2

Nonlinear Problems in One Variable

We begin our study of the solution of nonlinear problems by discussing problems in just one variable: finding the solution of one nonlinear equation in one unknown, and finding the minimum of a function of one variable. The reason for studying one-variable problems separately is that they allow us to see those principles for constructing good local, global, and derivative-approximating algorithms that will also be the basis of our algorithms for multivariable problems, without requiring knowledge of linear algebra or multivariable calculus. The algorithms for multivariable problems will be more complex than those in this chapter, but an understanding of the basic approach here should help in the multivariable case.

Some references that consider the problems of this chapter in detail are Avriel (1976), Brent (1973), Conte and de Boor (1980), and Dahlquist, Björck, and Anderson (1974).

2.1 WHAT IS NOT POSSIBLE

Consider the problem of finding the real roots of each of the following three nonlinear equations in one unknown:

$$f_1(x) = x^4 - 12x^3 + 47x^2 - 60x;$$

$$f_2(x) = x^4 - 12x^3 + 47x^2 - 60x + 24,$$

$$f_3(x) = x^4 - 12x^3 + 47x^2 - 60x + 24.1.$$

(see Figure 2.1.1). It would be wonderful if we had a general-purpose computer routine that would tell us: "The roots of $f_1(x)$ are $x = 0, 3, 4$, and 5; the real roots of $f_2(x)$ are $x = 1$ and $x \cong 0.888$; $f_3(x)$ has no real roots."

It is unlikely that there will ever be such a routine. In general, the questions of existence and uniqueness—does a given problem have a solution, and is it unique?—are beyond the capabilities one can expect of algorithms that solve nonlinear problems. In fact, we must readily admit that for any computer algorithm there exist nonlinear functions (infinitely continuously differentiable, if you wish) perverse enough to defeat the algorithm. Therefore, all a user can be guaranteed from any algorithm applied to a nonlinear problem is the answer, "An approximate solution to the problem is ______________," or, "No approximate solution to the problem was found in the alloted time." In many cases, however, the supplier of a nonlinear problem knows from practical considerations that it has a solution, and either that the solution is unique or that a solution in a particular region is desired. Thus the inability to determine the existence or uniqueness of solutions is usually not the primary concern in practice.

It is also apparent that one will be able to find only approximate solutions to most nonlinear problems. This is due not only to the finite precision of our computers, but also to the classical result of Galois that for some polynomials of degree $n \geq 5$, no closed-form solutions can be found using integers and the operations $+$, $-$, $\times$, $\div$, exponentiation, and second through nth roots. Therefore, we will develop methods that try to find one approximate solution of a nonlinear problem.

2.2 NEWTON'S METHOD FOR SOLVING ONE EQUATION IN ONE UNKNOWN

Our consideration of finding a root of one equation in one unknown begins with Newton's method, which is the prototype of the algorithms we will generate. Suppose we wish to calculate the square root of 3 to a reasonable number of places. This can be viewed as finding an approximate root x_* of the func-

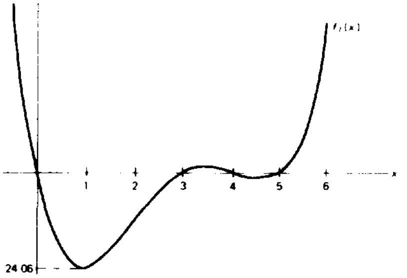

Figure 2.1.1 The equation $f_1(x) = x^4 - 12x^3 + 47x^2 - 60x$

tion $f(x) = x^2 - 3$ (see Figure 2.2.1). If our initial or current estimate of the answer is $x_c = 2$, we can get a better estimate x_+ by drawing the line that is tangent to $f(x)$ at $(2, f(2)) = (2, 1)$, and finding the point x_+ where this line crosses the x axis. Since

