & 12 \\ -2 & 14 \end{bmatrix}$$

has eigenvalues 1, 2, 3, and 4, with eigenvectors $\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$, and $\left[\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}\right]$, respectively. Moreover, there is no Schur form for three or more matrices, that is, we cannot in general find unitary matrices U and V such that U^*A_iV is triangular for $i = 0\colon 2$.

Associated with the QEP is the matrix $Q(X) = A_2X^2 + A_1X + A_0$, with $X \in \mathbb{C}^{n\times n}$. From the relation

$$\begin{aligned} Q(\lambda) - Q(X) &= A_2(\lambda^2 I - X^2) + A_1(\lambda I - X) \\ &= (\lambda A_2 + A_2 X + A_1)(\lambda I - X) \end{aligned}$$

it is clear that if we can find a matrix X such that $Q(X) = 0$, known as a *solvent*, then we have reduced the QEP to finding the eigenvalues of X and solving one $n \times n$ GEP. For the 2×2 Q above there are five solvents, one of which is $\left[\begin{smallmatrix} 3 & 0 \\ 1 & 2 \end{smallmatrix}\right]$. The existence and enumeration of solvents is nontrivial and leads into the theory of *matrix polynomials*. In general, matrix polynomials are matrices of the form $\sum_{i=0}^{k} \lambda^i A_i$ whose elements are polynomials in a complex variable; an older term for such matrices is *λ-matrices*.

The standard approach for numerical solution of the QEP mimics the conversion of the scalar polynomial root problem into a matrix eigenproblem described in section 5.3. From the relation

$$\begin{aligned} L(\lambda)z \equiv \left(\begin{bmatrix} A_1 & A_0 \\ I & 0 \end{bmatrix} + \lambda \begin{bmatrix} A_2 & 0 \\ 0 & -I \end{bmatrix} \right) \begin{bmatrix} \lambda x \\ x \end{bmatrix} \\ = \begin{bmatrix} Q(\lambda)x \\ 0 \end{bmatrix} \end{aligned}$$

we see that the eigenvalues of the quadratic Q are the eigenvalues of the $2n \times 2n$ linear polynomial $L(\lambda)$. This is an example of an exact linearization process—thanks to the hidden λ in the eigenvector! The eigenvalues of L can be found using the QZ algorithm. The eigenvectors of L have the form $z = \left[\begin{smallmatrix} \lambda x \\ x \end{smallmatrix}\right]$, where x is an eigenvector of Q, and so x can be obtained from either the first n (if $\lambda \neq 0$) or the last n components of z.

6 Sparse Linear Systems

For linear systems coming from discretization of differential equations it is common that A is *banded*, that is, the nonzero elements lie in a band about the main diagonal. An extreme case is a *tridiagonal matrix*, of which the classic example is the second-difference matrix, illustrated for $n = 4$ by

$$A = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}, \quad A^{-1} = -\frac{1}{5} \begin{bmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}.$$

This matrix corresponds to a centered finite difference approximation to a second derivative: $f''(x) \approx (f(x+h) - 2f(x) + f(x-h))/h^2$. Note that A^{-1} is a full matrix. For banded matrices, GE produces banded LU factors and its computational cost is proportional to n times the square of the bandwidth.

A matrix is *sparse* if advantage can be taken of the zero entries, because of either their number or their distribution. A banded matrix is a special case of a sparse matrix. Sparse matrices are stored on a computer not as a square array but in a special format that records only the nonzeros and their location in the matrix. This can be done with three vectors: one to store the nonzero entries and the other two to define the row and column indices of the elements in the first vector.

Sparse matrices help to explain the tenet: *never solve a linear system $Ax = b$ by computing $x = A^{-1} \times b$*. The reasons for eschewing A^{-1} are threefold:

- Computing A^{-1} requires three times as many flops as solving $Ax = b$ by GE with partial pivoting.

- GE with partial pivoting is backward stable for solving $Ax = b$ (see section 8) but solution via A^{-1} is not.
- If A is sparse, A^{-1} is generally dense and so requires much more storage than GE with partial pivoting.

When GE is applied to a sparse matrix *fill-in* occurs when the row operations cause a zero entry to become nonzero during the elimination. To minimize the storage and the computational cost, fill-in must be avoided as much as possible. This can be done by employing row and column interchanges to choose a suitable pivot from the active submatrix. The first such strategy was introduced by Markowitz in 1957. At the kth stage, with $c_j^{(k)}$ denoting the number of nonzeros in rows k to n of column j and $r_i^{(k)}$ the number of nonzeros in columns k to n of row i, the Markowitz strategy finds the pair (r,s) that minimizes $(r_i^{(k)}-1)(c_j^{(k)}-1)$ over all nonzero potential pivots $a_{ij}^{(k)}$ and then takes $a_{rs}^{(k)}$ as the pivot. The quantity being minimized is a bound on the fill-in. In practice, the potential pivots must be restricted to those not too much smaller in magnitude than the partial pivot, in order to preserve numerical stability. The result of GE with Markowitz pivoting is a factorization $PAQ = LU$, where P and Q are permutation matrices.

The analogue of the Markowitz strategy for Hermitian positive definite matrices chooses a diagonal entry $a_{ii}^{(k)}$ as the pivot, where $r_i^{(k)}$ is minimal. This is the *minimum degree algorithm*, which has been very successful in practice. Figure 3 shows in the first row a sparse and banded symmetric positive definite matrix A of dimension 225 followed to the right by its Cholesky factor. The Cholesky factor has many more nonzeros than A. The second row shows the matrix PAP^T produced by an approximate minimum degree ordering (produced by the MATLAB `symamd` function) and its Cholesky factor. We can see that the permutations have destroyed the band structure but have greatly reduced the fill-in, producing a much sparser Cholesky factor.

