# Trust-Region Methods

Line search methods and trust-region methods both generate steps with the help of a quadratic model of the objective function, but they use this model in different ways. Line search methods use it to generate a search direction, and then focus their efforts on finding a suitable step length $\alpha$ along this direction. Trust-region methods define a region around the current iterate within which they trust the model to be an adequate representation of the objective function, and then choose the step to be the approximate minimizer of the model in this trust region. In effect, they choose the direction and length of the step simultaneously. If a step is not acceptable, they reduce the size of the region and find a new minimizer. In general, the step direction changes whenever the size of the trust region is altered.

The size of the trust region is critical to the effectiveness of each step. If the region is too small, the algorithm misses an opportunity to take a substantial step that will move it much closer to the minimizer of the objective function. If too large, the minimizer of the model may be far from the minimizer of the objective function in the region, so we may have to reduce the size of the region and try again. In practical algorithms, we choose the size of the region according to the performance of the algorithm during previous iterations. If the model is generally reliable, producing good steps and accurately predicting the behavior of the objective function along these steps, the size of the trust region is steadily increased to

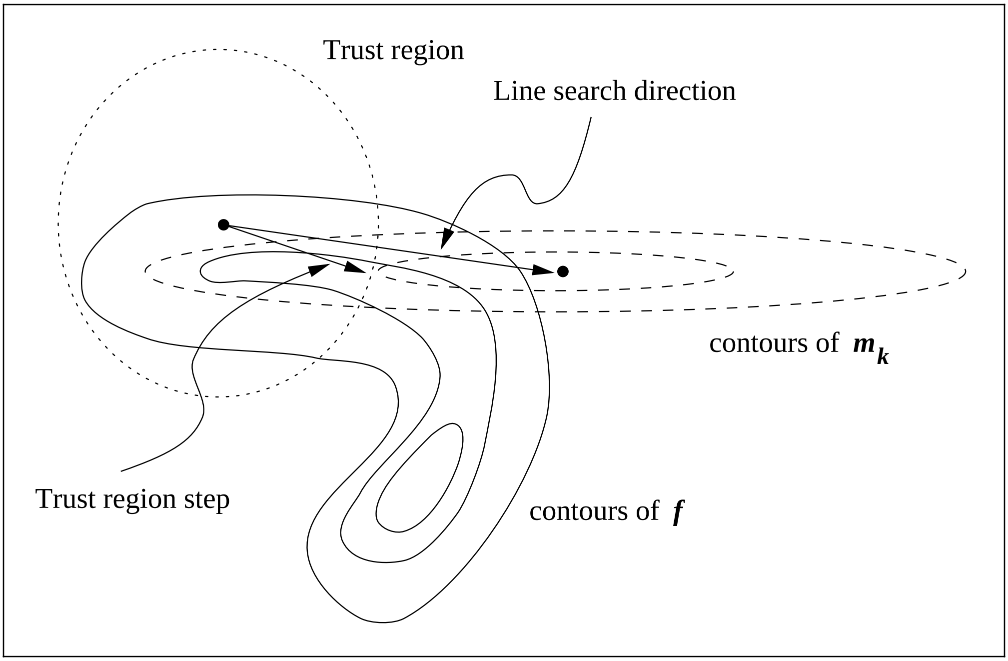


**Figure 4.1** Trust-region and line search steps.

allow longer, more ambitious, steps to be taken. On the other hand, a failed step indicates that our model is an inadequate representation of the objective function over the current trust region, so we reduce the size of the region and try again.

Figure 4.1 illustrates the trust-region approach on a function $f$ of two variables in which the current point lies at one end of a curved valley while the minimizer $x^*$ lies at the other end. The quadratic model function $m_k$, whose elliptical contours are shown as dashed lines, is based on function and derivative information at $x_k$ and possibly also on information accumulated from previous iterations and steps. A line search method based on this model searches along the step to the minimizer of $m_k$ (shown), but this direction allows only a small reduction in $f$ even if an optimal step is taken. A trust-region method, on the other hand, steps to the minimizer of $m_k$ within the dotted circle, which yields a more significant reduction in $f$ and a better step.

We will assume that the first two terms of the quadratic model functions $m_k$ at each iterate $x_k$ are identical to the first two terms of the Taylor-series expansion of $f$ around $x_k$. Specifically, we have

$$m_k(p) = f_k + \nabla f_k^T p + \tfrac{1}{2} p^T B_k p, \tag{4.1}$$

where $f_k = f(x_k)$, $\nabla f_k = \nabla f(x_k)$, and $B_k$ is some symmetric matrix. Since by (2.6) we have

$$f(x_k + p) = f_k + \nabla f_k^T p + \tfrac{1}{2} p^T \nabla^2 f(x_k + tp) p, \tag{4.2}$$

for some scalar $t \in (0, 1)$, and since $m_k(p) = f_k + \nabla f_k^T p + O\left(\|p\|^2\right)$, the difference between $m_k(p)$ and $f(x_k + p)$ is $O\left(\|p\|^2\right)$, so the approximation error is small when $p$ is small.

