AIMS Lecture Notes 2006

Peter J. Olver

2. Numerical Solution of Scalar Equations

Most numerical solution methods are based on some form of iteration. The basic idea is that repeated application of the algorithm will produce closer and closer approximations to the desired solution. To analyze such algorithms, our first task is to understand general iterative processes.

2.1. 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 $k = 0, 1, 2, 3, \ldots$ takes on non-negative integer values.

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:

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

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

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,

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

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.*

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. Fixed points are 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 $u^{(0)}, u^{(1)}, u^{(2)}, \ldots$ are all well-defined.

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

$$u^{(k+1)} = a\,u^{(k)}, \qquad u^{(0)} = c. \tag{2.7}$$

The general solution to (2.7) is easily found:

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

and, in general,

$$u^{(k)} = a^k\,c. \tag{2.8}$$

If the initial condition is $a = 0$, then the solution $u^{(k)} \equiv 0$ is constant. Therefore, 0 is a *fixed point* or *equilibrium solution* for the iterative system.

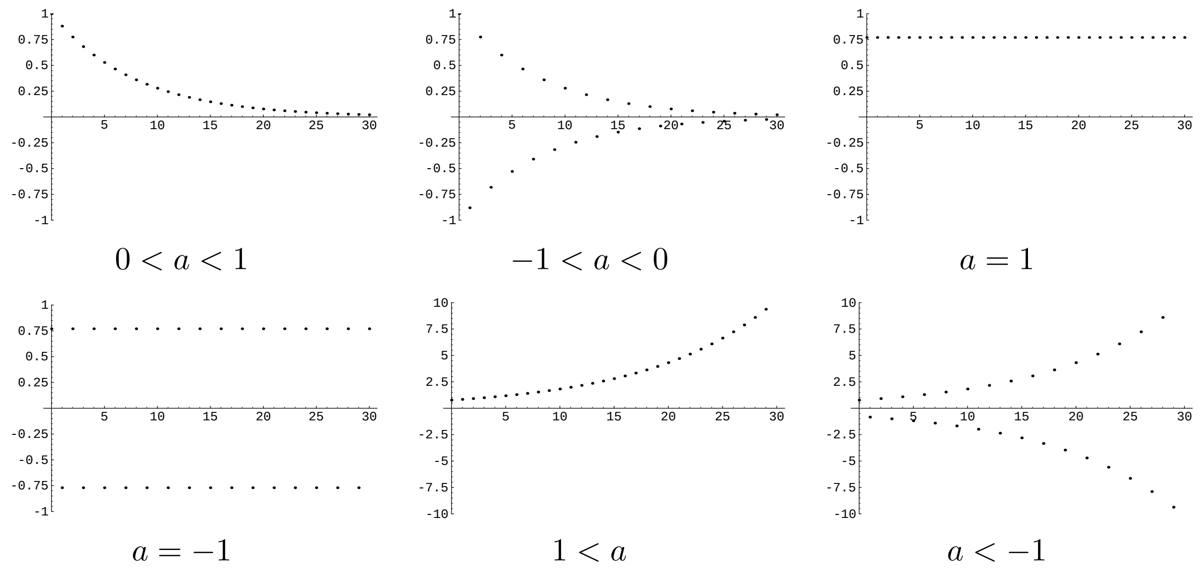

Figure 2.1. One Dimensional Real Linear Iterative Systems.

Example 2.3. Banks add interest to a savings account at discrete time intervals. For example, if the bank offers 5% interest compounded yearly, this means that the account balance will increase by 5% each year. Thus, assuming no deposits or withdrawals, the balance $u^{(k)}$ after k years will satisfy the iterative equation (2.7) with $a = 1 + r$ where $r = .05$ is the interest rate, and the 1 indicates that all the money remains in the account. Thus, after k years, your account balance is

$$u^{(k)} = (1+r)^k c, \qquad \text{where} \qquad c = u^{(0)} \tag{2.9}$$

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

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

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

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

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

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

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

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

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

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

In Figure 2.1 we exhibit representative *scatter plots* for the nontrivial cases (b – g). The horizontal axis indicates the index k and the vertical axis the solution value u. Each dot in the scatter plot represents an iterate $u^{(k)}$.