$$x_+ = x_c - \Delta x,$$

and

$$f'(x_c) = \frac{\Delta y}{\Delta x} = \frac{f(x_c)}{\Delta x},$$

we have that

$$f'(x_c)\Delta x = \Delta y = f(x_c)$$

or

$$x_+ = x_c - \frac{f(x_c)}{f'(x_c)} \qquad (2.2.1)$$

which gives

$$x_+ = 2 - \tfrac{1}{4} = 1.75.$$

The logical thing to do next is to apply the same process from the new current estimate $x_c = 1.75$. Using (2.2.1) gives $x_+ = 1.75 - (0.0625/3.5) = 1.732\frac{1}{7}$, which already has four correct digits of $\sqrt{3}$. One more iteration gives $x_+ \cong 1.7320508$, which has eight correct digits.

The method we have just developed is called the *Newton-Raphson method* or *Newton's method.* It is important to our understanding to take a more abstract view of what we have done. At each iteration we have constructed a *local model* of our function $f(x)$ and solved for the root of the model. In

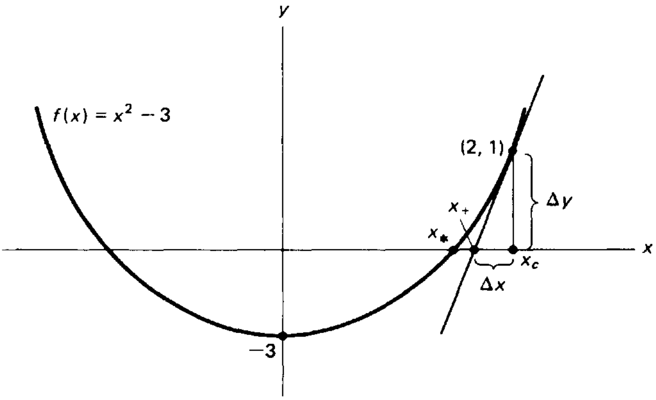

Figure 2.2.1 An iteration of Newton's method on $f(x) = x^2 - 3$ (not to scale)

the present case, our model

$$M_c(x) = f(x_c) + f'(x_c)(x - x_c) \qquad (2.2.2)$$

is just the unique line with function value $f(x_c)$ and slope $f'(x_c)$ at the point x_c. [We use capital M to be consistent with the multidimensional case and to differentiate from minimization problems where our model is denoted by $m_c(x)$.] It is easy to verify that $M_c(x)$ crosses the x axis at the point x_+ defined by (2.2.1).

Pedagogical tradition calls for us to say that we have obtained Newton's method by writing $f(x)$ as its Taylor series approximation around the current estimate x_c,

$$\begin{aligned} f(x) &= f(x_c) + f'(x_c)(x - x_c) + \frac{f''(x_c)(x - x_c)^2}{2!} + \cdots \\ &= \sum_{i=0}^{\infty} \frac{f^i(x_c)(x - x_c)^i}{i!}, \end{aligned} \qquad (2.2.3)$$

and then approximating $f(x)$ by the affine* portion of this series, which naturally is given also by (2.2.2). Again the root is given by (2.2.1). There are several reasons why we prefer a different approach. It is unappealing and unnecessary to make assumptions about derivatives of any higher order than those actually used in the iteration. Furthermore, when we consider multivariable problems, higher-order derivatives become so complicated that they are harder to understand than any of the algorithms we will derive.

Instead, Newton's method comes simply and naturally from Newton's theorem,

$$f(x) = f(x_c) + \int_{x_c}^{x} f'(z)\, dz.$$

It seems reasonable to approximate the indefinite integral by

$$\int_{x_c}^{x} f'(z)\, dz \cong f'(x_c)(x - x_c)$$

and once more obtain the affine approximation to $f(x)$ given by (2.2.2). This type of derivation will be helpful to us in multivariable problems, where geometrical derivations become less manageable.

