Nonlinear Systems

by Peter J. Olver
University of Minnesota

1. Introduction.

Nonlinearity is ubiquitous in physical phenomena. Fluid and plasma mechanics, gas dynamics, elasticity, relativity, chemical reactions, combustion, ecology, biomechanics, and many, many other phenomena are all governed by inherently nonlinear equations. (The one notable exception is quantum mechanics, which is a fundamentally linear theory, although recent attempts at grand unification of all fundamental physical theories, such as string theory and conformal field theory, [**8**], are nonlinear.) For this reason, an ever increasing proportion of modern mathematical research is devoted to the analysis of nonlinear systems and nonlinear phenomena.

Why, then, does one devote so much time studying linear mathematics? The facile answer is that nonlinear systems are vastly more difficult to analyze. In the nonlinear regime, many of the most basic questions remain unanswered: existence and uniqueness of solutions are not guaranteed; explicit formulae are difficult to come by; linear superposition is no longer available; numerical approximations are not always sufficiently accurate; etc., etc. A more intelligent answer is that a thorough understanding of linear phenomena and linear mathematics is an essential prerequisite for progress in the nonlinear arena. Therefore, one must first develop the proper linear foundations in sufficient depth before we can realistically confront the untamed nonlinear wilderness. Moreover, many important physical systems are "weakly nonlinear", in the sense that, while nonlinear effects do play an essential role, the linear terms tend to dominate the physics, and so, to a first approximation, the system is essentially linear. As a result, such nonlinear phenomena are best understood as some form of perturbation of their linear approximations. The truly nonlinear regime is, even today, only sporadically modeled and even less well understood.

The advent of powerful computers has fomented a veritable revolution in our understanding of nonlinear mathematics. Indeed, many of the most important modern analytical techniques drew their inspiration from early computer-aided investigations of nonlinear systems. However, despite dramatic advances in both hardware capabilities and sophisticated mathematical algorithms, many nonlinear systems — for instance, fully general Einsteinian gravitation, or the Navier–Stokes equations of fluid mechanics at high Reynolds numbers — still remain beyond the capabilities of today's computers.

The goal of these lecture notes is to provide a brief overview of some of the most important ideas, mathematical techniques, and new physical phenomena in the nonlinear realm. We start with iteration of nonlinear functions, also known as discrete dynamical systems. Building on our experience with iterative linear systems, as developed in Chapter 10 of [**14**], we will discover that functional iteration, when it converges, provides a powerful mechanism for solving equations and for optimization. On the other hand, even

very simple non-convergent nonlinear iterative systems may admit remarkably complex, chaotic behavior. The third section is devoted to basic solution techniques for nonlinear equations and nonlinear systems, and includes bisection, general iteration, and the very powerful Newton Method. The fourth section is devoted to finite-dimensional optimization principles, i.e., the minimization or maximization of nonlinear functions, and including systems with constraints leading to the method of Lagrange multipliers. Numerical optimization procedures rely on iterative procedures, and we concentrate on those associated with gradient descent.

2. Iteration of Functions.

Iteration, meaning repeated application of a function, can be viewed as a *discrete dynamical system* in which the continuous time variable has been "quantized" to assume integer values. Even iterating a very simple quadratic scalar function can lead to an amazing variety of dynamical phenomena, including multiply-periodic solutions and genuine chaos. Nonlinear iterative systems arise not just in mathematics, but also underlie the growth and decay of biological populations, predator-prey interactions, spread of communicable diseases such as AIDS, and host of other natural phenomena. Moreover, many numerical solution methods — for systems of algebraic equations, ordinary differential equations, partial differential equations, and so on — rely on iteration, and so the theory underlies the analysis of convergence and efficiency of such numerical approximation schemes.

In general, an iterative system has the form

$$\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)}), \tag{2.1}$$

where $\mathbf{g}\colon \mathbb{R}^n \to \mathbb{R}^n$ is a real vector-valued function. (One can similarly treat iteration of complex-valued functions $\mathbf{g}\colon \mathbb{C}^n \to \mathbb{C}^n$, but, for simplicity, we only deal with real systems here.) A solution is a discrete collection of points[†] $\mathbf{u}^{(k)} \in \mathbb{R}^n$, in which the index $k = 0, 1, 2, 3, \dots$ takes on non-negative integer values. Chapter 10 of [**14**] dealt with the case when $\mathbf{g}(\mathbf{u}) = A\,\mathbf{u}$ is a linear function, necessarily given by multiplication by an $n \times n$ matrix A. In this chapter, we enlarge our scope to the nonlinear case.

Once we specify the initial iterate,

$$\mathbf{u}^{(0)} = \mathbf{c}, \tag{2.2}$$

then the resulting solution to the discrete dynamical system (2.1) is easily computed:

$$\mathbf{u}^{(1)} = \mathbf{g}(\mathbf{u}^{(0)}) = \mathbf{g}(\mathbf{c}), \quad \mathbf{u}^{(2)} = \mathbf{g}(\mathbf{u}^{(1)}) = \mathbf{g}(\mathbf{g}(\mathbf{c})), \quad \mathbf{u}^{(3)} = \mathbf{g}(\mathbf{u}^{(2)}) = \mathbf{g}(\mathbf{g}(\mathbf{g}(\mathbf{c}))), \quad \dots$$

and so on. Thus, unlike continuous dynamical systems, the existence and uniqueness of solutions is not an issue. As long as each successive iterate $\mathbf{u}^{(k)}$ lies in the domain of definition of $\mathbf{g}$ one merely repeats the process to produce the solution,

$$\mathbf{u}^{(k)} = \overbrace{\mathbf{g} \circ \cdots \circ \mathbf{g}}^{k \text{ times}}(\mathbf{c}), \qquad k = 0, 1, 2, \dots, \tag{2.3}$$

[†] The superscripts on $\mathbf{u}^{(k)}$ refer to the iteration number, and do *not* denote derivatives.

which is obtained by composing the function $\mathbf{g}$ with itself a total of k times. In other words, the solution to a discrete dynamical system corresponds to repeatedly pushing the $\mathbf{g}$ key on your calculator. For example, entering 0 and then repeatedly hitting the `cos` key corresponds to solving the iterative system

$$u^{(k+1)} = \cos u^{(k)}, \qquad u^{(0)} = 0. \tag{2.4}$$

The first 10 iterates are displayed in the following table:

k	0	1	2	3	4	5	6	7	8	9
$u^{(k)}$	0	1	.540302	.857553	.65429	.79348	.701369	.76396	.722102	.750418

For simplicity, we shall always assume that the vector-valued function $\mathbf{g}\colon \mathbb{R}^n \to \mathbb{R}^n$ is defined on all of $\mathbb{R}^n$; otherwise, we must always be careful that the successive iterates $\mathbf{u}^{(k)}$ never leave its domain of definition, thereby causing the iteration to break down. To avoid technical complications, we will also assume that $\mathbf{g}$ is at least continuous; later results rely on additional smoothness requirements, e.g., continuity of its first and second order partial derivatives.

While the solution to a discrete dynamical system is essentially trivial, understanding its behavior is definitely not. Sometimes the solution converges to a particular value — the key requirement for numerical solution methods. Sometimes it goes off to ∞, or, more precisely, the norms[†] of the iterates are unbounded: $\| \mathbf{u}^{(k)} \| \to \infty$ as $k \to \infty$. Sometimes the solution repeats itself after a while. And sometimes the iterates behave in a seemingly random, chaotic manner — all depending on the function $\mathbf{g}$ and, at times, the initial condition $\mathbf{c}$. Although all of these cases may arise in real-world applications, we shall mostly concentrate upon understanding convergence.

Definition 2.1. A *fixed point* or *equilibrium* of a discrete dynamical system (2.1) is a vector $\mathbf{u}^\star \in \mathbb{R}^n$ such that

$$\mathbf{g}(\mathbf{u}^\star) = \mathbf{u}^\star. \tag{2.5}$$

We easily see that every fixed point provides a constant solution to the discrete dynamical system, namely $\mathbf{u}^{(k)} = \mathbf{u}^\star$ for all k. Moreover, it is not hard to prove that any convergent solution necessarily converges to a fixed point.

Proposition 2.2. *If a solution to a discrete dynamical system converges,*

$$\lim_{k\to\infty} \mathbf{u}^{(k)} = \mathbf{u}^\star,$$

then the limit $\mathbf{u}^\star$ is a fixed point.

Proof: This is a simple consequence of the continuity of $\mathbf{g}$. We have

$$\mathbf{u}^\star = \lim_{k\to\infty} \mathbf{u}^{(k+1)} = \lim_{k\to\infty} \mathbf{g}(\mathbf{u}^{(k)}) = \mathbf{g}\left(\lim_{k\to\infty} \mathbf{u}^{(k)} \right) = \mathbf{g}(\mathbf{u}^\star),$$

the last two equalities following from the continuity of $\mathbf{g}$. *Q.E.D.*

[†] In view of the equivalence of norms on finite-dimensional vector spaces, cf. [**14**], any norm will do here.

For example, continuing the cosine iteration (2.4), we find that the iterates gradually converge to the value $u^\star \approx .739085$, which is the unique solution to the fixed point equation

$$\cos u = u.$$

Later we will see how to rigorously prove this observed behavior.

Of course, not every solution to a discrete dynamical system will necessarily converge, but Proposition 2.2 says that if it does, then it must converge to a fixed point. Thus, a key goal is to understand when a solution converges, and, if so, to which fixed point — if there is more than one. (In the linear case, only the actual convergence is a significant issues since most linear systems admit exactly one fixed point, namely $\mathbf{u}^\star = \mathbf{0}$.)

Fixed points are roughly divided into three classes:

- *asymptotically stable*, with the property that all nearby solutions converge to it,
- *stable*, with the property that all nearby solutions stay nearby, and
- *unstable*, almost all of whose nearby solutions diverge away from the fixed point.

Thus, from a practical standpoint, convergence of the iterates of a discrete dynamical system requires asymptotic stability of the fixed point. Examples will appear in abundance in the following sections.

Scalar Functions

As always, the first step is to thoroughly understand the scalar case, and so we begin with a discrete dynamical system

$$u^{(k+1)} = g(u^{(k)}), \qquad u^{(0)} = c, \tag{2.6}$$

in which $g\colon \mathbb{R} \to \mathbb{R}$ is a continuous, scalar-valued function. As noted above, we will assume, for simplicity, that g is defined everywhere, and so we do not need to worry about whether the iterates $u^{(0)}, u^{(1)}, u^{(2)}, \dots$ are all well-defined.

The elementary linear case $g(u) = a\,u$ is treated in [**14**; Chapter 10]. The simplest "nonlinear" case is that of an affine function

$$g(u) = a\,u + b, \tag{2.7}$$

leading to an *affine discrete dynamical system*

$$u^{(k+1)} = a\,u^{(k)} + b. \tag{2.8}$$

The only fixed point is the solution to

$$u^\star = g(u^\star) = a\,u^\star + b, \qquad \text{namely,} \qquad u^\star = \frac{b}{1-a}\,. \tag{2.9}$$

The formula for $u^\star$ requires that $a \neq 1$, and, indeed, the case $a = 1$ has no fixed point, as the reader can easily confirm.

Since we already know the value of $u^\star$, we can readily analyze the differences

$$e^{(k)} = u^{(k)} - u^\star, \tag{2.10}$$

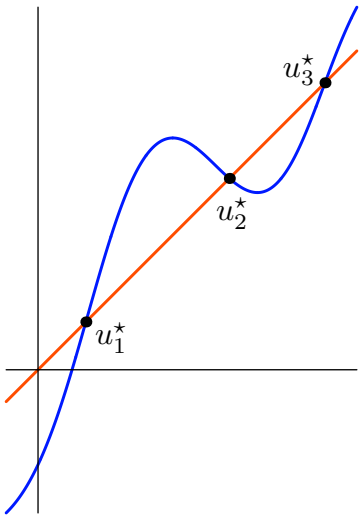


Figure 1. Fixed Points.

between successive iterates and the fixed point. Observe that, the smaller $e^{(k)}$ is, the closer $u^{(k)}$ is to the desired fixed point. In many applications, the iterate $u^{(k)}$ is viewed as an approximation to the fixed point $u^\star$, and so $e^{(k)}$ is interpreted as the *error* in the k^{th} iterate. Subtracting the fixed point equation (2.9) from the iteration equation (2.8), we find

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

Therefore the errors $e^{(k)}$ are related by a *linear iteration*

$$e^{(k+1)} = a\,e^{(k)}, \qquad \text{and hence} \qquad e^{(k)} = a^k e^{(0)}. \tag{2.11}$$

Therefore, the solutions to this scalar linear iteration converge:

$$e^{(k)} \longrightarrow 0 \qquad \text{and hence} \qquad u^{(k)} \longrightarrow u^\star, \qquad \text{if and only if} \qquad |\,a\,| < 1.$$

This is the criterion for *asymptotic stability* of the fixed point, or, equivalently, convergence of the affine iterative system (2.8). The magnitude of a determines the rate of convergence, and the closer it is to 0, the faster the iterates approach the fixed point.

Example 2.3. The affine function

$$g(u) = \tfrac{1}{4}u + 2$$

leads to the iterative scheme

$$u^{(k+1)} = \tfrac{1}{4}u^{(k)} + 2.$$

Starting with the initial condition $u^{(0)} = 0$, the ensuing values are

k	1	2	3	4	5	6	7	8
$u^{(k)}$	2.0	2.5	2.625	2.6562	2.6641	2.6660	2.6665	2.6666

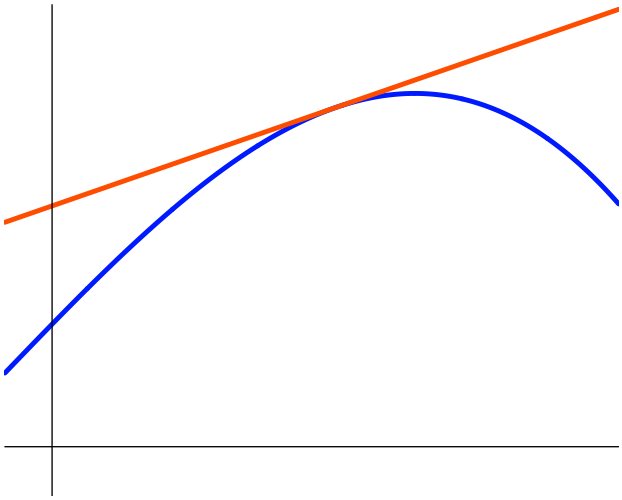


Figure 2. Tangent Line Approximation.

Thus, after 8 iterations, the iterates have produced the fixed point $u^\star = \frac{8}{3}$ to 4 decimal places. The rate of convergence is $\frac{1}{4}$, and indeed

$$| e^{(k)} | = | u^{(k)} - u^\star | = \left(\tfrac{1}{4}\right)^k | u^{(0)} - u^\star | = \tfrac{8}{3} \left(\tfrac{1}{4}\right)^k \longrightarrow 0 \qquad \text{as} \qquad k \longrightarrow \infty.$$

Let us now turn to the fully nonlinear case. First note that the fixed points of $g(u)$ correspond to the intersections of its graph with the graph of the function $i(u) = u$. For instance Figure 1 shows the graph of a function that has 3 fixed points, labeled $u_1^\star, u_2^\star, u_3^\star$.

In general, near any point in its domain, a (smooth) nonlinear function can be well approximated by its tangent line, which represents the graph of an affine function; see Figure 2. Therefore, if we are close to a fixed point $u^\star$, then we might expect the iterative system based on the nonlinear function $g(u)$ to behave very much like that of its affine tangent line approximation. And, indeed, this intuition turns out to be essentially correct. This result forms our first concrete example of *linearization*, in which the analysis of a nonlinear system is based on its linear (or, more precisely, affine) approximation.

The explicit formula for the tangent line to $g(u)$ near the fixed point $u = u^\star = g(u^\star)$ is

$$g(u) \approx g(u^\star) + g'(u^\star)(u - u^\star) \equiv a\, u + b, \tag{2.12}$$

where

$$a = g'(u^\star), \qquad\qquad b = g(u^\star) - g'(u^\star)\, u^\star = \big(\, 1 - g'(u^\star) \,\big)\, u^\star.$$

Note that $u^\star = b\,/(1 - a)$ remains a fixed point for the affine approximation: $a\, u^\star + b = u^\star$. According to the preceding discussion, the convergence of the iterates for the affine approximation is governed by the size of the coefficient $a = g'(u^\star)$. This observation inspires the basic stability criterion for fixed points of scalar iterative systems.

Theorem 2.4. *Let $g(u)$ be a continuously differentiable scalar function. Suppose $u^\star = g(u^\star)$ is a fixed point. If $| g'(u^\star) | < 1$, then $u^\star$ is an asymptotically stable fixed point, and hence any sequence of iterates $u^{(k)}$ which starts out sufficiently close to $u^\star$ will converge to $u^\star$. On the other hand, if $| g'(u^\star) | > 1$, then $u^\star$ is an unstable fixed point, and*

the only iterates which converge to it are those that land exactly on it, i.e., $u^{(k)} = u^\star$ for some $k \geq 0$.

Proof: The goal is to prove that the errors $e^{(k)} = u^{(k)} - u^\star$ between the iterates and the fixed point tend to 0 as $k \to \infty$. To this end, we try to estimate $e^{(k+1)}$ in terms of $e^{(k)}$. According to (2.6) and the Mean Value Theorem from calculus,

$$e^{(k+1)} = u^{(k+1)} - u^\star = g(u^{(k)}) - g(u^\star) = g'(v)\,(u^{(k)} - u^\star) = g'(v)\,e^{(k)}, \tag{2.13}$$

for some v lying between $u^{(k)}$ and $u^\star$. By continuity, if $|\,g'(u^\star)\,| < 1$ at the fixed point, then we can choose $\delta > 0$ and $|\,g'(u^\star)\,| < \sigma < 1$ such that the estimate

$$|\,g'(v)\,| \leq \sigma < 1 \qquad \text{whenever} \qquad |\,v - u^\star\,| < \delta \tag{2.14}$$

holds in a (perhaps small) interval surrounding the fixed point. Suppose

$$|\,e^{(k)}\,| = |\,u^{(k)} - u^\star\,| < \delta.$$

Then the point v in (2.13), which is closer to $u^\star$ than $u^{(k)}$, satisfies (2.14). Therefore,

$$|\,u^{(k+1)} - u^\star\,| \;\leq\; \sigma\,|\,u^{(k)} - u^\star\,|, \qquad \text{and hence} \qquad |\,e^{(k+1)}\,| \;\leq\; \sigma\,|\,e^{(k)}\,|. \tag{2.15}$$

In particular, since $\sigma < 1$, we have $|\,u^{(k+1)} - u^\star\,| < \delta$, and hence the subsequent iterate $u^{(k+1)}$ also lies in the interval where (2.14) holds. Repeating the argument, we conclude that, provided the initial iterate satisfies

$$|\,e^{(0)}\,| = |\,u^{(0)} - u^\star\,| < \delta,$$

the subsequent errors are bounded by

$$e^{(k)} \;\leq\; \sigma^k\,e^{(0)}, \qquad \text{and hence} \qquad e^{(k)} = |\,u^{(k)} - u^\star\,| \;\longrightarrow\; 0 \quad \text{as} \quad k \to \infty,$$

which completes the proof of the theorem in the stable case.

The proof in unstable case is left as an exercise for the reader. *Q.E.D.*

Remark: The constant σ governs the rate of convergence of the iterates to the fixed point. The closer the iterates are to the fixed point, the smaller we can choose δ in (2.14), and hence the closer we can choose σ to $|\,g'(u^\star)\,|$. Thus, roughly speaking, $|\,g'(u^\star)\,|$ governs the speed of convergence, once the iterates get close to the fixed point. This observation will be developed more fully in the following subsection.

Remark: The cases when $g'(u^\star) = \pm 1$ are *not* covered by the theorem. For a linear system, such fixed points are stable, but not asymptotically stable. For nonlinear systems, more detailed knowledge of the nonlinear terms is required in order to resolve the status — stable or unstable — of the fixed point. Despite their importance in certain applications, we will not try to analyze such borderline cases any further here.

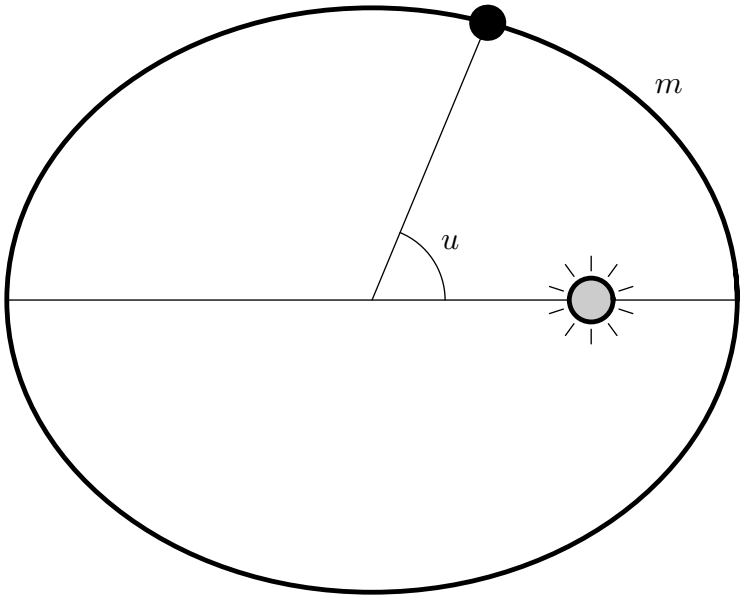


Figure 3. Planetary Orbit.

Example 2.5. Given constants ϵ, m, the trigonometric equation

$$u = m + \epsilon \sin u \tag{2.16}$$

is known as *Kepler's equation*. It arises in the study of planetary motion, in which $0 < \epsilon < 1$ represents the *eccentricity* of an elliptical planetary orbit, u is the *eccentric anomaly*, defined as the angle formed at the center of the ellipse by the planet and the major axis, and $m = 2\pi t\,/T$ is its *mean anomaly*, which is the time, measured in units of $T/(2\pi)$ where T is the period of the orbit, i.e., the length of the planet's year, since perihelion or point of closest approach to the sun; see Figure 3.

The solutions to Kepler's equation are the fixed points of the discrete dynamical system based on the function

$$g(u) = m + \epsilon \sin u.$$

Note that

$$|\, g'(u)\,| = |\, \epsilon \cos u\,| \le |\, \epsilon\,| < 1, \tag{2.17}$$

which automatically implies that the as yet unknown fixed point is stable. Indeed, condition (2.17) is enough to prove the existence of a unique stable fixed point; see Theorem 2.18 below. In the particular case $m = \epsilon = \frac{1}{2}$, the result of iterating $u^{(k+1)} = \frac{1}{2} + \frac{1}{2}\sin u^{(k)}$ starting with $u^{(0)} = \frac{1}{2}$ is

k	0	1	2	3	4	5	6	7	8
$u^{(k)}$	.5	.7397	.8370	.8713	.8826	.8862	.8873	.8877	.8878

After 12 iterations, we have converged sufficiently close to the solution (fixed point) $u^\star = .887862$ to have computed its value to 6 decimal places.