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

Let us also analyze complex scalar iterative systems. The coefficient a and the initial datum c in (2.7) are allowed to be complex numbers. The solution is the same, (2.8), but now we need to know what happens when we raise a complex number a to a high power. The secret is to write $a = r\,e^{\,\mathrm{i}\,\theta}$ in polar form, where $r = |\,a\,|$ is its modulus and $\theta = \operatorname{ph} a$ its angle or phase. Then $a^k = r^k\,e^{\,\mathrm{i}\,k\,\theta}$. Since $|\,e^{\,\mathrm{i}\,k\,\theta}\,| = 1$, we have $|\,a^k\,| = |\,a\,|^k$, and so the solutions (2.8) have modulus $|\,u^{(k)}\,| = |\,a^k\,a\,| = |\,a\,|^k\,|\,a\,|$. As a result, $u^{(k)}$ will remain bounded if and only if $|\,a\,| \leq 1$, and will tend to zero as $k \to \infty$ if and only if $|\,a\,| < 1$.

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

Theorem 2.4. *The zero solution to a* (real *or* complex) *scalar iterative system* $u^{(k+1)} = a\,u^{(k)}$ *is*

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

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

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

Nonlinear Scalar Iteration

The simplest "nonlinear" case is that of an affine function

$$g(u) = a\,u + b, \tag{2.10}$$

leading to an *affine discrete dynamical system*

$$u^{(k+1)} = a\,u^{(k)} + b. \tag{2.11}$$

The only fixed point is the solution to

$$u^\star = g(u^\star) = a\,u^\star + b, \qquad \text{namely,} \qquad u^\star = \frac{b}{1-a}\,. \tag{2.12}$$

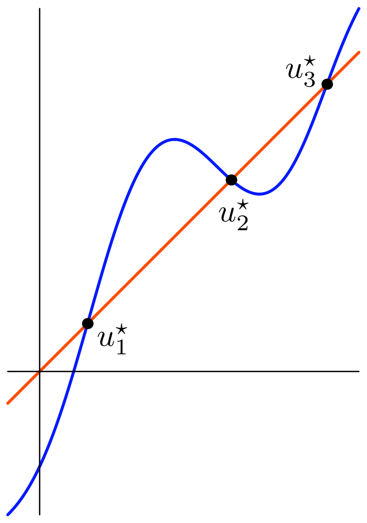

Figure 2.2. Fixed Points.

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

$$e^{(k)} = u^{(k)} - u^\star, \tag{2.13}$$

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 (2.12) from the iteration equation (2.11), we find

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

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

$$e^{(k+1)} = a\,e^{(k)}, \qquad \text{and hence} \qquad e^{(k)} = a^k e^{(0)}. \tag{2.14}$$

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 (2.11). 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 2.5. The affine function

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

leads to the iterative scheme

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

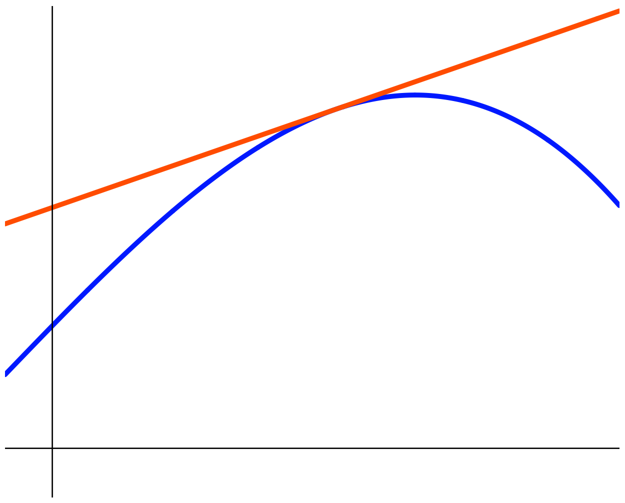

Figure 2.3. Tangent Line Approximation.