As an alternative to GE for solving sparse linear systems one can apply iterative methods, described in section 9; for sufficiently large problems these are the only feasible methods.

7 Overdetermined and Underdetermined Systems

Linear systems $Ax = b$ with a rectangular matrix $A \in \mathbb{C}^{m\times n}$ are very common. They break into two categories: *overdetermined systems*, with more equations than unknowns ($m > n$), and *underdetermined systems*, with fewer equations than unknowns ($m < n$). Since in general there is no solution when $m > n$ and there are many solutions when $m < n$, extra conditions must be imposed for the problems to be well-defined. These usually involve norms and different choices of norms are possible. We will restrict our discussion mainly to the 2-norm, which is the most important case, but other choices are also of practical interest.

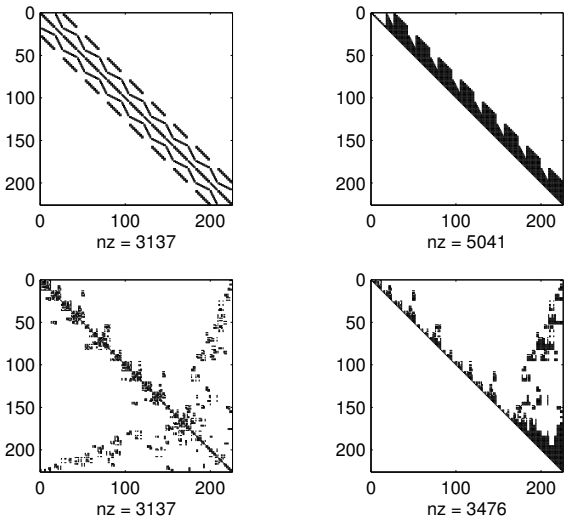

Figure 3 Sparsity plots of a symmetric positive definite matrix (left) and its Cholesky factor (right) for original matrix (first row) and reordered matrix (second row). `nz` is the number of nonzeros.

7.1 The Linear Least Squares Problem

When $m > n$ the residual $r = b - Ax$ cannot in general be made zero so we try to minimize its norm. The most common choice of norm is the 2-norm, which gives the *linear least squares problem*

$$\min_{x\in\mathbb{C}^n} \|b - Ax\|_2. \tag{6}$$

This choice can be motivated by statistical considerations (the Gauss–Markov theorem) or by the fact that the square of the 2-norm is differentiable, which makes the problem explicitly solvable. Indeed by setting the gradient of $\|b - Ax\|_2^2$ to zero we obtain the *normal equations* $A^*Ax = A^*b$, which any solution of the least squares problem must satisfy. If A has full rank then A^*A is positive definite and so there is a unique solution, which can be computed by solving the normal

equations using Cholesky factorization. For reasons of numerical stability, it is preferable to use a QR factorization: if $A = Q\begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ then the normal equations reduce to the triangular system $R_1 x = c$, where c is the first n components of $Q^* b$.

When A is rank deficient there are many least squares solutions, which vary widely in norm. A natural choice is one of minimal 2-norm, and in fact there is a unique minimal 2-norm solution, x_{LS}, given by

$$x_{LS} = \sum_{i=1}^{r} (u_i^* b/\sigma_i) v_i,$$

where

$$A = U\Sigma V^*, \quad U = [u_1, \ldots, u_m], \; V = [v_1, \ldots, v_n] \quad (7)$$

is an SVD and $r = \operatorname{rank}(A)$. The use of this formula in practice is not straightforward because a matrix stored in floating-point arithmetic will rarely have any zero singular values. Therefore r must be chosen by designating which singular values can be regarded as negligible and this choice should take account of the accuracy with which the elements of A are known.

Another choice of least squares solution in the rank-deficient case is a *basic solution*: one with at most r nonzeros. Such a solution can be computed via the QR factorization with column pivoting.

7.2 Underdetermined Systems

When $m < n$ and A has full rank, there are infinitely many solutions to $Ax = b$ and again it is natural to seek one of minimal 2-norm. There is a unique such solution $x_{LS} = A^*(AA^*)^{-1}b$, and it is best computed via a QR factorization, this time of A^*. A basic solution, with m nonzeros, can alternatively be computed. As a simple example, consider the problem "find two numbers whose sum is 5", that is, solve $[1\ 1]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 5$. A basic solution is $[5\ 0]^T$ while the minimal 2-norm solution is $[5/2\ 5/2]^T$. Minimal 1-norm solutions to underdetermined systems are important in compressed sensing.

7.3 Pseudoinverse

The analysis in the previous two subsections can be unified in a very elegant way by making use of the *Moore–Penrose pseudoinverse* A^+ of $A \in \mathbb{C}^{m\times n}$, which is defined as the unique $X \in \mathbb{C}^{n\times m}$ satisfying the Moore–Penrose conditions

$$AXA = A, \qquad XAX = X,$$
$$(AX)^* = AX, \qquad (XA)^* = XA.$$

(It is certainly not obvious that these equations have a unique solution.) In the case where A is square and nonsingular it is easily seen that A^+ is just A^{-1}. Moreover, if $\operatorname{rank}(A) = n$ then $A^+ = (A^*A)^{-1}A^*$, while if $\operatorname{rank}(A) = m$ then $A^+ = A^*(AA^*)^{-1}$. In terms of the SVD (7),

$$A^+ = V \operatorname{diag}(\sigma_1^{-1}, \ldots, \sigma_r^{-1}, 0, \ldots, 0) U^*,$$

where $r = \operatorname{rank}(A)$. The formula $x_{LS} = A^+ b$ holds for all m and n, so the pseudoinverse yields the minimal 2-norm solution to both the least squares (overdetermined) problem $Ax = b$ and an underdetermined system $Ax = b$. The pseudoinverse has many interesting properties, including $(A^+)^+ = A$, but it is not always true that $(AB)^+ = B^+A^+$.