When $B_k$ is equal to the true Hessian $\nabla^2 f(x_k)$, the model function actually agrees with the Taylor series to *three* terms. The approximation error is $O\left(\|p\|^3\right)$ in this case, so this model is especially accurate when $\|p\|$ is small. The algorithm based on setting $B_k = \nabla^2 f(x_k)$ is called the trust-region Newton method, and will be discussed further in Chapter 6. In the current chapter, we emphasize the generality of the trust-region approach by assuming little about $B_k$ except symmetry and uniform boundedness in the index $k$.

To obtain each step, we seek a solution of the subproblem

$$\min_{p \in \mathbb{R}^n} m_k(p) = f_k + \nabla f_k^T p + \tfrac{1}{2} p^T B_k p \qquad \text{s.t. } \|p\| \le \Delta_k, \tag{4.3}$$

where $\Delta_k > 0$ is the trust-region radius. For the moment, we define $\|\cdot\|$ to be the Euclidean norm, so that the solution $p_k^*$ of (4.3) is the minimizer of $m_k$ in the ball of radius $\Delta_k$. Thus, the trust-region approach requires us to solve a sequence of subproblems (4.3) in which the objective function and constraint (which can be written as $p^T p \le \Delta_k^2$) are both quadratic. When $B_k$ is positive definite and $\|B_k^{-1} \nabla f_k\| \le \Delta_k$, the solution of (4.3) is easy to identify—it is simply the unconstrained minimum $p_k^{\mathrm{B}} = -B_k^{-1} \nabla f_k$ of the quadratic $m_k(p)$. In this case, we call $p_k^{\mathrm{B}}$ the *full step*. The solution of (4.3) is not so obvious in other cases, but it can usually be found without too much expense. In any case, we need only an *approximate* solution to obtain convergence and good practical behavior.

### OUTLINE OF THE ALGORITHM

The first issue to arise in defining a trust-region method is the strategy for choosing the trust-region radius $\Delta_k$ at each iteration. We base this choice on the agreement between the model function $m_k$ and the objective function $f$ at previous iterations. Given a step $p_k$ we define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}; \tag{4.4}$$

the numerator is called the *actual reduction*, and the denominator is the *predicted reduction*. Note that since the step $p_k$ is obtained by minimizing the model $m_k$ over a region that includes the step $p = 0$, the predicted reduction will always be nonnegative. Thus if $\rho_k$ is negative, the new objective value $f(x_k + p_k)$ is greater than the current value $f(x_k)$, so the step must be rejected.

On the other hand, if $\rho_k$ is close to 1, there is good agreement between the model $m_k$ and the function $f$ over this step, so it is safe to expand the trust region for the next iteration. If $\rho_k$ is positive but not close to 1, we do not alter the trust region, but if it is close to zero or negative, we shrink the trust region. The following algorithm describes the process.

**Algorithm 4.1** (Trust Region).
Given $\bar{\Delta} > 0$, $\Delta_0 \in (0, \bar{\Delta})$, and $\eta \in \left[0, \frac{1}{4}\right)$:
**for** $k = 0, 1, 2, \ldots$
  Obtain $p_k$ by (approximately) solving (4.3);
  Evaluate $\rho_k$ from (4.4);
  **if** $\rho_k < \frac{1}{4}$
    $\Delta_{k+1} = \frac{1}{4}\|p_k\|$
  **else**
    **if** $\rho_k > \frac{3}{4}$ and $\|p_k\| = \Delta_k$
      $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$
    **else**
      $\Delta_{k+1} = \Delta_k$;
  **if** $\rho_k > \eta$
    $x_{k+1} = x_k + p_k$
  **else**
    $x_{k+1} = x_k$;
**end (for).**

Here $\bar{\Delta}$ is an overall bound on the step lengths. Note that the radius is increased only if $\|p_k\|$ actually reaches the boundary of the trust region. If the step stays strictly inside the region, we infer that the current value of $\Delta_k$ is not interfering with the progress of the algorithm, so we leave its value unchanged for the next iteration.

To turn Algorithm 4.1 into a practical algorithm, we need to focus on solving (4.3). We first describe three strategies for finding *approximate* solutions, which achieve at least as much reduction in $m_k$ as the reduction achieved by the so-called *Cauchy point*. This point is simply the minimizer of $m_k$ along the steepest descent direction $-\nabla f_k$, subject to the trust-region bound. The first approximate strategy is the *dogleg method*, which is appropriate when the model Hessian $B_k$ is positive definite. The second strategy, known as *two-dimensional subspace minimization*, can be applied when $B_k$ is indefinite, though it requires an estimate of the most negative eigenvalue of this matrix. The third strategy, due to Steihaug, is most appropriate when $B_k$ is the exact Hessian $\nabla^2 f(x_k)$ and when this matrix is large and sparse.