Newton's method is typical of methods for solving nonlinear problems; it is an iterative process that generates a sequence of points that we hope come increasingly close to a solution. The obvious question is, "Will it work?" The

* We will refer to (2.2.2) as an *affine* model, although colloquially it is often called a *linear* model. The reason is that an affine model corresponds to an affine subspace through $(x, f(x))$, a line that does not necessarily pass through the origin, whereas a linear subspace must pass through the origin.

answer is a qualified "Yes." Notice that if $f(x)$ were linear, Newton's method would find its root in one iteration. Now let us see what it will do for the general square-root problem;

$$\text{given } \alpha > 0, \qquad \text{find } x \text{ such that } f(x) = x^2 - \alpha = 0,$$

starting from a current guess $x_c \neq 0$. Since

$$x_+ = x_c - \frac{f(x_c)}{f'(x_c)} = x_c - \frac{x_c^2 - \alpha}{2x_c} = \frac{x_c}{2} + \frac{\alpha}{2x_c},$$

one has

$$x_+ - \sqrt{\alpha} = \frac{x_c}{2} + \frac{\alpha}{2x_c} - \sqrt{\alpha} = \frac{(x_c - \sqrt{\alpha})^2}{2x_c}, \tag{2.2.4a}$$

or, using relative error, one has

$$\frac{x_+ - \sqrt{\alpha}}{\sqrt{\alpha}} = \left(\frac{x_c - \sqrt{\alpha}}{\sqrt{\alpha}}\right)^2 \cdot \left(\frac{\sqrt{\alpha}}{2x_c}\right). \tag{2.2.4b}$$

Thus as long as the initial error $|x_c - \sqrt{\alpha}|$ is less than $|2x_c|$, the new error $|x_+ - \sqrt{\alpha}|$ will be smaller than the old error $|x_c - \sqrt{\alpha}|$, and eventually each new error will be much smaller than the previous error. This agrees with our experience for finding the square root of 3 in the example that began this section.

The pattern of decrease in error given by (2.2.4) is typical of Newton's method. The error at each iteration will be approximately the square of the previous error, so that, if the initial guess is good enough, the error will decrease and eventually decrease rapidly. This pattern is known as local q-quadratic convergence. Before deriving the general convergence theorem for Newton's method, we need to discuss rates of convergence.

2.3 CONVERGENCE OF SEQUENCES OF REAL NUMBERS

Given an iterative method that produces a sequence of points $x_1, x_2, \ldots$, from a starting guess x_0, we will want to know if the iterates converge to a solution x_*, and if so, how quickly. If we assume that we know what it means to write

$$\lim_{k \to \infty} a_k = 0$$

for a real sequence $\{a_k\}$, then the following definition characterizes the properties we will need.

Definition 2.3.1 Let $x_* \in \mathbb{R}$, $x_k \in \mathbb{R}$, $k = 0, 1, 2, \ldots$. Then the sequence

$\{x_k\} = \{x_0, x_1, x_2, \ldots\}$ is said to *converge* to x_* if

$$\lim_{k \to \infty} |x_k - x_*| = 0.$$

If in addition, there exists a constant $c \in [0, 1)$ and an integer $\hat{k} \geq 0$ such that for all $k \geq \hat{k}$,

$$|x_{k+1} - x_*| \leq c\,|x_k - x_*| \tag{2.3.1}$$

then $\{x_k\}$ is said to be *q-linearly convergent* to x_*. If for some sequence $\{c_k\}$ that converges to 0,

$$|x_{k+1} - x_*| \leq c_k\,|x_k - x_*|, \tag{2.3.2}$$

then $\{x_k\}$ is said to *converge q-superlinearly* to x_*. If there exist constants $p > 1$, $c \geq 0$, and $\hat{k} \geq 0$ such that $\{x_k\}$ converges to x_* and for all $k \geq \hat{k}$,

$$|x_{k+1} - x_*| \leq c\,|x_k - x_*|^p, \tag{2.3.3}$$

then $\{x_k\}$ is said to converge to x_* with *q-order at least p*. If $p = 2$ or 3, the convergence is said to be *q-quadratic* or *q-cubic*, respectively.