Although the pseudoinverse is a very useful theoretical tool it is rarely necessary to compute it explicitly (just as for its special case the matrix inverse).

The pseudoinverse is just one of many ways of generalizing the notion of inverse to rectangular matrices, but it is the right one for minimum 2-norm solutions to linear systems. Other generalized inverses can be obtained by requiring only a subset of the four Moore–Penrose conditions to hold.

8 Numerical Considerations

Prior to the introduction of the first digital computers in the 1940s, numerical computations were carried out by humans, sometimes with the aid of mechanical calculators. The human involvement in a sequence of calculations meant that potentially dangerous events such as dividing by a tiny number or subtracting two numbers that agree to almost all their significant digits could be observed, their effect monitored, and possible corrective action taken—such as temporarily increasing the precision of the calculations. On the very early computers intermediate results were observed on a cathode-ray tube monitor, but this became impossible as problem sizes increased (along with available computing power). Fears were raised in the 1940s that algorithms such as GE would suffer exponential growth of errors as the problem dimension increased, due to the rapidly increasing number of arithmetic operations, each having its associated rounding error. These fears were particularly concerning given that the error growth might be unseen and unsuspected.

The subject of *rounding error analysis* grew out of the need to understand the effect on algorithms of rounding errors. The person who did the most to

develop the subject was James Wilkinson, whose influential papers and 1961 and 1965 books showed how backward error analysis can be used to obtain deep insights into numerical stability. We will discuss just two particular examples.

Wilkinson showed that when a nonsingular linear system $Ax = b$ is solved by GE in floating-point arithmetic the computed solution $\hat{x}$ satisfies

$$(A + \Delta A)\hat{x} = b, \qquad \|\Delta A\|_\infty \leqslant p(n)\rho_n u\|A\|_\infty.$$

Here $p(n)$ is a cubic polynomial, the *growth factor*

$$\rho_n = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{\max_{i,j} |a_{ij}|} \geqslant 1$$

measures the growth of elements during the elimination, and u is the unit roundoff. This is a *backward stability result*: it says that the computed solution $\hat{x}$ is the exact solution of a perturbed system. Ideally, we would like $\|\Delta A\|_\infty \leqslant u\|A\|_\infty$, which reflects the uncertainty caused by converting the elements of A to floating-point numbers. The polynomial term $p(n)$ is pessimistic and might be more realistically replaced by its square root. The danger term is the growth factor ρ_n, and the conclusion from Wilkinson's analysis is that a pivoting strategy should aim to keep ρ_n small. If no pivoting is done, ρ_n can be arbitrarily large (e.g., for $A = \left[\begin{smallmatrix} \varepsilon & 1 \\ 1 & 1 \end{smallmatrix}\right]$ with $0 < \varepsilon \ll 1$, $\rho_n \approx 1/\varepsilon$). For partial pivoting however, it can be shown that $\rho_n \leqslant 2^{n-1}$ and that this bound is attainable. In practice, ρ_n is almost always of modest size for partial pivoting ($\rho_n \leqslant 50$, say); why this should be so remains one of the great mysteries of numerical analysis!

One of the benefits of Wilkinson's backward error analysis is that it enables us to identify classes of matrices for which pivoting is not necessary, that is, for which the LU factorization $A = LU$ exists and ρ_n is nicely bounded. One such class is the matrices that are diagonally dominant by either rows or columns, for which $\rho_n \leqslant 2$.

The potential instability of GE can be attributed to the fact that A is premultiplied by a sequence of nonunitary transformations, any of which can be ill conditioned. Many algorithms, including Householder QR factorization and the QR algorithm for eigenvalues, use exclusively unitary transformations. Such algorithms are usually (but not always) backward stable, essentially because unitary transformations do not magnify errors: $\|UAV\| = \|A\|$ for any unitary U and V for the 2-norm and the Frobenius norm. As an example, the QR algorithm applied to $A \in \mathbb{C}^{n\times n}$ produces a computed upper triangular matrix $\hat{T}$ such that

$$\widetilde{Q}^*(A + \Delta A)\widetilde{Q} = \hat{T}, \qquad \|\Delta A\|_F \leqslant p(n)u\|A\|_F,$$

where $\widetilde{Q}$ is some exactly unitary matrix and $p(n)$ is a cubic polynomial. The computed Schur factor $\hat{Q}$ is not necessarily close to $\widetilde{Q}$—which in turn is not necessarily close to the exact Q!— but it is close to being orthogonal: $\|\hat{Q}^*\hat{Q} - I\|_F \leqslant p(n)u$. This distinction between the different Q matrices is an indication of the subtleties of backward error analysis. For some problems it is not clear exactly what form of backward error result it is possible to prove while obtaining useful bounds. However, the purpose of a backward error analysis is always the same: either to show that an algorithm behaves in a numerically stable way or to shed light on how it might fail to do so and to indicate what quantities should be monitored in order to identify potential instability.

9 Iterative Methods

In numerical linear algebra methods can broadly be divided into two classes: direct and iterative. Direct methods, such as GE, solve a problem in a fixed number of arithmetic operations or a variable number that in practice is fairly constant, as for the QR algorithm for eigenvalues. Iterative methods are infinite processes that must be truncated at some point when the approximation they provide is "good enough". Usually, iterative methods do not transform the matrix in question and access it only through matrix–vector products; this makes them particularly attractive for large, sparse matrices, where applying a direct method may not be practical.

