Chapter 2

ONE-VARIABLE OPTIMIZATION

1. Optimality conditions

We begin with a formal statement of the conditions which hold at a minimum of a one-variable differentiable function. We have already made use of these conditions in the previous chapter.

Definition Suppose that $F(x)$ is a continuously differentiable function of the scalar variable x, and that, when $x = x^*$,

$$\frac{dF}{dx} = 0 \quad \text{and} \quad \frac{d^2F}{dx^2} > 0. \tag{2.1.1}$$

The function $F(x)$ is then said to have a *local minimum* at x^*.

Conditions (2.1.1) imply that $F(x^*)$ is the smallest value of F in some region near x^*. It may also be true that $F(x^*) \leq F(x)$ for *all* x but condition (2.1.1) does not guarantee this.

Definition If conditions (2.1.1) hold at $x = x^*$ and if $F(x^*) \leq F(x)$ for *all* x then x^* is said to be the *global minimum*.

In practice it is usually hard to establish that x^* is a global minimum and so we shall chiefly be concerned with methods of finding local minima.

Conditions (2.1.1) are called *optimality conditions*. For simple problems they can be used directly to find a minimum, as in section 3 and 1.3 of chapter 1. As another example, consider the function

$$F(x) = x^3 - 3x^2. \tag{2.1.2}$$

Here, and in what follows, we shall sometimes use the notation

$$F'(x) = \frac{dF}{dx} \quad \text{and} \quad F''(x) = \frac{d^2F}{dx^2}.$$

Thus, for (2.1.2), $F'(x) = 3x^2 - 6x$ and so $F'(x) = 0$ when $x = 0$ and $x = 2$. These values represent two *stationary points* of $F(x)$; and to determine which is a minimum we must consider $F''(x) = 6x - 6$. We see that F has a minimum at $x = 2$ because $F''(2) > 0$. However, $F''(0)$ is negative and so $F(x)$ has a local maximum at $x = 0$.

We can only use this analytical approach to solve minimization problems when it is easy to form and solve the equation $F'(x) = 0$. This may not be the case for functions $F(x)$ which occur in practical problems – for instance **Maxret1m**. Therefore we shall usually resort to iterative techniques.

Some iterative methods are called *direct search* techniques and are based on simple comparison of function values at trial points. Others are known as *gradient methods*. These use derivatives of the objective function and can be viewed as iterative algorithms for solving the nonlinear equation $F'(x) = 0$. Gradient methods tend to converge faster than direct search methods. They also have the advantage that they permit an obvious convergence test – namely stopping the iterations when the gradient is near zero. Gradient methods are not always suitable, however – i.e. when $F(x)$ has discontinuous derivatives, as on a piecewise linear function. We have already seen an example of this in function (1.7.8).

Exercises

1. Show that, if the optimality conditions (2.1.1) hold, $F(x^* + h) > F(x^*)$ for h sufficiently small.

2. Find the stationary points of $F(x) = 4\cos x^2 - \sin x^2 - 3$.

2. The bisection method

A simple (but inefficient) way of estimating the least value of $F(x)$ in a range $a \leq x \leq b$ would be calculate the function at many points in $[a,b]$ and then pick the one with lowest value. The *bisection method* uses a more systematic approach to the evaluation of F in $[a,b]$. Each iteration uses a comparison between function values at five points to reduce by half the size of a bracket containing the minimum. It can be shown that this approach will locate a minimum if it is applied to a function $F(x)$ that is *unimodal* – i.e., one that has only one minimum in the range $[a,b]$.

A statement of the algorithm is given below.

Bisection Method for minimizing $F(x)$ on the range $[a,b]$

Set $x_a = a$, $x_b = b$ and $x_m = \frac{1}{2}(a+b)$.
Calculate $F_a = F(x_a)$, $F_b = F(x_b)$, $F_m = F(x_m)$
Repeat
set $x_l = \frac{1}{2}(x_a + x_m)$, $x_r = \frac{1}{2}(x_m + x_b)$
calculate $F_l = F(x_l)$ and $F_r = F(x_r)$
let $F_{min} = \min\{F_a, F_b, F_m, F_l, F_r\}$
if $F_{min} = F_a$ or F_l then set $x_b = x_m$, $x_m = x_l$, $F_b = F_m$, $F_m = F_l$
else if $F_{min} = F_m$ then set $x_a = x_l$, $x_b = x_r$, $F_a = F_l$, $F_b = F_r$
else if $F_{min} = F_r$ or F_b then set $x_a = x_m$, $x_m = x_r$, $F_a = F_m$, $F_m = F_r$
until $|x_b - x_a|$ is sufficiently small