We also describe a strategy due to Moré and Sorensen that finds a "nearly exact" solution of (4.3). This strategy is based on the fact that the solution $p$ satisfies $(B_k + \lambda I)p = -\nabla f_k$ for some positive value of $\lambda > 0$. This strategy seeks the value of $\lambda$ that corresponds to the trust-region radius $\Delta_k$ and performs additional calculations in the special case in which the resulting modified Hessian $(B_k + \lambda I)$ is nonsingular. Details are given below.

## 4.1 THE CAUCHY POINT AND RELATED ALGORITHMS

### THE CAUCHY POINT

As we saw in the previous chapter, line search methods do not require optimal step lengths to be globally convergent. In fact, only a crude approximation to the optimal step length that satisfies certain loose criteria is needed. A similar situation applies in trust-region methods. Although in principle we are seeking the optimal solution of the subproblem (4.3), it is enough for global convergence purposes to find an approximate solution $p_k$ that lies within the trust region and gives a *sufficient reduction* in the model. The sufficient reduction can be quantified in terms of the Cauchy point, which we denote by $p_k^{\mathrm{C}}$ and define in terms of the following simple procedure:

**Algorithm 4.2** (Cauchy Point Calculation).

Find the vector $p_k^{\mathrm{S}}$ that solves a linear version of (4.3), that is,

$$p_k^{\mathrm{S}} = \arg\min_{p \in \mathbb{R}^n} f_k + \nabla f_k^T p \qquad \text{s.t. } \|p\| \le \Delta_k; \tag{4.5}$$

Calculate the scalar $\tau_k > 0$ that minimizes $m_k(\tau p_k^{\mathrm{S}})$ subject to satisfying the trust-region bound, that is,

$$\tau_k = \arg\min_{\tau > 0} m_k(\tau p_k^{\mathrm{S}}) \qquad \text{s.t. } \|\tau p_k^{\mathrm{S}}\| \le \Delta_k; \tag{4.6}$$

Set $p_k^{\mathrm{C}} = \tau_k p_k^{\mathrm{S}}$.

In fact, it is easy to write down a closed-form definition of the Cauchy point. The solution of (4.5) is simply

$$p_k^{\mathrm{S}} = -\frac{\Delta_k}{\|\nabla f_k\|} \nabla f_k.$$

To obtain $\tau_k$ explicitly, we consider the cases of $\nabla f_k^T B_k \nabla f_k \le 0$ and $\nabla f_k^T B_k \nabla f_k > 0$ separately. For the former case, the function $m_k(\tau p_k^{\mathrm{S}})$ decreases monotonically with $\tau$ whenever $\nabla f_k \ne 0$, so $\tau_k$ is simply the largest value that satisfies the trust-region bound, namely, $\tau_k = 1$. For the case $\nabla f_k^T B_k \nabla f_k > 0$, $m_k(\tau p_k^{\mathrm{S}})$ is a convex quadratic in $\tau$, so $\tau_k$ is either

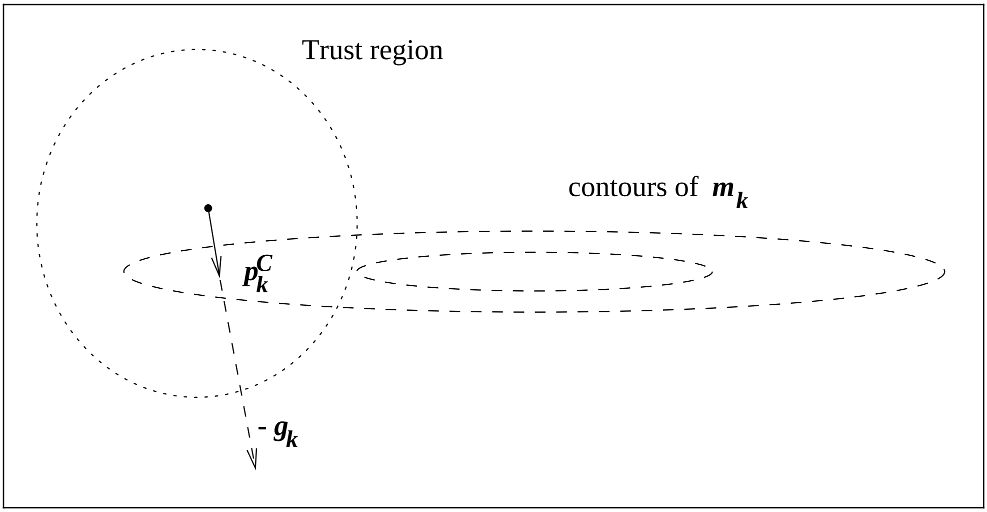


**Figure 4.2** The Cauchy point.

the unconstrained minimizer of this quadratic, $\|\nabla f_k\|^3/(\Delta_k \nabla f_k^T B_k \nabla f_k)$, or the boundary value 1, whichever comes first. In summary, we have

$$p_k^{\mathrm{C}} = -\tau_k \frac{\Delta_k}{\|\nabla f_k\|} \nabla f_k, \tag{4.7}$$

where

$$\tau_k = \begin{cases} 1 & \text{if } \nabla f_k^T B_k \nabla f_k \le 0; \\ \min\left(\|\nabla f_k\|^3/(\Delta_k \nabla f_k^T B_k \nabla f_k), 1\right) & \text{otherwise.} \end{cases} \tag{4.8}$$

Figure 4.2 illustrates the Cauchy point for a subproblem in which $B_k$ is positive definite. In this example, $p_k^{\mathrm{C}}$ lies strictly inside the trust region.