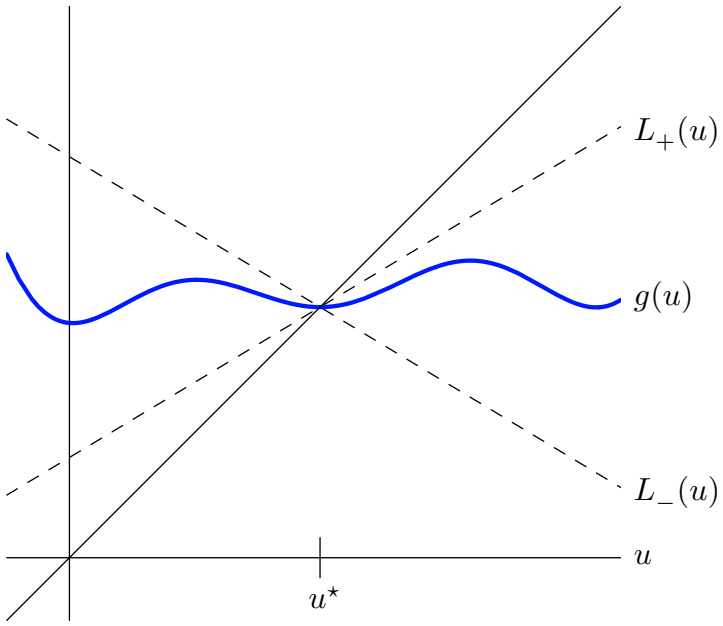


Figure 4. Graph of a Contraction.

Inspection of the proof of Theorem 2.4 reveals that we never really used the differentiability of g, except to verify the inequality

$$| g(u) - g(u^\star) | \leq \sigma \, | u - u^\star | \qquad \text{for some fixed } \sigma < 1. \tag{2.18}$$

A function that satisfies (2.18) for all nearby u is called a *contraction* at the point $u^\star$. *Any* function $g(u)$ whose graph lies between the two lines

$$L_\pm(u) = g(u^\star) \pm \sigma \, (u - u^\star) \qquad \text{for some} \quad \sigma < 1,$$

for all u sufficiently close to $u^\star$, i.e., such that $| u - u^\star | < \delta$ for some $\delta > 0$, defines a contraction, and hence fixed point iteration starting with $| u^{(0)} - u^\star | < \delta$ will converge to $u^\star$; see Figure 4. In particular, any function that is differentiable at $u^\star$ with $| g'(u^\star) | < 1$ defines a contraction at $u^\star$.

Example 2.6. The simplest truly nonlinear example is a quadratic polynomial. The most important case is the so-called *logistic map*

$$g(u) = \lambda \, u(1 - u), \tag{2.19}$$

where $\lambda \neq 0$ is a fixed non-zero parameter. (The case $\lambda = 0$ is completely trivial. Why?) In fact, an elementary change of variables can make any quadratic iterative system into one involving a logistic map.

The fixed points of the logistic map are the solutions to the quadratic equation

$$u = \lambda \, u(1 - u), \qquad \text{or} \qquad \lambda \, u^2 - \lambda \, u + 1 = 0.$$

Using the quadratic formula, we conclude that $g(u)$ has two fixed points:

$$u_1^\star = 0, \qquad u_2^\star = 1 - \frac{1}{\lambda} \, .$$

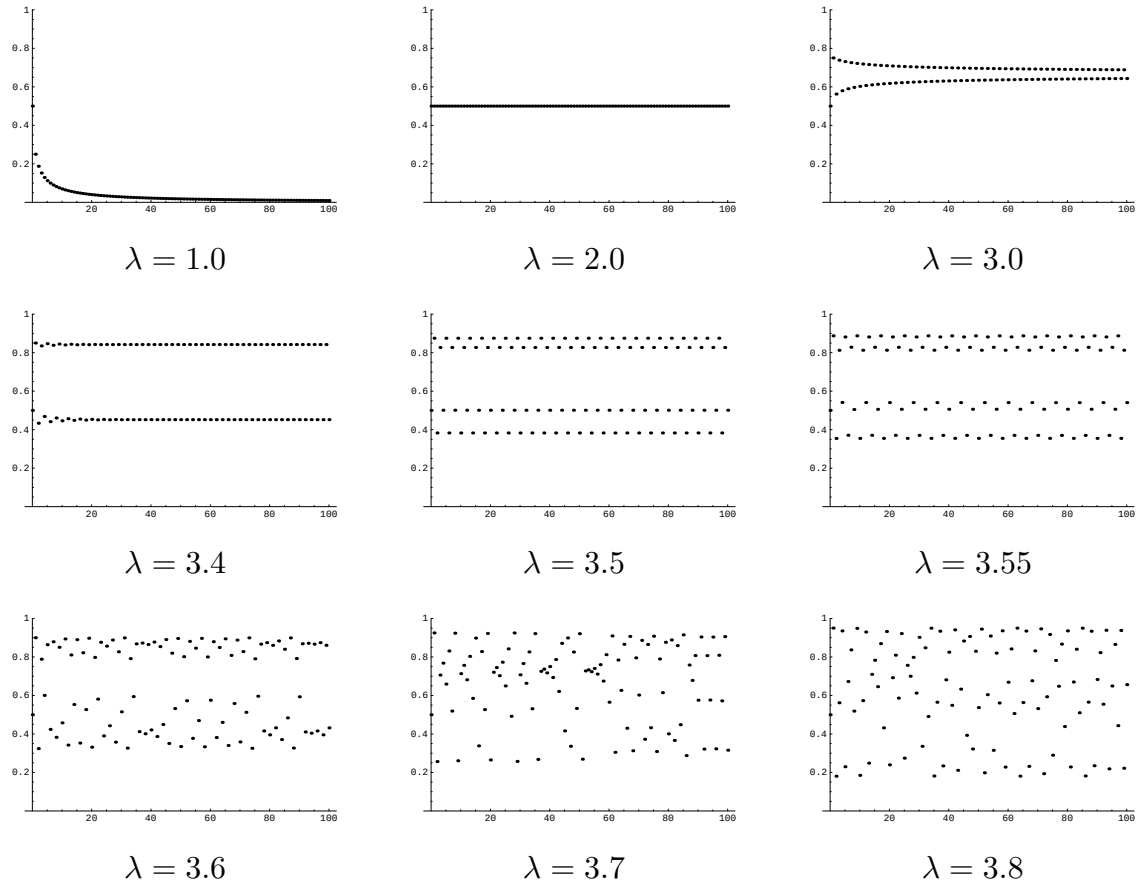


Figure 5. Logistic Iterates.

Let us apply Theorem 2.4 to determine their stability. The derivative is

$$g'(u) = \lambda - 2\,\lambda\,u, \qquad \text{and so} \qquad g'(u_1^\star) = \lambda, \qquad g'(u_2^\star) = 2 - \lambda.$$

Therefore, if $|\,\lambda\,| < 1$, the first fixed point is stable, while if $1 < \lambda < 3$, the second fixed point is stable. For $\lambda < -1$ or $\lambda > 3$ neither fixed point is stable, and we expect the iterates to not converge at all.

Numerical experiments with this example show that it is the source of an amazingly diverse range of behavior, depending upon the value of the parameter λ. In the accompanying Figure 5, we display the results of iteration starting with initial point $u^{(0)} = .5$ for several different values of λ; in each plot, the horizontal axis indicates the iterate number k and the vertical axis the iterate value $u^{(k)}$ for $k = 0, \ldots, 100$. As expected from Theorem 2.4, the iterates converge to one of the fixed points in the range $-1 < \lambda < 3$, except when $\lambda = 1$. For λ a little bit larger than $\lambda_1 = 3$, the iterates do not converge to a fixed point. But it does not take long for them to settle down, switching back and forth between two particular values. This behavior indicates the existence of a (stable) *period* 2 *orbit* for the discrete dynamical system, in accordance with the following definition.

Definition 2.7. A *period k orbit* of a discrete dynamical system is a solution that satisfies $u^{(n+k)} = u^{(n)}$ for all $n = 0, 1, 2, \ldots$. The (*minimal*) *period* is the smallest positive value of k for which this condition holds.

Thus, a fixed point

$$u^{(0)} = u^{(1)} = u^{(2)} = \cdots$$

is a period 1 orbit. A period 2 orbit satisfies

$$u^{(0)} = u^{(2)} = u^{(4)} = \cdots \qquad \text{and} \qquad u^{(1)} = u^{(3)} = u^{(5)} = \cdots,$$

but $u^{(0)} \neq u^{(1)}$, as otherwise the minimal period would be 1. Similarly, a period 3 orbit has

$$u^{(0)} = u^{(3)} = u^{(6)} = \cdots, \qquad u^{(1)} = u^{(4)} = u^{(7)} = \cdots, \qquad u^{(2)} = u^{(5)} = u^{(8)} = \cdots,$$

with $u^{(0)}, u^{(1)}, u^{(2)}$ distinct. Stability of a period k orbit implies that nearby iterates converge to this periodic solution.

For the logistic map, the period 2 orbit persists until $\lambda = \lambda_2 \approx 3.4495$, after which the iterates alternate between four values — a period 4 orbit. This again changes at $\lambda = \lambda_3 \approx 3.5441$, after which the iterates end up alternating between eight values. In fact, there is an increasing sequence of values

$$3 = \lambda_1 < \lambda_2 < \lambda_3 < \lambda_4 < \cdots,$$

where, for any $\lambda_n < \lambda \leq \lambda_{n+1}$, the iterates eventually follow a period 2^n orbit. Thus, as λ passes through each value λ_n the period of the orbit goes from 2^n to $2 \cdot 2^n = 2^{n+1}$, and the discrete dynamical system experiences a *bifurcation*. The bifurcation values λ_n are packed closer and closer together as n increases, piling up on an eventual limiting value

$$\lambda_\star = \lim_{n \to \infty} \lambda_n \approx 3.5699,$$

at which point the orbit's period has, so to speak, become infinitely large. The entire phenomena is known as a *period doubling cascade*.

Interestingly, the ratios of the distances between successive bifurcation points approaches a well-defined limit,

$$\frac{\lambda_{n+2} - \lambda_{n+1}}{\lambda_{n+1} - \lambda_n} \quad \longrightarrow \quad 4.6692\ldots, \tag{2.20}$$

known as *Feigenbaum's constant*. In the 1970's, the American physicist Mitchell Feigenbaum, [**6**], discovered that similar period doubling cascades appear in a broad range of discrete dynamical systems. Even more remarkably, in almost all cases, the corresponding ratios of distances between bifurcation points has the *same* limiting value. Feigenbaum's experimental observations were rigorously proved by Oscar Lanford in 1982, [**10**].

After λ passes the limiting value $\lambda_\star$, all hell breaks loose. The iterates become completely chaotic†, moving at random over the interval $[0, 1]$. But this is not the end of the

† The term "chaotic" does have a precise mathematical definition, [**5**], but the reader can take it more figuratively for the purposes of this elementary exposition.

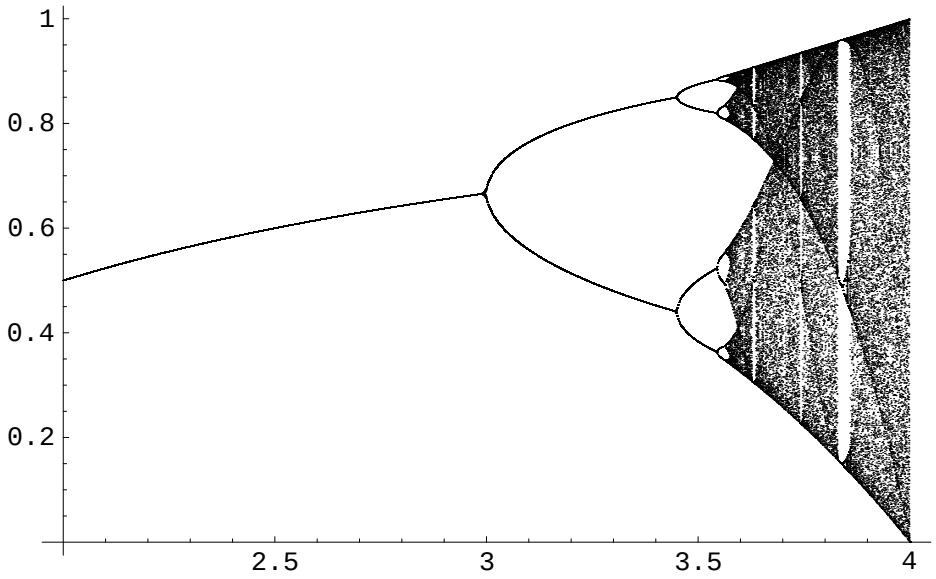


Figure 6. The Logistic Map.

story. Embedded within this chaotic regime are certain small ranges of λ where the system settles down to a stable orbit, whose period is no longer necessarily a power of 2. In fact, there exist values of λ for which the iterates settle down to a stable orbit of period k for *any* positive integer k. For instance, as λ increases past $\lambda_{3,\star} \approx 3.83$, a period 3 orbit appears over a small range of values, after which, as λ increses slightly further, there is a period doubling cascade where period 6, 12, 24, ... orbits successively appear, each persisting on a shorter and shorter range of parameter values, until λ passes yet another critical value where chaos breaks out yet again. There is a well-prescribed order in which the periodic orbits make their successive appearance, and each odd period k orbit is followed by a very closely spaced sequence of period doubling bifurcations, of periods $2^n k$ for $n = 1, 2, 3, \ldots$, after which the iterates revert to completely chaotic behavior until the next periodic case emerges. The ratios of distances between bifurcation points always have the same Feigenbaum limit (2.20). Finally, these periodic and chaotic windows all pile up on the ultimate parameter value $\lambda^\star_\star = 4$. And then, when $\lambda > 4$, all the iterates go off to ∞, and the system ceases to be interesting.

The reader is encouraged to write a simple computer program and perform some numerical experiments. In particular, Figure 6 shows the asymptotic behavior of the iterates for values of the parameter in the interesting range $2 < \lambda < 4$. The horizontal axis is λ, and the marked points show the ultimate fate of the iteration for the given value of λ. For instance, each point the single curve lying above the smaller values of λ represents a stable fixed point; this bifurcates into a pair of curves representing stable period 2 orbits, which then bifurcates into 4 curves representing period 4 orbits, and so on. Chaotic behavior is indicated by a somewhat random pattern of points lying above the value of λ. To plot this figure, we ran the logistic iteration $u^{(n)}$ for $0 \leq n \leq 100$, discarded the first 50 points, and then plotted the next 50 iterates $u^{(51)}, \ldots, u^{(100)}$. Investigation of the fine detailed structure of the logistic map requires yet more iterations with increased numerical accuracy. In addition one should discard more of the initial iterates so as to give

the system enough time to settle down to a stable periodic orbit or, alternatively, continue in a chaotic manner.

Remark: So far, we have only looked at real scalar iterative systems. Complex discrete dynamical systems display yet more remarkable and fascinating behavior. The complex version of the logistic iteration equation leads to the justly famous Julia and Mandelbrot sets, [**11**], with their stunning, psychedelic fractal structure, [**15**].

The rich range of phenomena in evidence, even in such extremely simple nonlinear iterative systems, is astounding. While intimations first appeared in the late nineteenth century research of the influential French mathematician Henri Poincaré, serious investigations were delayed until the advent of the computer era, which precipitated an explosion of research activity in the area of dynamical systems. Similar period doubling cascades and chaos are found in a broad range of nonlinear systems, [**1**], and are often encountered in physical applications, [**13**]. A modern explanation of fluid turbulence is that it is a (very complicated) form of chaos, [**1**].

Quadratic Convergence

Let us now return to the more mundane case when the iterates converge to a stable fixed point of the discrete dynamical system. In applications, we use the iterates to compute a precise† numerical value for the fixed point, and hence the efficiency of the algorithm depends on the speed of convergence of the iterates.

According to the remark following the proof Theorem 2.4, the convergence rate of an iterative system is essentially governed by the magnitude of the derivative $|\, g'(u^\star)\,|$ at the fixed point. The basic inequality (2.15) for the errors $e^{(k)} = u^{(k)} - u^\star$, namely

$$|\, e^{(k+1)}\,| \le \sigma\, |\, e^{(k)}\,|,$$

is known as a *linear convergence estimate*. It means that, once the iterates are close to the fixed point, the error decreases by a factor of (at least) $\sigma \approx |\, g'(u^\star)\,|$ at each step. If the k^{th} iterate $u^{(k)}$ approximates the fixed point $u^\star$ correctly to m decimal places, so its error is bounded by

$$|\, e^{(k)}\,| < .5 \times 10^{-m},$$

then the $(k+1)^{\text{st}}$ iterate satisfies the error bound

$$|\, e^{(k+1)}\,| \le \sigma\, |\, e^{(k)}\,| < .5 \times 10^{-m}\, \sigma = .5 \times 10^{-m + \log_{10} \sigma}.$$

More generally, for any $j > 0$,

$$|\, e^{(k+j)}\,| \le \sigma^j\, |\, e^{(k)}\,| < .5 \times 10^{-m}\, \sigma^j = .5 \times 10^{-m + j \log_{10} \sigma},$$

which means that the $(k+j)^{\text{th}}$ iterate $u^{(k+j)}$ has at least†

$$m - j\, \log_{10} \sigma = m + j\, \log_{10} \sigma^{-1}$$

† The degree of precision is to be specified by the user and the application.

† Note that since $\sigma < 1$, the logarithm $\log_{10} \sigma^{-1} = -\log_{10} \sigma > 0$ is positive.

correct decimal places. For instance, if $\sigma = .1$ then each new iterate produces one new decimal place of accuracy (at least), while if $\sigma = .9$ then it typically takes $22 \approx -1/\log_{10} .9$ iterates to produce just one additional accurate digit!

This means that there is a huge advantage — particularly in the application of iterative methods to the numerical solution of equations — to arrange that $|\, g'(u^\star)\,|$ be as small as possible. The fastest convergence rate of all will occur when $g'(u^\star) = 0$. In fact, in such a happy situation, the rate of convergence is not just slightly, but dramatically faster than linear.

Theorem 2.8. *Suppose that*[‡] $g \in \mathrm{C}^2$, *and* $u^\star = g(u^\star)$ *is a fixed point such that* $g'(u^\star) = 0$. *Then, for all iterates* $u^{(k)}$ *sufficiently close to* $u^\star$, *the errors* $e^{(k)} = u^{(k)} - u^\star$ *satisfy the* quadratic convergence estimate

$$|\, e^{(k+1)}\,| \;\le\; \tau\, |\, e^{(k)}\,|^2 \tag{2.21}$$

for some constant $\tau > 0$.

Proof: Just as that of the linear convergence estimate (2.15), the proof relies on approximating $g(u)$ by a simpler function near the fixed point. For linear convergence, an affine approximation sufficed, but here we require a higher order approximation. Thus, we replace the mean value formula (2.13) by the first order Taylor expansion

$$g(u) = g(u^\star) + g'(u^\star)\,(u - u^\star) + \tfrac{1}{2}\, g''(w)\,(u - u^\star)^2, \tag{2.22}$$

where the final error term depends on an (unknown) point w that lies between u and $u^\star$. At a fixed point, the constant term is $g(u^\star) = u^\star$. Furthermore, under our hypothesis $g'(u^\star) = 0$, and so (2.22) reduces to

$$g(u) - u^\star = \tfrac{1}{2}\, g''(w)\,(u - u^\star)^2.$$

Therefore,

$$|\, g(u) - u^\star\,| \le \tau\, |\, u - u^\star\,|^2, \tag{2.23}$$

where τ is chosen so that

$$\tfrac{1}{2}\, |\, g''(w)\,| \le \tau \tag{2.24}$$

for all w sufficiently close to $u^\star$. Therefore, the magnitude of τ is governed by the size of the *second derivative* of the iterative function $g(u)$ near the fixed point. We use the inequality (2.23) to estimate the error

$$|\, e^{(k+1)}\,| = |\, u^{(k+1)} - u^\star\,| = |\, g(u^{(k)}) - g(u^\star)\,| \;\le\; \tau\, |\, u^{(k)} - u^\star\,|^2 = \tau\, |\, e^{(k)}\,|^2,$$

which establishes the quadratic convergence estimate (2.21). *Q.E.D.*

[‡] The notation means that $g(u)$ is twice continuously differentiable, i.e., g, g', g'' are all defined and continuous near the fixed point $u^\star$.

Let us see how the quadratic estimate (2.21) speeds up the convergence rate. Following our earlier argument, suppose $u^{(k)}$ is correct to m decimal places, so

$$| e^{(k)} | < .5 \times 10^{-m}.$$

Then (2.21) implies that

$$| e^{(k+1)} | < .5 \times (10^{-m})^2 \tau = .5 \times 10^{-2m + \log_{10} \tau},$$

and so $u^{(k+1)}$ has $2m - \log_{10} \tau$ accurate decimal places. If $\tau \approx | g''(u^\star) |$ is of moderate size, we have essentially *doubled* the number of accurate decimal places in just a single iterate! A second iteration will double the number of accurate digits yet again. Thus, the convergence of a quadratic iteration scheme is *extremely* rapid, and, barring round-off errors, one can produce any desired number of digits of accuracy in a very short time. For example, if we start with an initial guess that is accurate in the first decimal digit, then a linear iteration with $\sigma = .1$ will require 49 iterations to obtain 50 decimal place accuracy, whereas a quadratic iteration (with $\tau = 1$) will only require 6 iterations to obtain $2^6 = 64$ decimal places of accuracy!

Example 2.9. Consider the function

$$g(u) = \frac{2u^3 + 3}{3u^2 + 3}.$$

There is a unique (real) fixed point $u^\star = g(u^\star)$, which is the real solution to the cubic equation

$$\tfrac{1}{3} u^3 + u - 1 = 0.$$

Note that

$$g'(u) = \frac{2u^4 + 6u^2 - 6u}{3(u^2 + 1)^2} = \frac{6u\left(\tfrac{1}{3} u^3 + u - 1\right)}{3(u^2 + 1)^2},$$

and hence $g'(u^\star) = 0$ vanishes at the fixed point. Theorem 2.8 implies that the iterations will exhibit quadratic convergence to the root. Indeed, we find, starting with $u^{(0)} = 0$, the following values:

k	1	2	3
$u^{(k)}$	1.00000000000000	.833333333333333	.817850637522769

4	5	6
.817731680821982	.817731673886824	.817731673886824

The convergence rate is dramatic: after only 5 iterations, we have produced the first 15 decimal places of the fixed point. In contrast, the linearly convergent scheme based on $\widetilde{g}(u) = 1 - \frac{1}{3} u^3$ takes 29 iterations just to produce the first 5 decimal places of the same solution.

In practice, the appearance of a quadratically convergent fixed point is a matter of luck. The construction of quadratically convergent iterative methods for solving equations will be the focus of the following Section 3.

Vector–Valued Iteration

Extending the preceding analysis to vector-valued iterative systems is not especially difficult. We will build on our experience with linear iterative systems, and so the reader is advised to review the basic concepts and results from Chapter 10 of [**14**] before proceeding to the nonlinear systems presented here.

We begin by fixing a norm $\| \cdot \|$ on $\mathbb{R}^n$. Since we will also be computing the associated matrix norm $\| A \|$, as defined in [**14**], it may be more convenient for computations to adopt either the 1 or the ∞ norms rather than the standard Euclidean norm.

