Chapter 9

Iteration

Iteration, meaning the repeated application of a process or function, appears in a surprisingly wide range of applications. Discrete dynamical systems, in which the time variable has been "quantized" into individual units (seconds, days, years, etc.) are modeled by iterative systems. Most numerical solution algorithms, for both linear and nonlinear systems, are based on iterative procedures. Starting with an initial guess, the successive iterates lead to closer and closer approximations to the true solution. For linear systems of equations, there are several iterative solution algorithms that can, in favorable situations, be employed as efficient alternatives to Gaussian Elimination. Iterative methods are particularly effective for solving the very large, sparse systems arising in the numerical solution of both ordinary and partial differential equations. All practical methods for computing eigenvalues and eigenvectors rely on some form of iteration. A detailed historical development of iterative methods for solving linear systems and eigenvalue problems can be found in the recent survey paper [**84**]. Probabilistic iterative models known as Markov chains govern basic stochastic processes and appear in genetics, population biology, scheduling, internet search, financial markets, and many more.

In this book, we will treat only iteration of linear systems. (Nonlinear iteration is of similar importance in applied mathematics and numerical analysis, and we refer the interested reader to [**40**, **66**, **79**] for details.) Linear iteration coincides with multiplication by successive powers of a matrix; convergence of the iterates depends on the magnitude of its eigenvalues. We present a variety of convergence criteria based on the spectral radius, on matrix norms, and on eigenvalue estimates provided by the Gershgorin Theorem.

We will then turn our attention to some classical iterative algorithms that can be used to accurately approximate the solutions to linear algebraic systems. The Jacobi Method is the simplest, while an evident serialization leads to the Gauss–Seidel Method. Completely general convergence criteria are hard to formulate, although convergence is assured for the important class of strictly diagonally dominant matrices that arise in many applications. A simple modification of the Gauss–Seidel Method, known as Successive Over-Relaxation (SOR), can dramatically speed up the convergence rate.

In the following Section 9.5 we discuss some practical methods for computing eigenvalues and eigenvectors of matrices. Needless to say, we completely avoid trying to solve (or even write down) the characteristic polynomial equation. The basic Power Method and its variants, which are based on linear iteration, are used to effectively approximate selected eigenvalues. To calculate the complete system of eigenvalues and eigenvectors, the remarkable $Q\,R$ algorithm, which relies on the Gram–Schmidt orthogonalization procedure, is the method of choice, and we include a new proof of its convergence.

The following section describes some more recent "semi-direct" iterative algorithms for finding eigenvalues and solving linear systems, that, in contrast to the classical iterative schemes, are guaranteed to eventually produce the exact solution. These are based on the idea of a Krylov subspace, spanned by the vectors generated by repeatedly multiplying an initial vector by the coefficient matrix. The Arnoldi and Lanczos algorithms are used to find a corresponding orthonormal basis for the Krylov subspaces, and thereby approximate (some of) the eigenvalues of the matrix. Two classes of solution methods are then

P. J. Olver, C. Shakiban, *Applied Linear Algebra*, Undergraduate Texts in Mathematics,
https://doi.org/10.1007/978-3-319-91041-3_9

presented: first, the Full Orthogonalization Method (FOM) which, for a positive definite matrix, produces the powerful technique known as Conjugate Gradients (CG), of particular importance in numerical approximation of partial differential equations. The second is the recent Generalized Minimal Residual Method (GMRES), which is effectively used for solving large sparse linear systems.

The final Section 9.7 introduces the basic ideas behind wavelets, a powerful and widely used alternative to Fourier methods for signal and image processing. While slightly off topic, it provides a nice application of orthogonality and iterative techniques, and is thus a fitting end to this chapter.

9.1 Linear Iterative Systems

We begin with the basic definition of an iterative system of linear equations.

Definition 9.1. A *linear iterative system* takes the form

$$\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}, \qquad \mathbf{u}^{(0)} = \mathbf{a}, \tag{9.1}$$

where the *coefficient matrix* T has size $n \times n$.

We will consider both real and complex systems, and so the *iterates*[†] $\mathbf{u}^{(k)}$ are vectors either in $\mathbb{R}^n$ (which assumes that the coefficient matrix T is also real) or in $\mathbb{C}^n$. A linear iterative system can be viewed as a discretized version of a first order system of linear ordinary differential equations, as in (8.9), in which the state of the system, as represented by the vector $\mathbf{u}^{(k)}$, changes at discrete time intervals, labeled by the index k.

Scalar Systems

As usual, to study systems one begins with an in-depth analysis of the scalar version. Consider the iterative equation

$$u^{(k+1)} = \lambda\,u^{(k)}, \qquad u^{(0)} = a, \tag{9.2}$$

where λ, a and the solution $u^{(k)}$ are all real or complex scalars. The general solution to (9.2) is easily found:

$$u^{(1)} = \lambda\,u^{(0)} = \lambda\,a, \qquad u^{(2)} = \lambda\,u^{(1)} = \lambda^2\,a, \qquad u^{(3)} = \lambda\,u^{(2)} = \lambda^3\,a,$$

and, in general,

$$u^{(k)} = \lambda^k\,a. \tag{9.3}$$

If the initial condition is $a = 0$, then the solution $u^{(k)} \equiv 0$ is constant. In other words, 0 is a *fixed point* or *equilibrium solution* for the iterative system because it does not change under iteration.

Example 9.2. Banks add interest to a savings account at discrete time intervals. For example, if the bank offers 5% interest compounded yearly, this means that the account balance will increase by 5% each year. Thus, assuming no deposits or withdrawals, the balance $u^{(k)}$ after k years will satisfy the iterative equation (9.2) with $\lambda = 1 + r$, where

[†] **Warning.** The superscripts on $\mathbf{u}^{(k)}$ refer to the iterate number, and should not be mistaken for derivatives.

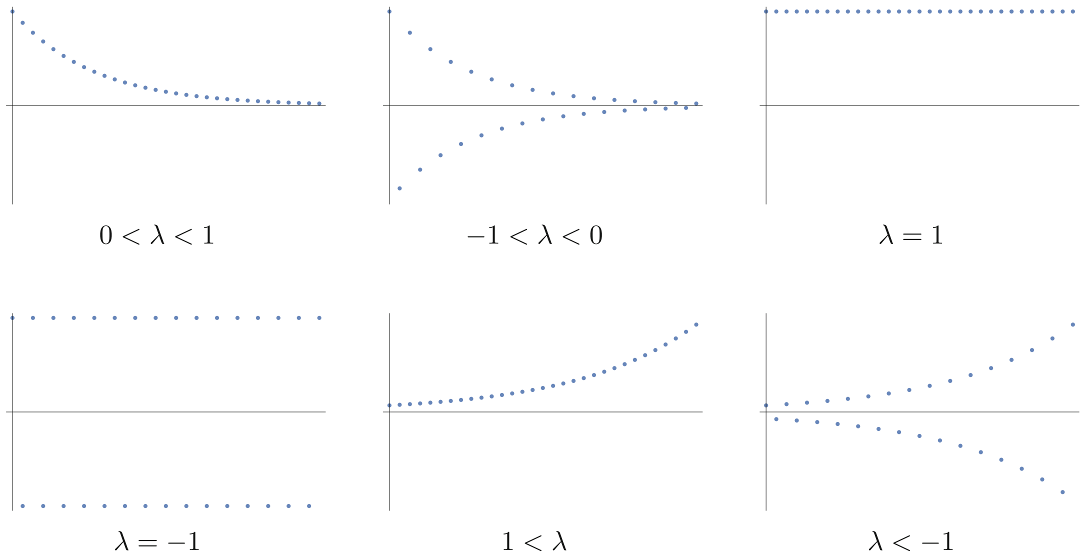


Figure 9.1. One–Dimensional Real Linear Iterative Systems.

$r = .05$ is the interest rate, and the 1 indicates that all the money remains in the account. Thus, after k years, your account balance is

$$u^{(k)} = (1+r)^k a, \qquad \text{where} \qquad a = u^{(0)} \tag{9.4}$$

is your initial deposit. For example, if $u^{(0)} = a = \$1{,}000$, after 1 year, your account has $u^{(1)} = \$1{,}050$, after 10 years $u^{(10)} = \$1{,}628.89$, after 50 years $u^{(50)} = \$11{,}467.40$, and after 200 years $u^{(200)} = \$17{,}292{,}580.82$, a gain of over 17,000%.

When the interest is compounded monthly, the rate is still quoted on a yearly basis, and so you receive $\frac{1}{12}$ of the interest each month. If $\widehat{u}^{(k)}$ denotes the balance after k months, then, after n years, the account balance will be $\widehat{u}^{(12n)} = \left(1 + \frac{1}{12} r\right)^{12n} a$. Thus, when the interest rate of 5% is compounded monthly, your account balance is $\widehat{u}^{(12)} = \$1{,}051.16$ after 1 year, $\widehat{u}^{(120)} = \$1,647.01$ after 10 years, $\widehat{u}^{(600)} = \$12{,}119.38$ after 50 years, and $\widehat{u}^{(2400)} = \$21{,}573{,}572.66$ dollars after 200 years. So, if you wait sufficiently long, compounding will have a dramatic effect. Similarly, daily compounding replaces 12 by 365.25, the number of days in a year. After 200 years, the balance wil be $22,011,396.03.

Let us analyze the solutions of scalar iterative equations, starting with the case when $\lambda \in \mathbb{R}$ is a real constant. Aside from the equilibrium solution $u^{(k)} \equiv 0$, the iterates exhibit six qualitatively different behaviors, depending on the size of the coefficient λ.

(*a*) If $\lambda = 0$, the solution immediately becomes zero, and stays there, whereby $u^{(k)} = 0$ for all $k \geq 1$.

(*b*) If $0 < \lambda < 1$, then the solution is of one sign, and tends monotonically to zero, so $u^{(k)} \to 0$ as $k \to \infty$.

(*c*) If $-1 < \lambda < 0$, then the solution tends to zero: $u^{(k)} \to 0$ as $k \to \infty$. Successive iterates have alternating signs.

(d) If $\lambda = 1$, the solution is constant: $u^{(k)} = a$, for all $k \geq 0$.

(e) If $\lambda = -1$, the solution bounces back and forth between two values; $u^{(k)} = (-1)^k a$.

(f) If $1 < \lambda < \infty$, then the iterates $u^{(k)}$ become unbounded. If $a > 0$, they tend monotonically to $+\infty$; if $a < 0$, they tend to $-\infty$.

(g) If $-\infty < \lambda < -1$, then the iterates $u^{(k)}$ also become unbounded, with alternating signs.

In Figure 9.1 we exhibit representative *scatter plots* for the nontrivial cases (b – g). The horizontal axis indicates the index k, and the vertical axis the solution value u. Each dot in the scatter plot represents an iterate $u^{(k)}$.

In the first three cases, the fixed point $u = 0$ is said to be *asymptotically stable*, since all solutions tend to 0 as $k \to \infty$. In cases (d) and (e), the zero solution is *stable*, since solutions with nearby initial data, $|\, a \,| \ll 1$, remain nearby. In the final two cases, the zero solution is *unstable*; every nonzero initial data $a \neq 0$ — no matter how small — will give rise to a solution that eventually goes arbitrarily far away from equilibrium.

Let us also investigate complex scalar iterative systems. The coefficient λ and the initial datum a in (9.2) are allowed to be complex numbers. The solution is the same, (9.3), but now we need to know what happens when we raise a complex number λ to a high power. The secret is to write $\lambda = r\, e^{\,\mathrm{i}\,\theta}$ in polar form (3.93), where $r = |\,\lambda\,|$ is its modulus and $\theta = \operatorname{ph} \lambda$ its angle or phase. Then $\lambda^k = r^k e^{\,\mathrm{i}\,k\,\theta}$. Since $|\, e^{\,\mathrm{i}\,k\,\theta} \,| = 1$, we have $|\,\lambda^k\,| = |\,\lambda\,|^k$, and so the solutions (9.3) have modulus $|\,u^{(k)}\,| = |\,\lambda^k\, a\,| = |\,\lambda\,|^k\,|\,a\,|$. As a result, $u^{(k)}$ will remain bounded if and only if $|\,\lambda\,| \leq 1$, and will tend to zero as $k \to \infty$ if and only if $|\,\lambda\,| < 1$.

We have thus established the basic stability criteria for scalar, linear systems.

Theorem 9.3. The zero solution to a (real or complex) scalar iterative system is

(a) *asymptotically stable* if and only if $|\,\lambda\,| < 1$,

(b) *stable* if and only if $|\,\lambda\,| \leq 1$,

(c) *unstable* if and only if $|\,\lambda\,| > 1$.

Exercises

9.1.1. Suppose $u^{(0)} = 1$. Find $u^{(1)}, u^{(10)}$, and $u^{(20)}$ when (a) $u^{(k+1)} = 2\,u^{(k)}$, (b) $u^{(k+1)} = -.9\,u^{(k)}$, (c) $u^{(k+1)} = \mathrm{i}\,u^{(k)}$, ($d$) $u^{(k+1)} = (1 - 2\,\mathrm{i}\,)\,u^{(k)}$. Is the system stable or unstable? If stable, is it asymptotically stable?

9.1.2. A bank offers 3.25% interest compounded yearly. Suppose you deposit \$100. (a) Set up a linear iterative equation to represent your bank balance. (b) How much money do you have after 10 years? (c) What if the interest is compounded monthly?

9.1.3. Show that the yearly balances of an account whose interest is compounded monthly satisfy a linear iterative system. How is the effective yearly interest rate determined from the original annual interest rate?

9.1.4. Show that, as the time interval of compounding goes to zero, the bank balance after k years approaches an exponential function $e^{r\,k}\, a$, where r is the yearly interest rate and a is the initial balance.

9.1.5. For which values of λ does the scalar iterative system (9.2) have a periodic solution, meaning that $u^{(k+m)} = u^{(k)}$ for some m?

9.1.6. Consider the iterative systems $u^{(k+1)} = \lambda\, u^{(k)}$ and $v^{(k+1)} = \mu\, v^{(k)}$, where $|\,\lambda\,| > |\,\mu\,|$. Prove that, for all nonzero initial data $u^{(0)} = a \neq 0$, $v^{(0)} = b \neq 0$, the solution to the first is eventually larger (in modulus) than that of the second: $|\,u^{(k)}\,| > |\,v^{(k)}\,|$, for $k \gg 0$.

9.1.7. Let $u(t)$ denote the solution to the linear ordinary differential equation $\dot{u} = \beta\, u$, $u(0) = a$. Let $h > 0$. Show that the sample values $u^{(k)} = u(k\, h)$ satisfy a linear iterative system. What is the coefficient λ? Compare the stability properties of the differential equation and the corresponding iterative system.

♠ 9.1.8. Investigate the solutions of the linear iterative equation $u^{(k+1)} = \lambda\, u^{(k)}$ when λ is a complex number with $|\,\lambda\,| = 1$, and look for patterns.

9.1.9. Let $\lambda, c \in \mathbb{R}$. Solve the *affine* or *inhomogeneous linear iterative equation*

$$u^{(k+1)} = \lambda\, u^{(k)} + c, \qquad\qquad u^{(0)} = a. \tag{9.5}$$

Discuss the possible behaviors of the solutions. *Hint*: Write the solution in the form $u^{(k)} = u^\star + v^{(k)}$, where $u^\star$ is the equilibrium solution.

9.1.10. A bank offers 5% interest compounded yearly. Suppose you deposit \$120 in the account each year. Set up an affine iterative equation (9.5) to represent your bank balance. How much money do you have after 10 years? After you retire in 50 years? After 200 years?

9.1.11. Redo Exercise 9.1.10 in the case that the interest is compounded monthly and you deposit \$10 each month.

♡ 9.1.12. Each spring, the deer in Minnesota produce offspring at a rate of roughly 1.2 times the total population, while approximately 5% of the population dies as a result of predators and natural causes. In the fall, hunters are allowed to shoot 3,600 deer. This winter the Department of Natural Resources (DNR) estimates that there are 20,000 deer. Set up an affine iterative equation (9.5) to represent the deer population each subsequent year. Solve the system and find the population in the next 5 years. How many deer in the long term will there be? Using this information, formulate a reasonable policy of how many deer hunting licenses the DNR should allow each fall, assuming one kill per license.

Powers of Matrices

The solution to the general linear iterative system

$$\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}, \qquad\qquad \mathbf{u}^{(0)} = \mathbf{a}, \tag{9.6}$$

is also, at least at first glance, immediate. Clearly,

$$\mathbf{u}^{(1)} = T\,\mathbf{u}^{(0)} = T\,\mathbf{a}, \qquad \mathbf{u}^{(2)} = T\,\mathbf{u}^{(1)} = T^2\mathbf{a}, \qquad \mathbf{u}^{(3)} = T\,\mathbf{u}^{(2)} = T^3\mathbf{a},$$

and, in general,

$$\mathbf{u}^{(k)} = T^k\mathbf{a}. \tag{9.7}$$

Thus, the iterates are simply determined by multiplying the initial vector $\mathbf{a}$ by the successive powers of the coefficient matrix T. And so, in contrast to differential equations, proving the existence and uniqueness of solutions to an iterative system is completely trivial.

However, unlike real or complex scalars, the general formulas for and qualitative behavior of the powers of a square matrix are not nearly so immediately apparent. (Before continuing, the reader is urged to experiment with simple 2×2 matrices, trying to detect patterns.) To make progress, recall how, in Section 8.1, we endeavored to solve linear systems of differential equations by suitably adapting the known exponential solution from the scalar version. In the iterative case, the scalar solution formula (9.3) is written in terms of powers, not exponentials. This motivates us to try the power ansatz

$$\mathbf{u}^{(k)} = \lambda^k\,\mathbf{v}, \tag{9.8}$$

in which λ is a scalar and $\mathbf{v}$ a vector, as a possible solution to the system. We find

$$\mathbf{u}^{(k+1)} = \lambda^{k+1}\,\mathbf{v}, \qquad \text{while} \qquad T\,\mathbf{u}^{(k)} = T(\lambda^k\,\mathbf{v}) = \lambda^k\,T\,\mathbf{v}.$$

These two expressions will be equal if and only if

$$T\,\mathbf{v} = \lambda\,\mathbf{v}.$$

This is precisely the defining eigenvalue equation (8.12), and thus, (9.8) is a nontrivial solution to (9.6) if and only if λ is an *eigenvalue* of the coefficient matrix T and $\mathbf{v} \neq \mathbf{0}$ an associated *eigenvector*.

Thus, for each eigenvector and eigenvalue of the coefficient matrix, we can construct a solution to the iterative system. We can then appeal to linear superposition, as in Theorem 7.30, to combine the basic eigensolutions to form more general solutions. In particular, if the coefficient matrix is complete, this method will produce the general solution.

Theorem 9.4. If the coefficient matrix T is complete, then the general solution to the linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ is given by

$$\mathbf{u}^{(k)} = c_1\,\lambda_1^k\,\mathbf{v}_1 + c_2\,\lambda_2^k\,\mathbf{v}_2 + \;\cdots\; + c_n\,\lambda_n^k\,\mathbf{v}_n, \tag{9.9}$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are the linearly independent eigenvectors and $\lambda_1, \ldots, \lambda_n$ the corresponding eigenvalues of T. The coefficients $c_1, \ldots, c_n$ are arbitrary scalars and are uniquely prescribed by the initial conditions $\mathbf{u}^{(0)} = \mathbf{a}$.

Proof: Since we already know, by linear superposition, that (9.9) is a solution to the system for arbitrary $c_1, \ldots, c_n$, it suffices to show that we can match any prescribed initial conditions. To this end, we need to solve the linear system

$$\mathbf{u}^{(0)} = c_1\,\mathbf{v}_1 + \;\cdots\; + c_n\,\mathbf{v}_n = \mathbf{a}. \tag{9.10}$$

Completeness of T implies that its eigenvectors form a basis of $\mathbb{C}^n$, and hence (9.10) always admits a solution. In matrix form, we can rewrite (9.10) as

$$S\,\mathbf{c} = \mathbf{a}, \qquad \text{so that} \qquad \mathbf{c} = S^{-1}\mathbf{a}, \qquad \text{where} \qquad S = (\,\mathbf{v}_1\;\mathbf{v}_2\;\ldots\;\mathbf{v}_n\,)$$

is the (nonsingular) matrix whose columns are the eigenvectors. *Q.E.D.*

Solutions in the incomplete cases are more complicated to write down, and rely on the Jordan bases of Section 8.6; see Exercise 9.1.40.

Example 9.5. Consider the iterative system

$$x^{(k+1)} = \tfrac{3}{5}\,x^{(k)} + \tfrac{1}{5}\,y^{(k)}, \qquad y^{(k+1)} = \tfrac{1}{5}\,x^{(k)} + \tfrac{3}{5}\,y^{(k)}, \tag{9.11}$$

with initial conditions

$$x^{(0)} = a, \qquad y^{(0)} = b. \tag{9.12}$$

The system can be rewritten in our matrix form (9.6), with

$$T = \begin{pmatrix} .6 & .2 \\ .2 & .6 \end{pmatrix}, \qquad \mathbf{u}^{(k)} = \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}, \qquad \mathbf{a} = \begin{pmatrix} a \\ b \end{pmatrix}.$$

Solving the characteristic equation

$$\det(T - \lambda\,\mathrm{I}\,) = \lambda^2 - 1.2\,\lambda - .32 = 0$$

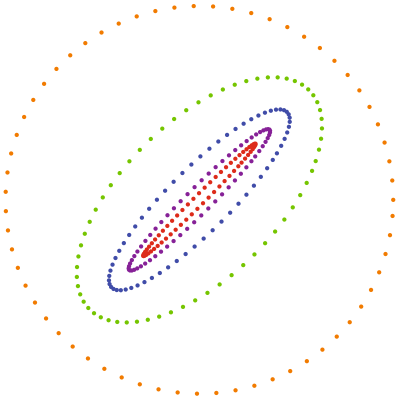


Figure 9.2. Stable Iterative System.

produces the eigenvalues $\lambda_1 = .8$, $\lambda_2 = .4$. We then solve the associated linear systems $(T - \lambda_j \mathrm{I})\mathbf{v}_j = \mathbf{0}$ for the corresponding eigenvectors:

$$\lambda_1 = .8\,, \qquad \mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad\qquad \lambda_2 = .4\,, \qquad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Therefore, the basic eigensolutions are

$$\mathbf{u}_1^{(k)} = (\,.8)^k \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad \mathbf{u}_2^{(k)} = (\,.4)^k \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

Theorem 9.4 tells us that the general solution is given as a linear combination,

$$\mathbf{u}^{(k)} = c_1\,\mathbf{u}_1^{(k)} + c_2\,\mathbf{u}_2^{(k)} = c_1\,(\,.8)^k \begin{pmatrix} 1 \\ 1 \end{pmatrix} + c_2\,(\,.4)^k \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} c_1\,(\,.8)^k - c_2\,(\,.4)^k \\ c_1\,(\,.8)^k + c_2\,(\,.4)^k \end{pmatrix},$$

where c_1, c_2 are determined by the initial conditions:

$$\mathbf{u}^{(0)} = \begin{pmatrix} c_1 - c_2 \\ c_1 + c_2 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \qquad \text{and hence} \qquad c_1 = \frac{a+b}{2}\,, \qquad c_2 = \frac{b-a}{2}\,.$$

Therefore, the explicit formula for the solution to the initial value problem (9.11–12) is

$$x^{(k)} = (\,.8)^k\,\frac{a+b}{2} + (\,.4)^k\,\frac{a-b}{2}, \qquad y^{(k)} = (\,.8)^k\,\frac{a+b}{2} + (\,.4)^k\,\frac{b-a}{2}.$$

In particular, as $k \to \infty$, the iterates $\mathbf{u}^{(k)} \to \mathbf{0}$ converge to zero at a rate governed by the dominant eigenvalue $\lambda_1 = .8$. Figure 9.2 illustrates the cumulative effect of the iteration; the initial data is colored orange, and successive iterates are colored green, blue, purple, red. The initial conditions consist of a large number of points on the unit circle $x^2 + y^2 = 1$, which are successively mapped to points on progressively smaller and flatter ellipses, that shrink down towards the origin.

Example 9.6. The *Fibonacci numbers* are defined by the second order[†] scalar iterative equation

$$u^{(k+2)} = u^{(k+1)} + u^{(k)}, \tag{9.13}$$

[†] In general, an iterative system $\mathbf{u}^{(k+j)} = T_1\mathbf{u}^{(k+j-1)} + \cdots + T_j\mathbf{u}^{(k)}$ in which the new iterate depends upon the preceding j values is said to have *order* j.

with initial conditions

$$u^{(0)} = a, \qquad u^{(1)} = b. \tag{9.14}$$

In short, to obtain the next Fibonacci number, add the previous two. The classical *Fibonacci integers* start with $a = 0$, $b = 1$; the next few are

$$u^{(0)} = 0,\ \ u^{(1)} = 1,\ \ u^{(2)} = 1,\ \ u^{(3)} = 2,\ \ u^{(4)} = 3,\ \ u^{(5)} = 5,\ \ u^{(6)} = 8,\ \ u^{(7)} = 13,\ \ \ldots .$$

The Fibonacci integers occur in a surprising variety of natural objects, including leaves, flowers, and fruit, [**83**]. They were originally introduced by the eleventh-/twelfth-century Italian mathematician Leonardo Pisano Fibonacci as a crude model of the growth of a population of rabbits. In Fibonacci's model, the k^{th} Fibonacci number $u^{(k)}$ measures the total number of pairs of rabbits at year k. We start the process with a single juvenile pair[‡] at year 0. Once a year, each pair of rabbits produces a new pair of offspring, but it takes a full year for a rabbit pair to mature enough to produce offspring of their own.

Every higher order iterative equation can be replaced by an equivalent first order iterative system. In this particular case, we define the vector

$$\mathbf{u}^{(k)} = \begin{pmatrix} u^{(k)} \\ u^{(k+1)} \end{pmatrix} \in \mathbb{R}^2,$$

and note that (9.13) is equivalent to the matrix system

$$\begin{pmatrix} u^{(k+1)} \\ u^{(k+2)} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u^{(k)} \\ u^{(k+1)} \end{pmatrix}, \quad \text{or} \quad \mathbf{u}^{(k+1)} = T\mathbf{u}^{(k)}, \quad \text{where} \quad T = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

To find the explicit formula for the Fibonacci numbers, we must determine the eigenvalues and eigenvectors of the coefficient matrix T. A straightforward computation produces

$$\lambda_1 = \frac{1+\sqrt{5}}{2} = 1.618034\ldots, \qquad \lambda_2 = \frac{1-\sqrt{5}}{2} = -.618034\ldots,$$
$$\mathbf{v}_1 = \begin{pmatrix} \frac{-1+\sqrt{5}}{2} \\ 1 \end{pmatrix}, \qquad \mathbf{v}_2 = \begin{pmatrix} \frac{-1-\sqrt{5}}{2} \\ 1 \end{pmatrix}.$$

Therefore, according to (9.9), the general solution to the Fibonacci system is

$$\mathbf{u}^{(k)} = \begin{pmatrix} u^{(k)} \\ u^{(k+1)} \end{pmatrix} = c_1 \left(\frac{1+\sqrt{5}}{2} \right)^k \begin{pmatrix} \frac{-1+\sqrt{5}}{2} \\ 1 \end{pmatrix} + c_2 \left(\frac{1-\sqrt{5}}{2} \right)^k \begin{pmatrix} \frac{-1-\sqrt{5}}{2} \\ 1 \end{pmatrix}. \tag{9.15}$$

The initial data

$$\mathbf{u}^{(0)} = c_1 \begin{pmatrix} \frac{-1+\sqrt{5}}{2} \\ 1 \end{pmatrix} + c_2 \begin{pmatrix} \frac{-1-\sqrt{5}}{2} \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

uniquely specifies the coefficients

$$c_1 = \frac{2a + (1+\sqrt{5})b}{2\sqrt{5}}, \qquad c_2 = -\,\frac{2a + (1-\sqrt{5})b}{2\sqrt{5}}.$$

The first entry of the solution vector (9.15) produces the explicit formula

$$u^{(k)} = \frac{(-1+\sqrt{5})a + 2b}{2\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^k + \frac{(1+\sqrt{5})a - 2b}{2\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^k \tag{9.16}$$

[‡] Fibonacci ignores some pertinent details like the sex of the offspring.

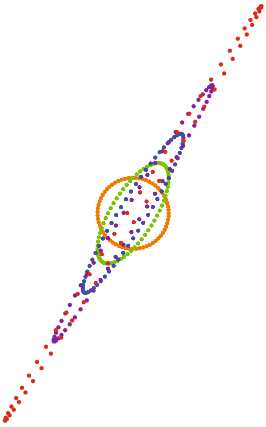


Figure 9.3. Fibonacci Iteration.

for the k^{th} Fibonacci number. For the particular initial conditions $a = 0$, $b = 1$, (9.16) reduces to the classical *Binet formula*

$$u^{(k)} = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^k - \left(\frac{1-\sqrt{5}}{2} \right)^k \right]. \tag{9.17}$$

It is a remarkable fact that, for every value of k, all the $\sqrt{5}$'s cancel out, and the Binet formula (9.17) does indeed produce the Fibonacci integers listed above. Another useful observation is that, since

$$0 < |\,\lambda_2\,| = \frac{\sqrt{5}-1}{2} < 1 < \lambda_1 = \frac{1+\sqrt{5}}{2},$$

the terms involving λ_1^k go to ∞ (and so the zero solution to this iterative system is unstable) while the terms involving λ_2^k go to zero. Therefore, even for k moderately large, the first term in (9.16) is an excellent approximation to the k^{th} Fibonacci number — and one that gets more and more accurate as k increases. A plot of the first 4 iterates, starting with the initial data consisting of equally spaced points on the unit circle, appears in Figure 9.3. As in the previous example, the circle is mapped to a sequence of progressively more eccentric ellipses; however, their major semi-axes become more and more stretched out, and almost all points end up going off to ∞ in the direction of the dominant eigenvector $\mathbf{v}_2$.

The dominant eigenvalue $\lambda_1 = \frac{1}{2}\left(1+\sqrt{5}\,\right) = 1.6180339\ldots$ is known as the *golden ratio* and plays an important role in spiral growth in nature, as well as in art, architecture, and design, [**83**]. It describes the overall growth rate of the Fibonacci integers, and, in fact, every sequence of Fibonacci numbers with initial conditions $b \neq \frac{1}{2}\left(1-\sqrt{5}\,\right) a$.

Example 9.7. Let $T = \begin{pmatrix} -3 & 1 & 6 \\ 1 & -1 & -2 \\ -1 & -1 & 0 \end{pmatrix}$ be the coefficient matrix for a three-dimensional iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$. Its eigenvalues and corresponding eigenvectors are

$$\lambda_1 = -2, \qquad \lambda_2 = -1+\mathrm{i}\,, \qquad \lambda_3 = -1-\mathrm{i}\,,$$
$$\mathbf{v}_1 = \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix}, \qquad \mathbf{v}_2 = \begin{pmatrix} 2-\mathrm{i} \\ -1 \\ 1 \end{pmatrix}, \qquad \mathbf{v}_3 = \begin{pmatrix} 2+\mathrm{i} \\ -1 \\ 1 \end{pmatrix}.$$

Therefore, according to (9.9), the general complex solution is

$$\mathbf{u}^{(k)} = b_1\,(-2)^k \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix} + b_2\,(-1+\mathrm{i}\,)^k \begin{pmatrix} 2-\mathrm{i} \\ -1 \\ 1 \end{pmatrix} + b_3\,(-1-\mathrm{i}\,)^k \begin{pmatrix} 2+\mathrm{i} \\ -1 \\ 1 \end{pmatrix},$$

where b_1, b_2, b_3 are arbitrary complex scalars.

If we are interested only in real solutions, we can break up any complex solution into its real and imaginary parts, each of which constitutes a real solution. (This is a manifestation of the general Reality Principle of Theorem 7.48, but is not hard to prove directly.) We begin by writing $\lambda_2 = -1+\mathrm{i} = \sqrt{2}\,e^{3\pi\,\mathrm{i}\,/4}$ in polar form, and hence

$$(-1+\mathrm{i}\,)^k = 2^{k/2}\,e^{3\,k\,\pi\,\mathrm{i}\,/4} = 2^{k/2}\left(\cos\tfrac{3}{4}\,k\pi + \mathrm{i}\,\sin\tfrac{3}{4}\,k\pi\right).$$

Therefore, the complex solution

$$(-1+\mathrm{i}\,)^k \begin{pmatrix} 2-\mathrm{i} \\ -1 \\ 1 \end{pmatrix} = 2^{k/2} \begin{pmatrix} 2\cos\frac{3}{4}\,k\pi + \sin\frac{3}{4}\,k\pi \\ -\cos\frac{3}{4}\,k\pi \\ \cos\frac{3}{4}\,k\pi \end{pmatrix} + \mathrm{i}\;2^{k/2} \begin{pmatrix} 2\sin\frac{3}{4}\,k\pi - \cos\frac{3}{4}\,k\pi \\ -\sin\frac{3}{4}\,k\pi \\ \sin\frac{3}{4}\,k\pi \end{pmatrix}$$

is a combination of two independent real solutions. The complex conjugate eigenvalue $\lambda_3 = -1-\mathrm{i}$ leads, as before, to the complex conjugate solution — and the same two real solutions. The general real solution $\mathbf{u}^{(k)}$ to the system can be written as a linear combination of the three independent real solutions:

$$c_1\,(-2)^k \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix} + c_2\,2^{k/2} \begin{pmatrix} 2\cos\frac{3}{4}\,k\pi+\sin\frac{3}{4}\,k\pi \\ -\cos\frac{3}{4}\,k\pi \\ \cos\frac{3}{4}\,k\pi \end{pmatrix} + c_3\,2^{k/2} \begin{pmatrix} 2\sin\frac{3}{4}\,k\pi-\cos\frac{3}{4}\,k\pi \\ -\sin\frac{3}{4}\,k\pi \\ \sin\frac{3}{4}\,k\pi \end{pmatrix}, \tag{9.18}$$

where c_1, c_2, c_3 are arbitrary real scalars, uniquely prescribed by the initial conditions.

Diagonalization and Iteration

An alternative, equally efficient approach to solving iterative systems is based on diagonalization of the coefficient matrix, cf. (8.30). Specifically, assuming the coefficient matrix T is complete, we can factor it as a product

$$T = S\,\Lambda\,S^{-1}, \tag{9.19}$$

in which $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix containing the eigenvalues of T, while the columns of $S = (\mathbf{v}_1 \ \cdots\ \mathbf{v}_n)$ are the corresponding eigenvectors. Consequently, the powers of T are given by

$$\begin{aligned} T^2 &= (S\,\Lambda\,S^{-1})\,(S\,\Lambda\,S^{-1}) = S\,\Lambda^2\,S^{-1}, \\ T^3 &= (S\,\Lambda\,S^{-1})\,(S\,\Lambda\,S^{-1})\,(S\,\Lambda\,S^{-1}) = S\,\Lambda^3\,S^{-1}, \end{aligned}$$

and, in general,

$$T^k = S\,\Lambda^k\,S^{-1}. \tag{9.20}$$

Moreover, since Λ is a diagonal matrix, its powers are trivial to compute:

$$\Lambda^k = \operatorname{diag}(\lambda_1^k, \lambda_2^k,\ \ldots\ ,\lambda_n^k). \tag{9.21}$$

Thus, by combining (9.20–21), we obtain an explicit formula for the powers of a complete matrix T. Furthermore, the solution to the associated linear iterative system

$$\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}, \qquad \mathbf{u}^{(0)} = \mathbf{a}, \qquad \text{is given by} \qquad \mathbf{u}^{(k)} = T^k\mathbf{a} = S\,\Lambda^k\,S^{-1}\mathbf{a}. \tag{9.22}$$

You should convince yourself that this gives precisely the same solution as before. Computationally, there is not a significant difference between the two solution methods, and the choice is left to the discretion of the user.

Example 9.8. Suppose $T = \begin{pmatrix} 7 & 6 \\ -9 & -8 \end{pmatrix}$. Its eigenvalues and eigenvectors are readily computed:

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

We assemble these into the diagonal eigenvalue matrix Λ and the eigenvector matrix S, given by

$$\Lambda = \begin{pmatrix} -2 & 0 \\ 0 & 1 \end{pmatrix}, \qquad S = \begin{pmatrix} -2 & -1 \\ 3 & 1 \end{pmatrix},$$

whence

$$\begin{pmatrix} 7 & 6 \\ -9 & -8 \end{pmatrix} = T = S\,\Lambda\,S^{-1} = \begin{pmatrix} -2 & -1 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} -2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -3 & -2 \end{pmatrix},$$

as you can readily check. Therefore, according to (9.20),

$$\begin{aligned} T^k &= S\,\Lambda^k\,S^{-1} \\ &= \begin{pmatrix} -2 & -1 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} (-2)^k & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -3 & -2 \end{pmatrix} = \begin{pmatrix} 3 - 2\,(-2)^k & 2 - 2\,(-2)^k \\ -3 + 3\,(-2)^k & -2 + 3\,(-2)^k \end{pmatrix}. \end{aligned}$$

You may wish to check this formula directly for the first few values of $k = 1, 2, \ldots$. As a result, the solution to the particular iterative system

$$\mathbf{u}^{(k+1)} = \begin{pmatrix} 7 & 6 \\ -9 & -8 \end{pmatrix} \mathbf{u}^{(k)}, \quad \mathbf{u}^{(0)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad \text{is} \qquad \mathbf{u}^{(k)} = T^k \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 - 4\,(-2)^k \\ -5 + 6\,(-2)^k \end{pmatrix}.$$

In this case, the eigenvalue $\lambda_1 = -2$ causes an instability, with solutions having arbitrarily large norm as $k \to \infty$.

Exercises

9.1.13. Find the explicit formula for the solution to the following linear iterative systems:

(a) $u^{(k+1)} = u^{(k)} - 2\,v^{(k)}, \;\; v^{(k+1)} = -2\,u^{(k)} + v^{(k)}, \;\; u^{(0)} = 1, \;\; v^{(0)} = 0.$

(b) $u^{(k+1)} = u^{(k)} - \frac{2}{3}\,v^{(k)}, \;\; v^{(k+1)} = \frac{1}{2}\,u^{(k)} - \frac{1}{6}\,v^{(k)}, \;\; u^{(0)} = -2, \;\; v^{(0)} = 3.$

(c) $u^{(k+1)} = u^{(k)} - v^{(k)}, \;\; v^{(k+1)} = -u^{(k)} + 5\,v^{(k)}, \;\; u^{(0)} = 1, \;\; v^{(0)} = 0.$

(d) $u^{(k+1)} = \frac{1}{2}\,u^{(k)} + v^{(k)}, \;\; v^{(k+1)} = v^{(k)} - 2\,w^{(k)}, \;\; w^{(k+1)} = \frac{1}{3}\,w^{(k)},$
$u^{(0)} = 1, \;\; v^{(0)} = -1, \;\; w^{(0)} = 1.$

(e) $u^{(k+1)} = -u^{(k)} + 2\,v^{(k)} - w^{(k)}, \;\; v^{(k+1)} = -6\,u^{(k)} + 7\,v^{(k)} - 4\,w^{(k)},$
$w^{(k+1)} = -6\,u^{(k)} + 6\,v^{(k)} - 4\,w^{(k)}, \quad u^{(0)} = 0, \quad v^{(0)} = 1, \quad w^{(0)} = 3.$

9.1.14. Find the explicit formula for the general solution to the linear iterative systems with the following coefficient matrices:

(a) $\begin{pmatrix} -1 & 2 \\ 1 & -1 \end{pmatrix}$, (b) $\begin{pmatrix} -2 & 7 \\ -1 & 3 \end{pmatrix}$, (c) $\begin{pmatrix} -3 & 2 & -2 \\ -6 & 4 & -3 \\ 12 & -6 & -5 \end{pmatrix}$, (d) $\begin{pmatrix} -\frac{5}{6} & \frac{1}{3} & -\frac{1}{6} \\ 0 & -\frac{1}{2} & \frac{1}{3} \\ 1 & -1 & \frac{2}{3} \end{pmatrix}$.

9.1.15. Prove that all the Fibonacci integers $u^{(k)}$, $k \geq 0$, can be found by just computing the first term in the Binet formula (9.17) and then rounding off to the nearest integer.

9.1.16. The k^{th} *Lucas number* is defined as $L^{(k)} = \left(\dfrac{1+\sqrt{5}}{2} \right)^k + \left(\dfrac{1-\sqrt{5}}{2} \right)^k$.

(*a*) Explain why the Lucas numbers satisfy the Fibonacci iterative equation $L^{(k+2)} = L^{(k+1)} + L^{(k)}$. (*b*) Write down the first 7 Lucas numbers.
(*c*) Prove that every Lucas number is a positive integer.

9.1.17. What happens to the Fibonacci integers $u^{(k)}$ if we go "backward in time", i.e., for $k < 0$? How is $u^{(-k)}$ related to $u^{(k)}$?