Proposition If $F(x)$ is unimodal for $a \le x \le b$ with a minimum at x^* then the number of iterations taken by the bisection method to locate x^* within a bracket of width less than 10^{-s} is K, where K is the smallest integer which exceeds

$$\frac{log_{10}(b-a)+s}{log_{10}(2)}. \tag{2.2.1}$$

Proof The size of the bracket containing the solution is halved on each iteration. Hence, after k iterations the width of the bracket is $2^{-k}(b-a)$. To find the value of k which gives

$$2^{-k}(b-a) = 10^{-s}$$

we take logs of both sides and get

$$log_{10}(b-a) - klog_{10}(2) = -s$$

and so the width of the bracket is less than 10^{-s} once k exceeds (2.2.1).

We can show how the bisection method works by applying it to the problem

$$\text{Minimize } F(x) = x^3 - 3x^2 \text{ for } 0 \le x \le 3.$$

Initially $x_a = 0, x_b = 3, x_m = 1.5$, and the first iteration adds the points $x_l = 0.75$ and $x_r = 2.25$. We then find

$$F_a = 0;\ F_l = -1.266;\ F_m = -3.375;\ F_r = -3.797;\ F_b = 0.$$

The least function value is at $x_r = 2.25$ and hence the search range for the next iteration is $[x_m,\ x_b] = [1.5,\ 3.0]$. After re-labelling the points and computing new values x_l,x_r we get

$$x_a = 1.5;\ x_l = 1.875;\ x_m = 2.25;\ x_r = 2.625;\ x_b = 3$$

and $\quad F_a = -3.375;\ F_l = -3.955;\ F_m = -3.797;\ F_r = -2.584;\ F_b = 0.$

Now the least function value is at x_l and the new range is $[x_a, x_m] = [1.5,\ 2.25]$. Re-labelling and adding the new x_l and x_r we get

$$x_a = 1.5;\ x_l = 1.6875;\ x_m = 1.875;\ x_r = 2.0625;\ x_b = 2.25$$

and $\quad F_a = -3.375;\ F_l = -3.737;\ F_m = -3.955;\ F_r = -3.988;\ F_b = -3.797.$

These values imply the minimum lies in $[x_l, x_r] = [1.875, 2.25]$. After a few more steps we have an acceptable approximation to the true solution at $x = 2$.

The application of the bisection method to a maximum-return problem is illustrated below in section 6 of this chapter.

Finding a bracket for a minimum

We now give a systematic way of finding a range $a < x < b$ which contains a minimum of $F(x)$. This method uses the slope F' to determine whether the search for a minimum should be to the left or the right of an initial point x_0. If $F'(x_0)$ is positive then lower values of the function will be found for $x < x_0$, while $F'(x_0) < 0$ implies that lower values of F occur when $x > x_0$. The algorithm merely takes larger and larger steps in a "downhill" direction until the function starts to increase, indicating that a minimum has been bracketed.

Finding a and b to bracket a local minimum of $F(x)$

Choose an initial point x_0 and a step size $\alpha(> 0)$
Set $\delta = -\alpha \times sign(F'(x_0))$
Repeat for $k = 0, 1, 2, ...$
$x_{k+1} = x_k + \delta, \quad \delta = 2\delta$
until $F(x_{k+1}) > F(x-k)$
if $k = 0$ then set $a = x_0$ and $b = x_1$
if $k > 0$ then set $a = x_{k-1}$ and $b = x_{k+1}$

Exercises

1. Apply the bisection method to $F(x) = e^x - 2x$ in the interval $0 \le x \le 1$.

2. Do two iterations of the bisection method for the function $F(x) = x^3 + x^2 - x$ in the range $0 \le x \le 1$. How close is the best point found to the exact minimum of F? What happens if we apply the bisection method in the range $-2 \le x \le 0$?

3. Use the bracketing technique with $x_0 = 1$ and $\alpha = 0.1$ to find a bracket for a minimum of $F(x) = e^x - 2x$.

4. Apply the bracketing algorithm to the function (1.7.5) with $V_a = 0.00123$ and $\rho = 10$ and using $x_0 = 0$ and $\alpha = 0.25$ to determine a starting range for

the bisection method. How does the result compare with the bracket obtained when $x_0 = 1$ and $\alpha = 0.2$?