We begin by defining the vector-valued counterpart of the basic linear convergence condition (2.18).

Definition 2.10. A function $\mathbf{g}\colon \mathbb{R}^n \to \mathbb{R}^n$ is a *contraction* at a point $\mathbf{u}^\star \in \mathbb{R}^n$ if there exists a constant $0 \le \sigma < 1$ such that

$$\| \mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{u}^\star) \| \;\le\; \sigma \, \| \mathbf{u} - \mathbf{u}^\star \| \tag{2.25}$$

for all $\mathbf{u}$ sufficiently close to $\mathbf{u}^\star$, i.e., $\| \mathbf{u} - \mathbf{u}^\star \| < \delta$ for some fixed $\delta > 0$.

Remark: The notion of a contraction depends on the underlying choice of matrix norm. Indeed, the linear function $\mathbf{g}(\mathbf{u}) = A\mathbf{u}$ if and only if $\| A \| < 1$, which implies that A is a convergent matrix. While every convergent matrix satisfies $\| A \| < 1$ in *some* matrix norm, and hence defines a contraction relative to that norm, it may very well have $\| A \| > 1$ in a particular norm, violating the contraction condition.

Theorem 2.11. *If $\mathbf{u}^\star = \mathbf{g}(\mathbf{u}^\star)$ is a fixed point for the discrete dynamical system (2.1) and $\mathbf{g}$ is a contraction at $\mathbf{u}^\star$, then $\mathbf{u}^\star$ is an asymptotically stable fixed point.*

Proof: The proof is a copy of the last part of the proof of Theorem 2.4. We write

$$\| \mathbf{u}^{(k+1)} - \mathbf{u}^\star \| = \| \mathbf{g}(\mathbf{u}^{(k)}) - \mathbf{g}(\mathbf{u}^\star) \| \le \sigma \, \| \mathbf{u}^{(k)} - \mathbf{u}^\star \|,$$

using the assumed estimate (2.25). Iterating this basic inequality immediately demonstrates that

$$\| \mathbf{u}^{(k)} - \mathbf{u}^\star \| \;\le\; \sigma^k \, \| \mathbf{u}^{(0)} - \mathbf{u}^\star \| \qquad \text{for} \qquad k = 0, 1, 2, 3, \dots . \tag{2.26}$$

Since $\sigma < 1$, the right hand side tends to 0 as $k \to \infty$, and hence $\mathbf{u}^{(k)} \to \mathbf{u}^\star$. *Q.E.D.*

In most interesting situations, the function $\mathbf{g}$ is differentiable, and so can be approximated by its first order Taylor polynomial:

$$\mathbf{g}(\mathbf{u}) \approx \mathbf{g}(\mathbf{u}^\star) + \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star) = \mathbf{u}^\star + \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star). \tag{2.27}$$

Here

$$\mathbf{g}'(\mathbf{u}) = \begin{pmatrix} \dfrac{\partial g_1}{\partial u_1} & \dfrac{\partial g_1}{\partial u_2} & \cdots & \dfrac{\partial g_1}{\partial u_n} \\ \dfrac{\partial g_2}{\partial u_1} & \dfrac{\partial g_2}{\partial u_2} & \cdots & \dfrac{\partial g_2}{\partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial g_n}{\partial u_1} & \dfrac{\partial g_n}{\partial u_2} & \cdots & \dfrac{\partial g_n}{\partial u_n} \end{pmatrix}, \tag{2.28}$$

denotes the $n \times n$ *Jacobian matrix* of the vector-valued function $\mathbf{g}$, whose entries are the partial derivatives of its individual components. Since $\mathbf{u}^\star$ is fixed, the the right hand side of (2.27) is an affine function of $\mathbf{u}$. Moreover, $\mathbf{u}^\star$ remains a fixed point of the affine approximation. Proposition 10.44 of [**14**] tells us that iteration of the affine function will converge to the fixed point if and only if its coefficient matrix, namely $\mathbf{g}'(\mathbf{u}^\star)$, is a convergent matrix, meaning that its spectral radius $\rho(\mathbf{g}'(\mathbf{u}^\star)) < 1$. This observation motivates the following theorem and corollary.

Theorem 2.12. *Let $\mathbf{u}^\star$ be a fixed point for the discrete dynamical system $\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)})$. If the Jacobian matrix norm $\| \mathbf{g}'(\mathbf{u}^\star) \| < 1$, then $\mathbf{g}$ is a contraction at $\mathbf{u}^\star$, and hence the fixed point $\mathbf{u}^\star$ is asymptotically stable.*

Proof: The first order Taylor expansion of $\mathbf{g}(\mathbf{u})$ at the fixed point $\mathbf{u}^\star$ takes the form

$$\mathbf{g}(\mathbf{u}) = \mathbf{g}(\mathbf{u}^\star) + \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star) + R(\mathbf{u} - \mathbf{u}^\star), \tag{2.29}$$

where the remainder term satisfies

$$\lim_{\mathbf{u} \to \mathbf{u}^\star} \frac{R(\mathbf{u} - \mathbf{u}^\star)}{\| \mathbf{u} - \mathbf{u}^\star \|} \;=\; 0.$$

Let $\varepsilon > 0$ be such that

$$\sigma = \| \mathbf{g}'(\mathbf{u}^\star) \| + \varepsilon < 1.$$

Choose $0 < \delta < 1$ such that $\| R(\mathbf{u} - \mathbf{u}^\star) \| \le \varepsilon \| \mathbf{u} - \mathbf{u}^\star \|$ whenever $\| \mathbf{u} - \mathbf{u}^\star \| \le \delta$. For such $\mathbf{u}$, we have, by the Triangle Inequality,

$$\begin{aligned} \| \mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{u}^\star) \| \;&\le\; \| \mathbf{g}'(\mathbf{u}^\star)\,(\mathbf{u} - \mathbf{u}^\star) \| + \| R(\mathbf{u} - \mathbf{u}^\star) \| \\ &\le\; \big(\| \mathbf{g}'(\mathbf{u}^\star) \| + \varepsilon \big) \| \mathbf{u} - \mathbf{u}^\star \| = \sigma \| \mathbf{u} - \mathbf{u}^\star \|, \end{aligned}$$

which establishes the contraction inequality (2.25). *Q.E.D.*

Corollary 2.13. *If the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star)$ is a convergent matrix, meaning that its spectral radius satisfies $\rho\big(\mathbf{g}'(\mathbf{u}^\star)\big) < 1$, then $\mathbf{u}^\star$ is an asymptotically stable fixed point.*

Proof: Corollary 10.32 in [**14**] assures us that $\| \mathbf{g}'(\mathbf{u}^\star) \| < 1$ in some matrix norm. Using this norm, the result immediately follows from the theorem. *Q.E.D.*

Theorem 2.12 tells us that initial values $\mathbf{u}^{(0)}$ that are sufficiently near a stable fixed point $\mathbf{u}^\star$ are guaranteed to converge to it. In the linear case, closeness of the initial data to the fixed point was not, in fact, an issue; all stable fixed points are, in fact, globally stable. For nonlinear iteration, it is of critical importance, and one does not typically expect iteration starting with far away initial data to converge to the desired fixed point. An interesting (and difficult) problem is to determine the so-called *basin of attraction* of a stable fixed point, defined as the set of all initial data that ends up converging to it. As in the elementary logistic map (2.19), initial values that lie outside a basin of attraction can lead to divergent iterates, periodic orbits, or even exhibit chaotic behavior. The full range of possible phenomena is a topic of contemporary research in dynamical systems theory, [**1**, **5**, **15**, **17**], and in numerical analysis, [**3**, **4**].

Alternatively, a fixed point is *unstable* when the spectral radius of its Jacobian matrix is strictly greater than one, meaning that it has at least one eigenvalue of modulus $|\,\lambda\,| > 1$. More precisely, the following result holds, cf. [**9**; Theorem 5.1.5].

Theorem 2.14. *Let $\mathbf{u}^\star$ be a fixed point for the discrete dynamical system $\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)})$. If the Jacobian matrix has spectral radius $\rho\big(\,\mathbf{g}'(\mathbf{u}^\star)\,\big) > 1$, then $\mathbf{u}^\star$ is an* unstable fixed point*, in the sense that there exists $r > 0$ with the property that one can find initial points $\mathbf{u}^{(0)}$ lying arbitrarily close to the fixed point $\mathbf{u}^\star$ that produce some eventual iterate $\mathbf{u}^{(k)}$ that lies a distance at least r away from $\mathbf{u}^\star$; that is, $\|\,\mathbf{u}^{(k)} - \mathbf{u}^\star\,\| \geq r$ for some $k \geq 1$, depending upon the initial point $\mathbf{u}^{(0)}$.*

In fact, most initial points near $\mathbf{u}^\star$ exhibit such instability. However, keep in mind that, while the theorem states that, for suitable initial data, *some* subsequent iterate will lie outside the ball of radius r centered at $\mathbf{u}^\star$, it is not claimed that this is necessarily true for *all* subsequent iterates, since it is possible that later iterates re-enter the ball and even (albeit highly unlikely), if they happen to then land on the so-called *stable manifold*[†] of the fixed point, eventually converging to it.

Example 2.15. Consider the function

$$\mathbf{g}(u,v) = \begin{pmatrix} -\frac{1}{4}\,u^3 + \frac{9}{8}\,u + \frac{1}{4}\,v^3 \\ \frac{3}{4}\,v - \frac{1}{2}\,uv \end{pmatrix}.$$

There are four (real) fixed points; stability is determined by the size of the eigenvalues of the Jacobian matrix

$$\mathbf{g}'(u,v) = \begin{pmatrix} \frac{9}{8} - \frac{3}{4}\,u^2 & -\frac{1}{2}\,v \\ \frac{3}{4}\,v^2 & \frac{3}{4} - \frac{1}{2}\,u \end{pmatrix}$$

[†] For linear iterative systems, the stable manifold of the origin coincides with the stable subspace spanned by the real and imaginary parts of eigenvectors corresponding to the stable eigenvalues of modulus $|\,\lambda\,| < 1$. In the nonlinear case, the stable manifold at $\mathbf{u}^\star$ is tangent to the stable subspace of its Jacobian matrix. Practically detecting such convergent points whose iterates eventually lie on the stable manifold is rather challenging, since (almost) any small numerical error can dislodge the iterate off the stable submanifold, causing it to eventually go away from the fixed point again.

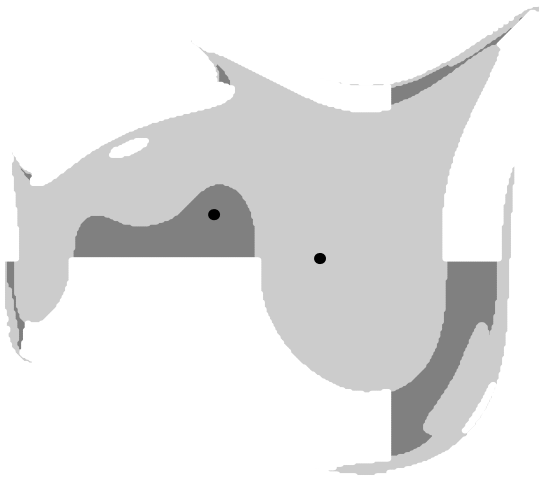


Figure 7. Basins of Attraction.

at each of the fixed points. The results are summarized in the following table:

fixed point	$\mathbf{u}_1^\star = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\mathbf{u}_2^\star = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$	$\mathbf{u}_3^\star = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$	$\mathbf{u}_4^\star = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$
Jacobian matrix	$\begin{pmatrix} \frac{9}{8} & 0 \\ 0 & \frac{3}{4} \end{pmatrix}$	$\begin{pmatrix} \frac{3}{4} & 0 \\ 0 & \frac{3}{4} - \frac{1}{2\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{3}{4} & 0 \\ 0 & \frac{3}{4} + \frac{1}{2\sqrt{2}} \end{pmatrix}$	$\begin{pmatrix} \frac{15}{16} & -\frac{1}{4} \\ \frac{3}{16} & 1 \end{pmatrix}$
eigenvalues	$1.125,\ .75$	$.75,\ .396447$	$1.10355,\ .75$	$.96875 \pm .214239\,\mathrm{i}$
spectral radius	1.125	.75	1.10355	.992157

Thus, $\mathbf{u}_2^\star$ and $\mathbf{u}_4^\star$ are stable fixed points, whereas $\mathbf{u}_1^\star$ and $\mathbf{u}_3^\star$ are both unstable. Indeed, starting with $\mathbf{u}^{(0)} = (\,.5, .5\,)^T$, it takes 24 iterates to converge to $\mathbf{u}_2^\star$ with 4 significant decimal digits, whereas starting with $\mathbf{u}^{(0)} = (\,-.7, .7\,)^T$, it takes 1049 iterates to converge to within 4 digits of $\mathbf{u}_4^\star$; the slower convergence rate is predicted by the larger Jacobian spectral radius. The two basins of attraction are plotted in Figure 7. The stable fixed points are indicated by black dots. The light gray region contains $\mathbf{u}_2^\star$ and indicates all the points that converge to it; the darker gray indicates points converging, more slowly, to $\mathbf{u}_4^\star$. All other initial points, except $\mathbf{u}_1^\star$ and $\mathbf{u}_3^\star$, have rapidly unbounded iterates: $\| \mathbf{u}^{(k)} \| \to \infty$.

The smaller the spectral radius or matrix norm of the Jacobian matrix at the fixed point, the faster the nearby iterates will converge to it. As in the scalar case, quadratic convergence will occur when the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star) = \mathrm{O}$ is the zero matrix, i.e., *all* first order partial derivatives of the components of $\mathbf{g}$ vanish at the fixed point. The quadratic convergence estimate

$$\| \mathbf{u}^{(k+1)} - \mathbf{u}^\star \| \ \le\ \tau \, \| \mathbf{u}^{(k)} - \mathbf{u}^\star \|^2 \tag{2.30}$$

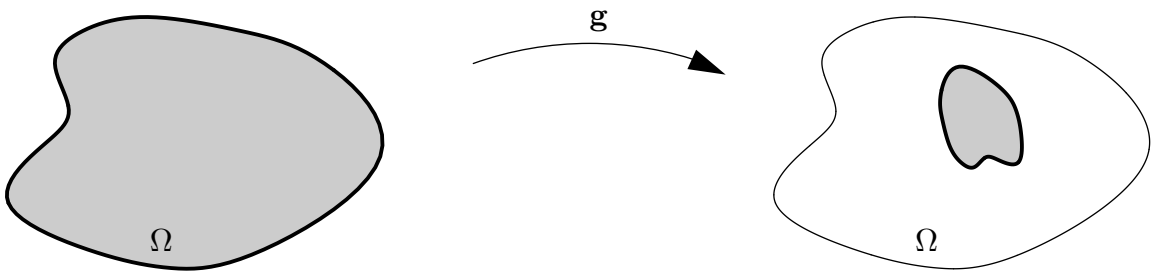


Figure 8. A Contraction Mapping.

is a consequence of the second order Taylor expansion at the fixed point. Details of the proof are left as an exercise.

Of course, in practice we don't know the norm or spectral radius of the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star)$ because we don't know where the fixed point is. This apparent difficulty can be easily circumvented by requiring that $\| \mathbf{g}'(\mathbf{u}) \| < 1$ for all $\mathbf{u}$ — or, at least, for all $\mathbf{u}$ in a domain Ω containing the fixed point. In fact, this hypothesis can be used to prove the existence and uniqueness of asymptotically stable fixed points. Rather than work with the Jacobian matrix, let us return to the contraction condition (2.25), but now imposed uniformly on an entire domain.

Definition 2.16. A function $\mathbf{g}\colon \mathbb{R}^n \to \mathbb{R}^n$ is called a *contraction mapping* on a domain $\Omega \subset \mathbb{R}^n$ if

(*a*) it maps Ω to itself, so $\mathbf{g}(\mathbf{u}) \in \Omega$ whenever $\mathbf{u} \in \Omega$, and

(*b*) there exists a *constant* $0 \le \sigma < 1$ such that

$$\| \mathbf{g}(\mathbf{u}) - \mathbf{g}(\mathbf{v}) \| \;\le\; \sigma \, \| \mathbf{u} - \mathbf{v} \| \qquad \text{for all} \qquad \mathbf{u}, \mathbf{v} \in \Omega. \tag{2.31}$$

In other words, applying a contraction mapping reduces the mutual distance between points. In view of Theorem 2.12, we can detect contraction mappings by looking at the norm of their Jacobian matrix.

Lemma 2.17. *If $\mathbf{g}\colon \Omega \to \Omega$ and $\| \mathbf{g}'(\mathbf{u}) \| < 1$ for all $\mathbf{u} \in \Omega$, then $\mathbf{g}$ is a contraction mapping.*

So, as its name indicates, a contraction mapping effectively shrinks the size of its domain; see Figure 8. As the iterations proceed, the successive image domains become smaller and smaller. If the original domain is closed and bounded, then it is forced to shrink down to a single point, which is the unique fixed point of the iterative system, leading to the *Contraction Mapping Theorem*.

Theorem 2.18. *If $\mathbf{g}$ is a contraction mapping on a closed bounded domain $\Omega \subset \mathbb{R}^n$, then $\mathbf{g}$ admits a unique fixed point $\mathbf{u}^\star \in \Omega$. Moreover, starting with any initial point $\mathbf{u}^{(0)} \in \Omega$, the iterates $\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)})$ necessarily converge to the fixed point: $\mathbf{u}^{(k)} \to \mathbf{u}^\star$.*

Proof: The basic contraction estimate (2.31) implies that, for any positive integers k, l,

$$\| \mathbf{u}^{(k+1)} - \mathbf{u}^{(l+1)} \| = \| \mathbf{g}(\mathbf{u}^{(k)}) - \mathbf{g}(\mathbf{u}^{(l)}) \| \le \sigma \, \| \mathbf{u}^{(k)} - \mathbf{u}^{(l)} \|.$$

Iterating, we find

$$\| \mathbf{u}^{(k+n)} - \mathbf{u}^{(l+n)} \| \leq \sigma^n \| \mathbf{u}^{(k)} - \mathbf{u}^{(l)} \| \leq \sigma^n D,$$

where

$$D = \max \{ \| \mathbf{u} - \mathbf{v} \| \mid \ \mathbf{u}, \mathbf{v} \in \Omega \} < \infty,$$

the *diameter* of Ω, is bounded since Ω is a bounded domain. Thus, to prove $\mathbf{u}^{(k)}$ forms a Cauchy sequence, given $\varepsilon > 0$, choose N such that $\sigma^N D < \varepsilon$, which is possible since $0 \leq \sigma < 1$. Then the preceding estimate show that

$$\| \mathbf{u}^{(k)} - \mathbf{u}^{(l)} \| \leq \sigma^N \| \mathbf{u}^{(k-N)} - \mathbf{u}^{(l-N)} \| \leq \sigma^N D < \varepsilon,$$

whenever $k, l \geq N$, proving the result. Since we know $\mathbf{u}^{(k)} \to \mathbf{u}^\star$ converges, the usual argument proves that $u\star$ is a fixed point.

To prove uniqueness of the fixed point, suppose that there are two fixed points, so $\mathbf{g}(\mathbf{u}^\star) = \mathbf{u}^\star$ and $\mathbf{g}(\widetilde{\mathbf{u}}^\star) = \widetilde{\mathbf{u}}^\star$. Then, by (2.25)

$$0 \leq \| \widetilde{\mathbf{u}}^\star - \mathbf{u}^\star \| = \| \mathbf{g}(\widetilde{\mathbf{u}}^\star) - \mathbf{g}(\mathbf{u}^\star) \| \leq \sigma \| \widetilde{\mathbf{u}}^\star - \mathbf{u}^\star \| < \| \widetilde{\mathbf{u}}^\star - \mathbf{u}^\star \|,$$

which implies $\| \widetilde{\mathbf{u}}^\star - \mathbf{u}^\star \| = 0$ and hence $\widetilde{\mathbf{u}}^\star = \mathbf{u}^\star$, proving the result. *Q.E.D.*

Example 2.19. The function

$$g(u) = u + \tfrac{1}{2}\pi - \tan^{-1} u \qquad \text{satisfies} \qquad | g'(u) | = \left| 1 - \frac{1}{1+u^2} \right| < 1$$

for all $u \in \mathbb{R}$, and hence defines a contraction mapping. However, $g(u)$ has no fixed point. Why does this not contradict Theorem 2.18?

3. Solution of Equations and Systems.

Solving nonlinear equations and systems of equations is, of course, a problem of utmost importance in mathematics and its manifold applications. It can also be extremely difficult. Indeed, finding a complete set of (numerical) solutions to a complicated nonlinear system can be an almost insurmountable challenge. In its most general version, we are given a collection of m functions $f_1, \dots, f_m$ depending upon n variables $u_1, \dots, u_n$, and are asked to determine all possible solutions $\mathbf{u} = (\, u_1, u_2, \dots, u_n \,)^T$ to the system

$$f_1(u_1, \dots, u_n) = 0, \qquad \dots \qquad f_m(u_1, \dots, u_n) = 0. \tag{3.1}$$

In many applications, the number of equations equals the number of unknowns, $m = n$, in which case one expects both existence and uniqueness of solutions. This point will be discussed in further detail below.

Here are some prototypical examples:

(*a*) Find the roots of the quintic polynomial equation

$$u^5 + u + 1 = 0. \tag{3.2}$$

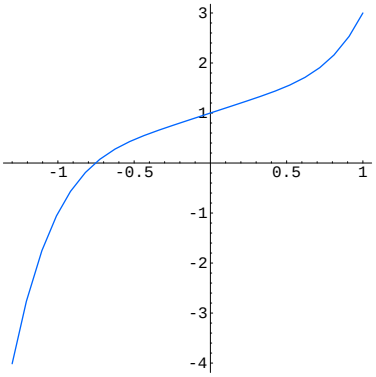


Figure 9. Graph of $u^5 + u + 1$.

Graphing the left hand side of the equation, as in Figure 9, convinces us that there is just one real root, lying somewhere between -1 and $-.5$. While there are explicit algebraic formulas for the roots of quadratic, cubic, and quartic polynomials, a famous theorem[†] due to the Norwegian mathematician Nils Henrik Abel in the early 1800's states that there is *no* such formula for generic fifth order polynomial equations.

(*b*) Any fixed point equation $u = g(u)$ has the form (3.4) where $f(u) = u - g(u)$. For example, the trigonometric Kepler equation

$$u - \epsilon \sin u = m$$

arises in the study of planetary motion, cf. Example 2.5. Here ϵ, m are fixed constants, and we seek a corresponding solution u.

(*c*) Suppose we are given chemical compounds A, B, C that react to produce a fourth compound D according to

$$2\,A + B \;\longleftrightarrow\; D, \qquad\qquad A + 3\,C \;\longleftrightarrow\; D.$$

Let a, b, c be the initial concentrations of the reagents A, B, C injected into the reaction chamber. If u denotes the concentration of D produced by the first reaction, and v that by the second reaction, then the final equilibrium concentrations

$$a_\star = a - 2\,u - v, \qquad b_\star = b - u, \qquad c_\star = c - 3\,v, \qquad d_\star = u + v,$$

of the reagents will be determined by solving the nonlinear system

$$(a - 2\,u - v)^2(b - u) = \alpha\,(u + v), \qquad (a - 2\,u - v)(c - 3\,v)^3 = \beta\,(u + v), \tag{3.3}$$

where α, β are the known equilibrium constants of the two reactions.