9.1.18. Use formula (9.20) to compute the k^{th} power of the following matrices:

$$(a)\ \begin{pmatrix} 5 & 2 \\ 2 & 2 \end{pmatrix}, \quad (b)\ \begin{pmatrix} 4 & 1 \\ -2 & 1 \end{pmatrix}, \quad (c)\ \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \quad (d)\ \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 1 \end{pmatrix}, \quad (e)\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 2 \end{pmatrix}.$$

9.1.19. Use your answer from Exercise 9.1.18 to solve the following iterative systems:
(*a*) $u^{(k+1)} = 5u^{(k)} + 2v^{(k)},\ v^{(k+1)} = 2u^{(k)} + 2v^{(k)},\ u^{(0)} = -1,\ v^{(0)} = 0,$
(*b*) $u^{(k+1)} = 4u^{(k)} + v^{(k)},\ v^{(k+1)} = -2u^{(k)} + v^{(k)},\ u^{(0)} = 1,\ v^{(0)} = -3,$
(*c*) $u^{(k+1)} = u^{(k)} + v^{(k)},\ v^{(k+1)} = -u^{(k)} + v^{(k)},\ u^{(0)} = 0,\ v^{(0)} = 2,$
(*d*) $u^{(k+1)} = u^{(k)} + v^{(k)} + 2w^{(k)},\ v^{(k+1)} = u^{(k)} + 2v^{(k)} + w^{(k)},$
$w^{(k+1)} = 2u^{(k)} + v^{(k)} + w^{(k)},\ u^{(0)} = 1,\ v^{(0)} = 0,\ w^{(0)} = 1,$
(*e*) $u^{(k+1)} = v^{(k)},\ v^{(k+1)} = w^{(k)},\ w^{(k+1)} = -u^{(k)} + 2w^{(k)},\ u^{(0)} = 1,\ v^{(0)} = 0,\ w^{(0)} = 0.$

9.1.20. (*a*) Given initial data $\mathbf{u}^{(0)} = (1,1,1)^T$, explain why the resulting solution $\mathbf{u}^{(k)}$ to the system in Example 9.7 has all integer entries. (*b*) Find the coefficients c_1, c_2, c_3 in the explicit solution formula (9.18). (*c*) Check the first few iterates to convince yourself that the solution formula does, in spite of appearances, always give an integer value.

9.1.21. (*a*) Show how to convert the higher order linear iterative equation

$$u^{(k+j)} = c_1 u^{(k+j-1)} + c_2 u^{(k+j-2)} + \cdots + c_j u^{(k)}$$

into a first order system $\mathbf{u}^{(k)} = T\mathbf{u}^{(k)}$. *Hint*: See Example 9.6.
(*b*) Write down initial conditions that guarantee a unique solution $u^{(k)}$ for all $k \geq 0$.

9.1.22. Apply the method of Exercise 9.1.21 to solve the following iterative equations:
(*a*) $u^{(k+2)} = -u^{(k+1)} + 2u^{(k)},\quad u^{(0)} = 1,\quad u^{(1)} = 2.$
(*b*) $12u^{(k+2)} = u^{(k+1)} + u^{(k)},\quad u^{(0)} = -1,\quad u^{(1)} = 2.$
(*c*) $u^{(k+2)} = 4u^{(k+1)} + u^{(k)},\quad u^{(0)} = 1,\quad u^{(1)} = -1.$
(*d*) $u^{(k+2)} = 2u^{(k+1)} - 2u^{(k)},\quad u^{(0)} = 1,\quad u^{(1)} = 3.$
(*e*) $u^{(k+3)} = 2u^{(k+2)} + u^{(k+1)} - 2u^{(k)},\quad u^{(0)} = 0,\quad u^{(1)} = 2,\quad u^{(2)} = 3.$
(*f*) $u^{(k+3)} = u^{(k+2)} + 2u^{(k+1)} - 2u^{(k)},\quad u^{(0)} = 0,\quad u^{(1)} = 1,\quad u^{(2)} = 1.$

9.1.23. Suppose you have n dollars and can buy coffee for \$1, milk for \$2, and orange juice for \$2. Let $C^{(n)}$ count the number of different ways of spending all your money. (*a*) Explain why $C^{(n)} = C^{(n-1)} + 2C^{(n-2)}$, $C^{(0)} = C^{(1)} = 1$. (*b*) Find an explicit formula for $C^{(n)}$.

9.1.24. Find the general solution to the iterative system $u_i^{(k+1)} = u_{i-1}^{(k)} + u_{i+1}^{(k)}$, $i = 1, \ldots, n$, where we set $u_0^{(k)} = u_{n+1}^{(k)} = 0$ for all k. *Hint*: Use Exercise 8.2.47.

♣ 9.1.25. Starting with $u^{(0)} = 0$, $u^{(1)} = 0$, $u^{(2)} = 1$, define the sequence of *tribonacci numbers* $u^{(k)}$ by adding the previous three to get the next one. For instance, $u^{(3)} = u^{(0)} + u^{(1)} + u^{(2)} = 1$. (*a*) Write out the next four tribonacci numbers. (*b*) Find a third order iterative equation for the k^{th} tribonacci number. (*c*) Explain why the tribonacci numbers are all integers. (*d*) Find an explicit formula for the solution, using a computer to approximate the eigenvalues. (*e*) Do they grow faster than the usual Fibonacci numbers? What is their overall rate of growth?

♣ 9.1.26. Suppose that Fibonacci's rabbits live for only eight years, [**44**]. (*a*) Write out an iterative equation to describe the rabbit population. (*b*) Write down the first few terms. (*c*) Convert your equation into a first order iterative system, using the method of Exercise 9.1.21. (*d*) At what rate does the rabbit population grow?

♣ 9.1.27. A well-known method of generating a sequence of "pseudo-random" integers $u^{(0)}, u^{(1)}, u^{(2)}, \ldots$ satisfying $0 \le u^{(i)} < n$ is based on the *modular Fibonacci equation* $u^{(k+2)} = u^{(k+1)} + u^{(k)} \bmod n$, with suitably chosen initial values $0 \le u^{(0)}, u^{(1)} < n$.

(*a*) Generate the sequence of pseudo-random numbers that result from the choices $n = 10$, $u^{(0)} = 3$, $u^{(1)} = 7$. Keep iterating until the sequence starts repeating.

(*b*) Experiment with other sequences of pseudo-random numbers generated by the method.

9.1.28. Prove that the curves $E_k = \{ T^k \mathbf{x} \mid \| \mathbf{x} \| = 1 \}$, $k = 0, 1, 2, \ldots$, sketched in Figure 9.2 form a family of ellipses with the same principal axes. What are the individual semi-axes? *Hint*: Use Exercise 8.7.23.

♠ 9.1.29. Plot the ellipses $E_k = \{ T^k \mathbf{x} \mid \| \mathbf{x} \| = 1 \}$ for $k = 1, 2, 3, 4$ for the following matrices T. Then determine their principal axes, semi-axes, and areas. *Hint*: Use Exercise 8.7.23.

$$(a)\ \begin{pmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{pmatrix}, \qquad (b)\ \begin{pmatrix} 0 & -1.2 \\ .4 & 0 \end{pmatrix}, \qquad (c)\ \begin{pmatrix} \frac{3}{5} & \frac{1}{5} \\ \frac{2}{5} & \frac{4}{5} \end{pmatrix}.$$

9.1.30. Let T be a positive definite 2×2 matrix. Let $E_n = \{ T^n \mathbf{x} \mid \| \mathbf{x} \| = 1 \}$, $n = 0, 1, 2, \ldots$, be the image of the unit circle under the n^{th} power of T. (*a*) Prove that E_n is an ellipse. *True or false*: (*b*) The ellipses E_n all have the same principal axes. (*c*) The semi-axes are given by $r_n = r_1^n$, $s_n = s_1^n$. (*d*) The areas are given by $A_n = \pi \alpha^n$ where $\alpha = A_1 / \pi$.

9.1.31. Answer Exercise 9.1.30 when T is an arbitrary nonsingular 2×2 matrix. *Hint*: Use Exercise 8.7.23.

9.1.32. Given the general solution (9.9) of the iterative system $\mathbf{u}^{(k+1)} = T \mathbf{u}^{(k)}$, write down the solution to $\mathbf{v}^{(k+1)} = \alpha T \mathbf{v}^{(k)} + \beta \mathbf{v}^{(k)}$, where $\alpha, \beta \in \mathbb{R}$.

♢ 9.1.33. Prove directly that if the coefficient matrix of a linear iterative system is real, both the real and imaginary parts of a complex solution are real solutions.

♢ 9.1.34. Explain why the solution $\mathbf{u}^{(k)}$, $k \ge 0$, to the initial value problem (9.6) exists and is uniquely defined. Does this hold if we allow negative $k < 0$?

9.1.35. Prove that if T is a symmetric matrix, then the coefficients in (9.9) are given by the formula $c_j = \mathbf{a}^T \mathbf{v}_j \,/\, \mathbf{v}_j^T \mathbf{v}_j$.

9.1.36. Explain why the j^{th} column $\mathbf{c}_j^{(k)}$ of the matrix power T^k satisfies the linear iterative system $\mathbf{c}_j^{(k+1)} = T \mathbf{c}_j^{(k)}$ with initial data $\mathbf{c}_j^{(0)} = \mathbf{e}_j$, the j^{th} standard basis vector.

9.1.37. Let $z^{(k+1)} = \lambda z^{(k)}$ be a complex scalar iterative equation with $\lambda = \mu + \mathrm{i}\,\nu$. Show that its real and imaginary parts $x^{(k)} = \operatorname{Re} z^{(k)}$, $y^{(k)} = \operatorname{Im} z^{(k)}$, satisfy a two-dimensional real linear iterative system. Use the eigenvalue method to solve the real 2×2 system, and verify that your solution coincides with the solution to the original complex equation.

♢ 9.1.38. Suppose $V \subset \mathbb{R}^n$ is an invariant subspace for the $n \times n$ matrix T governing the linear iterative system $\mathbf{u}^{(k+1)} = T \mathbf{u}^{(k)}$. Prove that if $\mathbf{u}^{(0)} \in V$, then so is the solution: $\mathbf{u}^{(k)} \in V$.

9.1.39. Suppose $\mathbf{u}^{(k)}$ and $\widetilde{\mathbf{u}}^{(k)}$ are two solutions to the same iterative system $\mathbf{u}^{(k+1)} = T \mathbf{u}^{(k)}$. (*a*) Suppose $\mathbf{u}^{(k_0)} = \widetilde{\mathbf{u}}^{(k_0)}$ for some $k_0 \ge 0$. Can you conclude that these are the same solution: $\mathbf{u}^{(k)} = \widetilde{\mathbf{u}}^{(k)}$ for all k? (*b*) What can you say if $\mathbf{u}^{(k_0)} = \widetilde{\mathbf{u}}^{(k_1)}$ for $k_0 \ne k_1$?

♢ 9.1.40. Let T be an incomplete matrix, and suppose $\mathbf{w}_1, \ldots, \mathbf{w}_j$ is a Jordan chain associated with an incomplete eigenvalue λ. (*a*) Prove that, for $i = 1, \ldots, j$,

$$T^k \mathbf{w}_i = \lambda^k \mathbf{w}_i + k\,\lambda^{k-1} \mathbf{w}_{i-1} + \binom{k}{2} \lambda^{k-2} \mathbf{w}_{i-2} + \;\cdots\; . \tag{9.23}$$

(*b*) Explain how to use a Jordan basis of T to construct the general solution to the linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$.

9.1.41. Use the method Exercise 9.1.40 to find the general real solution to the following linear iterative systems:
(*a*) $u^{(k+1)} = 2\,u^{(k)} + 3\,v^{(k)}, \;\; v^{(k+1)} = 2\,v^{(k)}$,
(*b*) $u^{(k+1)} = u^{(k)} + v^{(k)}, \;\; v^{(k+1)} = -4\,u^{(k)} + 5\,v^{(k)}$,
(*c*) $u^{(k+1)} = -u^{(k)} + v^{(k)} + w^{(k)}, \;\; v^{(k+1)} = -v^{(k)} + w^{(k)}, \;\; w^{(k+1)} = -w^{(k)}$,
(*d*) $u^{(k+1)} = 3\,u^{(k)} - v^{(k)}, \;\; v^{(k+1)} = -u^{(k)} + 3\,v^{(k)} + w^{(k)}, \;\; w^{(k+1)} = -v^{(k)} + 3\,w^{(k)}$,
(*e*) $u^{(k+1)} = u^{(k)} - v^{(k)} - w^{(k)}, \;\; v^{(k+1)} = 2\,u^{(k)} + 2\,v^{(k)} + 2\,w^{(k)}, \;\; w^{(k+1)} = -u^{(k)} + v^{(k)} + w^{(k)}$,
(*f*) $u^{(k+1)} = v^{(k)} + z^{(k)}, \;\; v^{(k+1)} = -u^{(k)} + w^{(k)}, \;\; w^{(k+1)} = z^{(k)}, \;\; z^{(k+1)} = -w^{(k)}$.

9.1.42. Find a formula for the k^{th} power of a Jordan block matrix. *Hint*: Use Exercise 9.1.40.

♡ 9.1.43. An *affine iterative system* has the form $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)} + \mathbf{b}, \;\; \mathbf{u}^{(0)} = \mathbf{c}$.
(*a*) Under what conditions does the system have an equilibrium solution $\mathbf{u}^{(k)} \equiv \mathbf{u}^\star$?
(*b*) In such cases, find a formula for the general solution. *Hint*: Look at $\mathbf{v}^{(k)} = \mathbf{u}^{(k)} - \mathbf{u}^\star$.
(*c*) Solve the following affine iterative systems:

$$(i) \quad \mathbf{u}^{(k+1)} = \begin{pmatrix} 6 & 3 \\ -3 & -4 \end{pmatrix} \mathbf{u}^{(k)} + \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \qquad \mathbf{u}^{(0)} = \begin{pmatrix} 4 \\ -3 \end{pmatrix},$$

$$(ii) \quad \mathbf{u}^{(k+1)} = \begin{pmatrix} -1 & 2 \\ 1 & -1 \end{pmatrix} \mathbf{u}^{(k)} + \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \mathbf{u}^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

$$(iii) \quad \mathbf{u}^{(k+1)} = \begin{pmatrix} -3 & 2 & -2 \\ -6 & 4 & -3 \\ 12 & -6 & -5 \end{pmatrix} \mathbf{u}^{(k)} + \begin{pmatrix} 1 \\ -3 \\ 0 \end{pmatrix}, \qquad \mathbf{u}^{(0)} = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix},$$

$$(iv) \quad \mathbf{u}^{(k+1)} = \begin{pmatrix} -\frac{5}{6} & \frac{1}{3} & -\frac{1}{6} \\ 0 & -\frac{1}{2} & \frac{1}{3} \\ 1 & -1 & \frac{2}{3} \end{pmatrix} \mathbf{u}^{(k)} + \begin{pmatrix} \frac{1}{6} \\ -\frac{1}{3} \\ -\frac{1}{2} \end{pmatrix}, \qquad \mathbf{u}^{(0)} = \begin{pmatrix} \frac{1}{6} \\ -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix}.$$

(*d*) Discuss what happens in cases in which there is no fixed point, assuming that T is complete.

9.2 Stability

With the solution formula (9.9) in hand, we are now in a position to understand the qualitative behavior of solutions to (complete) linear iterative systems. The most important case for applications is when all the iterates converge to $\mathbf{0}$.

Definition 9.9. The equilibrium solution $\mathbf{u}^\star = \mathbf{0}$ to a linear iterative system (9.1) is called *globally asymptotically stable* if all solutions $\mathbf{u}^{(k)} \to \mathbf{0}$ as $k \to \infty$.

Asymptotic stability relies on the following property of the coefficient matrix.

Definition 9.10. A matrix T is called *convergent* if its powers converge to the zero matrix, $T^k \to \mathrm{O}$, meaning that the individual entries of T^k all go to 0 as $k \to \infty$.

The equivalence of the convergence condition and stability of the iterative system follows immediately from the solution formula (9.7).

Theorem 9.11. The linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ has globally asymptotically stable zero solution if and only if T is a convergent matrix.

Proof: If $T^k \to \mathrm{O}$, and $\mathbf{u}^{(k)} = T^k\mathbf{a}$ is any solution, then clearly $\mathbf{u}^{(k)} \to \mathbf{0}$ as $k \to \infty$, proving stability. Conversely, the solution $\mathbf{u}_j^{(k)} = T^k\mathbf{e}_j$ is the same as the j^{th} column of T^k. If the origin is asymptotically stable, then $\mathbf{u}_j^{(k)} \to \mathbf{0}$. Thus, the individual columns of T^k all tend to $\mathbf{0}$, proving that $T^k \to \mathrm{O}$. *Q.E.D.*

To facilitate the analysis of convergence, we shall adopt a norm $\|\cdot\|$ on our underlying vector space, $\mathbb{R}^n$ or $\mathbb{C}^n$. The reader may be inclined to choose the Euclidean (or Hermitian) norm, but, in practice, the ∞ norm

$$\|\mathbf{u}\|_\infty = \max\{\,|u_1|,\ \ldots\ ,|u_n|\,\} \tag{9.24}$$

prescribed by the vector's maximal entry (in modulus) is often easier to work with. Convergence of the iterates is equivalent to convergence of their norms:

$$\mathbf{u}^{(k)} \to \mathbf{0} \qquad \text{if and only if} \qquad \|\mathbf{u}^{(k)}\| \to \mathbf{0} \qquad \text{as} \qquad k \to \infty.$$

The fundamental stability criterion for linear iterative systems relies on the size of the eigenvalues of the coefficient matrix.

Theorem 9.12. The matrix T is convergent, and hence the zero solution of the associated linear iterative system (9.1) is globally asymptotically stable, if and only if all its (complex) eigenvalues have modulus strictly less than one: $|\lambda_j| < 1$.

Proof: Let us prove this result assuming that the coefficient matrix T is complete. (The proof in the incomplete case relies on the Jordan canonical form, and is outlined in Exercise 9.2.18.) If λ_j is an eigenvalue such that $|\lambda_j| < 1$, then the corresponding basis solution $\mathbf{u}_j^{(k)} = \lambda_j^k\mathbf{v}_j$ tends to zero as $k \to \infty$; indeed,

$$\|\mathbf{u}_j^{(k)}\| = \|\lambda_j^k\mathbf{v}_j\| = |\lambda_j|^k\,\|\mathbf{v}_j\| \ \longrightarrow\ 0, \qquad \text{since} \qquad |\lambda_j| < 1.$$

Therefore, if all eigenvalues are less than 1 in modulus, all terms in the solution formula (9.9) tend to zero, which proves asymptotic stability: $\mathbf{u}^{(k)} \to \mathbf{0}$. Conversely, if any eigenvalue satisfies $|\lambda_j| \geq 1$, then the solution $\mathbf{u}^{(k)} = \lambda_j^k\mathbf{v}_j$ does not tend to $\mathbf{0}$ as $k \to \infty$, and hence $\mathbf{0}$ is not asymptotically stable. *Q.E.D.*

Spectral Radius

Consequently, the necessary and sufficient condition for asymptotic stability of a linear iterative system is that all the eigenvalues of the coefficient matrix lie strictly inside the unit circle in the complex plane: $|\lambda_j| < 1$. This criterion can be recast using the following important definition.

Definition 9.13. The *spectral radius* of a matrix T is defined as the maximal modulus of all of its real and complex eigenvalues: $\rho(T) = \max\{\,|\lambda_1|, \ldots, |\lambda_k|\,\}$.

Theorem 9.14. The matrix T is convergent if and only if its spectral radius is strictly less than one: $\rho(T) < 1$.

If T is complete, then we can apply the triangle inequality to (9.9) to estimate

$$\begin{aligned}\| \mathbf{u}^{(k)} \| &= \| c_1 \lambda_1^k \mathbf{v}_1 + \cdots + c_n \lambda_n^k \mathbf{v}_n \| \\ &\le | \lambda_1 |^k \, \| c_1 \mathbf{v}_1 \| + \cdots + | \lambda_n |^k \, \| c_n \mathbf{v}_n \| \\ &\le \rho(T)^k \bigl(| c_1 | \, \| \mathbf{v}_1 \| + \cdots + | c_n | \, \| \mathbf{v}_n \| \bigr) = C \, \rho(T)^k,\end{aligned} \tag{9.25}$$

for some constant $C > 0$ that depends only upon the initial conditions. In particular, if $\rho(T) < 1$, then

$$\| \mathbf{u}^{(k)} \| \;\le\; C \, \rho(T)^k \;\longrightarrow\; 0 \qquad \text{as} \qquad k \to \infty, \tag{9.26}$$

in accordance with Theorem 9.14. Thus, the spectral radius $\rho(T)$ prescribes the rate of convergence of the solutions to equilibrium; the smaller the spectral radius, the faster the solutions go to $\mathbf{0}$.

If T has only one largest (simple) eigenvalue, so $| \lambda_1 | > | \lambda_j |$ for all $j > 1$, then the first term in the solution formula (9.9) will eventually dominate all the others: $\| \lambda_1^k \mathbf{v}_1 \| \gg \| \lambda_j^k \mathbf{v}_j \|$ for $j > 1$ and $k \gg 0$. Therefore, provided that $c_1 \ne 0$, the solution (9.9) has the asymptotic formula

$$\mathbf{u}^{(k)} \;\approx\; c_1 \lambda_1^k \mathbf{v}_1, \tag{9.27}$$

and so most solutions end up parallel to $\mathbf{v}_1$. In particular, if $| \lambda_1 | = \rho(T) < 1$, such a solution approaches $\mathbf{0}$ along the direction of the dominant eigenvector $\mathbf{v}_1$ at a rate governed by the modulus of the dominant eigenvalue. The exceptional solutions, with $c_1 = 0$, tend to $\mathbf{0}$ at a faster rate, along one of the other eigendirections. In practical computations, one rarely observes the exceptional solutions. Indeed, even if the initial condition does not involve the dominant eigenvector, numerical errors during the iteration will almost inevitably introduce a small component in the direction of $\mathbf{v}_1$, which will, if you wait long enough, eventually dominate the solution.

The inequality (9.25) applies only to complete matrices. In the general case, one can prove, cf. Exercise 9.2.18, that the solution satisfies the slightly weaker inequality

$$\| \mathbf{u}^{(k)} \| \le C \, \sigma^k \qquad \text{for all} \qquad k \ge 0, \qquad \text{where} \qquad \sigma > \rho(T) \tag{9.28}$$

is any number larger than the spectral radius, while $C > 0$ is a positive constant (whose value may depend on how close σ is to ρ).

Example 9.15. According to Example 9.7, the matrix

$$T = \begin{pmatrix} -3 & 1 & 6 \\ 1 & -1 & -2 \\ -1 & -1 & 0 \end{pmatrix} \qquad \text{has eigenvalues} \qquad \begin{aligned} \lambda_1 &= -2, \\ \lambda_2 &= -1 + \mathrm{i}, \\ \lambda_3 &= -1 - \mathrm{i}. \end{aligned}$$

Since $| \lambda_1 | = 2 > | \lambda_2 | = | \lambda_3 | = \sqrt{2}$, the spectral radius is $\rho(T) = | \lambda_1 | = 2$. We conclude that T is not a convergent matrix. As the reader can check, either directly, or from the solution formula (9.18), the vectors $\mathbf{u}^{(k)} = T^k \mathbf{u}^{(0)}$ obtained by repeatedly multiplying any nonzero initial vector $\mathbf{u}^{(0)}$ by T rapidly go off to ∞, in successively opposite directions, at a rate roughly equal to $\rho(T)^k = 2^k$.

On the other hand, the matrix

$$\widetilde{T} = -\tfrac{1}{3} \, T = \begin{pmatrix} 1 & -\frac{1}{3} & -2 \\ -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix} \qquad \text{with eigenvalues} \qquad \begin{aligned} \lambda_1 &= \tfrac{2}{3}, \\ \lambda_2 &= \tfrac{1}{3} - \tfrac{1}{3}\,\mathrm{i}, \\ \lambda_3 &= \tfrac{1}{3} + \tfrac{1}{3}\,\mathrm{i}, \end{aligned}$$

has spectral radius $\rho(\widetilde{T}) = \frac{2}{3}$, and hence is a convergent matrix. According to (9.27), if we write the initial data $\mathbf{u}^{(0)} = c_1\,\mathbf{v}_1 + c_2\,\mathbf{v}_2 + c_3\,\mathbf{v}_3$ as a linear combination of the eigenvectors, then, provided $c_1 \neq 0$, the iterates have the asymptotic form $\mathbf{u}^{(k)} \approx c_1 \left(\frac{2}{3}\right)^k \mathbf{v}_1$, where $\mathbf{v}_1 = (\,4, -2, 1\,)^T$ is the eigenvector corresponding to the dominant eigenvalue $\lambda_1 = \frac{2}{3}$. Thus, for most initial vectors, the iterates end up decreasing in length by a factor of almost exactly $\frac{2}{3}$, eventually becoming parallel to the dominant eigenvector $\mathbf{v}_1$. This is borne out by a sample computation: starting with $\mathbf{u}^{(0)} = (\,1, 1, 1\,)^T$, we obtain, for instance,

$$\mathbf{u}^{(15)} = \begin{pmatrix} -.018216 \\ .009135 \\ -.004567 \end{pmatrix}, \qquad \mathbf{u}^{(16)} = \begin{pmatrix} -.012126 \\ .006072 \\ -.003027 \end{pmatrix}, \qquad \mathbf{u}^{(17)} = \begin{pmatrix} -.008096 \\ .004048 \\ -.002018 \end{pmatrix},$$

which form progressively more accurate scalar multiples of the dominant eigenvector $\mathbf{v}_1 = (\,4, -2, 1\,)^T$; moreover, the ratios between their successive entries, $\mathbf{u}_i^{(k+1)}/\mathbf{u}_i^{(k)}$, are approaching the dominant eigenvalue $\lambda_1 = \frac{2}{3}$.

Exercises

9.2.1. Determine the spectral radius of the following matrices:

(a) $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$, (b) $\begin{pmatrix} \frac{1}{3} & -\frac{1}{4} \\ \frac{1}{2} & -\frac{1}{3} \end{pmatrix}$, (c) $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & 1 & 2 \end{pmatrix}$, (d) $\begin{pmatrix} -1 & 5 & -9 \\ 4 & 0 & -1 \\ 4 & -4 & 3 \end{pmatrix}$.

9.2.2. Determine whether or not the following matrices are convergent:

(a) $\begin{pmatrix} 2 & -3 \\ 3 & 2 \end{pmatrix}$, (b) $\begin{pmatrix} .6 & .3 \\ .3 & .7 \end{pmatrix}$, (c) $\frac{1}{5}\begin{pmatrix} 5 & -3 & -2 \\ 1 & -2 & 1 \\ 1 & -5 & 4 \end{pmatrix}$, (d) $\begin{pmatrix} .8 & .3 & .2 \\ .1 & .2 & .6 \\ .1 & .5 & .2 \end{pmatrix}$.

9.2.3. Which of the listed coefficient matrices defines a linear iterative system with asymptotically stable zero solution?

(a) $\begin{pmatrix} -3 & 0 \\ -4 & -1 \end{pmatrix}$, (b) $\begin{pmatrix} \frac{1}{2} & \frac{3}{4} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix}$, (c) $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$, (d) $\begin{pmatrix} -1 & 3 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & -1 \end{pmatrix}$,

(e) $\begin{pmatrix} \frac{1}{2} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{3}{4} & -\frac{1}{2} \\ -\frac{1}{4} & -\frac{1}{4} & \frac{1}{2} \end{pmatrix}$, (f) $\begin{pmatrix} 3 & 0 & -1 \\ 0 & 1 & 0 \\ 2 & 0 & 0 \end{pmatrix}$, (g) $\begin{pmatrix} 1 & 0 & -3 & -2 \\ -\frac{1}{2} & \frac{1}{2} & 2 & \frac{3}{2} \\ -\frac{1}{6} & 0 & \frac{3}{2} & \frac{2}{3} \\ \frac{2}{3} & 0 & -3 & -\frac{5}{3} \end{pmatrix}$.

9.2.4. (a) Determine the eigenvalues and spectral radius of the matrix $T = \begin{pmatrix} 3 & 2 & -2 \\ -2 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$.

(b) Use part (a) to find the eigenvalues and spectral radius of $\widehat{T} = \begin{pmatrix} \frac{3}{5} & \frac{2}{5} & -\frac{2}{5} \\ -\frac{2}{5} & \frac{1}{5} & 0 \\ 0 & \frac{2}{5} & \frac{1}{5} \end{pmatrix}$.

(c) Write down an asymptotic formula for the solutions to $\mathbf{u}^{(k+1)} = \widehat{T}\,\mathbf{u}^{(k)}$.

9.2.5. (a) Show that the spectral radius of $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ is $\rho(T) = 1$.

(b) Show that most iterates $\mathbf{u}^{(k)} = T^k \mathbf{u}^{(0)}$ become unbounded as $k \to \infty$.

(c) Discuss why the inequality $\|\,\mathbf{u}^{(k)}\,\| \le C\,\rho(T)^k$ does not hold when the coefficient matrix is incomplete. (d) Can you prove that (9.28) holds in this example?

9.2.6. Given a linear iterative system with non-convergent matrix, which solutions, if any, will converge to $\mathbf{0}$?

♢ 9.2.7. Suppose T is a complete matrix. (*a*) Prove that every solution to the corresponding linear iterative system is bounded if and only if $\rho(T) \le 1$. (*b*) Can you generalize this result to incomplete matrices? *Hint*: Look at Exercise 9.1.40.

♡ 9.2.8. Discuss the asymptotic behavior of solutions to an iterative system that has two eigenvalues of largest modulus, e.g., $\lambda_1 = -\lambda_2$, or $\lambda_1 = \overline{\lambda_2}$ are complex conjugate eigenvalues. How would you detect this? How can you determine the eigenvalues and eigenvectors?

9.2.9. Suppose T has spectral radius $\rho(T)$. Can you predict the spectral radius of $a\,T + b\,\mathrm{I}$, where a, b are scalars? If not, what additional information do you need?

9.2.10. Prove that if A is any square matrix, then there exists $c \ne 0$ such that the scalar multiple $c\,A$ is a convergent matrix. Find a formula for the largest possible such c.

♡ 9.2.11. Let M_n be the $n \times n$ tridiagonal matrix with all 1's on the sub- and super-diagonals, and zeros on the main diagonal. (*a*) What is the spectral radius of M_n? *Hint*: Use Exercise 8.2.47. (*b*) Is M_n convergent? (*c*) Find the general solution to the iterative system $\mathbf{u}^{(k+1)} = M_n\,\mathbf{u}^{(k)}$.

♡ 9.2.12. Let α, β be scalars. Let $T_{\alpha,\beta}$ be the $n \times n$ tridiagonal matrix that has all α's on the sub- and super-diagonals, and β's on the main diagonal. (*a*) Solve the iterative system $\mathbf{u}^{(k+1)} = T_{\alpha,\beta}\,\mathbf{u}^{(k)}$. (*b*) For which values of α, β is the system asymptotically stable? *Hint*: Combine Exercises 9.2.11 and 9.1.32.

9.2.13. (*a*) Prove that if $|\det T\,| > 1$, then the iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ is unstable.
(*b*) If $|\det T\,| < 1$, is the system asymptotically stable? Prove or give a counterexample.

9.2.14. *True or false*: (*a*) $\rho(c\,A) = c\,\rho(A)$, (*b*) $\rho(S^{-1}AS) = \rho(A)$, (*c*) $\rho(A^2) = \rho(A)^2$, (*d*) $\rho(A^{-1}) = 1/\rho(A)$, (*e*) $\rho(A+B) = \rho(A) + \rho(B)$, (*f*) $\rho(AB) = \rho(A)\,\rho(B)$.

9.2.15. *True or false*: (*a*) If T is convergent, then T^2 is convergent.
(*b*) If A is convergent, then $T = A^T A$ is convergent.

9.2.16. Suppose $T^k \to P$ as $k \to \infty$. (*a*) Prove that P is idempotent: $P^2 = P$.
(*b*) Can you characterize all such matrices P?
(*c*) What are the conditions on the matrix A for this to happen?

9.2.17. Prove that a matrix T with all integer entries is convergent if and only if it is nilpotent, i.e., $T^k = \mathrm{O}$ for some $k \ge 0$. Give a nonzero example of such a matrix.

♢ 9.2.18. Prove the inequality (9.28) when T is incomplete. Use it to complete the proof of Theorem 9.14 in the incomplete case. *Hint*: Use Exercises 9.1.40, 9.2.22.

♢ 9.2.19. Suppose that M is a nonsingular matrix. (*a*) Prove that the *implicit iterative system* $M\,\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)}$ has globally asymptotically stable zero solution if and only if all the eigenvalues of M are strictly greater than one in magnitude: $|\,\mu_i\,| > 1$. (*b*) Let K be another matrix. Prove that more general implicit iterative system of the form $M\,\mathbf{u}^{(n+1)} = K\,\mathbf{u}^{(n)}$ has globally asymptotically stable zero solution if and only if all the generalized eigenvalues of the matrix pair K, M, as in Exercise 8.5.8, are strictly less than 1 in magnitude: $|\,\lambda_i\,| < 1$.

♢ 9.2.20. The *stable subspace* $S \subset \mathbb{R}^n$ for a linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ is defined as the set of all points $\mathbf{a}$ such that the solution with initial condition $\mathbf{u}^{(0)} = \mathbf{a}$ satisfies $u^{(k)} \to \mathbf{0}$ as $k \to \infty$. (*a*) Prove that S is an invariant subspace for the matrix T.
(*b*) Determine necessary and sufficient conditions for $\mathbf{a} \in S$.
(*c*) Find the stable subspace for the linear systems in Exercise 9.1.14

♡ 9.2.21. Consider a second order iterative system $\mathbf{u}^{(k+2)} = A\,\mathbf{u}^{(k+1)} + B\,\mathbf{u}^{(k)}$, where A, B are $n \times n$ matrices. Define a *quadratic eigenvalue* to be a complex number that satisfies $\det(\lambda^2\, \mathrm{I} - \lambda\, A - B) = 0$. Prove that the zero solution is globally asymptotically stable if and only if all its quadratic eigenvalues satisfy $|\,\lambda\,| < 1$.

◇ 9.2.22. Let $p(t)$ be a polynomial. Assume $0 < \lambda < \mu$. Prove that there is a positive constant C such that $p(n)\,\lambda^n < C\,\mu^n$ for all $n > 0$.

Fixed Points

The zero vector $\mathbf{0}$ is always a *fixed point* for a linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$, since $\mathbf{0} = T\,\mathbf{0}$, and so $\mathbf{u}^{(k)} \equiv \mathbf{0}$ is an equilibrium solution. Are there any others? The answer is immediate: $\mathbf{u}^\star$ is a fixed point if and only if $\mathbf{u}^\star = T\,\mathbf{u}^\star$, and hence $\mathbf{u}^\star$ satisfies the eigenvalue equation for T with for the unit eigenvalue $\lambda = 1$. Thus, the system admits a nonzero fixed point if and only if the coefficient matrix T has 1 as an eigenvalue. Since every nonzero scalar multiple of the eigenvector $\mathbf{u}^\star$ is also an eigenvector, in such cases the system has infinitely many fixed points, namely all elements of the eigenspace $V_1 = \ker(T - \mathrm{I}\,)$, including $\mathbf{0}$. We are interested in whether the fixed points are *stable* in the sense that solutions having nearby initial conditions remain nearby. More precisely:

Definition 9.16. A fixed point $\mathbf{u}^\star$ of an iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ is called *stable* if for every $\varepsilon > 0$ there exists a $\delta > 0$ such that whenever $\|\,\mathbf{u}^{(0)} - \mathbf{u}^\star\,\| < \delta$, then the resulting iterates satisfy $\|\,\mathbf{u}^{(k)} - \mathbf{u}^\star\,\| < \varepsilon$ for all k.

The stability of the fixed points, at least if the coefficient matrix is complete, is governed by the same solution formula (9.9). If the eigenvalue $\lambda_1 = 1$ is simple, and all other eigenvalues are less than one in modulus, so

$$1 = \lambda_1 > |\,\lambda_2\,| \geq \ \cdots\ \geq |\,\lambda_n\,|,$$

then the solution takes the asymptotic form

$$\mathbf{u}^{(k)} = c_1\,\mathbf{v}_1 + c_2\,\lambda_2^k\,\mathbf{v}_2 + \ \cdots\ + c_n\,\lambda_n^k\,\mathbf{v}_n \longrightarrow c_1\,\mathbf{v}_1, \qquad \text{as} \qquad k \longrightarrow \infty, \tag{9.29}$$

converging to one of the fixed points, i.e., to a multiple of the eigenvector $\mathbf{v}_1$. The coefficient c_1 is prescribed by the initial conditions, cf. (9.10). The rate of convergence of the solution is governed by the modulus $|\,\lambda_2\,|$ of the *subdominant eigenvalue*.

Proposition 9.17. Suppose that T has a simple (or, more generally, complete) eigenvalue $\lambda_1 = 1$, and, moreover, all other eigenvalues satisfy $|\,\lambda_j\,| < 1$. Then all solutions to the linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ converge to a vector $\mathbf{v} \in V_1$ that lies in the $\lambda_1 = 1$ eigenspace. Moreover, all the fixed points $\mathbf{v} \in V_1$ of T are *stable*.

Stability of a fixed point does not imply asymptotic stability, since nearby solutions may converge to a nearby fixed point, i.e., a slightly different element of the eigenspace V_1.

The general necessary and sufficient conditions for stability of the fixed points of a linear iterative system is governed by the spectral radius of its coefficient matrix, as follows. The proof is relegated to Exercise 9.2.28.

Theorem 9.18. The fixed points of an iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ are stable if and only if $\rho(T) \leq 1$ and, moreover, every eigenvalue of modulus $|\,\lambda\,| = 1$ is complete.

Thus, with regard to linear iterative systems, either all fixed points are stable or all are unstable. Keep in mind that the fixed points are the elements of the eigenspace V_1 corresponding to the eigenvalue $\lambda = 1$, if such exists. If 1 is not an eigenvalue of T, then $\mathbf{u}^\star = \mathbf{0}$ is the only fixed point.

Example 9.19. Consider the iterative system with coefficient matrix

$$T = \begin{pmatrix} \frac{3}{2} & -\frac{1}{2} & -3 \\ -\frac{1}{2} & \frac{1}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}.$$

The eigenvalues and corresponding eigenvectors are

$$\lambda_1 = 1, \qquad \lambda_2 = \frac{1+\mathrm{i}}{2}, \qquad \lambda_3 = \frac{1-\mathrm{i}}{2},$$

$$\mathbf{v}_1 = \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix}, \qquad \mathbf{v}_2 = \begin{pmatrix} 2-\mathrm{i} \\ -1 \\ 1 \end{pmatrix}, \qquad \mathbf{v}_3 = \begin{pmatrix} 2+\mathrm{i} \\ -1 \\ 1 \end{pmatrix}.$$

Since $\lambda_1 = 1$, every scalar multiple of the eigenvector $\mathbf{v}_1$ is a fixed point. The fixed points are stable, since the remaining eigenvalues have modulus $|\,\lambda_2\,| = |\,\lambda_3\,| = \frac{1}{2}\sqrt{2} \approx .7071 < 1$. Thus, the iterates $\mathbf{u}^{(k)} = T^k \mathbf{a} \longrightarrow c_1 \mathbf{v}_1$ will eventually converge to a multiple of the first eigenvector; in almost all cases the convergence rate is $\frac{1}{2}\sqrt{2}$. For example, starting with $\mathbf{u}^{(0)} = (\,1,1,1\,)^T$, leads to the iterates[†]

$$\mathbf{u}^{(5)} = \begin{pmatrix} -9.5 \\ 4.75 \\ -2.75 \end{pmatrix}, \qquad \mathbf{u}^{(10)} = \begin{pmatrix} -7.9062 \\ 3.9062 \\ -1.9062 \end{pmatrix}, \qquad \mathbf{u}^{(15)} = \begin{pmatrix} -7.9766 \\ 4.0 \\ -2.0 \end{pmatrix},$$

$$\mathbf{u}^{(20)} = \begin{pmatrix} -8.0088 \\ 4.0029 \\ -2.0029 \end{pmatrix}, \qquad \mathbf{u}^{(25)} = \begin{pmatrix} -7.9985 \\ 3.9993 \\ -1.9993 \end{pmatrix}, \qquad \mathbf{u}^{(30)} = \begin{pmatrix} -8.0001 \\ 4.0001 \\ -2.0001 \end{pmatrix},$$

which are gradually converging to the particular eigenvector $(\,-8,4,-2\,)^T = -2\,\mathbf{v}_1$. This can be predicted in advance by decomposing the initial vector into a linear combination of the eigenvectors:

$$\mathbf{u}^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -2 \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix} + \frac{3+3\,\mathrm{i}}{2} \begin{pmatrix} 2-\mathrm{i} \\ -1 \\ 1 \end{pmatrix} + \frac{3-3\,\mathrm{i}}{2} \begin{pmatrix} 2+\mathrm{i} \\ -1 \\ 1 \end{pmatrix},$$

whence

$$\mathbf{u}^{(k)} = \begin{pmatrix} -8 \\ 4 \\ -2 \end{pmatrix} + \frac{3+3\,\mathrm{i}}{2} \left(\frac{1+\mathrm{i}}{2} \right)^k \begin{pmatrix} 2-\mathrm{i} \\ -1 \\ 1 \end{pmatrix} + \frac{3-3\,\mathrm{i}}{2} \left(\frac{1-\mathrm{i}}{2} \right)^k \begin{pmatrix} 2+\mathrm{i} \\ -1 \\ 1 \end{pmatrix},$$

and so $\mathbf{u}^{(k)} \to (\,-8,4,-2\,)^T$ as $k \to \infty$. Despite the complex formula, the solution is, in fact, real.