The Cauchy step $p_k^{\mathrm{C}}$ is inexpensive to calculate—no matrix factorizations are required—and is of crucial importance in deciding if an approximate solution of the trust-region subproblem is acceptable. Specifically, a trust-region method will be globally convergent if its steps $p_k$ attain a sufficient reduction in $m_k$; that is, they give a reduction in the model $m_k$ that is at least some fixed multiple of the decrease attained by the Cauchy step at each iteration.

### IMPROVING ON THE CAUCHY POINT

Since the Cauchy point $p_k^{\mathrm{C}}$ provides sufficient reduction in the model function $m_k$ to yield global convergence, and since the cost of calculating it is so small, why should we look any further for a better approximate solution of (4.3)? The reason is that by always taking the Cauchy point as our step, we are simply implementing the steepest descent method with a particular choice of step length. As we have seen in Chapter 3, steepest descent performs poorly even if an *optimal* step length is used at each iteration.

The Cauchy point does not depend very strongly on the matrix $B_k$, which is used only in the calculation of the step length. Rapid convergence (superlinear, for instance) can be expected only if $B_k$ plays a role in determining the *direction* of the step as well as its length.

A number of algorithms for generating approximate solutions $p_k$ to the trust-region problem (4.3) start by computing the Cauchy point and then try to improve on it. The improvement strategy is often designed so that the full step $p_k^{\mathrm{B}} = -B_k^{-1}\nabla f_k$ is chosen whenever $B_k$ is positive definite and $\|p_k^{\mathrm{B}}\| \le \Delta_k$. When $B_k$ is the exact Hessian $\nabla^2 f(x_k)$ or a quasi-Newton approximation, this strategy can be expected to yield superlinear convergence.

We now consider three methods for finding approximate solutions to (4.3) that have the features just described. Throughout this section we will be focusing on the internal workings of a single iteration, so we drop the subscript "$k$" from the quantities $\Delta_k$, $p_k$, and $m_k$ to simplify the notation. With this simplification, we restate the trust-region subproblem (4.3) as follows:

$$\min_{p\in\mathbb{R}^n} m(p) \stackrel{\text{def}}{=} f + g^T p + \tfrac{1}{2} p^T B p \qquad \text{s.t. } \|p\| \le \Delta. \tag{4.9}$$

We denote the solution of (4.9) by $p^*(\Delta)$, to emphasize the dependence on $\Delta$.

### THE DOGLEG METHOD

We start by examining the effect of the trust-region radius $\Delta$ on the solution $p^*(\Delta)$ of the subproblem (4.9). When $B$ is positive definite, we have already noted that the unconstrained minimizer of $m$ is the full step $p^{\mathrm{B}} = -B^{-1}g$. When this point is feasible for (4.9), it is obviously a solution, so we have

$$p^*(\Delta) = p^{\mathrm{B}}, \qquad \text{when } \Delta \ge \|p^{\mathrm{B}}\|. \tag{4.10}$$

When $\Delta$ is tiny, the restriction $\|p\| \le \Delta$ ensures that the quadratic term in $m$ has little effect on the solution of (4.9). The true solution $p(\Delta)$ is approximately the same as the solution we would obtain by minimizing the linear function $f + g^T p$ over $\|p\| \le \Delta$, that is,

$$p^*(\Delta) \approx -\Delta \frac{g}{\|g\|}, \qquad \text{when } \Delta \text{ is small}. \tag{4.11}$$

For intermediate values of $\Delta$, the solution $p^*(\Delta)$ typically follows a curved trajectory like the one in Figure 4.3.

The *dogleg method* finds an approximate solution by replacing the curved trajectory for $p^*(\Delta)$ with a path consisting of two line segments. The first line segment runs from the