Our immediate goal is to develop numerical algorithms for solving such nonlinear equations. Unfortunately, there is no direct universal solution method for nonlinear systems comparable to Gaussian elimination. As a result, numerical solution techniques rely almost exclusively on iterative algorithms. This section presents the principal methods for

[†] A modern proof of this fact relies on Galois theory, [**7**].

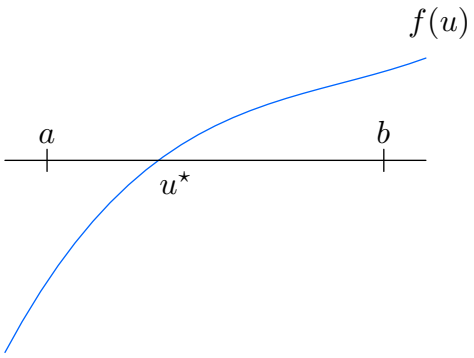


Figure 10. Intermediate Value Theorem.

numerically approximating the solution(s) to a nonlinear system. We shall only discuss general purpose algorithms; specialized methods for solving particular classes of equations, e.g., polynomial equations, can be found in numerical analysis texts, e.g., [**3**, **4**, **16**]. Of course, the most important specialized methods — those designed for solving linear systems — will continue to play a critical role, even in the nonlinear regime.

The Bisection Method

We begin, as always, with the scalar case. Thus, we are given a real-valued function $f\colon \mathbb{R} \to \mathbb{R}$, and seek its *roots*, i.e., the real[†] solution(s) to the scalar equation

$$f(u) = 0. \tag{3.4}$$

Our immediate goal is to develop numerical algorithms for solving such nonlinear scalar equations. The most primitive algorithm, and *the only one that is guaranteed to work in all cases*, is the Bisection Method. While it has an iterative flavor, it cannot be properly classed as a method governed by functional iteration as defined in the preceding section, and so must be studied directly in its own right.

The starting point is the Intermediate Value Theorem, which we state in simplified form. See Figure 10 for an illustration, and [**2**] for a proof.

Lemma 3.1. *Let $f(u)$ be a continuous scalar function. Suppose we can find two points $a < b$ where the values of $f(a)$ and $f(b)$ take* opposite *signs, so either $f(a) < 0$ and $f(b) > 0$, or $f(a) > 0$ and $f(b) < 0$. Then there exists at least one point $a < u^\star < b$ where $f(u^\star) = 0$.*

Note that if $f(a) = 0$ or $f(b) = 0$, then finding a root is trivial. If $f(a)$ and $f(b)$ have the same sign, then there may or may not be a root in between. Figure 11 plots the functions $u^2 + 1$, u^2 and $u^2 - 1$, on the interval $-2 \le u \le 2$. The first has two simple roots; the second has a single double root, while the third has no root. We also note

[†] Complex roots to complex equations will be discussed later.

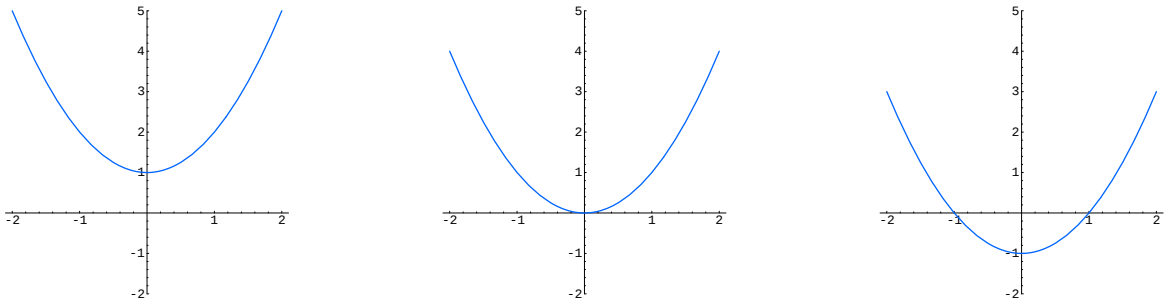


Figure 11. Roots of Quadratic Functions.

that continuity of the function on the entire interval $[a, b]$ is an essential hypothesis. For example, the function $f(u) = 1/u$ satisfies $f(-1) = -1$ and $f(1) = 1$, but there is no root to the equation $1/u = 0$.

Note carefully that the Lemma 3.1 does *not* say there is a unique root between a and b. There may be many roots, or even, in pathological examples, infinitely many. All the theorem guarantees is that, under the stated hypotheses, there is at least one root.

Once we are assured that a root exists, bisection relies on a "divide and conquer" strategy. The goal is to locate a root $a < u^\star < b$ between the endpoints. Lacking any additional evidence, one tactic would be to try the midpoint $c = \frac{1}{2}(a + b)$ as a first guess for the root. If, by some miracle, $f(c) = 0$, then we are done, since we have found a solution! Otherwise (and typically) we look at the sign of $f(c)$. There are two possibilities. If $f(a)$ and $f(c)$ are of opposite signs, then the Intermediate Value Theorem tells us that there is a root $u^\star$ lying between $a < u^\star < c$. Otherwise, $f(c)$ and $f(b)$ must have opposite signs, and so there is a root $c < u^\star < b$. In either event, we apply the same method to the interval in which we are assured a root lies, and repeat the procedure. Each iteration halves the length of the interval, and chooses the half in which a root is sure to lie. (There may, of course, be a root in the other half interval, but as we cannot be sure, we discard it from further consideration.) The root we home in on lies trapped in intervals of smaller and smaller width, and so convergence of the method is guaranteed.

Example 3.2. The roots of the quadratic equation

$$f(u) = u^2 + u - 3 = 0$$

can be computed exactly by the quadratic formula:

$$u_1^\star = \frac{-1 + \sqrt{13}}{2} \approx 1.302775\ldots, \qquad u_2^\star = \frac{-1 - \sqrt{13}}{2} \approx -2.302775\ldots.$$

Let us see how one might approximate them by applying the Bisection Algorithm. We start the procedure by choosing the points $a = u^{(0)} = 1$, $b = v^{(0)} = 2$, noting that $f(1) = -1$ and $f(2) = 3$ have opposite signs and hence we are guaranteed that there is at least one root between 1 and 2. In the first step we look at the midpoint of the interval $[1, 2]$, which is 1.5, and evaluate $f(1.5) = .75$. Since $f(1) = -1$ and $f(1.5) = .75$ have opposite signs, we know that there is a root lying between 1 and 1.5. Thus, we take $u^{(1)} = 1$ and $v^{(1)} = 1.5$ as the endpoints of the next interval, and continue. The next midpoint is at

k	$u^{(k)}$	$v^{(k)}$	$w^{(k)} = \frac{1}{2}(u^{(k)} + v^{(k)})$	$f(w^{(k)})$
0	1	2	1.5	.75
1	1	1.5	1.25	−.1875
2	1.25	1.5	1.375	.2656
3	1.25	1.375	1.3125	.0352
4	1.25	1.3125	1.2813	−.0771
5	1.2813	1.3125	1.2969	−.0212
6	1.2969	1.3125	1.3047	.0069
7	1.2969	1.3047	1.3008	−.0072
8	1.3008	1.3047	1.3027	−.0002
9	1.3027	1.3047	1.3037	.0034
10	1.3027	1.3037	1.3032	.0016
11	1.3027	1.3032	1.3030	.0007
12	1.3027	1.3030	1.3029	.0003
13	1.3027	1.3029	1.3028	.0001
14	1.3027	1.3028	1.3028	−.0000

1.25, where $f(1.25) = -.1875$ has the opposite sign to $f(1.5) = .75$, and so a root lies between $u^{(2)} = 1.25$ and $v^{(2)} = 1.5$. The process is then iterated as long as desired — or, more practically, as long as your computer's precision does not become an issue.

The table displays the result of the algorithm, rounded off to four decimal places. After 14 iterations, the Bisection Method has correctly computed the first four decimal digits of the positive root $u_1^\star$. A similar bisection starting with the interval from $u^{(1)} = -3$ to $v^{(1)} = -2$ will produce the negative root.

A formal implementation of the Bisection Algorithm appears in the accompanying pseudocode program. The endpoints of the k^{th} interval are denoted by $u^{(k)}$ and $v^{(k)}$. The midpoint is $w^{(k)} = \frac{1}{2}\big(u^{(k)} + v^{(k)}\big)$, and the key decision is whether $w^{(k)}$ should be the right or left hand endpoint of the next interval. The integer n, governing the number of iterations, is to be prescribed in accordance with how accurately we wish to approximate the root $u^\star$.

The algorithm produces two sequences of approximations $u^{(k)}$ and $v^{(k)}$ that both converge monotonically to $u^\star$, one from below and the other from above:

$$a = u^{(0)} \le u^{(1)} \le u^{(2)} \le \cdots \le u^{(k)} \longrightarrow u^\star \longleftarrow v^{(k)} \le \cdots \le v^{(2)} \le v^{(1)} \le v^{(0)} = b.$$

and $u^\star$ is trapped between the two. Thus, the root is trapped inside a sequence of intervals $[\,u^{(k)}, v^{(k)}\,]$ of progressively shorter and shorter length. Indeed, the length of each interval is exactly half that of its predecessor:

$$v^{(k)} - u^{(k)} = \tfrac{1}{2}\,(v^{(k-1)} - u^{(k-1)}).$$

The Bisection Method

```
start
    if f(a) f(b) < 0 set u^(0) = a, v^(0) = b
      else  print "Bisection Method not applicable"
    for k = 0 to n − 1
        set w^(k) = ½(u^(k) + v^(k))
        if f(w^(k)) = 0, stop; print u* = w^(k)
        if f(u^(k)) f(w^(k)) < 0, set u^(k+1) = u^(k), v^(k+1) = w^(k)
                        else     set u^(k+1) = w^(k), v^(k+1) = v^(k)
    next k
    print u* = w^(n) = ½(u^(n) + v^(n))
end
```

Iterating this formula, we conclude that

$$v^{(n)} - u^{(n)} = \left(\tfrac{1}{2}\right)^n (v^{(0)} - u^{(0)}) = \left(\tfrac{1}{2}\right)^n (b-a) \longrightarrow 0 \qquad \text{as} \qquad n \longrightarrow \infty.$$

The midpoint

$$w^{(n)} = \tfrac{1}{2}\,(u^{(n)} + v^{(n)})$$

lies within a distance

$$|\, w^{(n)} - u^\star \,| \le \tfrac{1}{2}\,(v^{(n)} - u^{(n)}) = \left(\tfrac{1}{2}\right)^{n+1} (b-a)$$

of the root. Consequently, if we desire to approximate the root within a prescribed tolerance ε, we should choose the number of iterations n so that

$$\left(\tfrac{1}{2}\right)^{n+1} (b-a) < \varepsilon, \qquad \text{or} \qquad n > \log_2 \frac{b-a}{\varepsilon} - 1\,. \tag{3.5}$$

Summarizing:

Theorem 3.3. *If $f(u)$ is a continuous function, with $f(a)\,f(b) < 0$, then the Bisection Method starting with $u^{(0)} = a$, $v^{(0)} = b$, will converge to a solution $u^\star$ to the equation $f(u) = 0$ lying between a and b. After n steps, the midpoint $w^{(n)} = \frac{1}{2}\,(u^{(n)} + v^{(n)})$ will be within a distance of $\varepsilon = 2^{-n-1}(b-a)$ from the solution.*

For example, in the case of the quadratic equation in Example 3.2, after 14 iterations, we have approximated the positive root to within

$$\varepsilon = \left(\tfrac{1}{2}\right)^{15} (2-1) \approx 3.052 \times 10^{-5},$$

reconfirming our observation that we have accurately computed its first four decimal places. If we are in need of 10 decimal places, we set our tolerance to $\varepsilon = .5 \times 10^{-10}$, and so, according to (3.5), must perform $n = 34 > 33.22 \approx \log_2 2 \times 10^{10} - 1$ successive bisections†.

Example 3.4. As noted at the beginning of this section, the quintic equation

$$f(u) = u^5 + u + 1 = 0$$

has one real root, whose value can be readily computed by bisection. We start the algorithm with the initial points $u^{(0)} = -1,\ v^{(0)} = 0$, noting that $f(-1) = -1 < 0$ while $f(0) = 1 > 0$ are of opposite signs. In order to compute the root to 6 decimal places, we set $\varepsilon = .5 \times 10^{-6}$ in (3.5), and so need to perform $n = 20 > 19.93 \approx \log_2 2 \times 10^6 - 1$ bisections. Indeed, the algorithm produces the approximation $u^\star \approx -.754878$ to the root, and the displayed digits are guaranteed to be accurate.

Fixed Point Methods

The Bisection Method has an ironclad guarantee to converge to a root of the function — provided it can be properly started by locating two points where the function takes opposite signs. This may be tricky if the function has two very closely spaced roots and is, say, negative only for a very small interval between them, and may be impossible for multiple roots, e.g., the root $u^\star = 0$ of the quadratic function $f(u) = u^2$. When applicable, its convergence rate is completely predictable, but not especially fast. Worse, it has no immediately apparent extension to systems of equations, since there is *no* obvious counterpart to the Intermediate Value Theorem for vector-valued functions.

Most other numerical schemes for solving equations rely on some form of fixed point iteration. Thus, we seek to replace the system of equations $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ with a fixed point system $\mathbf{u} = \mathbf{g}(\mathbf{u})$, that leads to the iterative solution scheme $\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)})$. For this to work, there are two key requirements:

(*a*) The solution $\mathbf{u}^\star$ to the equation $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ is also a fixed point for $\mathbf{g}(\mathbf{u})$, and

(*b*) $\mathbf{u}^\star$ is, in fact a stable fixed point, meaning that the Jacobian $\mathbf{g}'(\mathbf{u}^\star)$ is a convergent matrix, or, slightly more restrictively, $\| \mathbf{g}'(\mathbf{u}^\star) \| < 1$ for a prescribed matrix norm.

If both conditions hold, then, *provided we choose the initial iterate* $\mathbf{u}^{(0)} = \mathbf{c}$ *sufficiently close to* $\mathbf{u}^\star$, the iterates $\mathbf{u}^{(k)} \to \mathbf{u}^\star$ will converge to the desired solution. Thus, the key to the practical use of functional iteration for solving equations is the proper design of an iterative system — coupled with a reasonably good initial guess for the solution. Before implementing general procedures, let us discuss an elementary example.

Example 3.5. To solve the cubic equation

$$f(u) = u^3 - u - 1 = 0 \tag{3.6}$$

we note that $f(1) = -1$ while $f(2) = 5$, and so there is a root between 1 and 2. Indeed, the Bisection Method leads to the approximate value $u^\star \approx 1.3247$ after 17 iterations.

† This assumes we have sufficient precision on the computer to avoid round-off errors.

Let us try to find the same root by fixed point iteration. As a first, naïve, guess, we rewrite the cubic equation in fixed point form

$$u = u^3 - 1 = \widetilde{g}(u).$$

Starting with the initial guess $u^{(0)} = 1.5$, successive approximations to the solution are found by iterating

$$u^{(k+1)} = \widetilde{g}(u^{(k)}) = (u^{(k)})^3 - 1, \qquad k = 0, 1, 2, \dots .$$

However, their values

$$\begin{aligned} &u^{(0)} = 1.5, \qquad u^{(1)} = 2.375, \qquad u^{(2)} = 12.396, \\ &u^{(3)} = 1904, \qquad u^{(4)} = 6.9024 \times 10^9, \qquad u^{(5)} = 3.2886 \times 10^{29}, \qquad \dots \end{aligned}$$

rapidly become unbounded, and so fail to converge. This could, in fact, have been predicted by the convergence criterion in Theorem 2.4. Indeed, $\widetilde{g}'(u) = -3u^2$ and so $|\, \widetilde{g}'(u) \,| > 3$ for all $u \geq 1$, including the root $u^\star$. This means that $u^\star$ is an unstable fixed point, and the iterates cannot converge to it.

On the other hand, we can rewrite the equation (3.6) in the alternative iterative form

$$u = \sqrt[3]{1 + u} \; = g(u).$$

In this case

$$0 \;\leq\; g'(u) \;=\; \frac{1}{3(1+u)^{2/3}} \;\leq\; \frac{1}{3} \qquad \text{for} \qquad u > 0.$$

Thus, the stability condition (2.14) is satisfied, and we anticipate convergence at a rate of at least $\frac{1}{3}$. (The Bisection Method converges more slowly, at rate $\frac{1}{2}$.) Indeed, the first few iterates $u^{(k+1)} = \sqrt[3]{1 + u^{(k)}}$ are

$$1.5, \qquad 1.35721, \qquad 1.33086, \qquad 1.32588, \qquad 1.32494, \qquad 1.32476, \qquad 1.32473,$$

and we have converged to the root, correct to four decimal places, in only 6 iterations.

Newton's Method

Our immediate goal is to design an efficient iterative scheme $u^{(k+1)} = g(u^{(k)})$ whose iterates converge rapidly to the solution of the given scalar equation $f(u) = 0$. As we learned in Section 2, the convergence of the iteration is governed by the magnitude of its derivative at the fixed point. At the very least, we should impose the stability criterion $|\, g'(u^\star) \,| < 1$, and the smaller this quantity can be made, the faster the iterative scheme converges. if we are able to arrange that $g'(u^\star) = 0$, then the iterates will converge quadratically fast, leading, as noted in the discussion following Theorem 2.8, to a dramatic improvement in speed and efficiency.

Now, the first condition requires that $g(u) = u$ whenever $f(u) = 0$. A little thought will convince you that the iterative function should take the form

$$g(u) = u - h(u)\, f(u), \tag{3.7}$$

where $h(u)$ is a reasonably nice function. If $f(u^\star) = 0$, then clearly $u^\star = g(u^\star)$, and so $u^\star$ is a fixed point. The converse holds provided $h(u) \neq 0$ is never zero.

For quadratic convergence, the key requirement is that the derivative of $g(u)$ be zero at the fixed point solutions. We compute

$$g'(u) = 1 - h'(u)\, f(u) - h(u)\, f'(u).$$

Thus, $g'(u^\star) = 0$ at a solution to $f(u^\star) = 0$ if and only if

$$0 = 1 - h'(u^\star)\, f(u^\star) - h(u^\star)\, f'(u^\star) = 1 - h(u^\star)\, f'(u^\star).$$

Consequently, we should require that

$$h(u^\star) = \frac{1}{f'(u^\star)} \tag{3.8}$$

to ensure a quadratically convergent iterative scheme. This assumes that $f'(u^\star) \neq 0$, which means that $u^\star$ is a *simple root* of f. For here on, we leave aside multiple roots, which require a different approach.

Of course, there are many functions $h(u)$ that satisfy (3.8), since we only need to specify its value at a single point. The problem is that we do not know $u^\star$ — after all this is what we are trying to compute — and so cannot compute the value of the derivative of f there. However, we can circumvent this apparent difficulty by a simple device: we impose equation (3.8) at all points, setting

$$h(u) = \frac{1}{f'(u)}\,, \tag{3.9}$$

which certainly guarantees that it holds at the solution $u^\star$. The result is the function

$$g(u) = u \;-\; \frac{f(u)}{f'(u)}\,, \tag{3.10}$$

and the resulting iteration scheme is known as *Newton's Method*, which, as the name suggests, dates back to the founder of the calculus. To this day, Newton's Method remains *the* most important general purpose algorithm for solving equations. It starts with an initial guess $u^{(0)}$ to be supplied by the user, and then successively computes

$$u^{(k+1)} = u^{(k)} - \frac{f(u^{(k)})}{f'(u^{(k)})}\,. \tag{3.11}$$

As long as the initial guess is sufficiently close, the iterates $u^{(k)}$ are guaranteed to converge, quadratically fast, to the (simple) root $u^\star$ of the equation $f(u) = 0$.

Theorem 3.6. *Suppose $f(u) \in \mathrm{C}^2$ is twice continuously differentiable. Let $u^\star$ be a solution to the equation $f(u^\star) = 0$ such that $f'(u^\star) \neq 0$. Given an initial guess $u^{(0)}$ sufficiently close to $u^\star$, the Newton iteration scheme* (3.11) *converges at a quadratic rate to the solution $u^\star$.*

Proof: By continuity, if $f'(u^\star) \neq 0$, then $f'(u) \neq 0$ for all u sufficiently close to $u^\star$, and hence the Newton iterative function (3.10) is well defined and continuously differentiable near $u^\star$. Since $g'(u) = f(u)\, f''(u)/f'(u)^2$, we have $g'(u^\star) = 0$ when $f(u^\star) = 0$, as promised by our construction. The quadratic convergence of the resulting iterative scheme is an immediate consequence of Theorem 2.8. *Q.E.D.*

Example 3.7. Consider the cubic equation

$$f(u) = u^3 - u - 1 = 0,$$

that we already solved in Example 3.5. The function used in the Newton iteration is

$$g(u) = u - \frac{f(u)}{f'(u)} = u - \frac{u^3 - u - 1}{3\,u^2 - 1},$$

which is well-defined as long as $u \neq \pm \frac{1}{\sqrt{3}}$. We will try to avoid these singular points. The iterative procedure

$$u^{(k+1)} = g(u^{(k)}) = u^{(k)} - \frac{(u^{(k)})^3 - u^{(k)} - 1}{3\,(u^{(k)})^2 - 1}$$

with initial guess $u^{(0)} = 1.5$ produces the following values:

$$1.5, \qquad 1.34783, \qquad 1.32520, \qquad 1.32472,$$

and we have computed the root to 5 decimal places after only three iterations. The quadratic convergence of Newton's Method implies that, roughly, each new iterate doubles the number of correct decimal places. Thus, to compute the root accurately to 40 decimal places would only require 3 further iterations†. This underscores the tremendous advantage that the Newton algorithm offers over competing methods.

Example 3.8. Consider the cubic polynomial equation

$$f(u) = u^3 - \tfrac{3}{2}\,u^2 + \tfrac{5}{9}\,u - \tfrac{1}{27} = 0.$$

Since

$$f(0) = -\tfrac{1}{27}, \qquad f\big(\tfrac{1}{3}\big) = \tfrac{1}{54}, \qquad f\big(\tfrac{2}{3}\big) = -\tfrac{1}{27}, \qquad f(1) = \tfrac{1}{54},$$

the Intermediate Value Lemma 3.1 guarantees that there are three roots on the interval $[0, 1]$: one between 0 and $\frac{1}{3}$, the second between $\frac{1}{3}$ and $\frac{2}{3}$, and the third between $\frac{2}{3}$ and 1. The graph in Figure 12 reconfirms this observation. Since we are dealing with a cubic polynomial, there are no other roots. (Why?)

It takes sixteen iterations of the Bisection Method starting with the three subintervals $\big[0\,,\tfrac{1}{3}\big]$, $\big[\tfrac{1}{3}\,,\tfrac{2}{3}\big]$ and $\big[\tfrac{2}{3}\,,1\big]$, to produce the roots to six decimal places:

$$u_1^\star \approx .085119, \qquad u_2^\star \approx .451805, \qquad u_3^\star \approx .963076.$$

† This assumes we are working in a sufficiently high precision arithmetic so as to avoid round-off errors.