[†] Since the convergence is slow, we only display every fifth one.

Exercises

9.2.23. Find all fixed points for the iterative systems with the following coefficient matrices:

$$(a)\ \begin{pmatrix} .7 & .3 \\ .2 & .8 \end{pmatrix}, \quad (b)\ \begin{pmatrix} .6 & 1.0 \\ .3 & -.7 \end{pmatrix}, \quad (c)\ \begin{pmatrix} -1 & -1 & -4 \\ -2 & 0 & -4 \\ 1 & -1 & 0 \end{pmatrix}, \quad (d)\ \begin{pmatrix} 2 & 1 & -1 \\ 2 & 3 & -2 \\ -1 & -1 & 2 \end{pmatrix}.$$

9.2.24. Discuss the stability of each fixed point and the asymptotic behavior(s) of the solutions to the systems in Exercise 9.2.23. Which fixed point, if any, does the solution with initial condition $\mathbf{u}^{(0)} = \mathbf{e}_1$ converge to?

9.2.25. Suppose T is a symmetric matrix that satisfies the hypotheses of Proposition 9.17 with a simple eigenvalue $\lambda_1 = 1$. Prove that the solution $\mathbf{u}^{(k)}$ to the linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ has limiting value $\displaystyle\lim_{k\to\infty} \mathbf{u}^{(k)} = \frac{\mathbf{u}^{(0)} \cdot \mathbf{v}_1}{\|\,\mathbf{v}_1\,\|^2}\,\mathbf{v}_1$.

9.2.26. *True or false*: If T has a stable nonzero fixed point, then it is a convergent matrix.

9.2.27. *True or false*: If every point $\mathbf{u} \in \mathbb{R}^n$ is a fixed point, then they are all stable. Can you characterize such systems?

◇ 9.2.28. Prove Theorem 9.18: (*a*) assuming T is complete, (*b*) for general T.
Hint: Use Exercise 9.1.40.

♡ 9.2.29. (*a*) Under what conditions does the linear iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$ have a *period* 2 *solution*, meaning that the iterates repeat after every other iterate: $\mathbf{u}^{(k+2)} = \mathbf{u}^{(k)} \neq \mathbf{u}^{(k+1)}$? Give an example of such a system. (*b*) Under what conditions is there a unique period 2 solution? (*c*) What about a period m solution for $2 < m \in \mathbb{N}$?

Matrix Norms and Convergence

As we now know, the convergence of a linear iterative system is governed by the spectral radius, or, equivalently, the modulus of the largest eigenvalue of the coefficient matrix. Unfortunately, finding accurate approximations to the eigenvalues of most matrices is a nontrivial computational task. Indeed, as we will learn in Section 9.5, all practical numerical algorithms rely on some form of iteration. But using iteration to determine the spectral radius defeats the purpose, which is to predict the behavior of the iterative system in advance! One independent means of accomplishing this is through matrix norms, as introduced at the end of Section 3.3.

Let $\|\,\mathbf{v}\,\|$ denote a norm on[†] $\mathbb{R}^n$. Theorem 3.20 defines the induced natural matrix norm on the space of $n \times n$ matrices, denoted by $\|\,A\,\| = \max\{\,\|\,A\mathbf{u}\,\| \mid \|\,\mathbf{u}\,\| = 1\,\}$. The following result relates the magnitude of the norm of a matrix to convergence of the associated iterative system.

Proposition 9.20. If A is a square matrix, then $\|\,A^k\,\| \leq \|\,A\,\|^k$. In particular, if $\|\,A\,\| < 1$, then $\|\,A^k\,\| \to 0$ as $k \to \infty$, and hence A is a convergent matrix: $A^k \to \mathrm{O}$.

The first part is a restatement of Proposition 3.22, and the second part is an immediate consequence. The converse to this result is not quite true; a convergent matrix does not

[†] We work with real iterative systems throughout this chapter, but the methods readily extend to their complex counterparts.

necessarily have matrix norm less than 1, or even ≤ 1 — see Example 9.23 below. An alternative proof of Proposition 9.20 can be based on the following useful estimate:

Theorem 9.21. The spectral radius of a matrix is bounded by its matrix norm:

$$\rho(A) \leq \| A \|. \tag{9.30}$$

Proof: If λ is a real eigenvalue, and $\mathbf{u}$ a corresponding unit eigenvector, so that $A\mathbf{u} = \lambda\,\mathbf{u}$ with $\|\mathbf{u}\| = 1$, then

$$\| A\mathbf{u} \| = \| \lambda\,\mathbf{u} \| = |\,\lambda\,|\ \|\mathbf{u}\| = |\,\lambda\,|. \tag{9.31}$$

Since $\| A \|$ is the maximum of $\| A\mathbf{u} \|$ over all possible unit vectors, this implies that

$$|\,\lambda\,| \leq \| A \|. \tag{9.32}$$

If all the eigenvalues of A are real, then the spectral radius is the maximum of their absolute values, and so it too is bounded by $\| A \|$, proving (9.30).

If A has complex eigenvalues, then we need to work a little harder to establish (9.32). (This is because the matrix norm is defined by the effect of A on *real* vectors, and so we cannot directly use the complex eigenvectors to establish the required bound.) Let $\lambda = r\,e^{\,\mathrm{i}\,\theta}$ be a complex eigenvalue with complex eigenvector $\mathbf{z} = \mathbf{x} + \mathrm{i}\,\mathbf{y}$. Define

$$\mu = \min \left\{ \, \| \operatorname{Re}\,(e^{\,\mathrm{i}\,\varphi}\,\mathbf{z}) \| = \| (\cos\varphi)\,\mathbf{x} - (\sin\varphi)\,\mathbf{y} \| \;\middle|\; 0 \leq \varphi \leq 2\pi \, \right\}. \tag{9.33}$$

Since the indicated subset is a closed curve (in fact, an ellipse) that does not go through the origin[†], $\mu > 0$. Let φ_0 denote the value of the angle that produces the minimum, so

$$\mu = \| (\cos\varphi_0)\,\mathbf{x} - (\sin\varphi_0)\,\mathbf{y} \| = \| \operatorname{Re}\left(e^{\,\mathrm{i}\,\varphi_0}\,\mathbf{z} \right) \|.$$

Define the real unit vector

$$\mathbf{u} = \frac{\operatorname{Re}\,(e^{\,\mathrm{i}\,\varphi_0}\,\mathbf{z})}{\mu} = \frac{(\cos\varphi_0)\,\mathbf{x} - (\sin\varphi_0)\,\mathbf{y}}{\mu}, \qquad \text{so that} \qquad \|\mathbf{u}\| = 1.$$

Then

$$A\mathbf{u} = \frac{1}{\mu}\operatorname{Re}\left(e^{\,\mathrm{i}\,\varphi_0}\,A\,\mathbf{z} \right) = \frac{1}{\mu}\operatorname{Re}\left(e^{\,\mathrm{i}\,\varphi_0}\,r\,e^{\,\mathrm{i}\,\theta}\,\mathbf{z} \right) = \frac{r}{\mu}\operatorname{Re}\left(e^{\,\mathrm{i}\,(\varphi_0+\theta)}\,\mathbf{z} \right).$$

Therefore, keeping in mind that m is the minimal value in (9.33),

$$\| A \| \geq \| A\mathbf{u} \| = \frac{r}{\mu}\,\| \operatorname{Re}\left(e^{\,\mathrm{i}\,(\varphi_0+\theta)}\,\mathbf{z} \right) \| \geq r = |\,\lambda\,|, \tag{9.34}$$

and so (9.32) also holds for complex eigenvalues. *Q.E.D.*

Let us see what the convergence criterion of Proposition 9.20 says for a couple of our well-known matrix norms. First, the formula (3.44) for the ∞ norm implies the following convergence criterion.

Proposition 9.22. If all the absolute row sums of A are strictly less than 1, then $\| A \|_\infty < 1$ and hence A is a convergent matrix.

Example 9.23. Consider the symmetric matrix $A = \begin{pmatrix} \frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{1}{4} \end{pmatrix}$. Its two absolute row sums are $\left|\,\frac{1}{2}\,\right| + \left|\,-\frac{1}{3}\,\right| = \frac{5}{6}$, $\left|\,-\frac{1}{3}\,\right| + \left|\,\frac{1}{4}\,\right| = \frac{7}{12}$, so

$$\| A \|_\infty = \max\left\{ \tfrac{5}{6}, \tfrac{7}{12} \right\} = \tfrac{5}{6} = .83333\ldots\,.$$

[†] This relies on the fact that $\mathbf{x}, \mathbf{y}$ are linearly independent, which was shown in Exercise 8.3.12.

Since the norm is less than 1, A is a convergent matrix. Indeed, its eigenvalues are

$$\lambda_1 = \frac{9+\sqrt{73}}{24} = .731000\ldots\,, \qquad\qquad \lambda_2 = \frac{9-\sqrt{73}}{24} = .018999\ldots\,,$$

and hence the spectral radius is $\rho(A) = \lambda_1 = .731000\ldots$, which is slightly smaller than its ∞ norm.

The row sum test for convergence is not always conclusive. For example, the matrix

$$A = \begin{pmatrix} \frac{1}{2} & -\frac{3}{5} \\ -\frac{3}{5} & \frac{1}{4} \end{pmatrix} \qquad \text{has matrix norm} \qquad \|A\|_\infty = \tfrac{11}{10} > 1.$$

On the other hand, its eigenvalues are $\dfrac{15 \pm \sqrt{601}}{40}$ and hence its spectral radius is $\rho(A) = \dfrac{15+\sqrt{601}}{40} = .987882\ldots$, which implies that A is (just barely) convergent, even though its maximal row sum is larger than 1.

Similarly, using the formula (8.61) for the Euclidean matrix norm, one deduces a convergence criterion based on the magnitude of the singular values.

Proposition 9.24. If A is a square matrix whose largest singular value satisfies $\sigma_1 < 1$, then $\|A\|_2 < 1$ and hence A is a convergent matrix.

Example 9.25. Consider the matrix and associated Gram matrix

$$A = \begin{pmatrix} 0 & -\frac{1}{3} & \frac{1}{3} \\ \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{2}{5} & \frac{1}{5} & 0 \end{pmatrix}, \qquad A^T A = \begin{pmatrix} .2225 & .0800 & .1250 \\ .0800 & .1511 & -.1111 \\ .1250 & -.1111 & .3611 \end{pmatrix}.$$

Then $A^T A$ has eigenvalues $\lambda_1 = .4472$, $\lambda_2 = .2665$, $\lambda_3 = .0210$, and hence the singular values of A are their square roots: $\sigma_1 = .6687$, $\sigma_2 = .5163$, $\sigma_3 = .1448$. The Euclidean matrix norm of A is the largest singular value, and so $\|A\|_2 = .6687$, proving that A is a convergent matrix. Note that, as always, the matrix norm overestimates the spectral radius, which is $\rho(A) = .5$.

Unfortunately, as we discovered in Example 9.23, matrix norms are not a foolproof test of convergence. There exist convergent matrices such that $\rho(A) < 1$ that yet have matrix norm $\|A\| \geq 1$. In such cases, the matrix norm is not able to predict convergence of the iterative system, although one should expect the convergence to be quite slow. Although such pathology might show up in the chosen matrix norm, it turns out that one can always rig up some matrix norm for which $\|A\| < 1$. This follows from a more general result, whose proof can be found in [**62**].

Theorem 9.26. Let A have spectral radius $\rho(A)$. If $\varepsilon > 0$ is any positive number, then there exists a matrix norm $\|\cdot\|$ such that

$$\rho(A) \leq \|A\| < \rho(A) + \varepsilon. \tag{9.35}$$

Corollary 9.27. If A is a convergent matrix, then there exists a matrix norm such that $\|A\| < 1$.

Proof: By definition, A is convergent if and only if $\rho(A) < 1$. Choose $\varepsilon > 0$ such that $\rho(A) + \varepsilon < 1$. Any norm that then satisfies (9.35) has the desired property. *Q.E.D.*

It can also be proved, [**48**], that, given a matrix norm, $\lim_{n\to\infty} \| A^n \|^{1/n} = \rho(A)$, and hence, if A is convergent, then $\| A^n \| < 1$ for n sufficiently large.

Warning. Based on the accumulated evidence, one might be tempted to speculate that the spectral radius itself defines a matrix norm. Unfortunately, this is not the case. For example, the nonzero matrix $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ has zero spectral radius, $\rho(A) = 0$, in violation of a basic norm axiom.

Exercises

9.2.30. Compute the ∞ matrix norm of the following matrices. Which are guaranteed to be convergent? (*a*) $\begin{pmatrix} \frac12 & \frac14 \\ \frac13 & \frac16 \end{pmatrix}$, (*b*) $\begin{pmatrix} \frac53 & \frac43 \\ -\frac76 & -\frac56 \end{pmatrix}$, (*c*) $\begin{pmatrix} \frac27 & -\frac27 \\ -\frac27 & \frac67 \end{pmatrix}$, (*d*) $\begin{pmatrix} \frac14 & \frac32 \\ -\frac12 & \frac54 \end{pmatrix}$, (*e*) $\begin{pmatrix} \frac27 & \frac27 & -\frac47 \\ 0 & \frac27 & \frac67 \\ \frac27 & \frac47 & \frac27 \end{pmatrix}$, (*f*) $\begin{pmatrix} 0 & .1 & .8 \\ -.1 & 0 & .1 \\ -.8 & -.1 & 0 \end{pmatrix}$, (*g*) $\begin{pmatrix} 1 & -\frac23 & -\frac23 \\ 1 & -\frac13 & -1 \\ \frac13 & -\frac23 & 0 \end{pmatrix}$, (*h*) $\begin{pmatrix} \frac13 & 0 & 0 \\ -\frac13 & 0 & \frac13 \\ 0 & \frac23 & \frac13 \end{pmatrix}$.

9.2.31. Compute the Euclidean matrix norm of each matrix in Exercise 9.2.30. Have your convergence conclusions changed?

9.2.32. Compute the spectral radii of the matrices in Exercise 9.2.30. Which are convergent? Compare your conclusions with those of Exercises 9.2.30 and 9.2.31.

9.2.33. Let k be an integer and set $A_k = \begin{pmatrix} k & -1 \\ k^2 & -k \end{pmatrix}$. Compute (a) $\| A_k \|_\infty$, (*b*) $\| A_k \|_2$, (*c*) $\rho(A_k)$. (*d*) Explain why every A_k is a convergent matrix, even though their matrix norms can be arbitrarily large. (*e*) Why does this not contradict Corollary 9.27?

9.2.34. Show that if $| c | < 1/\| A \|$, then $c A$ is a convergent matrix.

♢ 9.2.35. Prove that the spectral radius function does *not* satisfy the triangle inequality by finding matrices A, B such that $\rho(A + B) > \rho(A) + \rho(B)$.

9.2.36. Find a convergent matrix that has dominant singular value $\sigma_1 > 1$.

♢ 9.2.37. Prove that if A is a real symmetric matrix, then its Euclidean matrix norm is equal to its spectral radius.

♢ 9.2.38. Let A be a square matrix. Let $s = \max\{s_1, \ldots, s_n\}$ be the maximal absolute row sum of A and let $t = \min\left\{ | a_{ii} | - r_i \right\}$, with r_i given by (8.27). Prove that $\max\{0, t\} \leq \rho(A) \leq s$.

9.2.39. Suppose the largest entry (in modulus) of A is $| a_{ij} | = a_\star$. Can you bound its radius of convergence?

9.2.40. (a) Suppose that every entry of the $n \times n$ matrix A is bounded by $| a_{ij} | < 1/n$. Prove that A is a convergent matrix. *Hint*: Use Exercise 9.2.38. (*b*) Produce a matrix of size $n \times n$ with one or more entries satisfying $| a_{ij} | = 1/n$ that is not convergent.

9.2.41. Write down an example of a strictly diagonally dominant matrix that is also convergent.

9.2.42. *True or false*: If $B = S^{-1} A S$ are similar matrices, then

$$\text{(a)}\ \| B \|_\infty = \| A \|_\infty, \quad \text{(b)}\ \| B \|_2 = \| A \|_2, \quad \text{(c)}\ \rho(B) = \rho(A).$$

9.2.43. Prove that the curve parametrized in (9.33) is an ellipse. What are its semi-axes?

◇ 9.2.44. (a) Prove that the individual entries a_{ij} of a matrix A are bounded in absolute value by its ∞ matrix norm: $|\,a_{ij}\,| \le \|\,A\,\|_\infty$. (b) Prove that if the series $\sum_{n=0}^{\infty} \|\,A_n\,\|_\infty < \infty$ converges, then the *matrix series* $\sum_{n=0}^{\infty} A_n = A^\star$ converges to some matrix $A^\star$.
(c) Let $\|\,A\,\|$ denote any natural matrix norm. Prove that if the series $\sum_{n=0}^{\infty} \|\,A_n\,\| < \infty$ converges, then the matrix series $\sum_{n=0}^{\infty} A_n = A^\star$ converges.

9.2.45. (a) Use Exercise 9.2.44 to prove that the *geometric matrix series* $\sum_{n=0}^{\infty} A^n$ converges whenever $\rho(A) < 1$. *Hint*: Apply Corollary 9.27.
(b) Prove that the sum equals $(\,\mathrm{I} - A)^{-1}$. How do you know $\mathrm{I} - A$ is invertible?

9.3 Markov Processes

A discrete probabilistic process in which the future state of a system depends only upon its current configuration is known as a *Markov chain*, to honor the pioneering early twentieth studies of the Russian mathematician Andrei Markov. Markov chains are described by linear iterative systems whose coefficient matrices have a special form. They define the simplest examples of stochastic processes, [**4**, **23**], which have many profound physical, biological, economic, and statistical applications, including networks, internet search engines, speech recognition, and routing.

To take a very simple (albeit slightly artificial) example, suppose you would like to be able to predict the weather in your city. Consulting local weather records over the past decade, you determine that

(a) If today is sunny, there is a 70% chance that tomorrow will also be sunny,

(b) But, if today is cloudy, the chances are 80% that tomorrow will also be cloudy.

Question: given that today is sunny, what is the probability that next Saturday's weather will also be sunny?

To formulate this process mathematically, we let $s^{(k)}$ denote the probability that day k is sunny and $c^{(k)}$ the probability that it is cloudy. If we assume that these are the only possibilities, then the individual probabilities must sum to 1, so

$$s^{(k)} + c^{(k)} = 1.$$

According to our data, the probability that the next day is sunny or cloudy is expressed by the equations

$$s^{(k+1)} = .7\, s^{(k)} + .2\, c^{(k)}, \qquad\qquad c^{(k+1)} = .3\, s^{(k)} + .8\, c^{(k)}. \tag{9.36}$$

Indeed, day $k+1$ could be sunny either if day k was, with a 70% chance, or, if day k was cloudy, there is still a 20% chance of day $k+1$ being sunny. We rewrite (9.36) in a more convenient matrix form:

$$\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}, \qquad \text{where} \qquad T = \begin{pmatrix} .7 & .2 \\ .3 & .8 \end{pmatrix}, \qquad \mathbf{u}^{(k)} = \begin{pmatrix} s^{(k)} \\ c^{(k)} \end{pmatrix}. \tag{9.37}$$

In a Markov process, the vector of probabilities $\mathbf{u}^{(k)}$ is known as the k^{th} *state vector* and the matrix T is known as the *transition matrix*, whose entries fix the transition probabilities between the various states.

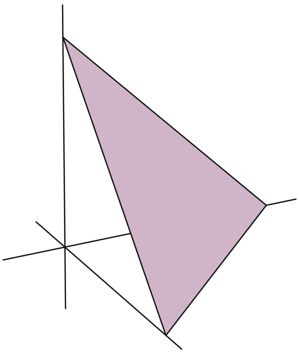


Figure 9.4. The Set of Probability Vectors in $\mathbb{R}^3$.

By assumption, the initial state vector is $\mathbf{u}^{(0)} = (1, 0)^T$, since we know for certain that today is sunny. Rounded off to three decimal places, the subsequent state vectors are

$$\mathbf{u}^{(1)} \simeq \begin{pmatrix} .7 \\ .3 \end{pmatrix}, \qquad \mathbf{u}^{(2)} \simeq \begin{pmatrix} .55 \\ .45 \end{pmatrix}, \qquad \mathbf{u}^{(3)} \simeq \begin{pmatrix} .475 \\ .525 \end{pmatrix}, \qquad \mathbf{u}^{(4)} \simeq \begin{pmatrix} .438 \\ .563 \end{pmatrix},$$
$$\mathbf{u}^{(5)} \simeq \begin{pmatrix} .419 \\ .581 \end{pmatrix}, \qquad \mathbf{u}^{(6)} \simeq \begin{pmatrix} .410 \\ .591 \end{pmatrix}, \qquad \mathbf{u}^{(7)} \simeq \begin{pmatrix} .405 \\ .595 \end{pmatrix}, \qquad \mathbf{u}^{(8)} \simeq \begin{pmatrix} .402 \\ .598 \end{pmatrix}.$$

The iterates converge fairly rapidly to $(.4, .6)^T$, which is, in fact, a fixed point for the iterative system (9.37). Thus, in the long run, 40% of the days will be sunny and 60% will be cloudy. Let us explain why this happens.

Definition 9.28. A vector $\mathbf{u} = (u_1, u_2, \ldots, u_n)^T \in \mathbb{R}^n$ is called a *probability vector* if all its entries lie between 0 and 1, so $0 \le u_i \le 1$ for $i = 1, \ldots, n$, and, moreover, their sum is $u_1 + \cdots + u_n = 1$.

We interpret the entry u_i of a probability vector as the probability that the system is in state number i. The fact that the entries add up to 1 means that they represent a complete list of probabilities for the possible states of the system. The set of probability vectors defines an $(n-1)$-dimensional *simplex* in $\mathbb{R}^n$. For example, the possible probability vectors $\mathbf{u} \in \mathbb{R}^3$ fill the equilateral triangle plotted in Figure 9.4.

Remark. Every nonzero vector $\mathbf{0} \ne \mathbf{v} = (v_1, v_2, \ldots, v_n)^T$ with all non-negative entries, $v_i \ge 0$ for $i = 1, \ldots, n$, can be converted into a parallel probability vector by dividing by the sum of its entries:

$$\mathbf{u} = \frac{\mathbf{v}}{v_1 + \cdots + v_n}. \tag{9.38}$$

For example, if $\mathbf{v} = (3, 2, 0, 1)^T$, then $\mathbf{u} = \left(\frac{1}{2}, \frac{1}{3}, 0, \frac{1}{6}\right)^T$ is the corresponding probability vector.

In general, a *Markov chain* is represented by a first order linear iterative system

$$\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}, \tag{9.39}$$

whose initial state $\mathbf{u}^{(0)}$ is a probability vector. The entries of the *transition matrix* T must satisfy

$$0 \le t_{ij} \le 1, \qquad\qquad t_{1j} + \cdots + t_{nj} = 1. \tag{9.40}$$

The entry t_{ij} represents the *transitional probability* that the system will switch from state j to state i. (Note the reversal of indices.) Since this covers all possible transitions, the *column sums* of the transition matrix are all equal to 1, and hence each column of T is a probability vector, which is equivalent to condition (9.40). In Exercise 9.3.24 you are asked to show that, under these assumptions, if $\mathbf{u}^{(k)}$ is a probability vector, then so is $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)}$, and hence, given our assumption on the initial state, the solution $\mathbf{u}^{(k)} = T^k\,\mathbf{u}^{(0)}$ to the Markov process defines a sequence, or "chain", of probability vectors.

Let us now investigate the convergence of the Markov chain. Not all Markov chains converge — see Exercise 9.3.9 for an example — and so we impose some additional mild restrictions on the transition matrix.

Definition 9.29. A transition matrix (9.40) is *regular* if some power T^k contains no zero entries. In particular, if T itself has no zero entries, then it is regular.

Warning. The term "regular transition matrix" has nothing to do with our earlier term "regular matrix", which was used to describe matrices with an LU factorization.

The entries of T^k describe the transition probabilities of getting from one state to another in k steps. Thus, regularity of the transition matrix means that there is a nonzero probability of getting from any state to any other state in exactly k steps for some $k \geq 1$.

The asymptotic behavior of a regular Markov chain is governed by the following basic result, originally due to the German mathematicians Oskar Perron and Georg Frobenius in the early part of the twentieth century. A proof can be found at the end of this section.

Theorem 9.30. If T is a regular transition matrix, then it admits a unique *probability eigenvector* $\mathbf{u}^\star$ with eigenvalue $\lambda_1 = 1$. Moreover, a Markov chain with coefficient matrix T will converge to the probability eigenvector: $\mathbf{u}^{(k)} \to \mathbf{u}^\star$ as $k \to \infty$.

Example 9.31. The eigenvalues and eigenvectors of the weather transition matrix (9.37) are

$$\lambda_1 = 1, \qquad \mathbf{v}_1 = \begin{pmatrix} \frac{2}{3} \\ 1 \end{pmatrix}, \qquad\qquad \lambda_2 = .5, \qquad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

The first eigenvector is then converted into a probability vector via formula (9.38):

$$\mathbf{u}^\star = \mathbf{u}_1 = \frac{1}{1+\frac{2}{3}} \begin{pmatrix} \frac{2}{3} \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2}{5} \\ \frac{3}{5} \end{pmatrix}.$$

This distinguished probability eigenvector represents the final asymptotic state of the system after many iterations, *no matter what the initial state is*. Thus, our earlier observation that about 40% of the days will be sunny and 60% will be cloudy does not depend upon today's weather.

Example 9.32. A taxi company in Minnesota serves the cities of Minneapolis and St. Paul, as well as the nearby suburbs. Records indicate that, on average, 10% of the customers taking a taxi in Minneapolis go to St. Paul and 30% go to the suburbs. Customers boarding in St. Paul have a 30% chance of going to Minneapolis and a 30% chance of going to the suburbs, while suburban customers choose Minneapolis 40% of the time and St. Paul 30% of the time. The owner of the taxi company is interested in knowing where the taxis will end up, on average. Let us write this as a Markov process. The entries of the state vector $\mathbf{u}^{(k)} = (u_1^{(k)}, u_2^{(k)}, u_3^{(k)})^T$ tell what proportion of the taxi fleet is, respectively, in Minneapolis, St. Paul, and the suburbs, or, equivalently, the probability that an individual taxi will be in one of the three locations. Using the given data, we construct the relevant transition matrix

$$T = \begin{pmatrix} .6 & .3 & .4 \\ .1 & .4 & .3 \\ .3 & .3 & .3 \end{pmatrix}.$$

Note that T is regular since it has no zero entries. The probability eigenvector

$$\mathbf{u}^\star \simeq (\, .4714, \;\; .2286, \;\; .3 \,)^T$$

corresponding to the unit eigenvalue $\lambda_1 = 1$ is found by first solving the linear system $(T - \mathrm{I}\,)\mathbf{v}^\star = 0$ and then converting the solution[†] $\mathbf{v}^\star$ into a valid probability vector $\mathbf{u}^\star$ by use of formula (9.38). According to Theorem 9.30, no matter how the taxis are initially distributed, eventually about 47% of the taxis will be in Minneapolis, 23% in St. Paul, and 30% in the suburbs. This can be confirmed by running numerical experiments. Moreover, if the owner places this fraction of the taxis in the three locations, then they will more or less remain in such proportions forever.

Remark. As noted earlier — see Proposition 9.17 — the convergence rate of the Markov chain to its steady state is governed by the size of the *subdominant eigenvalue* λ_2. The smaller $|\, \lambda_2 \,|$ is, the faster the process converges. In the taxi example, $\lambda_2 = .3$ (and $\lambda_3 = 0$), and so the convergence to steady state is fairly rapid.

A Markov process can also be viewed as a weighted digraph. Each state corresponds to a vertex. A nonzero transition probability from one state to another corresponds to a weighted directed edge between the two vertices. Note that the digraph is typically not simple, since vertices can have two edges connecting them, one representing the transition probability of getting from the first to the second, and the second edge representing the transition probability of going in the other direction. The original PageRank algorithm that underlies Google's search engine, [**64**, **52**], starts with the internet digraph, whose vertices are web pages and whose directed edges represent links from one web page to another, which are weighted according to the number of such links. To be effective, the resulting weighted internet digraph is supplemented by adding in a number of random low weight edges. One then computes the probability eigenvector associated with the resulting digraph-based Markov process, the magnitudes of whose entries, indexed by the nodes, effectively rank the corresponding web pages.

Proof of Theorem 9.30: We begin the proof by replacing T by its transpose[‡] $M = T^T$, keeping in mind that every eigenvalue of T is also an eigenvalue of M albeit with different eigenvectors, cf. Proposition 8.12. The conditions (9.40) tell us that the matrix M has entries $0 \leq m_{ij} = t_{ji} \leq 1$, and, moreover, the *row sums* $s_i = \sum_{i=1}^n m_{ij} = 1$ of M, being the same as the corresponding column sums of T, are all equal to 1. Since $M^k = (T^k)^T$, regularity of T implies that some power M^k has all positive entries.

According to Exercise 1.2.29, if $\mathbf{z} = (\,1, \ldots, 1\,)^T$ is the column vector all of whose entries are equal to 1, then the entries of $M\mathbf{z}$ are the row sums of M. Therefore, $M\mathbf{z} = \mathbf{z}$, which implies that $\mathbf{z}$ is an eigenvector of M with eigenvalue $\lambda_1 = 1$. As a consequence, T also has

[†] Theorem 9.30 guarantees that there is an eigenvector $\mathbf{v}$ with all non-negative entries.

[‡] We apologize for the unfortunate clash of notation when writing the transpose of the matrix T.

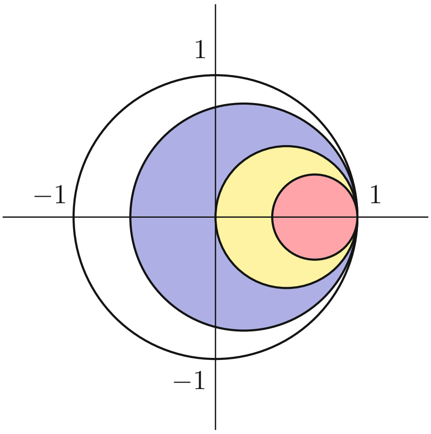


Figure 9.5. Gershgorin Disks for a Regular Transition Matrix.

1 as an eigenvalue. Observe that $\mathbf{z}$ is *not* in general an eigenvector of T; indeed, it satisfies the *co-eigenvector equation* $M\,\mathbf{z} = T^T\mathbf{z} = \mathbf{z}$.

We claim that $\lambda_1 = 1$ is a simple eigenvalue. To this end, we prove that $\mathbf{z}$ spans the one-dimensional eigenspace V_1. In other words, we need to show that if $M\,\mathbf{v} = \mathbf{v}$, then its entries $v_1 = \cdots = v_n = a$ are all equal, and so $\mathbf{v} = a\,\mathbf{z}$ is a scalar multiple of the known eigenvector $\mathbf{z}$. Let us first prove this assuming that all of the entries of M are strictly positive, and so $0 < m_{ij} = t_{ji} < 1$ for all i, j. Suppose $\mathbf{v}$ is an eigenvector with not all equal entries. Let v_k be the minimal entry of $\mathbf{v}$, so $v_k \leq v_i$ for all $i \neq k$, and at least one inequality is strict, say $v_k < v_j$. Then the k^{th} entry of the eigenvector equation $\mathbf{v} = M\,\mathbf{v}$ is

$$v_k = \sum_{j=1}^{n} m_{kj}\, v_j \ > \left(\sum_{j=1}^{n} m_{kj} \right) v_k = v_k,$$

where the strict inequality follows from the assumed positivity of the entries of M, and the final equality follows from the fact that M has unit row sums. Thus, we are led to a contradiction, and the claim follows. If M has one or more 0 entries, but M^k has all positive entries, then we apply the previous argument to the equation $M^k\mathbf{v} = \mathbf{v}$ which follows from $M\,\mathbf{v} = \mathbf{v}$. If $\lambda_1 = 1$ is a complete eigenvalue, then we are finished. The proof that this is indeed the case is a bit technical, and we refer the reader to [**4**] for the details.

Finally, let us prove that all the other eigenvalues of M are less than 1 in modulus. For this we appeal to the Gershgorin Circle Theorem 8.16. Suppose M^k has all positive entries, denoted by $m_{ij}^{(k)} > 0$. Its Gershgorin disk D_i is centered at $m_{ii}^{(k)} > 0$ and has radius $r_i = 1 - m_{ii}^{(k)} < 1$ since the i^{th} row sum of M^k equals 1. Thus the disk lies strictly inside the open unit disk $|\,z\,| < 1$ *except* for a single boundary point at $z = 1$; see Figure 9.5. The Circle Theorem 8.16 implies that all eigenvalues of M^k except the unit eigenvalue $\lambda_1 = 1$ must lie strictly inside the unit disk. Since these are just the k^{th} powers of the eigenvalues of M, the same holds for the eigenvalues themselves, so $|\,\lambda_j\,| < 1$ for $j \geq 2$.

Therefore, the matrix M, and, hence, also T, satisfies the hypotheses of Proposition 9.17. We conclude that the iterates $\mathbf{u}^{(k)} = T^k\mathbf{u}^{(0)} \to \mathbf{u}^\star$ converge to a multiple of the probability eigenvector of T. If the initial condition $\mathbf{u}^{(0)}$ is a probability vector, then so is every subsequent state vector $\mathbf{u}^{(k)}$, and so their limit $\mathbf{u}^\star$ must also be a probability vector. This completes the proof of the theorem. *Q.E.D.*

Exercises

9.3.1. Determine if the following matrices are regular transition matrices. If so, find the associated probability eigenvector. (*a*) $\begin{pmatrix} \frac{1}{2} & \frac{1}{3} \\ \frac{3}{4} & \frac{2}{3} \end{pmatrix}$, (*b*) $\begin{pmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix}$, (*c*) $\begin{pmatrix} \frac{1}{4} & \frac{2}{3} \\ \frac{3}{4} & \frac{1}{3} \end{pmatrix}$,

(*d*) $\begin{pmatrix} 0 & \frac{1}{5} \\ 1 & \frac{4}{5} \end{pmatrix}$, (*e*) $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, (*f*) $\begin{pmatrix} .3 & .5 & .2 \\ .3 & .2 & .5 \\ .4 & .3 & .3 \end{pmatrix}$, (*g*) $\begin{pmatrix} .1 & .5 & .4 \\ .6 & .1 & .3 \\ .3 & 0 & .7 \end{pmatrix}$,

(*h*) $\begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & \frac{1}{3} \end{pmatrix}$, (*i*) $\begin{pmatrix} 0 & .2 & 0 & 1 \\ .5 & 0 & .3 & 0 \\ 0 & .8 & 0 & 0 \\ .5 & 0 & .7 & 0 \end{pmatrix}$, (*j*) $\begin{pmatrix} .1 & .2 & .3 & .4 \\ .2 & .5 & .3 & .1 \\ .3 & .3 & .1 & .3 \\ .4 & .1 & .3 & .2 \end{pmatrix}$, (*k*) $\begin{pmatrix} 0 & .6 & 0 & .4 \\ .5 & 0 & .3 & .1 \\ 0 & .4 & 0 & .5 \\ .5 & 0 & .7 & 0 \end{pmatrix}$.

9.3.2. A business executive is managing three branches, labeled A, B, and C, of a corporation. She never visits the same branch on consecutive days. If she visits branch A one day, she visits branch B the next day. If she visits either branch B or C that day, then the next day she is twice as likely to visit branch A as to visit branch B or C. Explain why the resulting transition matrix is regular. Which branch does she visit the most often in the long run?

9.3.3. A study has determined that, on average, a man's occupation depends on that of his father. If the father is a farmer, there is a 30% chance that the son will be a blue collar laborer, a 30% chance he will be a white collar professional, and a 40% chance he will also be a farmer. If the father is a laborer, there is a 30% chance that the son will also be one, a 60% chance he will be a professional, and a 10% chance he will be a farmer. If the father is a professional, there is a 70% chance that the son will also be one, a 25% chance he will be a laborer, and a 5% chance he will be a farmer. (*a*) What is the probability that the grandson of a farmer will also be a farmer? (*b*) In the long run, what proportion of the male population will be farmers?

9.3.4. The population of an island is divided into city and country residents. Each year, 5% of the residents of the city move to the country and 15% of the residents of the country move to the city. In 2003, 35,000 people live in the city and 25,000 in the country. Assuming no growth in the population, how many people will live in the city and how many will live in the country between the years 2004 and 2008? What is the eventual population distribution of the island?

9.3.5. A certain plant species has either red, pink, or white flowers, depending on its genotype. If you cross a pink plant with any other plant, the probability distribution of the offspring is prescribed by the transition matrix $T = \begin{pmatrix} .5 & .25 & 0 \\ .5 & .5 & .5 \\ 0 & .25 & .5 \end{pmatrix}$. On average, if you continue crossing with only pink plants, what percentage of the three types of flowers would you expect to see in your garden?

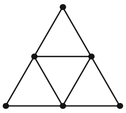
9.3.6. A genetic model describing inbreeding, in which mating takes place only between individuals of the same genotype, is given by the Markov process $\mathbf{u}^{(n+1)} = T\,\mathbf{u}^{(n)}$, where $T = \begin{pmatrix} 1 & \frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{4} & 1 \end{pmatrix}$ is the transition matrix and $\mathbf{u}^{(n)} = \begin{pmatrix} p_n \\ q_n \\ r_n \end{pmatrix}$, whose entries are, respectively, the proportion of populations of genotype AA, Aa, aa in the n^{th} generation. Find the solution to this Markov process and analyze your result.

9.3.7. A student has the habit that if she doesn't study one night, she is 70% certain of studying the next night. Furthermore, the probability that she studies two nights in a row is 50%. How often does she study in the long run?

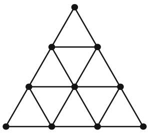
9.3.8. A traveling salesman visits the three cities of Atlanta, Boston, and Chicago. The matrix $\begin{pmatrix} 0 & .5 & .5 \\ 1 & 0 & .5 \\ 0 & .5 & 0 \end{pmatrix}$ describes the transition probabilities of his trips. Describe his travels in words, and calculate how often he visits each city on average.

9.3.9. Explain why the irregular Markov process with transition matrix $T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ does not reach a steady state. Use a population model, as in Exercise 9.3.4, to interpret what is going on.

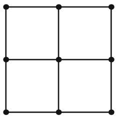
9.3.10. A bug crawls along the edges of the pictured triangular lattice with six vertices. Upon arriving at a vertex, there is an equal probability of its choosing any edge to leave the vertex. Set up the Markov chain described by the bug's motion, and determine how often, on average, it visits each vertex.



9.3.11. Answer Exercise 9.3.10 for the larger triangular lattice.



9.3.12. Suppose the bug of Exercise 9.3.10 crawls along the edges of the pictured square lattice. What can you say about its behavior?



♢ 9.3.13. Let T be a regular transition matrix with probability eigenvector $\mathbf{v}$.
(*a*) Prove that $\lim_{k\to\infty} T^k = P = (\,\mathbf{v}\ \mathbf{v}\ \ldots\ \mathbf{v}\,)$ is a matrix with every column equal to $\mathbf{v}$.
(*b*) Explain why $(\,\mathbf{v}\ \mathbf{v}\ \ldots\ \mathbf{v}\,)\,\mathbf{v} = \mathbf{v}$. (*c*) Prove directly that P is idempotent: $P^2 = P$.

9.3.14. Find $\lim_{k\to\infty} T^k$ when $T = \begin{pmatrix} .8 & .1 & .1 \\ .1 & .8 & .1 \\ .1 & .1 & .8 \end{pmatrix}$.

9.3.15. Prove that, for all $0 \le p, q \le 1$ with $p + q > 0$, the probability eigenvector of the transition matrix $T = \begin{pmatrix} 1-p & q \\ p & 1-q \end{pmatrix}$ is $\mathbf{v} = \left(\dfrac{q}{p+q}, \dfrac{p}{p+q} \right)^T$.

9.3.16. Describe the final state of a Markov chain with symmetric transition matrix $T = T^T$.

9.3.17. *True or false*: If T and T^T are both transition matrices, then $T = T^T$.

9.3.18. *True or false*: If T is a transition matrix, so is T^{-1}.

9.3.19. A transition matrix is called *doubly stochastic* if both its row and column sums are equal to 1. What is the limiting probability state of a Markov chain with doubly stochastic transition matrix?

9.3.20. *True or false*: The set of all probability vectors forms a subspace of $\mathbb{R}^n$.

9.3.21. *Multiple choice*: Every probability vector in $\mathbb{R}^n$ lies on the unit sphere for the
(*a*) 1 norm, (*b*) 2 norm, (*c*) ∞ norm, (*d*) all of the above, (*e*) none of the above.

9.3.22. *True or false*: Every probability eigenvector of a regular transition matrix has eigenvalue equal to 1.

9.3.23. Write down an example of (*a*) an irregular transition matrix; (*b*) a regular transition matrix that has one or more zero entries.

♢ 9.3.24. Let T be a transition matrix. Prove that if $\mathbf{u}$ is a probability vector, then so is $\mathbf{v} = T\mathbf{u}$.