We have already seen in section 5.5 a simple iterative method for the eigenvalue problem: the power method. The *stationary iterative methods* are an important class of iterative methods for solving a nonsingular linear system $Ax = b$. These methods are best described in terms of a *splitting*

$$A = M - N,$$

with M nonsingular. The system $Ax = b$ can be rewritten $Mx = Nx + b$, which suggests constructing a sequence $\{x^{(k)}\}$ from a given starting vector $x^{(0)}$ via

$$Mx^{(k+1)} = Nx^{(k)} + b. \tag{8}$$

Different choices of M and N yield different methods. The aim is to choose M in such a way that it is inexpensive to solve (8) while M is a good enough approximation to A that convergence is fast. It is easy to analyze convergence. Denote by $e^{(k)} = x^{(k)} - x$ the error in the

kth iterate. Subtracting $Mx = Nx + b$ from (8) gives $M(x^{(k+1)} - x) = N(x^{(k)} - x)$, so

$$e^{(k+1)} = M^{-1}Ne^{(k)} = \cdots = (M^{-1}N)^{k+1}e^{(0)}. \quad (9)$$

If $\rho(M^{-1}N) < 1$ then $(M^{-1}N)^k \to 0$ as $k \to \infty$ (see Jordan canonical form) and so $x^{(k)}$ converges to x, at a linear rate. In practice, for convergence in a reasonable number of iterations we need $\rho(M^{-1}N)$ to be sufficiently less than 1 and the powers of $M^{-1}N$ should not grow too large initially before eventually decaying; in other words, $M^{-1}N$ must not be too nonnormal.

Three standard choices of splitting are, with $D = \mathrm{diag}(A)$ and L and U denoting the strictly lower and strictly upper triangular parts of A, respectively,

- $M = D$, $N = -(L + U)$: *Jacobi iteration*;
- $M = D + L$, $N = -U$: *Gauss–Seidel iteration*;
- $M = \frac{1}{\omega}D + L$, $N = \frac{1-\omega}{\omega}D - U$, where $\omega \in (0, 2)$ is a relaxation parameter: *successive overrelaxation (SOR) iteration*.

Sufficient conditions for convergence are that A is strictly diagonally dominant by rows for the Jacobi iteration and that A is symmetric positive definite for the Gauss–Seidel iteration. How to choose ω so that $\rho(M^{-1}N|_\omega)$ is minimized for the SOR iteration was elucidated in the landmark 1950 PhD thesis of David Young.

The Google PageRank algorithm, which underlies Google's ordering of search results, can be interpreted as an application of the Jacobi iteration to a certain linear system involving the adjacency matrix of the graph corresponding to the whole world wide web. However, the most common use of stationary iterative methods is as preconditioners within other iterative methods.

The aim of *preconditioning* is to convert a given linear system $Ax = b$ into one that can be solved more cheaply by a particular iterative method. The basic idea is to use a nonsingular matrix W to transform the system to $(W^{-1}A)x = W^{-1}b$ in such a way that (a) the preconditioned system can be solved in fewer iterations than the original system and (b) matrix–vector multiplications with $W^{-1}A$ (which require the solution of a linear system with coefficient matrix W) are not significantly more expensive than matrix–vector multiplications with A. In general, this is a difficult or impossible task, but in many applications the matrix A has structure that can be exploited. For example, many elliptic PDE problems lead to a positive definite matrix A of the form

$$A = \begin{bmatrix} M_1 & F \\ F^T & M_2 \end{bmatrix},$$

where $M_1 z = d_1$ and $M_2 z = d_2$ are easy to solve. In this case it is natural to take $W = \mathrm{diag}(M_1, M_2)$ as the preconditioner. When A is Hermitian positive definite the preconditioned system is written in a way that preserves the structure. For example, for the Jacobi preconditioner, $D = \mathrm{diag}(A)$, the preconditioned system would be written $D^{-1/2}AD^{-1/2}\tilde{x} = \tilde{b}$, where $\tilde{x} = D^{1/2}x$ and $\tilde{b} = D^{-1/2}b$. Here, the matrix $D^{-1/2}AD^{-1/2}$ has unit diagonal and off-diagonal elements lying between -1 and 1.

The most powerful iterative methods for linear systems $Ax = b$ are the Krylov methods. In these methods each iterate $x^{(k)}$ is chosen from the shifted subspace $x^{(0)} + \mathcal{K}_k(A, r^{(0)})$ where

$$\mathcal{K}_k(A, r^{(0)}) = \mathrm{span}\{r^{(0)}, Ar^{(0)}, \ldots, A^{k-1}r^{(0)}\}$$

is a Krylov subspace of dimension k, with $r^{(k)} = b - Ax^{(k)}$. Different strategies for choosing approximations from within the Krylov subspaces yield different methods. For example, the *conjugate gradient method* (CG, for Hermitian positive definite A) and the *full orthogonalization method* (FOM, for general A) make the residual $r^{(k)}$ orthogonal to the Krylov subspace $\mathcal{K}_k(A, r^{(0)})$, while the *minimal residual method* (MINRES, for Hermitian A) and the *generalized minimal residual method* (GMRES, for general A) minimize the 2-norm of the residual over all vectors in the Krylov subspace. How to compute the vectors defined in these ways is nontrivial. It turns out that CG can be implemented with a recurrence requiring just one matrix–vector multiplication and three inner products per iteration, and MINRES is just a little more expensive. GMRES, being applicable to non-Hermitian matrices, is significantly more expensive, and it is also much harder to analyze its convergence behavior. For general matrices there are alternatives to GMRES that employ short recurrences. We mention just *BiCGSTAB*, which has the distinction that the 1992 paper by Henk van der Vorst that introduced it was the most-cited paper in mathematics of the 1990s.