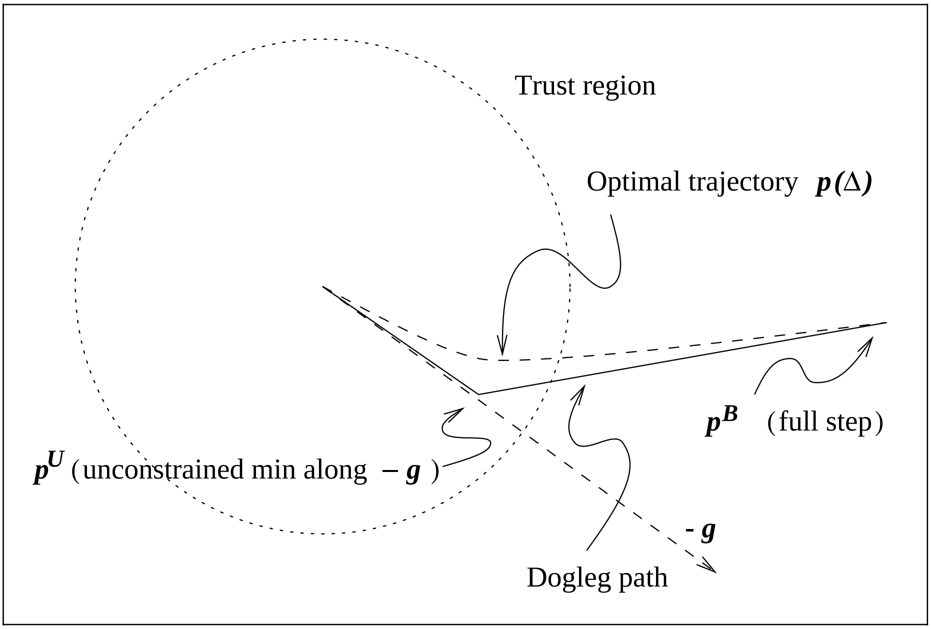


**Figure 4.3** Exact trajectory and dogleg approximation.

origin to the unconstrained minimizer along the steepest descent direction defined by

$$p^{\mathrm{U}} = -\frac{g^T g}{g^T B g} g, \tag{4.12}$$

while the second line segment runs from $p^{\mathrm{U}}$ to $p^{\mathrm{B}}$ (see Figure 4.3). Formally, we denote this trajectory by $\tilde{p}(\tau)$ for $\tau \in [0, 2]$, where

$$\tilde{p}(\tau) = \begin{cases} \tau p^{\mathrm{U}}, & 0 \le \tau \le 1, \\ p^{\mathrm{U}} + (\tau - 1)(p^{\mathrm{B}} - p^{\mathrm{U}}), & 1 \le \tau \le 2. \end{cases} \tag{4.13}$$

The dogleg method chooses $p$ to minimize the model $m$ along this path, subject to the trust-region bound. In fact, it is not even necessary to carry out a search, because the dogleg path intersects the trust-region boundary at most once and the intersection point can be computed analytically. We prove these claims in the following lemma.

**Lemma 4.1.**

*Let $B$ be positive definite. Then*

**(i)** $\|\tilde{p}(\tau)\|$ *is an increasing function of* $\tau$, *and*

**(ii)** $m(\tilde{p}(\tau))$ *is a decreasing function of* $\tau$.

PROOF. It is easy to show that (i) and (ii) both hold for $\tau \in [0, 1]$, so we restrict our attention to the case of $\tau \in [1, 2]$.

For (i), define $h(\alpha)$ by

$$\begin{aligned} h(\alpha) &= \tfrac{1}{2}\|\tilde{p}(1+\alpha)\|^2 \\ &= \tfrac{1}{2}\|p^{\mathrm{U}} + \alpha(p^{\mathrm{B}} - p^{\mathrm{U}})\|^2 \\ &= \tfrac{1}{2}\|p^{\mathrm{U}}\|^2 + \alpha p^{\mathrm{U}T}(p^{\mathrm{B}} - p^{\mathrm{U}}) + \tfrac{1}{2}\alpha^2\|p^{\mathrm{B}} - p^{\mathrm{U}}\|^2. \end{aligned}$$

Our result is proved if we can show that $h'(\alpha) \ge 0$ for $\alpha \in (0, 1)$. Now,

$$\begin{aligned} h'(\alpha) &= -p^{\mathrm{U}T}(p^{\mathrm{U}} - p^{\mathrm{B}}) + \alpha\|p^{\mathrm{U}} - p^{\mathrm{B}}\|^2 \\ &\ge -p^{\mathrm{U}T}(p^{\mathrm{U}} - p^{\mathrm{B}}) \\ &= \frac{g^T g}{g^T B g} g^T \left( -\frac{g^T g}{g^T B g} g + B^{-1} g \right) \\ &= g^T g \frac{g B^{-1} g}{g^T B g} \left[ 1 - \frac{(g^T g)^2}{(g^T B g)(g^T B^{-1} g)} \right] \\ &\ge 0, \end{aligned}$$

where the final inequality follows from Exercise 3.