If $\{x_k\}$ converges to x_* and, in place of (2.3.2),

$$|x_{k+j} - x_*| \leq c_k\,|x_k - x_*|$$

for some fixed integer j, then $\{x_k\}$ is said to be *j-step q-superlinearly convergent* to x_*. If $\{x_k\}$ converges to x_* and, in place of (2.3.3), for $k > \hat{k}$,

$$|x_{k+j} - x_*| \leq c\,|x_k - x_*|^p$$

for some fixed integer j, then $\{x_k\}$ is said to have *j-step q-order convergence* of order *at least p*.

An example of a q-linearly convergent sequence is

$$x_0 = 2,\ x_1 = \tfrac{3}{2},\ x_2 = \tfrac{5}{4},\ x_3 = \tfrac{9}{8},\ \ldots,\ x_i = 1 + 2^{-i},\ \ldots.$$

This sequence converges to $x_* = 1$ with $c = \frac{1}{2}$; on a CDC machine it will take 48 iterations until $\mathrm{fl}(x_k) = 1$. An example of a q-quadratically convergent sequence is

$$x_0 = \tfrac{3}{2},\ x_1 = \tfrac{5}{4},\ x_2 = \tfrac{17}{16},\ x_3 = \tfrac{257}{256},\ \ldots,\ x_k = 1 + 2^{-2^k},\ \ldots,$$

which converges to $x_* = 1$ with $c = 1$; on a CDC machine, $\mathrm{fl}(x_6)$ will equal 1. In practice, q-linear convergence can be fairly slow, whereas q-quadratic or q-superlinear convergence is eventually quite fast. However, actual behavior also depends upon the constants c in (2.3.1–2.3.3); for example, q-linear convergence with $c = 0.001$ is probably quite satisfactory, but with $c = 0.9$ it is not. (For further examples see Exercises 2 and 3). It is worth emphasizing that the utility of q-superlinear convergence is directly related to how many iterations are needed for c_k to become small.

The prefix "q" stands for quotient and is used to differentiate from "r" (root) orders of convergence. R-order* is a weaker type of convergence rate; all that is said of the errors $|x_k - x_*|$, of a sequence with r-order p, is that they are bounded above by another sequence of q-order p. A definitive reference is Ortega and Rheinboldt [1970]. An iterative method that will converge to the correct answer at a certain rate, provided it is started close enough to the correct answer, is said to be *locally convergent* at that rate. In this book we will be interested mainly in methods that are locally q-superlinearly or q-quadratically convergent and for which this behavior is apparent in practice.

2.4 CONVERGENCE OF NEWTON'S METHOD

We now show that, for most problems, Newton's method will converge q-quadratically to the root of one nonlinear equation in one unknown, provided it is given a good enough starting guess. However, it may not converge at all from a poor start, so that we need to incorporate the global methods of Section 2.5. The local convergence proof for Newton's method hinges on an estimate of the errors in the sequence of affine models $M_c(x)$ as approximations to $f(x)$. Since we obtained the approximations by using $f'(x_c)(x - x_c)$ to approximate

$$\int_{x_c}^{x} f'(z)\, dz$$

we are going to need to make some smoothness assumptions on f' in order to estimate the error in the approximation, which is

$$f(x) - M_c(x) = \int_{x_c}^{x} [f'(z) - f'(x_c)]\, dz.$$

First we define the notion of Lipschitz continuity.

Definition 2.4.1 A function g is Lipschitz continuous with constant γ in a set X, written $g \in \mathrm{Lip}_\gamma(X)$, if for every $x, y \in X$,

$$|g(x) - g(y)| \le \gamma |x - y|.$$

In order to prove the convergence of Newton's method, we first prove a simple lemma showing that if $f'(x)$ is Lipschitz continuous, then we can obtain a bound on how close the affine approximation $f(x) + f'(x)(y - x)$ is to $f(y)$.