♢ 9.3.25. (*a*) Prove that if T and S are transition matrices, then so is their product $T\,S$.
(*b*) Prove that if T is a transition matrix, then so is T^k for all $k \ge 0$.

9.4 Iterative Solution of Linear Algebraic Systems

In this section, we return to the most basic problem in linear algebra: solving the linear algebraic system

$$A\mathbf{u} = \mathbf{b}, \tag{9.41}$$

consisting of n equations in n unknowns. We assume that the $n \times n$ coefficient matrix A is nonsingular, and so the solution $\mathbf{u} = A^{-1}\mathbf{b}$ is unique. For simplicity, we shall only consider real systems here.

We will introduce several popular iterative methods that can be used to approximate the solution for certain classes of coefficient matrices. The resulting algorithms will provide an attractive alternative to Gaussian Elimination, particularly when one is dealing with the large, sparse systems that arise in the numerical solution to differential equations. One major advantage of an iterative technique is that, in favorable situations, it produces progressively more and more accurate approximations to the solution, and hence, by prolonging the iterations, can, at least in principle, compute the solution to any desired order of accuracy. Moreover, even performing just a few iterations may produce a reasonable approximation to the true solution — in stark contrast to Gaussian Elimination, where one must continue the process through to the bitter end before any useful information can be extracted. A partially completed Gaussian Elimination is of scant use! A significant weakness is that iterative methods are not universally applicable, and their design relies upon the detailed structure of the coefficient matrix.

We shall be attempting to solve the linear system (9.41) by replacing it with an iterative system of the form

$$\mathbf{u}^{(k+1)} = T\mathbf{u}^{(k)} + \mathbf{c}, \qquad \mathbf{u}^{(0)} = \mathbf{u}_0, \tag{9.42}$$

in which T is an $n \times n$ matrix and $\mathbf{c} \in \mathbb{R}^n$. This represents a slight generalization of our earlier iterative system (9.1), in that the right-hand side is now an affine function of $\mathbf{u}^{(k)}$. Suppose that the solutions to the affine iterative system converge: $\mathbf{u}^{(k)} \to \mathbf{u}^\star$ as $k \to \infty$. Then, by taking the limit of both sides of (9.42), we discover that the limit point $\mathbf{u}^\star$ solves the *fixed-point equation*

$$\mathbf{u}^\star = T\mathbf{u}^\star + \mathbf{c}. \tag{9.43}$$

Thus, we need to design our iterative system so that

(*a*) the solution to the fixed-point system $\mathbf{u} = T\mathbf{u} + \mathbf{c}$ coincides with the solution to the original system $A\mathbf{u} = \mathbf{b}$, and

(*b*) the iterates defined by (9.42) are known to converge to the fixed point. The more rapid the convergence, the better.

Before exploring these issues in depth, let us look at a simple example.

Example 9.33. Consider the linear system

$$3x + y - z = 3, \qquad x - 4y + 2z = -1, \qquad -2x - y + 5z = 2, \tag{9.44}$$

which has the vectorial form $A\mathbf{u} = \mathbf{b}$, with

$$A = \begin{pmatrix} 3 & 1 & -1 \\ 1 & -4 & 2 \\ -2 & -1 & 5 \end{pmatrix}, \qquad \mathbf{u} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 3 \\ -1 \\ 2 \end{pmatrix}.$$

One easy way to convert a linear system into a fixed-point form is to rewrite it as

$$\mathbf{u} = \mathrm{I}\,\mathbf{u} - A\mathbf{u} + A\mathbf{u} = (\mathrm{I} - A)\mathbf{u} + \mathbf{b} = T\mathbf{u} + \mathbf{c}, \qquad \text{where} \qquad T = \mathrm{I} - A, \qquad \mathbf{c} = \mathbf{b}.$$

k	$\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)} + \mathbf{b}$			$\mathbf{u}^{(k+1)} = \widehat{T}\,\mathbf{u}^{(k)} + \widehat{\mathbf{c}}$		
0	0	0	0	0	0	0
1	3	−1	2	1	.25	.4
2	0	−13	−1	1.05	.7	.85
3	15	−64	−7	1.05	.9375	.96
4	30	−322	−4	1.0075	.9925	1.0075
5	261	−1633	−244	1.005	1.00562	1.0015
6	870	−7939	−133	.9986	1.002	1.0031
7	6069	−40300	−5665	1.0004	1.0012	.9999
8	22500	−196240	−5500	.9995	1.0000	1.0004
9	145743	−992701	−129238	1.0001	1.0001	.9998
10	571980	−4850773	−184261	.9999	.9999	1.0001
11	3522555	−24457324	−2969767	1.0000	1.0000	1.0000

In the present case,

$$T = \mathrm{I} - A = \begin{pmatrix} -2 & -1 & 1 \\ -1 & 5 & -2 \\ 2 & 1 & -4 \end{pmatrix}, \qquad \mathbf{c} = \mathbf{b} = \begin{pmatrix} 3 \\ -1 \\ 2 \end{pmatrix}.$$

The resulting iterative system $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)} + \mathbf{c}$ has the explicit form

$$\begin{aligned} x^{(k+1)} &= -2\,x^{(k)} - y^{(k)} + z^{(k)} + 3, \\ y^{(k+1)} &= \;\; - x^{(k)} + 5\,y^{(k)} - 2\,z^{(k)} - 1, \\ z^{(k+1)} &= \;\;\; 2\,x^{(k)} + y^{(k)} - 4\,z^{(k)} + 2. \end{aligned} \tag{9.45}$$

Another possibility is to solve the first equation in (9.44) for x, the second for y, and the third for z, so that

$$x = -\tfrac{1}{3}\,y + \tfrac{1}{3}\,z + 1, \qquad y = \tfrac{1}{4}\,x + \tfrac{1}{2}\,z + \tfrac{1}{4}\,, \qquad z = \tfrac{2}{5}\,x + \tfrac{1}{5}\,y + \tfrac{2}{5}\,.$$

The resulting equations have the form of a fixed-point system

$$\mathbf{u} = \widehat{T}\,\mathbf{u} + \widehat{\mathbf{c}}, \qquad \text{in which} \qquad \widehat{T} = \begin{pmatrix} 0 & -\frac{1}{3} & \frac{1}{3} \\ \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{2}{5} & \frac{1}{5} & 0 \end{pmatrix}, \qquad \widehat{\mathbf{c}} = \begin{pmatrix} 1 \\ \frac{1}{4} \\ \frac{2}{5} \end{pmatrix}.$$

The corresponding iterative system $\mathbf{u}^{(k+1)} = \widehat{T}\,\mathbf{u}^{(k)} + \widehat{\mathbf{c}}$ is

$$\begin{aligned} x^{(k+1)} &= -\tfrac{1}{3}\,y^{(k)} + \tfrac{1}{3}\,z^{(k)} + 1, \\ y^{(k+1)} &= \;\; \tfrac{1}{4}\,x^{(k)} + \tfrac{1}{2}\,z^{(k)} + \tfrac{1}{4}, \\ z^{(k+1)} &= \;\; \tfrac{2}{5}\,x^{(k)} + \tfrac{1}{5}\,y^{(k)} + \tfrac{2}{5}. \end{aligned} \tag{9.46}$$

Do the resulting iterative methods converge to the solution $x = y = z = 1$, i.e., to $\mathbf{u}^\star = (\,1,1,1\,)^T$? The results, starting with initial guess $\mathbf{u}^{(0)} = (\,0,0,0\,)^T$, are tabulated in the accompanying table.

For the first method, the answer is clearly no — the iterates become wilder and wilder. Indeed, this occurs no matter how close the initial guess $\mathbf{u}^{(0)}$ is to the actual solution — unless $\mathbf{u}^{(0)}$ happens to be exactly equal to $\mathbf{u}^\star$. In the second case, the iterates do converge to the solution, and it does not take too long, even starting from a poor initial guess, to obtain a reasonably accurate approximation. Of course, in such a simple example, it would be silly to use iteration, when Gaussian Elimination can be done by hand and produces the solution almost immediately. However, we use the small examples for illustrative purposes, in order to prepare us to bring the full power of iterative algorithms to bear on the large linear systems arising in applications.

The convergence of solutions to (9.42) to the fixed point $\mathbf{u}^\star$ is based on the behavior of the *error vectors*

$$\mathbf{e}^{(k)} = \mathbf{u}^{(k)} - \mathbf{u}^\star, \tag{9.47}$$

which measure how close the iterates are to the true solution. Let us find out how the successive error vectors are related. We compute

$$\mathbf{e}^{(k+1)} = \mathbf{u}^{(k+1)} - \mathbf{u}^\star = (T\,\mathbf{u}^{(k)} + \mathbf{a}) - (T\,\mathbf{u}^\star + \mathbf{a}) = T(\mathbf{u}^{(k)} - \mathbf{u}^\star) = T\,\mathbf{e}^{(k)},$$

showing that the error vectors satisfy a *linear* iterative system

$$\mathbf{e}^{(k+1)} = T\,\mathbf{e}^{(k)}, \tag{9.48}$$

with the *same* coefficient matrix T. Therefore, they are given by the explicit formula

$$\mathbf{e}^{(k)} = T^k\,\mathbf{e}^{(0)}.$$

Now, the solutions to (9.42) converge to the fixed point, $\mathbf{u}^{(k)} \to \mathbf{u}^\star$, if and only if the error vectors converge to zero: $\mathbf{e}^{(k)} \to \mathbf{0}$ as $k \to \infty$. Our analysis of linear iterative systems, as summarized in Theorem 9.11, establishes the following basic convergence result.

Proposition 9.34. The solutions to the affine iterative system (9.42) will all converge to the solution to the fixed point equation (9.43) if and only if T is a convergent matrix, or, equivalently, its spectral radius satisfies $\rho(T) < 1$.

The spectral radius $\rho(T)$ of the coefficient matrix will govern the speed of convergence. Therefore, our goal is to construct an iterative system whose coefficient matrix has as small a spectral radius as possible. At the very least, the spectral radius must be less than 1. For the two iterative systems presented in Example 9.33, the spectral radii of the coefficient matrices are found to be

$$\rho(T) \simeq 4.9675, \qquad \rho(\,\widehat{T}\,) = .5.$$

Therefore, T is not a convergent matrix, which explains the wild behavior of its iterates, whereas $\widehat{T}$ is convergent, and one expects the error to decrease by a factor of roughly $\frac{1}{2}$ at each step, which is what is observed in practice.

The Jacobi Method

The first general iterative method for solving linear systems is based on the same simple idea used in our illustrative Example 9.33. Namely, we solve the i^{th} equation in the system $A\,\mathbf{u} = \mathbf{b}$, which is

$$\sum_{j=1}^{n} a_{ij}\,u_j = b_i,$$

for the i^{th} variable u_i. To do this, we need to assume that all the diagonal entries of A are nonzero: $a_{ii} \neq 0$. The result is

$$u_i = -\frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\, u_j + \frac{b_i}{a_{ii}} = \sum_{j=1}^{n} t_{ij}\, u_j + c_i, \tag{9.49}$$

where

$$t_{ij} = \begin{cases} -\dfrac{a_{ij}}{a_{ii}}, & i \neq j, \\ 0, & i = j, \end{cases} \qquad \text{and} \qquad c_i = \frac{b_i}{a_{ii}}. \tag{9.50}$$

The result has the form of a fixed-point system $\mathbf{u} = T\,\mathbf{u} + \mathbf{c}$, and forms the basis of the *Jacobi Method*

$$\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)} + \mathbf{c}, \qquad \mathbf{u}^{(0)} = \mathbf{u}_0, \tag{9.51}$$

named after the influential nineteenth-century German analyst Carl Jacobi. The explicit form of the Jacobi iterative algorithm is

$$u_i^{(k+1)} = -\frac{1}{a_{ii}} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\, u_j^{(k)} + \frac{b_i}{a_{ii}}. \tag{9.52}$$

It is instructive to rederive the Jacobi Method in matrix form. We begin by decomposing the coefficient matrix

$$A = L + D + U \tag{9.53}$$

into the sum of a strictly lower triangular matrix L, meaning all its diagonal entries are 0, a diagonal matrix D, and a strictly upper triangular matrix U, each of which is uniquely specified; see Exercise 1.3.11. For example, when

$$A = \begin{pmatrix} 3 & 1 & -1 \\ 1 & -4 & 2 \\ -2 & -1 & 5 \end{pmatrix}, \tag{9.54}$$

the decomposition (9.53) yields

$$L = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ -2 & -1 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 3 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & 5 \end{pmatrix}, \qquad U = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}.$$

Warning. The L, D, U in the elementary additive decomposition (9.53) have nothing to do with the L, D, U appearing in factorizations arising from Gaussian Elimination. The latter play no role in the iterative solution methods considered in this section.

We then rewrite the system

$$A\,\mathbf{u} = (L + D + U)\,\mathbf{u} = \mathbf{b} \qquad \text{in the alternative form} \qquad D\,\mathbf{u} = -\,(L + U)\,\mathbf{u} + \mathbf{b}.$$

The Jacobi fixed point system (9.49) amounts to solving the latter for

$$\mathbf{u} = T\,\mathbf{u} + \mathbf{c}, \qquad \text{where} \qquad T = -\,D^{-1}(L + U), \qquad \mathbf{c} = D^{-1}\mathbf{b}. \tag{9.55}$$

For the example (9.54), we recover the Jacobi iteration matrix

$$T = -\,D^{-1}(L + U) = \begin{pmatrix} 0 & -\frac{1}{3} & \frac{1}{3} \\ \frac{1}{4} & 0 & \frac{1}{2} \\ \frac{2}{5} & \frac{1}{5} & 0 \end{pmatrix}.$$

Deciding in advance whether the Jacobi Method will converge is not easy. However, it can be shown that Jacobi iteration *is* guaranteed to converge when the original coefficient matrix has large diagonal entries, in accordance with Definition 8.18.

Theorem 9.35. If the coefficient matrix A is strictly diagonally dominant, then the associated Jacobi iteration converges.

Proof: We shall prove that $\| T \|_\infty < 1$, and so Proposition 9.22 implies that T is a convergent matrix. The absolute row sums of the Jacobi matrix $T = -D^{-1}(L+U)$ are, according to (9.50),

$$s_i = \sum_{j=1}^{n} |\, t_{ij} \,| = \frac{1}{|\, a_{ii} \,|} \sum_{\substack{j=1 \\ j \neq i}}^{n} |\, a_{ij} \,| < 1, \tag{9.56}$$

because A is strictly diagonally dominant, and hence satisfies (8.28). This implies that $\| T \|_\infty = \max\{s_1, \ldots, s_n\} < 1$, and the result follows. *Q.E.D.*

Example 9.36. Consider the linear system

$$\begin{aligned} 4x + y + w &= 1, \\ x + 4y + z + v &= 2, \\ y + 4z + w &= -1, \\ x + z + 4w + v &= 2, \\ y + w + 4v &= 1. \end{aligned}$$

The Jacobi Method solves the respective equations for x, y, z, w, v, leading to the iterative equations

$$\begin{aligned} x^{(k+1)} &= -\tfrac{1}{4}\, y^{(k)} - \tfrac{1}{4}\, w^{(k)} + \tfrac{1}{4}, \\ y^{(k+1)} &= -\tfrac{1}{4}\, x^{(k)} - \tfrac{1}{4}\, z^{(k)} - \tfrac{1}{4}\, v^{(k)} + \tfrac{1}{2}, \\ z^{(k+1)} &= -\tfrac{1}{4}\, y^{(k)} - \tfrac{1}{4}\, w^{(k)} - \tfrac{1}{4}, \\ w^{(k+1)} &= -\tfrac{1}{4}\, x^{(k)} - \tfrac{1}{4}\, z^{(k)} - \tfrac{1}{4}\, v^{(k)} + \tfrac{1}{2}, \\ v^{(k+1)} &= -\tfrac{1}{4}\, y^{(k)} - \tfrac{1}{4}\, w^{(k)} + \tfrac{1}{4}. \end{aligned}$$

The coefficient matrix of the original system,

$$A = \begin{pmatrix} 4 & 1 & 0 & 1 & 0 \\ 1 & 4 & 1 & 0 & 1 \\ 0 & 1 & 4 & 1 & 0 \\ 1 & 0 & 1 & 4 & 1 \\ 0 & 1 & 0 & 1 & 4 \end{pmatrix},$$

is strictly diagonally dominant, and so we are guaranteed that the Jacobi iterations will eventually converge to the solution. Indeed, the Jacobi scheme takes the iterative form (9.55), with

$$T = \begin{pmatrix} 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 0 & -\frac{1}{4} & 0 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 & -\frac{1}{4} & 0 \end{pmatrix}, \qquad \mathbf{c} = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \\ -\frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{pmatrix}.$$

Note that $\| T \|_\infty = \frac{3}{4} < 1$, validating convergence. Thus, to obtain, say, four decimal place accuracy in the solution, we estimate that it will take fewer than $\log(.5 \times 10^{-4}) / \log .75 \simeq 34$ iterates, assuming a moderate initial error. But the matrix norm always underestimates the true rate of convergence, as prescribed by the spectral radius $\rho(T) = .6124$, which would imply about $\log(.5 \times 10^{-4}) / \log .6124 \simeq 20$ iterations to obtain the desired accuracy. Indeed, starting with the initial guess $x^{(0)} = y^{(0)} = z^{(0)} = w^{(0)} = v^{(0)} = 0$, the Jacobi iterates converge to the exact solution

$$x = -.1, \qquad y = .7, \qquad z = -.6, \qquad w = .7, \qquad v = -.1,$$

to within four decimal places in exactly 20 iterations.

Exercises

9.4.1. (*a*) Find the spectral radius of the matrix $T = \begin{pmatrix} 1 & 1 \\ -1 & -\frac{7}{6} \end{pmatrix}$. (*b*) Predict the long term behavior of the iterative system $\mathbf{u}^{(k+1)} = T\mathbf{u}^{(k)} + \mathbf{b}$, where $\mathbf{b} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$, in as much detail as you can.

9.4.2. Answer Exercise 9.4.1 when (*a*) $T = \begin{pmatrix} 1 & -\frac{1}{2} \\ -1 & \frac{3}{2} \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$;
(*b*) $T = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} \\ 1 & 1 & \frac{1}{4} \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}$; (*c*) $T = \begin{pmatrix} -.05 & .15 & .15 \\ .35 & .15 & -.35 \\ -.2 & -.2 & .3 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} -1.5 \\ 1.6 \\ 1.7 \end{pmatrix}$.

9.4.3. Which of the following systems have a strictly diagonally dominant coefficient matrix?
(*a*) $\begin{aligned} 5x - y &= 1, \\ -x + 3y &= -1; \end{aligned}$ (*b*) $\begin{aligned} \tfrac{1}{2}x + \tfrac{1}{3}y &= 1, \\ \tfrac{1}{5}x + \tfrac{1}{4}y &= 6; \end{aligned}$ (*c*) $\begin{aligned} -5x + y &= 3, \\ -3x + 2y &= -2; \end{aligned}$ (*d*) $\begin{aligned} -2x + y + z &= 1, \\ -x + 2y - z &= -2, \\ x - y + 3z &= 1; \end{aligned}$
(*e*) $\begin{aligned} -x + \tfrac{1}{2}y + \tfrac{1}{3}z &= 1, \\ \tfrac{1}{3}x + 2y + \tfrac{3}{4}z &= -3, \\ \tfrac{2}{3}x + \tfrac{1}{4}y - \tfrac{3}{2}z &= 2; \end{aligned}$ (*f*) $\begin{aligned} x - 2y + z &= 1, \\ -x + 2y + z &= -1, \\ x + 3y - 2z &= 3; \end{aligned}$ (*g*) $\begin{aligned} -4x + 2y + z &= 2, \\ -x + 3y + z &= -1, \\ x + 4y - 6z &= 3. \end{aligned}$

♠ 9.4.4. For the strictly diagonally dominant systems in Exercise 9.4.3, starting with the initial guess $x = y = z = 0$, compute the solution to 2 decimal places using the Jacobi Method. Check your answer by solving the system directly by Gaussian Elimination.

♠ 9.4.5. (*a*) Do any of the non-strictly diagonally dominant systems in Exercise 9.4.3 lead to convergent Jacobi algorithms? *Hint*: Check the spectral radius of the Jacobi matrix. (*b*) For the convergent systems in Exercise 9.4.3, starting with the initial guess $x = y = z = 0$, compute the solution to 2 decimal places by using the Jacobi Method, and check your answer by solving the system directly by Gaussian Elimination.

9.4.6. The following linear systems have positive definite coefficient matrices. Use the Jacobi Method starting with $\mathbf{u}^{(0)} = \mathbf{0}$ to find the solution to 4 decimal place accuracy.
(*a*) $\begin{pmatrix} 3 & -1 \\ -1 & 5 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, (*b*) $\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \mathbf{u} = \begin{pmatrix} -3 \\ 1 \end{pmatrix}$, (*c*) $\begin{pmatrix} 6 & -1 & -3 \\ -1 & 7 & 4 \\ -3 & 4 & 9 \end{pmatrix} \mathbf{u} = \begin{pmatrix} -1 \\ -2 \\ 7 \end{pmatrix}$,
(*d*) $\begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 5 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 1 \\ -5 \\ 0 \end{pmatrix}$, (*e*) $\begin{pmatrix} 5 & 1 & 1 & 1 \\ 1 & 5 & 1 & 1 \\ 1 & 1 & 5 & 1 \\ 1 & 1 & 1 & 5 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, (*f*) $\begin{pmatrix} 3 & 1 & 0 & -1 \\ 1 & 3 & 1 & 0 \\ 0 & 1 & 3 & 1 \\ -1 & 0 & 1 & 3 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ -1 \end{pmatrix}$.

♣ 9.4.7. Let A be the $n \times n$ tridiagonal matrix with all its diagonal entries equal to c and all 1's on the sub- and super-diagonals. (a) For which values of c is A strictly diagonally dominant? (b) For which values of c does the Jacobi iteration for $A\mathbf{u} = \mathbf{b}$ converge to the solution? What is the rate of convergence? *Hint*: Use Exercise 8.2.48. (c) Set $c = 2$ and use the Jacobi Method to solve the linear systems $K\mathbf{u} = \mathbf{e}_1$, for $n = 5, 10$, and 20. Starting with an initial guess of $\mathbf{0}$, how many Jacobi iterations does it take to obtain 3 decimal place accuracy? Does the convergence rate agree with what you computed in part (c)?

9.4.8. Prove that $\mathbf{0} \neq \mathbf{u} \in \ker A$ if and only if $\mathbf{u}$ is an eigenvector of the Jacobi iteration matrix with eigenvalue 1. What does this imply about convergence?

♢ 9.4.9. Prove that if A is a nonsingular coefficient matrix, then one can always arrange that all its diagonal entries are nonzero by suitably permuting its rows.

9.4.10. Consider the iterative system (9.42) with spectral radius $\rho(T) < 1$. Explain why it takes roughly $-1/\log_{10}\rho(T)$ iterations to produce one further decimal digit of accuracy in the solution.

9.4.11. *True or false*: If a system $A\mathbf{u} = \mathbf{b}$ has a strictly diagonally dominant coefficient matrix A, then the equivalent system obtained by applying an elementary row operation to A also has a strictly diagonally dominant coefficient matrix.

The Gauss–Seidel Method

The Gauss–Seidel Method relies on a slightly refined implementation of the Jacobi process. To understand how it works, it will help to write out the Jacobi iteration algorithm (9.51) in full detail:

$$\begin{array}{rlllllll}
u_1^{(k+1)} = & & t_{12}\,u_2^{(k)} & + t_{13}\,u_3^{(k)} & + \cdots & + t_{1,n-1}\,u_{n-1}^{(k)} & + t_{1n}\,u_n^{(k)} & + c_1, \\
u_2^{(k+1)} = & t_{21}\,u_1^{(k)} & & + t_{23}\,u_3^{(k)} & + \cdots & + t_{2,n-1}\,u_{n-1}^{(k)} & + t_{2n}\,u_n^{(k)} & + c_2, \\
u_3^{(k+1)} = & t_{31}\,u_1^{(k)} & + t_{32}\,u_2^{(k)} & & \cdots & + t_{3,n-1}\,u_{n-1}^{(k)} & + t_{3n}\,u_n^{(k)} & + c_3, \\
\vdots & \vdots & \vdots & \ddots & & & \ddots & \vdots \\
u_n^{(k+1)} = & t_{n1}\,u_1^{(k)} & + t_{n2}\,u_2^{(k)} & + t_{n3}\,u_3^{(k)} & + \cdots & + t_{n,n-1}\,u_{n-1}^{(k)} & & + c_n,
\end{array} \tag{9.57}$$

where we are explicitly noting the fact that all the diagonal entries of the coefficient matrix T vanish. Observe that we are using the entries of the current iterate $\mathbf{u}^{(k)}$ to compute *all* of the updated values of $\mathbf{u}^{(k+1)}$. Presumably, if the iterates $\mathbf{u}^{(k)}$ are converging to the solution $\mathbf{u}^\star$, then their individual entries are also converging, and so each $u_j^{(k+1)}$ should be a better approximation to $u_j^\star$ than $u_j^{(k)}$ is. Therefore, if we begin the k^{th} Jacobi iteration by computing $u_1^{(k+1)}$ using the first equation, then we are tempted to use this new and improved value to replace $u_1^{(k)}$ in each of the subsequent equations. In particular, we employ the modified equation

$$u_2^{(k+1)} = t_{21}\,u_1^{(k+1)} + t_{23}\,u_3^{(k)} + \cdots + t_{1n}\,u_n^{(k)} + c_2$$

to update the second component of our iterate. This more accurate value should then be used to update $u_3^{(k+1)}$, and so on.

The upshot of these considerations is the *Gauss–Seidel Method*

$$u_i^{(k+1)} = t_{i1}\,u_1^{(k+1)} + \cdots + t_{i,i-1}\,u_{i-1}^{(k+1)} + t_{i,i+1}\,u_{i+1}^{(k)} + \cdots + t_{in}\,u_n^{(k)} + c_i, \quad i = 1, \ldots, n, \tag{9.58}$$

named after Gauss (as usual!) and the German astronomer/mathematician Philipp von Seidel. At the k^{th} stage of the iteration, we use (9.58) to compute the revised entries $u_1^{(k+1)}, u_2^{(k+1)}, \ldots, u_n^{(k+1)}$ in their numerical order. Once an entry has been updated, the new value is immediately used in all subsequent computations.

Example 9.37. For the linear system

$$3\,x + y - z = 3, \qquad x - 4\,y + 2\,z = -1, \qquad -2\,x - y + 5\,z = 2,$$

the Jacobi iteration method was given in (9.46). To construct the corresponding Gauss–Seidel algorithm we use updated values of x, y, and z as they become available. Explicitly,

$$\begin{aligned} x^{(k+1)} &= -\tfrac{1}{3}\,y^{(k)} + \tfrac{1}{3}\,z^{(k)} + 1, \\ y^{(k+1)} &= \tfrac{1}{4}\,x^{(k+1)} + \tfrac{1}{2}\,z^{(k)} + \tfrac{1}{4}, \\ z^{(k+1)} &= \tfrac{2}{5}\,x^{(k+1)} + \tfrac{1}{5}\,y^{(k+1)} + \tfrac{2}{5}. \end{aligned} \tag{9.59}$$

Starting with $\mathbf{u}^{(0)} = \mathbf{0}$, the resulting iterates are

$$\mathbf{u}^{(1)} = \begin{pmatrix} 1.0000 \\ .5000 \\ .9000 \end{pmatrix}, \qquad \mathbf{u}^{(2)} = \begin{pmatrix} 1.1333 \\ .9833 \\ 1.0500 \end{pmatrix}, \qquad \mathbf{u}^{(3)} = \begin{pmatrix} 1.0222 \\ 1.0306 \\ 1.0150 \end{pmatrix}, \qquad \mathbf{u}^{(4)} = \begin{pmatrix} .9948 \\ 1.0062 \\ .9992 \end{pmatrix},$$

$$\mathbf{u}^{(5)} = \begin{pmatrix} .9977 \\ .9990 \\ .9989 \end{pmatrix}, \qquad \mathbf{u}^{(6)} = \begin{pmatrix} 1.0000 \\ .9994 \\ .9999 \end{pmatrix}, \qquad \mathbf{u}^{(7)} = \begin{pmatrix} 1.0001 \\ 1.0000 \\ 1.0001 \end{pmatrix}, \qquad \mathbf{u}^{(8)} = \begin{pmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \end{pmatrix},$$

and have converged to the solution, to 4 decimal place accuracy, after only 8 iterations — as opposed to the 11 iterations required by the Jacobi Method.

Gauss–Seidel iteration is particularly suited to implementation on a serial computer, since one can immediately replace each component $u_i^{(k)}$ by its updated value $u_i^{(k+1)}$, thereby also saving on storage in the computer's memory. In contrast, the Jacobi Method requires us to retain all the old values $\mathbf{u}^{(k)}$ until the new approximation $\mathbf{u}^{(k+1)}$ has been computed. Moreover, Gauss–Seidel typically (although not always) converges faster than Jacobi, making it the iterative algorithm of choice for serial processors. On the other hand, with the advent of parallel processing machines, variants of the parallelizable Jacobi scheme have been making a comeback.

What is Gauss–Seidel really up to? Let us rewrite the basic iterative equation (9.58) by multiplying by a_{ii} and moving the terms involving $\mathbf{u}^{(k+1)}$ to the left-hand side. In view of the formula (9.50) for the entries of T, the resulting equation is

$$a_{i1}\,u_1^{(k+1)} + \;\cdots\; + a_{i,i-1}\,u_{i-1}^{(k+1)} + a_{ii}\,u_i^{(k+1)} = -\,a_{i,i+1}\,u_{i+1}^{(k)} - \;\cdots\; - a_{in}\,u_n^{(k)} + b_i.$$

In matrix form, taking (9.53) into account, this reads

$$(L + D)\mathbf{u}^{(k+1)} = -\,U\,\mathbf{u}^{(k)} + \mathbf{b}, \tag{9.60}$$

and so can be viewed as a linear system of equations for $\mathbf{u}^{(k+1)}$ with lower triangular coefficient matrix $L + D$. Note that the fixed point of (9.60), namely the solution to

$$(L + D)\,\mathbf{u} = -\,U\,\mathbf{u} + \mathbf{b},$$

coincides with the solution to the original system

$$A\,\mathbf{u} = (L + D + U)\,\mathbf{u} = \mathbf{b}.$$

In other words, the Gauss–Seidel procedure is merely implementing Forward Substitution to solve the lower triangular system (9.60) for the next iterate:

$$\mathbf{u}^{(k+1)} = -\,(L+D)^{-1} U\, \mathbf{u}^{(k)} + (L+D)^{-1}\, \mathbf{b}.$$

The latter is in our more usual iterative form

$$\mathbf{u}^{(k+1)} = \widetilde{T}\, \mathbf{u}^{(k)} + \widetilde{\mathbf{c}}, \qquad \text{where} \qquad \widetilde{T} = -\,(L+D)^{-1} U, \qquad \widetilde{\mathbf{c}} = (L+D)^{-1}\, \mathbf{b}. \tag{9.61}$$

Consequently, the convergence of the Gauss–Seidel iterates is governed by the spectral radius of their coefficient matrix $\widetilde{T}$.

Returning to Example 9.37, we have

$$A = \begin{pmatrix} 3 & 1 & -1 \\ 1 & -4 & 2 \\ -2 & -1 & 5 \end{pmatrix}, \qquad L+D = \begin{pmatrix} 3 & 0 & 0 \\ 1 & -4 & 0 \\ -2 & -1 & 5 \end{pmatrix}, \qquad U = \begin{pmatrix} 0 & 1 & -1 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{pmatrix}.$$

Therefore, the Gauss–Seidel matrix is

$$\widetilde{T} = -\,(L+D)^{-1} U = \begin{pmatrix} 0 & -.3333 & .3333 \\ 0 & -.0833 & .5833 \\ 0 & -.1500 & .2500 \end{pmatrix}.$$

Its eigenvalues are 0 and $.0833 \pm .2444\, \mathrm{i}$, and hence its spectral radius is $\rho(\widetilde{T}\,) \simeq .2582$. This is roughly the square of the Jacobi spectral radius of .5, which tells us that the Gauss–Seidel iterations will converge about twice as fast to the solution. This can be verified by more extensive computations. Although examples can be constructed in which the Jacobi Method converges faster, in many practical situations Gauss–Seidel tends to converge roughly twice as fast as Jacobi.

Completely general conditions guaranteeing convergence of the Gauss–Seidel Method are also hard to establish. But, like the Jacobi Method, it is guaranteed to converge when the original coefficient matrix is strictly diagonally dominant.

Theorem 9.38. If A is strictly diagonally dominant, then the Gauss–Seidel iteration algorithm for solving $A\mathbf{u} = \mathbf{b}$ converges.

Proof: Let $\mathbf{e}^{(k)} = \mathbf{u}^{(k)} - \mathbf{u}^{\star}$ denote the k^{th} Gauss–Seidel error vector. As in (9.48), the error vectors satisfy the linear iterative system $\mathbf{e}^{(k+1)} = \widetilde{T}\mathbf{e}^{(k)}$, but a direct estimate of $\|\,\widetilde{T}\,\|_\infty$ is not so easy. Instead, let us write out the linear iterative system in components:

$$e_i^{(k+1)} = t_{i1}\, e_1^{(k+1)} + \;\cdots\; + t_{i,i-1}\, e_{i-1}^{(k+1)} + t_{i,i+1}\, e_{i+1}^{(k)} + \;\cdots\; + t_{in}\, e_n^{(k)}. \tag{9.62}$$

Let

$$m^{(k)} = \|\,\mathbf{e}^{(k)}\,\|_\infty = \max\bigl\{\,|\,e_1^{(k)}\,|,\;\ldots\;,|\,e_n^{(k)}\,|\,\bigr\} \tag{9.63}$$

denote the ∞ norm of the k^{th} error vector. To prove convergence, $\mathbf{e}^{(k)} \to \mathbf{0}$, it suffices to show that $m^{(k)} \to 0$ as $k \to \infty$. We claim that diagonal dominance of A implies that

$$m^{(k+1)} \le s\, m^{(k)}, \qquad \text{where} \qquad s = \|\, T \,\|_\infty < 1 \tag{9.64}$$

denotes the ∞ matrix norm of the *Jacobi* matrix T — not the Gauss–Seidel matrix $\widetilde{T}$ — which, by (9.56), is less than 1. We infer that $m^{(k)} \le s^k\, m^{(0)} \to 0$ as $k \to \infty$, demonstrating the theorem.

To prove (9.64), we use induction on $i = 1, \ldots, n$. Our induction hypothesis is

$$|\, e_j^{(k+1)} \,| \le s\, m^{(k)} < m^{(k)} \qquad \text{for} \qquad j = 1, \ldots, i-1.$$

(When $i = 1$, there is no assumption.) Moreover, by (9.63),

$$|\, e_j^{(k)} \,| \le m^{(k)} \qquad \text{for all} \qquad j = 1, \ldots, n.$$

We use these two inequalities to estimate $|\, e_i^{(k+1)} \,|$ from (9.62):

$$\begin{aligned} |\, e_i^{(k+1)} \,| &\le |\, t_{i1} \,|\, |\, e_1^{(k+1)} \,| + \cdots + |\, t_{i,i-1} \,|\, |\, e_{i-1}^{(k+1)} \,| + |\, t_{i,i+1} \,|\, |\, e_{i+1}^{(k)} \,| + \cdots + |\, t_{in} \,|\, |\, e_n^{(k)} \,| \\ &\le \bigl(\, |\, t_{i1} \,| + \cdots + |\, t_{in} \,| \,\bigr)\, m^{(k)} \le s\, m^{(k)}, \end{aligned}$$

which completes the induction step. As a result, the maximum

$$m^{(k+1)} = \max\bigl\{\, |\, e_1^{(k+1)} \,|, \ \ldots\ , |\, e_n^{(k+1)} \,| \,\bigr\} \le s\, m^{(k)}$$

also satisfies the same bound, and hence (9.64) follows. *Q.E.D.*

Example 9.39. For the linear system considered in Example 9.36, the Gauss–Seidel iterations take the form

$$\begin{aligned} x^{(k+1)} &= -\tfrac14\, y^{(k)} - \tfrac14\, w^{(k)} + \tfrac14\,, \\ y^{(k+1)} &= -\tfrac14\, x^{(k+1)} - \tfrac14\, z^{(k)} - \tfrac14\, v^{(k)} + \tfrac12\,, \\ z^{(k+1)} &= -\tfrac14\, y^{(k+1)} - \tfrac14\, w^{(k)} - \tfrac14\,, \\ w^{(k+1)} &= -\tfrac14\, x^{(k+1)} - \tfrac14\, z^{(k+1)} - \tfrac14\, v^{(k)} + \tfrac12\,, \\ v^{(k+1)} &= -\tfrac14\, y^{(k+1)} - \tfrac14\, w^{(k+1)} + \tfrac14\,. \end{aligned}$$

Starting with $x^{(0)} = y^{(0)} = z^{(0)} = w^{(0)} = v^{(0)} = 0$, the Gauss–Seidel iterates converge to the solution $x = -.1,\ y = .7,\ z = -.6,\ w = .7,\ v = -.1$, to four decimal places in 11 iterations, again roughly twice as fast as the Jacobi Method. Indeed, the convergence rate is governed by the corresponding Gauss–Seidel matrix $\widetilde{T}$, which is

$$\begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 \\ 1 & 0 & 1 & 4 & 0 \\ 0 & 1 & 0 & 1 & 4 \end{pmatrix}^{-1} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & -.2500 & 0 & -.2500 & 0 \\ 0 & .0625 & -.2500 & .0625 & -.2500 \\ 0 & -.0156 & .0625 & -.2656 & .0625 \\ 0 & .0664 & -.0156 & .1289 & -.2656 \\ 0 & -.0322 & .0664 & -.0479 & .1289 \end{pmatrix}.$$

Its spectral radius is $\rho(\widetilde{T}) = .3936$, which is, as in the previous example, approximately the square of the spectral radius of the Jacobi coefficient matrix, which explains the speedup in convergence.

Exercises

♡ 9.4.12. Consider the linear system $A\mathbf{x} = \mathbf{b}$, where $A = \begin{pmatrix} 4 & 1 & -2 \\ -1 & 4 & -1 \\ 1 & -1 & 4 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} -2 \\ -1 \\ 7 \end{pmatrix}$.

(*a*) First, solve the equation directly by Gaussian Elimination. (*b*) Write the Jacobi iteration in the form $\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)} + \mathbf{c}$. Find the 3×3 matrix T and the vector $\mathbf{c}$ explicitly. (*c*) Using the initial approximation $\mathbf{x}^{(0)} = \mathbf{0}$, carry out three iterations of the Jacobi algorithm to compute $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ and $\mathbf{x}^{(3)}$. How close are you to the exact solution? (*d*) Write the Gauss–Seidel iteration in the form $\mathbf{x}^{(k+1)} = \widetilde{T}\mathbf{x}^{(k)} + \widetilde{\mathbf{c}}$. Find the 3×3 matrix $\widetilde{T}$ and the vector $\widetilde{\mathbf{c}}$ explicitly. (*e*) Using the initial approximation $\mathbf{x}^{(0)} = \mathbf{0}$, carry out three iterations of the Gauss–Seidel algorithm. Which is a better approximation to the solution — Jacobi or Gauss–Seidel? (*f*) Determine the spectral radius of the Jacobi matrix T, and use this to prove that the Jacobi Method will converge to the solution of $A\mathbf{x} = \mathbf{b}$ for any choice of the initial approximation $\mathbf{x}^{(0)}$. (*g*) Determine the spectral radius of the Gauss–Seidel matrix $\widetilde{T}$. Which method converges faster? (*h*) For the faster method, how many iterations would you expect to need to obtain 5 decimal place accuracy? (*i*) Test your prediction by computing the solution to the desired accuracy.

♠ 9.4.13. For the strictly diagonally dominant systems in Exercise 9.4.3, starting with the initial guess $x = y = z = 0$, compute the solution to 3 decimal places using the Gauss–Seidel Method. Check your answer by solving the system directly by Gaussian Elimination.

9.4.14. Which of the systems in Exercise 9.4.3 lead to convergent Gauss–Seidel algorithms? In each case, which converges faster, Jacobi or Gauss–Seidel?

9.4.15. (*a*) Solve the positive definite linear systems in Exercise 9.4.6 using the Gauss–Seidel Method to achieve 4 decimal place accuracy.
(*b*) Compare the convergence rate with that of the Jacobi Method.

♣ 9.4.16. Let $A = \begin{pmatrix} c & 1 & 0 & 0 \\ 1 & c & 1 & 0 \\ 0 & 1 & c & 1 \\ 0 & 0 & 1 & c \end{pmatrix}$. (*a*) For what values of c is A strictly diagonally dominant?

(*b*) Use a computer to find the smallest positive value of $c > 0$ for which Jacobi iteration converges. (*c*) Find the smallest positive value of $c > 0$ for which Gauss–Seidel iteration converges. Is your answer the same? (*d*) When they both converge, which converges faster — Jacobi or Gauss–Seidel? How much faster? Does your answer depend upon the value of c?

♠ 9.4.17. Consider the linear system

$$2.4x - .8y + .8z = 1, \qquad -.6x + 3.6y - .6z = 0, \qquad 15x + 14.4y - 3.6z = 0.$$

Show, by direct computation, that Jacobi iteration converges to the solution, but Gauss–Seidel does not.