Theoretically, Krylov methods converge in at most n iterations for a system of dimension n. However, in practical computation rounding errors intervene and the methods behave as truly iterative methods not having finite termination. Since n is potentially huge, a Krylov method would not be used unless a good approximate solution was obtained in many fewer than n iterations, and preconditioning plays a crucial role here. Available error bounds for a method help to guide

the choice of preconditioner, but care is needed in interpreting the bounds. To illustrate this, consider the CG method for $Ax = b$, where A is Hermitian positive definite. In the A-norm, $\|z\|_A = (z^*Az)^{1/2}$, the error on the kth step satisfies

$$\|x - x^{(k)}\|_A \leqslant 2\|x - x^{(0)}\|_A \left(\frac{\kappa_2(A)^{1/2} - 1}{\kappa_2(A)^{1/2} + 1}\right)^k,$$

where $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2$. If we can precondition A so that its 2-norm condition number is very close to 1 then fast convergence is guaranteed. However, another result says that if A has k distinct eigenvalues then CG converges in at most k iterations. Therefore a better approach might be to choose the preconditioner so that the eigenvalues of the preconditioned matrix are clustered into a small number of groups.

Another important class of iterative methods is multigrid methods, which work on a hierarchy of grids that come from a discretization of an underlying PDE (geometric multigrid) or are constructed artificially from a given matrix (algebraic multigrid).

An important practical issue is how to terminate an iteration. Popular approaches are to stop when the residual $r^{(k)} = b - Ax^{(k)}$ (suitably scaled) is small or when an estimate of the error $x - x^{(k)}$ is small. Complicating factors include the fact that the preconditioner can change the norm and a possible desire to match the error in the iterations with the discretization error in the PDE from which the linear system might have come (as there is no point solving the system to greater accuracy than the data warrants). Research in recent years has led to good understanding of these issues.

The ideas of Krylov methods and preconditioners can be applied to problems other than linear systems. A popular Krylov method for solving the least squares problem (6) is *LSQR*, which is mathematically equivalent to applying CG to the normal equations. In large-scale eigenvalue problems only a few eigenpairs are usually required. A number of methods project the original matrix onto a Krylov subspace and then solve a smaller eigenvalue problem. These include the *Lanczos method* for Hermitian matrices and the *Arnoldi method* for general matrices. Also of much current research interest are *rational Krylov methods* based on rational generalizations of Krylov subspaces.

10 Nonnormality and Pseudospectra

Normal matrices $A \in \mathbb{C}^{n\times n}$ (defined in section 5.1) have the property that they are unitarily diagonalizable: $A = QDQ^*$ for some unitary Q and diagonal $D = \mathrm{diag}(\lambda_i)$ containing the eigenvalues on its diagonal. In many respects, normal matrices have very predictable behavior. For example, $\|A^k\|_2 = \rho(A)^k$ and $\|\,\mathrm{e}^{tA}\,\|_2 = \mathrm{e}^{\alpha(tA)}$, where the *spectral abscissa* $\alpha(tA)$ is the largest real part of any eigenvalue of tA. However, matrices that arise in practice are often very nonnormal. The adjective "very" can be quantified in various ways, of which one is the Frobenius norm of the strictly upper triangular part of the upper triangular matrix T in the Schur decomposition $A = QTQ^*$. For example, the matrix $\left[\begin{smallmatrix} t_{11} & \theta \\ 0 & t_{22} \end{smallmatrix}\right]$ is nonnormal for $\theta \neq 0$ and grows increasingly nonnormal as $|\theta|$ increases.

Consider the moderately nonnormal matrix

$$A = \begin{bmatrix} -0.97 & 25 \\ 0 & -0.3 \end{bmatrix}. \tag{10}$$

While the powers of A ultimately decay to zero, since $\rho(A) = 0.97 < 1$, we see from figure 4 that initially they increase in norm. Likewise, since $\alpha(A) = -0.3 < 0$ the norm $\|\,\mathrm{e}^{tA}\,\|_2$ tends to zero as $t \to \infty$, but figure 4 shows that there is an initial hump in the plot. In stationary iterations the hump caused by a nonnormal iteration matrix $M^{-1}N$ can delay convergence, as is clear from (9). In finite precision arithmetic it can even happen that, for a sufficiently large hump, rounding errors cause the norms of the powers to plateau at the hump level and never actually converge to zero.

How can we predict the shape of the curves in figure 4? Let us concentrate on $\|A^k\|_2$. Initially it grows like $\|A\|_2^k$ and ultimately it decays like $\rho(A)^k$, the decay rate following from (4). The height of the hump is related to pseudospectra, which have been popularized by Nick Trefethen.

