Chapter 5

Life Is Really Nonlinear

5.1 Nonlinear Problems and Shooting

We consider in this chapter the solution of nonlinear problems. Suppose, for example, that the coefficients b, c and d in (3.1.1) are functions of v as well as x:

$$v''(x) = b(x, v(x))v'(x) + c(x, v(x))v(x) + d(x, v(x)). \tag{5.1.1}$$

Then (5.1.1) is a nonlinear equation for v. A simple example of this, which we will discuss later, is

$$v''(x) = 3v(x) + x^2 + 10[v(x)]^3, \qquad 0 \le x \le 1. \tag{5.1.2}$$

With either (5.1.1) or (5.1.2) we can have any of the boundary conditions discussed in Section 3.1, as well as others which are themselves nonlinear; for example,

$$[v(0)]^2 + v'(0) = 1, \qquad v(1) = [v'(1)]^2. \tag{5.1.3}$$

There are two basic approaches to such nonlinear problems. One is to discretize the differential equation as was done in Chapter 3 for linear equations; we will study this approach in Section 5.3. In the present section we will consider a method based on the solution of initial-value problems, and for this purpose we will first treat as an example the projectile problem of Chapter 2.

The Projectile Problem and Shooting

Recall that the projectile problem was given by equations (2.1.15), (2.1.17), and (2.1.18) with $\dot{m} = 0$ and $T = 0$:

$$\begin{aligned} \dot{x} &= v\cos\theta, \qquad \dot{y} = v\sin\theta, \\ \dot{v} &= \frac{-1}{2m}c\rho s v^2 - g\sin\theta, \qquad \dot{\theta} = -\frac{g}{v}\cos\theta. \end{aligned} \tag{5.1.4}$$

As before, we have the initial conditions

$$x(0) = y(0) = 0, \qquad v(0) = \bar{v}. \tag{5.1.5}$$

In Chapter 2 we also prescribed

$$\theta(0) = \bar{\theta}, \tag{5.1.6}$$

so that (5.1.4) - (5.1.6) was an initial-value problem. Suppose now, however, that in place of (5.1.6) we require that the projectile hit the ground at a given time t_f; that is,

$$y(t_f) = 0. \tag{5.1.7}$$

Since the equations (5.1.4) are nonlinear in the unknowns x, y, v, and θ, this is a nonlinear two-point boundary value problem. Note that it may not have a solution; for example, t_f may be too large for the given initial velocity $\bar{v}$.

We can base a numerical solution of this problem on the trial-and-error method that an artillery gunner might employ: choose a value of the launch angle, say $\bar{\theta}_1$, and "shoot," which, mathematically, means to solve the initial-value problem (5.1.4),(5.1.5) together with

$$\theta(0) = \bar{\theta}_1. \tag{5.1.8}$$

We follow the trajectory until $t = t_f$ and record the corresponding value of y at t_f, say y_1. If $y_1 \neq 0$ we choose another value of $\theta(0)$ and shoot again, continuing the process until a $\theta(0)$ has been found such that $y(t_f)$ is suitably close to zero. Shortly we will discuss systematic ways to choose new values of $\theta(0)$.

Other Boundary Value Problems

We can apply the above *shooting method* to other boundary value problems even though there may be no physical analogy to shooting. Consider, for example, the equation (5.1.2) with the boundary conditions

$$v(0) = \alpha, \qquad v(1) = \beta. \tag{5.1.9}$$

In this case we choose a trial value s for $v'(0)$ and solve the initial value problem

$$v'' = 3v + 10v^3 + x^2, \qquad v(0) = \alpha, \qquad v'(0) = s \tag{5.1.10}$$

up to $x = 1$. If the value of v at $x = 1$ is not sufficiently close to β, we adjust s and try again.

A key concern in the use of the shooting method is the adjustment of the parameter before the next shoot. We can address this question by recognizing that finding the right value of s is equivalent to finding a root of a nonlinear

function. To see why this is so, consider (5.1.10). Let $v(x;s)$ be the solution of the initial-value problem with $v'(0) = s$ and define

$$f(s) = v(1;s) - \beta.$$

Then in the shooting method we need to find a value of s for which $f(s) = 0$. We can, in principle, use any number of numerical methods for finding solutions of equations; some of these methods will be discussed in the following section.

Systems of Equations

The shooting method can also be applied to two-point boundary-value problems for general first-order systems. Consider the system

$$\mathbf{u}' = R(\mathbf{u}, t), \qquad 0 \le t \le 1, \tag{5.1.11}$$

where $\mathbf{u}(t)$ is the n -vector with components $u_i(t)$, $i = 1, \ldots, n$. Assume that m of the functions $u_1, \ldots, u_n$ are prescribed at $t = 1$ and that $n-m$ are prescribed at $t = 0$ so that we have the correct number, n, of boundary conditions. We will denote the set of functions prescribed at $t = 0$ by U_0 and those prescribed at $t = 1$ by U_1. Note that these sets may overlap; for example, u_1 may be given at both $t = 0$ and $t = 1$, but u_2 may not be given at either end point. To apply the shooting method, we select initial values $s_1, \ldots, s_m$ for the m functions not prescribed at $t = 0$ and solve numerically the initial-value problem

$$\mathbf{u}' = R(\mathbf{u}, t), \qquad \mathbf{u}(0) \text{ given by boundary conditions or } \{s_1, \ldots, s_m\}.$$

Next, we compare the values of those $u_i \in U_1$ with the integrated values $\mathbf{u}(1;\mathbf{s})$, where $\mathbf{s} = (s_1, \ldots, s_m)$. To solve the boundary-value problem the initial values s_i must be such that

$$u_i(1;\mathbf{s}) = \text{given value}, \qquad u_i \in U_1.$$

This is a system of m nonlinear equations in the m unknowns $s_1, \ldots, s_m$. We will consider methods for the solution of systems of nonlinear equations in Section 5.3.

Instability

Although the shooting method is simple in concept, it can suffer from instabilities in the initial-value problems. Instabilities of this type were discussed in Chapter 2, and we give here another simple example similar to the one in Section 2.5. Consider the problem

$$u'' - 100u = 0, \tag{5.1.12}$$

with the boundary conditions

$$u(0) = 1, \qquad u(1) = 0. \tag{5.1.13}$$

It is easy to verify that the exact solution of this boundary-value problem is

$$u(t) = \frac{1}{1 - e^{-20}} e^{-10t} - \frac{e^{-20}}{1 - e^{-20}} e^{10t}. \tag{5.1.14}$$

Now we attempt to obtain the solution by the shooting method using

$$u'(0) = s. \tag{5.1.15}$$

The exact solution of the corresponding initial-value problem is

$$u(t; s) = \frac{10 - s}{20} e^{-10t} + \frac{10 + s}{20} e^{10t}, \tag{5.1.16}$$

and we see that the value $u(1; s)$ at the end point $t = 1$ is very sensitive to s. The value of s that will give the exact solution (5.1.14) of the boundary-value problem is

$$s = -10 \left(\frac{1 + e^{-20}}{1 - e^{-20}} \right) \doteq -10.$$

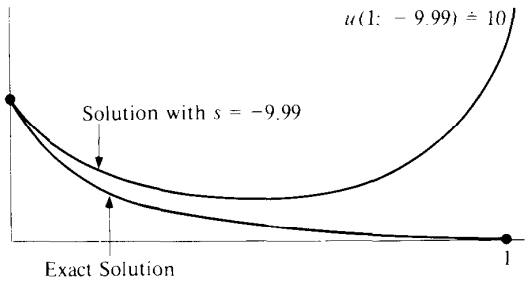


Figure 5.1: *Solutions of Exact and Nearby Problems*

If we solve the initial-value problem with the value of s correct to two decimal places, say $s = -9.99$, the solution of the initial-value problem is shown in Figure 5.1. The difficulty, of course, is that the solution of the initial-value problem grows like e^{10t}, and to suppress this fast-growing component, it is necessary to obtain a very accurate value of the initial condition. Even then,

rounding and discretization error in the solution of the initial value problem will tend to cancel out this accuracy in s.

Supplementary Discussion and References: 5.1

The shooting method for two-point boundary-value problems is described in many books on numerical analysis. A particularly detailed treatment is given in Roberts and Shipman [1972]. See also Ascher et al. [1988].

Another approach to solving two-point boundary-value problems by means of initial-value problems is the method of invariant embedding. Here, however, the initial-value problems are for partial differential equations, rather than ordinary differential equations. For a thorough discussion of invariant embedding, see Meyer [1973].

EXERCISES 5.1

5.1.1. Solve the two-point boundary-value problem $y'' + y' + y = -(x^2 + x + 1)$, $y(0) = y(1) = 0$, by the shooting method using one of the methods of Chapter 2 for initial-value problems. Check your numerical results by finding the exact analytical solution to this problem. (*Hint:* For the analytical solution, try the method of undetermined coefficients for a quadratic polynomial.)

5.1.2. Solve the projectile boundary-value problem (5.1.4),(5.1.5),(5.1.7) numerically using one of the methods of Chapter 2 for initial-value problems.

5.1.3. Attempt to solve problem (5.1.12),(5.1.13) using the same method you used for Exercise 5.1.2. Compare your best result with the exact solution given by (5.1.16) and discuss the discrepancies. Also discuss any difficulties you encountered in obtaining your numerical solution.

5.2 Solution of a Single Nonlinear Equation

In the last section, we saw that the shooting method with one free parameter can be viewed as a problem of finding a solution of a nonlinear equation

$$f(x) = 0. \tag{5.2.1}$$

We also saw in Chapter 2 that the use of implicit methods required solving a nonlinear equation. Many other areas in scientific computing lead to the problem of finding roots of equations, or, more generally, solutions of a system of nonlinear equations, which we will discuss in the next section. In the present section we restrict our attention to functions of a single variable.