* We will capitalize the prefix letters R and Q when they begin a sentence, but not otherwise.

LEMMA 2.4.2 For an open interval D, let $f: D \to \mathbb{R}$ and let $f' \in \mathrm{Lip}_\gamma(D)$. Then for any $x, y \in D$,

$$|f(y) - f(x) - f'(x)(y - x)| \le \frac{\gamma(y - x)^2}{2}. \tag{2.4.1}$$

Proof. From basic calculus, $f(y) - f(x) = \int_x^y f'(z)\, dz$, or equivalently,

$$f(y) - f(x) - f'(x)(y - x) = \int_x^y [f'(z) - f'(x)]\, dz. \tag{2.4.2}$$

Making the change of variables

$$z = x + t(y - x), \qquad dz = dt(y - x),$$

(2.4.2) becomes

$$f(y) - f(x) - f'(x)(y - x) = \int_0^1 [f'(x + t(y - x)) - f'(x)](y - x)\, dt,$$

and so by the triangle inequality applied to the integral and the Lipschitz continuity of f',

$$|f(y) - f(x) - f'(x)(y - x)| \le |y - x| \int_0^1 \gamma\, |t(y - x$$

$$= \gamma |y - x|^2/2. \qquad \square$$

Note that (2.4.1) closely resembles the error bound given by the Taylor series with remainder, with the Lipschitz constant γ taking the place of a bound on $|f''(\xi)|$ for $\xi \in D$. The main advantage of using Lipschitz continuity is that we do not need to discuss this next higher derivative. This is especially convenient in multiple dimensions.

We are now ready to state and prove a fundamental theorem of numerical mathematics. We will prove the most useful form of the result and leave the more general ones as exercises (see Exercises 13-14.)

THEOREM 2.4.3 Let $f: D \to \mathbb{R}$, for an open interval D, and let $f' \in \mathrm{Lip}_\gamma(D)$. Assume that for some $\rho > 0$, $|f'(x)| \ge \rho$ for every $x \in D$. If $f(x) = 0$ has a solution $x_* \in D$, then there is some $\eta > 0$ such that: if $|x_0 - x_*| < \eta$, then the sequence $\{x_k\}$ generated by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \qquad k = 0, 1, 2, \ldots$$

exists and converges to x_*. Furthermore, for $k = 0, 1, \ldots,$

$$|x_{k+1} - x_*| \le \frac{\gamma}{2\rho} |x_k - x_*|^2. \tag{2.4.3}$$

Proof. Let $\tau \in (0, 1)$, let $\hat{\eta}$ be the radius of the largest open interval around x_* that is contained in D, and define $\eta = \min\{\hat{\eta}, \tau(2\rho/\gamma)\}$. We will show by induction that for $k = 0, 1, 2, \ldots$, (2.4.3) holds, and

$$|x_{k+1} - x_*| \le \tau |x_k - x_*| < \eta.$$

The proof simply shows at each iteration that the new error $|x_{k+1} - x_*|$ is bounded by a constant times the error the affine model makes in approximating f at x_*, which from Lemma 2.4.2 is $O(|x_k - x_*|^2)$. For $k = 0$,

$$x_1 - x_* = x_0 - x_* - \frac{f(x_0)}{f'(x_0)} = x_0 - x_* - \frac{f(x_0) - f(x_*)}{f'(x_0)}$$

$$= \frac{1}{f'(x_0)} [f(x_*) - f(x_0) - f'(x_0)(x_* - x_0)].$$

The term in brackets is $f(x_*) - M_0(x_*)$, the error at x_* in the local affine model at $x_c = x_0$. Thus from Lemma 2.4.2,

$$|x_1 - x_*| \le \frac{\gamma}{2|f'(x_0)|} |x_0 - x_*|^2,$$

and by the assumptions on $f'(x)$

$$|x_1 - x_*| \le \frac{\gamma}{2\rho} |x_0 - x_*|^2.$$

Since $|x_0 - x_*| \le \eta \le \tau \cdot 2\rho/\gamma$, we have $|x_1 - x_*| \le \tau |x_0 - x_*| < \eta$. The proof of the induction step then proceeds identically. □