The *ε-pseudospectrum* of $A \in \mathbb{C}^{n\times n}$ is defined, for a given $\varepsilon > 0$, to be the set

$$\Lambda_\varepsilon(A) = \{\, z \in \mathbb{C} : z \text{ is an eigenvalue of } A + E \text{ for some } E \text{ with } \|E\|_2 < \varepsilon \,\}, \tag{11}$$

and it can also be represented, in terms of the *resolvent* $(zI - A)^{-1}$, as

$$\Lambda_\varepsilon(A) = \{\, z \in \mathbb{C} : \|(zI - A)^{-1}\|_2 > \varepsilon^{-1} \,\}.$$

The 0.001-pseudospectrum, for example, tells us the uncertainty in the eigenvalues of A if the elements are known only to three decimal places. Pseudospectra provide much insight into the effects of nonnormality of matrices and (with an appropriate extension of the definition) linear operators. For nonnormal matrices the pseudospectra are much bigger than a perturbation of

the spectrum by ε. It can be shown that for any $\varepsilon > 0$,

$$\sup_{k\geqslant 0} \|A^k\| \geqslant \frac{\rho_\varepsilon(A) - 1}{\varepsilon}, \qquad \|A^k\| \leqslant \frac{\rho_\varepsilon(A)^{k+1}}{\varepsilon},$$

where the *pseudospectral radius* $\rho_\varepsilon(A) = \max\{\, |\lambda| : \lambda \in \Lambda_\varepsilon(A)\,\}$. For A in (10) and $\varepsilon = 10^{-2}$ these inequalities give an upper bound of 230 for $\|A^3\|$ and a lower bound of 23 for $\sup_{k\geqslant 0}\|A^k\|$, and figure 5 plots the corresponding ε-pseudospectrum.

11 Structured Matrices

In a wide variety of applications the matrices have a special structure. The matrix elements might form a pattern, as for a Toeplitz matrix or a Hamiltonian matrix, the matrix may satisfy a nonlinear equation such as $A^*\Sigma A = \Sigma$, where $\Sigma = \mathrm{diag}(\pm 1)$, which yields the *pseudo-unitary matrices* A, or the submatrices may satisfy certain rank conditions (as for *quasiseparable matrices*). We discuss here two of the oldest and most studied classes of structured matrices, both of which were historically important in the analysis of iterative methods for linear systems arising from the discretization of differential equations.

11.1 Nonnegative Matrices

A *nonnegative matrix* is a real matrix all of whose entries are nonnegative. A number of important classes of matrices are subsets of the nonnegative matrices. These include adjacency matrices, stochastic matrices, and Leslie matrices (used in population modeling). Nonnegative matrices have a large body of theory, which originates with Perron in 1907 and Frobenius in 1908.

To state the celebrated Perron–Frobenius theorem we need the definition that $A \in \mathbb{R}^{n\times n}$ with $n \geqslant 2$ is *reducible* if there is a permutation matrix P such that

$$P^T A P = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix},$$

where A_{11} and A_{22} are square, nonempty submatrices, and it is *irreducible* if it is not reducible. A matrix with positive entries is trivially irreducible. A useful characterization is that A is irreducible if and only if the directed graph associated with A (which has n vertices, with an an edge connecting the ith vertex to the jth vertex if $a_{ij} \neq 0$) is strongly connected.

Theorem 3 (Perron–Frobenius). *If $A \in \mathbb{R}^{n\times n}$ is nonnegative and irreducible then*

1. *$\rho(A) > 0$,*
2. *$\rho(A)$ is an eigenvalue of A,*
3. *there is a positive vector x such that $Ax = \rho(A)x$,*
4. *$\rho(A)$ is an eigenvalue of algebraic multiplicity 1.*

To illustrate the theorem consider the following two irreducible matrices and their eigenvalues:

$$A = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}, \qquad \Lambda(A) = \{15, \pm 2\sqrt{6}\},$$

$$B = \begin{bmatrix} 0 & 0 & 6 \\ \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{3} & 0 \end{bmatrix}, \qquad \Lambda(B) = \{1, \tfrac{1}{2}(-1 \pm \sqrt{3}\mathrm{i})\}.$$

The Perron–Frobenius theorem correctly tells us that $\rho(A) = 15$ is a distinct eigenvalue of A, and that it has a corresponding positive eigenvector, which is known as the *Perron vector*. The Perron vector of A is the vector of all ones, as A forms a magic square and $\rho(A)$ is the magic sum! The Perron vector of B, which is both a Leslie matrix and a companion matrix, is $[6\ 3\ 1]^T$. There is one notable difference between A and B: for A, $\rho(A)$ exceeds the other eigenvalues in modulus, but all three eigenvalues of B have modulus 1. In fact, Perron's original version of Theorem 3 says that if A has all positive elements then $\rho(A)$ is not only an eigenvalue of A but is larger in modulus than every other eigenvalue. Note that $B^3 = I$, which provides another way to see that the eigenvalues of B all have modulus 1.

We saw in the section 9 that the spectral radius plays an important role in the convergence of stationary iterative methods, through $\rho(M^{-1}N)$, where $A = M - N$ is a splitting. In comparing different splittings we can use the result that for $A, B \in \mathbb{R}^{n\times n}$, with $|A|$ denoting the matrix $(|a_{ij}|)$,

$$|a_{ij}| \leqslant b_{ij} \quad \forall i,j \quad \Rightarrow \quad \rho(A) \leqslant \rho(|A|) \leqslant \rho(B).$$

11.2 *M*-Matrices

$A \in \mathbb{R}^{n\times n}$ is an *M-matrix* if it can be written in the form $A = sI - B$, where B is nonnegative and $s > \rho(B)$. *M*-matrices arise in many applications, a classic one being Leontief's input–output models in economics.