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

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 2.2 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 repre4sents the graph of an affine function; see Figure 2.3. 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

$$g(u) \approx g(u^\star) + g'(u^\star)(u - u^\star) \equiv a\,u + b, \tag{2.15}$$

where

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

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 2.6. *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,

$$e^{(k+1)} = u^{(k+1)} - u^\star = g(u^{(k)}) - g(u^\star) = g'(v)\,(u^{(k)} - u^\star) = g'(v)\,e^{(k)}, \tag{2.16}$$

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

$$|\, g'(v) \,| \leq \sigma < 1 \qquad \text{whenever} \qquad |\, v - u^\star \,| < \delta \tag{2.17}$$

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

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

Then the point v in (2.16), which is closer to $u^\star$ than $u^{(k)}$, satisfies (2.17). Therefore,

$$|\, u^{(k+1)} - u^\star \,| \;\leq\; \sigma\,|\, u^{(k)} - u^\star \,|, \qquad \text{and hence} \qquad |\, e^{(k+1)} \,| \;\leq\; \sigma\,|\, e^{(k)} \,|. \tag{2.18}$$

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 (2.17) 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 (2.17), 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.

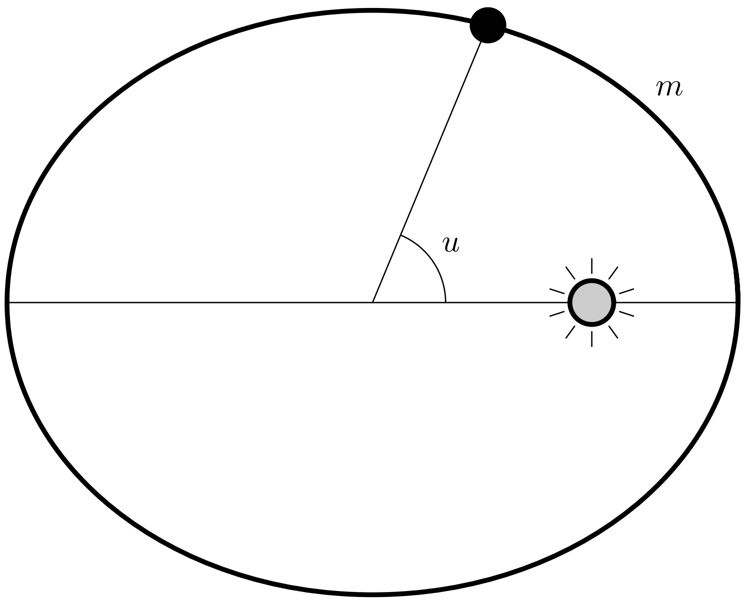

Figure 2.4. Planetary Orbit.

Example 2.7. Given constants ϵ, m, the trigonometric equation

$$u = m + \epsilon \sin u \tag{2.19}$$

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 2.4.

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

$$|\, g'(u) \,| = |\, \epsilon \cos u \,| = |\, \epsilon \,| < 1, \tag{2.20}$$

which automatically implies that the as yet unknown fixed point is stable. Indeed, condition (2.20) is enough to prove the existence of a unique stable fixed point; see the remark after Theorem 9.7. 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 $u^{(0)} = 0$ is

k	1	2	3	4	5	6	7	8	9
$u^{(k)}$	.5	.7397	.8370	.8713	.8826	.8862	.8873	.8877	.8878

After 13 iterations, we have converged sufficiently close to the solution (fixed point) $u^\star = .887862$ to have computed its value to 6 decimal places.