3. The secant method

We now consider an iterative method for solving $F'(x) = 0$. This will find a local minimum of $F(x)$ *provided we use it in a region where the second derivative $F''(x)$ remains positive*. The approach is based on linear interpolation. If we have evaluated F' at two points $x = x_1$ and $x = x_2$ then the calculation

$$x = x_1 - \frac{F'(x_1)}{F'(x_2) - F'(x_1)}(x_2 - x_1) \tag{2.3.1}$$

gives an estimate of the point where F' vanishes.

We can show that formula (2.3.1) is exact if F is a quadratic function. Consider $F(x) = x^2 - 3x - 1$ for which $F'(x) = 2x - 3$ and suppose we use $x_1 = 0$ and $x_2 = 2$. Then (2.3.1) gives

$$x = 0 - \frac{F'(0)}{F'(2) - F'(0)} \times 2 = 0 - \frac{-3}{4} \times 2 = 1.5.$$

Hence (2.3.1) has yielded the minimum of $F(x)$. However, when F is not quadratic, we have to apply the interpolation formula iteratively. The algorithm below shows how this might be done.

Secant method for solving $F'(x) = 0$

Choose x_0, x_1 as two estimates of the minimum of $F(x)$
Repeat for $k = 0, 1, 2...$

$$x_{k+2} = x_k - \frac{F'(x_k)}{F'(x_{k+1}) - F'(x_k)}(x_{k+1} - x_k)$$

until $|F'(x_{k+2})|$ is sufficiently small.

This algorithm makes repeated use of formula (2.3.1) based upon the two most recently calculated points. In fact, this may not be the most efficient way to proceed. When $k > 1$, we would normally calculate x_{k+2} using x_{k+1} together with *either* x_k *or* x_{k-1} according to one of a number of possible strategies:
(a) Choose whichever of x_k and x_{k-1} gives the smaller value of $|F'|$.
(b) Choose whichever of x_k and x_{k-1} gives F' with opposite sign to $F'(x_{k+1})$
(c) Choose whichever of x_k and x_{k-1} gives the smaller value of F.
Strategies *(a)* and *(c)* are based on using points which seem closer to the minimum; strategy *(b)* reflects the fact that linear interpolation is more reliable than

linear extrapolation (which occurs when $F'(x_{k+1})$ and $F'(x_k)$ have the same sign). Strategy *(b)*, however, can only be employed if we have chosen our initial x_0 and x_1 so that $F'(x_0)$ and $F'(x_1)$ have opposite signs.

We can demonstrate the secant method with strategy *(a)* on $F(x) = x^3 - 3x^2$ for which $F'(x) = 3x^2 - 6x$. If $x_0 = 1.5$ and $x_1 = 3$ then $F'(x_0) = -2.25$ and $F'(x_1) = 9$. Hence the first iteration gives

$$x_2 = 1.5 - \frac{-2.25}{11.25} \times 1.5 = 1.8$$

and so $F'(x_2) = -1.08$. For the next iteration we re-assign $x_1 = 1.5$, since $|F'(1.5)| < |F'(3)|$, and so we get

$$x_3 = 1.5 - \frac{-2.25}{-1.17} \times 0.3 \approx 2.077.$$

The iterations appear to be converging towards the correct solution at $x = 2$, and the reader can perform further calculations to confirm this.

Exercises

1. Apply the secant method to $F(x) = e^x - 2x$ in the range $0 \le x \le 1$.

2. Show that (2.3.1) will give $F'(x) = 0$ when applied to *any* quadratic function

$$F(x) = ax^2 + bx + c.$$

3. Use the secant method with strategy *(b)* on $F(x) = x^3 - 3x^2$ with $x_0 = 1.5$ and $x_1 = 3$. What happens if the starting values are $x_0 = 0.5$ and $x_1 = 1.5$?

4. The Newton method

This method seeks the minimum of $F(x)$ using both first *and* second derivatives. In its simplest form it is described as follows.

Newton method for minimizing $F(x)$

Choose x_0 as an estimate of the minimum of $F(x)$
Repeat for $k = 0, 1, 2...$

$$x_{k+1} = x_k - \frac{F'(x_k)}{F''(x_k)} \tag{2.4.1}$$

until $|F'(x_{k+1})|$ is sufficiently small.