The special sign pattern of an *M*-matrix—positive diagonal elements and nonpositive off-diagonal elements—combines with the spectral radius condition to give many interesting characterizations and properties. For example, a nonsingular matrix A with nonpositive off-diagonal elements is an *M*-matrix if and only if A^{-1} is nonnegative. Another characterization, which makes connections with section 1, is that A is an *M*-matrix if and only if A has positive diagonal

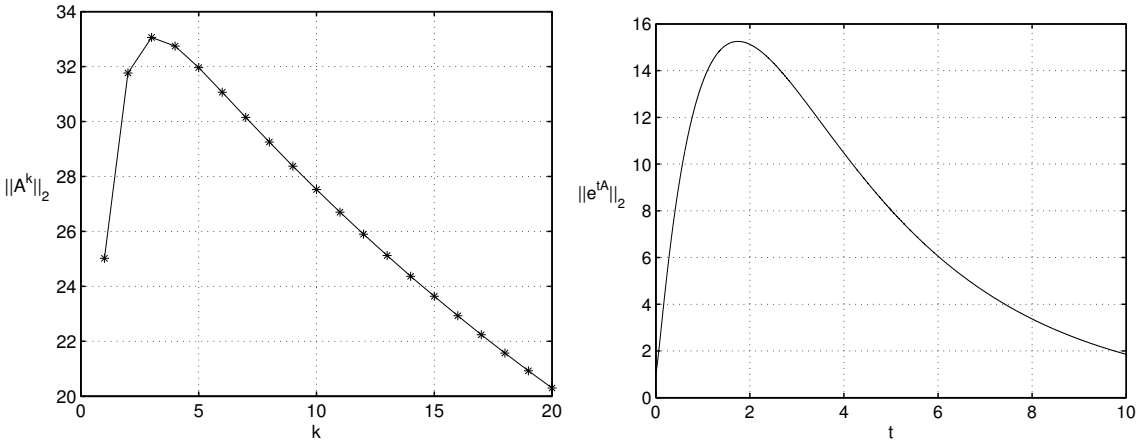

Figure 4 2-norms of powers and exponentials of 2×2 matrix A in (10).

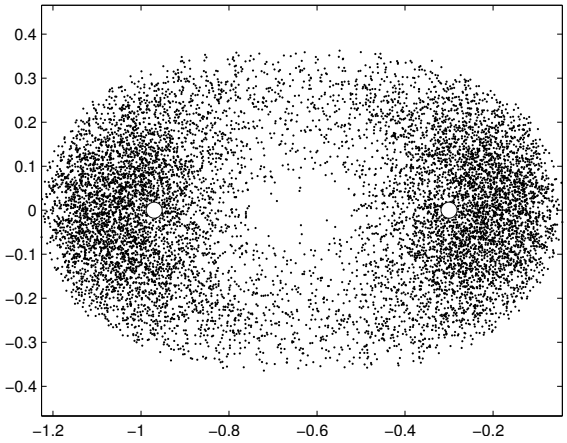

Figure 5 Approximation to 10^{-2}-pseudospectrum of A in (10) comprising eigenvalues of 5000 randomly perturbed matrices $A + E$ in (11). The eigenvalues of A are marked by white circles.

entries and AD is diagonally dominant by rows for some nonsingular diagonal matrix D.

An important source of M-matrices is discretizations of differential equations, and the archetypal example is the second-difference matrix, described at the start of section 6, which is an M-matrix multiplied by -1. For this application it is an important result that when A is an M-matrix the Jacobi and Gauss–Seidel iterations for $Ax = b$ both converge for any starting vector—a result that is part of the more general theory of regular splittings.

Another important property of M-matrices is immediate from the definition: the eigenvalues all lie in the open right half-plane. This means that M-matrices are special cases of positive stable matrices, which in turn are of great interest due to the fact that the stability of various mathematical processes is equivalent to positive (or negative) stability of an associated matrix.

The class of matrices whose inverses are M-matrices is also much-studied. To indicate why, we state a result about matrix roots. It is known that if A is an M-matrix then $A^{1/2}$ is also an M-matrix. But if A is stochastic (that is, it is nonnegative and has unit row sums), $A^{1/2}$ may not be stochastic. However, if A is both stochastic *and* the inverse of an M-matrix then $A^{1/p}$ is stochastic for all positive integers p.

12 Matrix Inequalities

There is a large body of work on matrix inequalities, ranging from classical 19th and early 20th century inequalities (some of which are described in section 5.4) to more recent contributions, which are often motivated by applications, notably in statistics, physics, and control theory. In this section we describe just a few examples, chosen for their interest or practical usefulness.

An important class of inequalities on Hermitian matrices is expressed using the *Löwner (partial) ordering* in which for Hermitian X and Y, $X \geqslant Y$ denotes that $X - Y$ is positive semidefinite while $X > Y$ denotes that $X - Y$ is positive definite. Many inequalities between real numbers generalize to Hermitian matrices in this ordering. For example, if A, B, C are Hermitian and A commutes with B and C then

$$A \geqslant 0, \quad B \leqslant C \quad \Rightarrow \quad AB \leqslant AC.$$

A function f is *matrix monotone* if it preserves the order, that is, $A \leqslant B$ implies $f(A) \leqslant f(B)$, where $f(A)$ denotes a function of a matrix. Much is known about this class of functions, including that $t^{1/2}$ and $\log t$ are matrix monotone but t^2 is not.

Many matrix inequalities involve norms. One example is

$$\| \, |A| - |B| \, \|_F \leqslant \sqrt{2} \|A - B\|_F,$$

where $A, B \in \mathbb{C}^{m\times n}$ and $|\cdot|$ is the matrix absolute value defined in section 2. This inequality can be regarded as a perturbation result that shows the matrix absolute value to be very well conditioned.

An example of an inequality that finds use in the analysis of convergence of methods in nonlinear optimization is the *Kantorovich inequality*, which for Hermitian positive definite A with eigenvalues $\lambda_n \leqslant \cdots \leqslant \lambda_1$ and $x \neq 0$ is

$$\frac{(x^*Ax)(x^*A^{-1}x)}{(x^*x)^2} \leqslant \frac{(\lambda_1 + \lambda_n)^2}{4\lambda_1\lambda_n}.$$

This inequality is attained for some x, and the left-hand side is always at least 1.