For (ii), we define $\hat{h}(\alpha) = m(\tilde{p}(1+\alpha))$ and show that $\hat{h}'(\alpha) \le 0$ for $\alpha \in (0, 1)$. Substitution of (4.13) into (4.9) and differentiation with respect to the argument leads to

$$\begin{aligned} \hat{h}'(\alpha) &= (p^{\mathrm{B}} - p^{\mathrm{U}})^T(g + Bp^{\mathrm{U}}) + \alpha(p^{\mathrm{B}} - p^{\mathrm{U}})^T B (p^{\mathrm{B}} - p^{\mathrm{U}}) \\ &\le (p^{\mathrm{B}} - p^{\mathrm{U}})^T(g + Bp^{\mathrm{U}} + B(p^{\mathrm{B}} - p^{\mathrm{U}})) \\ &= (p^{\mathrm{B}} - p^{\mathrm{U}})^T(g + Bp^{\mathrm{B}}) = 0, \end{aligned}$$

giving the result. $\square$

It follows from this lemma that the path $\tilde{p}(\tau)$ intersects the trust-region boundary $\|p\| = \Delta$ at exactly one point if $\|p^{\mathrm{B}}\| \ge \Delta$, and nowhere otherwise. Since $m$ is decreasing along the path, the chosen value of $p$ will be at $p^{\mathrm{B}}$ if $\|p^{\mathrm{B}}\| \le \Delta$, otherwise at the point of intersection of the dogleg and the trust-region boundary. In the latter case, we compute the appropriate value of $\tau$ by solving the following scalar quadratic equation:

$$\|p^{\mathrm{U}} + (\tau - 1)(p^{\mathrm{B}} - p^{\mathrm{U}})\|^2 = \Delta^2.$$

The dogleg strategy can be adapted to handle indefinite $B$, but there is not much point in doing so because the full step $p^{\mathrm{B}}$ is not the unconstrained minimizer of $m$ in this case. Instead, we now describe another strategy, which aims to include directions of negative curvature (that is, directions $d$ for which $d^T B d < 0$) in the space of candidate trust-region steps.

### TWO-DIMENSIONAL SUBSPACE MINIMIZATION

When $B$ is positive definite, the dogleg method strategy can be made slightly more sophisticated by widening the search for $p$ to the entire two-dimensional subspace spanned by $p^{\mathrm{U}}$ and $p^{\mathrm{B}}$ (equivalently, $g$ and $-B^{-1}g$). The subproblem (4.9) is replaced by

$$\min_p m(p) = f + g^T p + \tfrac{1}{2} p^T B p \quad \text{s.t. } \|p\| \le \Delta, \; p \in \operatorname{span}[g, B^{-1}g]. \tag{4.14}$$

This is a problem in two variables that can be solved without much effort (see the exercises). Clearly, the Cauchy point $p^{\mathrm{C}}$ is feasible for (4.14), so the optimal solution of this subproblem yields at least as much reduction in $m$ as the Cauchy point, resulting in global convergence of the algorithm. The two-dimensional subspace minimization strategy is obviously an extension of the dogleg method as well, since the entire dogleg path lies in $\operatorname{span}[g, B^{-1}g]$.

An advantage of this strategy is that it can be modified to handle the case of indefinite $B$ in a way that is intuitive, practical, and theoretically sound. We mention just the salient points of the handling of the indefiniteness here, and refer the reader to papers by Byrd, Schnabel, and Schultz (see [39] and [226]) for details. When $B$ has negative eigenvalues, the two-dimensional subspace in (4.14) is changed to

$$\operatorname{span}[g, (B + \alpha I)^{-1} g], \qquad \text{for some } \alpha \in (-\lambda_1, -2\lambda_1], \tag{4.15}$$

where $\lambda_1$ denotes the most negative eigenvalue of $B$. (This choice of $\alpha$ ensures that $B + \alpha I$ is positive definite, and the flexibility in this definition allows us to use a numerical procedure such as the Lanczos method to compute an acceptable value of $\alpha$.) When $\|(B+\alpha I)^{-1} g\| \le \Delta$, we discard the subspace search of (4.14), (4.15) and instead define the step to be

$$p = -(B + \alpha I)^{-1} g + v, \tag{4.16}$$

where $v$ is a vector that satisfies $v^T (B + \alpha I)^{-1} g \le 0$. (This condition ensures that $v$ does not move $p$ back toward zero, but instead continues to move roughly in the direction of $-(B + \alpha I)^{-1} g$).

When $B$ has zero eigenvalues but no negative eigenvalues, the Cauchy step $p = p^{\mathrm{C}}$ is used as the approximate solution of (4.9).

The reduction in model function $m$ achieved by the two-dimensional minimization strategy often is close to the reduction achieved by the exact solution of (4.9). Most of the computational effort lies in a single factorization of $B$ or $B + \alpha I$ (estimation of $\alpha$ and solution of (4.14) are less significant), while strategies that find nearly exact solutions of (4.9) typically require two or three such factorizations.

### STEIHAUG'S APPROACH

Both methods described above require the solution of a single linear system involving $B$ or $(B + \alpha I)$. When $B$ is large, this operation may be quite costly, so we are motivated to consider other techniques for finding an approximate solution of (4.9) that do not require exact solution of a linear system but still produce an improvement on the Cauchy point. Steihaug [231] proposed a technique with these properties. Steihaug's implementation is based on the conjugate gradient algorithm, an iterative algorithm for solving linear systems with symmetric positive definite coefficient matrices. The conjugate gradient (CG) algorithm is the subject of Chapter 5, and the interested reader should look ahead to that chapter for further details. Our comments in this section focus on the differences between standard CG and Steihaug's approach, which are essentially that the algorithm terminates when it either exits the trust region $\|p\| \le \Delta$ or when it encounters a direction of negative curvature in $B$.