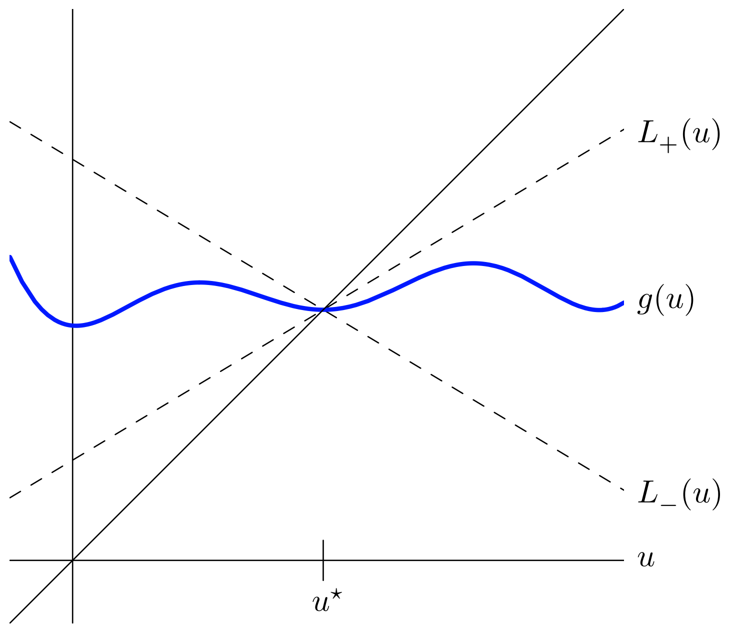

Figure 2.5. Graph of a Contraction.

Inspection of the proof of Theorem 2.6 reveals that we never really used the differentiability of g, except to verify the inequality

$$|\,g(u) - g(u^\star)\,| \le \sigma\,|\,u - u^\star\,| \qquad \text{for some fixed } \sigma < 1. \tag{2.21}$$

A function that satisfies (2.21) 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 2.5. In particular, any function that is differentiable at $u^\star$ with $|\,g'(u^\star)\,| < 1$ defines a contraction at $u^\star$.

Example 2.8. The simplest truly nonlinear example is a quadratic polynomial. The most important case is the so-called *logistic map*

$$g(u) = \lambda\,u(1-u), \tag{2.22}$$

where $\lambda \ne 0$ 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:

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

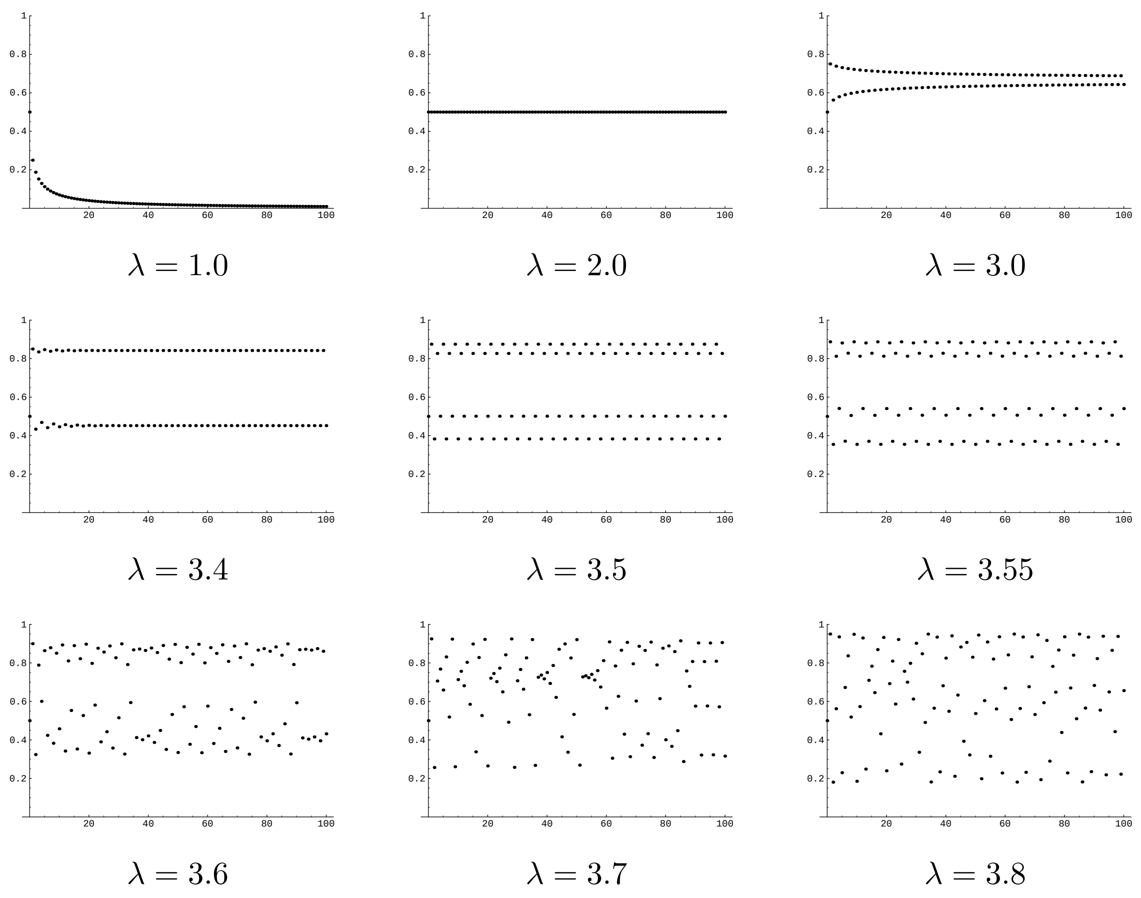

Figure 2.6. Logistic Iterates.

Let us apply Theorem 2.6 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 2.6, 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 valoue $u^{(k)}$ for $k = 0, \ldots, 100$. As expected from Theorem 2.6, 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 exitence of a (stable) *period* 2 *orbit* for the discrete dynamical system, in accordance with the following definition.

Definition 2.9. 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,

$$\frac{\lambda_{n+2} - \lambda_{n+1}}{\lambda_{n+1} - \lambda_n} \quad \longrightarrow \quad 4.6692\ldots , \tag{2.23}$$

known as *Feigenbaum's constant*. In the 1970's, the American physicist Mitchell Feigenbaum, [**16**], 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, [**32**].

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, but the reader can take it more figuratively for the purposes of this elementary exposition.

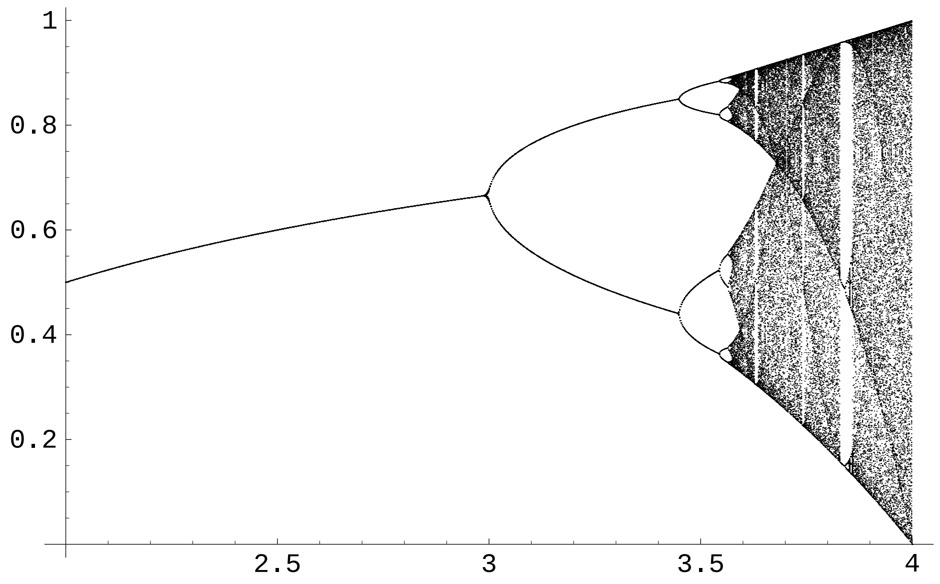

Figure 2.7. The Logistic Map.