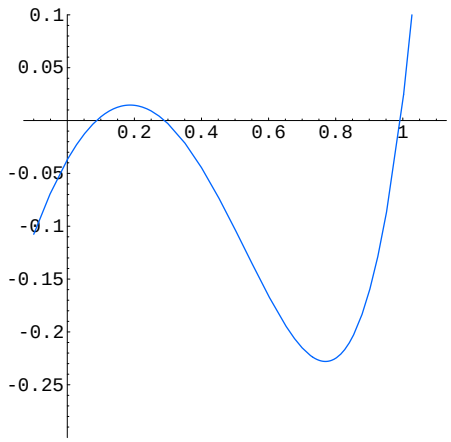


Figure 12. The function $f(u) = u^3 - \frac{3}{2}\,u^2 + \frac{5}{9}\,u - \frac{1}{27}$.

Incidentally, if we start with the interval $[0, 1]$ and apply bisection, we converge (perhaps surprisingly) to the largest root $u_3^\star$ in 17 iterations.

Fixed point iteration based on the formulation

$$u = g(u) = -u^3 + \tfrac{3}{2}\,u^2 + \tfrac{4}{9}\,u + \tfrac{1}{27}$$

can be used to find the first and third roots, but not the second root. For instance, starting with $u^{(0)} = 0$ produces $u_1^\star$ to 5 decimal places after 23 iterations, whereas starting with $u^{(0)} = 1$ produces $u_3^\star$ to 5 decimal places after 14 iterations. The reason we cannot produce $u_2^\star$ is due to the magnitude of the derivative

$$g'(u) = -3\,u^2 + 3\,u + \tfrac{4}{9}$$

at the roots, which is

$$g'(u_1^\star) \approx 0.678065, \qquad g'(u_2^\star) \approx 1.18748, \qquad g'(u_3^\star) \approx 0.551126.$$

Thus, $u_1^\star$ and $u_3^\star$ are stable fixed points, but $u_2^\star$ is unstable. However, because $g'(u_1^\star)$ and $g'(u_3^\star)$ are both bigger than .5, this iterative algorithm actually converges *slower* than ordinary bisection!

Finally, Newton's Method is based upon iteration of the rational function

$$g(u) = u - \frac{f(u)}{f'(u)} = u - \frac{u^3 - \frac{3}{2}\,u^2 + \frac{5}{9}\,u - \frac{1}{27}}{3\,u^2 - 3\,u + \frac{5}{9}}\,.$$

Starting with an initial guess of $u^{(0)} = 0$, the method computes $u_1^\star$ to 6 decimal places after only 4 iterations; starting with $u^{(0)} = .5$, it produces $u_2^\star$ to similar accuracy after 2 iterations; while starting with $u^{(0)} = 1$ produces $u_3^\star$ after 3 iterations — a dramatic speed up over the other two methods.

Newton's Method has a very pretty graphical interpretation, that helps us understand what is going on and why it converges so fast. Given the equation $f(u) = 0$, suppose we

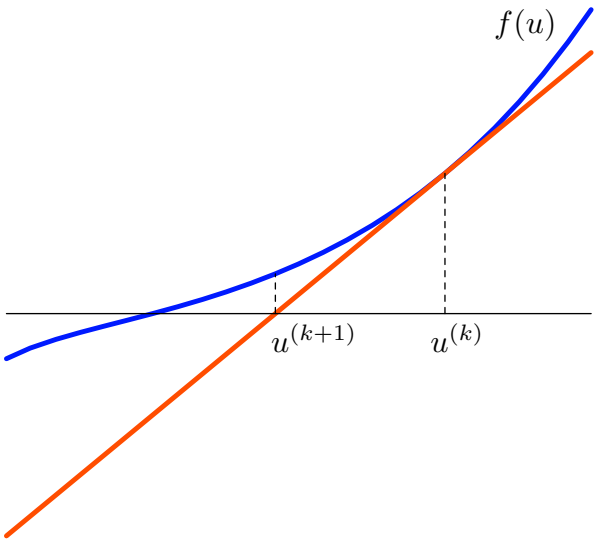


Figure 13. Newton's Method.

know an approximate value $u = u^{(k)}$ for a solution. Nearby $u^{(k)}$, we can approximate the nonlinear function $f(u)$ by its tangent line

$$y = f(u^{(k)}) + f'(u^{(k)})(u - u^{(k)}). \tag{3.12}$$

As long as the tangent line is not horizontal — which requires $f'(u^{(k)}) \neq 0$ — it crosses the axis at

$$u^{(k+1)} = u^{(k)} - \frac{f(u^{(k)})}{f'(u^{(k)})},$$

which represents a new, and, presumably more accurate, approximation to the desired root. The procedure is illustrated pictorially in Figure 13. Note that the passage from $u^{(k)}$ to $u^{(k+1)}$ is exactly the Newton iteration step (3.11). Thus, Newtonian iteration is the same as the approximation of function's root by those of its successive tangent lines.

Given a sufficiently accurate initial guess, Newton's Method will rapidly produce highly accurate values for the simple roots to the equation in question. In practice, barring some kind of special exploitable structure, Newton's Method is the root-finding algorithm of choice. The one caveat is that we need to start the process reasonably close to the root we are seeking. Otherwise, there is no guarantee that a particular set of iterates will converge, although if they do, the limiting value is necessarily a root of our equation. The behavior of Newton's Method as we change parameters and vary the initial guess is very similar to the simpler logistic map that we studied in Section 2, including period doubling bifurcations and chaotic behavior. The reader is invited to experiment with simple examples; further details can be found in [**15**].

Example 3.9. For fixed values of the eccentricity ϵ, Kepler's equation

$$u - \epsilon \sin u = m \tag{3.13}$$

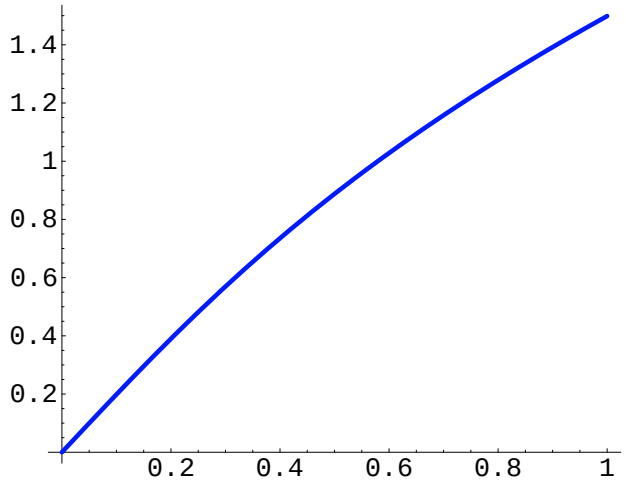


Figure 14. The Solution to the Kepler Equation for Eccentricity $\epsilon = .5$.

can be viewed as a implicit equation defining the eccentric anomaly u as a function of the mean anomaly m. To solve Kepler's equation by Newton's Method, we introduce the iterative function

$$g(u) \;=\; u \;-\; \frac{u - \epsilon \sin u - m}{1 - \epsilon \cos u}\,.$$

Notice that when $|\,\epsilon\,| < 1$, the denominator never vanishes and so the iteration remains well-defined everywhere. Starting with a sufficiently close initial guess $u^{(0)}$, we are assured that the method will quickly converge to the solution.

Fixing the eccentricity ϵ, we can employ the method of *continuation* to determine how the solution $u^\star = h(m)$ depends upon the mean anomaly m. Namely, we start at $m = m_0 = 0$ with the obvious solution $u^\star = h(0) = 0$. Then, to compute the solution at successive closely spaced values $0 < m_1 < m_2 < m_3 < \cdots$, we use the previously computed value as an initial guess $u^{(0)} = h(m_k)$ for the value of the solution at the next mesh point m_{k+1}, and run the Newton scheme until it converges to a sufficiently accurate approximation to the value $u^\star = h(m_{k+1})$. As long as m_{k+1} is reasonably close to m_k, Newton's Method will converge to the solution quite quickly.

The continuation method will quickly produce the values of u at the sample points. Intermediate values can either be determined by an interpolation scheme, e.g., a cubic spline fit of the data, or by running the Newton scheme using the closest known value as an initial condition. A plot for $0 \le m \le 1$ using the value $\epsilon = .5$ appears in Figure 14.

Systems of Equations

Let us now turn our attention to nonlinear systems of equations. We shall only consider the case when there are the same number of equations as unknowns:

$$f_1(u_1, \dots, u_n) = 0, \qquad \dots \qquad f_n(u_1, \dots, u_n) = 0. \tag{3.14}$$

We shall rewrite the system in vector form

$$\mathbf{f}(\mathbf{u}) = \mathbf{0}, \tag{3.15}$$

where $\mathbf{f}:\mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued function of n variables. In practice, we do not necessarily require that $\mathbf{f}$ be defined on all of $\mathbb{R}^n$, although this does simplify the exposition.

We shall only consider solutions that are separated from any others. More formally:

Definition 3.10. A solution $\mathbf{u}^\star$ to a system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ is called *isolated* if there exists $\delta > 0$ such that $\mathbf{f}(\mathbf{u}) \neq \mathbf{0}$ for all $\mathbf{u}$ satisfying $0 < \| \mathbf{u} - \mathbf{u}^\star \| < \delta$.

Example 3.11. Consider the planar equation

$$x^2 + y^2 = (x^2 + y^2)^2.$$

Rewriting the equation in polar coordinates as

$$r = r^2 \qquad \text{or} \qquad r(r-1) = 0,$$

we immediately see that the solutions consist of the origin $x = y = 0$ and all points on the unit circle $r^2 = x^2 + y^2 = 1$. Only the origin is an isolated solution, since every solution lying on the circle has plenty of other points on the circle that lie arbitrarily close to it.

Typically, solutions to a system of n equations in n unknowns are isolated, although this is not always true. For example, if A is a singular $n \times n$ matrix, then the solutions to the homogeneous linear system $A\,\mathbf{u} = \mathbf{0}$ form a nontrivial subspace, and so are not isolated. Nonlinear systems with non-isolated solutions can similarly be viewed as exhibiting some form of degeneracy. In general, the numerical computation of non-isolated solutions, e.g., solving the implicit equations for a curve or surface, is a much more difficult problem, and we will not attempt to discuss these issues in this introductory presentation. (However, our continuation approach to the Kepler equation in Example 3.9 indicates how one might proceed in such situations.)

In the case of a single scalar equation, the simple roots, meaning those for which $f'(u^\star) \neq 0$, are the easiest to compute. In higher dimensions, the role of the derivative of the function is played by the Jacobian matrix (2.28), and this motivates the following definition.

Definition 3.12. A solution $\mathbf{u}^\star$ to a system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ is called *nonsingular* if the associated Jacobian matrix is nonsingular there: $\det \mathbf{f}'(\mathbf{u}^\star) \neq 0$.

Note that the Jacobian matrix is square if and only if the system has the same number of equations as unknowns, which is thus one of the requirements for a solution to be nonsingular in our sense. Moreover, the Inverse Function Theorem from multivariable calculus, [**2**, **12**], implies that a nonsingular solution is necessarily isolated.

Theorem 3.13. *Every nonsingular solution $\mathbf{u}^\star$ to a system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ is isolated.*

Being the multivariate counterparts of simple roots also means that nonsingular solutions of systems are the most amenable to practical computation. Computing non-isolated solutions, as well as isolated solutions with a singular Jacobian matrix, is a considerable challenge, and practical algorithms remain much less well developed. For this reason, we focus exclusively on numerical solution techniques for nonsingular solutions.

Now, let us turn to numerical solution techniques. The first remark is that, unlike the scalar case, proving existence of a solution to a system of equations is often a challenging issue. There is no counterpart to the Intermediate Value Lemma 3.1 for vector-valued functions. It is not hard to find vector-valued functions whose entries take on both positive and negative values, but admit no solutions; a simple example follows. This precludes any simple analog of the Bisection Method for nonlinear systems in more than one unknown.

Example 3.14. Let $\mathbf{f}(x,y) = (\,x, 2\,x\,y - 1\,)^T$. Note that

$$\mathbf{f}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1\\1\end{pmatrix}, \qquad \mathbf{f}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\\-1\end{pmatrix}, \qquad \mathbf{f}\begin{pmatrix}-1\\-1\end{pmatrix} = \begin{pmatrix}-1\\1\end{pmatrix}, \qquad \mathbf{f}\begin{pmatrix}-1\\0\end{pmatrix} = \begin{pmatrix}-1\\-1\end{pmatrix},$$

and hence the entries of $\mathbf{f}(x,y)$ take on all possible combinations of signs, i.e., both entries positive, the first positive and the second negative, etc. On the other hand, there is no solution to the system $\mathbf{f}(x,y) = \mathbf{0}$. This implies that one cannot detect roots of a vector-valued function by looking at sign changes in the entries.

On the other hand, Newton's Method can be straightforwardly adapted to compute nonsingular solutions to systems of equations, and is *the* most widely used method for this purpose. The derivation proceeds in very similar manner to the scalar case. First, we replace the system (3.15) by a fixed point system

$$\mathbf{u} = \mathbf{g}(\mathbf{u}) \tag{3.16}$$

having the same solutions. By direct analogy with (3.7), any (reasonable) fixed point method will take the form

$$\mathbf{g}(\mathbf{u}) = \mathbf{u} - L(\mathbf{u})\,\mathbf{f}(\mathbf{u}), \tag{3.17}$$

where $L(\mathbf{u})$ is an $n \times n$ matrix-valued function. Clearly, if $\mathbf{f}(\mathbf{u}) = \mathbf{0}$ then $\mathbf{g}(\mathbf{u}) = \mathbf{u}$; conversely, if $\mathbf{g}(\mathbf{u}) = \mathbf{u}$, then $L(\mathbf{u})\,\mathbf{f}(\mathbf{u}) = \mathbf{0}$. If we further require that the matrix $L(\mathbf{u})$ be nonsingular, i.e., $\det L(\mathbf{u}) \neq 0$, then every fixed point of the iterator (3.17) will be a solution to the system (3.15) and vice versa.

According to Theorem 2.12, the speed of convergence (if any) of the iterative method

$$\mathbf{u}^{(k+1)} = \mathbf{g}(\mathbf{u}^{(k)}) \tag{3.18}$$

is governed by the matrix norm (or, more precisely, the spectral radius) of the Jacobian matrix $\mathbf{g}'(\mathbf{u}^\star)$ at the fixed point. In particular, if

$$\mathbf{g}'(\mathbf{u}^\star) = \mathrm{O} \tag{3.19}$$

is the zero matrix, then the method converges quadratically fast. Let's figure out how this can be arranged. Computing the derivative using the matrix version of the Leibniz rule for the derivative of a matrix product, we find

$$\mathbf{g}'(\mathbf{u}^\star) = \mathrm{I} - L(\mathbf{u}^\star)\,\mathbf{f}'(\mathbf{u}^\star), \tag{3.20}$$

where I is the $n \times n$ identity matrix. (Fortunately, all the terms that involve derivatives of the entries of $L(\mathbf{u})$ go away since $\mathbf{f}(\mathbf{u}^\star) = \mathbf{0}$ by assumption.) Therefore, the quadratic convergence criterion (3.19) holds if and only if

$$L(\mathbf{u}^\star)\,\mathbf{f}'(\mathbf{u}^\star) = \mathrm{I}\,, \qquad \text{and hence} \qquad L(\mathbf{u}^\star) = \mathbf{f}'(\mathbf{u}^\star)^{-1} \tag{3.21}$$

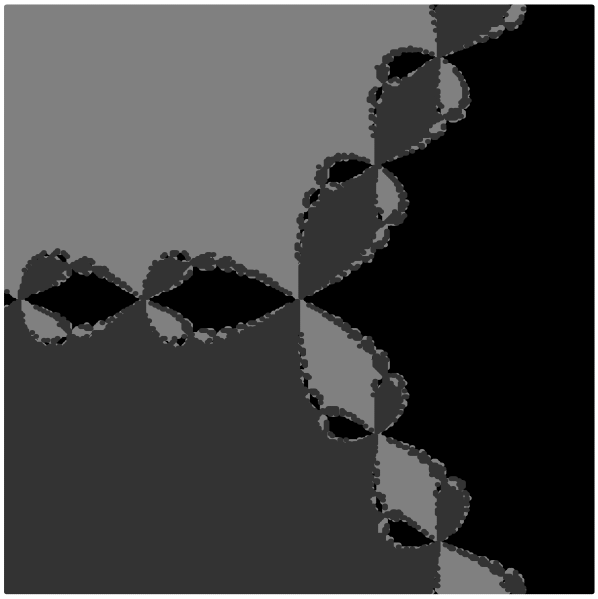


Figure 15. Computing the Cube Roots of Unity by Newton's Method.

should be the inverse of the Jacobian matrix of $\mathbf{f}$ at the solution, which, fortuitously, was already assumed to be nonsingular.

As in the scalar case, we don't know the solution $\mathbf{u}^\star$, but we can arrange that condition (3.21) holds by setting

$$L(\mathbf{u}) = \mathbf{f}'(\mathbf{u})^{-1}$$

everywhere — or at least everywhere that $\mathbf{f}$ has a nonsingular Jacobian matrix. The resulting fixed point system

$$\mathbf{u} = \mathbf{g}(\mathbf{u}) = \mathbf{u} - \mathbf{f}'(\mathbf{u})^{-1}\,\mathbf{f}(\mathbf{u}), \tag{3.22}$$

leads to the quadratically convergent *Newton iteration scheme*

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - \mathbf{f}'(\mathbf{u}^{(k)})^{-1}\,\mathbf{f}(\mathbf{u}^{(k)}). \tag{3.23}$$

All it requires is that we guess an initial value $\mathbf{u}^{(0)}$ that is sufficiently close to the desired solution $\mathbf{u}^\star$. We are then guaranteed that the iterates $\mathbf{u}^{(k)}$ will converge quadratically fast to $\mathbf{u}^\star$.

Theorem 3.15. *Let $\mathbf{u}^\star$ be a nonsingular solution to the system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$. Then, provided $\mathbf{u}^{(0)}$ is sufficiently close to $\mathbf{u}^\star$, the Newton iteration scheme* (3.23) *converges at a quadratic rate to the solution: $\mathbf{u}^{(k)} \to \mathbf{u}^\star$.*

Example 3.16. Consider the pair of simultaneous cubic equations

$$f_1(u,v) = u^3 - 3\,u\,v^2 - 1 = 0, \qquad f_2(u,v) = 3\,u^2\,v - v^3 = 0. \tag{3.24}$$

It is not difficult to prove that there are precisely three solutions:

$$\mathbf{u}_1^\star = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_2^\star = \begin{pmatrix} -.5 \\ .866025\ldots \end{pmatrix}, \qquad \mathbf{u}_3^\star = \begin{pmatrix} -.5 \\ -.866025\ldots \end{pmatrix}. \tag{3.25}$$

The Newton scheme relies on the Jacobian matrix

$$\mathbf{f}'(\mathbf{u}) = \begin{pmatrix} 3\,u^2 - 3\,v^2 & -6\,u\,v \\ 6\,u\,v & 3\,u^2 - 3\,v^2 \end{pmatrix}.$$

Since $\det \mathbf{f}'(\mathbf{u}) = 9(u^2 + v^2)$ is non-zero except at the origin, all three solutions are non-singular, and hence, for a sufficiently close initial value, Newton's Method will converge to the nearby solution. We explicitly compute the inverse Jacobian matrix:

$$\mathbf{f}'(\mathbf{u})^{-1} = \frac{1}{9(u^2 + v^2)} \begin{pmatrix} 3\,u^2 - 3\,v^2 & 6\,u\,v \\ -6\,u\,v & 3\,u^2 - 3\,v^2 \end{pmatrix}.$$

Hence, in this particular example, the Newton iterator (3.22) is

$$\mathbf{g}(\mathbf{u}) = \begin{pmatrix} u \\ v \end{pmatrix} - \frac{1}{9(u^2 + v^2)} \begin{pmatrix} 3\,u^2 - 3\,v^2 & 6\,u\,v \\ -6\,u\,v & 3\,u^2 - 3\,v^2 \end{pmatrix} \begin{pmatrix} u^3 - 3\,u\,v^2 - 1 \\ 3\,u^2\,v - v^3 \end{pmatrix}.$$

A complete diagram of the three basins of attraction, consisting of points whose Newton iterates converge to each of the three roots, has a remarkably complicated, fractal-like structure, as illustrated in Figure 15. In this plot, the x and y coordinates run from -1.5 to 1.5. The points in the black region all converge to $\mathbf{u}_1^\star$; those in the light gray region all converge to $\mathbf{u}_2^\star$; while those in the dark gray region all converge to $\mathbf{u}_3^\star$. The closer one is to the root, the sooner the iterates converge. On the interfaces between the basins of attraction are points for which the Newton iterates fail to converge, but exhibit a random, chaotic behavior. However, round-off errors will cause such iterates to fall into one of the basins, making it extremely difficult to observe such behavior over the long run.

Remark: The alert reader may notice that in this example, we are in fact merely computing the cube roots of unity, i.e., equations (3.24) are the real and imaginary parts of the complex equation $z^3 = 1$ when $z = u + \mathrm{i}\,v$.

Example 3.17. A robot arm consists of two rigid rods that are joined end-to-end to a fixed point in the plane, which we take as the origin $\mathbf{0}$. The arms are free to rotate, and the problem is to configure them so that the robot's hand ends up at the prescribed position $\mathbf{a} = (\,a, b\,)^T$. The first rod has length ℓ and makes an angle α with the horizontal, so its end is at position $\mathbf{v}_1 = (\,\ell\cos\alpha, \ell\sin\alpha\,)^T$. The second rod has length m and makes an angle β with the horizontal, and so is represented by the vector $\mathbf{v}_2 = (\,m\cos\beta, m\sin\beta\,)^T$. The hand at the end of the second arm is at position $\mathbf{v}_1 + \mathbf{v}_2$, and the problem is to find values for the angles α, β so that $\mathbf{v}_1 + \mathbf{v}_2 = \mathbf{a}$; see Figure 16. To this end, we need to solve the system of equations

$$\ell\cos\alpha + m\cos\beta = a, \qquad \ell\sin\alpha + m\sin\beta = b, \tag{3.26}$$

for the angles α, β.