Steihaug's approach can be stated formally as follows:

**Algorithm 4.3** (CG–Steihaug).

Given $\epsilon > 0$;
Set $p_0 = 0, r_0 = g, d_0 = -r_0$;
**if** $\|r_0\| < \epsilon$
    **return** $p = p_0$;
**for** $j = 0, 1, 2, \ldots$
    **if** $d_j^T B d_j \le 0$
        Find $\tau$ such that $p = p_j + \tau d_j$ minimizes $m(p)$ in (4.9)
            and satisfies $\|p\| = \Delta$;
        **return** $p$;
    Set $\alpha_j = r_j^T r_j / d_j^T B d_j$;
    Set $p_{j+1} = p_j + \alpha_j d_j$;
    **if** $\|p_{j+1}\| \ge \Delta$
        Find $\tau \ge 0$ such that $p = p_j + \tau d_j$ satisfies $\|p\| = \Delta$;
        **return** $p$;
    Set $r_{j+1} = r_j + \alpha_j B d_j$;
    **if** $\|r_{j+1}\| < \epsilon \|r_0\|$
        **return** $p = p_{j+1}$;
    Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$;
    Set $d_{j+1} = r_{j+1} + \beta_{j+1} d_j$;
**end (for)**.

To connect this algorithm with Algorithm CG of Chapter 5, we note that $m(\cdot)$ takes the place of $\phi(\cdot)$, $p$ takes the place of $x$, $B$ takes the place of $A$, and $-g$ takes the place of $b$. The change of sign in the substitution $b \to -g$ propagates through the algorithm.

Algorithm 4.3 differs from standard CG in that two extra stopping criteria are present—the first two **if** statements inside the **for** loop. The first **if** statement stops the method if its current search direction $d_j$ is a direction of zero curvature or negative curvature along $B$. The second one causes termination if $p_{j+1}$ violates the trust-region bound. In both cases, a final point $p$ is found by intersecting the current search direction with the trust-region boundary.

The initialization of $p_0$ to zero is a crucial feature of the algorithm. After the first iteration (assuming $\|r_0\|_2 \geq \epsilon$), we have

$$p_1 = \alpha_0 d_0 = \frac{r_0^T r_0}{d_0^T B d_0} d_0 = -\frac{g^T g}{g^T B g} g,$$

which is exactly the Cauchy point! Since each iteration of the conjugate gradient method reduces $m(\cdot)$, this algorithm fulfills the necessary condition for global convergence.

Another crucial property of the method is that each iterate $p_j$ is larger in norm than its predecessor. This property is another consequence of the initialization $p_0 = 0$. Its main implication is that it is acceptable to stop iterating as soon as the trust-region boundary is reached, because no further iterates giving a lower value of $\phi$ will be inside the trust region. We state and prove this property formally in the following theorem. (The proof makes use of the expanding subspace property of the CG algorithm, which we do not describe until Chapter 5, so it can be skipped on the first pass.)

**Theorem 4.2.**

*The sequence of vectors generated by Algorithm 4.3 satisfies*

$$0 = \|p_0\|_2 < \cdots < \|p_j\|_2 < \|p_{j+1}\|_2 < \cdots < \|p\|_2 \leq \Delta.$$

PROOF. We first show that the sequences of vectors generated by Algorithm 4.3 satisfy $p_j^T r_j = 0$ for $j \geq 0$ and $p_j^T d_j > 0$ for $j \geq 1$.

Algorithm 4.3 computes $p_{j+1}$ recursively in terms of $p_j$, but when all the terms of this recursion are written explicitly, we see that

$$p_j = p_0 + \sum_{i=0}^{j-1} \alpha_i d_i = \sum_{i=0}^{j-1} \alpha_i d_i,$$

since $p_0 = 0$. Multiplying by $r_j$ and applying the expanding subspace property of CG gives

$$p_j^T r_j = \sum_{i=0}^{j-1} \alpha_i d_i^T r_j = 0.$$

An induction proof establishes the relation $p_j^T d_j > 0$. By applying the expanding subspace property again, we obtain

$$p_1^T d_1 = (\alpha_0 d_0)^T (r_1 + \beta_1 d_0) = \alpha_0 \beta_1 \, d_0^T d_0 > 0. \tag{4.17}$$

We now make the inductive hypothesis that $p_j^T d_j > 0$ and show that this implies $p_{j+1}^T d_{j+1} > 0$. From (4.17), we have $p_{j+1}^T r_{j+1} = 0$, and therefore we have

$$\begin{aligned} p_{j+1}^T d_{j+1} &= p_{j+1}^T (r_{j+1} + \beta_{j+1} d_j) \\ &= \beta_{j+1} p_{j+1}^T d_j \\ &= \beta_{j+1} (p_j + \alpha_j d_j)^T d_j \\ &= \beta_{j+1} p_j^T d_j + \alpha_j \beta_{j+1} d_j^T d_j. \end{aligned}$$

Because of the inductive hypothesis, the last expression is positive.