An important special case of (5.2.1) occurs when f is a polynomial:

$$f(x) = a_n x^n + \cdots + a_1 x + a_0. \tag{5.2.2}$$

In this case we know from the fundamental theorem of algebra that f has exactly n real or complex roots if we count multiplicities of the roots. For a general function f it is usually difficult to ascertain how many solutions equation (5.2.1) has: there may be none, one, finitely many, or infinitely many. A simple condition that ensures that there is at most one solution in a given interval (a, b) is that

$$f'(x) > 0 \qquad \text{for all} \quad x \in (a, b) \tag{5.2.3}$$

(or $f'(x) < 0$ in the interval), although this does not guarantee that a root exists in the interval. (The proof of these statements is left to Exercises 5.2.1 and 5.2.2.) If, however, f is continuous, and

$$f(a) < 0, \qquad f(b) > 0, \tag{5.2.4}$$

then it is intuitively clear (and rigorously proved by a famous theorem of the calculus) that f must have at least one root in the interval (a, b).

The Bisection Method

Let us now assume that (5.2.4) holds and, for simplicity, that there is just one root in the interval (a, b). We do not necessarily assume that (5.2.3) holds; the situation might be as shown in Figure 5.2. One of the simplest ways of approximating a root of f in this situation is the *bisection method*, which we now describe.

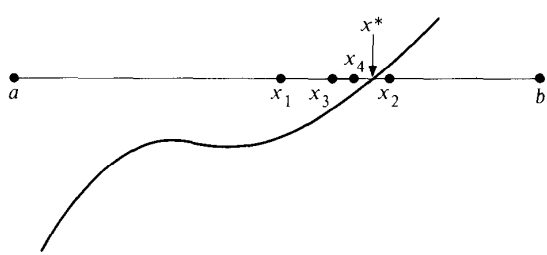


Figure 5.2: *The Bisection Method*

Let $x_1 = \frac{1}{2}(a + b)$ be the midpoint of the interval (a, b) and evaluate $f(x_1)$. If $f(x_1) > 0$, then the root, x^*, must lie between a and x_1; if $f(x_1) < 0$, which is the situation shown in Figure 5.2, then x^* is between x_1 and b. We now continue this process, always keeping the interval in which x^* is known to lie and evaluating f at its midpoint to obtain the next interval. For the function

shown in Figure 5.2, the steps would be as follows:

$$
\begin{aligned}
&f(x_1) < 0. \quad \text{Hence, } x^\star \in (x_1, b). \quad \text{Set } x_2 = \tfrac{1}{2}(x_1 + b).\\
&f(x_2) > 0. \quad \text{Hence, } x^\star \in (x_1, x_2). \quad \text{Set } x_3 = \tfrac{1}{2}(x_1 + x_2).\\
&f(x_3) < 0. \quad \text{Hence, } x^\star \in (x_3, x_2). \quad \text{Set } x_4 = \tfrac{1}{2}(x_2 + x_3).\\
&f(x_4) < 0. \quad \text{Hence, } x^\star \in (x_4, x_2). \quad \text{Set } x_5 = \tfrac{1}{2}(x_2 + x_4).\\
&\qquad\qquad\vdots
\end{aligned}
$$

Clearly, each step of the bisection procedure reduces the length of the interval known to contain $x^\star$ by a factor of 2. Therefore after m steps the length of the interval will be $(b-a)2^{-m}$, and this provides a bound on the error in our current approximation to the root; that is,

$$|x_m - x^\star| \leq \frac{|b-a|}{2^m}. \tag{5.2.5}$$

This bound has been obtained under the tacit assumption that the function values $f(x_i)$ are computed exactly. Of course, on a computer this will not be the case because of the rounding error (and possibly also discretization error – recall that the evaluation of the function f for the shooting method requires the solution of an initial-value problem). However, the bisection method does not use the value of $f(x_i)$ but only the sign of $f(x_i)$; therefore the bisection method is impervious to errors in evaluating the function f as long as the sign of $f(x_i)$ is evaluated correctly. One might think that the round-off error could not be so severe as to change the sign of the function, but this is not the case when the function values become sufficiently small. If the sign of $f(x_i)$ is incorrect, a wrong decision will be made in choosing the next subinterval, and the error bound (5.2.5) does not necessarily hold.

It is clear that if one makes a maximum error of E in evaluating f at any point in the interval (a, b), then the sign of f will be correctly evaluated as long as

$$|f(x)| > |E|.$$

Since the function f will be close to zero near the root $x^\star$, we can also argue the converse: there will be an *interval of uncertainty*, say, $(x^\star - \varepsilon, x^\star + \varepsilon)$, about the root in which the sign of f may not be correctly evaluated (see Figure 5.3). When our approximations reach this interval, their further progress toward the root is at best problematical. Unfortunately, it is extremely difficult to determine this interval in advance. It depends on the unknown root $x^\star$, the "flatness" of f in the neighborhood of the root, and the magnitude of the errors made in evaluating f. On the other hand, the interval is usually detectable during the course of the computation by an erratic behavior of the iterates; when this occurs, there is no longer any point in continuing the computation.

The fact that the sign of the function f may not be evaluated correctly near the root affects not only the bisection method but also the other methods we shall discuss later in the section.

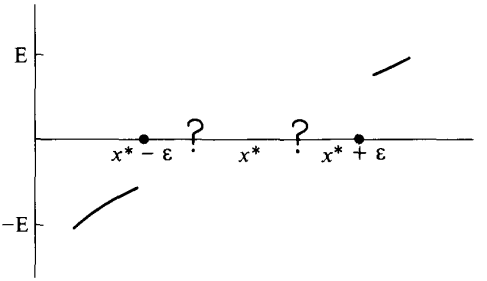


Figure 5.3: *The Interval of Uncertainty*

The Secant Method

One drawback of the bisection method is that it may be rather slow. To reduce the initial interval by a large factor, say 10^6, which may correspond to about six-decimal-digit accuracy, we would expect to require, from the error bound (5.2.5),

$$m = \frac{6}{\log_{10} 2} \doteq 20$$

evaluations of f. When each evaluation is expensive, as in the case of the shooting method, we would like to keep the number of evaluations as small as possible.

One possible way to speed up the bisection method is to use the values of the function f (instead of only its signs), and the simplest way to utilize this information is to choose the next point x_{i+1} as the zero of the linear function that interpolates f at x_{i-1} and x_i. This is shown in Figure 5.4. In the somewhat favorable situation shown in the figure, it is clear that x_{i+1} is a considerably better approximation to the root than would be the midpoint of the interval (x_{i-1}, x_i).

The linear interpolating function is given by (see Section 2.3, although it is easily checked directly)

$$l(x) = \frac{(x - x_{i-1})}{(x_i - x_{i-1})} f(x_i) - \frac{(x - x_i)}{(x_i - x_{i-1})} f(x_{i-1}), \tag{5.2.6}$$

and the root of this linear function is

$$x_{i+1} = \frac{x_{i-1} f(x_i) - x_i f(x_{i-1})}{f(x_i) - f(x_{i-1})}. \tag{5.2.7}$$

We may now proceed as in the bisection method, retaining x_{i+1} and either x_i or x_{i-1} so that the function values at the two retained points have different signs. This is the *regula falsi method.* Alternatively, in the *secant method*, we simply carry out (5.2.7) sequentially as indicated, keeping the last two iterates regardless of whether their function values have different signs.

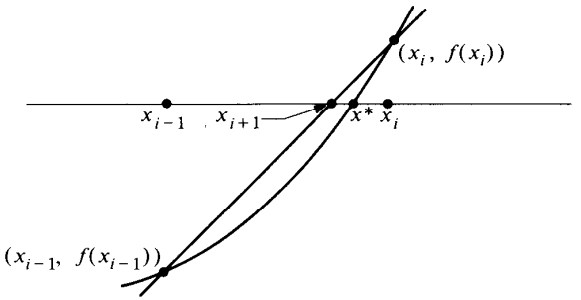


Figure 5.4: *The Secant Method*

It is better to rewrite (5.2.7) as

$$x_{i+1} = x_i - \frac{f(x_i)}{d_i}, \qquad d_i = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}, \tag{5.2.8}$$

which is easily verified as mathematically identical to (5.2.7). This form is preferable to (5.2.7) for computation since there is less cancellation.

Newton's Method

We can consider the quantity d_i in (5.2.8) to be a difference approximation to $f'(x_i)$, and, thus (5.2.8) may be viewed as a "discrete form" of the iterative method

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \tag{5.2.9}$$

This is known as *Newton's method* and is the most famous iterative method for obtaining roots of equations (as well as for solving systems of nonlinear equations, as we shall see in the next section). Geometrically, Newton's method can be interpreted as approximating the function f by the linear function

$$l_i(x) = f(x_i) + (x - x_i)f'(x_i),$$

which is tangent to f at x_i, and then taking the next iterate x_{i+1} to be the zero of $l_i(x)$; this is shown in Figure 5.5.

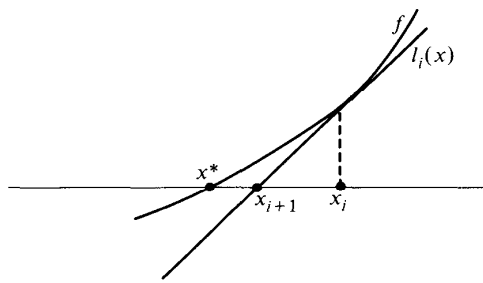


Figure 5.5: *Newtons's Method*

Iteration Functions and Convergence

The Newton iteration (5.2.9) can be written in the form

$$x_{i+1} = g(x_i), \tag{5.2.10}$$

where

$$g(x) = x - \frac{f(x)}{f'(x)}. \tag{5.2.11}$$

Many other iterative methods may also be written in the general form (5.2.10) for some *iteration function* g. For example, a very simple method is given by defining g to be

$$g(x) = x - \alpha f(x) \tag{5.2.12}$$

for some scalar α. This is sometimes called the *chord method* and is illustrated in Figure 5.6.