♠ 9.4.18. Discuss convergence of Gauss–Seidel iteration for the system

$$\begin{aligned} 5x + 7y + 6z + 5w &= 23, & 6x + 8y + 10z + 9w &= 33, \\ 7x + 10y + 8z + 7w &= 32, & 5x + 7y + 9z + 10w &= 31. \end{aligned}$$

9.4.19. Let $A = \begin{pmatrix} 2 & 4 & -4 \\ 3 & 3 & 3 \\ 2 & 2 & 1 \end{pmatrix}$. Find the spectral radius of the Jacobi and Gauss–Seidel iteration matrices, and discuss their convergence.

♠ 9.4.20. Consider the linear system $H_5\mathbf{u} = \mathbf{e}_1$, where H_5 is the 5×5 Hilbert matrix. Does the Jacobi Method converge to the solution? If so, how fast? What about Gauss–Seidel?

◇ 9.4.21. How many arithmetic operations are needed to perform k steps of the Jacobi iteration? What about Gauss–Seidel? Under what conditions is Jacobi or Gauss–Seidel more efficient than Gaussian Elimination?

♣ 9.4.22. Consider the linear system $A\,\mathbf{x} = \mathbf{e}_1$ based on the 10×10 pentadiagonal matrix

$$A = \begin{pmatrix} z & -1 & 1 & 0 & & & \\ -1 & z & -1 & 1 & 0 & & \\ 1 & -1 & z & -1 & 1 & 0 & \ddots \\ 0 & 1 & -1 & z & -1 & 1 & \ddots \\ & 0 & 1 & -1 & z & -1 & \ddots \\ & & 0 & 1 & -1 & z & \ddots \\ & & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

(a) For what values of z are the Jacobi and Gauss–Seidel Methods guaranteed to converge? (b) Set $z = 4$. How many iterations are required to approximate the solution to 3 decimal places? (c) How small can $|\,z\,|$ be before the methods diverge?

♣ 9.4.23. The *naïve iterative method* for solving $A\,\mathbf{u} = \mathbf{b}$ is to rewrite it in fixed point form $\mathbf{u} = T\,\mathbf{u} + \mathbf{c}$, where $T = \mathrm{I} - A$ and $\mathbf{c} = \mathbf{b}$. (a) What conditions on the eigenvalues of A ensure convergence of the naïve method? (b) Use the Gershgorin Theorem 8.16 to prove that the naïve method converges to the solution to $\begin{pmatrix} .8 & -.1 & -.1 \\ .2 & 1.5 & -.1 \\ .2 & -.1 & 1.0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$. ($c$) Check part ($b$) by implementing the method.

Successive Over-Relaxation

As we know, the smaller the spectral radius (or matrix norm) of the coefficient matrix, the faster the convergence of the iterative algorithm. One of the goals of researchers in numerical linear algebra is to design new methods for accelerating the convergence. In his 1950 thesis, the American mathematician David Young discovered a simple modification of the Jacobi and Gauss–Seidel Methods that can, in favorable situations, lead to a dramatic speedup in the rate of convergence. The method, known as *Successive Over-Relaxation*, and often abbreviated SOR, has become the iterative method of choice in a range of modern applications, [**21**, **86**]. In this subsection, we provide a brief overview.

In practice, finding the optimal iterative algorithm to solve a given linear system is as hard as solving the system itself. Therefore, numerical analysts have relied on a few tried and true techniques for designing iterative schemes that can be used in the more common applications. Consider a linear algebraic system $A\,\mathbf{u} = \mathbf{b}$. Every decomposition of the coefficient matrix into the difference of two matrices,

$$A = M - N, \tag{9.65}$$

leads to an equivalent system of the form

$$M\,\mathbf{u} = N\,\mathbf{u} + \mathbf{b}. \tag{9.66}$$

Provided that M is nonsingular, we can rewrite the preceding system in fixed point form:

$$\mathbf{u} = M^{-1}N\,\mathbf{u} + M^{-1}\mathbf{b} = T\,\mathbf{u} + \mathbf{c}, \qquad \text{where} \qquad T = M^{-1}N, \qquad \mathbf{c} = M^{-1}\mathbf{b}.$$

Now, we are free to choose any such M, which then specifies $N = A - M$ uniquely. However, for the resulting iterative method $\mathbf{u}^{(k+1)} = T\,\mathbf{u}^{(k)} + \mathbf{c}$ to be practical we must arrange that

(a) $T = M^{-1}N$ is a convergent matrix, and
(b) M can be easily inverted.

The second requirement ensures that the iterative equations

$$M\,\mathbf{u}^{(k+1)} = N\,\mathbf{u}^{(k)} + \mathbf{b} \tag{9.67}$$

can be solved for $\mathbf{u}^{(k+1)}$ with minimal computational effort. Typically, this requires that M be either a diagonal matrix, in which case the inversion is immediate, or lower or upper triangular, in which case one employs Forward or Back Substitution to solve for $\mathbf{u}^{(k+1)}$.

With this in mind, we now introduce the SOR Method. It relies on a slight generalization of the Gauss–Seidel decomposition (9.60) of the matrix into lower triangular and strictly upper triangular parts. The starting point is to write

$$A = L + D + U = \bigl[\,L + \alpha\,D\,\bigr] - \bigl[\,(\alpha-1)\,D - U\,\bigr], \tag{9.68}$$

where $0 \neq \alpha$ is an adjustable scalar parameter. We decompose the system $A\,\mathbf{u} = \mathbf{b}$ as

$$(L + \alpha\,D)\mathbf{u} = \bigl[\,(\alpha-1)\,D - U\,\bigr]\mathbf{u} + \mathbf{b}. \tag{9.69}$$

It turns out to be slightly more convenient to divide (9.69) through by α and write the resulting iterative system in the form

$$(\omega\,L + D)\mathbf{u}^{(k+1)} = \bigl[\,(1-\omega)\,D - \omega\,U\,\bigr]\mathbf{u}^{(k)} + \omega\,\mathbf{b}, \tag{9.70}$$

where $\omega = 1/\alpha$ is called the *relaxation parameter*. Assuming, as usual, that all diagonal entries of A are nonzero, the matrix $\omega\,L + D$ is an invertible lower triangular matrix, and so we can use Forward Substitution to solve the iterative system (9.70) to recover $\mathbf{u}^{(k+1)}$. The explicit formula for its i^{th} entry is

$$\begin{aligned} u_i^{(k+1)} = \omega\,t_{i1}\,u_1^{(k+1)} + \;\cdots\; + \omega\,t_{i,i-1}\,u_{i-1}^{(k+1)} + (1-\omega)\,u_i^{(k)} \\ + \omega\,t_{i,i+1}\,u_{i+1}^{(k)} + \;\cdots\; + \omega\,t_{in}\,u_n^{(k)} + \omega\,c_i, \end{aligned} \tag{9.71}$$

where t_{ij} and c_i denote the original Jacobi values (9.50). As in the Gauss–Seidel approach, we update the entries $u_i^{(k+1)}$ in numerical order $i = 1, \ldots, n$. Thus, to obtain the SOR scheme (9.71), we merely multiply the right-hand side of the Gauss–Seidel system (9.58) by the adjustable relaxation parameter ω and append the diagonal term $(1-\omega)\,u_i^{(k)}$. In particular, if we set $\omega = 1$, then the SOR Method reduces to the Gauss–Seidel Method. Choosing $\omega < 1$ leads to an *under-relaxed* method, while $\omega > 1$, known as *over-relaxation*, is the preferred choice in most practical instances.

To analyze the SOR algorithm in detail, we rewrite (9.70) in the fixed point form

$$\mathbf{u}^{(k+1)} = T_\omega\,\mathbf{u}^{(k)} + \mathbf{c}_\omega, \tag{9.72}$$

where

$$T_\omega = (\omega\,L + D)^{-1}\bigl[\,(1-\omega)\,D - \omega\,U\,\bigr], \qquad \mathbf{c}_\omega = (\omega\,L + D)^{-1}\,\omega\,\mathbf{b}. \tag{9.73}$$

The rate of convergence is governed by the spectral radius of the matrix T_ω. The goal is to choose the relaxation parameter ω so as to make the spectral radius of T_ω as small as possible. As we will see, a clever choice of ω can result in a dramatic speedup in the convergence rate. Let us look at an elementary example.

Example 9.40. Consider the matrix $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$, which we decompose as $A = L + D + U$, where

$$L = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \qquad U = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}.$$

Jacobi iteration is based on the coefficient matrix

$$T = -D^{-1}(L+U) = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}.$$

Its spectral radius is $\rho(T) = .5$, and hence the Jacobi Method takes, on average, roughly $-1/\log_{10} .5 \simeq 3.3$ iterations to produce each new decimal place in the solution.

The SOR Method (9.70) takes the explicit form

$$\begin{pmatrix} 2 & 0 \\ -\omega & 2 \end{pmatrix} \mathbf{u}^{(k+1)} = \begin{pmatrix} 2(1-\omega) & \omega \\ 0 & 2\,(1-\omega) \end{pmatrix} \mathbf{u}^{(k)} + \omega\, \mathbf{b},$$

where Gauss–Seidel is the particular case $\omega = 1$. The SOR coefficient matrix is

$$T_\omega = \begin{pmatrix} 2 & 0 \\ -\omega & 2 \end{pmatrix}^{-1} \begin{pmatrix} 2\,(1-\omega) & \omega \\ 0 & 2\,(1-\omega) \end{pmatrix} = \begin{pmatrix} 1-\omega & \frac{1}{2}\omega \\ \frac{1}{2}\omega(1-\omega) & \frac{1}{4}\,(2-\omega)^2 \end{pmatrix}.$$

To compute the eigenvalues of T_ω, we form its characteristic equation:

$$0 = \det(T_\omega - \lambda\, \mathrm{I}\,) = \lambda^2 - \bigl(2 - 2\omega + \tfrac{1}{4}\omega^2\bigr)\lambda + (1-\omega)^2 = (\lambda + \omega - 1)^2 - \tfrac{1}{4}\,\lambda\,\omega^2. \qquad (9.74)$$

Our goal is to choose ω such that

(*a*) both eigenvalues are less than 1 in modulus, so $|\,\lambda_1\,|, |\,\lambda_2\,| < 1$. This is the minimal requirement for convergence of the method.

(*b*) the largest eigenvalue (in modulus) is as small as possible. This will give the smallest spectral radius for T_ω and hence the fastest convergence rate.

By (8.26), the product of the two eigenvalues is the determinant,

$$\lambda_1\,\lambda_2 = \det T_\omega = (1-\omega)^2.$$

If $\omega \le 0$ or $\omega \ge 2$, then $\det T_\omega \ge 1$, and hence at least one of the eigenvalues would have modulus larger than 1. Thus, in order to ensure convergence, we must require $0 < \omega < 2$. For Gauss–Seidel, at $\omega = 1$, the eigenvalues are $\lambda_1 = \frac{1}{4}$, $\lambda_2 = 0$, and the spectral radius is $\rho(T_1) = .25$. This is exactly the square of the Jacobi spectral radius, and hence the Gauss–Seidel iterates converge twice as fast; so it takes, on average, only about $-1/\log_{10} .25 \simeq 1.66$ Gauss–Seidel iterations to produce each new decimal place of accuracy. It can be shown (Exercise 9.4.32) that as ω increases above 1, the two eigenvalues move along the real axis towards each other. They coincide when

$$\omega = \omega_\star = 8 - 4\sqrt{3} \ \simeq\ 1.07, \qquad \text{at which point} \qquad \lambda_1 = \lambda_2 = \omega_\star - 1 = .07 = \rho(T_\omega),$$

which is the convergence rate of the optimal SOR Method. Each iteration produces slightly more than one new decimal place in the solution, which represents a significant improvement over the Gauss–Seidel convergence rate. It takes about twice as many Gauss–Seidel iterations (and four times as many Jacobi iterations) to produce the same accuracy as this optimal SOR Method.

Of course, in such a simple 2×2 example, it is not so surprising that we can construct the best value for the relaxation parameter by hand. Young was able to find the optimal value of the relaxation parameter for a broad class of matrices that includes most of those arising in the finite difference and finite element numerical solutions to ordinary and partial differential equations, [**61**]. For the matrices in Young's class, the Jacobi eigenvalues occur in signed pairs. If $\pm\mu$ are a pair of eigenvalues for the Jacobi Method, then the corresponding eigenvalues of the SOR iteration matrix satisfy the quadratic equation

$$(\lambda + \omega - 1)^2 = \lambda\,\omega^2\,\mu^2. \tag{9.75}$$

If $\omega = 1$, so we have standard Gauss–Seidel, then $\lambda^2 = \lambda\,\mu^2$, and so the eigenvalues are $\lambda = 0$, $\lambda = \mu^2$. The Gauss–Seidel spectral radius is therefore the square of the Jacobi spectral radius, and so (at least for matrices in the Young class) its iterates converge twice as fast. The quadratic equation (9.75) has the same properties as in the 2×2 version (9.74) (which corresponds to the case $\mu = \frac{1}{2}$), and hence the optimal value of ω will be the one at which the two roots are equal:

$$\lambda_1 = \lambda_2 = \omega - 1, \qquad \text{which occurs when} \qquad \omega = \frac{2 - 2\sqrt{1-\mu^2}}{\mu^2} = \frac{2}{1+\sqrt{1-\mu^2}}\,.$$

Therefore, if $\rho_J = \max|\,\mu\,|$ denotes the spectral radius of the Jacobi Method, then the Gauss–Seidel has spectral radius $\rho_{GS} = \rho_J^2$, while the SOR Method with optimal relaxation parameter

$$\omega_\star = \frac{2}{1+\sqrt{1-\rho_J^2}}, \qquad \text{has spectral radius} \qquad \rho_\star = \omega_\star - 1. \tag{9.76}$$

For example, if $\rho_J = .99$, which is rather slow convergence (but common for iterative numerical solution schemes for partial differential equations), then $\rho_{GS} = .9801$, which is twice as fast, but still quite slow, while SOR with $\omega_\star = 1.7527$ has $\rho_\star = .7527$, which is dramatically faster[†]. Indeed, since $\rho_\star \simeq (\rho_{GS})^{14} \simeq (\rho_J)^{28}$, it takes about 14 Gauss–Seidel (and 28 Jacobi) iterations to produce the same accuracy as one SOR step. It is amazing that such a simple idea can have such a dramatic effect.

Exercises

♡ 9.4.24. Consider the linear system $A\mathbf{u} = \mathbf{b}$, where $A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$.
(*a*) What is the solution? (*b*) Discuss the convergence of the Jacobi iteration method. (*c*) Discuss the convergence of the Gauss–Seidel iteration method. (*d*) Write down the explicit formulas for the SOR Method. (*e*) What is the optimal value of the relaxation parameter ω for this system? How much faster is the convergence as compared to the Jacobi and Gauss–Seidel Methods? (*f*) Suppose your initial guess is $\mathbf{u}^{(0)} = \mathbf{0}$. Give an estimate as to how many steps each iterative method (Jacobi, Gauss–Seidel, SOR) would require in order to approximate the solution to the system to within 5 decimal places. (*g*) Verify your answer by direct computation.

♠ 9.4.25. In Exercise 9.4.18 you were asked to solve a system by Gauss–Seidel. How much faster can you design an SOR scheme to converge? Experiment with several values of the relaxation parameter ω, and discuss what you find.

♠ 9.4.26. Investigate the three basic iterative techniques — Jacobi, Gauss–Seidel, SOR — for solving the linear system $K^\star\mathbf{u}^\star = \mathbf{f}^\star$ for the cubical circuit in Example 6.4.

[†] More precisely, since the SOR matrix is not necessarily diagonalizable, the overall convergence rate is slightly slower than the spectral radius. However, this technical detail does not affect the overall conclusion.

♣ 9.4.27. Consider the linear system

$$4x - y - z = 1, \quad -x + 4y - w = 2, \quad -x + 4z - w = 0, \quad -y - z + 4w = 1.$$

(*a*) Find the solution by using Gaussian Elimination and Back Substitution. (*b*) Using $\mathbf{0}$ as your initial guess, how many iterations are required to approximate the solution to within five decimal places using (*i*) Jacobi iteration? (*ii*) Gauss–Seidel iteration? Can you estimate the spectral radii of the relevant matrices in each case? (*c*) Try to find the solution by using the SOR Method with the parameter ω taking various values between .5 and 1.5. Which value of ω gives the fastest convergence? What is the spectral radius of the SOR matrix?

♠ 9.4.28. (*a*) Find the spectral radius of the Jacobi and Gauss–Seidel iteration matrices when $A = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$. (*b*) Is A strictly diagonally dominant? (*c*) Use (9.76) to fix the optimal value of the SOR parameter. Verify that the spectral radius of the resulting iteration matrix agrees with the second formula in (9.76). (*d*) For each iterative method, predict how many iterations are needed to solve the linear system $A\mathbf{x} = \mathbf{e}_1$ to 4 decimal places, and then verify your predictions by direct computation.

♠ 9.4.29. Change the matrix in Exercise 9.4.28 to $A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ 1 & 2 & -1 & 0 \\ 0 & 1 & 2 & -1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$, and answer the same questions. Does the SOR Method with parameter given by (9.76) speed the iterations up? Why not? Can you find a value of the SOR parameter that does?

♠ 9.4.30. Consider the linear system $A\mathbf{u} = \mathbf{e}_1$ in which A is the 8×8 tridiagonal matrix with all 2's on the main diagonal and all -1's on the sub- and super-diagonals. (*a*) Use Exercise 8.2.47 to find the spectral radius of the Jacobi iteration method to solve $A\mathbf{u} = \mathbf{b}$. Does the Jacobi Method converge? (*b*) What is the optimal value of the SOR parameter based on (9.76)? How many Jacobi iterations are needed to match the effect of a single SOR step? (*c*) Test out your conclusions by using both Jacobi and SOR to approximate the solution to 3 decimal places.

♣ 9.4.31. How much can you speed up the convergence of the iterative solution to the pentadiagonal linear system in Exercise 9.4.22 when $z = 4$ using SOR? Discuss.

♢ 9.4.32. For the matrix treated in Example 9.40, prove that (*a*) as ω increases from 1 to $8 - 4\sqrt{3}$, the two eigenvalues move towards each other, with the larger one decreasing in magnitude; (*b*) if $\omega > 8 - 4\sqrt{3}$, the eigenvalues are complex conjugates, with larger modulus than the optimal value. (*c*) Can you conclude that $\omega_\star = 8 - 4\sqrt{3}$ is the optimal value for the SOR parameter?

♣ 9.4.33. The matrix $A = \begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix}$ arises in the finite difference (and finite element) discretization of the Poisson equation on a nine point square grid. Solve the linear system $A\mathbf{u} = \mathbf{e}_5$ using (*a*) Gaussian Elimination; (*b*) Jacobi iteration; (*c*) Gauss–Seidel iteration; (*d*) SOR based on the Jacobi spectral radius.

♣ 9.4.34. The generalization of Exercise 9.4.33 to an $n \times n$ grid results in an $n^2 \times n^2$ matrix in block tridiagonal form $A = \begin{pmatrix} K & -\mathrm{I} & & & \\ -\mathrm{I} & K & -\mathrm{I} & & \\ & -\mathrm{I} & K & -\mathrm{I} & \\ & & \ddots & \ddots & \ddots \end{pmatrix}$, in which K is the tridiagonal $n \times n$ matrix with 4's on the main diagonal and -1's on the sub- and super-diagonals, while I denotes the $n \times n$ identity matrix. Use the known value of the Jacobi spectral radius $\rho_J = \cos \dfrac{\pi}{n+1}$, [**86**], to design an SOR Method to solve the linear system $A\mathbf{u} = \mathbf{f}$. Run your method on the cases $n = 5$ and $\mathbf{f} = \mathbf{e}_{13}$ and $n = 25$ and $\mathbf{f} = \mathbf{e}_{313}$ corresponding to a unit force at the center of the grid. How much faster is the convergence rate of SOR than Jacobi and Gauss–Seidel?

♡ 9.4.35. If $\mathbf{u}^{(k)}$ is an approximation to the solution to $A\mathbf{u} = \mathbf{b}$, then the *residual vector* $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{u}^{(k)}$ measures how accurately the approximation solves the system.
(*a*) Show that the Jacobi iteration can be written in the form $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + D^{-1}\mathbf{r}^{(k)}$.
(*b*) Show that the Gauss–Seidel iteration has the form $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (L + D)^{-1}\mathbf{r}^{(k)}$.
(*c*) Show that the SOR iteration has the form $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (\omega L + D)^{-1}\mathbf{r}^{(k)}$.
(*d*) If $\| \mathbf{r}^{(k)} \|$ is small, does this mean that $\mathbf{u}^{(k)}$ is close to the solution? Explain your answer and illustrate with a couple of examples.

9.4.36. Let K be a positive definite $n \times n$ matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. For what values of ε does the iterative system $\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \varepsilon\, \mathbf{r}^{(k)}$, where $\mathbf{r}^{(k)} = \mathbf{f} - K\mathbf{u}^{(k)}$ is the current residual vector, converge to the solution to the linear system $K\mathbf{u} = \mathbf{f}$? What is the optimal value of ε, and what is the convergence rate?

9.5 Numerical Computation of Eigenvalues

The importance of the eigenvalues of a square matrix in a broad range of applications is amply demonstrated in this chapter and its successor. However, finding the eigenvalues and associated eigenvectors is not such an easy task. The direct method of constructing the characteristic equation of the matrix through the determinantal formula, then solving the resulting polynomial equation for the eigenvalues, and finally producing the eigenvectors by solving the associated homogeneous linear system, is hopelessly inefficient, and fraught with numerical pitfalls. We are in need of a completely new idea if we have any hopes of designing efficient numerical approximation schemes.

In this section, we develop a few of the most basic numerical algorithms for computing eigenvalues and eigenvectors. All are iterative in nature. The most direct are based on the connections between the eigenvalues and the high powers of a matrix. A more sophisticated approach, based on the QR factorization that we learned in Section 4.3, will be presented at the end of the section. Additional computational methods for eigenvalues will appear in the following Section 9.6.

The Power Method

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

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

$$\mathbf{v}^{(k+1)} = A\,\mathbf{v}^{(k)}, \qquad \mathbf{v}^{(0)} = \mathbf{v}, \tag{9.77}$$

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

$$\mathbf{v} = c_1\,\mathbf{v}_1 + \;\cdots\; + c_n\,\mathbf{v}_n, \tag{9.78}$$

then the solution takes the explicit form given in Theorem 9.4, namely

$$\mathbf{v}^{(k)} = A^k\,\mathbf{v} = c_1\,\lambda_1^k\,\mathbf{v}_1 + \;\cdots\; + c_n\,\lambda_n^k\,\mathbf{v}_n. \tag{9.79}$$

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

$$|\,\lambda_1\,| > |\,\lambda_j\,| \qquad \text{for all} \qquad j > 1. \tag{9.80}$$

As its name implies, this eigenvalue will eventually dominate the iteration (9.79). Indeed, since

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

the first term in the iterative formula (9.79) will eventually be much larger than the rest, and so, provided $c_1 \neq 0$,

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

Therefore, the solution to the iterative system (9.77) will, almost always, end up being a multiple of the dominant eigenvector of the coefficient matrix.

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

$$\lambda_1 \;\simeq\; \frac{v_i^{(k)}}{v_i^{(k-1)}}\,, \qquad \text{provided that} \qquad v_i^{(k-1)} \neq 0. \tag{9.81}$$

Example 9.41. Consider the matrix $A = \begin{pmatrix} -1 & 2 & 2 \\ -1 & -4 & -2 \\ -3 & 9 & 7 \end{pmatrix}$. As you can check, its eigenvalues and eigenvectors are

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

Repeatedly multiplying the initial vector $\mathbf{v} = (\,1, 0, 0\,)^T$ by A results in the iterates $\mathbf{v}^{(k)} = A^k\mathbf{v}$ listed in the accompanying table. The last column indicates the ratio $\lambda^{(k)} = v_1^{(k)}/v_1^{(k-1)}$ between the first components of successive iterates. (One could equally

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

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

well use the second or third components.) The ratios are converging to the dominant eigenvalue $\lambda_1 = 3$, while the vectors $\mathbf{v}^{(k)}$ are converging to a very large multiple of the corresponding eigenvector $\mathbf{v}_1 = (1, -1, 3)^T$.

The success of the Power Method lies in the assumption that A has a unique dominant eigenvalue of maximal modulus, which, by definition, equals its spectral radius: $|\lambda_1| = \rho(A)$. The rate of convergence of the method is governed by the ratio $|\lambda_2/\lambda_1|$ between the subdominant and dominant eigenvalues. Thus, the farther the dominant eigenvalue lies away from the rest, the faster the Power Method converges. We also assumed that the initial vector $\mathbf{v}^{(0)}$ includes a nonzero multiple of the dominant eigenvector, i.e., $c_1 \neq 0$. As we do not know the eigenvectors, it is not so easy to guarantee this in advance, although one must be quite unlucky to make such a poor choice of initial vector. (Of course, the stupid choice $\mathbf{v}^{(0)} = \mathbf{0}$ is not counted.) Moreover, even if c_1 happens to be 0 initially, numerical round-off error will typically come to one's rescue, since it will almost inevitably introduce a tiny component of the eigenvector $\mathbf{v}_1$ into some iterate, and this component will eventually dominate the computation. The trick is to wait long enough for it to appear!

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

$$\mathbf{u}^{(0)} = \frac{\mathbf{v}^{(0)}}{\|\mathbf{v}^{(0)}\|}, \qquad \text{and} \qquad \mathbf{u}^{(k+1)} = \frac{A\mathbf{u}^{(k)}}{\|A\mathbf{u}^{(k)}\|}. \tag{9.82}$$

If the dominant eigenvalue is positive, $\lambda_1 > 0$, then $\mathbf{u}^{(k)} \to \mathbf{u}_1$ will converge to one of the

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

two dominant unit eigenvectors (the other is $-\mathbf{u}_1$). If $\lambda_1 < 0$, then the iterates will switch back and forth between the two eigenvectors, so $\mathbf{u}^{(k)} \simeq \pm\mathbf{u}_1$. In either case, the dominant eigenvalue λ_1 is obtained as a limiting ratio between nonzero entries of $A\mathbf{u}^{(k)}$ and $\mathbf{u}^{(k)}$. If some other sort of behavior is observed, it means that one of our assumptions is not valid; either A has more than one dominant eigenvalue of maximum modulus, e.g., it has a complex conjugate pair of eigenvalues of largest modulus, or it is not complete. In such cases, one can apply the more general long term behavior described in Exercise 9.2.8 to pin down the dominant eigenvalues.

Example 9.42. For the matrix considered in Example 9.41, starting the iterative system (9.82) with $\mathbf{u}^{(k)} = (1, 0, 0)^T$, the resulting unit vectors are tabulated above. The last column, being the ratio between the first components of $A\mathbf{u}^{(k-1)}$ and $\mathbf{u}^{(k-1)}$, again converges to the dominant eigenvalue $\lambda_1 = 3$.

Variants of the Power Method for computing the other eigenvalues of the matrix are explored in the exercises.

Remark. See Wilkinson, [**90**; Chapter 2] for the perturbation theory of eigenvalues, i.e., how they can behave under small perturbations of the matrix. Wilkinson defines a *spectral condition number* to equal the product of the norms of the matrix used to place the matrix in Jordan canonical form and its inverse. The larger the spectral condition number, the more the eigenvalues deviate under perturbation. In particular symmetric matrices have spectral condition number = 1, and so their eigenvalues are well behaved under perturbations. He also gives examples of highly ill-conditioned matrices. Similarly, in [**69**; Section 3.3], Saad defines a condition number for an individual simple eigenvalue, and proves that it is the reciprocal of the cosine of the angle between its eigenvectors and co-eigenvectors (left eigenvectors).

Exercises

♠ 9.5.1. Use the Power Method to find the dominant eigenvalue and associated eigenvector of the following matrices:

$(a)\ \begin{pmatrix} -1 & -2 \\ 3 & 4 \end{pmatrix}$, $(b)\ \begin{pmatrix} -5 & 2 \\ -3 & 0 \end{pmatrix}$, $(c)\ \begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix}$, $(d)\ \begin{pmatrix} -2 & 0 & 1 \\ -3 & -2 & 0 \\ -2 & 5 & 4 \end{pmatrix}$,

$(e)\ \begin{pmatrix} -1 & -2 & -2 \\ 1 & 2 & 5 \\ -1 & 4 & 0 \end{pmatrix}$, $(f)\ \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 1 \\ 2 & 2 & 2 \end{pmatrix}$, $(g)\ \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$, $(h)\ \begin{pmatrix} 4 & 1 & 0 & 1 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 1 & 0 & 1 & 4 \end{pmatrix}$.

♠ 9.5.2. Use the Power Method to find the largest singular value of the following matrices:

$(a)\ \begin{pmatrix} 1 & 2 \\ -1 & 3 \end{pmatrix}$, $(b)\ \begin{pmatrix} 2 & 1 & -1 \\ -2 & 3 & 1 \end{pmatrix}$, $(c)\ \begin{pmatrix} 2 & 2 & 1 & -1 \\ 1 & -2 & 0 & 1 \end{pmatrix}$, $(d)\ \begin{pmatrix} 3 & 1 & -1 \\ 1 & -2 & 2 \\ 2 & -1 & 1 \end{pmatrix}$.

♠ 9.5.3. Let T_n be the tridiagonal matrix whose diagonal entries are all equal to 2 and whose sub- and super-diagonal entries all equal 1. Use the Power Method to find the dominant eigenvalue of T_n for $n = 10, 20, 50$. Do your values agree with those in Exercise 8.2.47? How many iterations do you require to obtain 4 decimal place accuracy?

♢ 9.5.4. Prove that, for the iterative method (9.82), $\| A\mathbf{u}^{(k)} \| \to |\lambda_1|$. Assuming λ_1 is real, explain how to deduce its sign.

♢ 9.5.5. *The Inverse Power Method*. Let A be a nonsingular matrix. (a) Show that the eigenvalues of A^{-1} are the reciprocals $1/\lambda$ of the eigenvalues of A. How are the eigenvectors related? (b) Show how to use the Power Method on A^{-1} to produce the smallest (in modulus) eigenvalue of A. (c) What is the rate of convergence of the algorithm? (d) Design a practical iterative algorithm based on the (permuted) LU decomposition of A.

♠ 9.5.6. Apply the Inverse Power Method of Exercise 9.5.7 to the find the smallest eigenvalue of the matrices in Exercise 9.5.1.

♢ 9.5.7. *The Shifted Inverse Power Method*. Suppose that μ is *not* an eigenvalue of A. (a) Show that the iterative system $\mathbf{u}^{(k+1)} = (A - \mu\,\mathrm{I})^{-1}\mathbf{u}^{(k)}$ converges to the eigenvector of A corresponding to the eigenvalue $\lambda^\star$ that is *closest* to μ. Explain how to find the eigenvalue $\lambda^\star$. (b) What is the rate of convergence of the algorithm? (c) What happens if μ is an eigenvalue?

♠ 9.5.8. Apply the Shifted Inverse Power Method of Exercise 9.5.7 to the find the eigenvalue closest to $\mu = .5$ of the matrices in Exercise 9.5.1.

9.5.9. Suppose that $A\mathbf{u}^{(k)} = \mathbf{0}$ in the iterative procedure (9.82). What does this indicate?

♠ 9.5.10. (i) Explain how to use the Deflation Method of Exercise 8.2.51 to find the subdominant eigenvalue of a nonsingular matrix A. (ii) Apply your method to the matrices listed in Exercise 9.5.1.

The QR Algorithm

As stated, the Power Method produces only the dominant (largest in magnitude) eigenvalue of a matrix A. The Inverse Power Method of Exercise 9.5.5 can be used to find the smallest eigenvalue. Additional eigenvalues can be found by using the Shifted Inverse Power Method of Exercise 9.5.7, or the Deflation Method of Exercise 9.5.10. However, if we need to know *all* the eigenvalues, such piecemeal methods are too time-consuming to be of much practical value.

The most popular scheme for simultaneously approximating all the eigenvalues of a matrix A is the remarkable $Q\,R$ algorithm, first proposed in 1961 by John Francis, [**29**], and Vera Kublanovskaya, [**51**]. The underlying idea is simple, but surprising. The first step is to factor the matrix

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

into a product of an orthogonal matrix Q_0 and a positive (i.e., with all positive entries along the diagonal) upper triangular matrix R_0 by using the Gram–Schmidt orthogonalization procedure of Theorem 4.24, or, even better, the numerically stable version described in (4.28). Next, multiply the two factors together *in the wrong order*! The result is the new matrix

$$A_1 = R_0\,Q_0.$$

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

$$A_1 = Q_1\,R_1$$

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

$$A_2 = R_1\,Q_1.$$

The complete algorithm can be written as

$$A = A_0 = Q_0\,R_0, \qquad A_{k+1} = R_k\,Q_k = Q_{k+1}\,R_{k+1}, \qquad k = 0, 1, 2, \ldots\,, \tag{9.83}$$

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

The astonishing fact is that, for many matrices A with all real eigenvalues, the iterates $A_k \longrightarrow V$ converge to an upper triangular matrix V whose diagonal entries are the eigenvalues of A. Thus, after a sufficient number of iterations, say m, the matrix A_m will have very small entries below the diagonal, and one can read off a complete system of (approximate) eigenvalues along its diagonal. For each eigenvalue, the computation of the corresponding eigenvector can be most efficiently accomplished by applying the Shifted Inverse Power Method of Exercise 9.5.7 with parameter μ chosen near the computed eigenvalue.

Example 9.43. Consider the matrix $A = \begin{pmatrix} 2 & 1 \\ 2 & 3 \end{pmatrix}$. The initial Gram–Schmidt factorization $A = Q_0\,R_0$ yields

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

These are multiplied in the reverse order to give

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

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

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

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

The next iteration yields

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

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

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

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

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

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

$$S_k = S_{k-1} Q_k = Q_0 Q_1 \ \cdots \ Q_{k-1} Q_k, \tag{9.84}$$

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

Example 9.44. Consider the symmetric matrix $A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & -1 \\ 0 & -1 & 6 \end{pmatrix}$. The initial $A = Q_0 R_0$ factorization produces

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

and so

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

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

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

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

Continuing in this manner, after 10 iterations we have

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

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

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

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

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

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

We will devote the remainder of this section to a justification of the QR algorithm for a class of matrices. We will assume that A is symmetric, and that its (necessarily real) eigenvalues satisfy

$$|\,\lambda_1\,| > |\,\lambda_2\,| > \ \cdots\ > |\,\lambda_n\,| > 0. \tag{9.85}$$

According to the Spectral Theorem 8.38, the corresponding unit eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ (in the Euclidean norm) form an orthonormal basis of $\mathbb{R}^n$. Our analysis can be adapted to a broader class of matrices, but this will suffice to expose the main ideas without unduly complicating the exposition.

The secret is that the QR algorithm is, in fact, a well-disguised adaptation of the more primitive Power Method. If we were to use the Power Method to capture all the eigenvectors and eigenvalues of A, the first thought might be to try to perform it simultaneously on a complete basis $\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_n^{(0)}$ of $\mathbb{R}^n$ instead of just one individual vector. The problem is that, for almost all vectors, the power iterates $\mathbf{v}_j^{(k)} = A^k\,\mathbf{v}_j^{(0)}$ all tend to a multiple of the dominant eigenvector $\mathbf{u}_1$. Normalizing the vectors at each step, as in (9.82), is not any better, since then they merely converge to one of the two dominant unit eigenvectors $\pm\,\mathbf{u}_1$. However, if, inspired by the form of the eigenvector basis, we *orthonormalize* the vectors at each step, then we effectively prevent them from all accumulating at the same dominant unit eigenvector, and so, with some luck, the resulting vectors will converge to the full system of eigenvectors. Since orthonormalizing a basis via the Gram–Schmidt process is equivalent to a QR matrix factorization, the mechanics of the algorithm becomes less surprising.

In detail, we start with any orthonormal basis, which, for simplicity, we take to be the standard basis vectors of $\mathbb{R}^n$, and so $\mathbf{u}_1^{(0)} = \mathbf{e}_1, \ \ldots\ , \mathbf{u}_n^{(0)} = \mathbf{e}_n$. At the k^{th} stage of the algorithm, we set $\mathbf{u}_1^{(k)}, \ldots, \mathbf{u}_n^{(k)}$ to be the orthonormal vectors that result from applying the Gram–Schmidt algorithm to the power vectors $\mathbf{v}_j^{(k)} = A^k\,\mathbf{e}_j$. In matrix language, the vectors $\mathbf{v}_1^{(k)}, \ldots, \mathbf{v}_n^{(k)}$ are merely the columns of A^k, and the orthonormal basis $\mathbf{u}_1^{(k)}, \ldots, \mathbf{u}_n^{(k)}$ are the columns of the orthogonal matrix S_k in the QR decomposition of the k^{th} power of A, which we denote by

$$A^k = S_k\,P_k, \tag{9.86}$$

where P_k is positive upper triangular, meaning all its diagonal entries are positive. Note that, in view of (9.83)

$$A = Q_0\,R_0, \qquad\qquad A^2 = Q_0\,R_0\,Q_0\,R_0 = Q_0\,Q_1\,R_1\,R_0,$$

$$A^3 = Q_0\,R_0\,Q_0\,R_0\,Q_0\,R_0 = Q_0\,Q_1\,R_1\,Q_1\,R_1\,R_0 = Q_0\,Q_1\,Q_2\,R_2\,R_1\,R_0,$$

and, in general,

$$A^k = \bigl(\,Q_0\,Q_1\ \cdots\ Q_{k-1}\,\bigr)\,\bigl(\,R_{k-1}\ \cdots\ R_1\,R_0\,\bigr). \tag{9.87}$$

Proposition 4.23 tells us that the product of orthogonal matrices is also orthogonal. The product of positive upper triangular matrices is also positive upper triangular. Therefore, comparing (9.86, 87) and invoking the uniqueness of the $Q\,R$ factorization, we conclude that

$$S_k = Q_0\,Q_1 \,\cdots\, Q_{k-1} = S_{k-1}\,Q_{k-1}, \qquad P_k = R_{k-1}\,\cdots\,R_1\,R_0 = R_{k-1}\,P_{k-1}. \tag{9.88}$$

Let $S = (\,\mathbf{u}_1\;\mathbf{u}_2\,\ldots\,\mathbf{u}_n\,)$ denote an orthogonal matrix whose columns are unit eigenvectors of A. The Spectral Theorem 8.38 tells us that

$$A = S\,\Lambda\,S^T, \qquad \text{where} \qquad \Lambda = \operatorname{diag}(\lambda_1,\ldots,\lambda_n)$$

is the diagonal eigenvalue matrix. Substituting the spectral factorization into (9.86) yields

$$A^k = S\,\Lambda^k\,S^T = S_k\,P_k.$$

We now make one additional assumption on the matrix A by requiring that S^T be a regular matrix, meaning that it can be factored, $S^T = L\,U$, as the product of a lower unitriangular matrix and an upper triangular matrix. We can further assume, without loss of generality, that the diagonal entries of U — that is, the pivots of S^T — are all positive. Indeed, by Exercise 1.3.31, this can be arranged by multiplying each row of S^T by the sign of its pivot, which amounts to possibly replacing some of the unit eigenvectors $\mathbf{u}_j$ by their negatives $-\mathbf{u}_j$, which is allowed, since it does not affect their status as an orthonormal eigenvector basis. Regularity of S^T holds generically, and is the analogue of the condition that our initial vector in the Power Method includes a nonzero component of the dominant eigenvector.

Under these two assumptions,

$$A^k = S\,\Lambda^k\,L\,U = S_k\,P_k, \qquad \text{and hence} \qquad S\,\Lambda^k\,L = S_k\,P_k\,U^{-1}.$$

Multiplying on the right by Λ^{-k}, we obtain

$$S\,\Lambda^k\,L\,\Lambda^{-k} = S_k\,T_k, \qquad \text{where} \qquad T_k = P_k\,U^{-1}\,\Lambda^{-k} \tag{9.89}$$

is also a positive upper triangular matrix, since P_k, U, Λ are all of that form.

Now consider what happens as $k \to \infty$. The entries of the lower triangular matrix $N = \Lambda^k\,L\,\Lambda^{-k}$ are

$$n_{ij} = \begin{cases} l_{ij}(\lambda_i/\lambda_j)^k, & i > j, \\ l_{ii} = 1, & i = j, \\ 0, & i < j. \end{cases}$$

Since we are assuming $|\,\lambda_i\,| < |\,\lambda_j\,|$ when $i > j$, we immediately deduce that

$$\Lambda^k\,L\,\Lambda^{-k} \longrightarrow \mathrm{I}, \qquad \text{and hence} \qquad S_k\,T_k = S\,\Lambda^k\,L\,\Lambda^{-k} \longrightarrow S \qquad \text{as} \qquad k \longrightarrow \infty.$$

We now appeal to the following lemma, whose proof will be given after we finish the justification of the $Q\,R$ algorithm.

Lemma 9.45. Let $S_1, S_2, \ldots$ and S be orthogonal matrices and $T_1, T_2, \ldots$ positive upper triangular matrices. Then $S_k\,T_k \to S$ as $k \to \infty$ if and only if $S_k \to S$ and $T_k \to \mathrm{I}$.