The condition in Theorem 2.4.3 that $f'(x)$ have a nonzero lower bound in D simply means that $f'(x_*)$ must be nonzero for Newton's method to converge quadratically. Indeed, if $f'(x_*) = 0$, then x_* is a multiple root, and Newton's method converges only linearly (see Exercise 12). To appreciate the difference, we give below sample iterations of Newton's method applied to $f_1(x) = x^2 - 1$ and $f_2(x) = x^2 - 2x + 1$, both starting from $x_0 = 2$. Notice how much more slowly Newton's method converges on $f_2(x)$ because $f_2'(x_*) = 0$.

EXAMPLE 2.4.4 Newton's Method Applied to Two Quadratics (CDC, Single Precision)

$f_1(x) = x^2 - 1$		$f_2(x) = x^2 - 2x + 1$
2	x_0	2
1.25	x_1	1.5
1.025	x_2	1.25
1.0003048780488	x_3	1.125
1.0000000464611	x_4	1.0625
1.0	x_5	1.03125

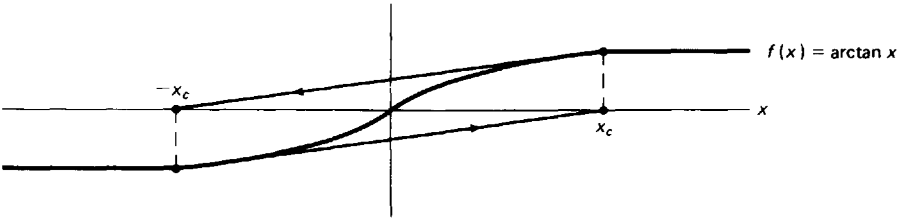

Figure 2.4.1 Newton's method applied to $f(x) = \arctan(x)$

It is also informative to examine the constant $\gamma/2\rho$ involved in the q-quadratic convergence relation (2.4.3). The numerator γ, a Lipschitz constant for f' on D, can be considered a measure of the nonlinearity of f. However, γ is a scale-dependent measure; multiplying f or changing the units of x by a constant will scale f' by that constant without making the function more or less nonlinear. A partially scale-free measure of nonlinearity is the relative rate of change in $f'(x)$, which is obtained by dividing γ by $f'(x)$. Thus, since ρ is a lower bound on $f'(x)$ for $x \in D$, γ/ρ is an upper bound on the relative nonlinearity of $f(x)$, and Theorem 2.4.3 says that the smaller this measure of relative nonlinearity, the faster Newton's method will converge. If f is linear, then $\gamma = 0$ and $x_1 = x_*$.

Theorem 2.4.3 guarantees the convergence of Newton's method only from a good starting point x_0, and indeed it is easy to see that Newton's method may not converge at all if $|x_0 - x_*|$ is large. For example, consider the function $f(x) = \arctan x$ (see Figure 2.4.1). For some $x_c \in [1.39, 1.40]$, if $x_0 = x_c$, then Newton's method will produce the cycle $x_1 = -x_c$, $x_2 = x_c$, $x_3 = -x_c, \ldots$. If $|x_0| < x_c$, Newton's method will converge to $x_* = 0$, but if $|x_0| > x_c$, Newton's method will diverge; i.e., the error $|x_k - x_*|$ will increase at each iteration. Thus Newton's method is useful to us for its fast local convergence, but we need to incorporate it into a more robust method that will be successful from farther starting points.