Iterative methods of the form (5.2.10) are called *one-step methods* since x_{i+1} depends only on the previous iterate x_i. On the other hand, the secant method (5.2.8) depends on both x_i and x_{i-1} and is an example of a *multistep method.* (Note the analogy with one-step and multistep methods for initial-value problems.)

To be useful the iteration function g must have the property

$$x^\star = g(x^\star) \tag{5.2.13}$$

for a root $x^\star$ of f. This is clearly the case for (5.2.11) and (5.2.12). (This is true for (5.2.11) even when $f'(x^\star) = 0$, although in this case we must define $g(x^\star)$ as the limit as $x \to x^\star$; see Exercise 5.2.3.) A value of $x^\star$ that satisfies (5.2.13) is called a *fixed point* of the function g. Assuming that the function g has a fixed point x^*, an important question is the convergence of the iterates x_i to x^*.

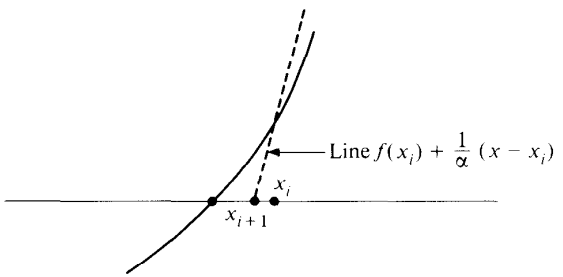


Figure 5.6: *The Chord Method*

We now discuss a basic property that ensures convergence of the iterates (5.2.10), at least when the starting iterate is sufficiently close to $x^\star$. We assume that g is continuously differentiable in a neighborhood of $x^\star$, that (5.2.13) holds, and that

$$|g'(x)| \leq \gamma < 1, \qquad \text{if } |x - x^\star| \leq \beta. \tag{5.2.14}$$

By the mean-value theorem of the calculus, we can write

$$g(x) - g(x^\star) = g'(\xi)(x - x^\star), \tag{5.2.15}$$

where ξ is between x and $x^\star$. Therefore if $|x - x^\star| \leq \beta$, then we can apply (5.2.14) to conclude that

$$|g(x) - g(x^\star)| \leq \gamma |x - x^\star|, \qquad \text{if } |x - x^\star| \leq \beta. \tag{5.2.16}$$

Suppose now that $|x_0 - x^\star| \leq \beta$. Then, using (5.2.10) and (5.2.13), we see from (5.2.16) that

$$|x_1 - x^\star| = |g(x_0) - g(x^\star)| \leq \gamma |x_0 - x^\star|.$$

Since $\gamma < 1$, this shows that x_1 is closer to $x^\star$ than x_0. Thus, $|x_1 - x^\star| \leq \beta$, and we can do the same thing again to obtain

$$|x_2 - x^\star| \leq \gamma |x_1 - x^\star| \leq \gamma^2 |x_0 - x^\star|,$$

and, in general,

$$|x_n - x^\star| \leq \gamma |x_{n-1} - x^\star| \leq \cdots \leq \gamma^n |x_0 - x^\star|. \tag{5.2.17}$$

Since $\gamma < 1$, this shows that $x_n \to x^\star$ as $n \to \infty$ (assuming no rounding or other errors).

It will be argued that (5.2.14) is an uncheckable condition since it requires knowing something about g' near x^*, which is unknown. Surprisingly, however, we can obtain valuable information from the preceding analysis even without knowing x^*. As a first illustration of this we consider an analysis of the second-order Adams-Moulton formula described in Section 2.4 for the solution of the ordinary differential equation $y' = f(y)$, where, for simplicity, we have dropped the dependence of f on x. The implicit formula is then given in (2.4.12) as

$$y_{k+1} = y_k + \frac{h}{2}[f(y_{k+1}) + f_k]. \tag{5.2.18}$$

This is a nonlinear equation for y_{k+1}, although it was used in Section 2.4 only as a "corrector formula"; that is, a predicted value $y_{k+1}^{(0)}$ was computed by an explicit method and then used in (5.2.18) to obtain a new estimate of y_{k+1} by

$$y_{k+1}^{(1)} = y_k + \frac{h}{2}[f(y_{k+1}^{(0)}) + f_k]. \tag{5.2.19}$$

Now we can correct this value again by using it in place of $y_{k+1}^{(0)}$ in (5.2.19). If we do this repeatedly, we obtain the sequence defined by

$$y_{k+1}^{(i+1)} = y_k + \frac{h}{2}[f(y_{k+1}^{(i)}) + f_k], \qquad i = 0, 1, \ldots . \tag{5.2.20}$$

Clearly, (5.2.20) is just the iteration process $y_{k+1}^{(i+1)} = g(y_{k+1}^{(i)})$, where

$$g(y) = y_k + \frac{h}{2}[f(y) + f_k].$$

If y_{k+1} is the exact solution of (5.2.18), we can apply the previous analysis to conclude that the sequence of (5.2.20) will converge to y_{k+1} provided that $y_{k+1}^{(0)}$ (the predicted value) is sufficiently close to y_{k+1} and that

$$|g'(y)| = \left|\frac{h}{2}f'(y)\right| < 1$$

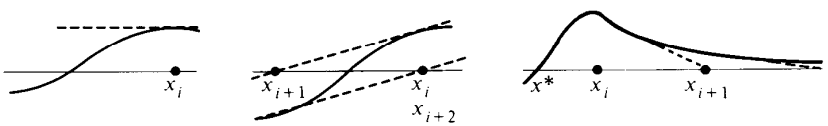
in a neighborhood of y_{k+1}; this will hold if h is sufficiently small.

As another illustration of the use of the convergence analysis, we consider Newton's method. Assume that $f'(x^*) \neq 0$ and that f is twice continuously differentiable in a neighborhood of x^*. Thus, by continuity, $f'(x) \neq 0$ in some neighborhood of x^*, and we can differentiate the Newton iteration function (5.2.11) to obtain

$$g'(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} = \frac{f(x)f''(x)}{[f'(x)]^2}.$$

Hence $g'(x^\star) = 0$, since $f(x^\star) = 0$. Therefore, by continuity, (5.2.14) must hold in a neighborhood of $x^\star$, and we conclude that the Newton iterates converge if x_0 is sufficiently close to $x^\star$. This shows that, under rather mild assumptions, the Newton iterates *must* converge to a root provided that x_0 (or any iterate x_k) is sufficiently close to $x^\star$. Although this type of convergence theorem, known as a *local convergence theorem*, does not help one decide if the iterates will converge from a given x_0, it gives an important intrinsic property of the iterative method.

When an iterate is not sufficiently close to a solution, various types of "bad" behavior can occur with Newton's method, as shown in Figure 5.7. Figure 5.7(a) illustrates that if $f'(x_i) = 0$, the next Newton iterate is not defined and the tangent line to f at x_i is horizontal. Figure 5.7(b) indicates the possibility of "cycling," in which $x_{i+2} = x_i$, and this cycle then repeats (see Exercise 5.2.5); thus there is no convergence but no divergence either. Cycles of order higher than 2 are also possible. Figure 5.7.(c) shows divergence to infinity, as would be the case if x_i is outside the domain of convergence to the solution of interest and the function behaves like, for example, e^{-x} as $x \to \infty$.



(a) $f'(x_i) = 0$ (b) Oscillation (c) Divergence

Figure 5.7: *Possible "Bad" Behavior of Newton's Method*

Convexity

In contrast to the above instances of bad behavior, there are situations in which Newton's method will converge for any starting approximation, no matter how far from the solution. In this case we speak of *global convergence.* Perhaps the simplest functions for which global convergence is obtained are those that are convex. A function is *convex* if it satisfies any one of the following equivalent properties, depending upon the differentiability of the function:

$$f''(x) \geq 0, \qquad \text{for all } x, \tag{5.2.21a}$$

$$f'(y) \geq f'(x), \qquad \text{if } y \geq x, \tag{5.2.21b}$$

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha) f(y), \tag{5.2.21c}$$

where (5.2.21c) holds for any $\alpha \in (0,1)$ and all x, y.

A linear function $f(x) = ax + b$ is always convex, as is easily checked by any of the definitions of (5.2.21). However, we are mostly interested in functions that actually "bend upwards," as illustrated in Figure 5.8. Such functions are *strictly convex* and satisfy (5.2.21b,c) with strict inequality whenever $x \neq y$. Strict inequality in (5.2.21a) is also sufficient for strict convexity, but not necessary; the function $f(x) = x^4$ is strictly convex although $f''(0) = 0$.

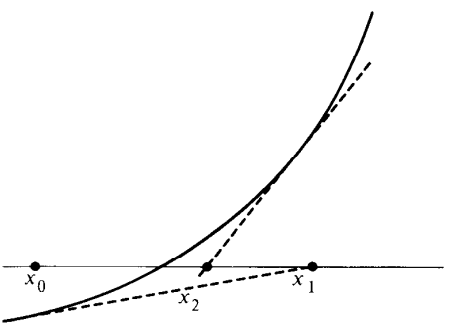


Figure 5.8: *Convergence of Newton's Method for a Convex Function*

A convex function may have infinitely many roots ($f(x) \equiv 0$) and even a strictly convex function may have no roots (for example, $f(x) = e^{-x}$). In the sequel, we will assume that f is strictly convex, $f'(x) > 0$ for all x, and $f(x) = 0$ has a solution; as noted previously, the condition on f' ensures that the solution is unique. These assumptions are illustrated by the function in Figure 5.8. In this case if x_0 is to the right of the solution, the Newton iterates converge monotonically to the solution, as is intuitively clear by drawing the tangent lines to the curve (see also Exercise 5.2.10). If x_0 is to the left of the solution, as shown in Figure 5.8, then the next Newton iterate is to the right of the solution, and thereafter the Newton iterates again converge monotonically to the solution. Figure 5.8 shows a function for which $f'(x) > 0$. If $f'(x) < 0$, the corresponding situation holds, but monotone convergence is now from left to right (Exercise 5.2.11). Similar convergence statements can be made if f is *concave*, that is, if $-f$ is convex.