Many inequalities are available that generalize scalar inequalities for means. For example, the arithmetic–geometric mean inequality $(ab)^{1/2} \leqslant \frac{1}{2}(a+b)$ for positive scalars has an analogue for Hermitian positive definite A and B in the inequality $A \# B \leqslant \frac{1}{2}(A+B)$, where $A \# B$ is the geometric mean defined as the unique Hermitian positive definite solution to $XA^{-1}X = B$. The geometric mean also satisfies the extremal property

$$A \# B = \max \left\{ X : X = X^*, \, \begin{bmatrix} A & X \\ X & B \end{bmatrix} \geqslant 0 \right\},$$

which hints at *matrix completion problems*, in which the aim is to choose missing elements of a matrix in order to achieve some goal, which could be to satisfy a particular matrix property or, as here, to maximize an objective function. Another mean for Hermitian positive definite matrices (and applicable more generally), is the *log-Euclidean mean*, $\exp(\frac{1}{2}(\log A + \log B))$, where log is the principal logarithm, which is used in image registration, for example.

Finally, we mention an inequality for the matrix exponential. Although there is no simple relation between e^{A+B} and $\mathrm{e}^A \, \mathrm{e}^B$ in general, for Hermitian A and B the inequality $\mathrm{trace}(\mathrm{e}^{A+B}) \leqslant \mathrm{trace}(\mathrm{e}^A \, \mathrm{e}^B)$ was proved independently by S. Golden and J. Thompson in 1965. Originally of interest in statistical mechanics, the Golden–Thompson inequality has more recently found use in random matrix theory. Again for Hermitian A and B, the related inequalities $\| \, \mathrm{e}^{A+B} \, \| \leqslant \| \, \mathrm{e}^{A/2} \, \mathrm{e}^B \, \mathrm{e}^{A/2} \, \| \leqslant \| \, \mathrm{e}^A \, \mathrm{e}^B \, \|$ hold for any unitarily invariant norm.

13 Library Software

From the early days of digital computing the benefits of providing library subroutines for carrying out basic operations such as the addition of vectors and the formation of vector inner products was recognized. Over the ensuing years many matrix computation research codes were published, including in the *Linear Algebra* volume of the *Handbook for Automatic Computation* (1971) and in the *Collected Algorithms of the ACM*. Starting in the 1970s the concept of standardized subprograms was developed in the form of the *Basic Linear Algebra Subprograms* (BLAS), which are specifications for vector (level 1), matrix–vector (level 2), and matrix–matrix (level 3) operations. The BLAS have been widely adopted, and highly optimized implementations are available for most machines. The freely-available *LAPACK* library of Fortran codes represents the current state of the art for solving dense linear equations, least squares problems, and eigenvalue and singular value problems. Many modern programming packages and environments build on LAPACK.

It is interesting to note that the TOP500 list (`http://www.top500.org`) ranks the world's fastest computers by their speed (measured in flops per second) in solving a random linear system $Ax = b$ by GE. This benchmark has its origins in the 1970s *LINPACK* project, a precursor to LAPACK, in which the performance of contemporary machines was compared by running the LINPACK GE code on a 100×100 system.

14 Outlook

Matrix analysis and numerical linear algebra remain very active areas of research. Many problems in applied mathematics and scientific computing require the solution of a matrix problem at some stage, so there is always a demand for better understanding of matrix problems and faster and more accurate algorithms for their solution. As the overarching applications evolve, new problem variants are generated, often involving new assumptions on the data, different requirements on the solution, or new metrics for measuring the success of an algorithm. A further driver of research is computer hardware. With the advent of processors with many cores, the use of accelerators such as graphics processing units (GPUs), and the harnessing of vast numbers of processors for parallel computing, the

standard algorithms in numerical linear algebra are having to be reorganized and possibly even replaced, so we are likely to see significant changes in the coming years.

15 Further Reading

Three must-haves for researchers are Golub and Van Loan's influential treatment of numerical linear algebra and the two volumes by Horn and Johnson, which contain a comprehensive treatment of matrix analysis.

[1] Rajendra Bhatia. *Matrix Analysis*. Springer-Verlag, New York, 1997. xi+347 pp. ISBN 0-387-94846-5.

[2] Rajendra Bhatia. Linear algebra to quantum cohomology: The story of Alfred Horn's inequalities. *Amer. Math. Monthly*, 108(4):289–318, 2001.

[3] Rajendra Bhatia. *Positive Definite Matrices*. Princeton University Press, Princeton, NJ, USA, 2007. ix+254 pp. ISBN 0-691-12918-5.

[4] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Fourth edition, Johns Hopkins University Press, Baltimore, MD, USA, 2013. xxi+756 pp. ISBN 978-1-4214-0794-4.

[5] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. xxx+680 pp. ISBN 0-89871-521-0.

[6] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1991. viii+607 pp. ISBN 0-521-30587-X.

[7] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Second edition, Cambridge University Press, Cambridge, UK, 2013. xviii+643 pp. ISBN 978-0-521-83940-2.

[8] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998. xxiv+398 pp. Unabridged, amended version of book first published by Prentice-Hall in 1980. ISBN 0-89871-402-8.

[9] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003. xviii+528 pp. ISBN 0-89871-534-2.

[10] G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, London, 1990. xv+365 pp. ISBN 0-12-670230-6.

[11] Françoise Tisseur and Karl Meerbergen. The quadratic eigenvalue problem. *SIAM Rev.*, 43(2):235–286, 2001.