Lemma 9.45 implies that, as claimed, the orthogonal matrices S_k do converge to the orthogonal eigenvector matrix S. Moreover, by (9.88–89),

$$R_k = P_k\,P_{k-1}^{-1} = \left(T_k\,\Lambda^k\,U^{-1}\right)\left(T_{k-1}\,\Lambda^{k-1}\,U^{-1}\right)^{-1} = T_k\,\Lambda\,T_{k-1}^{-1}.$$

Since both T_k and T_{k-1} converge to the identity matrix, R_k converges to the diagonal eigenvalue matrix Λ, as claimed. The eigenvalues appear in decreasing order along the diagonal — this is a consequence of our regularity assumption on the transposed eigenvector matrix S^T.

Theorem 9.46. If A is positive definite with all simple eigenvalues, and its transposed eigenvector matrix S^T is regular, then the matrices $S_k \to S$ and $R_k \to \Lambda$ appearing in the QR algorithm applied to A converge to, respectively, the eigenvector matrix S and the diagonal eigenvalue matrix Λ.

Remark. If A is symmetric and has all simple eigenvalues, then, for suitably large $\alpha \gg 0$, the *shifted matrix* $\widetilde{A} = A + \alpha\,\mathrm{I}$ is positive definite, has the same eigenvectors as A, and has simple shifted eigenvalues $\widetilde{\lambda}_k = \lambda_k + \alpha$. Thus, one can run the QR algorithm to determine the eigenvalues and eigenvectors of $\widetilde{A}$, and hence those of A by undoing the shift.

The last remaining item is a proof of Lemma 9.45. We write

$$S = (\,\mathbf{u}_1\ \mathbf{u}_2\ \ldots\ \mathbf{u}_n\,), \qquad S_k = \left(\mathbf{u}_1^{(k)}\ \mathbf{u}_2^{(k)}\ \ldots\ \mathbf{u}_n^{(k)}\right),$$

in columnar form. Let $t_{ij}^{(k)}$ denote the entries of the positive upper triangular matrix T_k. The last column of the limiting equation $S_k\,T_k \to S$ reads $t_{nn}^{(k)}\,\mathbf{u}_n^{(k)} \to \mathbf{u}_n$. Since both $\mathbf{u}_n^{(k)}$ and $\mathbf{u}_n$ are unit vectors, and $t_{nn}^{(k)} > 0$, it follows that

$$\|\, t_{nn}^{(k)}\,\mathbf{u}_n^{(k)} \,\| = t_{nn}^{(k)} \longrightarrow \|\,\mathbf{u}_n\,\| = 1, \quad \text{and hence the last column} \quad \mathbf{u}_n^{(k)} \longrightarrow \mathbf{u}_n.$$

The next to last column reads

$$t_{n-1,n-1}^{(k)}\,\mathbf{u}_{n-1}^{(k)} + t_{n-1,n}^{(k)}\,\mathbf{u}_n^{(k)} \longrightarrow \mathbf{u}_{n-1}.$$

Taking the inner product with $\mathbf{u}_n^{(k)} \to \mathbf{u}_n$ and using orthonormality, we deduce $t_{n-1,n}^{(k)} \to 0$, and so $t_{n-1,n-1}^{(k)}\,\mathbf{u}_{n-1}^{(k)} \to \mathbf{u}_{n-1}$, which, by the previous reasoning, implies $t_{n-1,n-1}^{(k)} \to 1$ and $\mathbf{u}_{n-1}^{(k)} \to \mathbf{u}_{n-1}$. The proof is completed by working backwards through the remaining columns, using a similar argument at each step. The remaining details are left to the interested reader.

Exercises

9.5.11. Apply the QR algorithm to the following symmetric matrices to find their eigenvalues and eigenvectors to 2 decimal places: (a) $\begin{pmatrix} 1 & 2 \\ 2 & 6 \end{pmatrix}$, (b) $\begin{pmatrix} 3 & -1 \\ -1 & 5 \end{pmatrix}$, (c) $\begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 3 \\ 0 & 3 & 1 \end{pmatrix}$, (d) $\begin{pmatrix} 2 & 5 & 0 \\ 5 & 0 & -3 \\ 0 & -3 & 3 \end{pmatrix}$, (e) $\begin{pmatrix} 3 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 \\ 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 3 \end{pmatrix}$, (f) $\begin{pmatrix} 6 & 1 & -1 & 0 \\ 1 & 8 & 1 & -1 \\ -1 & 1 & 4 & 1 \\ 0 & -1 & 1 & 3 \end{pmatrix}$.

9.5.12. Show that applying the QR algorithm to the matrix $A = \begin{pmatrix} 4 & -1 & 1 \\ -1 & 7 & 2 \\ 1 & 2 & 7 \end{pmatrix}$ results in a diagonal matrix with the eigenvalues on the diagonal, but not in decreasing order. Explain.

9.5.13. Apply the $Q\,R$ algorithm to the following non-symmetric matrices to find their eigenvalues to 3 decimal places:

(a) $\begin{pmatrix} -1 & -2 \\ 3 & 4 \end{pmatrix}$, (b) $\begin{pmatrix} 2 & 3 \\ 1 & 5 \end{pmatrix}$, (c) $\begin{pmatrix} 2 & 1 & 0 \\ 2 & 0 & -3 \\ 0 & -2 & 1 \end{pmatrix}$, (d) $\begin{pmatrix} 2 & 5 & 1 \\ 2 & -1 & 3 \\ 4 & 5 & 3 \end{pmatrix}$, (e) $\begin{pmatrix} 6 & 1 & 7 & 9 \\ 6 & 8 & 14 & 9 \\ 3 & 1 & 4 & 6 \\ 3 & 2 & 5 & 3 \end{pmatrix}$.

9.5.14. The matrix $A = \begin{pmatrix} -1 & 2 & 1 \\ -2 & 3 & 1 \\ -2 & 2 & 2 \end{pmatrix}$ has a double eigenvalue of 1, and so our proof of convergence of the $Q\,R$ algorithm doesn't apply. Does the $Q\,R$ algorithm find its eigenvalues?

9.5.15. Explain why the $Q\,R$ algorithm fails to find the eigenvalues of the matrices

$$(a)\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad (b)\ \begin{pmatrix} -2 & 1 & 0 \\ 0 & -2 & 1 \\ 1 & 0 & -2 \end{pmatrix}, \qquad (c)\ \begin{pmatrix} 5 & -4 & 2 \\ -4 & 5 & 2 \\ 2 & 2 & -1 \end{pmatrix}.$$

♢ 9.5.16. Prove that all of the matrices A_k defined in (9.83) have the same eigenvalues.

♢ 9.5.17. (a) Prove that if A is symmetric and tridiagonal, then all matrices A_k appearing in the $Q\,R$ algorithm are also symmetric and tridiagonal. *Hint*: First prove symmetry.
(b) Is the result true if A is not symmetric — only tridiagonal?

Tridiagonalization

In practical implementations, the direct $Q\,R$ algorithm often takes overly long before providing reasonable approximations to the eigenvalues of large matrices. Fortunately, the algorithm can be made much more efficient by a simple preprocessing step. The key observation is that the $Q\,R$ algorithm preserves the class of symmetric tridiagonal matrices, and, like Gaussian Elimination, is much faster when applied to this class. Moreover, by applying a sequence of Householder reflection matrices (4.35), we can convert any symmetric matrix into tridiagonal form while preserving all the eigenvalues. Thus, by first using the Householder tridiagonalization process, and then applying the $Q\,R$ Method to the resulting tridiagonal matrix, we obtain an efficient and practical algorithm for computing eigenvalues of large symmetric matrices. Generalizations to non-symmetric matrices will be briefly considered at the end of the section.

In Householder's approach to the $Q\,R$ factorization, we were able to convert the matrix A to upper triangular form R by a sequence of elementary reflection matrices. Unfortunately, this procedure does not preserve the eigenvalues of the matrix — the diagonal entries of R are *not* the eigenvalues — and so we need to be a bit more clever here. We begin by recalling, from Exercise 8.2.32, that similar matrices have the same eigenvalues (but not the same eigenvectors).

Lemma 9.47. If $H = \mathrm{I} - 2\,\mathbf{u}\,\mathbf{u}^T$ is an elementary reflection matrix, with $\mathbf{u} \in \mathbb{R}^n$ a unit vector (under the Euclidean norm), then A and $B = H\,A\,H$ are similar matrices and hence have the same eigenvalues.

Proof: It suffices to note that, according to (4.37), $H^{-1} = H$, and hence $B = H^{-1}A\,H$ is similar to A. *Q.E.D.*

Now, starting with a symmetric $n \times n$ matrix A, our goal is to devise a similar tridiagonal matrix by applying a sequence of Householder reflections. Using the Euclidean norm, we begin by setting

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

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

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

be the corresponding elementary reflection matrix that maps $\mathbf{x}_1$ to $\mathbf{y}_1$. Either the plus or the minus sign in the formula for $\mathbf{y}_1$ works in the algorithm; a good choice is to set it to be the opposite of the sign of the entry a_{21}, which helps minimize the possible effects of round-off error in computing the unit vector $\mathbf{u}_1$. By direct computation, based on Lemma 4.28 and the fact that the first entry of $\mathbf{u}_1$ is zero, we obtain

$$A_2 = H_1 A H_1 = \begin{pmatrix} a_{11} & r_1 & 0 & \ldots & 0 \\ r_1 & \widetilde{a}_{22} & \widetilde{a}_{23} & \ldots & \widetilde{a}_{2n} \\ 0 & \widetilde{a}_{32} & \widetilde{a}_{33} & \ldots & \widetilde{a}_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \widetilde{a}_{n2} & \widetilde{a}_{n3} & \ldots & \widetilde{a}_{nn} \end{pmatrix} \tag{9.90}$$

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

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

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

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

we construct the similar matrix

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

whose first two rows and columns are now in tridiagonal form. The remaining steps in the algorithm should now be clear. Thus, the final result is a tridiagonal matrix $T = A_n$ that has the *same eigenvalues* (but not the same eigenvectors) as the original symmetric matrix A. Let us illustrate the method by an example.

Example 9.48. To tridiagonalize $A = \begin{pmatrix} 4 & 1 & -1 & 2 \\ 1 & 4 & 1 & -1 \\ -1 & 1 & 4 & 1 \\ 2 & -1 & 1 & 4 \end{pmatrix}$, we begin with its first column. We set $\mathbf{x}_1 = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 2 \end{pmatrix}$, so that $\mathbf{y}_1 = \begin{pmatrix} 0 \\ \sqrt{6} \\ 0 \\ 0 \end{pmatrix} \simeq \begin{pmatrix} 0 \\ 2.4495 \\ 0 \\ 0 \end{pmatrix}$. Therefore, the unit vector and corresponding Householder matrix are

$$\mathbf{u}_1 = \frac{\mathbf{x}_1 - \mathbf{y}_1}{\|\mathbf{x}_1 - \mathbf{y}_1\|} = \begin{pmatrix} 0 \\ .8391 \\ -.2433 \\ .4865 \end{pmatrix}, \quad H_1 = \mathrm{I} - 2\,\mathbf{u}_1\,\mathbf{u}_1^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -.4082 & .4082 & -.8165 \\ 0 & .4082 & .8816 & .2367 \\ 0 & -.8165 & .2367 & .5266 \end{pmatrix}.$$

We compute

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

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

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

The resulting matrix

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

is now in tridiagonal form.

Since the final tridiagonal matrix T has the same eigenvalues as A, we can apply the QR algorithm to T to approximate the common eigenvalues. According to Exercise 9.5.17, if $A = A_1$ is tridiagonal, so are all its QR iterates $A_2, A_3, \ldots$. Moreover, far fewer arithmetic operations are required; in Exercise 9.5.25, you are asked to quantify this. For instance, in the preceding example, after we apply 20 iterations of the QR algorithm directly to T, the upper triangular factor has become

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

The eigenvalues of T, and hence also of A, appear along the diagonal, and are correct to 4 decimal places. As noted earlier, with the eigenvalues in hand the corresponding eigenvectors can then be found via the Shifted Inverse Power Method of Exercise 9.5.7.

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

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

where the starred entries can be anything. It can be proved that the QR algorithm maintains the upper Hessenberg form, and, while not as efficient as in the tridiagonal case, still yields a significant savings in computational effort required to find the common eigenvalues.

If A has no eigenvalues of the same magnitude, which, in particular, requires all its eigenvalues to be simple, then application of the tridiagonal QR algorithm to its tridiagonalization will, usually, produce its eigenvalues. More generally, if A has k eigenvalues of the same magnitude, then the QR algorithm, applied either directly to A, or to its tridiagonalization, will, again generically, converge to a block upper triangular matrix, with an $k \times k$ matrix in the block diagonal slot that has these same eigenvalues. Thus, for example, if A is a real matrix with simple real and complex eigenvalues, then each complex conjugate pair will be the eigenvalues of one of the 2×2 matrices appearing on the diagonal of the eventual QR iterates, while the real eigenvalues will appear directly (in a 1×1 "block") on the diagonal.

Further details and results can be found in [**21**, **66**, **69**, **89**, **90**].

Exercises

9.5.18. Use Householder matrices to convert the following matrices into tridiagonal form:

$$(a)\ \begin{pmatrix} 8 & -7 & 2 \\ -7 & 17 & -7 \\ 2 & -7 & 8 \end{pmatrix}, \quad (b)\ \begin{pmatrix} 5 & 1 & -2 & 1 \\ 1 & 5 & 1 & -2 \\ -2 & 1 & 5 & 1 \\ 1 & -2 & 1 & 5 \end{pmatrix}, \quad (c)\ \begin{pmatrix} 4 & 0 & -1 & 1 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 2 & 0 \\ 1 & -1 & 0 & 3 \end{pmatrix}.$$

♠ 9.5.19. Find the eigenvalues, to 2 decimal places, of the matrices in Exercise 9.5.18 by applying the QR algorithm to the tridiagonal form.

♠ 9.5.20. Use the tridiagonal QR Method to find the singular values of $A = \begin{pmatrix} 2 & 2 & 1 & -1 \\ 1 & -2 & 0 & 1 \\ 0 & -1 & 2 & 2 \end{pmatrix}$.

9.5.21. Use Householder matrices to convert the following matrices into upper Hessenberg form:

$$(a)\ \begin{pmatrix} 3 & -1 & 2 \\ 1 & 3 & -4 \\ 2 & -1 & -1 \end{pmatrix}, \quad (b)\ \begin{pmatrix} 3 & 2 & -1 & 1 \\ 2 & 4 & 0 & 1 \\ 0 & 1 & 2 & -6 \\ 1 & 0 & -5 & 1 \end{pmatrix}, \quad (c)\ \begin{pmatrix} 1 & 0 & -1 & 1 \\ 2 & 1 & 1 & -1 \\ -1 & 0 & 1 & 3 \\ 3 & -1 & 1 & 4 \end{pmatrix}.$$

♠ 9.5.22. Find the eigenvalues, to 2 decimal places, of the matrices in Exercise 9.5.21 by applying the QR algorithm to the upper Hessenberg form.

9.5.23. Prove that the effect of the first Householder reflection is as given in (9.90).

9.5.24. What is the effect of tridiagonalization on the eigenvectors of the matrix?

◇ 9.5.25. (*a*) How many arithmetic operations — multiplications/divisions and additions/subtractions — are required to place a generic $n \times n$ symmetric matrix into tridiagonal form? (*b*) How many operations are needed to perform one iteration of the QR algorithm on an $n \times n$ tridiagonal matrix? (*c*) How much faster, on average, is the tridiagonal algorithm than the direct QR algorithm for finding the eigenvalues of a symmetric matrix?

9.5.26. Write out a pseudocode program to tridiagonalize a matrix. The input should be an $n \times n$ matrix A, and the output should be the Householder unit vectors $\mathbf{u}_1, \ldots, \mathbf{u}_{n-1}$ and the tridiagonal matrix R. Does your program produce the upper Hessenberg form when the input matrix is not symmetric?

◇ 9.5.27. Prove that in the $H = LU$ factorization of a regular upper Hessenberg matrix, the lower triangular factor L is bidiagonal, as in (1.67).

9.6 Krylov Subspace Methods

So far, we have established two broad classes of algorithms for solving linear systems. The first, known as *direct methods*, are based on some version of Gaussian Elimination or matrix factorization. Direct methods eventually[†] obtain the exact solution, but must be carried through to completion before any useful information is obtained. The second class contains the *iterative methods* discussed above that lead to closer and closer approximations to the solution, but almost never reach the exact value. One might ask whether there are algorithms that combine the best of both: *semi-direct methods* whose intermediate computations lead to closer and closer approximations, and, moreover, are guaranteed to terminate in a finite number of steps with the exact solution in hand.

In recent years, for dealing with large sparse linear systems, such as those arising from the numerical solution of partial differential equations, semi-direct iterative methods based on Krylov subspaces have become quite popular. The original ideas were introduced in the 1930's by the Russian naval engineer Alexei Krylov, who was in search of an efficient and reliable method for numerically computing eigenvalues. Krylov methods have seen much development in a variety of directions, [**32**, **70**, **85**], and we will show how they can be used to iteratively solve linear systems and to compute eigenvalues.

Krylov Subspaces

The starting point is an $n \times n$ matrix A, assumed to be real, although extensions to complex matrices are relatively straightforward. In applications, A is both large and sparse, meaning that most of its entries are 0, and so multiplying A by a vector $\mathbf{v} \in \mathbb{R}^n$ to produce the vector $A\mathbf{v}$ is an efficient operation.

Recall that the Power Method for computing the dominant eigenvalue and eigenvector of A is based on successive iterates applied to a randomly chosen initial vector: $\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, A^3\mathbf{v}, \ldots$. We will employ these particular vectors to span a collection of subspaces.

Definition 9.49. Given an $n \times n$ real matrix A, the *Krylov subspace* of *order* $k \geq 1$ generated by a nonzero vector $\mathbf{0} \neq \mathbf{v} \in \mathbb{R}^n$ is the subspace $V^{(k)} \subset \mathbb{R}^n$ spanned by the vectors $\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{k-1}\mathbf{v}$. We also set $V^{(0)} = \{\mathbf{0}\}$ by convention.

[†] This assumes that we are dealing with a fully accurate implementation, i.e., without round-off or other numerical error. In this discussion, numerical instability will be left aside as a separate, albeit ultimately important, concern.

For example, if $\mathbf{v}$ is an eigenvector of A, so $A\mathbf{v} = \lambda\mathbf{v}$, then $V^{(2)} = V^{(1)}$ is the one-dimensional eigenspace spanned by $\mathbf{v}$; conversely, if $V^{(2)}$ is one-dimensional, then $\mathbf{v}$ is necessarily an eigenvector, and hence $V^{(k)} = V^{(1)}$ for all $k \geq 1$. More generally, if $V^{(j+1)} = V^{(j)}$ for some $j \geq 0$, then $V^{(k)} = V^{(j)}$ for all $k \geq j$. This is easily proved by induction: by assumption, $A^j\mathbf{v} \in V^{(j)}$, and thus can be written as a linear combination

$$A^j\mathbf{v} = c_1\mathbf{v} + c_2 A\mathbf{v} + \cdots + c_{j-1}A^{j-2}\mathbf{v} + c_j A^{j-1}\mathbf{v} \in V^{(j)}$$

for some scalars $c_1, \ldots, c_j$. Thus,

$$\begin{aligned} A^{j+1}\mathbf{v} &= c_1 A\mathbf{v} + c_2 A^2\mathbf{v} + \cdots + c_{j-1}A^{j-1}\mathbf{v} + c_j A^j\mathbf{v} \\ &= c_j c_1\mathbf{v} + (c_1 + c_j c_2)A\mathbf{v} + \cdots + (c_{j-2} + c_j c_{j-1})A^{j-2}\mathbf{v} + (c_j + c_j^2)A^{j-1}\mathbf{v} \in V^{(j)} \end{aligned}$$

also, proving that $V^{(j+2)} = V^{(j)}$. The general induction step is clear.

Since we assumed $\mathbf{v} \neq \mathbf{0}$, as otherwise all $V^{(k)} = \{\mathbf{0}\}$ are trivial and not of interest, this argument implies the existence of an integer $m \in \mathbb{N}$, called the *stabilization order*, such that $\dim V^{(k)} = k$ for $k = 1, \ldots, m$, while $V^{(k)} = V^{(m)}$ has dimension m for all $k \geq m$. Since we are working in $\mathbb{R}^n$, clearly $m \leq n$; Exercise 9.6.3 gives a stricter bound for m in terms of the degree of the minimal polynomial of the matrix A, as defined in Exercise 8.6.23. We also note the following useful result.

Lemma 9.50. Suppose $V^{(k)} \neq V^{(k-1)}$. Let $\mathbf{w} \in V^{(k)} \setminus V^{(k-1)}$. Then $A\mathbf{w} \in V^{(k+1)}$ and, moreover, $V^{(k+1)}$ is spanned by $A\mathbf{w}$ and (a basis of) $V^{(k)}$. Moreover, if $A\mathbf{w} \in V^{(k)}$, then $V^{(k+1)} = V^{(k)}$ and the Krylov subspaces stabilize at order k.

Proof: By assumption,

$$\mathbf{w} = c_1\mathbf{v} + c_2 A\mathbf{v} + \cdots + c_{k-1}A^{k-2}\mathbf{v} + c_k A^{k-1}\mathbf{v}$$

for some scalars $c_1, \ldots, c_k$ with $c_k \neq 0$. Thus, as above,

$$A\mathbf{w} = c_1 A\mathbf{v} + c_2 A^2\mathbf{v} + \cdots + c_{k-1}A^{k-1}\mathbf{v} + c_k A^k\mathbf{v} \in V^{(k+1)}. \tag{9.91}$$

If $A\mathbf{w} \in V^{(k)}$, the left-hand side of (9.91) is a linear combination of $\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{k-1}\mathbf{v}$, and hence, since $c_k \neq 0$, so is $A^k\mathbf{v}$, which implies $V^{(k+1)} = V^{(k)}$. Otherwise, (9.91) implies that $A^k\mathbf{v}$ is a linear combination of $A\mathbf{w}$ and $A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{k-1}\mathbf{v}$, and thus every vector in $V^{(k+1)}$ can be written as a linear combination of $A\mathbf{w}$ and the Krylov vectors $\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, \ldots, A^{k-1}\mathbf{v} \in V^{(k)}$. *Q.E.D.*

For simplicity in what follows, we will assume that A has all real eigenvalues; for example A might be a symmetric matrix. We further assume that A has a unique dominant eigenvalue λ_1, so that λ_1 is a simple eigenvalue, and $|\lambda_1| > |\lambda_j|$ for all $j > 1$. In this case, as we know from our earlier analysis, for most initial choices of the vector $\mathbf{v}$, the vectors used to define the Krylov subspace tend to scalar multiples of a dominant eigenvector $\mathbf{v}_1$, meaning that $A^k\mathbf{v} \to \lambda_1^k\mathbf{v}_1$ as $k \to \infty$. Thus, the Krylov vectors in and of themselves contain increasingly little information, particularly in a numerical environment. As with the Power Method, matrices with several dominant eigenvalues, including real matrices with complex conjugate eigenvalues and matrices for which $\pm\lambda_1$ are both eigenvalues, require suitable modifications of the methods.

Arnoldi Iteration

The way to get around the pure power behavior was already introduced in the design of the QR algorithm: instead of the Krylov vectors, one constructs an orthonormal basis of the Krylov subspace using the Gram–Schmidt process. (As above, we work with the dot product $\mathbf{v} \cdot \mathbf{w} = \mathbf{v}^T \mathbf{w}$ and corresponding Euclidean norm throughout this presentation, leaving the investigation of other inner products to the motivated reader.) To this end, we may as well start with a unit vector, and so replace the initial vector $\mathbf{v}$ by the unit vector $\mathbf{u}_1 = \mathbf{v}/\|\mathbf{v}\|$, so $\|\mathbf{u}_1\| = 1$, which spans the initial Krylov subspace $V^{(1)}$. The second order subspace $V^{(2)}$ will be spanned by the vectors $\mathbf{u}_1$ and $A\mathbf{u}_1$, and we extract an orthonormal basis by projection. First, according to our orthogonal projection formulas, the vector

$$\mathbf{v}_2 = A\mathbf{u}_1 - h_{11}\mathbf{u}_1, \qquad \text{where} \qquad h_{11} = \mathbf{u}_1^T A\mathbf{u}_1,$$

satisfies the desired orthogonality condition $\mathbf{u}_1 \cdot \mathbf{v}_2 = 0$. If $\mathbf{v}_2 = \mathbf{0}$, then $\mathbf{u}_1$ is an eigenvector of A, and the process terminates, since the Krylov subspaces would immediately stabilize: $V^{(k)} = V^{(1)}$ for all $k \geq 1$. Otherwise, we replace $\mathbf{v}_2$ by the unit vector

$$\mathbf{u}_2 = \frac{\mathbf{v}_2}{h_{21}}, \qquad \text{where} \qquad h_{21} = \|\mathbf{v}_2\|,$$

and deduce that $\mathbf{u}_1$ and $\mathbf{u}_2$ form an orthonormal basis for $V^{(2)}$. Proceeding in this manner, assuming that $k \leq m$, the stabilization order, at the k^{th} stage, we have already computed orthonormal vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ such that $\mathbf{u}_1, \ldots, \mathbf{u}_j$ form an orthonormal basis of $V^{(j)}$ for each $j = 1, \ldots, k$. Taking $\mathbf{w} = \mathbf{u}_k$ in Lemma 9.50, we deduce that $\mathbf{u}_1, \ldots, \mathbf{u}_k$ and $A\mathbf{u}_k$ span $V^{(k+1)}$. Our orthogonal projection formula (4.41) implies that

$$\mathbf{v}_{k+1} = A\mathbf{u}_k - \sum_{j=1}^{k} h_{jk}\mathbf{u}_j, \qquad \text{where} \qquad h_{jk} = \mathbf{u}_j^T A\mathbf{u}_k \tag{9.92}$$

lies in $V^{(k+1)}$ and is orthogonal to $\mathbf{u}_0, \ldots, \mathbf{u}_k$. If $\mathbf{v}_{k+1} = \mathbf{0}$, then $A\mathbf{u}_k \in V^{(k)}$, and, again by Lemma 9.50, the Krylov spaces have stabilized with $V^{(k+1)} = V^{(k)}$. Otherwise, let

$$\mathbf{u}_{k+1} = \frac{\mathbf{v}_{k+1}}{h_{k+1,k}}, \qquad \text{where} \qquad h_{k+1,k} = \|\mathbf{v}_{k+1}\|, \tag{9.93}$$

be the corresponding unit vector, so that $\mathbf{u}_0, \ldots, \mathbf{u}_{k+1}$ form an orthonormal basis of $V^{(k+1)}$, as desired.

While the preceding algorithm will work in favorable situations, the preferred method, known as *Arnoldi iteration*, named after the mid-twentieth-century American engineer Walter Arnoldi, employs the stabilized Gram–Schmidt process described in Section 4.2, thereby ameliorating, as much as possible, potential numerical instabilities. Thus, at step $k \geq 1$, having $\mathbf{u}_1, \ldots, \mathbf{u}_k$ in hand, one iteratively computes

$$\mathbf{v}_{k+1}^{(1)} = A\mathbf{u}_k, \;\; \mathbf{v}_{k+1}^{(j+1)} = \mathbf{v}_{k+1}^{(j)} - h_{jk}\mathbf{u}_j, \quad \text{where} \quad h_{jk} = \mathbf{u}_j^T \mathbf{v}_{k+1}^{(j)}, \quad \text{for} \quad j = 1, \ldots, k-1. \tag{9.94}$$

We then set $\mathbf{v}_{k+1} = \mathbf{v}_{k+1}^{(k)}$ and, if it is nonzero, use (9.93) to define the next orthonormal basis vector $\mathbf{u}_{k+1}$. In Exercise 9.6.6 you are asked to prove that the resulting *Arnoldi vectors* $\mathbf{u}_k$ and coefficients h_{jk} are the same as in (9.92, 93) (if computed exactly).

It is instructive to formulate the Arnoldi orthonormalization process in matrix form. First note that we can rewrite (9.92–93) as

$$A\mathbf{u}_k = \sum_{j=0}^{k+1} h_{jk}\mathbf{u}_j, \tag{9.95}$$

and hence, by orthonormality

$$h_{jk} = \begin{cases} \mathbf{u}_j^T A \mathbf{u}_k, & 1 \le j \le k+1, \\ 0, & j \ge k+2. \end{cases} \tag{9.96}$$

Let $Q_k = (\, \mathbf{u}_1 \; \mathbf{u}_2 \; \ldots \; \mathbf{u}_k \,)$ denote the $n \times k$ matrix whose columns are the first k Arnoldi vectors. Since these are orthonormal, it follows that

$$Q_k^T Q_k = \mathrm{I}. \tag{9.97}$$

(However, keep in mind that Q_k is a rectangular matrix, and so $Q_k Q_k^T$ is in general *not* the identity matrix.) Let

$$H_k = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & \cdots & h_{1,k-2} & h_{1,k-1} & h_{1k} \\ h_{21} & h_{22} & h_{23} & h_{24} & \cdots & h_{2,k-2} & h_{2,k-1} & h_{2k} \\ 0 & h_{32} & h_{33} & h_{34} & \cdots & h_{3,k-2} & h_{3,k-1} & h_{3k} \\ 0 & 0 & h_{43} & h_{44} & \cdots & h_{4,k-2} & h_{4,k-1} & h_{4k} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & h_{k-1,k-2} & h_{k-1,k-1} & h_{k-1,k} \\ 0 & 0 & 0 & \cdots & 0 & 0 & h_{k,k-1} & h_{kk} \end{pmatrix} \tag{9.98}$$

be the $k \times k$ upper Hessenberg matrix formed by the coefficients h_{jk} given in (9.96), which implies that

$$H_k = Q_k^T A Q_k. \tag{9.99}$$

In particular, if A is symmetric, then so is H_k, which implies that it is also tridiagonal. In this case, the Arnoldi algorithm is known as the *symmetric Lanczos algorithm*, after the Hungarian mathematician Cornelius Lanczos.

Equation (9.99) yields an alternative interpretation of the Arnoldi iteration as a (partial) orthogonal reduction of A to Hessenberg or, in the symmetric case, tridiagonal form. The matrix H_k can be viewed as the representation of the orthogonal projection of A onto the Krylov subspace $V^{(k)}$ in terms of the basis formed by the Arnoldi vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$. Thus, we can identify H_k with the (projected) action of A on the subspace $V^{(k)}$ and, as such, its dominant eigenvalues and eigenvectors, which can be computed using the QR algorithm, are expected to form good approximations to those of A itself. Since its predecessor, H_{k-1}, coincides with the upper left $(k-1) \times (k-1)$ submatrix of H_k, the QR factorizations of the Hessenberg coefficient matrices H_k can be speeded up by an iterative procedure; see [**70**] for details. One can also use Householder reflections to tridiagonalize H_k before applying QR. Of course, if A is symmetric, then, as noted above, H_k is already tridiagonal and so this step is superfluous. Moreover, if the method is carried out to the stabilization order m, the resulting Krylov subspace is invariant under A, and hence the eigenvalues of H_m coincide with those of A restricted to $V^{(m)}$, cf. Exercise 8.4.5. In this manner, the Arnoldi/Lanczos algorithm produces a semi-direct method for approximating eigenvalues of the matrix A. Again, the Shifted Inverse Power Method of Exercise 9.5.7 can then be used to compute each corresponding eigenvector.

We further note, as a consequence of the first equation in (9.95), the following formula relating the Arnoldi matrix Q_k to its successor Q_{k+1}:

$$A Q_k = Q_{k+1} \widetilde{H}_k, \tag{9.100}$$

where

$$\widetilde{H}_k = \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & \cdots & h_{1,k-2} & h_{1,k-1} & h_{1k} \\ h_{21} & h_{22} & h_{23} & h_{24} & \cdots & h_{2,k-2} & h_{2,k-1} & h_{2k} \\ 0 & h_{32} & h_{33} & h_{34} & \cdots & h_{3,k-2} & h_{3,k-1} & h_{3k} \\ 0 & 0 & h_{43} & h_{44} & \cdots & h_{4,k-2} & h_{4,k-1} & h_{4k} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & h_{k-1,k-2} & h_{k-1,k-1} & h_{k-1,k} \\ 0 & 0 & 0 & \cdots & 0 & 0 & h_{k,k-1} & h_{kk} \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & h_{k+1,1} \end{pmatrix} \tag{9.101}$$

is the $(k+1) \times k$ matrix formed by appending the indicated bottom row to H_k.

Finally, we note the useful formula

$$Q_k^T \mathbf{v} = \| \mathbf{v} \| \, \mathbf{e}_1, \tag{9.102}$$

with $\mathbf{e}_1 = (\,1, 0, 0, \ldots, 0\,)^T \in \mathbb{R}^k$ the first standard basis vector. This is a consequence of the orthonormality of the Arnoldi vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$, which form the rows of Q_k^T, along with the fact that $\mathbf{v} = \| \mathbf{v} \| \, \mathbf{u}_1$.

Remark. In numerical applications, the best results are obtained by maximizing the stabilization order of the Krylov subspaces generated by the initial vector, and so a random choice of the initial vector $\mathbf{v}$, or, equivalently, the initial unit vector $\mathbf{u}_1$ is preferred so as to minimize chances of low order degeneration and consequent inaccuracies. In the unlucky event that stabilization occurs prematurely, one should restart the method with a different choice of initial vector, [**70**].

The Full Orthogonalization Method

Krylov subspaces can also be applied to generate powerful semi-direct iterative algorithms for solving linear systems. There are two different approaches. The first starts with the concept of a *weak* or *Galerkin formulation* of a linear system, which is the elementary observation that that the only vector that is orthogonal to every vector in an inner product space is the zero vector; see Exercise 3.1.10(a). As above, we concentrate on the case $V = \mathbb{R}^n$ with the standard dot product. The observation means that $\mathbf{x} \in \mathbb{R}^n$ solves the linear system $A\,\mathbf{x} = \mathbf{b}$ if and only if

$$\mathbf{v}^T (A\,\mathbf{x} - \mathbf{b}) = \mathbf{0} \qquad \text{for all} \qquad \mathbf{v} \in \mathbb{R}^n. \tag{9.103}$$

Solution techniques based on this formulation were first studied in depth, in the context of the mechanics of thin elastic plates, by the Russian engineer Boris Galerkin in the first half of the twentieth century, and often bear his name.

In the case of linear algebraic systems, the Galerkin formulation per se does not add anything to what we already know. However, it becomes important for the numerical approximation of solutions by restricting (9.103) to a smaller-dimensional subspace $V \subset \mathbb{R}^n$. Specifically, one seeks a vector $\mathbf{x} \in V$ such that the Galerkin formulation (9.103) holds for all $\mathbf{v} \in V$. In other words, the approximate solution is the vector $\mathbf{x} \in V$ such that the residual $\mathbf{r} = \mathbf{b} - A\,\mathbf{x}$ is orthogonal to the subspace V. With a suitably inspired choice of the subspace V, the Galerkin formulation may well provide a decent approximation to the actual solution.

Remark. One can easily adapt the Galerkin formulation to general linear systems $L[u] = f$, where $L: U \to V$ is any linear operator between vector spaces. The corresponding weak formulation, as described in Exercise 7.5.9, has become an extremely important tool in the modern mathematical analysis of differential equations, which take place in infinite-dimensional function spaces. Moreover, the restriction of the weak formulation to a finite-dimensional subspace $V \subset U$ is the basis of the powerful finite element solution method for boundary value problems; see [**8**, **61**] for details.

Remark. The question of existence and uniqueness of the Galerkin approximate solution depends upon the matrix A and the choice of subspace V. Given a basis $\mathbf{v}_1, \ldots, \mathbf{v}_k$ of V, we express $\mathbf{x} = y_1 \mathbf{v}_1 + \;\cdots\; + y_k \mathbf{v}_k = S\mathbf{y}$, where $S = (\,\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_k\,)$ is the $n \times k$ matrix whose columns are the basis vectors, while $\mathbf{y} = (\,y_1, y_2, \ldots, y_k\,)^T \in \mathbb{R}^k$ contains the coordinates of $\mathbf{x} = S\mathbf{y} \in V$ with respect to the given basis. Then the Galerkin conditions on V can be written as

$$\mathbf{v}^T(A\mathbf{x} - \mathbf{b}) = \mathbf{v}^T(AS\mathbf{y} - \mathbf{b}) = \mathbf{0} \qquad \text{for all} \qquad \mathbf{v} \in V.$$

Expressing $\mathbf{v} = S\mathbf{z}$ for $\mathbf{z} \in \mathbb{R}^k$ in the same fashion, this becomes

$$\mathbf{z}^T S^T (AS\mathbf{y} - \mathbf{b}) = \mathbf{z}^T (S^T AS\mathbf{y} - S^T\mathbf{b}) = \mathbf{0} \qquad \text{for all} \qquad \mathbf{z} \in \mathbb{R}^k,$$

which clearly holds if and only if

$$S^T AS\mathbf{y} = S^T\mathbf{b}. \tag{9.104}$$

This is a linear system of k equations in the k unknowns $\mathbf{y} \in \mathbb{R}^k$. Thus, a solution exists and is uniquely determined if and only if the $k \times k$ coefficient matrix $S^T AS$ is nonsingular, which requires, at the very least, $\operatorname{rank} A \geq k$, and places additional constraints on S.

As you may suspect, in the case of a linear algebraic system, a particularly good choice of subspace for a Galerkin approximation to the solution is a Krylov subspace. The resulting solution method is known as the *Full Orthogonalization Method*, abbreviated FOM, [**70**]. In detail, the method proceeds as follows. Let $V^{(k)} \subset \mathbb{R}^n$ be the order k Krylov subspace generated by the right-hand side $\mathbf{b}$, and thus spanned by $\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots, A^{k-1}\mathbf{b}$. The k^{th} *Krylov approximation* to the solution $\mathbf{x}$ is the vector $\mathbf{x}_k \in V^{(k)}$ whose residual $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ satisfies the Galerkin condition of being orthogonal to the subspace:

$$\mathbf{v} \cdot \mathbf{r}_k = \mathbf{v}^T(\mathbf{b} - A\mathbf{x}_k) = \mathbf{0} \qquad \text{for all} \qquad \mathbf{v} \in V^{(k)}.$$

In particular, the initial approximation is taken to be $\mathbf{x}_0 = \mathbf{0} \in V^{(1)}$, with residual vector $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = \mathbf{b}$. Moreover, Lemma 9.50 implies $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k \in V^{(k+1)}$. Since it is orthogonal to $V^{(k)}$, it must be a scalar multiple of the $(k+1)^{\text{st}}$ Arnoldi vector:

$$\mathbf{r}_k = c_{k+1}\mathbf{u}_{k+1}, \qquad \text{where} \qquad c_{k+1} = \|\,\mathbf{r}_k\,\|. \tag{9.105}$$

This implies that the residual vectors are also mutually orthogonal:

$$\mathbf{r}_j \cdot \mathbf{r}_k = 0, \qquad j \neq k. \tag{9.106}$$

Using the orthonormal Arnoldi basis vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k \in V^{(k)}$, which form the columns of the matrix Q_k, we write $\mathbf{x}_k = Q_k\mathbf{y}_k$, and hence, recalling (9.99), equation (9.104) becomes

$$Q_k^T A Q_k \mathbf{y}_k = H_k \mathbf{y}_k = Q_k^T \mathbf{b} = \|\,\mathbf{b}\,\|\,\mathbf{e}_1, \tag{9.107}$$

where H_k is the upper Hessenberg matrix (9.99), and we use (9.102) (with $\mathbf{b}$ replacing $\mathbf{v}$, as per our initial supposition) to obtain the final expression. Solving the resulting system (9.107), assuming H_k is invertible, for $\mathbf{y}_k = \|\,\mathbf{b}\,\|\, H_k^{-1}\mathbf{e}_1$ produces the k^{th} order Krylov approximation to the solution

$$\mathbf{x}_k = Q_k\mathbf{y}_k = \|\,\mathbf{b}\,\|\, Q_k H_k^{-1}\mathbf{e}_1. \tag{9.108}$$

Of course, in applications one does not explicitly compute the inverse H_k^{-1} but rather uses, say, its LU factorization $H_k = L_k U_k$ (assuming regularity), coupled with forward and back substitution to solve (9.107). Moreover, according to Exercise 9.5.27, the lower unitriangular factor L_k is bidiagonal, meaning that all entries not on the diagonal or subdiagonal are zero. Of course, because the upper left $(k-1)\times(k-1)$ entries of H_k are the same as those of its predecessor, whose factorization $H_{k-1} = L_{k-1}U_{k-1}$ can be assumed to already be known, we can quickly factorize H_k. Namely, we write

$$H_k = \begin{pmatrix} H_{k-1} & \mathbf{f}_k \\ \mathbf{g}_k^T & h_{kk} \end{pmatrix}, \qquad L_k = \begin{pmatrix} L_{k-1} & \mathbf{0} \\ \mathbf{m}_k^T & 1 \end{pmatrix}, \qquad U_k = \begin{pmatrix} U_{k-1} & \mathbf{z}_k \\ \mathbf{0} & u_{kk} \end{pmatrix},$$

where $\mathbf{f}_k, \mathbf{g}_k, \mathbf{m}_k, \mathbf{z}_k \in \mathbb{R}^{k-1}$, while $h_{kk}, u_{kk} \in \mathbb{R}$. Moreover, since H_k is upper Hessenberg, both $\mathbf{g}_k = h_{k,k-1}\mathbf{e}_{k-1}$ and $\mathbf{m}_k = l_{k,k-1}\mathbf{e}_{k-1}$ are multiples of the $(k-1)^{\text{st}}$ basis vector $\mathbf{e}_{k-1} \in \mathbb{R}^{k-1}$. Multiplying out $H_k = L_k U_k$ implies that we need only solve a single triangular linear system, via forward substitution, along with a pair of scalar linear equations, resulting in

$$L_{k-1}\mathbf{z}_k = \mathbf{f}_k, \qquad l_{k,k-1} = h_{k,k-1}/u_{k-1,k-1}, \qquad u_{kk} = h_{kk} - l_{k,k-1}u_{k-1,k}. \tag{9.109}$$

Remark. Suppose you happen to know a good initial guess $\mathbf{x}_0$ for the solution. The convergence can then be speeded up by setting $\widetilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$, which converts the original system to $A\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}$, where $\widetilde{\mathbf{b}} = \mathbf{r}_0 = \mathbf{b} - A\,\mathbf{x}_0$ is the initial residual. On applying the FOM algorithm to the modified system, the resulting $\widetilde{\mathbf{x}}_k \in \widetilde{V}^{(k)}$ in the Krylov subspaces generated by $\widetilde{\mathbf{b}}$ provide the improved approximations $\mathbf{x}_k = \widetilde{\mathbf{x}}_k + \mathbf{x}_0$ to the solution $\mathbf{x}$ to the original system.