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 $6, 12, 24, \ldots$ 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 (2.23). 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 2.7 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, [**33**], with their stunning, psychedelic fractal structure, [**42**].

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, [**35**]. 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 2.6, 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 (2.18) for the errors $e^{(k)} = u^{(k)} - u^\star$, namely

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

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

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

More generally, for any $j > 0$,

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

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 2.10. *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

$$|\, e^{(k+1)} \,| \;\leq\; \tau \,|\, e^{(k)} \,|^2 \tag{2.24}$$

for some constant $\tau > 0$.

Proof: Just as that of the linear convergence estimate (2.18), 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 (2.16) by the first order Taylor expansion

$$g(u) = g(u^\star) + g'(u^\star)\,(u - u^\star) + \tfrac{1}{2}\, g''(w)\,(u - u^\star)^2, \tag{2.25}$$

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 (2.25) reduces to

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

Therefore,

$$|\, g(u) - u^\star \,| \leq \tau \,|\, u - u^\star \,|^2, \tag{2.26}$$

where τ is chosen so that

$$\tfrac{1}{2} \,|\, g''(w) \,| \leq \tau \tag{2.27}$$

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 (2.26) to estimate the error

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

which establishes the quadratic convergence estimate (2.24). *Q.E.D.*

Let us see how the quadratic estimate (2.24) 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 (2.24) 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 2.11. 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

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

and hence $g'(u^\star) = 0$ vanishes at the fixed point. Theorem 2.10 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.

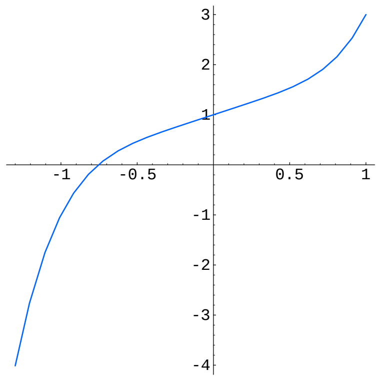

Figure 2.8. Graph of $u^5 + u + 1$.

2.2. Numerical Solution of Equations.

Solving nonlinear equations and systems of equations is, of course, a problem of utmost importance in mathematics and its manifold applications. We begin by studying the scalar case. Thus, we are given a real-valued function $f: \mathbb{R} \to \mathbb{R}$, and seek its *roots*, i.e., the real solution(s) to the scalar equation

$$f(u) = 0. \tag{2.28}$$

Here are some prototypical examples:

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

$$u^5 + u + 1 = 0. \tag{2.29}$$

Graphing the left hand side of the equation, as in Figure 2.8, 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 (2.28) 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 2.7. 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

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

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

[†] A modern proof of this fact relies on Galois theory, [**19**].

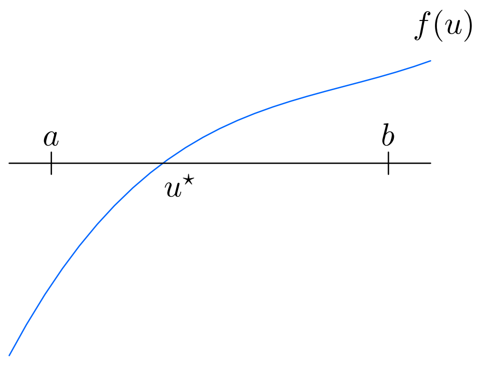

Figure 2.9. Intermediate Value Theorem.

by the second reaction, then the final equilibrium concentrations

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

of the reagents will be determined by solving the nonlinear system

$$(a - 2u - v)^2 (b - u) = \alpha (u + v), \qquad (a - 2u - v)(c - 3v)^3 = \beta (u + v), \tag{2.30}$$

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

Our immediate goal is to develop numerical algorithms for solving such nonlinear scalar equations.

The Bisection Method

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 2.9 for an illustration, and [**2**] for a proof.