This algorithm is derived by expanding $F(x)$ as a Taylor series about x_k

$$F(x_k + h) = F(x_k) + hF'(x_k) + \frac{h^2}{2}F''(x_k) + O(h^3). \tag{2.4.2}$$

Differentiation with respect to h gives a Taylor series for $F'(x)$

$$F'(x_k+h) = F'(x_k) + hF''(x_k) + O(h^2). \tag{2.4.3}$$

If we assume that h is small enough for the $O(h^2)$ term in (2.4.3) to be neglected then it follows that the step $h = -F'(x_k)/F''(x_k)$ will give $F'(x_k+h) \approx 0$.

As an illustration, we apply the Newton method to $F(x) = x^3 - 3x^2$ for which $F'(x) = 3x^2 - 6x$ and $F''(x) = 6x - 6$. At the initial guess $x_0 = 3$, $F' = 9$ and $F'' = 12$ and so the next iterate is given by

$$x_1 = 3 - \frac{9}{12} = 2.25.$$

Iteration two uses $F'(2.25) = 1.6875$ and $F''(2.25) = 7.5$ to give

$$x_2 = 2.25 - \frac{1.6875}{7.5} = 2.025.$$

After one more iteration $x_3 \approx 2.0003$ and so Newton's method is converging to the solution $x^* = 2$ more quickly than either bisection or the secant method.

Convergence of the Newton method

Because the Newton iteration is important in the development of optimization methods we study its convergence more formally. We define

$$e_k = x^* - x_k \tag{2.4.4}$$

as the error in the approximate minimum after k iterations.

Proposition Suppose that the Newton iteration (2.4.1) converges to x^*, a local minimum of $F(x)$, and that $F''(x^*) = m > 0$. Suppose also that there is some neighbourhood, N, of x^* in which the third derivatives of F are bounded, so that, for some $M > 0$,

$$M \geq F'''(x) \geq -M \text{ for all } x \in N. \tag{2.4.5}$$

If e_k is defined by (2.4.4) then there exists an integer K such that, for all $k > K$,

$$\frac{e_k^2 M}{m} > e_{k+1} > -\frac{e_k^2 M}{m}. \tag{2.4.6}$$

Proof Since the iterates x_k converge to x^* there exists an integer K such that

$$x_k \in N \text{ and } |e_k| < \frac{m}{2M} \quad \text{for } k > K.$$

Then the bounds (2.4.5) on F''' imply

$$m + M|e_k| > F''(x_k) > m - M|e_k|.$$

Combining this with the bound on $|e_k|$, we get

$$F''(x_k) > \frac{m}{2}. \tag{2.4.7}$$

Now, by the mean value form of Taylor's theorem ,

$$F'(x^*) = F'(x_k) + e_k F''(x_k) + \frac{1}{2} e_k^2 F'''(\xi),$$

for some ξ between x^* and x_k; and since $F'(x^*) = 0$ we deduce

$$F'(x_k) = -e_k F''(x_k) + \frac{1}{2} e_k^2 F'''(\xi).$$

The next estimate of the minimum is $x_{k+1} = x_k - \delta x_k$ where

$$\delta x_k = \frac{F'(x_k)}{F''(x_k)} = -e_k + \frac{e_k^2 F'''(\xi)}{2F''(x_k)}.$$

Hence the error after $k+1$ iterations is

$$e_{k+1} = x^* - x_{k+1} = e_k + \delta x_k = \frac{e_k^2 F'''(\xi)}{2F''(x_k)}.$$

Thus (2.4.6) follows, using (2.4.5) and (2.4.7).

This result shows that, when x_k is near to x^*, the error e_{k+1} is proportional to e_k^2 and so the Newton method ultimately approaches the minimum very rapidly.

Definition If, for some constant C, the errors e_k, e_{k+1} on successive steps of an iterative method satisfy

$$|e_{k+1}| \leq C e_k^2 \quad \text{as} \quad k \to \infty$$

then the iteration is said to have a *quadratic* rate of ultimate convergence.

Implementation of the Newton method

The convergence result in section 4 depends on certain assumptions about higher derivatives and this should warn us that the basic Newton iteration may not always converge. For instance, if the iterations reach a point where $F''(x)$ is zero the calculation will break down. It is not only this extreme case which can cause difficulties, however, as the following examples show.