To find the solution, we shall apply Newton's Method. First, we compute the Jacobian matrix of the system with respect to α, β, which is

$$\mathbf{f}'(\alpha, \beta) = \begin{pmatrix} -\ell\sin\alpha & -m\sin\beta \\ \ell\cos\alpha & m\cos\beta \end{pmatrix},$$

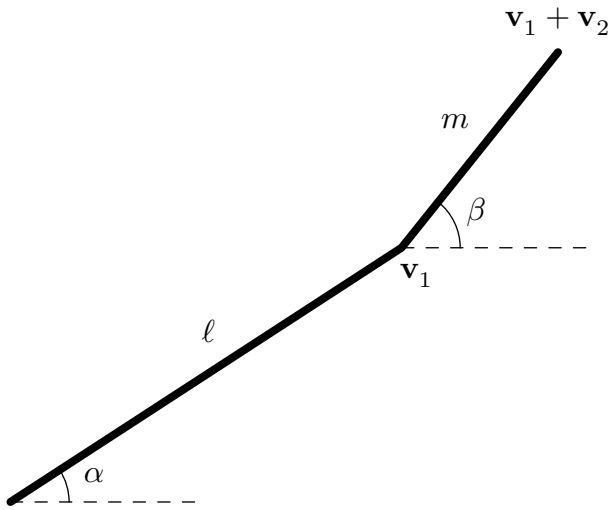


Figure 16. Robot Arm.

with inverse

$$\mathbf{f}'(\alpha, \beta)^{-1} = \frac{1}{\ell m \sin(\beta - \alpha)} \begin{pmatrix} -\ell \sin\alpha & m \sin\beta \\ -\ell \cos\alpha & m \cos\beta \end{pmatrix}.$$

As a result, the Newton iteration equation (3.23) has the explicit form

$$\begin{pmatrix} \alpha^{(k+1)} \\ \beta^{(k+1)} \end{pmatrix} = \begin{pmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{pmatrix} - \\ - \frac{1}{\ell m \sin(\beta^{(k)} - \alpha^{(k)})} \begin{pmatrix} -\ell \cos\alpha^{(k)} & m \sin\beta^{(k)} \\ -\ell \cos\alpha^{(k)} & m \sin\beta^{(k)} \end{pmatrix} \begin{pmatrix} \ell \cos\alpha^{(k)} + m \cos\beta^{(k)} - a \\ \ell \sin\alpha^{(k)} + m \sin\beta^{(k)} - b \end{pmatrix}.$$

when running the iteration, one must be careful to avoid points at which $\alpha^{(k)} - \beta^{(k)} = 0$ or π, i.e., where the robot arm has straightened out.

As an example, let us assume that the rods have lengths $\ell = 2$, $m = 1$, and the desired location of the hand is at $\mathbf{a} = (\,1, 1\,)^T$. We start with an initial guess of $\alpha^{(0)} = 0$, $\beta^{(0)} = \frac{1}{2}\pi$, so the first rod lies along the x–axis and the second is perpendicular. The first few Newton iterates are given in the accompanying table. The first column is the iterate number k; the second and third columns indicate the angles $\alpha^{(k)}$, $\beta^{(k)}$ of the rods. The fourth and fifth give the position $(x^{(k)}, y^{(k)})^T$ of the joint or elbow, while the final two indicate the position $(z^{(k)}, w^{(k)})^T$ of the robot's hand.

k	$\alpha^{(k)}$	$\beta^{(k)}$	$x^{(k)}$	$y^{(k)}$	$z^{(k)}$	$w^{(k)}$
0	.0000	1.5708	2.0000	.0000	2.0000	1.0000
1	.0000	2.5708	2.0000	.0000	1.1585	.5403
2	.3533	2.8642	1.8765	.6920	.9147	.9658
3	.2917	2.7084	1.9155	.5751	1.0079	.9948
4	.2987	2.7176	1.9114	.5886	1.0000	1.0000
5	.2987	2.7176	1.9114	.5886	1.0000	1.0000

Observe that the robot has rapidly converged to one of the two possible configurations. (Can you figure out what the second equilibrium is?) In general, convergence depends on the choice of initial configuration, and the Newton iterates do not always settle down to a fixed point. For instance, if $\| \mathbf{a} \| > \ell + m$, there is no possible solution, since the arms are too short for the hand to reach to desired location; thus, no choice of initial conditions will lead to a convergent scheme and the robot arm flaps around in a chaotic manner.

Now that we have gained a little experience with Newton's Method for systems of equations, some supplementary remarks are in order. As we know, [**14**], except perhaps in very low-dimensional situations, one should not directly invert a matrix, but rather use Gaussian elimination, or, in favorable situations, a linear iterative scheme, e.g., Jacobi, Gauss–Seidel or even SOR. So a better strategy is to leave the Newton system (3.23) in unsolved, implicit form

$$\mathbf{f}'(\mathbf{u}^{(k)})\,\mathbf{v}^{(k)} = -\,\mathbf{f}(\mathbf{u}^{(k)}), \qquad \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{v}^{(k)}. \tag{3.27}$$

Given the iterate $\mathbf{u}^{(k)}$, we compute the Jacobian matrix $\mathbf{f}'(\mathbf{u}^{(k)})$ and the right hand side $-\,\mathbf{f}(\mathbf{u}^{(k)})$, and then use our preferred linear systems solver to find $\mathbf{v}^{(k)}$. Adding $\mathbf{u}^{(k)}$ to the result immediately yields the updated approximation $\mathbf{u}^{(k+1)}$ to the solution.

The main bottleneck in the implementation of the Newton scheme, particularly for large systems, is solving the linear system in (3.27). The coefficient matrix $\mathbf{f}'(\mathbf{u}^{(k)})$ must be recomputed at each step of the iteration, and hence knowing the solution to the k^{th} linear system does not appear to help us solve the subsequent system. Performing a complete Gaussian elimination at every step will tend to slow down the algorithm, particularly in high dimensional situations involving many equations in many unknowns.

One simple dodge for speeding up the computation is to note that, once we start converging, $\mathbf{u}^{(k)}$ will be very close to $\mathbf{u}^{(k-1)}$ and so we will probably not go far wrong by using $\mathbf{f}'(\mathbf{u}^{(k-1)})$ in place of the updated Jacobian matrix $\mathbf{f}'(\mathbf{u}^{(k)})$. Since we have already solved the linear system with coefficient matrix $\mathbf{f}'(\mathbf{u}^{(k-1)})$, we know its $L\,U$ factorization, and hence can use Forward and Back Substitution to quickly solve the modified system

$$\mathbf{f}'(\mathbf{u}^{(k-1)})\,\mathbf{v}^{(k+1)} = -\,\mathbf{f}(\mathbf{u}^{(k)}), \qquad \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{v}^{(k)}. \tag{3.28}$$

If $\mathbf{u}^{(k+1)}$ is still close to $\mathbf{u}^{(k-1)}$, we can continue to use $\mathbf{f}'(\mathbf{u}^{(k-1)})$ as the coefficient matrix when proceeding on to the next iterate $\mathbf{u}^{(k+2)}$. We proceed in this manner until there

has been a notable change in the iterates, at which stage we can revert to solving the correct, unmodified linear system (3.27) by Gaussian Elimination. This strategy may dramatically reduce the total amount of computation required to approximate the solution to a prescribed accuracy. The down side is that this *quasi-Newton scheme* is usually only linearly convergent, and so does not home in on the root as fast as the unmodified implementation. The user needs to balance the trade-off between speed of convergence versus amount of time needed to solve the linear system at each step in the process. See [**16**] for further discussion.

4. Optimization.

We have already noted the importance of quadratic minimization principles for characterizing the equilibrium solutions of linear systems of physical significance. In nonlinear systems, optimization — either maximization or minimization — retains its centrality, and the wealth of practical applications has spawned an entire sub-discipline of applied mathematics. Physical systems naturally seek to minimize the potential energy function, and so determination of the possible equilibrium configurations requires solving a nonlinear minimization principle. Engineering design is guided by a variety of optimization constraints, such as performance, longevity, safety, and cost. Non-quadratic minimization principles also arise in the fitting of data by schemes that go beyond the simple linear least squares approximation method discussed in [**14**; Section 4.3]. Additional applications naturally appear in economics and financial mathematics — one often wishes to minimize expenses or maximize profits, in biological and ecological systems, in pattern recognition and signal processing, in statistics, and so on. In this section, we will describe the basic mathematics underlying simple nonlinear optimization problems along with basic numerical techniques.

The Objective Function

Throughout this section, the real-valued function $F(\mathbf{u}) = F(u_1, \dots, u_n)$ to be optimized — the energy, cost, entropy, performance, etc. — will be called the *objective function*. As such, it depends upon one or more variables $\mathbf{u} = (u_1, u_2, \dots, u_n)^T$ that belong to a prescribed subset $\Omega \subset \mathbb{R}^n$.

Definition 4.1. A point $\mathbf{u}^\star \in \Omega$ is a *global minimum* of the objective function

$$F(\mathbf{u}^\star) \le F(\mathbf{u}) \qquad \text{for all} \qquad \mathbf{u} \in \Omega. \tag{4.1}$$

The minimum is called *strict* if

$$F(\mathbf{u}^\star) < F(\mathbf{u}) \qquad \text{for} \qquad \mathbf{u}^\star \neq \mathbf{u} \in \Omega. \tag{4.2}$$

The point $\mathbf{u}^\star$ is called a (*strict*) *local minimum* if the relevant inequality holds just for points $\mathbf{u} \in \Omega$ nearby $\mathbf{u}^\star$, i.e., satisfying $\| \mathbf{u} - \mathbf{u}^\star \| < \delta$ for some $\delta > 0$. In particular, strict local minima are *isolated*.

The definition of a *maximum* — local or global — is the same, but with the reversed inequality: $F(\mathbf{u}^\star) \ge F(\mathbf{u})$ or, in the strict case, $F(\mathbf{u}^\star) > F(\mathbf{u})$. Alternatively, a maximum of $F(\mathbf{u})$ is the same as a minimum of the negative $-F(\mathbf{u})$. Therefore, every result that

applies to minimization of a function can easily be translated into a result on maximization, which allows us to concentrate exclusively on the minimization problem without any loss of generality. We will use *extremum* as a shorthand term for either a minimum or a maximum.

Remark: In fact, *any* system of equations can be readily converted into a minimization principle. Given a system $\mathbf{f}(\mathbf{u}) = \mathbf{0}$, we introduce the objective function

$$F(\mathbf{u}) = \| \mathbf{f}(\mathbf{u}) \|^2, \tag{4.3}$$

where $\| \cdot \|$ is any convenient norm on $\mathbb{R}^n$. By the basic properties of the norm, the minimum value is $F(\mathbf{u}) = 0$, and this is achieved if and only if $\mathbf{f}(\mathbf{u}) = \mathbf{0}$, i.e., at a solution to the system. More generally, if there is no solution to the system, the minimizer(s) of $F(\mathbf{u})$ play the role of a least squares solution, at least for an inner product-based norm, along with the extensions to more general norms.

In contrast to the rather difficult question of existence of solutions to systems of equations, there is a general theorem that guarantees the existence of minima (and, hence, maxima) for a broad class of optimization problems.

Theorem 4.2. *If $F\colon \Omega \to \mathbb{R}$ is continuous, and $\Omega \subset \mathbb{R}^n$ is a* compact, *meaning closed and bounded, subset, then F has at least one global minimum $\mathbf{u}^\star \in \Omega$.*

Proof: Let $m^\star = \min\{ F(\mathbf{u}) \,|\, \mathbf{u} \in \Omega \}$, which may, *a priori*, be $-\infty$. Choose points $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots \in \Omega$, such that $F(\mathbf{u}^{(k)}) \to m$ as $k \to \infty$. By the basic properties of compact sets, [**18**], there is a convergent subsequence $\mathbf{u}^{(k_i)} \to \mathbf{u}^\star \in \Omega$. By continuity,

$$F(\mathbf{u}^\star) = F\left(\lim_{k_i \to \infty} \mathbf{u}^{(k_i)} \right) = \lim_{k_i \to \infty} F\left(\mathbf{u}^{(k_i)} \right) = m^\star,$$

and hence $\mathbf{u}^\star$ is a minimizer. *Q.E.D.*

Although Theorem 4.2 assures us of the existence of a global minimum of any continuous function on a bounded domain, it does not guarantee uniqueness, nor does it indicate how to go about finding it. Just as with the solution of nonlinear systems of equations, it is quite rare that one can extract explicit formulae for the minima of non-quadratic functions. Our goal, then, is to formulate practical algorithms that can accurately compute the minima of general nonlinear functions.

The most naïve algorithm, but one that is often successful in small scale problems, [**16**], is to select a reasonably dense set of sample points $\mathbf{u}^{(k)}$ in the domain and choose the one that provides the smallest value for $F(\mathbf{u}^{(k)})$. If the points are sufficiently densely distributed and the function is not too wild, this will give a reasonable approximation to the minimum. The algorithm can be speeded up by appealing to more sophisticated means of selecting the sample points.

In the rest of this section, we will discuss optimization strategies that exploit the differential calculus. Let us first review the basic procedure for optimizing functions that you learned in first and second year calculus. As you no doubt remember, there are two different possible types of minima. An *interior minimum* occurs at an interior point of the domain of definition of the function, whereas a *boundary minimum* occurs on its boundary

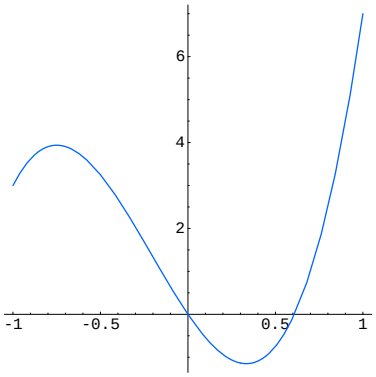


Figure 17. The function $8\,u^3 + 5\,u^2 - 6\,u$.

$\partial\Omega$. Interior local minima are easier to find, and, to keep the presentation simple, we shall focus our efforts on them. Let us begin with a simple scalar example.

Example 4.3. Let us optimize the scalar objective function

$$F(u) = 8\,u^3 + 5\,u^2 - 6\,u$$

on the domain $-1 \le u \le 1$. To locate the minimum, the first step is to look at the *critical points* where the derivative vanishes:

$$F'(u) = 24\,u^2 + 10\,u - 6 = 0, \qquad \text{and hence} \qquad u = \tfrac{1}{3},\ -\tfrac{3}{4}.$$

To ascertain the local nature of the two critical points, we apply the second derivative test. Since $F''(u) = 48\,u + 10$, we have

$$F''\left(\tfrac{1}{3}\right) = 26 > 0, \qquad \text{whereas} \qquad F''\left(-\tfrac{3}{4}\right) = -26 < 0.$$

We conclude that $\frac{1}{3}$ is a local minimum, while $\frac{3}{4}$ is a local maximum.

To find the global minimum and maximum on the interval $[-1, 1]$, we must also take into account the boundary points ± 1. Comparing the function values at the four points,

$$F(-1) = 3, \qquad F\left(\tfrac{1}{3}\right) = -\tfrac{31}{27} \approx -1.148, \qquad F\left(-\tfrac{3}{4}\right) = \tfrac{63}{16} = 3.9375, \qquad F(1) = 7,$$

we see that $\frac{1}{3}$ is the global minimum, whereas 1 is the global maximum — which occurs on the boundary of the interval. This is borne out by the graph of the function, as displayed in Figure 17.

The Gradient

As you first learn in multi-variable calculus, [**2**, **12**], the interior extrema — minima and maxima — of a smooth function $F(\mathbf{u}) = F(u_1, \dots, u_n)$ are necessarily *critical points*, meaning places where the gradient of F vanishes. The standard gradient is the vector field whose entries are its first order partial derivatives:

$$\nabla F(\mathbf{u}) = \left(\frac{\partial F}{\partial u_1}, \ \dots \ , \frac{\partial F}{\partial u_n} \right)^T. \tag{4.4}$$

Let us, in preparation for the more general minimization problems over infinite-dimensional function spaces that arise in the calculus of variations, reformulate the definition of the gradient in a more intrinsic manner. An important but subtle point is that the gradient operator, in fact, relies upon the introduction of an inner product on the underlying vector space. The version (4.4) is, in fact, based upon on the Euclidean dot product on $\mathbb{R}^n$. Altering the inner product will change the formula for the gradient!

Definition 4.4. Let V be an inner product space. The *gradient* of a function $F\colon V \to \mathbb{R}$ at a point $\mathbf{u} \in V$ is the vector $\nabla F(\mathbf{u}) \in V$ that satisfies

$$\langle\, \nabla F(\mathbf{u})\,;\mathbf{v}\,\rangle = \frac{d}{dt}\, F(\mathbf{u} + t\,\mathbf{v})\,\Big|_{t=0} \qquad \text{for all} \qquad \mathbf{v} \in V. \tag{4.5}$$

Remark: The function F does not have to be defined on all of the space V in order for this definition to make sense.

The quantity displayed in the preceding formula is known as the *directional derivative* of F with respect to $\mathbf{v} \in V$, and typically denoted by $\partial F/\partial \mathbf{v}$. Thus, by definition, the directional derivative equals the inner product with the gradient vector. The directional derivative measures the rate of change of F in the direction of the vector $\mathbf{v}$, scaled in proportion to its length.

In the Euclidean case, when $F(\mathbf{u}) = F(u_1, \dots, u_n)$ is a function of n variables, defined for $\mathbf{u} = (\,u_1, u_2, \dots, u_n\,)^T \in \mathbb{R}^n$, we can use the chain rule to compute

$$\begin{aligned} \frac{d}{dt}\, F(\mathbf{u} + t\,\mathbf{v}) &= \frac{d}{dt}\, F(u_1 + t\,v_1, \dots, u_n + t\,v_n) \\ &= \frac{\partial F}{\partial u_1}(\mathbf{u} + t\,\mathbf{v})\, v_1 \;+\; \cdots \;+\; \frac{\partial F}{\partial u_n}(\mathbf{u} + t\,\mathbf{v})\, v_n. \end{aligned} \tag{4.6}$$

Setting $t = 0$, the right hand side of (4.5) reduces to

$$\frac{d}{dt}\, F(\mathbf{u} + t\,\mathbf{v})\,\Big|_{t=0} = \frac{\partial F}{\partial u_1}\,(\mathbf{u})\, v_1 \;+\; \cdots \;+\; \frac{\partial F}{\partial u_n}\,(\mathbf{u})\, v_n = \nabla F(\mathbf{u}) \cdot \mathbf{v}.$$

Therefore, the directional derivative equals the Euclidean dot product between the usual gradient of the function (4.4) and the direction vector $\mathbf{v}$, justifying (4.5) in the Euclidean case.

Remark: In this chapter, we will only deal with the standard Euclidean dot product, which results in the usual gradient formula (4.4). If we introduce an alternative inner product on $\mathbb{R}^n$, then the notion of gradient, as defined in (4.5) will change.

A function $F(\mathbf{u})$ is *continuously differentiable* if and only if its gradient $\nabla F(\mathbf{u})$ is a continuously varying vector-valued function of $\mathbf{u}$. This is equivalent to the requirement that its first order partial derivatives $\partial F/\partial u_i$ are all continuous. As usual, we use $\mathrm{C}^1(\Omega)$ to denote the vector space of all continuously differentiable scalar-valued functions defined on a domain $\Omega \subset \mathbb{R}^n$. From now on, all objective functions are assumed to be continuously differentiable on their domain of definition.

If $\mathbf{u}(t)$ represents a parametrized curve contained within the domain of definition of $F(\mathbf{u})$, then the instantaneous rate of change in the scalar quantity F as we move along the curve is given by

$$\frac{d}{dt}\, F(\mathbf{u}(t)) = \left\langle\, \nabla F(\mathbf{u})\,;\, \frac{d\mathbf{u}}{dt}\,\right\rangle, \tag{4.7}$$

which is the directional derivative of F with respect to the velocity or tangent vector $\mathbf{v} = \dot{\mathbf{u}}$ to the curve. For instance, suppose $F(u_1, u_2)$ represents the elevation of a mountain range at position $\mathbf{u} = (\, u_1, u_2\,)^T$. If we travel through the mountains along the path $\mathbf{u}(t) = (\, u_1(t), u_2(t)\,)^T$, then our instantaneous rate of ascent or descent (4.7) is equal to the dot product of our velocity vector $\dot{\mathbf{u}}(t)$ with the gradient of the elevation function. This observation leads to an important interpretation of the gradient vector.

Theorem 4.5. *The gradient $\nabla F(\mathbf{u})$ of a scalar function $F(\mathbf{u})$ points in the direction of its steepest increase at the point $\mathbf{u}$. The negative gradient, $-\nabla F(\mathbf{u})$, which points in the opposite direction, indicates the direction of steepest decrease.*

Thus, when F represents elevation, ∇F tells us the direction that is steepest uphill, while $-\nabla F$ points directly downhill — the direction water will flow. Similarly, if F represents the temperature of a solid body, then ∇F tells us the direction in which it is heating up the quickest. Heat energy (like water) will flow in the opposite, coldest direction, namely that of the negative gradient vector $-\nabla F$.

But you need to be careful in how you interpret Theorem 4.5. Clearly, the faster you move along a curve, the faster the function $F(\mathbf{u})$ will vary, and one needs to take this into account when comparing the rates of change along different curves. The easiest way to effect the comparison is to assume that the tangent vector $\mathbf{a} = \dot{\mathbf{u}}$ has unit norm, so $\| \mathbf{a} \| = 1$, which means that we are passing through the point $\mathbf{u}(t)$ with unit speed. Once this is done, Theorem 4.5 is an immediate consequence of the Cauchy–Schwarz inequality, cf. [**14**]. Indeed,

$$\left| \frac{\partial F}{\partial \mathbf{a}} \right| = | \, \mathbf{a} \cdot \nabla F \, | \le \| \, \mathbf{a} \, \| \, \| \, \nabla F \, \| = \| \, \nabla F \, \|, \qquad \text{when} \qquad \| \, \mathbf{a} \, \| = 1,$$

with equality if and only if $\mathbf{a}$ points in the same direction as the gradient. Therefore, the maximum rate of change is when $\mathbf{a} = \nabla F / \| \, \nabla F \, \|$ is the unit vector in the gradient direction, while the minimum is achieved when $\mathbf{a} = -\nabla F / \| \, \nabla F \, \|$ points in the opposite direction.

Critical Points

Thus, the only points at which the gradient fails to indicate directions of increase/decrease of the objective function are where it vanishes. Such points play a critical role in the analysis, whence the following definition.

Definition 4.6. A point $\mathbf{u}^\star$ is called a *critical point* of the objective function $F(\mathbf{u})$ if

$$\nabla F(\mathbf{u}^\star) = \mathbf{0}. \tag{4.8}$$

Let us prove that all local minima are indeed critical points. The most important thing about this proof is that it only relies on the intrinsic definition of gradient, and therefore applies to any function on any inner product space. Moreover, even though the gradient will change if we alter the underlying inner product, the requirement that it vanish at a local minimum does not.

Theorem 4.7. *Every local (interior) minimum* $\mathbf{u}^\star$ *of a continuously differentiable function* $F(\mathbf{u})$ *is a critical point*: $\nabla F(\mathbf{u}^\star) = \mathbf{0}$.

Proof: Let $\mathbf{0} \neq \mathbf{v} \in \mathbb{R}^n$ be any vector. Consider the scalar function

$$g(t) = F(\mathbf{u}^\star + t\,\mathbf{v}) = F(u_1^\star + t\,v_1, \dots, u_n^\star + t\,v_n), \tag{4.9}$$

where $t \in \mathbb{R}$ is sufficiently small to ensure that $\mathbf{u}^\star + t\,\mathbf{v}$ remains strictly inside the domain of F. Note that g measures the values of F along a straight line passing through $\mathbf{u}^\star$ in the direction prescribed by $\mathbf{v}$. Since $\mathbf{u}^\star$ is a local minimum,

$$F(\mathbf{u}^\star) \le F(\mathbf{u}^\star + t\,\mathbf{v}), \qquad \text{and hence} \qquad g(0) \le g(t)$$

for all t sufficiently close to zero. In other words, $g(t)$, as a function of the single variable t, has a local minimum at $t = 0$. By the basic calculus result on minima of functions of one variable, the derivative of $g(t)$ must vanish at $t = 0$. Therefore, by the definition (4.5) of gradient,

$$0 = g'(0) = \frac{d}{dt}\, F(\mathbf{u}^\star + t\,\mathbf{v}) \,\Big|_{t=0} = \langle\, \nabla F(\mathbf{u}^\star)\,;\mathbf{v}\,\rangle.$$

We conclude that the gradient vector $\nabla F(\mathbf{u}^\star)$ at the critical point must be orthogonal to *every* vector $\mathbf{v} \in \mathbb{R}^n$, which is only possible if $\nabla F(\mathbf{u}^\star) = \mathbf{0}$. *Q.E.D.*

Thus, provided the objective function is continuously differentiable, every interior minimum, both local and global, is necessarily a critical point. The converse is not true; critical points can be maxima; they can also be saddle points or of some degenerate form. The basic analytical method[†] for determining the (interior) minima of a given function is to first find all its critical points by solving the system of equations (4.8). Each critical point then needs to be examined more closely — as it could be either a minimum, or a maximum, or neither.