The Conjugate Gradient Method

The most important case of the FOM algorithm is that in which the coefficient matrix A is symmetric, and hence, as noted above, H_k is symmetric, tridiagonal, which means that the system (9.107) can be quickly solved by the tridiagonal version of Gaussian Elimination, cf. (1.69–70). In particular, if $A > 0$ is positive definite, then so is $H_k > 0$, and the resulting algorithm is known as the *Conjugate Gradient Method*, often abbreviated CG, first introduced in 1952 by Hestenes and Stiefel, [**39**]. It is now the most widely used method for solving linear systems with positive definite coefficient matrices, e.g., those arising in the numerical solution to boundary value problems for elliptic systems of partial differential equations, [**8**, **61**].

There is a simpler direct way to formulate the CG algorithm, which is the one that is used in practice. First, we apply Theorems 1.29 and 1.34 to refine the factorization of the tridiagonal matrix:

$$H_k = L_k D_k L_k^T, \tag{9.110}$$

where L_k is lower unitriangular and D_k is diagonal. Let C_k be the $k\times k$ diagonal matrix with diagonal entries $c_j = \|\,\mathbf{r}_{j-1}\,\|$ for $j = 1, \ldots, k$, so that, according to (9.105),

$$Q_k C_k = R_k = \left(\,\mathbf{r}_0\ \mathbf{r}_1\ \ldots\ \mathbf{r}_{k-1}\,\right)$$

is the matrix of residual vectors. Define

$$W_k = (\, \mathbf{w}_1 \ \mathbf{w}_2 \ \ldots \ \mathbf{w}_k \,) = Q_k L_k^{-T} C_k = R_k V_k, \tag{9.111}$$

where the columns $\mathbf{w}_1, \ldots, \mathbf{w}_k$ of W_k are known as the *conjugate directions*, and where

$$V_k = C_k^{-1} L_k^{-T} C_k = \begin{pmatrix} 1 & s_1 & & & \\ & 1 & s_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & s_{k-1} \\ & & & & 1 \end{pmatrix}$$

is upper unitriangular. Note that, for $j \geq 1$, the $(j+1)^{\text{st}}$ column of the matrix equation $R_k = W_k V_k^{-1}$ implies

$$\mathbf{r}_j = \mathbf{w}_{j+1} - s_j \mathbf{w}_j. \tag{9.112}$$

We claim that the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$ are *conjugate*, which means that they mutually orthogonal with respect to the inner product[†] $\langle\!\langle \, \mathbf{v} \, , \mathbf{w} \, \rangle\!\rangle = \mathbf{v}^T A \mathbf{w}$ induced by A, and so

$$\langle\!\langle \, \mathbf{w}_i \, , \mathbf{w}_j \, \rangle\!\rangle = \mathbf{w}_i^T A \mathbf{w}_j = 0, \qquad i \neq j. \tag{9.113}$$

To verify (9.113), we use (9.110, 111) to compute the corresponding Gram matrix, whose entries are the inner products:

$$W_k^T A W_k = C_k L_k^{-1} Q_k^T A Q_k L_k^{-T} C_k = C_k L_k^{-1} H_k L_k^{-T} C_k = C_k D_k C_k = C_k^2 D_k,$$

the final result being a diagonal matrix. We deduce that all the off-diagonal entries of the Gram matrix $W_k^T A W_k$ vanish, which proves (9.113).

Let us write the k^{th} approximate solution $\mathbf{x}_k \in V^{(k)}$ in the form

$$\mathbf{x}_k = Q_k \mathbf{y}_k = W_k \mathbf{t}_k = t_1 \mathbf{w}_1 + \ \cdots \ + t_k \mathbf{w}_k, \qquad \text{where} \qquad \mathbf{t}_k = C_k^{-1} L_k^T \mathbf{y}_k.$$

As a consequence of (9.112) with[‡] k replacing j, along with (9.113), its residual vector $\mathbf{r}_k = \mathbf{b} - A\,\mathbf{x}_k$ satisfies

$$\begin{aligned} \langle\!\langle \, \mathbf{r}_k \, , \mathbf{w}_k \, \rangle\!\rangle &= \langle\!\langle \, \mathbf{w}_{k+1} - s_k \mathbf{w}_k \, , \mathbf{w}_k \, \rangle\!\rangle = -\, s_k \langle\!\langle \, \mathbf{w}_k \, , \mathbf{w}_k \, \rangle\!\rangle, \\ \langle\!\langle \, \mathbf{r}_k \, , \mathbf{w}_{k+1} \, \rangle\!\rangle &= \langle\!\langle \, \mathbf{w}_{k+1} - s_k \mathbf{w}_k \, , \mathbf{w}_{k+1} \, \rangle\!\rangle = \langle\!\langle \, \mathbf{w}_{k+1} \, , \mathbf{w}_{k+1} \, \rangle\!\rangle. \end{aligned} \tag{9.114}$$

The $(k+1)^{\text{st}}$ approximation can be written in the iterative form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + t_{k+1} \mathbf{w}_{k+1}, \tag{9.115}$$

meaning that we move from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ by adding a suitable scalar mutiple of the conjugate direction $\mathbf{w}_{k+1}$. The updated residual is

$$\mathbf{r}_{k+1} = \mathbf{b} - A\,\mathbf{x}_{k+1} = \mathbf{b} - A\,\mathbf{x}_k - t_{k+1} A \mathbf{w}_{k+1} = \mathbf{r}_k - t_{k+1} A \mathbf{w}_{k+1}. \tag{9.116}$$

[†] Of course, (9.113) defines a genuine inner product only if $A > 0$. On the other hand, the ensuing calculations only require symmetry of the coefficient matrix, although there is no guarantee that the resulting linear systems can be solved when A is not positive definite.

[‡] To be completely accurate, the resulting equation appears as the $(k+1)^{\text{st}}$ column of the subsequent matrix equations $R_l = W_l V_l^{-1}$ for all $l \geq k+1$.

Conjugate Gradient Method for Solving $A\mathbf{x} = \mathbf{b}$ with $A > 0$

```
start
    choose an initial guess $\mathbf{x}_0$, e.g., $\mathbf{x}_0 = \mathbf{0}$
    for $k = 0$ to $m - 1$
        set $\mathbf{r}_k = \mathbf{b} - A\,\mathbf{x}_k$
        if $\mathbf{r}_k = \mathbf{0}$ print "$\mathbf{x}_k$ is the exact solution"; end
        if $k = 0$ set $\mathbf{w}_1 = \mathbf{r}_0$ else set $\mathbf{w}_{k+1} = \mathbf{r}_k + \dfrac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}\,\mathbf{w}_k$
        set $\mathbf{x}_{k+1} = \mathbf{x}_k + \dfrac{\|\mathbf{r}_k\|^2}{\mathbf{w}_{k+1}^T A\,\mathbf{w}_{k+1}}\,\mathbf{w}_{k+1}$
    next $k$
end
```

Orthogonality of the residuals, (9.106), coupled with (9.114) implies

$$\begin{aligned} 0 = \mathbf{r}_k^T\mathbf{r}_{k+1} = \|\mathbf{r}_k\|^2 - t_{k+1}\mathbf{r}_k^T A\,\mathbf{w}_{k+1} &= \|\mathbf{r}_k\|^2 - t_{k+1}\langle\!\langle\,\mathbf{r}_k\,,\mathbf{w}_{k+1}\,\rangle\!\rangle \\ &= \|\mathbf{r}_k\|^2 - t_{k+1}\langle\!\langle\,\mathbf{w}_{k+1}\,,\mathbf{w}_{k+1}\,\rangle\!\rangle, \end{aligned}$$

hence

$$t_{k+1} = \frac{\|\mathbf{r}_k\|^2}{\langle\!\langle\,\mathbf{w}_{k+1}\,,\mathbf{w}_{k+1}\,\rangle\!\rangle} = \frac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{w}_{k+1}^T A\,\mathbf{w}_{k+1}}\,. \tag{9.117}$$

Finally, using (9.106) and (9.116, 117), with k replaced by $k-1$, yields

$$\|\mathbf{r}_k\|^2 = \mathbf{r}_k^T(\mathbf{r}_{k-1} - t_k A\,\mathbf{w}_k) = -\,t_k\,\mathbf{r}_k^T A\,\mathbf{w}_k = -\,t_k\,\langle\!\langle\,\mathbf{r}_k\,,\mathbf{w}_k\,\rangle\!\rangle = -\,\frac{\langle\!\langle\,\mathbf{r}_k\,,\mathbf{w}_k\,\rangle\!\rangle\,\|\mathbf{r}_{k-1}\|^2}{\langle\!\langle\,\mathbf{w}_k\,,\mathbf{w}_k\,\rangle\!\rangle}\,.$$

Thus, referring back to (9.114),

$$s_k = -\,\frac{\langle\!\langle\,\mathbf{r}_k\,,\mathbf{w}_k\,\rangle\!\rangle}{\langle\!\langle\,\mathbf{w}_k\,,\mathbf{w}_k\,\rangle\!\rangle} = \frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}\,. \tag{9.118}$$

The iterative equations (9.115, 117, 118) constitute the Conjugate Gradient algorithm, which is summarized in the accompanying pseudocode. The algorithm can also be applied if A is merely symmetric, although it may break down if the denominator $\mathbf{w}_{k+1}^T A\,\mathbf{w}_{k+1} = 0$, which will not occur in the positive definite case (why?). At each stage, $\mathbf{x}_k$ is the current approximation to the solution. The initial guess $\mathbf{x}_0$ can be chosen by the user, with $\mathbf{x}_0 = \mathbf{0}$ the default. The number of iterations $m \le n$ can be specified in advance; alternatively, one can impose a stopping criterion based on the size of the residual vector, $\|\mathbf{r}_k\|$, or, alternatively, the amount of change between successive iterates, as measured by, say, their distance $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ in either the Euclidean norm or the ∞ norm. Because the process is semi-direct, eventually $\mathbf{r}_k = 0$ for some $k \le n$, and so, in the absence of round-off errors, the result will be the exact solution to the system. Of course, in examples, one would not carry through the algorithm to the bitter end, since a decent approximation to the solution is typically obtained with relatively few iterations. For further developments and applications, see [**21**, **66**, **70**, **89**].

Remark. The reason for the name "conjugate gradient" is as follows. The term gradient stems from the minimization principle characterizing the solutions to linear systems with positive definite coefficient matrices. According to Theorem 5.2, if $A > 0$, the solution to the linear system $A\mathbf{x} = \mathbf{b}$ is the unique minimizer of the quadratic function

$$p(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}. \tag{9.119}$$

One approach to solving the system is to try to successively minimize $p(\mathbf{x})$ as much as possible. Suppose we find ourselves at a point $\mathbf{x}$ that is not the minimizer. In which direction should we travel? Multivariable calculus tells us that the gradient vector $\nabla p(\mathbf{x})$ of a function points in the direction of its steepest increase at the point, while its negative $-\nabla p(\mathbf{x})$ points in the direction of steepest decrease, [**2**, **78**]. The gradient of the particular quadratic function (9.119) is easily found:

$$-\nabla p(\mathbf{x}) = \mathbf{b} - A\mathbf{x} = \mathbf{r}.$$

Thus, the *residual vector* specifies the direction of steepest decrease in the quadratic function, and is thus a good choice of direction in which to head off in search of the true minimizer. (If one views the graph of p as a mountain range, then, at any given location $\mathbf{x}$ with elevation $p(\mathbf{x})$, the negative gradient $-\nabla p(\mathbf{x}) = \mathbf{r}$ points in the steepest downhill direction.) This idea leads to the *gradient descent algorithm*, in which each successive approximation $\mathbf{x}_k$ to the solution is obtained by going a certain distance in the residual direction:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + d_k\,\mathbf{r}_k, \qquad \text{where} \qquad \mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k. \tag{9.120}$$

The scalar factor d_k is to be specified so that the resulting $p(\mathbf{x}_{k+1})$ is as small as possible; in Exercise 9.6.14 you are asked to find this value. Gradient descent is a reasonable algorithm, and will lead to the solution in favorable situations. It is also effectively used to find minima of more general nonlinear functions. However, in certain circumstances, the iterative method based on gradient descent can take a long time to converge to an accurate approximation to the solution, and so is typically not competitive. To obtain the speedier Conjugate Gradient algorithm, we modify the gradient descent idea by requiring that the next descent direction be chosen so that it is *conjugate* to the preceding directions, i.e., satisfies (9.113). This idea can be used to produce an independent direct derivation of the Conjugate Gradient algorithm.

Example 9.51. Consider the linear system $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}.$$

The exact solution is $\mathbf{x}_\star = (2, 5, -6)^T$. Let us implement the method of conjugate gradients, starting with the initial guess $\mathbf{x}_0 = (0, 0, 0)^T$. The corresponding residual vector is merely $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = \mathbf{b} = (1, 2, -1)^T$. The first conjugate direction is $\mathbf{w}_1 = \mathbf{r}_0 = (1, 2, -1)^T$, and we use formula (9.115) to obtain the updated approximation to the solution

$$\mathbf{x}_1 = \mathbf{x}_0 + \frac{\|\mathbf{r}_0\|^2}{\langle\!\langle \mathbf{w}_1, \mathbf{w}_1 \rangle\!\rangle}\,\mathbf{w}_1 = \frac{6}{4}\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ 3 \\ -\frac{3}{2} \end{pmatrix},$$

noting that $\langle\!\langle \mathbf{w}_1, \mathbf{w}_1 \rangle\!\rangle = \mathbf{w}_1^T A\mathbf{w}_1 = 4$. For the next stage of the algorithm, we compute the corresponding residual $\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1 = \left(-\frac{1}{2}, -1, -\frac{5}{2}\right)^T$. The conjugate direction is

$$\mathbf{w}_2 = \mathbf{r}_1 + \frac{\|\mathbf{r}_1\|^2}{\|\mathbf{r}_0\|^2}\,\mathbf{w}_1 = \begin{pmatrix} -\frac{1}{2} \\ -1 \\ -\frac{5}{2} \end{pmatrix} + \frac{\frac{15}{2}}{6}\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix} = \begin{pmatrix} \frac{3}{4} \\ \frac{3}{2} \\ -\frac{15}{4} \end{pmatrix},$$

which, as designed, satisfies the conjugacy condition $\langle\!\langle\, \mathbf{w}_1\,, \mathbf{w}_2 \,\rangle\!\rangle = \mathbf{w}_1^T A\mathbf{w}_2 = 0$. Each entry of the ensuing approximation

$$\mathbf{x}_2 = \mathbf{x}_1 + \frac{\|\mathbf{r}_1\|^2}{\langle\!\langle\, \mathbf{w}_2\,, \mathbf{w}_2 \,\rangle\!\rangle}\,\mathbf{w}_2 = \begin{pmatrix} \frac{3}{2} \\ 3 \\ -\frac{3}{2} \end{pmatrix} + \frac{\frac{15}{2}}{\frac{27}{4}}\begin{pmatrix} \frac{3}{4} \\ \frac{3}{2} \\ -\frac{15}{4} \end{pmatrix} = \begin{pmatrix} \frac{7}{3} \\ \frac{14}{3} \\ -\frac{17}{3} \end{pmatrix} \simeq \begin{pmatrix} 2.3333 \\ 4.6667 \\ -5.6667 \end{pmatrix}$$

is now within a $\frac{1}{3}$ of the exact solution $\mathbf{x}_\star$.

Since we are dealing with a 3×3 system, we will recover the exact solution by one more iteration of the algorithm. The new residual is $\mathbf{r}_2 = \mathbf{b} - A\mathbf{x}_2 = \left(-\frac{4}{3}, \frac{2}{3}, 0\right)^T$. The final conjugate direction is

$$\mathbf{w}_3 = \mathbf{r}_2 + \frac{\|\mathbf{r}_2\|^2}{\|\mathbf{r}_1\|^2}\,\mathbf{w}_2 = \begin{pmatrix} -\frac{4}{3} \\ \frac{2}{3} \\ 0 \end{pmatrix} + \frac{\frac{20}{9}}{\frac{15}{2}}\begin{pmatrix} \frac{3}{4} \\ \frac{3}{2} \\ -\frac{15}{4} \end{pmatrix} = \begin{pmatrix} -\frac{10}{9} \\ \frac{10}{9} \\ -\frac{10}{9} \end{pmatrix},$$

which, as you can check, is conjugate to both $\mathbf{w}_1$ and $\mathbf{w}_2$. The solution is obtained from

$$\mathbf{x}_3 = \mathbf{x}_2 + \frac{\|\mathbf{r}_2\|^2}{\langle\!\langle\, \mathbf{w}_3\,, \mathbf{w}_3 \,\rangle\!\rangle}\,\mathbf{w}_3 = \begin{pmatrix} \frac{7}{3} \\ \frac{14}{3} \\ -\frac{17}{3} \end{pmatrix} + \frac{\frac{20}{9}}{\frac{200}{27}}\begin{pmatrix} -\frac{10}{9} \\ \frac{10}{9} \\ -\frac{10}{9} \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \\ -6 \end{pmatrix}.$$

The Generalized Minimal Residual Method

A natural alternative to the Galerkin weak approach is to try to directly minimize the norm of the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ when the approximate solution $\mathbf{x}$ is required to lie in a specified subspace $\mathbf{x} \in V$. When V is a Krylov subspace, this idea results in the Generalized Minimal Residual Method (usually abbreviated GMRES), which was developed by the Algerian and American mathematicians/computer scientists[†] Yousef Saad and Martin Schultz, [**71**].

As in the FOR Method, we choose the Krylov subspaces generated by $\mathbf{b}$, the right-hand side of the system to be solved, but now seek the vector $\mathbf{x}_k^\star \in V^{(k)}$ that minimizes the Euclidean norm $\|A\mathbf{x} - \mathbf{b}\|$ over all vectors $\mathbf{x} \in V^{(k)}$. This approach corresponds to the initial approximation $\mathbf{x}_0 = \mathbf{0} \in V^{(1)}$; as before, if we know a better initial guess $\mathbf{x}_0$, we set $\widetilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$, which converts the original system to $A\widetilde{\mathbf{x}} = \widetilde{\mathbf{b}}$, where $\widetilde{\mathbf{b}} = \mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ is the initial residual, and then apply the method to the new system.

Again, we express the vectors

$$\mathbf{x}_k = y_1\mathbf{u}_1 + \cdots + y_k\mathbf{u}_k = Q_k\mathbf{y} \in V^{(k)}$$

[†] Coincidentally, the first author of the book you are reading is at the same university, Minnesota, as Saad, and had the same thesis advisor, Garrett Birkhoff, as Schultz.

as linear combinations of the orthonormal Arnoldi basis vectors, with coefficients $\mathbf{y}_k = (\,y_1, \ldots, y_k\,)^T \in \mathbb{R}^k$. In view of (9.100) and (9.97), with k replaced by $k+1$, the squared residual norm is given by

$$\begin{aligned}\|\,\mathbf{r}\,\|_k^2 &= \|\,A\,\mathbf{x}_k - \mathbf{b}\,\|^2 = \|\,A\,Q_k\,\mathbf{y}_k - \mathbf{b}\,\|^2 = \|\,Q_{k+1}\widetilde{H}_k\mathbf{y}_k - \mathbf{b}\,\|^2 \\ &= (Q_{k+1}\widetilde{H}_k\,\mathbf{y}_k - \mathbf{b})^T(Q_{k+1}\widetilde{H}_k\,\mathbf{y}_k - \mathbf{b}) = \mathbf{y}_k^T\widetilde{H}_k^T\widetilde{H}_k\,\mathbf{y}_k - 2\,\mathbf{y}_k^T\widetilde{H}_k^TQ_{k+1}^T\mathbf{b} + \|\,\mathbf{b}\,\|^2 \\ &= \mathbf{y}_k^T\widetilde{H}_k^T\widetilde{H}_k\,\mathbf{y}_k - 2\,\mathbf{y}_k^T\widetilde{H}_k^T\,\mathbf{c}_k + \|\,\mathbf{c}_k\,\|^2 = \|\,\widetilde{H}_k\,\mathbf{y}_k - \mathbf{c}_k\,\|^2,\end{aligned} \tag{9.121}$$

where, according to (9.107) again with k replaced by $k+1$,

$$\mathbf{c}_k = Q_{k+1}^T\mathbf{b} = \|\,\mathbf{b}\,\|\,\mathbf{e}_1 \in \mathbb{R}^{k+1}, \qquad \text{so that} \qquad \|\,\mathbf{c}_k\,\| = \|\,\mathbf{b}\,\|. \tag{9.122}$$

We deduce that minimizing $\|\,A\,\mathbf{x} - \mathbf{b}\,\|$ over all $\mathbf{x} \in V^{(k)}$ is the same as minimizing $\|\,\widetilde{H}_k\,\mathbf{y} - \mathbf{c}_k\,\|$ over all $\mathbf{y} \in \mathbb{R}^k$. The latter is a standard least squares minimization problem, whose solution $\mathbf{y}_k$ is found by solving the corresponding normal equations

$$\widetilde{H}_k^T\widetilde{H}_k\,\mathbf{y}_k = \widetilde{H}_k^T\mathbf{c}_k = \|\,\mathbf{b}\,\|\,\widetilde{H}_k^T\mathbf{e}_1 = \|\,\mathbf{b}\,\|\;(\,h_{11}, h_{12},\;\ldots\;, h_{1k}\,)^T\,. \tag{9.123}$$

Solving (9.123), produces the desired minimizer $\mathbf{x}_k = Q_k\mathbf{y}_k \in V^{(k)}$, and hence the desired approximation to the solution to the original linear system.

The result of this calculation is the *Generalized Minimal Residual Method* (GMRES) algorithm. To successively approximate the solution to $A\,\mathbf{x} = \mathbf{b}$, on the k^{th} iteration, we set $\mathbf{c} = \|\,\mathbf{b}\,\|\,\mathbf{e}_1$, and then perform the following steps:

(a) calculate $\mathbf{u}_k$ and $\widetilde{H}_k$ using the Arnoldi Method;

(b) use least squares to find the vector $\mathbf{y} = \mathbf{y}_k$ that minimizes $\|\,\widetilde{H}_k\,\mathbf{y} - \mathbf{c}\,\|$;

(c) let $\mathbf{x}_k = Q_k\mathbf{y}_k$ be the k^{th} approximate solution.

The process is repeated until the residual norm $\|\,\mathbf{r}_k\,\| = \|\,A\,\mathbf{x}_k - \mathbf{b}\,\| = \|\,\widetilde{H}_k\,\mathbf{y} - \mathbf{c}\,\|$ is below a pre-assigned threshhold. Again, because of the iterative structure of the Krylov vectors, and hence the upper Hessenberg matrices H_k, knowing the solution to the order k minimization roblem allows one to rather quickly construct that of the order $k+1$ version. As with all Krylov methods, GMRES is a semi-direct method and hence, if performed in exact arithmetic, will eventually produce the exact solution once the Krylov stabilization order is reached. As with FOM/CG, this is rarely required, and one typically imposes a stopping criterion based on either the norm of the residual vector or the size of the difference between successive iterates $\|\,\mathbf{x}_{k+1} - \mathbf{x}_k\,\|$. The method works very well in practice, particularly with the sparse coefficient matrices arising in many numerical solution algorithms for partial differential equations and beyond, including finite difference, finite element, collocation, and multipole expansion.

Exercises

9.6.1. Find an orthonormal basis for the Krylov subspaces $V^{(1)}, V^{(2)}, V^{(3)}$ for the following matrices and vectors:

(a) $A = \begin{pmatrix} 0 & 1 \\ 3 & 1 \end{pmatrix}$, $\mathbf{v} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$; ($b$) $A = \begin{pmatrix} 2 & 2 & -1 \\ 2 & -1 & 0 \\ 2 & 1 & 3 \end{pmatrix}$, $\mathbf{v} = \begin{pmatrix} -1 \\ 2 \\ 0 \end{pmatrix}$;

(c) $A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 2 & -3 \\ 2 & -1 & 0 \end{pmatrix}$, $\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$; ($d$) $A = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$, $\mathbf{v} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$.

9.6.2. Let $\mathbf{v} = \mathbf{x} + \mathrm{i}\,\mathbf{y}$ be an eigenvector corresponding to a complex, non-real eigenvalue of the real $n \times n$ matrix A. (*a*) Prove that the Krylov subspaces $V^{(k)}$ for $k \geq 2$ generated by both $\mathbf{x}$ and $\mathbf{y}$ are all two-dimensional. (*b*) Is the converse valid? Specifically, if $\dim V^{(3)} = 2$, then all $V^{(k)}$ are two-dimensional for $k \geq 1$ and spanned by the real and imaginary parts of a complex eigenvector of A.

♢ 9.6.3. (*a*) Prove that the dimension of a Krylov subspace is bounded by the degree of the minimal polynomial of the matrix A, as defined in Exercise 8.6.23. (*b*) Is there always a Krylov subspace whose dimension equals the degree of the minimal polynomial?

9.6.4. *True or false*: A Krylov subspace is an invariant subspace for the matrix A.

9.6.5. Prove that the invertibility of the coefficient matrix $S^T A S$ in (9.104) depends only on the subspace V and not on the choice of basis thereof.

♢ 9.6.6. Prove that (9.92, 93, 94) give the same Arnoldi vectors $\mathbf{u}_k$ and the same coefficients h_{jk} when computed exactly.

9.6.7. Solve the following linear systems by the Conjugate Gradient Method, keeping track of the residual vectors and solution approximations as you iterate.

$$(a)\ \begin{pmatrix} 3 & -1 \\ -1 & 5 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad (b)\ \begin{pmatrix} 6 & 2 & 1 \\ 2 & 3 & -1 \\ 1 & -1 & 2 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}, \quad (c)\ \begin{pmatrix} 6 & -1 & -3 \\ -1 & 7 & 4 \\ -3 & 4 & 9 \end{pmatrix} \mathbf{u} = \begin{pmatrix} -1 \\ -2 \\ 7 \end{pmatrix},$$

$$(d)\ \begin{pmatrix} 6 & -1 & -1 & 5 \\ -1 & 7 & 1 & -1 \\ -1 & 1 & 3 & -3 \\ 5 & -1 & -3 & 6 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 1 \\ 2 \\ 0 \\ -1 \end{pmatrix}, \qquad (e)\ \begin{pmatrix} 5 & 1 & 1 & 1 \\ 1 & 5 & 1 & 1 \\ 1 & 1 & 5 & 1 \\ 1 & 1 & 1 & 5 \end{pmatrix} \mathbf{u} = \begin{pmatrix} 4 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

♣ 9.6.8. Use the Conjugate Gradient Method to solve the system in Exercise 9.4.33. How many iterations do you need to obtain the solution that is accurate to 2 decimal places? How does this compare to the Jacobi and SOR Methods?

♣ 9.6.9. According to Example 3.39, the $n \times n$ Hilbert matrix H_n is positive definite, and hence we can apply the Conjugate Gradient Method to solve the linear system $H_n \mathbf{u} = \mathbf{f}$. For the values $n = 5, 10, 30$, let $\mathbf{u}^\star \in \mathbb{R}^n$ be the vector with all entries equal to 1.
(*a*) Compute $\mathbf{f} = H_n \mathbf{u}^\star$. (*b*) Use Gaussian Elimination to solve $H_n \mathbf{u} = \mathbf{f}$. How close is your solution to $\mathbf{u}^\star$? (*c*) Does pivoting improve the solution in part (*b*)?
(*d*) Does the conjugate gradient algorithm do any better?

9.6.10. Try applying the Conjugate Gradient algorithm to the system $-x + 2y + z = -2$, $y + 2z = 1$, $3x + y - z = 1$. Do you obtain the solution? Why or why not?

9.6.11. *True or false*: If the residual vector $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ satisfies $\|\mathbf{r}\| < .01$, then $\mathbf{x}$ approximates the true solution to within two decimal places.

♢ 9.6.12. How many arithmetic operations are needed to implement one iteration of the Conjugate Gradient Method? How many iterations can you perform before the method becomes more work than direct Gaussian Elimination?

Remark. If the matrix is sparse, the number of operations can decrease dramatically.

♢ 9.6.13. Fill in the details in a direct derivation of the Conjugate Gradient algorithm following the ideas outlined in the text: starting with the initial guess $\mathbf{x}_0$ and corresponding residual vector $\mathbf{w}_1 = \mathbf{r}_0 = \mathbf{b}$, at the k^{th} step in the algorithm, given the approximation $\mathbf{x}_k$ and residual $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, the k^{th} conjugate direction is chosen so that $\mathbf{w}_{k+1} = \mathbf{r}_k + s_k \mathbf{w}_k$ satisfies the conjugacy conditions (9.113). The next approximation $\mathbf{x}_{k+1} = \mathbf{x}_k + t_{k+1} \mathbf{w}_{k+1}$ is chosen so that its residual $\mathbf{r}_{k+1} = \mathbf{b} - A\mathbf{x}_{k+1}$ is as small as possible.

♢ 9.6.14. In (9.120), find the value of d_k that minimizes $p(\mathbf{x}_{k+1})$.

♠ 9.6.15. Use the direct gradient descent algorithm (9.120) using the value of d_k found in Exercise 9.6.14 to solve the linear systems in Exercise 9.6.7. Compare the speed of convergence with that of the Conjugate Gradient Method.

♣ 9.6.16. Use GMRES to solve the system in Exercise 9.4.33. Compare the rate of convergence with the CG algorithm in Exercise 9.6.8.

♣ 9.6.17. Is GMRES able to solve the system in Exercise 9.6.10?

9.6.18. Explain in what sense the GMRES approximation $\mathbf{x}_{k+1}$ of order $k+1$ is a better approximation to the true solution than that of order k, namely $\mathbf{x}_k$.

9.6.19. (*a*) Explain what happens to the GMRES algorithm if the right-hand side $\mathbf{b}$ of the linear system $A\,\mathbf{x} = \mathbf{b}$ is an eigenvector of A. (*b*) More generally, prove that if the Krylov subspaces generated by $\mathbf{b}$ stabilize at order m, then the solution ot the linear system lies in $V^{(m)}$ and so the GMRES algorithm converges to the solution at order m.

9.7 Wavelets

Trigonometric Fourier series, both continuous and discrete, are amazingly powerful, but they do suffer from one potentially serious defect. The complex exponential basis functions $e^{\,\mathrm{i}\,k\,x} = \cos k\,x + \mathrm{i}\,\sin k\,x$ are spread out over the entire interval $[-\pi, \pi]$, and so are not well suited to processing localized signals — meaning data that are concentrated in a relatively small regions. Ideally, one would like to construct a system of functions that is orthogonal, and so has all the advantages of the Fourier basis functions, but, in addition, adapts to localized structures in signals. This dream was the inspiration for the development of the modern theory of wavelets.

The Haar Wavelets

Although the modern era of wavelets started in the mid 1980's, the simplest example of a wavelet basis was discovered by the Hungarian mathematician Alfréd Haar in 1910, [**35**]. We consider the space of functions (signals) defined the interval $[0,1]$, equipped with the standard L^2 inner product

$$\langle\, f\,,g\,\rangle = \int_0^1 f(x)\,g(x)\,dx. \tag{9.124}$$

The usual scaling arguments can be used to adapt the wavelet formulas to any other interval.

The *Haar wavelets* are certain piecewise constant functions. The initial four are graphed in Figure 9.6. The first is the *box function*

$$\varphi_1(x) = \varphi(x) = \begin{cases} 1, & 0 < x \le 1, \\ 0, & \text{otherwise}, \end{cases} \tag{9.125}$$

known as the *scaling function*, for reasons that shall appear shortly. Although we are interested in the value of $\varphi(x)$ only on the interval $[0,1]$, it will be convenient to extend it, and all the other wavelets, to be zero outside the basic interval. The second Haar function

$$\varphi_2(x) = w(x) = \begin{cases} 1, & 0 < x \le \frac{1}{2}, \\ -1, & \frac{1}{2} < x \le 1, \\ 0, & \text{otherwise}, \end{cases} \tag{9.126}$$

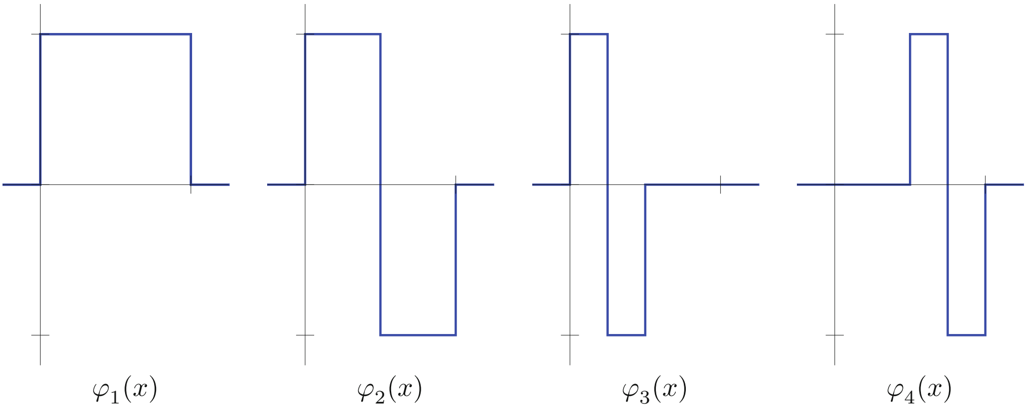


Figure 9.6. The First Four Haar Wavelets.

is known as the *mother wavelet*. The third and fourth Haar functions are compressed versions of the mother wavelet:

$$\varphi_3(x) = w(2x) = \begin{cases} 1, & 0 < x \le \frac{1}{4}, \\ -1, & \frac{1}{4} < x \le \frac{1}{2}, \\ 0, & \text{otherwise}, \end{cases} \qquad \varphi_4(x) = w(2x-1) = \begin{cases} 1, & \frac{1}{2} < x \le \frac{3}{4}, \\ -1, & \frac{3}{4} < x \le 1, \\ 0, & \text{otherwise}, \end{cases}$$

called *daughter wavelets*. One can easily check, by direct evaluation of the integrals, that the four Haar wavelet functions are orthogonal with respect to the L^2 inner product (9.124): $\langle \varphi_i , \varphi_j \rangle = 0$ when $i \ne j$.

The scaling transformation $x \mapsto 2x$ serves to compress the wavelet function, while the translation $2x \mapsto 2x - 1$ moves the compressed version to the right by a half a unit. Furthermore, we can represent the mother wavelet by compressing and translating the scaling function:

$$w(x) = \varphi(2x) - \varphi(2x-1). \tag{9.127}$$

It is these two operations of scaling and compression — coupled with the all-important orthogonality — that underlies the power of wavelets.

The Haar wavelets have an evident discretization. If we decompose the interval $(0,1]$ into the four subintervals

$$\left(0, \tfrac{1}{4}\right], \qquad \left(\tfrac{1}{4}, \tfrac{1}{2}\right], \qquad \left(\tfrac{1}{2}, \tfrac{3}{4}\right], \qquad \left(\tfrac{3}{4}, 1\right], \tag{9.128}$$

on which the four wavelet functions are constant, then we can represent each of them by a vector in $\mathbb{R}^4$ whose entries are the values of each wavelet function sampled at the left endpoint of each subinterval. In this manner, we obtain the wavelet sample vectors

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad \mathbf{v}_2 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}, \qquad \mathbf{v}_3 = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{v}_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}. \tag{9.129}$$

which form the orthogonal wavelet basis of $\mathbb{R}^4$ we first encountered in Examples 2.35 and 4.10. Orthogonality of the vectors (9.129) with respect to the standard Euclidean dot product is equivalent to orthogonality of the Haar wavelet functions with respect to the inner product (9.124). Indeed, if

$$f(x) \;\sim\; \mathbf{f} = (f_1, f_2, f_3, f_4) \qquad \text{and} \qquad g(x) \;\sim\; \mathbf{g} = (g_1, g_2, g_3, g_4)$$

are *piecewise constant* real functions that achieve the indicated values on the four subintervals (9.128), then their L^2 inner product

$$\langle\, f\,, g\,\rangle = \int_0^1 f(x)\, g(x)\, dx = \tfrac{1}{4}\big(\, f_1\, g_1 + f_2\, g_2 + f_3\, g_3 + f_4\, g_4\,\big) = \tfrac{1}{4}\, \mathbf{f}\cdot\mathbf{g},$$

is equal to the averaged dot product of their sample values — the real form of the inner product (5.104) that was used in the discrete Fourier transform.

Since the vectors (9.129) form an orthogonal basis of $\mathbb{R}^4$, we can uniquely decompose such a piecewise constant function as a linear combination of wavelets

$$f(x) = c_1\, \varphi_1(x) + c_2\, \varphi_2(x) + c_3\, \varphi_3(x) + c_4\, \varphi_4(x),$$

or, equivalently, in terms of the sample vectors,

$$\mathbf{f} = c_1 \mathbf{v}_1 + c_2 \mathbf{v}_2 + c_3 \mathbf{v}_3 + c_4 \mathbf{v}_4.$$

The required coefficients

$$c_k = \frac{\langle\, f\,, \varphi_k\,\rangle}{\|\, \varphi_k \,\|^2} = \frac{\mathbf{f}\cdot\mathbf{v}_k}{\|\, \mathbf{v}_k \,\|^2}$$

are fixed by our usual orthogonality formula (4.7). Explicitly,

$$\begin{aligned} c_1 &= \tfrac{1}{4}\,(f_1 + f_2 + f_3 + f_4), & \qquad c_3 &= \tfrac{1}{2}\,(f_1 - f_2), \\ c_2 &= \tfrac{1}{4}\,(f_1 + f_2 - f_3 - f_4), & \qquad c_4 &= \tfrac{1}{2}\,(f_3 - f_4). \end{aligned}$$

Before proceeding to the more general case, let us introduce an important analytical definition that quantifies precisely how localized a function is.

Definition 9.52. The *support* of a function $f(x)$, written $\operatorname{supp} f$, is the closure of the set where $f(x) \neq 0$.

Thus, a point will belong to the support of $f(x)$, if f is not zero there, or at least is not zero at nearby points. More precisely:

Lemma 9.53. If $f(a) \neq 0$, then $a \in \operatorname{supp} f$. More generally, a point $a \in \operatorname{supp} f$ if and only if there exists a convergent sequence $x_n \to a$ such that $f(x_n) \neq 0$. Conversely, $a \notin \operatorname{supp} f$ if and only if $f(x) \equiv 0$ on an interval $a - \delta < x < a + \delta$ for some $\delta > 0$.

Intuitively, the smaller the support of a function, the more localized it is. For example, the support of the Haar mother wavelet (9.126) is $\operatorname{supp} w = [0, 1]$ — the point $x = 0$ is included, even though $w(0) = 0$, because $w(x) \neq 0$ at nearby points. The two daughter wavelets have smaller support:

$$\operatorname{supp} \varphi_3 = \left[\, 0\,, \tfrac{1}{2}\,\right], \qquad \operatorname{supp} \varphi_4 = \left[\, \tfrac{1}{2}\,, 1\,\right],$$

and so are twice as localized.

The effect of scalings and translations on the support of a function is easily discerned.

Lemma 9.54. If $\operatorname{supp} f = [a, b]$, and

$$g(x) = f(r\,x - \delta), \qquad \text{then} \qquad \operatorname{supp} g = \left[\, \frac{a+\delta}{r}\,, \frac{b+\delta}{r}\,\right].$$

In other words, scaling x by a factor r compresses the support of the function by a factor $1/r$, while translating x translates the support of the function.