The above discussion assumes that the properties of f hold for all x, in which case we obtain global convergence. They may, however, hold only in a neighborhood of a solution, and this will again ensure monotone convergence of the Newton iterates for suitable starting approximations x_0. For example, in Figure 5.7(c) there will be an interval $[x^*, b]$ for which the Newton iterates

will converge monotonically to $x^\star$ if $x_0 \in [x^\star, b]$. See also Exercises 5.2.5 and 5.2.6.

Rate of Convergence

From the standpoint of economical computation, the rate at which iterates converge to a root is almost as important as whether they converge at all. Suppose, in analogy to the estimate (5.2.17), that the errors behave as

$$|x^\star - x_{i+1}| = \gamma |x^\star - x_i|,$$

where γ is very close to 1, say $\gamma = 0.999$. Then, reducing the error in a given iterate by a factor of 10 would require well over two thousand iterations. Clearly, we wish the γ in the estimate (5.2.17) to be as small as possible, and from the derivation of (5.2.16) we see that it can be no smaller than $|g'(x^\star)|$. If $g'(x^\star) \neq 0$, then the rate of convergence is said to be *linear* or *geometric*, and $|g'(x^\star)|$ is the *asymptotic convergence factor*. Recall, however, that for Newton's method we showed that $g'(x^\star) = 0$ under the assumption that $f'(x^\star) \neq 0$. This does *not* imply, of course, that the iterates converge in one step, but it signals that the rate of convergence is faster than linear. In particular, it can be shown (see Exercise 5.2.4) that close to the solution the errors in Newton's method satisfy

$$|x^\star - x_{i+1}| \leq c|x^\star - x_i|^2, \tag{5.2.22}$$

where c depends on the ratio of f'' to f' near $x^\star$. The relation (5.2.22) defines *quadratic convergence*; the iterates converge very rapidly once they begin to get close to a root. For example, suppose $c = 1$ and $|x^\star - x_i| \doteq 10^{-3}$. Then $|x^\star - x_{i+1}| \doteq 10^{-6}$, so that the number of correct decimal places has been doubled in one iteration. It is this property of quadratic convergence that makes Newton's method of central importance. Quadratic convergence is lost, however, when $f'(x^\star) = 0$, so that $x^\star$ is a multiple root (see Exercise 5.2.12).

As an illustration of quadratic convergence in Newton's method, consider the problem of finding the zeros of $f(x) = 1/x + \ln x - 2$. This function is defined for all positive values of x and has two zeros: one between $x = 0$ and $x = 1$ and the other between $x = 6$ and $x = 7$, as illustrated in Figure 5.9. Table 5.1 contains a summary of the first six iterations of Newton's method using the starting value of $x = 0.1$. Note that once an approximation is "close enough" (in this case, after three iterations), the number of correct digits doubles in each iteration, which shows the quadratic convergence.

Rounding Error

So far the discussion of Newton's method has been predicated upon exact computation of the iterates, but rounding or other errors will inevitably cause

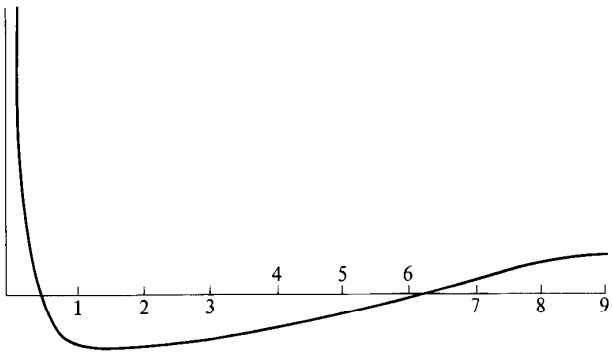


Figure 5.9: *The Function* $f(x) = 1/x + lnx - 2$

Table 5.1: *Convergence of Newton's Method for* $f(x) = 1/x + lnx - 2$

Iteration	x_{i-1}	$f(x_{i-1})$	x_i	Number of Correct Digits
1	0.1	5.6974149	0.16330461	0
2	0.16330461	2.3113878	0.23697659	0
3	0.23697659	0.7800322	0.29438633	1
4	0.29438633	0.1740346	0.31576121	2
5	0.31576121	0.0141811	0.31782764	4
6	0.31782764	0.0001134	0.31784443	8

the iterates to be computed inaccurately. For example, if ε_i and ε_i' are the errors made in computing $f(x_i)$ and $f'(x_i)$, respectively, then the computed next iterate $\hat{x}_{i+1}$ is

$$\hat{x}_{i+1} = x_i \ominus (f(x_i) + \varepsilon_i) \oslash (f'(x_i) + \varepsilon_i'),$$

where the circled operations indicate that rounding errors are also made in the subtraction and division. A full analysis of the effects of these errors is difficult, if even possible, and we content ourselves with the following remarks. If the errors ε_i and ε_i' are small, we can expect the computed iterates to behave roughly as the exact iterates would, at least as long as we are not close to the root. However, when $f(x_i)$ becomes so small that it is comparable in size to ε_i, then the computed iterates no longer behave like the exact ones. In particular, we saw in the case of the bisection method that when the sign of f can no longer be evaluated correctly, the method breaks down in the

sense that a wrong decision may be made as to which interval the root is in. An analogous thing happens with Newton's method: if the sign of $f(x_i)$ is evaluated incorrectly, but that of $f'(x_i)$ correctly (a reasonable assumption if $f'(x^*)$ is not particularly small), then the computed value of $f(x_i)/f'(x_i)$ has the wrong sign, and the computed next iterate moves in the wrong direction.

As with the bisection method, the notion of an interval of uncertainty about the root x^* applies equally well to Newton's method (as well as to essentially all iterative methods).

Ill-conditioning

In Chapter 4 we discussed ill-conditioning of a solution of a system of linear equations; an analogous problem can occur with roots of nonlinear equations. The simplest example of this is given by the trivial polynomial equation

$$x^n = 0,$$

which has an n-fold root equal to zero, and the polynomial equation

$$x^n = \varepsilon, \qquad \varepsilon > 0,$$

whose n roots are $\varepsilon^{1/n}$ times the nth roots of unity and therefore all have absolute value of $\varepsilon^{1/n}$. If, for example, $n = 10$ and $\varepsilon = 10^{-10}$, the roots of the second polynomial have absolute value 10^{-1}; thus, a change of 10^{-10} in one coefficient (the constant term) of the original polynomial has caused changes 10^9 times as great in the roots.

This simple example is a special case of the general observation that if a root x^* of a polynomial f is of multiplicity m, then small changes of order ε in the coefficients of f may cause a change of order $\varepsilon^{1/m}$ in x^*; an analogous result holds for functions other than polynomials by expanding in a Taylor series about x^*.

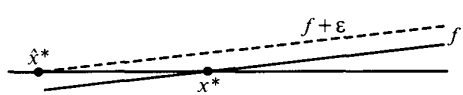


Figure 5.10: *A Large Change in x^* Due to a Small Change in f*

A necessary condition for a multiple root at x^* is that $f'(x^*) = 0$. If $f'(x^*) \neq 0$ but $f'(x)$ is small in the neighborhood of x^*, then small changes in f can still cause large changes in x^*, as Figure 5.10 illustrates. Perhaps the most famous example of how ill-conditioned nonmultiple roots can be is given by the following. Let f be the polynomial of degree 20 with roots $1, \ldots, 20$, and let $\hat{f}$ be the same polynomial but with the coefficient of x^{19} changed by $2^{-23} \doteq 10^{-7}$. Then the roots of $\hat{f}$ to one decimal place are given by

1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 8.9

10.1 ± 0.6i 11.8 ± 1.7i 14.0 ± 2.5i 16.7 ± 2.8i 19.5 ± 1.9i 20.8

Since the coefficient of x^{19} in $f(x)$ is 210, we see that a change of about $10^{-7}\%$ in this one coefficient has caused such large changes in the roots that some have even become complex!

The Shooting Method

We end this section with a discussion of the shooting method introduced in Section 5.1. Consider the boundary value problem

$$v''(x) = g(x, v(x)), \qquad v(0) = \alpha, \qquad v(1) = \beta. \tag{5.2.23}$$

The equation (5.1.2) is a special case of (5.2.23) in which $g(x, v) = x^2 + 3v + 10v^3$. To apply the shooting method we solve the initial value problem

$$v''(x) = g(x, v(x)), \qquad v(0) = \alpha, \qquad v'(0) = s, \tag{5.2.24}$$

and denote the solution by $v(x; s)$. Then, as in Section 5.1, we define the function

$$f(s) = v(1; s) - \beta, \tag{5.2.25}$$

and we wish to solve the equation $f(s) = 0$.

Consider first the bisection method, in which we only need to evaluate the function f. Each such evaluation requires that we solve the initial value problem (5.2.24) so as to obtain $v(1; s)$. In general, the solution of (5.2.24) can be accomplished by any of the methods of Chapter 2 (after first converting the second order equation to a system of two first-order equations). To begin the bisection method we would need to find s_0 and s_1 so that $f(s_0)$ and $f(s_1)$ have different signs. This might require a little trial and error but then the bisection method proceeds systematically. Note that, in general, the computed $f(s)$ will be inaccurate because of both discretization error in solving the initial value problem as well as rounding error, and the bisection method will break down when the sign of f can no longer be evaluated correctly.