Lemma 2.12. *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 2.10 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 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 $1/u = 0$.

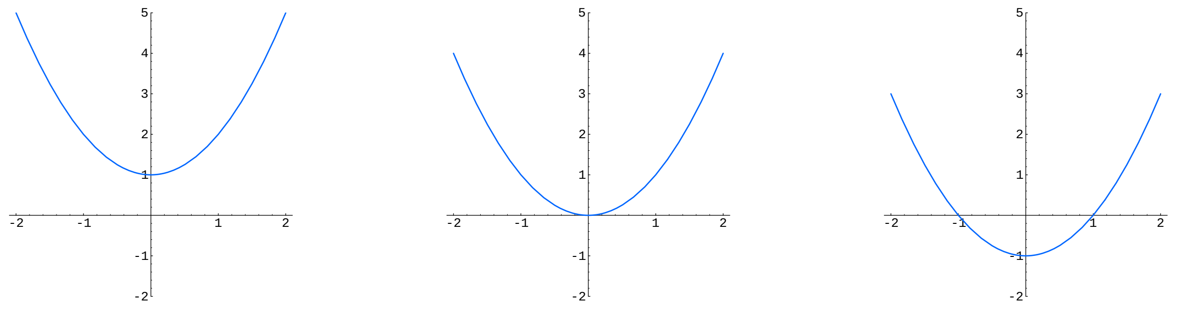

Figure 2.10. Roots of Quadratic Functions.

Note carefully that the Lemma 2.12 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 2.13. The roots of the quadratic equation

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

can be computed exactly by the quadratic formula:

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

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 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.

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	$-.0000$

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 kth 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:

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

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 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 $w^{(k)} = \frac{1}{2}(u^{(k)} + v^{(k)})$
        if $f(w^{(k)}) = 0$, stop; print $u^\star = w^{(k)}$
        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 $u^\star = w^{(n)} = \frac{1}{2}(u^{(n)} + v^{(n)})$
end
```

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

$$\left(\tfrac{1}{2}\right)^{n+1}(b-a) < \varepsilon, \qquad \text{or} \qquad n > \log_2 \frac{b-a}{\varepsilon} - 1\,. \tag{2.31}$$

Summarizing:

Theorem 2.14. *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 2.13, 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 (2.31), must perform $n = 34 > 33.22 \approx \log_2 2 \times 10^{10} - 1$ successive bisections[†].

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

Example 2.15. 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 (2.31), 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 a naïve example.

Example 2.16. To solve the cubic equation

$$f(u) = u^3 - u - 1 = 0 \tag{2.32}$$

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.

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

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

However, their values

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

rapidly become unbounded, and so fail to converge. This could, in fact, have been predicted by the convergence criterion in Theorem 2.6. 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 (2.32) 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 (2.17) 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 2.1, 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 2.10, 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

$$g(u) = u - h(u)\, f(u), \tag{2.33}$$

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

$$h(u^\star) = \frac{1}{f'(u^\star)} \tag{2.34}$$

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 (2.34), 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 (2.34) at all points, setting

$$h(u) = \frac{1}{f'(u)}\,, \tag{2.35}$$

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

$$g(u) = u \;-\; \frac{f(u)}{f'(u)}\,, \tag{2.36}$$

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

$$u^{(k+1)} = u^{(k)} - \frac{f(u^{(k)})}{f'(u^{(k)})}\,. \tag{2.37}$$

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 2.17. *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* (2.37) *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 (2.36) 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 2.10. *Q.E.D.*

Example 2.18. Consider the cubic equation

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

that we already solved in Example 2.16. The function used in the Newton iteration is

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

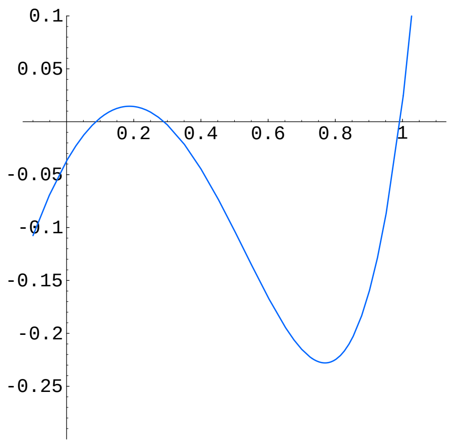

Figure 2.11. The function $f(u) = u^3 - \frac{3}{2}u^2 + \frac{5}{9}u - \frac{1}{27}$.

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 2.19. Consider the cubic polynomial equation

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

Since

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

the Intermediate Value Lemma 2.12 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 2.11 reconfirms this observation. Since we are dealing with a cubic polynomial, there are no other roots. (Why?)

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

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

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

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

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

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

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

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 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

$$y = f(u^{(k)}) + f'(u^{(k)})(u - u^{(k)}). \tag{2.38}$$

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

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

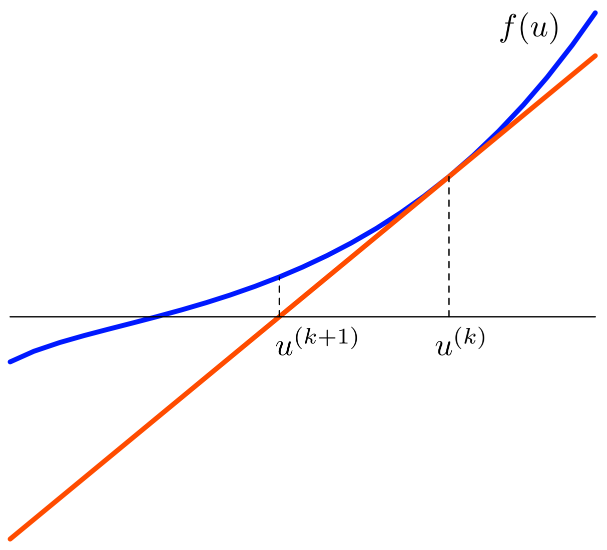

Figure 2.12. Newton's Method.

which represents a new, and, presumably more accurate, approximation to the desired root. The procedure is illustrated pictorially in Figure 2.12. Note that the passage from $u^{(k)}$ to $u^{(k+1)}$ is exactly the Newton iteration step (2.37). 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 2.1, including period doubling bifurcations and chaotic behavior. The reader is invited to experiment with simple examples; further details can be found in [**42**].

Example 2.20. For fixed values of the eccentricity ϵ, Kepler's equation

$$u - \epsilon \sin u = m \tag{2.39}$$

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 tghe method of *continuation* to determine how the solution $u^\star = h(m)$ depends upon the mean anomaly m. Namely, we start at

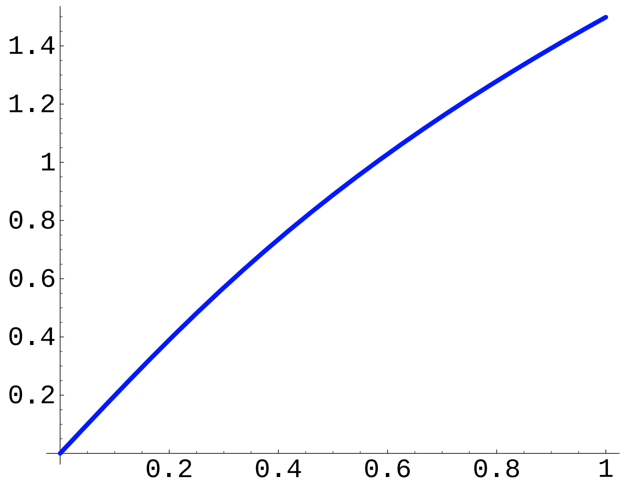

Figure 2.13. The Solution to the Kepler Equation for Eccentricity $\epsilon = .5$.

$m = m_0 = 0$ 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 $0 \leq m \leq 1$ using the value $\epsilon = .5$ appears in Figure 2.13.