Example 4.8. Consider the function

$$F(u, v) = u^4 - 2\,u^2 + v^2,$$

which is defined and continuously differentiable on all of $\mathbb{R}^2$. Since $\nabla F = \big(\,4\,u^3 - 4\,u, 2\,v\,\big)^T$, its critical points are obtained by solving the pair of equations

$$4\,u^3 - 4\,u = 0, \qquad 2\,v = 0.$$

[†] Numerical methods are discussed below.

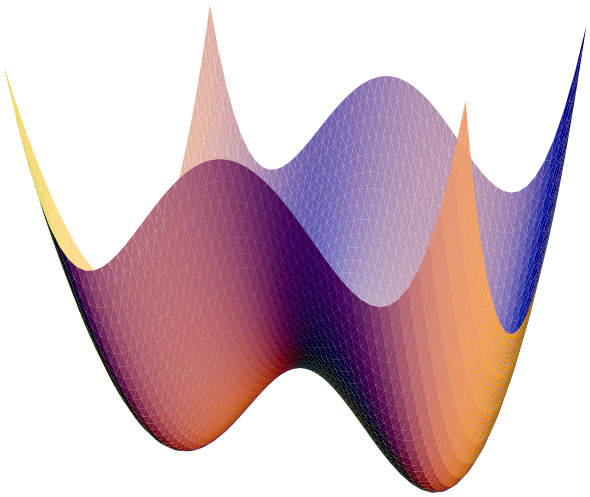


Figure 18. The Function $u^4 - 2u^2 + v^2$.

The solutions to the first equation are $u = 0, \pm 1$, while the second equation requires $v = 0$. Therefore, F has three critical points:

$$\mathbf{u}_1^\star = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_2^\star = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_3^\star = \begin{pmatrix} -1 \\ 0 \end{pmatrix}. \tag{4.10}$$

Inspecting its graph in Figure 18, we suspect that the first critical point $\mathbf{u}_1^\star$ is a saddle point, whereas the other two appear to be local minima, having the same value $F(\mathbf{u}_2^\star) = F(\mathbf{u}_3^\star) = -1$. This will be confirmed once we learn how to rigorously distinguish critical points.

The student should also pay attention to the distinction between local minima and global minima. In the absence of theoretical justification, the only practical way to determine whether or not a minimum is global is to find all the different local minima, including those on the boundary, and see which one gives the smallest value. If the domain is unbounded, one must also worry about the asymptotic behavior of the objective function for large $\mathbf{u}$.

The Second Derivative Test

The status of critical point — minimum, maximum, or neither — can often be resolved by analyzing the second derivative of the objective function at the critical point. Let us first review the one variable second derivative test you learned in first year calculus.

Proposition 4.9. *Let $g(t) \in \mathrm{C}^2$ be a scalar function, and suppose that $t^\star$ a critical point: $g'(t^\star) = 0$. If $t^\star$ is a local minimum, then $g''(t^\star) \geq 0$. Conversely, if $g''(t^\star) > 0$, then $t^\star$ is a strict local minimum. Similarly, $g''(t^\star) \leq 0$ is required at a local maximum, while $g''(t^\star) < 0$ implies that $t^\star$ is a strict local maximum.*

The proof of this result relies on the fact that we can approximate the function by its quadratic Taylor polynomial near the critical point:

$$g(t) \approx g(t^\star) + \tfrac{1}{2}(t - t^\star)^2 g''(t^\star),$$

since $g'(t^\star) = 0$, and so the linear terms in the Taylor polynomial vanish. If $g''(t^\star) \neq 0$, then the quadratic Taylor polynomial has a minimum or maximum at $t^\star$ according to the sign of the second derivative, and this provides the key to the proof. In the borderline case, when $g''(t^\star) = 0$, the second derivative test is inconclusive, and the point could be either maximum or minimum or neither. One must analyze the higher order terms in the Taylor expansion to resolve the status of the critical point.

In multi-variate calculus, the "second derivative" of a function $F(\mathbf{u}) = F(u_1, \dots, u_n)$ is represented by the $n \times n$ *Hessian*[†] *matrix*, whose entries are its second order partial derivatives:

$$\nabla^2 F(\mathbf{u}) = \begin{pmatrix} \dfrac{\partial^2 F}{\partial u_1^2} & \dfrac{\partial^2 F}{\partial u_1\, \partial u_2} & \cdots & \dfrac{\partial^2 F}{\partial u_1\, \partial u_n} \\ \dfrac{\partial^2 F}{\partial u_2\, \partial u_1} & \dfrac{\partial^2 F}{\partial u_2^2} & \cdots & \dfrac{\partial^2 F}{\partial u_2\, \partial u_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 F}{\partial u_n\, \partial u_1} & \dfrac{\partial^2 F}{\partial u_n\, \partial u_2} & \cdots & \dfrac{\partial^2 F}{\partial u_n^2} \end{pmatrix}, \tag{4.11}$$

We will always assume that $F(\mathbf{u}) \in \mathrm{C}^2$ has continuous second order partial derivatives. In this case, its mixed partial derivatives are equal: $\partial^2 F/\partial u_i\, \partial u_j = \partial^2 F/\partial u_j\, \partial u_i$, cf. [**2**, **12**]. As a result, the Hessian is a symmetric matrix: $\nabla^2 F(\mathbf{u}) = \nabla^2 F(\mathbf{u})^T$.

The second derivative test for a local minimum of scalar function relies on the positivity of its second derivative. For a function of several variables, the corresponding condition is that the Hessian matrix be positive definite, cf. [**14**]. More specifically:

Theorem 4.10. *Let $F(\mathbf{u}) = F(u_1, \dots, u_n) \in \mathrm{C}^2(\Omega)$ be a real-valued, twice continuously differentiable function defined on an open domain $\Omega \subset \mathbb{R}^n$. If $\mathbf{u}^\star \in \Omega$ is a (local, interior) minimum for F, then it is necessarily a critical point, so $\nabla F(\mathbf{u}^\star) = \mathbf{0}$. Moreover, the Hessian matrix* (4.11) *must be positive semi-definite at the minimum, so $\nabla^2 F(\mathbf{u}^\star) \geq 0$. Conversely, if $\mathbf{u}^\star$ is a critical point with positive definite Hessian matrix $\nabla^2 F(\mathbf{u}^\star) > 0$, then $\mathbf{u}^\star$ is a strict local minimum of F.*

A maximum requires a negative semi-definite Hessian matrix. If, moreover, the Hessian at the critical point is negative definite, then the critical point is a strict local maximum. If the Hessian matrix is indefinite, then the critical point is a saddle point — neither minimum nor maximum. In the borderline case, when the Hessian is only positive or negative semi-definite at the critical point, the second derivative test is inconclusive. Resolving

[†] Named after the early eighteenth century German mathematician Ludwig Otto Hesse.

the nature of the critical point requires more detailed knowledge of the objective function, e.g., its higher order derivatives.

We defer the proof of Theorem 4.10 until the end of this section.

Example 4.11. As a first, elementary example, consider the quadratic function

$$F(u,v) = u^2 - 2\,u\,v + 3\,v^2.$$

To minimize F, we begin by computing its gradient

$$\nabla F(u,v) = \begin{pmatrix} 2\,u - 2\,v \\ -2\,u + 6\,v \end{pmatrix}.$$

Solving the pair of equations $\nabla F = \mathbf{0}$, namely

$$2\,u - 2\,v = 0, \qquad\qquad -2\,u + 6\,v = 0,$$

we see that the only critical point is the origin $u = v = 0$. To test whether the origin is a maximum or minimum, we further compute the Hessian matrix

$$H = \nabla^2 F(u,v) = \begin{pmatrix} F_{uu} & F_{uv} \\ F_{uv} & F_{vv} \end{pmatrix} = \begin{pmatrix} 2 & -2 \\ -2 & 6 \end{pmatrix}.$$

Using the methods of [**14**; Section 3.5], we easily prove that the Hessian matrix is positive definite. Therefore, by Theorem 4.10, $\mathbf{u}^\star = \mathbf{0}$ is a strict local minimum of F.

Indeed, we recognize $F(u,v)$ to be, in fact, a homogeneous positive definite quadratic form, which can be written in the form

$$F(u,v) = \mathbf{u}^T K\,\mathbf{u}, \qquad \text{where} \qquad K = \begin{pmatrix} 1 & -1 \\ -1 & 3 \end{pmatrix} = \tfrac{1}{2}\,H, \qquad \mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix}.$$

Positive definiteness of the coefficient matrix K implies that $F(u,v) > 0$ for all $\mathbf{u} = (\,u, v\,)^T \neq \mathbf{0}$, and hence $\mathbf{0}$ is, in fact, a global minimum.

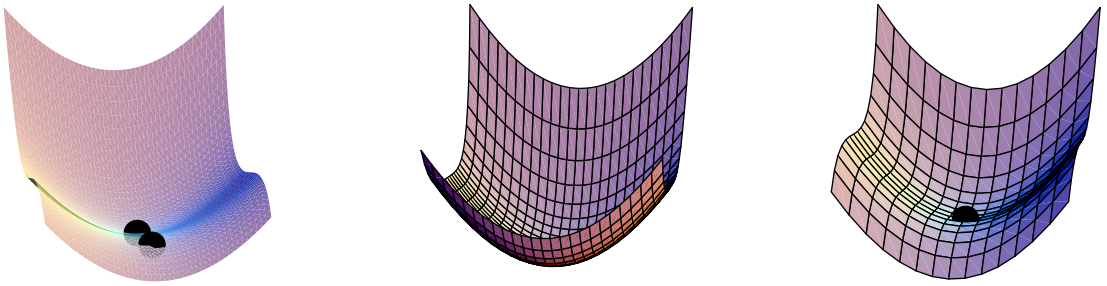
In general, any quadratic function $Q(\mathbf{u}) = Q(u_1, \ldots, u_n)$ can be written in the form

$$Q(\mathbf{u}) = \mathbf{u}^T K\,\mathbf{u} - 2\,\mathbf{b}^T\mathbf{u} + c = \sum_{i,j=1}^{m} k_{ij}\,u_i\,u_j - 2\sum_{i=1}^{n} b_i\,u_i + c, \tag{4.12}$$

where $K = K^T$ is a symmetric $n \times n$ matrix, $\mathbf{b} \in \mathbb{R}^n$ is a fixed vector, and $c \in \mathbb{R}$ is a scalar. A straightforward computation produces the formula for its gradient and Hessian matrix:

$$\nabla Q(\mathbf{u}) = 2\,K\,\mathbf{u} - 2\,\mathbf{b}, \qquad\qquad \nabla^2 Q(\mathbf{u}) = 2\,K. \tag{4.13}$$

As a result, the critical points of the quadratic function are the solutions to the linear system $K\,\mathbf{u} = \mathbf{b}$. If K is nonsingular, there is a unique critical point $\mathbf{u}^\star$, which is a strict local minimum if and only if $K > 0$ is positive definite. In fact, [**14**; Theorem 4.1] tells us that, in the positive definite case, $\mathbf{u}^\star$ is a strict *global* minimum for $Q(\mathbf{u})$. Thus, the algebraic approach of [**14**; Chapter 4] provides additional, global information that cannot be gleaned directly from the local, multivariable calculus Theorem 4.10. But algebra is only able to handle quadratic minimization problems with ease. The analytical classification of minima and maxima of more complicated objective functions necessarily relies the gradient and Hessian criteria of Theorem 4.10.



$u^2 + v^2 - v^3$ $\qquad$ $u^2 + v^4$ $\qquad$ $u^2 + v^3$

Figure 19. Critical Points.

Example 4.12. The function

$$F(u,v) = u^2 + v^2 - v^3 \qquad \text{has gradient} \qquad \nabla F(u,v) = \begin{pmatrix} 2\,u \\ 2\,v - 3\,v^2 \end{pmatrix}.$$

The critical point equation $\nabla F = \mathbf{0}$ has two solutions: $\mathbf{u}_1^\star = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\mathbf{u}_2^\star = \begin{pmatrix} 0 \\ \frac{2}{3} \end{pmatrix}$. The Hessian matrix of the objective function is

$$\nabla^2 F(u,v) = \begin{pmatrix} 2 & 0 \\ 0 & 2 - 6\,v \end{pmatrix}.$$

At the first critical point, the Hessian $\nabla^2 F(0,0) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ is positive definite. Therefore, the origin is a strict local minimum. On the other hand, $\nabla^2 F\left(0, \frac{2}{3}\right) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$ is indefinite, and hence $\mathbf{u}_2^\star = \begin{pmatrix} 0 \\ \frac{2}{3} \end{pmatrix}$ a saddle point. The function is graphed in Figure 19, with the critical points indicated by the small solid balls. The origin is, in fact, only a local minimum, since $F(0,0) = 0$, whereas $F(0,v) < 0$ for all $v > 1$. Thus, this particular function has no global minimum or maximum on $\mathbb{R}^2$.

Next, consider the function

$$F(u,v) = u^2 + v^4, \qquad \text{with gradient} \qquad \nabla F(u,v) = \begin{pmatrix} 2\,u \\ 4\,v^3 \end{pmatrix}.$$

The only critical point is the origin $u = v = 0$. The origin is a strict global minimum because $F(u,v) > 0 = F(0,0)$ for all $(u,v) \neq (\,0,0\,)^T$. However, its Hessian matrix

$$\nabla^2 F(u,v) = \begin{pmatrix} 2 & 0 \\ 0 & 12\,v^2 \end{pmatrix}$$

is only positive semi-definite at the origin, $\nabla^2 F(0,0) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$.

On the other hand, the origin $u = v = 0$ is also the only critical point for the function

$$F(u,v) = u^2 + v^3 \qquad \text{with} \qquad \nabla F(u,v) = \begin{pmatrix} 2\,u \\ 3\,v^2 \end{pmatrix}.$$

The Hessian matrix is

$$\nabla^2 F(u,v) = \begin{pmatrix} 2 & 0 \\ 0 & 6\,v \end{pmatrix}, \qquad \text{and so} \qquad \nabla^2 F(0,0) = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}$$

is the same positive semi-definite matrix at the critical point. However, in this case $(0,0)$ is not a local minimum; indeed

$$F(0,v) < 0 = F(0,0) \qquad \text{whenever} \qquad v < 0,$$

and so there exist points arbitrarily close to the origin where F takes on smaller values. As illustrated in Figure 19, the origin is, in fact, a degenerate saddle point.

Finally, the function

$$F(u,v) = u^2 - 2\,u\,v + v^2 \qquad \text{has gradient} \qquad \nabla F(u,v) = \begin{pmatrix} 2\,u - 2\,v \\ -2\,u + 2\,v \end{pmatrix},$$

and so every point $u = v$ is a critical point. The Hessian matrix

$$\nabla^2 F(u,v) = \begin{pmatrix} F_{uu} & F_{uv} \\ F_{uv} & F_{vv} \end{pmatrix} = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$$

is positive semi-definite everywhere. Since $F(u,u) = 0$, while $F(u,v) = (u-v)^2 > 0$ when $u \neq v$, each of these critical points is a non-isolated (and hence non-strict) local minimum. Thus, comparing the three preceding examples, we see that a semi-definite Hessian is unable to distinguish between different types of degenerate critical points.

Finally, the reader should always keep in mind that first and second derivative tests only determine the local behavior of the function near the critical point. They cannot be used to determine whether or not we are at a global minimum. This requires some additional analysis, and, often, a fair amount of ingenuity.

Proof of Theorem 4.10: We return to the proof of Theorem 4.7. Given a local minimum $\mathbf{u}^\star$, the scalar function $g(t) = F(\mathbf{u}^\star + t\,\mathbf{v})$ in (4.9) has a local minimum at $t = 0$. As noted above, basic calculus tells us that its derivatives at $t = 0$ must satisfy

$$g'(0) = 0, \qquad g''(0) \geq 0. \tag{4.14}$$

The first condition leads to the critical point equation $\nabla F(\mathbf{u}^\star) = \mathbf{0}$. A straightforward chain rule calculation produces the formula

$$g''(0) = \sum_{i,j=1}^{n} \frac{\partial^2 F}{\partial u_i\, \partial u_j}(\mathbf{u}^\star)\, v_i\, v_j = \mathbf{v}^T\, \nabla^2 F(\mathbf{u}^\star)\, \mathbf{v}.$$

As a result, the second condition in (4.14) requires that

$$\mathbf{v}^T\, \nabla^2 F(\mathbf{u}^\star)\, \mathbf{v} \geq 0.$$

Since this condition is required for every direction $\mathbf{v} \in \mathbb{R}^n$, the Hessian matrix $\nabla^2 F(\mathbf{u}^\star) \geq 0$ satisfies the criterion for positive semi-definiteness, proving the first part of the theorem.

The proof of the converse relies[†] on the second order Taylor expansion of the function:

$$\begin{aligned} F(\mathbf{u}) &= F(\mathbf{u}^\star) + \nabla F(\mathbf{u}^\star)\cdot \mathbf{v} + \tfrac{1}{2}\,\mathbf{v}^T \nabla^2 F(\mathbf{u}^\star)\,\mathbf{v} + S(\mathbf{v},\mathbf{u}^\star) \\ &= F(\mathbf{u}^\star) + \tfrac{1}{2}\,\mathbf{v}^T \nabla^2 F(\mathbf{u}^\star)\,\mathbf{v} + S(\mathbf{v},\mathbf{u}^\star), \end{aligned} \qquad \text{where} \qquad \mathbf{v} = \mathbf{u} - \mathbf{u}^\star, \tag{4.15}$$

at the critical point, whence $\nabla F(\mathbf{u}^\star) = \mathbf{0}$. The remainder term in the Taylor formula goes to 0 as $\mathbf{u} \to \mathbf{u}^\star$ at a rate faster than quadratic:

$$\frac{S(\mathbf{v},\mathbf{u}^\star)}{\| \mathbf{v} \|^2} \;\longrightarrow\; 0 \qquad \text{as} \qquad \mathbf{v} \;\longrightarrow\; \mathbf{0}. \tag{4.16}$$

Assuming $\nabla^2 F(\mathbf{u}^\star)$, there is a constant $C > 0$ such that

$$\mathbf{v}^T \nabla^2 F(\mathbf{u}^\star)\,\mathbf{v} \geq C \,\| \mathbf{v} \|^2 \qquad \text{for all} \qquad \mathbf{v} \in \mathbb{R}^n.$$

This is a consequence of Theorem 3.7 of [**14**] on the equivalence of norms, coupled with the fact that every positive definite matrix defines a norm. By (4.16), we can find $\delta > 0$ such that

$$|\, S(\mathbf{v},\mathbf{u}^\star)\,| < \tfrac{1}{2} C \,\| \mathbf{v} \|^2 \qquad \text{whenever} \qquad 0 < \| \mathbf{v} \| = \| \mathbf{u} - \mathbf{u}^\star \| < \delta.$$

But then the Taylor formula (4.15) implies that, for all $\mathbf{u}$ satisfying the preceding inequality,

$$0 < \tfrac{1}{2}\,\mathbf{v}^T \nabla^2 F(\mathbf{u}^\star)\,\mathbf{v} + S(\mathbf{v},\mathbf{u}^\star) = F(\mathbf{u}) - F(\mathbf{u}^\star),$$

which implies $\mathbf{u}^\star$ is a strict local minimum of $F(\mathbf{u})$. *Q.E.D.*

Constrained Optimization and Lagrange Multipliers

In many applications, the function to be minimized is subject to constraints. For instance, finding boundary minima requires constraining the minima to the boundary of the domain. Another example would be to find the minimum temperature on the surface of the earth. Assuming the earth is a perfect sphere of radius R, the temperature function $T(u,v,w)$ is then to be minimized subject to the constraint $u^2 + v^2 + w^2 = R^2$.

Let us focus on finding the minimum value of an objective function $F(\mathbf{u}) = F(u,v,w)$ when its arguments (u,v,w) are constrained to lie on a regular surface $S \subset \mathbb{R}^3$. Suppose $\mathbf{u}^\star = (\,u^\star, v^\star, w^\star\,)^T \in S$ is a (local) minimum for the constrained objective function. Let $\mathbf{u}(t) = (\,u(t), v(t), w(t)\,)^T \subset S$ be any curve contained within the surface that passes through the minimum, with $\mathbf{u}(0) = \mathbf{u}^\star$. Then the scalar function $g(t) = F(\mathbf{u}(t))$ must have a local minimum at $t = 0$, and hence, in view of (4.7),

$$0 = g'(0) = \left.\frac{d}{dt}\, F(\mathbf{u}(t))\,\right|_{t=0} = \nabla F(\mathbf{u}(0))\cdot \dot{\mathbf{u}}(0) = \nabla F(\mathbf{u}^\star)\cdot \dot{\mathbf{u}}(0)\,. \tag{4.17}$$

[†] Actually, it is not hard to prove the first part using the first order Taylor expansion without resorting to the scalar function g. On the other hand, when we look at infinite-dimensional minimization problems arising in the calculus of variations, we will no longer have the luxury of appealing to the finite-dimensional Taylor expansion, whereas the previous argument continues to apply in general contexts.

Thus, the gradient of the objective function at the surface minimum must be orthogonal to the tangent vector to the curve. Since the curve was constrained to lies entirely in S, its tangent vector $\dot{\mathbf{u}}(0)$ is tangent to the surface at the point $\mathbf{u}^\star$. Since every tangent vector to the surface is tangent to some curve contained in the surface, $\nabla F(\mathbf{u}^\star)$ must be orthogonal to every tangent vector, and hence point in the normal direction to the surface. Thus, a *constrained critical point* $\mathbf{u}^\star \in S$ of a function on a surface is defined so that

$$\nabla F(\mathbf{u}^\star) = \lambda\, \mathbf{n}, \tag{4.18}$$

where $\mathbf{n}$ denotes the normal to the surface at the point $\mathbf{u}^\star$. The scalar factor λ is known as the *Lagrange multiplier* in honor or Lagrange, one of the pioneers of constrained optimization. The value of the Lagrange multiplier is not fixed a priori., but must be determined by solving the critical point system (4.18). The same reasoning applies to local maxima, which are also constrained critical points. The nature of a constrained critical point — local minimum, local maximum, local saddle point, etc. — is fixed by a constrained second derivative test.