For this type of problem the evaluation of $f(s)$ can be time consuming since it requires the solution of the initial value problem (5.2.24). Therefore we would like to utilize the potentially rapid convergence of Newton's method. For Newton's method we need $f'(s)$ and we differentiate (5.2.25) to obtain

$$f'(s) = v_s(1; s), \tag{5.2.26}$$

where $v_s(1; s)$ is the partial derivative of $v(x; s)$ with respect to s and evaluated at $x = 1$. In order to obtain a way of computing $v_s(1; s)$, we differentiate

$$v''(x; s) = g(x, v(x; s))$$

with respect to s to obtain

$$\frac{\partial}{\partial s}(v''(x;s)) = g_v(x, v(x;s))v_s(x;s), \tag{5.2.27}$$

where $g_v(x,v)$ is the partial derivative of $g(x,v)$ with respect to v. Assuming that differentiation with respect to s and x can be interchanged on the left side of (5.2.27), we then have

$$v_s''(x;s) = g_v(x, v(x,s))v_s(x;s), \tag{5.2.28}$$

which is called the *adjoint equation* for v_s. In (5.2.28) we are assuming that s is held fixed so that this is a differential equation for $v_s(x;s)$ considered only as a function of x. The initial conditions for (5.2.28) are obtained by differentiating those of (5.2.24) with respect to s; thus

$$v_s(0;s) = 0, \qquad v_s'(0;s) = 1. \tag{5.2.29}$$

If we knew the exact solution $v(x;s)$ of (5.2.24), we could put this into g_v in (5.2.28) to obtain a known function of x, and then (5.2.28), (5.2.29) would be a linear initial-value problem for $v_s(x;s)$. Upon solving this initial value problem we obtain $f'(s) = v_s(1;s)$. Of course, we do not know the exact solution of (5.2.24), but we solve this initial-value problem approximately to obtain values $v_i \doteq v(x_i;s)$ at the grid points x_i. We can then use these approximate values to evaluate g_v in (5.2.28) and in this way we can obtain an approximate solution to (5.2.28) at the same time we obtain the approximate solution to (5.2.24). Thus we will be able to carry out Newton's method, at least approximately, for $f(s) = 0$. We could also approximate Newton's method by the secant or regula falsi methods in which only values of $f(s)$ are required.

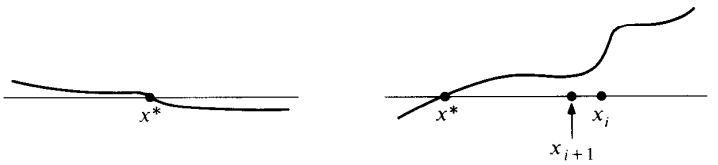
We should caution, however, that the shooting method is not always viable, as discussed in Section 5.1, and the finite difference method to be discussed in Section 5.3 may be preferable for two-point boundary value problems.

Supplementary Discussion and References: 5.2

For a thorough treatment of the theory of iterative methods for roots of equations, see Traub [1964], and for an excellent discussion of rounding error, see Wilkinson ([1963], [1965]). In particular, the example of the ill-conditioned polynomial of degree 20 is due to Wilkinson.

For roots of polynomials there are a number of specialized methods, such as Bairstow's method for polynomials with real coefficients but complex roots, and Laguerre's method, which has the property of cubic convergence (that is, an error estimate of the form (5.2.22) holds with $|x^* - x_i|^3$ on the right-hand side). Also, roots of polynomials are eigenvalues of the corresponding companion matrix and can be obtained in principle by the methods of Chapter 7.

When to stop the iteration is a problem for which there is still no definitive solution. The usual simplest tests are $|f(x_i)| < \varepsilon$ or $|x_{i+1} - x_i| < \varepsilon$, where ε is some given tolerance. The first can be misleading when the function f is very "flat" near the root, as is the case of a multiple root, and the second can fail in a variety of situations depending on the iterative method. For example, for Newton's method it can fail when the derivative is very large at the current iterate. These two possibilities are depicted in the following figure:



EXERCISES 5.2

5.2.1. If f is a continuously differentiable function, use the mean-value theorem (see Appendix 1) to show that if (5.2.3) holds then f has at most one root in the interval (a, b).

5.2.2. Let $f(x) = e^x$. Show that $f'(x) > 0$ for all x but f does not have any finite roots.

5.2.3. Let f be twice continuously differentiable and suppose that $f'(x^*) = 0$ at a root x^* of f but $f'(x) \neq 0$ in a neighborhood of x^*. Show that the limit of the iteration function g of (5.2.11) exists and equals x^* as $x \to x^*$.

5.2.4. Let f be twice continuously differentiable and suppose that $f'(x^*) \neq 0$ at a root x^* of f. Show that the error relation (5.2.22) holds for Newton's method in some interval about x^*. *Hint:* Expand $0 = f(x^*) = f(x) + f'(x)(x^* - x) + \frac{1}{2}f''(\xi)(x^* - x)^2$, solve this for x^*, and then use (5.2.9).

5.2.5. Consider the function $f(x) \equiv x - x^3$ with roots at 0 and ± 1.

a. Show that Newton's method is locally convergent to each of the three roots.

b. Carry out several steps of Newton's method starting with the initial approximation $x_0 = 2$. Discuss the rate of convergence that you observe in your computed iterates.

c. Carry out several steps of both the bisection and secant methods starting with the interval $(\frac{3}{4}, 2)$. Compare the rate of convergence of the iterates from these methods with that of the Newton iterates.

d. Determine the set of points S for which the Newton iterates will converge (in the absence of rounding errors) to the root 1 for any starting approximation x_0 in S. Do the same for the roots 0 and -1.

5.2.6. Consider the equation $x - 2\sin x = 0$.

a. Show graphically that this equation has precisely three roots: 0, and one in each of the intervals $(\pi/2, 2)$ and $(-2, -\pi/2)$.

b. Show that the iterates $x_{i+1} = 2\sin x_i$, $i = 0, 1, \ldots$, converge to the root in $(\pi/2, 2)$ for any x_0 in this interval.

c. Apply the Newton iteration to this equation and ascertain for what starting values the iterates will converge to the root in $(\pi/2, 2)$. Compare the rate of convergence of the Newton iterates with those of part **b**.

5.2.7. Let n be a positive integer and α a positive number. Show that Newton's method for the equation $x^n - \alpha = 0$ is

$$x_{k+1} = \frac{1}{n}\left[(n-1)x_k + \frac{\alpha}{x_k^{n-1}}\right], \qquad k = 0, 1, \cdots,$$

and that this Newton sequence converges for any $x_0 > 0$. Discuss the case $n = 2$.

5.2.8. Ascertain whether the following statements are true or false and prove your assertions:

a. Let $\{x_k\}$ be a sequence of Newton iterates for a continuously differentiable function f. If for some i, $|f(x_i)| \leq 0.01$ and $|x_{i+1} = x_i| \leq 0.01$, then x_{i+1} is within 0.01 of a root of $f(x) = 0$.

b. The Newton iterates converge to the unique solution of $x^2 - 2x + 1 = 0$ for any $x_0 \neq 1$. (Ignore rounding error.)

5.2.9. Consider the equation $x^2 - 2x + 2 = 0$. What is the behavior of the Newton iterates for various real starting values?

5.2.10. Show that the Newton iterates converge to the unique solution of $e^{2x} + 3x + 2 = 0$ for any starting value x_0.

5.2.11. Assume that f is differentiable, convex, and $f'(x) < 0$ for all x. If $f(x) = 0$ has a solution x^*, show that x^* is unique and that the Newton iterates converge monotonically upward to x^* if $x_0 < x^*$. What happens if $x_0 > x^*$?

5.2.12. Show that the Newton iterates for $f(x) \equiv x^p$ converge to the solution $x^* = 0$ only linearly with an asymptotic convergence factor of $(p-1)/p$.

5.2.13. Newton's method can be used for determining the reciprocal of numbers when division is not available.

a. Show how Newton's method can be applied to the equation

$$f(x) = \frac{1}{x} - a,$$

without using division.

b. Give an equation for the error term, $e_k = x_k - a^{-1}$, and show that the convergence is quadratic.

c. Give conditions on the initial approximation so that $x_k \to a^{-1}$ as $k \to \infty$. If $0 < a < 1$, give a numerical value of x_0 which will guarantee convergence.

5.3 Systems of Nonlinear Equations

We mentioned in Section 5.1 that the shooting method applied to a system of ordinary differential equations can lead to the problem of solving a system of nonlinear algebraic equations in order to supply the missing initial conditions. As we shall see later, the application of the finite difference method to nonlinear boundary-value problems also leads to nonlinear systems of equations. Therefore we consider in this section how the methods discussed in the previous section for a single equation can be extended to systems of equations.

The problem is to obtain an approximate solution to the system of equations

$$f_i(x_1, x_2, \ldots, x_n) = 0, \qquad i = 1, \ldots, n, \tag{5.3.1}$$

where $f_1, f_2, \ldots, f_n$ are given functions of the n variables $x_1, \ldots, x_n$. We shall usually use vector notation and write (5.3.1) as

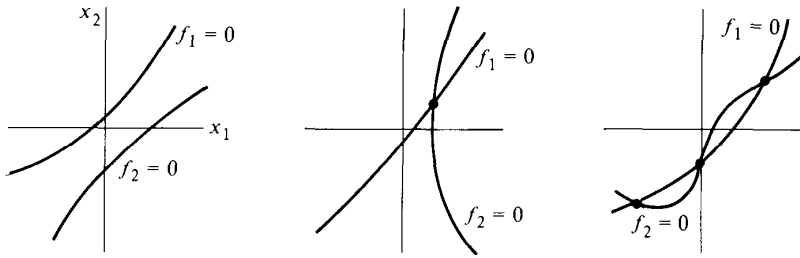
$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \tag{5.3.2}$$

where, as usual, $\mathbf{x}$ is the vector with components $x_1, \ldots, x_n$, and $\mathbf{F}$ is the vector function with components $f_1, \ldots, f_n$. The special case of solving (5.3.1) when $n = 1$ is just the problem of finding roots of a single equation that was considered in the previous section. On the other hand, the special case in which

$$\mathbf{F}(\mathbf{x}) \equiv A\mathbf{x} - \mathbf{b},$$

where A is a given matrix and $\mathbf{b}$ a given vector, is that of solving a system of linear equations, which was treated in Chapter 4.