Consider the function $F(x) = x^3 - 3x^2$, and suppose the Newton iteration is started from $x_0 = 1.1$. Since $F'(x) = 3x^2 - 6x$ and $F''(x) = 6x - 6$, we get

$$x_1 = 1.1 - \frac{(-2.97)}{0.6} = 6.05.$$

However, the minimum of $x^3 - 3x^2$ is at $x = 2$; and hence the method has overshot the minimum and given x_1 *further away* from the solution than x_0.

Suppose now that the Newton iteration is applied to $x^3 - 3x^2$ starting from $x_0 = 0.9$. The new estimate of the minimum turns out to be

$$x_1 = 0.9 - \frac{(-1.89)}{(-0.6)} = -2.25,$$

and the direction of the Newton step is away from the minimum. The iteration is being attracted to the maximum of $F(x)$ at $x = 0$. (Since the Newton method solves $F'(x) = 0$, this is not an unreasonable outcome.)

These two examples show that convergence of the basic Newton iteration depends upon the behaviour of $F''(x)$ and a practical algorithm should include safeguards against divergence. We should only use (2.4.1) if $F''(x)$ is strictly positive; and even then we should also check that the new point produced by the Newton formula is "better" than the one it replaces. These ideas are included in the following algorithm which applies the Newton method within a bracket $[a, b]$ such as can be found by the algorithm in section 2.

Safeguarded Newton method for minimizing $F(x)$ in $[a, b]$

Make a guess x_0 ($a < x_0 < b$) for the minimum of $F(x)$
Repeat for $k = 0, 1, 2, ..$
  if $F''(x_k) > 0$ then $\delta x = -F'(x_k)/F''(x_k)$
  else $\delta x = -F'(x_k)$
  if $\delta x < 0$ then $\alpha = \min(1, (a - x_k)/\delta x)$
  if $\delta x > 0$ then $\alpha = \min(1, (b - x_k)/\delta x)$
  Repeat for $j = 0, 1, ...$
    $\alpha = 0.5^j \alpha$
  until $F(x_k + \alpha\delta x) < F(x_k)$
  Set $x_{k+1} = x_k + \alpha\delta x$
until $|F'(x_{k+1})|$ is sufficiently small.

As well as giving an alternative choice of δx when $F'' \leq 0$, the safeguarded Newton algorithm includes a step-size, α. This is chosen first to prevent the correction steps from going outside the bracket $[a, b]$ and then, by repeated halving, to ensure that each new point has a lower value of F than the previous one. The algorithm always tries the full step ($\alpha = 1$) first and hence it can have the same fast ultimate convergence as the basic Newton method.

We can show the working of the safeguarded Newton algorithm on the function $F(x) = x^3 - 3x^2$ in the range [1,4] with $x_0 = 1.1$. Since

$$F(1.1) = -2.299,\ F'(1.1) = -2.97 \text{ and } F''(1.1) = 0.6$$

the first iteration gives $\delta x = 4.95$. The full step, $\alpha = 1$, gives $x_k + \alpha\delta x = 6.05$ which is outside the range we are considering and so we must re-set

$$\alpha = \frac{(4-1.1)}{4.95} \approx 0.5859.$$

However, $F(4) = 16 > F(1.1)$ and α is reduced again (to about 0.293) so that

$$x_k + \alpha\delta x = 1.1 + 0.293 \times 4.95 \approx 2.45.$$

Now $F(2.45) \approx -3.301$ which is less than $F(1.1)$. Therefore the inner loop of the algorithm is complete and the next iteration can begin.

Under certain assumptions, we can show that the inner loop of the safeguarded Newton algorithm will always terminate and hence that the safeguarded Newton method will converge.

Exercises

1. Use Newton's method to estimate the minimum of $e^x - 2x$ in $0 \le x \le 1$. Compare the rate of convergence with that of the bisection method.

2. Show that, for any starting guess, the basic Newton algorithm converges in one step when applied to a quadratic function.

3. Do one iteration of the basic Newton method on the function $F(x) = x^3 - 3x^2$ starting from each of the following initial guesses:

$$x_0 = 2.1, \quad x_0 = 1, \quad x_0 = -1.$$

Explain what happens in each case.

4. Do two iterations of the safeguarded Newton method applied to the function $x^3 - 3x^2$ and starting from $x_0 = 0.9$.