2.5 GLOBALLY CONVERGENT METHODS* FOR SOLVING ONE EQUATION IN ONE UNKNOWN

We will use a simple philosophy to incorporate Newton's method into a globally convergent algorithm: use Newton's method whenever it seems to be working well, otherwise fall back on a slower but sure global method. This strategy produces globally convergent algorithms with the fast local conver-

* For our definition of "global method," see the last paragraph of Section 1.1.

gence of Newton's method. In this section we discuss two global methods and then show how to combine a global method with Newton's method into a hybrid algorithm. We also discuss the stopping tests and other computer-dependent criteria necessary to successful computational algorithms.

The simplest global method is the method of bisection. It makes the somewhat reasonable assumption that one starts with an interval $[x_0, z_0]$ that contains a root. It sets x_1 to the midpoint of this interval, chooses the new interval to be the one of $[x_0, x_1]$ or $[x_1, z_0]$ that contains a root, and continues to halve the interval until a root is found (see Figure 2.5.1). This is expressed algebraically as:

```
given $x_0$, $z_0$ such that $f(x_0) \cdot f(z_0) < 0$,
for $k = 0, 1, 2, \ldots$, do
    $x_{k+1} = \dfrac{x_k + z_k}{2}$
    $z_{k+1} = \begin{cases} x_k, & \text{if } f(x_k) \cdot f(x_{k+1}) < 0, \\ z_k, & \text{otherwise.} \end{cases}$
```

The method of bisection always works in theory, but it is guaranteed only to reduce the error bound by $\frac{1}{2}$ for each iteration. This makes the method very marginal for practical use. Programs that use bisection generally do so only until an x_k is obtained from which some variant of Newton's method will converge. The method of bisection also does not extend naturally to multiple dimensions.

A method more indicative of how we will proceed in n-space is the following. Think of Newton's method as having suggested not only the step $x_N = x_c - f(x_c)/f'(x_c)$, but also the direction in which that step points. [Assume $f'(x_c) \neq 0$.] Although the Newton step may actually cause an increase in the absolute value of the function, its direction always will be one in which the absolute function value decreases initially (see Figure 2.5.2). This should be obvious geometrically; for the simple proof, see Exercise 16. Thus, if the Newton point x_N doesn't produce a decrease in $|f(x)|$, a reasonable strategy is to backtrack from x_N toward x_c until one finds a point x_+ for

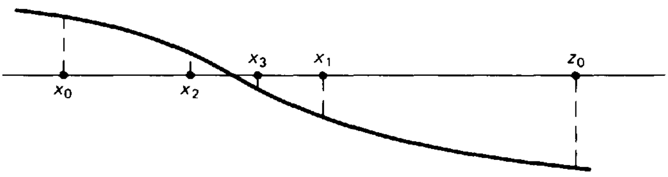

Figure 2.5.1 The method of bisection

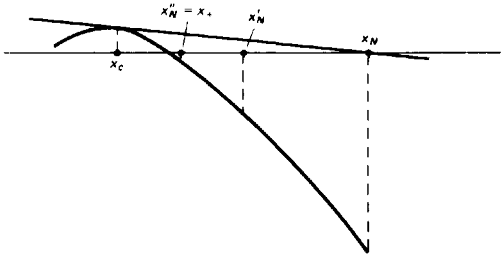

Figure 2.5.2 Backtracking from the Newton step

which $|f(x_+)| < |f(x_c)|$. A possible iteration is

$$x_+ = x_c - \frac{f(x_c)}{f'(x_c)}$$

$$\text{while } |f(x_+)| \geq |f(x_c)| \text{ do}$$

$$x_+ \leftarrow \frac{x_+ + x_c}{2}. \tag{2.5.1}$$

Note that this strategy does not require an initial interval bracketing a root.