The problem of ascertaining when (5.3.2) has solutions, and how many, is generally very difficult. In the relatively simple case $n = 2$, it is easy to see the various possibilities geometrically, at least in principle. For example, if we plot in the x_1, x_2 plane the set of points for which $f_1(x_1, x_2) = 0$, and then the set of points for which $f_2(x_1, x_2) = 0$, the intersection of these sets is precisely the set of solutions of (5.3.2). (Here, and henceforth, we are restricting our attention to only real solutions.) Figure 5.11 illustrates a few possible situations. Later



(a) No solutions (b) A unique solution (c) Several solutions

Figure 5.11: *Possible Solutions for* $n = 2$

we shall assume that (5.3.2) has a solution $\mathbf{x}^*$ that is the one of interest to us, although the system may have additional solutions.

Picard Iterations

In many situations the system (5.3.2) has the form

$$\mathbf{F}(\mathbf{x}) \equiv A\mathbf{x} + \mathbf{H}(\mathbf{x}) = 0, \tag{5.3.3}$$

where A is a given nonsingular matrix and $\mathbf{H}$ is a given vector of nonlinear functions. In this case a somewhat natural (although not necessarily good) iterative procedure is

$$\mathbf{x}^{i+1} = -A^{-1}\mathbf{H}(\mathbf{x}^i), \qquad i = 0, 1, \ldots, \tag{5.3.4}$$

where the superscript indicates iteration number. Here, as well as later, we mean by (5.3.4) that at each step of the iteration the linear system of equations

$$A\mathbf{x}^{i+1} = -\mathbf{H}(\mathbf{x}^i)$$

is to be solved to obtain the next iterate. The iteration (5.3.4) is known as a *Picard iteration.* It may be considered a special case of the extension of the chord method of the previous section to n equations; this would take the form

$$\mathbf{x}^{i+1} = \mathbf{x}^i - B\mathbf{F}(\mathbf{x}^i), \qquad i = 0, 1, \ldots, \tag{5.3.5}$$

for a given nonsingular matrix B. It is easy to see (Exercise 5.3.2) that (5.3.5) reduces to (5.3.4) if $\mathbf{F}$ is of the form (5.3.3) and $B = A^{-1}$.

Convergence

When will the iterations (5.3.5) or (5.3.4) converge? The situation is precisely analogous to that in the scalar case but complicated by the need to work with vector-valued functions. Consider the general one-step iteration

$$\mathbf{x}^{i+1} = \mathbf{G}(\mathbf{x}^i), \qquad i = 0, 1, \ldots , \tag{5.3.6}$$

where $\mathbf{G}$ is a given iteration function; for example, for (5.3.5)

$$\mathbf{G}(\mathbf{x}) \equiv \mathbf{x} - B\mathbf{F}(\mathbf{x}). \tag{5.3.7}$$

We shall assume that the solution $\mathbf{x}^\star$ of $\mathbf{F}(\mathbf{x}) = 0$ satisfies $\mathbf{x}^\star = \mathbf{G}(\mathbf{x}^\star)$ and, conversely, that if $\mathbf{x}^\star = \mathbf{G}(\mathbf{x}^\star)$, then $\mathbf{F}(\mathbf{x}^\star) = 0$; it is clear that this is the case for (5.3.7) if B is nonsingular.

In the previous section, the convergence theory was based on $|g'(x)| < 1$ in a neighborhood of the solution. For systems of equations the corresponding result is the following. If

$$||\mathbf{G}'(\mathbf{x})|| \leq \gamma < 1, \qquad \text{for } ||\mathbf{x} - \mathbf{x}^\star|| \leq \beta, \tag{5.3.8}$$

then the iterates (5.3.6) converge if $||\mathbf{x}^0 - \mathbf{x}^\star|| \leq \beta$ (or if $||\mathbf{x}^k - \mathbf{x}^\star|| \leq \beta$ for any k). Here, as in Section 4.4, $||\ ||$ denotes a vector norm or the corresponding matrix norm, and $\mathbf{G}'(\mathbf{x})$ is the Jacobian matrix (see Appendix 1) of $\mathbf{G}$ evaluated at $\mathbf{x}$. We shall not prove this convergence statement but only note that it can be rather easily proven after a proper extension of the mean-value theorem to n dimensions.

If we apply the criterion (5.3.8) to the iteration (5.3.5), we obtain (see Exercise 5.3.4)

$$||I - B\mathbf{F}'(\mathbf{x})|| \leq \gamma < 1 \qquad \text{for } ||\mathbf{x} - \mathbf{x}^\star|| < \beta \tag{5.3.9}$$

and, in particular, for the iteration (5.3.4),

$$||A^{-1}\mathbf{H}'(\mathbf{x})|| \leq \gamma < 1 \qquad \text{for } ||\mathbf{x} - \mathbf{x}^\star|| \leq \beta. \tag{5.3.10}$$

Intuitively, (5.3.10) says that the iteration (5.3.4) will converge provided that $A^{-1}\mathbf{H}'(\mathbf{x})$ is "small" when $\mathbf{x}$ is close to $\mathbf{x}^\star$. Similarly, the iteration (5.3.5) will converge if $B\mathbf{F}'(\mathbf{x})$ is close to the identity, or, equivalently, if B^{-1} is close to $\mathbf{F}'(\mathbf{x})$. Since $\mathbf{x}^\star$ is not known, these criteria are not meant to be used to check whether a given iteration will converge, but rather to give some insight as to what factors govern the convergence.

Newton's Method

Analogously to the previous section, the size of $||\mathbf{G}'(\mathbf{x})||$ will tend to determine the rate of convergence, and we would like this quantity to be as small as

possible, at least near the solution $\mathbf{x}^\star$. Suppose that for the iteration (5.3.5) we could choose $B = [\mathbf{F}'(\mathbf{x}^\star)]^{-1}$; then $\mathbf{G}'(\mathbf{x}^\star) = 0$, and the rate of convergence will be rapid near the solution. Of course, this choice of B is essentially impossible since $\mathbf{x}^\star$ is not known, but we can achieve this effect by the following *Newton iteration:*

$$\mathbf{x}^{i+1} = \mathbf{x}^i - [\mathbf{F}'(\mathbf{x}^i)]^{-1}\mathbf{F}(\mathbf{x}^i), \qquad i = 0, 1, \ldots . \tag{5.3.11}$$

Here, we are assuming, of course, that the matrices $\mathbf{F}'(\mathbf{x}^i)$ are nonsingular, and we would carry out (5.3.11) by the following steps:

1. Solve the linear system $\mathbf{F}'(\mathbf{x}^i)\mathbf{y}^i = -\mathbf{F}(\mathbf{x}^i)$.
2. Set $\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{y}^i$. (5.3.12)

The iteration (5.3.12) can be derived as follows. We approximate the functions f_i at $\mathbf{x}^k$ by a first-order Taylor expansion:

$$f_i(\mathbf{x}) \doteq l_i(\mathbf{x}) \equiv f_i(\mathbf{x}^k) + f_i'(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k), \quad i = 1, \cdots, n. \tag{5.3.13}$$

Here, $f_i'(\mathbf{x}^k)$ is the ith row of the Jacobian matrix $\mathbf{F}'(\mathbf{x}^k)$ and (5.3.13) can be written as

$$\mathbf{F}(\mathbf{x}) \doteq L(\mathbf{x}) \equiv \mathbf{F}(\mathbf{x}^k) + \mathbf{F}'(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k).$$

The solution of the linear system $L(\mathbf{x}) = 0$ then gives the next Newton iterate $\mathbf{x}^{k+1}$. Geometrically, $l_i(\mathbf{x}) = 0$ is the equation of the "hyperplane" tangent to f_i at $\mathbf{x}^k$, and $\mathbf{x}^{k+1}$ is the intersection of the n sets $\{\mathbf{x} : l_i(\mathbf{x}) = 0\}$. This generalizes to n dimensions the property of Newton's method in a single variable that the next Newton iterate is the intersection of the x-axis with the tangent line to f at x_i.

Clearly, the iteration (5.3.11) reduces to Newton's method of the previous section if $n = 1$. We would hope that (5.3.11) retains the basic property of quadratic convergence. This is true, and we state the following result without proof.

THEOREM 5.3.1 (Newton Convergence) *If* $\mathbf{F}$ *is two times continuously differentiable in a neighborhood of* $\mathbf{x}^\star$, *and if* $\mathbf{F}'(\mathbf{x}^\star)$ *is nonsingular, then the iterates (5.3.11) will converge to* $\mathbf{x}^\star$ *provided that* $\mathbf{x}^0$ *is sufficiently close to* $\mathbf{x}^\star$ *(local convergence theorem), and they will have the property of quadratic convergence:*

$$\|\mathbf{x}^{i+1} - \mathbf{x}^\star\| \leq c\|\mathbf{x}^i - \mathbf{x}^\star\|^2. \tag{5.3.14}$$

As an example of Newton's method (5.3.12), we give in Table 5.2 the first four iterations for the system of nonlinear equations

$$x_1^2 + x_2^2 - 1 = 0, \qquad x_1^2 - x_2 = 0, \tag{5.3.15}$$

Table 5.2: *Convergence of Newton's Method for* (5.3.15)

Iteration	x_1	x_2	Number of Correct Digits
0	0.5	0.5	0,0
1	0.87499999	0.62499999	0,1
2	0.79067460	0.61805555	1,4
3	0.78616432	0.61803399	4,8
4	0.78615138	0.61803399	8,8

using the starting values $x_1 = 0.5$ and $x_2 = 0.5$. Note that we can observe the approximate quadratic convergence by the number of correct digits in the iterates.