5. Methods using quadratic or cubic interpolation

Each iteration of Newton's method generates x_{k+1} as a stationary point of the interpolating quadratic function defined by the values of $F(x_k)$, $F'(x_k)$ and $F''(x_k)$. In a similar way, a direct-search iterative approach can be based on locating the minimum of the quadratic defined by values of F at three points x_k, x_{k-1}, x_{k-2}; and a gradient approach could minimize the local quadratic approximation given by, say, $F(x_{k-1})$, $F'(x_{k-1})$ and $F(x_k)$. If a quadratically predicted minimum x_{k+1} is found to be "close enough" to x^* (e.g. because $F'(x_{k+1}) \approx 0$) then the iteration terminates; otherwise x_{k+1} is used instead of one of the current points to generate a new quadratic model and hence predict a new minimum.

As with the Newton method, the practical implementation of this basic idea requires certain safeguards, mostly for dealing with cases where the interpolated quadratic has negative curvature and therefore does not have a minimum. The bracketing algorithm given earlier may prove useful in locating a group of points which implies a suitable quadratic model.

A similar approach is based on repeated location of the minimum of a *cubic* polynomial fitted either to values of F at four points or to values of F and F' at two points. This method can give faster convergence, but it also requires fall-back options to avoid the search being attracted to a maximum rather than a minimum of the interpolating polynomial.

Exercises

1. Suppose that $F(x)$ is a quadratic function and that, for any two points x_a, x_b, the ratio D is defined by

$$D = \frac{F(x_b) - F(x_a)}{(x_b - x_a)F'(x_a)}.$$

Show that $D = 0.5$ when x_b is the minimum of $F(x)$. What is the expression for D if $F(x)$ is a cubic function?

2. Explain why the secant method can be viewed as being equivalent to quadratic interpolation for the function $F(x)$.

3. Design an algorithm for minimizing $F(x)$ by quadratic interpolation based on function values only.

6. Solving maximum-return problems

The `SAMPO` software was mentioned briefly in section 5 of chapter 1. The program `sample2` is designed to read asset data like that in Table 1.3 and then to solve problem **Maxret1m** using either the bisection, secant or Newton method. In this section we shall quote some results from `sample2` in order to compare the performance of the three techniques. The reader should be able to obtain similar comparative results using other implementations of these one-variable optimization algorithms. (It should be understood, however, that two different versions of the same iterative optimization method may not give *exactly* the same sequence of iterates even though they eventually converge to the same solution. Minor discrepancies can arise because of rounding errors in computer arithmetic when the same calculation is expressed in two different ways or when two computer systems work to different precision.)

We recall that the maximum-return problem **Maxret1m** involves minimizing the function (1.7.4). In particular, using data in Table 1.3, (1.7.4) becomes

$$-0.06667x - 1.1167 + \frac{\rho}{V_a^2}(0.1256x^2 - 0.1589x + 0.05139 - V_a)^2 \quad (2.6.1)$$

where ρ is a weighting parameter and V_a denotes an acceptable value for risk. In order to choose a sensible value for V_a it can be helpful, first of all, to solve **Minrisk0** to obtain the *least possible* value of risk, V_{min}. For the data in Table 1.3 we find that $V_{min} \approx 0.00112$. Suppose now that we seek the maximum return when the acceptable risk is $V_a = 1.5V_{min} \approx 0.00168$. Table 2.1 shows results obtained when we use the bisection, secant and Newton methods to minimize (2.6.1) with $\rho = 1$.

Method	itns	y_1	y_2	V	R
Bisection	14	0.7	0.3	0.00169	1.233%
Secant	10	0.7	0.3	0.00169	1.233%
Newton	7	0.7	0.3	0.00169	1.233%

Table 2.1. Solutions of **Maxret1m** for assets in Table 1.3

We see that all three minimization methods find the same solution but that they use different numbers of iterations.

When drawing conclusions from a comparison like that in Table 2.1 it is important to ensure that all the methods have used the same (or similar) initial guessed solutions and convergence criteria. In the above results, both bisection and the secant method were given the starting values $x = 0$ and $x = 1$. The Newton method was started from $x = 1$ (it only needs one initial point). The bisection iterations terminate when the bracket containing the optimum has been reduced to a width less than 10^{-4}. Convergence of the secant and Newton methods occurs when the gradient $|F'(x)| < 10^{-5}$. Therefore it seems reasonable to conclude that the Newton method is indeed more efficient than the secant method which in turn is better than bisection.