The key requirement for a wavelet basis is that it contains functions with arbitrarily small support. To this end, the full Haar wavelet basis is obtained from the mother wavelet by iterating the scaling and translation processes. We begin with the scaling function

$$\varphi(x), \tag{9.130}$$

from which we construct the mother wavelet via (9.127). For each "generation" $j \geq 0$, we form the wavelet offspring by first compressing the mother wavelet so that its support fits into an interval of length 2^{-j},

$$w_{j,0}(x) = w(2^j x), \qquad \text{so that} \qquad \operatorname{supp} w_{j,0} = [0, 2^{-j}], \tag{9.131}$$

and then translating $w_{j,0}$ so as to fill up the entire interval $[0,1]$ by 2^j subintervals, each of length 2^{-j}, defining

$$w_{j,k}(x) = w_{j,0}(x-k) = w(2^j x - k), \qquad \text{where} \qquad k = 0, 1, \ldots, 2^j - 1. \tag{9.132}$$

Lemma 9.54 implies that $\operatorname{supp} w_{j,k} = [\, 2^{-j} k, 2^{-j}(k+1) \,]$, and so the combined supports of all the j^{th} generation of wavelets is the entire interval: $\displaystyle\bigcup_{k=0}^{2^j-1} \operatorname{supp} w_{j,k} = [0,1]$. The primal generation, $j=0$, consists of just the mother wavelet

$$w_{0,0}(x) = w(x).$$

The first generation, $j = 1$, consists of the two daughter wavelets already introduced as φ_3 and φ_4, namely

$$w_{1,0}(x) = w(2x), \qquad\qquad w_{1,1}(x) = w(2x-1).$$

The second generation, $j = 2$, appends four additional granddaughter wavelets to our basis:

$$w_{2,0}(x) = w(4x), \quad w_{2,1}(x) = w(4x-1), \quad w_{2,2}(x) = w(4x-2), \quad w_{2,3}(x) = w(4x-3).$$

The 8 Haar wavelets $\varphi, w_{0,0}, w_{1,0}, w_{1,1}, w_{2,0}, w_{2,1}, w_{2,2}, w_{2,3}$ are constant on the 8 subintervals of length $\frac{1}{8}$, taking the successive sample values indicated by the columns of the *wavelet matrix*

$$W_8 = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \end{pmatrix}. \tag{9.133}$$

Orthogonality of the wavelets is manifested in the orthogonality of the columns of W_8. (Unfortunately, terminological constraints prevent us from calling W_8 an orthogonal matrix, because its columns are not orthonormal!)

The n^{th} stage consists of 2^{n+1} different wavelet functions comprising the scaling functions and all the generations up to the n^{th} : $w_0(x) = \varphi(x)$ and $w_{j,k}(x)$ for $0 \leq j \leq n$ and $0 \leq k < 2^j$. They are all constant on each subinterval of length 2^{-n-1}.

Theorem 9.55. The wavelet functions $\varphi(x)$, $w_{j,k}(x)$ form an orthogonal system with respect to the inner product (9.124).

Proof: First, note that each wavelet $w_{j,k}(x)$ is equal to $+1$ on an interval of length 2^{-j-1} and to -1 on an adjacent interval of the same length. Therefore,

$$\langle\, w_{j,k}\,,\varphi\,\rangle = \int_0^1 w_{j,k}(x)\,dx = 0, \tag{9.134}$$

since the $+1$ and -1 contributions cancel each other. If two different wavelets $w_{j,k}$ and $w_{l,m}$ with, say $j \leq l$, have supports that are either disjoint, or just overlap at a single point, then their product $w_{j,k}(x)\,w_{l,m}(x) \equiv 0$, and so their inner product is clearly zero:

$$\langle\, w_{j,k}\,, w_{l,m}\,\rangle = \int_0^1 w_{j,k}(x)\,w_{l,m}(x)\,dx = 0.$$

Otherwise, except in the case when the two wavelets are identical, the support of $w_{l,m}$ is entirely contained in an interval where $w_{j,k}$ is constant, and so $w_{j,k}(x)\,w_{l,m}(x) = \pm\, w_{l,m}(x)$. Therefore, by (9.134),

$$\langle\, w_{j,k}\,, w_{l,m}\,\rangle = \int_0^1 w_{j,k}(x)\,w_{l,m}(x)\,dx = \pm \int_0^1 w_{l,m}(x)\,dx = 0.$$

Finally, we compute

$$\|\,\varphi\,\|^2 = \int_0^1 dx = 1, \qquad\qquad \|\,w_{j,k}\,\|^2 = \int_0^1 w_{j,k}(x)^2\,dx = 2^{-j}. \tag{9.135}$$

The second formula follows from the fact that $|\,w_{j,k}(x)\,| = 1$ on an interval of length 2^{-j} and is 0 elsewhere. *Q.E.D.*

The *wavelet series* of a signal $f(x)$ is given by

$$f(x) \;\sim\; c_0\,\varphi(x) \;+\; \sum_{j=0}^{\infty}\sum_{k=0}^{2^j-1} c_{j,k}\,w_{j,k}(x). \tag{9.136}$$

Orthogonality implies that the wavelet coefficients $c_0, c_{j,k}$ can be immediately computed using the standard inner product formula coupled with (9.135):

$$\begin{aligned} c_0 &= \frac{\langle\, f\,,\varphi\,\rangle}{\|\,\varphi\,\|^2} = \int_0^1 f(x)\,dx, \\ c_{j,k} &= \frac{\langle\, f\,, w_{j,k}\,\rangle}{\|\,w_{j,k}\,\|^2} = 2^j \int_{2^{-j}k}^{2^{-j}k+2^{-j-1}} f(x)\,dx \;-\; 2^j \int_{2^{-j}k+2^{-j-1}}^{2^{-j}(k+1)} f(x)\,dx. \end{aligned} \tag{9.137}$$

The convergence properties of the Haar wavelet series (9.136) are similar to those of Fourier series, [**61**, **77**]; full details can be found [**18**, **88**].

Example 9.56. In Figure 9.7, we plot the Haar expansions of the signal displayed in the first plot. The following plots show the partial sums for the Haar wavelet series (9.136) over $j = 0, \ldots, r$ with $r = 2, 3, 4, 5, 6$. Since the wavelets are themselves discontinuous, they do not have any difficulty converging to a discontinuous function. On the other hand, it takes quite a few wavelets to begin to accurately reproduce the signal — in the last plot, we are combining a total of $2^6 = 64$ Haar wavelets.

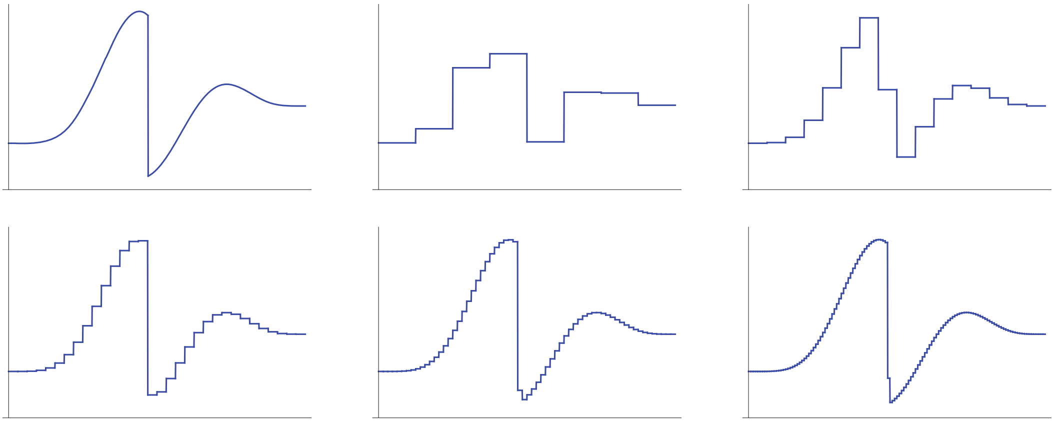


Figure 9.7. Haar Wavelet Expansion.

Exercises

♠ 9.7.1. Let $f(x) = x$. (a) Determine its Haar wavelet coefficients $c_{j,k}$. (b) Graph the partial sums $s_r(x)$ of the Haar wavelet series (9.136) where j goes from 0 to $r = 2$, 5, and 10. Compare your graphs with that of f and discuss what you observe. Is the series converging to the function? Can you prove this? (c) What is the maximal deviation $\| f - s_r \|_\infty = \max\{\, | f(x) - s_r(x) | \mid 0 \le x \le 1 \,\}$ for each of your partial sums?

♠ 9.7.2. Answer Exercise 9.7.1 for the functions

$$(a)\ \ x^2 - x, \qquad (b)\ \ \cos \pi x, \qquad (c)\ \begin{cases} e^{-x}, & 0 < x < \frac{1}{2}, \\ -e^{-x}, & \frac{1}{2} < x < 1. \end{cases}$$

♣ 9.7.3. In this exercise, we investigate the compression capabilities of the Haar wavelets. Let

$$f(x) = \begin{cases} -x, & 0 \le x \le \frac{1}{3}\pi, \\ x - \frac{2}{3}\pi, & \frac{1}{3}\pi \le x \le \frac{4}{3}\pi, \\ -x + 2\pi, & \frac{4}{3}\pi \le x \le 2\pi, \end{cases} \qquad \text{represent a signal defined on } 0 \le x \le 1. \text{ Let}$$

$s_r(x)$ denote the n^{th} partial sum, from $j = 0$ to r, of the Haar wavelet series (9.136). (a) How many different Haar wavelet coefficients $c_{j,k}$ appear in $s_r(x)$? If our criterion for compression is that $\| f - s_r \|_\infty < \varepsilon$, how large do you need to choose r when $\varepsilon = .1$? $\varepsilon = .01$? $\varepsilon = .001$? (b) Compare the Haar wavelet compression with the discrete Fourier method of Exercise 5.6.10.

♡ 9.7.4. (a) Explain why the wavelet expansion (9.136) defines a linear transformation on $\mathbb{R}^n$ that takes a wavelet coefficient vector $\mathbf{c} = \left(c_0, c_1, \ldots, c_{n-1} \right)^T$ to the corresponding sample vector $\mathbf{f} = \left(f_0, f_1, \ldots, f_{n-1} \right)^T$. (b) According to Theorem 7.5, the wavelet map must be given by matrix multiplication $\mathbf{f} = W_n \mathbf{c}$ by a 2×2^n matrix $W = W_n$. Construct W_2, W_3 and W_4. (c) Prove that the columns of W_n are obtained as the values of the wavelet basis functions on the 2^n sample intervals. (d) Prove that the columns of W_n are orthogonal. (e) Is W_n an orthogonal matrix? Find a formula for W_n^{-1}. (f) Explain why the wavelet transform is given by the linear map, $\mathbf{c} = W_n^{-1} \mathbf{f}$.

♠ 9.7.5. Test the noise removal features of the Haar wavelets by adding random noise to one of the functions in Exercises 9.7.1 and 9.7.2, computing the wavelet series, and then setting the high "frequency" modes to zero. What do you observe? Is this a reasonable denoising algorithm when compared with a Fourier method?

9.7.6. Write the Haar scaling function and mother wavelet as linear combinations of step functions.

◇ 9.7.7. Prove Lemma 9.54.

Modern Wavelets

The main defect of the Haar wavelets is that they do not provide a very efficient means of representing even very simple functions — it takes quite a large number of wavelets to reproduce signals with any degree of precision. The reason for this is that the Haar wavelets are piecewise constant, and so even an affine function $y = \alpha\, x + \beta$ requires many sample values, and hence a relatively extensive collection of Haar wavelets, to be accurately reproduced. In particular, compression and denoising algorithms based on Haar wavelets are either insufficiently precise or hopelessly inefficient, and hence of minor practical value.

For a long time it was thought that it was impossible to simultaneously achieve the requirements of localization, orthogonality and accurate reproduction of simple functions. The breakthrough came in 1988, when the Dutch mathematician Ingrid Daubechies produced the first examples of wavelet bases that realized all three basic criteria. Since then, wavelets have developed into a sophisticated and burgeoning industry with major impact on modern technology. Significant applications include compression, storage and recognition of fingerprints in the FBI's data base, and the JPEG2000 image format, which, unlike earlier Fourier-based JPEG standards, incorporates wavelet technology in its image compression and reconstruction algorithms. In this section, we will present a brief outline of the basic ideas underlying Daubechies' remarkable construction.

The recipe for any wavelet system involves two basic ingredients — a scaling function and a mother wavelet. The latter can be constructed from the scaling function by a prescription similar to that in (9.127), and therefore we first concentrate on the properties of the scaling function. The key requirement is that the scaling function must solve a *dilation equation* of the form

$$\varphi(x) = \sum_{k=0}^{p} c_k\, \varphi(2\,x - k) = c_0\, \varphi(2\,x) + c_1\, \varphi(2\,x - 1) + \;\cdots\; + c_p\, \varphi(2\,x - p) \qquad (9.138)$$

for some collection of constants $c_0, \ldots, c_p$. The dilation equation relates the function $\varphi(x)$ to a finite linear combination of its compressed translates. The coefficients $c_0, \ldots, c_p$ are not arbitrary, since the properties of orthogonality and localization will impose certain rather stringent requirements.

Example 9.57. The Haar or box scaling function (9.125) satisfies the dilation equation (9.138) with $c_0 = c_1 = 1$, namely

$$\varphi(x) = \varphi(2\,x) + \varphi(2\,x - 1). \qquad (9.139)$$

We recommend that you convince yourself of the validity of this identity before continuing.

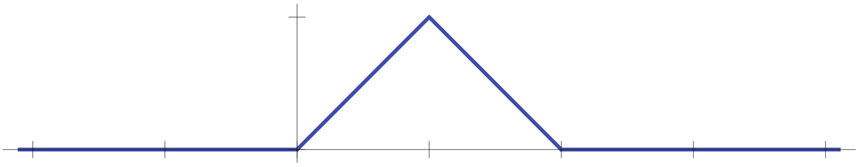


Figure 9.8. The Hat Function.

Example 9.58. Another example of a scaling function is the *hat function*

$$\varphi(x) = \begin{cases} x, & 0 \le x \le 1, \\ 2 - x, & 1 \le x \le 2, \\ 0, & \text{otherwise}, \end{cases} \tag{9.140}$$

graphed in Figure 9.8. The hat function satisfies the dilation equation

$$\varphi(x) = \tfrac{1}{2}\,\varphi(2x) + \varphi(2x-1) + \tfrac{1}{2}\,\varphi(2x-2), \tag{9.141}$$

which is (9.138) with $c_0 = \frac{1}{2}$, $c_1 = 1$, $c_2 = \frac{1}{2}$. Again, the reader should be able to check this identity by hand.

The dilation equation (9.138) is a kind of *functional equation*, and, as such, is not so easy to solve. Indeed, the mathematics of functional equations remains much less well developed than that of differential equations or integral equations. Even to prove that (nonzero) solutions exist is a nontrivial analytical problem. Since we already know two explicit examples, let us defer the discussion of solution techniques until we understand how the dilation equation can be used to construct a wavelet basis.

Given a solution to the dilation equation, we define the *mother wavelet* to be

$$\begin{aligned} w(x) &= \sum_{k=0}^{p} (-1)^k c_{p-k}\,\varphi(2x-k) \\ &= c_p\,\varphi(2x) - c_{p-1}\,\varphi(2x-1) + c_{p-2}\,\varphi(2x-2) + \;\cdots\; \pm c_0\,\varphi(2x-p). \end{aligned} \tag{9.142}$$

This formula directly generalizes the Haar wavelet relation (9.127), in light of its dilation equation (9.139). The daughter wavelets are then all found, as in the Haar basis, by iteratively compressing and translating the mother wavelet:

$$w_{j,k}(x) = w(2^j\,x - k). \tag{9.143}$$

In the general framework, we do not necessarily restrict our attention to the interval $[0,1]$, and so j and k can, in principle, be arbitrary integers.

Let us investigate what sort of conditions should be imposed on the dilation coefficients $c_0, \ldots, c_p$ in order that we obtain a viable wavelet basis by this construction. First, localization of the wavelets requires that the scaling function have bounded support, and so $\varphi(x) \equiv 0$ when x lies outside some bounded interval $[a,b]$. Integrating both sides of (9.138) produces

$$\int_a^b \varphi(x)\,dx = \int_{-\infty}^{\infty} \varphi(x)\,dx = \sum_{k=0}^{p} c_k \int_{-\infty}^{\infty} \varphi(2x-k)\,dx. \tag{9.144}$$

Performing the change of variables $y = 2x - k$, with $dx = \frac{1}{2}\,dy$, we obtain

$$\int_{-\infty}^{\infty} \varphi(2x-k)\,dx = \frac{1}{2}\int_{-\infty}^{\infty} \varphi(y)\,dy = \frac{1}{2}\int_a^b \varphi(x)\,dx, \tag{9.145}$$

where we revert to x as our (dummy) integration variable. We substitute this result back into (9.144). Assuming that $\displaystyle\int_a^b \varphi(x)\,dx \neq 0$, we discover that the dilation coefficients must satisfy

$$c_0 + \;\cdots\; + c_p = 2. \tag{9.146}$$

The second condition we require is orthogonality of the wavelets. For simplicity, we only consider the standard L^2 inner product[†]

$$\langle\, f\,,g\,\rangle = \int_{-\infty}^{\infty} f(x)\,g(x)\,dx.$$

It turns out that the orthogonality of the complete wavelet system is guaranteed once we know that the scaling function $\varphi(x)$ is orthogonal to all its integer translates:

$$\langle\, \varphi(x)\,,\varphi(x-m)\,\rangle = \int_{-\infty}^{\infty} \varphi(x)\,\varphi(x-m)\,dx = 0 \qquad \text{for all} \qquad m \neq 0. \tag{9.147}$$

We first note the formula

$$\begin{aligned}\langle\, \varphi(2x-k)\,,\varphi(2x-l)\,\rangle &= \int_{-\infty}^{\infty} \varphi(2x-k)\,\varphi(2x-l)\,dx \\ &= \frac{1}{2}\int_{-\infty}^{\infty} \varphi(x)\,\varphi(x+k-l)\,dx = \frac{1}{2}\,\langle\, \varphi(x)\,,\varphi(x+k-l)\,\rangle\end{aligned} \tag{9.148}$$

follows from the same change of variables $y = 2x-k$ used in (9.145). Therefore, since φ satisfies the dilation equation (9.138), we have

$$\langle\, \varphi(x)\,,\varphi(x-m)\,\rangle = \left\langle\, \sum_{j=0}^{p} c_j\,\varphi(2x-j)\,,\, \sum_{k=0}^{p} c_k\,\varphi(2x-2m-k) \right\rangle \tag{9.149}$$

$$= \sum_{j,k=0}^{p} c_j\,c_k\,\langle\, \varphi(2x-j)\,,\varphi(2x-2m-k)\,\rangle = \frac{1}{2}\sum_{j,k=0}^{p} c_j\,c_k\,\langle\, \varphi(x)\,,\varphi(x+j-2m-k)\,\rangle.$$

If we require orthogonality (9.147) of all the integer translates of φ, then the left-hand side of this identity will be 0 unless $m=0$, while only the summands with $j = 2m+k$ will be nonzero on the right. Therefore, orthogonality requires that

$$\sum_{0\,\leq\, k\,\leq\, p-2m} c_{2m+k}\,c_k = \begin{cases} 2, & m=0, \\ 0, & m\neq 0. \end{cases} \tag{9.150}$$

The algebraic equations (9.146, 150) for the dilation coefficients are the key requirements for the construction of an orthogonal wavelet basis.

For example, if we have just two nonzero coefficients c_0, c_1, then (9.146, 150) reduce to

$$c_0 + c_1 = 2, \qquad\qquad c_0^2 + c_1^2 = 2,$$

and so $c_0 = c_1 = 1$ is the only solution, resulting in the Haar dilation equation (9.139). If we have three coefficients c_0, c_1, c_2, then (9.146), (9.150) require

$$c_0 + c_1 + c_2 = 2, \qquad\qquad c_0^2 + c_1^2 + c_2^2 = 2, \qquad\qquad c_0\,c_2 = 0.$$

[†] In all instances, the functions have bounded support, and so the inner product integral can be reduced to an integral over a finite interval where both f and g are nonzero.

Thus either $c_2 = 0$, $c_0 = c_1 = 1$, and we are back to the Haar case, or $c_0 = 0$, $c_1 = c_2 = 1$, and the resulting dilation equation is a simple reformulation of the Haar case. In particular, the hat function (9.140) does *not* give rise to orthogonal wavelets.

The remarkable fact, discovered by Daubechies, is that there *is* a nontrivial solution for four (and, indeed, any even number) of nonzero coefficients c_0, c_1, c_2, c_3. The basic equations (9.146), (9.150) require

$$c_0 + c_1 + c_2 + c_3 = 2, \qquad c_0^2 + c_1^2 + c_2^2 + c_3^2 = 2, \qquad c_0\, c_2 + c_1\, c_3 = 0. \tag{9.151}$$

The particular values

$$c_0 = \frac{1+\sqrt{3}}{4}, \qquad c_1 = \frac{3+\sqrt{3}}{4}, \qquad c_2 = \frac{3-\sqrt{3}}{4}, \qquad c_3 = \frac{1-\sqrt{3}}{4}, \tag{9.152}$$

solve (9.151). These coefficients correspond to the *Daubechies dilation equation*

$$\varphi(x) = \frac{1+\sqrt{3}}{4}\,\varphi(2x) + \frac{3+\sqrt{3}}{4}\,\varphi(2x-1) + \frac{3-\sqrt{3}}{4}\,\varphi(2x-2) + \frac{1-\sqrt{3}}{4}\,\varphi(2x-3). \tag{9.153}$$

A nonzero solution of bounded support to this remarkable functional equation will give rise to a scaling function $\varphi(x)$, a mother wavelet

$$w(x) = \frac{1-\sqrt{3}}{4}\,\varphi(2x) - \frac{3-\sqrt{3}}{4}\,\varphi(2x-1) + \frac{3+\sqrt{3}}{4}\,\varphi(2x-2) - \frac{1+\sqrt{3}}{4}\,\varphi(2x-3), \tag{9.154}$$

and then, by compression and translation (9.143), the complete system of orthogonal wavelets $w_{j,k}(x)$.

Before explaining how to solve the Daubechies dilation equation, let us complete the proof of orthogonality. It is easy to see that, by translation invariance, since $\varphi(x)$ and $\varphi(x-m)$ are orthogonal whenever $m \neq 0$, so are $\varphi(x-k)$ and $\varphi(x-l)$ for all $k \neq l$. Next we prove orthogonality of $\varphi(x-m)$ and $w(x)$:

$$\begin{aligned}\langle\, w(x)\,,\varphi(x-m)\,\rangle &= \left\langle\ \sum_{j=0}^{p} (-1)^{j+1}\, c_j\, \varphi(2x-1+j)\,,\ \sum_{k=0}^{p} c_k\, \varphi(2x-2m-k)\ \right\rangle \\ &= \sum_{j,k=0}^{p} (-1)^{j+1}\, c_j\, c_k\, \langle\, \varphi(2x-1+j)\,,\varphi(2x-2m-k)\,\rangle \\ &= \frac{1}{2}\sum_{j,k=0}^{p} (-1)^{j+1}\, c_j\, c_k\, \langle\, \varphi(x)\,,\varphi(x-1+j-2m-k)\,\rangle,\end{aligned}$$

using (9.148). By orthogonality (9.147) of the translates of φ, the only summands that are nonzero are those for which $j = 2m+k+1$; the resulting coefficient of $\|\,\varphi(x)\,\|^2$ is

$$\sum_{k} (-1)^k\, c_{1-2m-k}\, c_k = 0,$$

where the sum is over all $0 \leq k \leq p$ such that $0 \leq 1-2m-k \leq p$. Each term in the sum appears twice, with opposite signs, and hence the result is always zero — no matter what the coefficients $c_0, \ldots, c_p$ are! The proof of orthogonality of the translates $w(x-m)$ of the mother wavelet, along with all her wavelet descendants $w(2^j x - k)$, relies on a similar argument, and the details are left as an exercise for the reader.

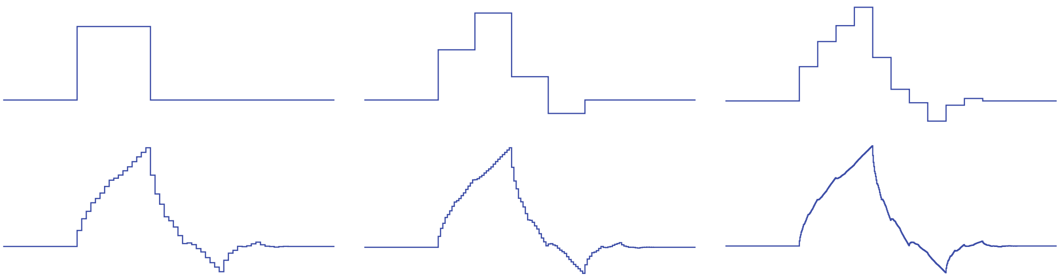


Figure 9.9. Approximating the Daubechies Wavelet.

Solving the Dilation Equation

Let us next discuss how to solve the dilation equation (9.138). The solution we are after does not have an elementary formula, and we require a slightly sophisticated approach to recover it. The key observation is that (9.138) has the form of a fixed point equation

$$\varphi = F[\varphi],$$

not in ordinary Euclidean space, but in an infinite-dimensional function space. With luck, the fixed point (or, more correctly, fixed function) will be stable, and so starting with a suitable initial guess $\varphi_0(x)$, the successive iterates

$$\varphi_{n+1} = F[\varphi_n]$$

will converge to the desired solution: $\varphi_n(x) \longrightarrow \varphi(x)$. In detail, the iterative version of the dilation equation (9.138) reads

$$\varphi_{n+1}(x) = \sum_{k=0}^{p} c_k\, \varphi_n(2x-k), \qquad n = 0, 1, 2, \ldots\,. \tag{9.155}$$

Before attempting to prove convergence of this iterative procedure to the Daubechies scaling function, let us experimentally investigate what happens.

A reasonable choice for the initial guess might be the Haar scaling or box function

$$\varphi_0(x) = \begin{cases} 1, & 0 < t \le 1. \\ 0, & \text{otherwise.} \end{cases}$$

In Figure 9.9 we graph the subsequent iterates $\varphi_1(x), \varphi_2(x), \varphi_4(x), \varphi_5(x), \varphi_7(x)$. There clearly appears to be convergence to some function $\varphi(x)$, although the final result looks a little bizarre. Bolstered by this preliminary experimental evidence, we can now try to prove convergence of the iterative scheme. This turns out to be true; a fully rigorous proof relies on the Fourier transform, and can be found in [**18**].

Theorem 9.59. The functions $\varphi_n(x)$ defined by the iterative functional equation (9.155) converge uniformly to a continuous function $\varphi(x)$, called the *Daubechies scaling function.*

Once we have established convergence, we are now able to verify that the scaling function and consequential system of wavelets form an orthogonal system of functions.

Proposition 9.60. All integer translates $\varphi(x-k)$, for $k \in \mathbb{Z}$ of the Daubechies scaling function, and all wavelets $w_{j,k}(x) = w(2^j x - k)$, $j \geq 0$, are mutually orthogonal functions with respect to the L^2 inner product. Moreover, $\| \varphi \|^2 = 1$, while $\| w_{j,k} \|^2 = 2^{-j}$.

Proof: As noted earlier, the orthogonality of the entire wavelet system will follow once we know the orthogonality (9.147) of the scaling function and its integer translates. We use induction to prove that this holds for all the iterates $\varphi_n(x)$, and so, in view of uniform convergence, the limiting scaling function also satisfies this property. We already know that the orthogonality property holds for the Haar scaling function $\varphi_0(x)$. To demonstrate the induction step, we repeat the computation in (9.149), but now the left-hand side is $\langle \varphi_{n+1}(x), \varphi_{n+1}(x-m) \rangle$, while all other terms involve the previous iterate φ_n. In view of the the algebraic constraints (9.150) on the wavelet coefficients and the induction hypothesis, we deduce that $\langle \varphi_{n+1}(x), \varphi_{n+1}(x-m) \rangle = 0$ whenever $m \neq 0$, while when $m = 0$, $\| \varphi_{n+1} \|^2 = \| \varphi_n \|^2$. Since $\| \varphi_0 \| = 1$, we further conclude that all the iterates, and hence the limiting scaling function, all have unit L^2 norm. The proof of the formula for the norms of the mother and daughter wavelets is left for Exercise 9.7.19. *Q.E.D.*

In practical computations, the limiting procedure for constructing the scaling function is not so convenient, and an alternative means of computing its values is employed. The starting point is to determine its values at integer points. First, the initial box function has values $\varphi_0(m) = 0$ for all integers $m \in \mathbb{Z}$ except $\varphi_0(1) = 1$. The iterative functional equation (9.155) will then produce the values of the iterates $\varphi_n(m)$ at integer points $m \in \mathbb{Z}$. A simple induction will convince you that $\varphi_n(m) = 0$ except for $m = 1$ and $m = 2$, and, therefore, by (9.155),

$$\varphi_{n+1}(1) = \frac{3+\sqrt{3}}{4}\,\varphi_n(1) + \frac{1+\sqrt{3}}{4}\,\varphi_n(2), \qquad \varphi_{n+1}(2) = \frac{1-\sqrt{3}}{4}\,\varphi_n(1) + \frac{3-\sqrt{3}}{4}\,\varphi_n(2),$$

since all other terms are 0. This has the form of a linear iterative system

$$\mathbf{v}^{(n+1)} = A\,\mathbf{v}^{(n)} \tag{9.156}$$

with coefficient matrix

$$A = \begin{pmatrix} \dfrac{3+\sqrt{3}}{4} & \dfrac{1+\sqrt{3}}{4} \\ \dfrac{1-\sqrt{3}}{4} & \dfrac{3-\sqrt{3}}{4} \end{pmatrix} \qquad \text{and where} \qquad \mathbf{v}^{(n)} = \begin{pmatrix} \varphi_n(1) \\ \varphi_n(2) \end{pmatrix}.$$

As we know, the solution to such an iterative system is specified by the eigenvalues and eigenvectors of the coefficient matrix, which are

$$\lambda_1 = 1, \qquad \mathbf{v}_1 = \begin{pmatrix} \frac{1+\sqrt{3}}{4} \\ \frac{1-\sqrt{3}}{4} \end{pmatrix}, \qquad \lambda_2 = \tfrac{1}{2}, \qquad \mathbf{v}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}.$$

We write the initial condition as a linear combination of the eigenvectors

$$\mathbf{v}^{(0)} = \begin{pmatrix} \varphi_0(1) \\ \varphi_0(2) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 2\,\mathbf{v}_1 - \frac{1-\sqrt{3}}{2}\,\mathbf{v}_2.$$

The solution is

$$\mathbf{v}^{(n)} = A^n\,\mathbf{v}^{(0)} = 2\,A^n\mathbf{v}_1 - \frac{1-\sqrt{3}}{2}\,A^n\mathbf{v}_2 = 2\,\mathbf{v}_1 - \frac{1}{2^n}\,\frac{1-\sqrt{3}}{2}\,\mathbf{v}_2.$$

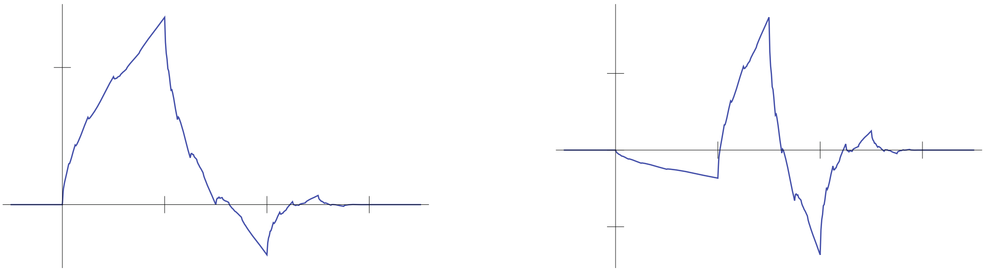


Figure 9.10. The Daubechies Scaling Function and Mother Wavelet.

The limiting vector

$$\begin{pmatrix} \varphi(1) \\ \varphi(2) \end{pmatrix} = \lim_{n \to \infty} \mathbf{v}^{(n)} = 2\,\mathbf{v}_1 = \begin{pmatrix} \dfrac{1+\sqrt{3}}{2} \\ \dfrac{1-\sqrt{3}}{2} \end{pmatrix}$$

gives the desired values of the scaling function:

$$\begin{gathered} \varphi(1) = \frac{1+\sqrt{3}}{2} = 1.366025\ldots\,, \qquad \varphi(2) = \frac{1-\sqrt{3}}{2} = -.366025\ldots\,, \\ \varphi(m) = 0, \qquad \text{for all} \qquad m \neq 1, 2. \end{gathered} \tag{9.157}$$

With this in hand, the Daubechies dilation equation (9.153) then prescribes the function values $\varphi\left(\frac{1}{2}m\right)$ at all half-integers, because if $x = \frac{1}{2}m$, then $2x - k = m - k$ is an integer. Once we know its values at the half-integers, we can reuse equation (9.153) to give its values at quarter-integers $\frac{1}{4}m$. Continuing onward, we determine the values of $\varphi(x)$ at all *dyadic points*, meaning rational numbers of the form $x = m/2^j$ for $m, j \in \mathbb{Z}$. Continuity will then prescribe its value at all other $x \in \mathbb{R}$ since x can be written as the limit of dyadic numbers x_n — namely those obtained by truncating its binary (base 2) expansion at the n^{th} digit beyond the decimal (or, rather "binary") point. But, in practice, this latter step is unnecessary, since all computers are ultimately based on the binary number system, and so only dyadic numbers actually reside in a computer's memory. Thus, there is no real need to determine the value of φ at non-dyadic points.

The preceding scheme was used to produce the graphs of the Daubechies scaling function in Figure 9.10. It is a continuous, but non-differentiable, function — and its graph has a very jagged, fractal-like appearance when viewed at close range. The Daubechies scaling function is, in fact, a close relative of the famous example of a continuous, nowhere differentiable function originally due to Weierstrass, [**42**, **53**], whose construction also relies on a similar scaling argument.

Given the values of the Daubechies scaling function on a sufficiently dense set of dyadic points, the consequential values of the mother wavelet are given by formula (9.154). Note that $\operatorname{supp}\varphi = \operatorname{supp} w = [0,3]$. The daughter wavelets are then found by the usual compression and translation procedure (9.143).

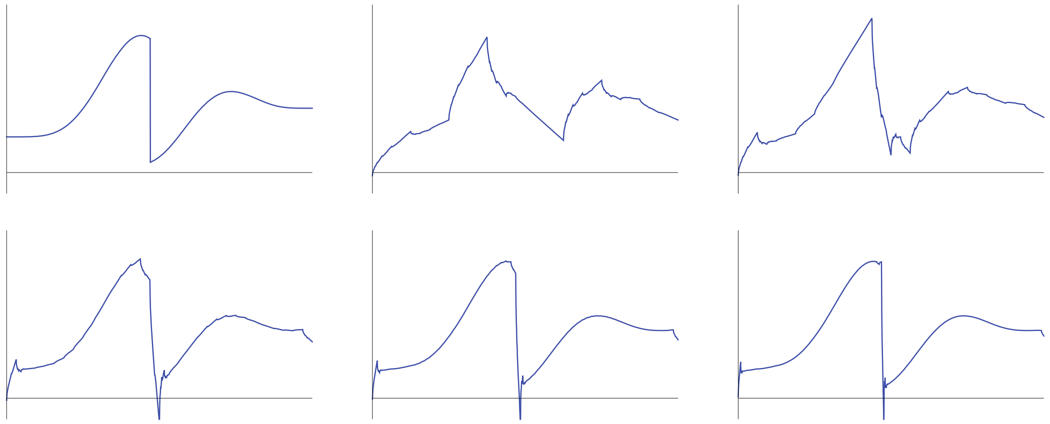


Figure 9.11. Daubechies Wavelet Expansion.

The Daubechies wavelet expansion of a function whose support is contained in[†] $[0,1]$ is then given by

$$f(x) \sim c_0\,\varphi(x) + \sum_{j=0}^{\infty} \sum_{k=-2}^{2^j-1} c_{j,k}\, w_{j,k}(x). \tag{9.158}$$

The inner summation begins at $k=-2$ so as to include *all* the wavelet offspring $w_{j,k}$ whose supports have a nontrivial intersection with the interval $[0,1]$. The wavelet coefficients $c_0, c_{j,k}$ are computed by the usual orthogonality formula

$$\begin{aligned} c_0 &= \langle f, \varphi \rangle = \int_0^3 f(x)\,\varphi(x)\,dx, \\ c_{j,k} &= \langle f, w_{j,k} \rangle = 2^j \int_{2^{-j}k}^{2^{-j}(k+3)} f(x)\,w_{j,k}(x)\,dx = \int_0^3 f\big(2^{-j}(x+k)\big)\,w(x)\,dx, \end{aligned} \tag{9.159}$$

where we agree that $f(x)=0$ whenever $x<0$ or $x>1$. In practice, one employs a numerical integration procedure, e.g., the trapezoid rule, based on dyadic points to speedily evaluate the integrals (9.159). A proof of completeness of the resulting wavelet basis functions can be found in [**18**]. Compression and denoising algorithms based on retaining only low-frequency modes proceed as in Section 5.6, and are left as projects for the motivated reader to implement.

Example 9.61. In Figure 9.11, we plot the Daubechies wavelet expansions of the same signal for Example 9.56. The first plot is the original signal, and the following show the partial sums of (9.158) over $j=0,\ldots,r$ with $r=2,3,4,5,6$. Unlike the Haar expansion, the Daubechies wavelets do exhibit a nonuniform Gibbs phenomenon, where the expansion noticeably overshoots near the discontinuity, [**61**], which can be observed at the interior discontinuity as well as the endpoints, since the function is set to 0 outside the interval

[†] For functions with larger support, one should include additional terms in the expansion corresponding to further translates of the wavelets so as to cover the entire support of the function. Alternatively, one can translate and rescale x to fit the function's support inside $[0,1]$.

$[0,1]$. Indeed, the Daubechies wavelets are continuous, and so cannot converge uniformly to a discontinuous function.

Exercises

♠ 9.7.8. Answer Exercises 9.7.1 and 9.7.2 using the Daubechies wavelets instead of the Haar wavelets. Do you see any improvement in your approximations? Discuss the advantages and disadvantages of both in light of these examples.

♠ 9.7.9. Answer Exercise 9.7.3 using the Daubechies wavelets to compress the data. Compare your results.

♢ 9.7.10. Verify formulas (9.139) and (9.141).

9.7.11. Prove that the most general solution to the functional equation $\varphi(x) = 2\,\varphi(2x)$ is $\varphi(x) = f(\log_2 x)/x$ where $f(z+1) = f(z)$ is any 1 periodic function.

♢ 9.7.12. Consider the dilation equation (9.138) with $c_0 = 0$, $c_1 = c_2 = 1$, so $\varphi(x) = \varphi(2x-1) + \varphi(2x-2)$. Prove that $\psi(x) = \varphi(x+1)$ satisfies the Haar dilation equation (9.139). Generalize this result to prove that we can always, without loss of generality, assume that $c_0 \neq 0$ in the general dilation equation (9.138).

9.7.13. Prove that a cubic B spline, as defined in Exercise 5.5.76, solves the dilation equation (9.138) for $c_0 = c_4 = \frac{1}{8}$, $c_1 = c_3 = \frac{1}{2}$, $c_2 = \frac{3}{4}$.

9.7.14. Explain why the scaling function $\varphi(x)$ and the mother wavelet $w(x)$ have the same support: $\operatorname{supp}\varphi = \operatorname{supp} w$.

9.7.15. Prove that (9.147) implies $\langle\, \varphi(x-l)\,, \varphi(x-m)\,\rangle = 0$ for all $l \neq m$.

♢ 9.7.16. Let $\varphi(x)$ be any scaling function, $w(x)$ the corresponding mother wavelet and $w_{j,k}(x)$ the wavelet descendants. Prove that (a) $\|\,\varphi\,\| = \|\,w\,\|$. (b) $\|\,w_{j,k}\,\| = 2^{-j}\,\|\,\varphi\,\|$.

♢ 9.7.17. (a) Prove that the scaling function $\varphi(x)$ and the mother wavelet $w(x)$ are orthogonal. (b) Prove that the integer translates $w(x-m)$ of the mother wavelet are mutually orthogonal. (c) Prove orthogonality of all the wavelet offspring $w_{j,k}(x)$.

9.7.18. Find the values of the Daubechies scaling function $\varphi(x)$ and mother wavelet $w(x)$ at $x =$ (a) $\frac{1}{2}$, (b) $\frac{1}{4}$, (c) $\frac{5}{16}$.

♢ 9.7.19. Prove the formulas in Proposition 9.60 for the norms of the mother and daughter wavelets.

♠ 9.7.20. Write a computer program to zoom in on the Daubechies scaling function and discuss what you see.

9.7.21. *True or false*: The iterative system (9.156) is a Markov process.

♢ 9.7.22. Let $\varphi(x)$ satisfy the Daubechies scaling equation (9.153). Prove that if $\varphi(i) \neq 0$ for any $i \leq 0$ or $i \geq p$, then $\operatorname{supp}\varphi$ is unbounded.

9.7.23. (a) Use (9.142) to construct the "mother wavelet" corresponding to the hat function (9.140). (b) Is the hat function orthogonal to the mother wavelet? (c) Is the hat function orthogonal to its integer translates?

9.7.24. Prove that a real number x is dyadic if and only if its binary (base 2) expansion terminates, i.e., is eventually all zeros.

9.7.25. Find dyadic approximations, with error at most 2^{-8}, to

(a) $\frac{3}{4}$, (b) $\frac{1}{3}$, (c) $\sqrt{2}$, (d) e, (e) π.