The quadratic convergence property (5.3.14) (which is lost if $\mathbf{F}'(\mathbf{x}^*)$ is singular) is highly desirable and makes Newton's method of central importance in the solution of nonlinear systems of equations. But there are three obstacles to its successful use. The first is the need to compute the Jacobian matrix at each step, and this requires evaluation of the n^2 partial derivatives $\partial f_i/\partial x_j$. If n is large and/or the functions f_i are complicated, it can be drudgery to work out by hand – and then convert to computer code – the expressions for these derivatives; this can sometimes be mitigated by the use of symbolic differentiation techniques, as discussed in Chapter 1. Another commonly used approach is to approximate the partial derivatives by finite differences; for example,

$$\frac{\partial f_i}{\partial x_j}(\mathbf{x}) \doteq \frac{1}{h}[f_i(x_1, \ldots, x_{j-1}, x_j + h, x_{j+1}, \ldots, x_n) - f_i(\mathbf{x})]. \tag{5.3.16}$$

This has the advantage of requiring only the expressions for the f_i, which are needed in any case. But the actual numerical evaluation of the Jacobian matrix, either by expressions for the partial derivatives or by approximations such as (5.3.16), can be costly in computer time. This leads to a frequently used modification of Newton's method in which the Jacobian matrix is reevaluated only periodically rather than at each iteration. For example, the iteration might be:

1. Evaluate $\mathbf{F}'(\mathbf{x}^0)$.
2. Compute $\mathbf{x}^{i+1} = \mathbf{x}^i - [\mathbf{F}'(\mathbf{x}^0)]^{-1}\mathbf{F}(\mathbf{x}^i), \quad i = 0, 1, \ldots, k.$
3. Evaluate $\mathbf{F}'(\mathbf{x}^{k+1})$. (5.3.17)
4. Compute $\mathbf{x}^{i+1} = \mathbf{x}^i - [\mathbf{F}'(\mathbf{x}^{k+1})]^{-1}\mathbf{F}(\mathbf{x}^i), \quad i = k+1, \ldots, 2k.$

The modified Newton iteration (5.3.17) can also be useful in alleviating the second disadvantage of Newton's method: the need to solve a system of linear equations at each step. The advantage of (5.3.17) in this regard is that the actual implementation involves, of course, solving a number of linear systems (as in step 2),

$$\mathbf{F}'(\mathbf{x}^0)\mathbf{y}^i = -\mathbf{F}(\mathbf{x}^i), \qquad i = 0, 1, \ldots, k,$$

where the coefficient matrix $\mathbf{F}'(\mathbf{x}^0)$ is the same. Hence, as discussed in Section 4.2, the LU factors of $\mathbf{F}'(\mathbf{x}^0)$ from the Gaussian elimination process can be retained and used for all $k+1$ right-hand sides.

The third – and most troublesome – difficulty with Newton's method is that the iterates may not converge from a given starting approximation $\mathbf{x}^0$; the local convergence theorem only insures convergence once $\mathbf{x}^0$ (or some other iterate) is "sufficiently close" to $\mathbf{x}^\star$. One remedy for this difficulty is to obtain the best possible first approximation using any physical or other knowledge about the problem. However, this is not always sufficient. An approach that often – but certainly not always – works is the *continuation method*, which we describe briefly in the Supplementary Discussion.

Nonlinear Boundary Value Problems

We discuss now the extension to nonlinear two-point boundary-value problems of the finite difference method presented in Chapter 3. We shall consider the equation

$$v'' = g(x, v), \qquad 0 \le x \le 1, \tag{5.3.18}$$

with the boundary conditions

$$v(0) = \alpha, \qquad v(1) = \beta. \tag{5.3.19}$$

Here g is a given function of two variables, and α and β are given constants.

We proceed exactly as in Section 3.1. The interval $[0, 1]$ is partitioned by grid points

$$0 = x_0 < x_1 < \cdots < x_n < x_{n+1} = 1$$

with spacing h. At each interior grid point x_i we approximate the second derivative by central differences and use these approximations in (5.3.18). This leads to the system of equations (corresponding to (3.1.8))

$$-v_{i+1} + 2v_i - v_{i-1} + h^2 g(x_i, v_i) = 0, \qquad i = 1, \ldots, n, \tag{5.3.20}$$

where $v_0 = \alpha$ and $v_{n+1} = \beta$ are known by the boundary conditions (5.3.18). This is a system of n equations in the n unknowns $v_1, \ldots, v_n$ and is nonlinear if the function g is nonlinear in v. A solution $v_1^\star, \ldots, v_n^\star$ of (5.3.19), if it exists, is an approximation to the corresponding solution v of (5.3.18) at the grid points

We can write the system (5.3.19) in matrix-vector form as

$$\mathbf{F}(\mathbf{v}) \equiv A\mathbf{v} + \mathbf{H}(\mathbf{v}) = 0, \tag{5.3.21}$$

where $\mathbf{v}$ is the vector with components $v_1, \ldots, v_n$, A is the $(2, -1)$ tridiagonal matrix of (3.1.10) and

$$\mathbf{H}(\mathbf{v}) = h^2 \begin{bmatrix} g(x_1, v_1) \\ \\ \vdots \\ \\ g(x_n, v_n) \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix}. \tag{5.3.22}$$

As an example, consider the problem

$$v''(x) = 3v(x) + x^2 + 10[v(x)]^3, \qquad 0 \le x \le 1, \qquad v(0) = v(1) = 0. \tag{5.3.23}$$

Here

$$g(x, v) = 3v + x^2 + 10v^3, \tag{5.3.24}$$

and with $h = 1/(n+1)$ and

$$x_i = ih, \qquad i = 0, 1, \ldots, n+1, \tag{5.3.25}$$

the difference equations (5.3.20) are

$$-v_{i+1} + 2v_i - v_{i-1} + h^2(3v_i + i^2h^2 + 10v_i^3) = 0, \qquad i = 1, \ldots, n, \tag{5.3.26}$$

where, from the boundary conditions, $v_0 = v_{n+1} = 0$. Hence the ith component of the function $\mathbf{H}(\mathbf{v})$ of (5.3.22) is $h^2(3v_i + i^2h^2 + 10v_i^3)$.

We now consider some numerical methods for the system (5.3.21). The Picard iteration discussed earlier is

$$A\mathbf{v}^{k+1} = -\mathbf{H}(\mathbf{v}^k). \tag{5.3.27}$$

The time to carry out one of these iterative steps depends almost entirely on the complexity of $\mathbf{H}$ since the solution of tridiagonal linear systems is very rapid, as we saw in Section 3.2. Moreover, in this case the LU decomposition of A can be done once and for all. Whether the iteration (5.3.26) even converges, however, will depend upon the properties of H.

Next consider Newton's method for (5.3.21). The Jacobian matrix will be (see Exercise 5.3.8)

$$\mathbf{F}'(\mathbf{v}) = A + \mathbf{H}'(\mathbf{v}). \tag{5.3.28}$$

Since the ith component, $H_i(\mathbf{v}) = h^2 g(x_i, v_i)$, of $\mathbf{H}$ depends only on v_i, we have that $\dfrac{\partial H_i}{\partial v_j} = 0$, $j \neq i$. Thus the matrix $\mathbf{H}'(\mathbf{v})$ is diagonal and $\mathbf{F}'(\mathbf{v})$ is tridiagonal with a typical row given by

$$-1 \qquad 2 + h^2 \frac{\partial g}{\partial v}(x_i, v_i) \qquad -1.$$

The Newton iteration is then

$$\begin{aligned} &1.\ \text{Solve } [A + \mathbf{H}'(\mathbf{v}^k)]\mathbf{y}^k = -[A\mathbf{v}^k + \mathbf{H}(\mathbf{v}^k)], \\ &2.\ \text{Set } \mathbf{v}^{k+1} = \mathbf{v}^k + \mathbf{y}^k, \end{aligned} \tag{5.3.29}$$

so that at each iteration a tridiagonal linear system is to be solved. If the function g is complicated, a major portion of the work of each Newton iteration will be the evaluation of $\mathbf{H}(\mathbf{v}^k)$ and $\mathbf{H}'(\mathbf{v}^k)$.

For the boundary-value problem (5.3.23) g is given by (5.3.24), so that

$$\frac{\partial g}{\partial v}(x, v) = 3 + 30v^2,$$

and the ith diagonal element of the Jacobian matrix (5.3.28) is $2+h^2(3+30v_i^2)$. Since the $(2, -1)$ tridiagonal matrix A is diagonally dominant, it is clear that the addition of the positive terms $h^2(3 + 30v_i^2)$ to the diagonal only enhances the diagonal dominance. More generally, whenever (Exercise 5.3.11)

$$\frac{\partial g}{\partial v}(x, v) \geq 0, \qquad 0 \leq x \leq 1, \qquad -\infty < v < \infty, \tag{5.3.30}$$

and A is the $(2, -1)$ matrix, then

$$A + \mathbf{H}'(\mathbf{v}) \text{ is diagonally dominant,} \tag{5.3.31}$$

$$A + \mathbf{H}'(\mathbf{v}) \text{ is symmetric positive-definite.} \tag{5.3.32}$$

As we saw in Section 4.3, either of these properties is sufficient to ensure that the solution of the tridiagonal systems (5.3.29) of Newton's method can be carried out by Gaussian elimination without any need for interchanging rows to preserve numerical stability.

It is also true (but beyond the scope of this book to prove) that either of the conditions (5.3.31) or (5.3.32) ensures that the system (5.3.21) has a unique solution. On the other hand, if (5.3.30) does not hold the differential equation (5.3.18) need not have a unique solution, and this will be reflected in the discrete system (5.3.21). For example, if $g(x, v) = v^4$ there will be two solutions of both (5.3.18) and (5.3.21). This is explored further in Exercise 5.3.13.

For the difference equations (5.3.26) we tabulate in Table 5.3 the results of Newton's method at the grid points $0.1, 0.2, \ldots, 0.9$ for $h = 0.1$, 0.01, and 0.001 ($n = 9$, 99, and 999). In all cases the initial approximation for Newton's method was taken to be $\mathbf{v}^0 = 0$, and the iteration terminated when all components of the Newton correction vector $\mathbf{y}^k$ of (5.3.28) were less than 10^{-6} in magnitude.