We now consider another problem to see if similar behaviour occurs. This time we use data for the first two assets in Table 1.2 and we consider the minimization of (1.7.4) with $\rho = 10$ and $V_a = 0.00072$. Table 2.2 summarises results obtained with `sample2` using the same starting guessed values and convergence criteria as for Table 2.1 together with an additional result for Newton's method when the initial solution estimate is $x_0 = 1$.

Method	itns	y_1	y_2	V	R
Bisection	14	0.23	0.77	0.00072	0.035%
Secant	6	0.23	0.77	0.00072	0.035%
Newton ($x_0 = 0$)	5	0.23	0.77	0.00072	0.035%
Newton ($x_0 = 1$)	7	0.564	0.436	0.00072	0.08%

Table 2.2. Solutions of **Maxret1m** for first two assets in Table 1.2

The first three rows of Table 2.2 again show the secant and Newton methods outperforming bisection by finding the same solution in fewer iterations. In fact the number of bisection steps depends *only* on the size of the starting bracket and the convergence criterion while the iteration count for the secant and Newton methods can vary from one problem to another.

The last row of Table 2.2 shows that a different solution is obtained if Newton's method is started from $x_0 = 1$ instead of $x_0 = 0$. This alternative solution is actually better than the one in the first three rows of the table because the return R is larger. However, both the solutions are valid local minima of (1.7.4) and we could say, therefore, that the bisection and secant methods have been "unlucky" in converging to the inferior one. As mentioned in section 1, most of the methods covered in this book will terminate when the iterations reach a *local* optimum. If there are several minima, it is partly a matter of chance which one is found, although the one "nearest" to the starting guess is probably the strongest contender.

Exercises (To be solved using `sample2` or other suitable software.)

1. Using the data for the first two assets in Table 1.2, determine the coefficients of the function (1.7.4). Hence find the maximum return for an acceptable risk $V_a = 0.0005$ by using the bisection method to minimize (1.7.4). How does your solution change for different values of ρ in the range $0.1 \le \rho \le 10$?

2. Solve the maximum-return problem in question 1 but using the bisection method to minimize the non-smooth function (1.7.7). Explain why the results differ from those in question 1.

3. Using data for the first two assets in Table 1.2, form the function (1.7.4) and minimize it by the secant method when $V_a = 0.002$ and $\rho = 10$. Use starting guesses 0 and 1 for x. Can you explain why a different solution is obtained

when the starting guesses for x are 0.5 and 1? (It may help to sketch the function being minimized.)

4. Minimize (2.6.1) by Newton's method using the initial guess $x_0 = 0.5$ and explain why the solution is different from the ones quoted in Table 2.1. Find starting ranges for the bisection and secant methods from which they too will converge to this alternative local minimum.

5. Plot the graph of (2.6.1) for $0.55 \le x \le 0.75$, when $V_a = 0.00123$ and $\rho = 10$ and observe the two local minima. Also plot the graph when $V_a = 0.001$. Can you explain why there is a unique minimum in this case? What is the smallest value of V_a for which two minima occur?

6. Plot the graph of (1.7.8) with $\rho = 1$ in the range $0.55 \le x \le 0.75$ first for $V_a = 0.001$ and then for $V_a = 0.00123$. Does the function have a continuous first derivative in both cases? Explain your answer.

7. Using any suitable optimization method, minimize (2.6.1) and hence solve the maximum-return problem for the data in Table 1.3 for values of V_a in the range $0.0011 \le V_a \le 0.0033$. Plot the resulting values of y_1 and y_2 against V_a. Do these show a linear relationship? How does the graph of maximum-return R against V_a compare with the efficient frontier you would obtain by solving **Minrisk1m** for the data in Table 1.3?

8. As an alternative to the composite function in **Risk-Ret1**, another function whose minimum value will give a low value of risk coupled with a high value for expected return is

$$F = \frac{y^T Q y}{\bar{r}^T y}. \tag{2.6.2}$$

For a two-asset problem, use similar ideas to those in section 1.3 of chapter 1 and express F as a function of invested fraction y_1 only. Use the bisection method to minimize this one-variable form of (2.6.2) for the data in Table 1.3. Also obtain an expression for dF/dy_1 and use the secant method to estimate the minimum of F for the same set of data.

http://www.springer.com/978-1-4020-8110-1

Nonlinear Optimization with Financial Applications
Bartholomew-Biggs, M.
2005, XVII, 261 p., Hardcover
ISBN: 978-1-4020-8110-1