Example 4.13. Our problem is to find the minimum value of the objective function $F(u,v,w) = u^2 - 2\,w^3$ when u, v, w are restricted to the unit sphere $S = (u^2 + v^2 + w^2 = 1)$. The radial vector $\mathbf{n} = (\,u, v, w\,)^T$ is normal to the sphere, and so the critical point condition (4.18) is

$$\nabla F = \begin{pmatrix} 2\,u \\ 0 \\ -6\,w^2 \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \\ w \end{pmatrix}.$$

Thus, we must solve the system of equations

$$2\,u = \lambda\, u, \qquad 0 = \lambda\, v, \qquad -6\,w^2 = \lambda\, w, \qquad \text{subject to} \qquad u^2 + v^2 + w^2 = 1,$$

for the unknowns u, v, w and λ. This needs to be done carefully to avoid missing any cases. First, if $u \neq 0$, then $\lambda = 2$, $v = 0$, and either $w = 0$ whence $u = \pm 1$, or $w = -\frac{1}{3}$ and so $u = \pm\sqrt{1 - w^2} = \pm\frac{2\sqrt{2}}{3}$. On the other hand, if $u = 0$, then either $\lambda = 0$, $w = 0$ and so $v = \pm 1$, or $v = 0$, $w = \pm 1$, and $\lambda = \mp 6$. Collecting these together, we discover that there are a total of 8 critical points on the unit sphere:

$$\mathbf{u}_1^\star = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_2^\star = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_3^\star = \begin{pmatrix} \frac{2\sqrt{2}}{3} \\ 0 \\ -\frac{1}{3} \end{pmatrix}, \qquad \mathbf{u}_4^\star = \begin{pmatrix} -\frac{2\sqrt{2}}{3} \\ 0 \\ -\frac{1}{3} \end{pmatrix},$$

$$\mathbf{u}_5^\star = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_6^\star = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}, \qquad \mathbf{u}_7^\star = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad \mathbf{u}_8^\star = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

Since the unit sphere is closed and bounded, we are assured that F has a global maximum and a global minimum when restricted to S, which are both to be found among our candidate critical points. Thus, we merely compute the value of the objective function at

each critical point,

$$\begin{aligned} &F(\mathbf{u}_1^\star) = 1, \qquad F(\mathbf{u}_2^\star) = 1, \qquad F(\mathbf{u}_3^\star) = \tfrac{22}{27}\,, \qquad F(\mathbf{u}_4^\star) = \tfrac{26}{27}\,, \\ &F(\mathbf{u}_5^\star) = 0, \qquad F(\mathbf{u}_6^\star) = 0, \qquad F(\mathbf{u}_7^\star) = -2, \qquad F(\mathbf{u}_8^\star) = 2. \end{aligned}$$

Therefore, $\mathbf{u}_7^\star$ must be the global minimum and $\mathbf{u}_8^\star$ the global maximum of the objective function restricted to the unit sphere. The status of the other six critical points — constrained local maximum, minimum, or neither — is less evident, and a full classification requires the second derivative test for constrained extrema.

If the surface is given as the level set of a function

$$G(u, v, w) = c, \tag{4.19}$$

then at any point $\mathbf{u}^\star \in S$, the gradient vector $\nabla G(\mathbf{u}^\star)$ points in the normal direction to the surface, and hence, *provided* $\mathbf{n} = \nabla G(\mathbf{u}^\star) \neq \mathbf{0}$, the surface critical point condition can be rewritten as

$$\nabla F(\mathbf{u}^\star) = \lambda\, \nabla G(\mathbf{u}^\star), \tag{4.20}$$

or, in full detail, the critical point $(\, u^\star, v^\star, w^\star \,)^T$ must satisfy

$$\begin{aligned} \frac{\partial F}{\partial u}\,(u, v, w) &= \lambda\, \frac{\partial G}{\partial u}\,(u, v, w), \\ \frac{\partial F}{\partial v}\,(u, v, w) &= \lambda\, \frac{\partial G}{\partial v}\,(u, v, w), \\ \frac{\partial F}{\partial w}\,(u, v, w) &= \lambda\, \frac{\partial G}{\partial w}\,(u, v, w). \end{aligned} \tag{4.21}$$

Thus, to find the constrained critical points, one needs to solve the combined system (4.19, 21) of 4 equations for the four unknowns u, v, w and the Lagrange multiplier λ.

Formally, one can reformulate the problem as an unconstrained optimization problem by introducing the *augmented objective function*

$$E(u, v, w, \lambda) = F(u, v, w) - \lambda\big(\, G(u, v, w) - c\,\big). \tag{4.22}$$

The critical points of the augmented function are where its gradient, with respect to all four arguments, vanishes. Setting the partial derivatives with respect to u, v, w to 0 reproduces the system (4.21), while it partial derivative with respect to λ reproduces the constraint (4.19).

If $F(\mathbf{u})$ is defined on a closed subdomain $\Omega \subset \mathbb{R}^n$, then its minima may also occur at boundary points $\mathbf{u} \in \partial\Omega$. When the boundary is smooth, there is an analogous critical point condition for local boundary extrema.

Theorem 4.14. *Let $\Omega \subset \mathbb{R}^n$ be a domain with smooth boundary $\partial\Omega$. Suppose $F(\mathbf{u})$ is co;ntinuously differentiable at all points in $\overline{\Omega} = \Omega \cup \partial\Omega$. If the boundary point $\mathbf{u}_0 \in \partial\Omega$ is a (local) minimum for F when restricted to the closed domain $\overline{\Omega}$, then the gradient vector $\nabla F(\mathbf{u}_0)$ is either $\mathbf{0}$ or points inside the domain in the normal direction to $\partial\Omega$.*

Proof: Let $\mathbf{u}(t) \subset \partial\Omega$ be any curve that is entirely contained in the boundary, with $\mathbf{u}(0) = \mathbf{u}_0$. Then the scalar function $g(t) = F(\mathbf{u}(t))$ must have a local minimum at $t = 0$, and hence, in view of (4.7),

$$0 = g'(0) = \frac{d}{dt} F(\mathbf{u}(t)) \Big|_{t=0} = \big\langle \nabla F(\mathbf{u}(0)) \,;\, \dot{\mathbf{u}}(0) \big\rangle \,.$$

Since the curve lies entirely in $\partial\Omega$, its tangent vector $\dot{\mathbf{u}}(0)$ is tangent to the boundary at the point $\mathbf{u}_0$; moreover, we can realize any such tangent vector by a suitable choice of curve. We conclude that $\nabla F(\mathbf{u}_0)$ is orthogonal to every tangent vector to $\partial\Omega$ at the point $\mathbf{u}_0$, and hence must point in the normal direction to the boundary. Moreover, if non-zero, it cannot point outside Ω since then $-\nabla F(\mathbf{u}_0)$, which is the direction of decrease in the objective function, would point inside the domain, which would preclude $\mathbf{u}_0$ from being a local minimum. *Q.E.D.*

The same ideas can be applied to optimization problems involving functions of several variables subject to one or more constraints. Suppose the objective function $F(\mathbf{u}) = F(u_1, \dots, u_n)$ is subject to the constraints

$$G_1(\mathbf{u}) = c_1, \qquad \cdots \qquad G_k(\mathbf{u}) = c_k. \tag{4.23}$$

A point $\mathbf{u}$ satisfying the constraints is termed *regular* if the corresponding gradient vectors $\nabla G_1(\mathbf{u}), \dots, \nabla G_k(\mathbf{u})$ are linearly independent. (Irregular points are more tricky, and must be handled separately.) A regular constrained critical point necessarily satisfies the vector equation

$$\nabla F(\mathbf{u}) = \lambda_1 \nabla G_1(\mathbf{u}) + \, \cdots \, + \lambda_1 \nabla G_1(\mathbf{u}), \tag{4.24}$$

where the unspecified scalars $\lambda_1, \dots, \lambda_k$ are called the Lagrange multipliers for the constrained optimization problem. The critical points are thus found by solving the combined system (4.23–24) for the $n + k$ variables $u_1, \dots, u_n$ and $\lambda_1, \dots, \lambda_k$. As in (4.22) we can reformulate this as an unconstrained optimization problem for the *augmented objective function*

$$E(\mathbf{u}, \boldsymbol{\lambda}) = F(\mathbf{u}) - \sum_{i=1}^{k} \lambda_i \big(G_i(\mathbf{u}) - c_i \big). \tag{4.25}$$

The gradient with respect to $\mathbf{u}$ reproduces the critical point system (4.24), while its gradient with respect to to $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_k)$ recovers the constraints (4.23).

Theorem 4.15. *Every regular constrained local minimum and maximum is a constrained critical point.*

Example 4.16. The problem is to find the point or points on the intersection of the elliptical cylinders

$$u^2 + 4\,v^2 = 1, \qquad u^2 + 9\,w^2 = 4, \tag{4.26}$$

that is the closest to the origin. Thus, we seek to minimize the squared[†] distance function

$$F(u,v,w) = u^2 + v^2 + w^2$$

subject to the constraints

$$G(u,v,w) = u^2 + 4\,v^2 = 1, \qquad H(u,v,w) = u^2 + 9\,w^2 = 4,$$

The augmented objective function (4.22) is

$$E(u,v,w,\lambda,\mu) = u^2 + v^2 + w^2 - \lambda\big(\,u^2 + 4\,v^2 - 1\,\big) + \mu\big(\,u^2 + 9\,w^2 - 4\,\big).$$

To find its critical points, we set all its partial derivatives to zero:

$$\frac{\partial E}{\partial u} = 2\,u + 2\,\lambda\,u + 2\,\mu\,u = 0, \qquad \frac{\partial E}{\partial v} = 2\,v + 8\,\lambda\,v = 0, \qquad \frac{\partial E}{\partial w} = 2\,w + 18\,\lambda\,w = 0,$$

while the partial derivatives with respect to the Lagrange multipliers λ, μ reproduce the two constraints (4.26). Thus,

$$\text{either } u = 0 \text{ or } \lambda + \mu = -1, \quad \text{either } v = 0 \text{ or } \lambda = -\tfrac{1}{4}\,, \quad \text{and} \quad \text{either } w = 0 \text{ or } \mu = -\tfrac{1}{18}\,.$$

Thus, at least one of u, v, w must be zero. If $u = 0$, then $v = \pm\frac{1}{2}$, $w = \pm\frac{2}{3}$; if $v = 0$, then $u = \pm 1$, $w = \pm\frac{1}{\sqrt{3}}$; while there are no real solutions to the constraints when $w = 0$. The first four critical points, $\big(\,0, \pm\frac{1}{2}, \pm\frac{2}{3}\,\big)^T$, all lie a distance $\frac{5}{6} \approx .8333$ from the origin, while the second four, $\big(\,\pm 1, 0, \pm\frac{2}{3}\,\big)^T$, are further away, at distance $\frac{\sqrt{2}}{3} \approx 1.1547$. Thus, the closest points on intersection of the cylinders are the first four, while the furthest points from the origin are the last four. (The latter comment relies on the fact that the intersection is a bounded subset of $\mathbb{R}^3$.)

Remark: A second derivative test for constrained minima and maxima can be found in [**12**].

Numerical Minimization of Scalar Functions

In practical optimization, one typically bypasses the preliminary characterization of minima as critical points, and instead implements a direct iterative procedure that constructs a sequence of successively better approximations to the desired minimum. As the computation progresses, the approximations are adjusted so that the objective function is made smaller and smaller, which, we hope, will ensure that we are converging to some form of minimum.

As always, to understand the issues involved, it is essential to consider the simplest scalar situation. Thus, we are given the problem of minimizing a scalar function $F(u)$ on a bounded interval $a \le u \le b$. The minimum value can either be at an endpoint or an interior minimum. Let us first state a result that plays a similar role to the Intermediate Value Lemma 3.1 that formed the basis of the Bisection Method for locating roots.

[†] Any distance minimizer also minimizes the squared distance; we work with the latter in order to avoid square roots in the computation.

Lemma 4.17. *Suppose that $F(u)$ is defined and continuous for all $a \leq u \leq b$. Suppose that we can find a point $a < c < b$ such that $F(c) < F(a)$ and $F(c) < F(b)$. Then $F(u)$ has a minimum at some point $a < u^\star < b$.*

The proof is an easy consequence of Theorem 4.2. Therefore, if we find three points $a < c < b$ satisfying the conditions of the lemma, we are assured of the existence of a local minimum for the function between the two endpoints. Once this is done, we can design an algorithm to home in on the minimum $u^\star$. We choose another point, say d between a and c and evaluate $F(d)$. If $F(d) < F(c)$, then $F(d) < F(a)$ also, and so the points $a < d < c$ satisfy the hypotheses of Lemma 4.17. Otherwise, if $F(d) > F(c)$ then the points $d < c < b$ satisfy the hypotheses of the lemma. In either case, a local minimum has been narrowed down to a smaller interval, either $[\,a, c\,]$ or $[\,d, b\,]$. In the unlikely even that $F(d) = F(c)$, one can try another point instead — unless the objective function is constant, one will eventually find a suitable value of d. Iterating the method will produce a sequence of progressively smaller and smaller intervals in which the minimum is trapped, and, just like the Bisection Method, the endpoints of the intervals get closer and closer to the local minimum $u^\star$.

The one question is how to choose the point d. We described the algorithm when it was selected to lie between a and c, but one could equally well try a point between c and b. To speed up the algorithm, it makes sense to place d in the larger of the two subintervals $[\,a, c\,]$ and $[\,c, b\,]$. One could try placing d in the midpoint of the interval, but a more inspired choice is to place it at a fraction $\theta = \frac{5}{\sqrt{2}} - \frac{1}{2} \approx .61803$ of the way along the interval, i.e., at $\theta\, a + (1 - \theta)\, c$ if $[\,a, c\,]$ is the longer interval. (One could equally well take the point $(1 - \theta)\, a + \theta\, c$.) The result is the *Golden Section Method*. At each stage, the length of the interval has been reduced by a factor of θ, so the convergence rate is linear, although a bit slower than bisection.

Another strategy is to use an interpolating polynomial passing through the three points on the graph of $F(u)$ and use its minimum value as the next approximation to the minimum. The minimizing value occurs at

$$d = \frac{m\, s - n\, t}{s - t}\,,$$

where

$$s = \frac{F(c) - F(a)}{c - a}\,, \qquad t = \frac{F(b) - F(c)}{b - c}\,, \qquad m = \frac{a + c}{2}\,, \qquad n = \frac{c + b}{2}\,.$$

As long as $a < c < b$ satisfy the hypothesis of Lemma 4.17, we are assured that the quadratic interpolant has a minimum (and not a maximum!), and that the minimum remains between the endpoints of the interval: $a < d < b$. If the length of the interval is small, the minimum value should be a good approximation to the minimizer $u^\star$ of $F(u)$ itself. Once d is determined, the algorithm proceeds as before. In this case, convergence is not quite guaranteed, or, in unfavorable situations, could be much slower than the Golden Section Method. One can even try using the method when the function values do not satisfy the hypothesis of Lemma 4.17, although now the new point d will not necessarily lie between a and b. Worse, the quadratic interpolant may have a maximum at d, and one

ends up going in the wrong direction, which can even happen in the minimizing case due to the discrepancy between it and the objective function $F(u)$. Thus, this method must be handled with care.

A final idea is to focus not on the objective function $F(u)$ but rather its derivative $f(u) = F'(u)$. The critical points of F are the roots of $f(u) = 0$, and so one can use one of the solution methods, e.g., Bisection or Newton's Method, to find the critical points. Of course, one must then take care that the critical point $u^\star$ is indeed a minimum, as it could equally well be a maximum of the original objective function. (It will probably not be an inflection point, as these do not correspond to simple roots of $f(u)$.) The status of the critical point can be checked by looking at the sign of $F''(u^\star) = f'(u^\star)$; indeed, if we use Newton's Method we will be computing the derivative at each stage of the algorithm, and can stop looking if the derivative turns out to be of the wrong sign.

Gradient Descent

Now, let us turn our attention to multi-dimensional optimization problems. We are seeking to minimize a (smooth) scalar objective function $F(\mathbf{u}) = F(u_1, \dots, u_n)$. According to Theorem 4.5, at any given point $\mathbf{u}$ in the domain of definition of F, the negative gradient vector $-\nabla F(\mathbf{u})$, if nonzero, points in the direction of the steepest decrease in F. Thus, to minimize F, an evident strategy is to "walk downhill", and, to be efficient, walk downhill as fast as possible, namely in the direction $-\nabla F(\mathbf{u})$. After walking in this direction for a little while, we recompute the gradient, and this tells us the new direction to head downhill. With luck, we will eventually end up at the bottom of the valley, i.e., at a (local) minimum value of the objective function.

This simple idea forms the basis of the *Gradient Descent Method* for minimizing the objective function $F(\mathbf{u})$. In a numerical implementation, we start the iterative procedure with an initial guess $\mathbf{u}^{(0)}$, and let $\mathbf{u}^{(k)}$ denote the k^{th} approximation to the minimum $\mathbf{u}^\star$. To compute the next approximation, we set out from $\mathbf{u}^{(k)}$ in the direction of the negative gradient, and set

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} - t_k \, \nabla F(\mathbf{u}^{(k)}). \tag{4.27}$$

for some positive scalar $t_k > 0$. We are free to adjust t_k so as to optimize our descent path, and this is the key to the success of the method.

If $\nabla F(\mathbf{u}^{(k)}) \neq \mathbf{0}$, then, at least when $t_k > 0$ is sufficiently small,

$$F(\mathbf{u}^{(k+1)}) < F(\mathbf{u}^{(k)}), \tag{4.28}$$

and so $\mathbf{u}^{(k+1)}$ is, presumably, a better approximation to the desired minimum. Clearly, we cannot choose t_k too large or we run the risk of overshooting the minimum and reversing the inequality (4.28). Think of walking downhill in the Swiss Alps. If you walk too far in a straight line, which is what happens as t_k increases, then you might very well miss the valley and end up higher than you began — not a good strategy for descending to the bottom! On the other hand, if we choose t_k too small, taking very tiny steps, then the method may end up converging to the minimum much too slowly to be of practical use.

How should we choose an optimal value for the factor t_k? Keep in mind that the goal is to minimize $F(\mathbf{u})$. Thus, a good strategy would be to set t_k equal to the value of $t > 0$

that minimizes the scalar objective function

$$g(t) = F\big(\, \mathbf{u}^{(k)} - t\, \nabla F(\mathbf{u}^{(k)}) \,\big) \tag{4.29}$$

obtained by restricting $F(\mathbf{u})$ to the ray emanating from $\mathbf{u}^{(k)}$ that lies in the negative gradient direction. Physically, this corresponds to setting off in a straight line in the direction of steepest decrease, and continuing on until we cannot go down any further. Barring luck, we will not have reached the actual bottom of the valley, but must then readjust our direction and continue on down the hill in a series of straight line paths.

In practice, one can rarely compute the minimizing value $t^\star$ of (4.29) exactly. Instead, we employ one of the scalar minimization algorithms presented in the previous subsection. Note that we only need to look for a minimum among positive values of $t > 0$, since our choice of the negative gradient direction assures us that, at least for t sufficiently small and positive, $g(t) < g(0)$.

A more sophisticated approach is to employ the second order Taylor polynomial to approximate the function and then use its minimum (assuming such exists) as the next approximation to the desired minimizer. Specifically, if $\mathbf{u}^{(k)}$ is the current approximation to the minimum, then we approximate

$$F(\mathbf{u}) \approx c^{(k)} + (\mathbf{u} - \mathbf{u}^{(k)})^T \mathbf{g}^{(k)} + \tfrac{1}{2} (\mathbf{u} - \mathbf{u}^{(k)})^T H^{(k)} (\mathbf{u} - \mathbf{u}^{(k)}) \tag{4.30}$$

near $\mathbf{u}^{(k)}$, where

$$c^{(k)} = F(\mathbf{u}^{(k)}), \qquad \mathbf{g}^{(k)} = \nabla F(\mathbf{u}^{(k)}), \qquad H^{(k)} = \nabla^2 F(\mathbf{u}^{(k)}), \tag{4.31}$$

are, respectively, the function value, the gradient and the Hessian at the current iterate. If $\mathbf{u}^\star$ is a strict local minimum, then $\nabla^2 F(\mathbf{u}^\star)$ is positive definite, and hence, assuming $\mathbf{u}^{(k)}$ is close, so is $H^{(k)} = \nabla^2 F(\mathbf{u}^{(k)})$. Thus, the quadratic Taylor approximation has a unique minimum value $\mathbf{u}^{(k+1)}$ which satisfies the linear system

$$H^{(k)} (\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}) = - \mathbf{g}^{(k)}. \tag{4.32}$$

The solution serves to define the next approximation $\mathbf{u}^{(k+1)}$. While not guaranteed to converge, the method does perform well in all reasonable situations.

References

[**1**] Alligood, K.T., Sauer, T.D., and Yorke, J.A., *Chaos. An Introduction to Dynamical Systems*, Springer-Verlag, New York, 1997.

[**2**] Apostol, T.M., *Calculus*, Blaisdell Publishing Co., Waltham, Mass., 1967–69.

[**3**] Bradie, B., *A Friendly Introduction to Numerical Analysis*, Prentice–Hall, Inc., Upper Saddle River, N.J., 2006.

[**4**] Burden, R.L., and Faires, J.D., *Numerical Analysis*, Seventh Edition, Brooks/Cole, Pacific Grove, CA, 2001.

[**5**] Devaney, R.L., *An Introduction to Chaotic Dynamical Systems*, Addison–Wesley, Redwood City, Calif., 1989.

[**6**] Feigenbaum, M.J., Qualitative universality for a class of nonlinear transformations, *J. Stat. Phys.* **19** (1978), 25–52.

[**7**] Gaal, L., *Classical Galois theory*, 4th ed., Chelsea Publ. Co., New York, 1988.

[**8**] Greene, B., *The Elegant Universe*: *Superstrings, Hidden Dimensions, and the Quest for the Ultimate Theory*, W. W. Norton, New York, 1999..

[**9**] Henry, D., *Geometric Theory of Semilinear Parabolic Equations*, Lecture Notes in Math., vol. 840, Springer–Verlag, Berlin, 1981.

[**10**] Lanford, O., A computer-assisted proof of the Feigenbaum conjecture, *Bull. Amer. Math. Soc.* **6** (1982), 427–434.

[**11**] Mandelbrot, B.B., *The Fractal Geometry of Nature*, W.H. Freeman, New York, 1983.

[**12**] Marsden, J.E., and Tromba, A.J., *Vector Calculus*, 4th ed., W.H. Freeman, New York, 1996.

[**13**] Moon, F.C., *Chaotic Vibrations*, John Wiley & Sons, New York, 1987.

[**14**] Olver, P.J., and Shakiban, C., *Applied Linear Algebra*, Prentice–Hall, Inc., Upper Saddle River, N.J., 2005.

[**15**] Peitgen, H.-O., and Richter, P.H., *The Beauty of Fractals*: *Images of Complex Dynamical Systems*,Springer–Verlag, New York, 1986.

[**16**] Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., *Numerical Recipes*: *The Art of Scientific Computing*, 3rd ed., Cambridge University Press, Cambridge, 2007.

[**17**] Robinson, R.C., *An Introduction to Dynamical Systems: Continuous and Discrete*, 2nd ed., Pure and Applied Undergraduate Texts, vol. 19, Amer. Math. Soc., Providence, R.I., 2012.

[**18**] Royden, H.L., *Real Analysis*, Macmillan Co., New York, 1988.