Table 5.3: *Newton's Method for the Difference Equations* (5.3.26)

x	$h = 0.1$	$h = 0.01$	$h = 0.001$
0.1	-0.0058	-0.0058	-0.0058
0.2	-0.0116	-0.0118	-0.0118
0.3	-0.0174	-0.0176	-0.0176
0.4	-0.0223	-0.0230	-0.0230
0.5	-0.0274	-0.0276	-0.0276
0.6	-0.0302	-0.0304	-0.0304
0.7	-0.0303	-0.0305	-0.0305
0.8	-0.0265	-0.0266	-0.0266
0.9	-0.0170	-0.0171	-0.0171

Supplementary Discussion and References: 5.3

For a more detailed discussion and analysis of a variety of methods for solving systems of nonlinear equations numerically, see Ortega and Rheinboldt [1970] and Dennis and Schnabel [1983]. In particular, these references contain various discrete forms of Newton's method where the Jacobian matrix is approximated in some fashion. Certain of these approximations lead to natural generalizations of the secant method to systems of equations, and others give what are known as *quasi-Newton* methods, which are among the most promising methods for nonlinear systems. For a review of quasi-Newton methods, see also Dennis and Moré [1977].

An attractive alternative to symbolic differentiation or approximation of the partial derivatives in the Jacobian matrix by finite differences is *automatic differentiation*. See Griewank [1989] for a review and Griewank [1990] for application to Newton's method.

Many systems of equations arise in the attempt to minimize (or maximize) a function g of n variables. From the calculus we know that if g is continuously differentiable, then a necessary condition for a local minimum is that the gradient vector vanishes:

$$\left(\frac{\partial g}{\partial x_1}, \ldots, \frac{\partial g}{\partial x_n} \right) = 0.$$

Thus by solving this system of equations, one obtains a possible local minimizer of g, and in many situations it will be known that this vector must indeed minimize g. Alternatively, if we are given an arbitrary system of equations $f_i(\mathbf{x}) = 0$, $i = 1, \ldots, n$, we can convert the solution of this system to a minimization problem by defining a function

$$g(\mathbf{x}) = \sum_{i=1}^{n} [f_i(\mathbf{x})]^2.$$

Clearly, g takes on a minimum value of zero only when all $f_i(\mathbf{x})$ are zero. This conversion, however, is usually not recommended for obtaining a numerical solution of the system since the ill-conditioning of the problem will be increased.

In many problems the equations are to be solved for various values of one or more parameters. Suppose there is a single parameter α and we write the system of equations as

$$\mathbf{F}(\mathbf{x}; \alpha) = \mathbf{0}. \tag{5.3.33}$$

Assume that we wish solutions $\mathbf{x}_0^\star, \cdots, \mathbf{x}_N^\star$ for values $\alpha_0 < \alpha_1 < \cdots < \alpha_N$, where α_0 corresponds to a trivial, or at least an easy, problem; for example, the equations for α_0 may be linear. If $\mathbf{x}_0^\star$ can be computed and if $|\alpha_1 - \alpha_0|$ is small, then we hope that $\mathbf{x}_0^\star$ is sufficiently close to $\mathbf{x}_1^\star$ so that $\mathbf{x}_0^\star$ is a suitable starting approximation for the equation $\mathbf{F}(\mathbf{x}; \alpha_1) = 0$. Continuing in this way, we use each previous solution as a starting approximation for the next problem. This is called the *continuation method.*

If the equations to be solved do not contain a parameter, we can always introduce one artificially. For example, let $\mathbf{F}(\mathbf{x}) = 0$ be the system and let $\mathbf{x}^0$ be our best approximation to the solution (but not good enough for the Newton iteration to converge). Define a new set of equations depending on a parameter α by

$$\hat{\mathbf{F}}(\mathbf{x}; \alpha) = \mathbf{F}(\mathbf{x}) + (\alpha - 1)\mathbf{F}(\mathbf{x}^0) = 0, \qquad 0 \le \alpha \le 1. \tag{5.3.34}$$

Then $\hat{\mathbf{F}}(\mathbf{x}; 0) = \mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x}^0) = 0$, for which $\mathbf{x}^0$ is a solution, and $\hat{\mathbf{F}}(\mathbf{x}; 1) = \mathbf{F}(\mathbf{x}) = 0$, which are the equations to be solved. Hence, we proceed as in the previous paragraph for parameters $0 = \alpha_0 < \alpha_1 < \cdots < \alpha_N = 1$.

The continuation method is closely related to *Davidenko's method.* Consider (5.3.33) and assume that for each $\alpha \in [0, 1]$ the equation defines a solution $\mathbf{x}(\alpha)$ that is continuously differentiable in α. Then if we differentiate

$$\mathbf{F}(\mathbf{x}(\alpha)) + (\alpha - 1)\mathbf{F}(\mathbf{x}^0) = 0$$

with respect to α, we obtain by the chain rule

$$\mathbf{F}'(\mathbf{x}(\alpha))\mathbf{x}'(\alpha) + \mathbf{F}(\mathbf{x}^0) = 0,$$

or, assuming that the Jacobian matrix $\mathbf{F}'(\mathbf{x}(\alpha))$ is nonsingular,

$$\mathbf{x}'(\alpha) = -[\mathbf{F}'(\mathbf{x}(\alpha))]^{-1}\mathbf{F}(\mathbf{x}^0),$$

with the initial condition $\mathbf{x}(0) = \mathbf{x}^0$. The solution $\mathbf{x}(\alpha)$ of this initial-value problem at $\alpha = 1$ will, we hope, be the desired solution of the original system of equations $\mathbf{F}(\mathbf{x}) = 0$. In practice we will have to solve the differential equations numerically, and we can, in principle, use any of the methods of Chapter 2. Although Davidenko's method and the continuation method are attractive possibilities, their reliability in practice has been less than desired. In particular, it is possible that the Jacobian matrix will become singular for some $\mathbf{x}(\alpha)$ with $\alpha < 1$, or even that the solution curve itself will blow up prematurely. For a review of possible ways of overcoming some of these difficulties, see Allgower and Georg [1990].

The proof that the system (5.3.21) has a unique solution under the conditions (5.3.31) or (5.3.32) can be found, for example, in Ortega and Rheinboldt [1970, Section 4.4].

EXERCISES 5.3

5.3.1. Show graphically that the system of equations $x_1^2 + x_2^2 = 1$, $x_1^2 - x_2 = 0$ has precisely two solutions.

5.3.2. Show that (5.3.5) reduces to (5.3.4) when $\mathbf{F}$ is of the form (5.3.3) and $B = A^{-1}$.

5.3.3. Compute the Jacobian matrix $\mathbf{G}'(\mathbf{x})$ for

$$\mathbf{G}(\mathbf{x}) = \begin{bmatrix} x_1^2 + x_1x_2x_3 + x_3^3 \\ x_1^3x_2 + x_2x_3^2 \\ x_1/x_2^3 \end{bmatrix}.$$

5.3.4. If $\mathbf{G}(\mathbf{x}) = \mathbf{x} - B\mathbf{F}(\mathbf{x})$, show that $\mathbf{G}'(\mathbf{x}) = I - B\mathbf{F}'(\mathbf{x})$ and conclude that (5.3.9) and (5.3.10) follow from (5.3.8).

5.3.5. For the functions of Exercise 5.3.1, compute the tangent planes at $x_1 = 2$, $x_2 = 2$.

5.3.6. Give the Newton iteration for the equations of Exercise 5.3.1. For what points $\mathbf{x}$ is the Jacobian matrix nonsingular?

5.3.7. Write a program for Newton's method to solve n equations in n unknowns. Use Gaussian elimination with partial pivoting to solve the linear equations.

5.3.8. If $\mathbf{F}(\mathbf{v}) = A\mathbf{v} + \mathbf{H}(\mathbf{v})$ for some matrix A, verify that $\mathbf{F}'(\mathbf{v}) = A + \mathbf{H}'(\mathbf{v})$. Apply this to obtaining the Jacobian matrices used in (5.3.9) and (5.3.10).

5.3.9. Write out the difference equations (5.3.20) and the corresponding Jacobian matrices for:

a. $g(x, v) = v + v^2$

b. $g(x, v) = xv^3$

5.3.10 Write out the Newton iteration (5.3.29) explicitly for the difference equations (5.3.20) with g given by Exercise 5.3.9.

5.3.11. Let D be a diagonal matrix with non-negative elements.

a. If A is symmetric positive definite, show that $A + D$ is positive definite.

b. If A is diagonally dominant and has positive diagonal elements, show that $A + D$ is diagonally dominant.

c. Apply parts **a.** and **b.** to show that (5.3.31) and (5.3.32) follow from (5.3.30).

5.3.12. Consider the two-point boundary-value problem $v'' = e^v + 2 - e^{x^2}$, $0 \le x \le 1$, $v(0) = 0$, $v(1) = 1$.

a. Write the finite difference equations for this problem in matrix-vector form for $h = 0.01$.

b. Discuss in detail how you would solve the system of equations in part **a** on a computer. The discussion should include a clear description of the method, what, if any, problems you expect the method to have, how much computer time you would expect the method to use, and so on.

5.3.13. Consider the two-point boundary-value problem $v'' = v^4$, $0 \le x \le 1$, $v(0) = 1$, $v(1) = \frac{1}{2}$.

a. Find an approximation to a solution by a third degree polynomial obtained by using the data $v(0)$, $v(1)$, $v''(0)$, $v''(1)$.

b. Obtain an approximate solution by the shooting method using both bisection and a chord method for the resulting single nonlinear equation.

c. Use the finite difference method and obtain a solution to the discrete system by the Picard method and Newton's method. For an initial approximation for these iterative methods, use the approximate solution of part **a**.

d. As mentioned in the text, a boundary value problem need not have a unique solution. Can you find a second approximate solution for this problem?

5.3.14. Consider the initial value problem

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0,$$

and the backward Euler method

$$\mathbf{y}_{k+1} = \mathbf{y}_k + h\mathbf{f}(x_{k+1}, \mathbf{y}_{k+1}), \quad k = 0, 1, \ldots.$$

a. Discuss Newton's method applied to the backward Euler system for $\mathbf{y}_{k+1}$. How might you obtain an initial approximation for Newton's method?

b. Describe the Picard iteration for this system, and give conditions under which the Picard iterates will converge.