Iteration (2.5.1) is an example of a hybrid algorithm, one that attempts to combine global convergence and fast local convergence by first trying the Newton step at each iteration, but always insisting that the iteration decreases some measure of the closeness to a solution. Constructing such hybrid algorithms is the key to practical success in solving multivariable nonlinear problems. Below is the general form of a class of hybrid algorithms for finding a root of one nonlinear equation; it is meant to introduce and emphasize those basic techniques for constructing globally and fast locally convergent algorithms that will be the foundation of all the algorithms in this book.

ALGORITHM 2.5.1 General hybrid quasi-Newton algorithm for solving one nonlinear equation in one unknown:

```
given $f: \mathbb{R} \rightarrow \mathbb{R}$, $x_0$,
for $k = 0, 1, 2, \ldots,$ do
    1. *decide* whether to stop; if not:
    2. *make a local model* of $f$ around $x_k$, and find the point $x_N$ that
       solves (or comes closest to solving) the model problem.
    3. (a) *decide* whether to take $x_{k+1} = x_N$, if not,
       (b) choose $x_{k+1}$ using a *global strategy* (make more conser-
           vative use of the solution to the model problem).
```

Step 1 is discussed below; it requires our first use of computer-dependent and problem-dependent tolerances. Step 2 usually involves calculating the Newton step, or a variant without derivatives (see Section 2.6). Equation (2.5.1) is an example of Step 3(a)–(b). We will see in Chapter 6 that the criterion in Step 3(a) has to be chosen with only a little bit of care to assure the global convergence in most cases of the hybrid algorithm to a solution.

Deciding when to stop is a somewhat ad hoc process that can't be perfect for every problem, yet it calls for considerable care. Since there may be no computer-representable x_* such that $f(x_*) = 0$, one must decide when one is "close enough." This decision usually takes two parts: "Have you approximately solved the problem?" or "Have the last two (or few) iterates stayed in virtually the same place?" The first question is represented by a test such as, "Is $|f(x_+)| < \tau_1$?" where the tolerance τ_1 is chosen to reflect the user's idea of being close enough to zero for this problem. For example, τ_1 might be set to $(\text{macheps})^{1/2}$. Naturally this test is very sensitive to the scale of $f(x)$, and so it is important that a routine instruct the user to choose τ_1, or scale f, so that an x_+ that satisfies $|f(x_+)| < \tau_1$ will be a satisfactory solution to the problem. Partly to guard against this condition's being too restrictive, the second question is included and it is tested by a relation such as, "Is $(|x_+ - x_c|/|x_+|) < \tau_2$?" A reasonable tolerance is $\tau_2 = (\text{macheps})^{1/2}$, which corresponds to stopping whenever the left half of the digits of x_c and x_+ agree, though any τ_2 greater than macheps can be selected. Since x_+ might be close to zero, the second test is usually modified to something like, "Is $(|x_+ - x_c|/\max\{|x_+|, |x_c|\} < \tau_2$?" A better test uses a user-supplied variable typ$x$ containing the typical size of x in the place of the $|x_c|$ in the denominator (see Exercise 17), so that the stopping condition on a CDC machine might be,

$$\text{if} \quad |f(x_+)| \le 10^{-5} \quad \text{or} \quad \frac{|x_+ - x_c|}{\max\{\text{typ}x, |x_+|\}} \le 10^{-7}, \quad \text{stop.}$$

In practice, $f(x_+)$ usually gets small before the step does in any problem for which local convergence is fast, but for a problem on which convergence is only linear, the step may become small first. The reader can already see that the choice of stopping rules is quite a can of worms, especially for poorly scaled problems. We will treat it more completely in Chapter 7.

2.6 METHODS WHEN DERIVATIVES ARE UNAVAILABLE

In many practical applications, $f(x)$ is not given by a formula; rather it is the output from some computational or experimental procedure. Since $f'(x)$ usually is not available then, our methods that use values of $f'(x)$ to solve $f(x) = 0$ must be modified to require only values of $f(x)$.