We now prove the theorem. If the algorithm stops because $d_j^T B d_j \leq 0$ or $\|p_{j+1}\|_2 \geq \Delta$, then the final point $p$ is chosen to make $\|p\|_2 = \Delta$, which is the largest possible length any point can have. To cover all other possibilities in the algorithm we must show that $\|p_j\|_2 < \|p_{j+1}\|_2$ when $p_{j+1} = p_j + \alpha_j d_j$ and $j \geq 1$. Observe that

$$\|p_{j+1}\|_2^2 = (p_j + \alpha_j d_j)^T (p_j + \alpha_j d_j) = \|p_j\|_2^2 + 2\alpha_j p_j^T d_j + \alpha_j^2 \|d_j\|_2^2.$$

It follows from this expression and our intermediate result that $\|p_j\|_2 < \|p_{j+1}\|_2$, so our proof is complete. □

From this theorem we see that the iterates of Algorithm 4.3 sweep out points $p_j$ that move on some interpolating path from $p_1$ to $p$, a path in which every step increases its total distance from the start point. When $B$ is positive definite, this path may be compared to the path of the dogleg method, because both methods move from the Cauchy step $p^{\mathrm{C}}$ to the full step $p^{\mathrm{B}}$, until the trust-region boundary intervenes.

A Newton trust-region method chooses $B$ to be the exact Hessian $\nabla^2 f(x)$, which may be indefinite during the course of the iteration (hence our focus on the case of indefinite $B$). This method has excellent local and global convergence properties, as we see in Chapter 6.

## 4.2 USING NEARLY EXACT SOLUTIONS TO THE SUBPROBLEM

### CHARACTERIZING EXACT SOLUTIONS

The methods discussed above make no serious attempt to seek the exact solution of the subproblem (4.9). They do, however, make some use of the information in the Hessian $B$, and they have advantages of low cost and global convergence, since they all generate a point that is at least as good as the Cauchy point.

When the problem is relatively small (that is, $n$ is not too large), it may be worthwhile to exploit the model more fully by looking for a closer approximation to the solution of the subproblem. In the next few pages we describe an approach for finding a good approximation at the cost of about three factorizations of the matrix $B$, as compared with a single factorization for the dogleg and two-dimensional subspace minimization methods. This approach is based on a convenient characterization of the exact solution of (4.9) (we need to be able to recognize an exact solution when we see it, after all) and an ingenious application of Newton's method in one variable. Essentially, we see that a solution $p$ of the trust-region problem satisfies the formula

$$(B + \lambda I)p^* = -g$$

for some $\lambda \geq 0$, and our algorithm for finding $p^*$ aims to identify the appropriate value of $\lambda$.

The following theorem gives the precise characterization of the solution of (4.9).

**Theorem 4.3.**

*The vector $p^*$ is a global solution of the trust-region problem*

$$\min_{p \in \mathbb{R}^n} m(p) = f + g^T p + \tfrac{1}{2} p^T B p, \qquad \text{s.t. } \|p\| \leq \Delta, \tag{4.18}$$

*if and only if $p^*$ is feasible and there is a scalar $\lambda \geq 0$ such that the following conditions are satisfied:*

$$(B + \lambda I)p^* = -g, \tag{4.19a}$$

$$\lambda(\Delta - ||p^*||) = 0, \tag{4.19b}$$

$$(B + \lambda I) \quad \text{is positive semidefinite.} \tag{4.19c}$$

We delay the proof of this result until later in the chapter, and instead discuss just its key features here with the help of Figure 4.4. The condition (4.19b) is a complementarity condition that states that at least one of the nonnegative quantities $\lambda$ and $(\Delta - \|p^*\|)$ must be zero. Hence, when the solution lies strictly inside the trust region (as it does when $\Delta = \Delta_1$ in Figure 4.4), we must have $\lambda = 0$ and so $Bp^* = -g$ with $B$ positive semidefinite, from (4.19a) and (4.19c), respectively. In the other cases $\Delta = \Delta_2$ and $\Delta = \Delta_3$, we have $\|p^*\| = \Delta$, and so $\lambda$ is allowed to take a positive value. Note from (4.19a) that

$$\lambda p^* = -Bp^* - g = -\nabla m(p^*),$$

that is, the solution $p^*$ is collinear with the negative gradient of $m$ and normal to its contours. These properties can be seen clearly in Figure 4.4.

### CALCULATING NEARLY EXACT SOLUTIONS

The characterization of Theorem 4.3 suggests an algorithm for finding the solution $p$ of (4.18). Either $\lambda = 0$ satisfies (4.19a) and (4.19c) with $\|p\| \leq \Delta$, or else we define

$$p(\lambda) = -(B + \lambda I)^{-1} g$$

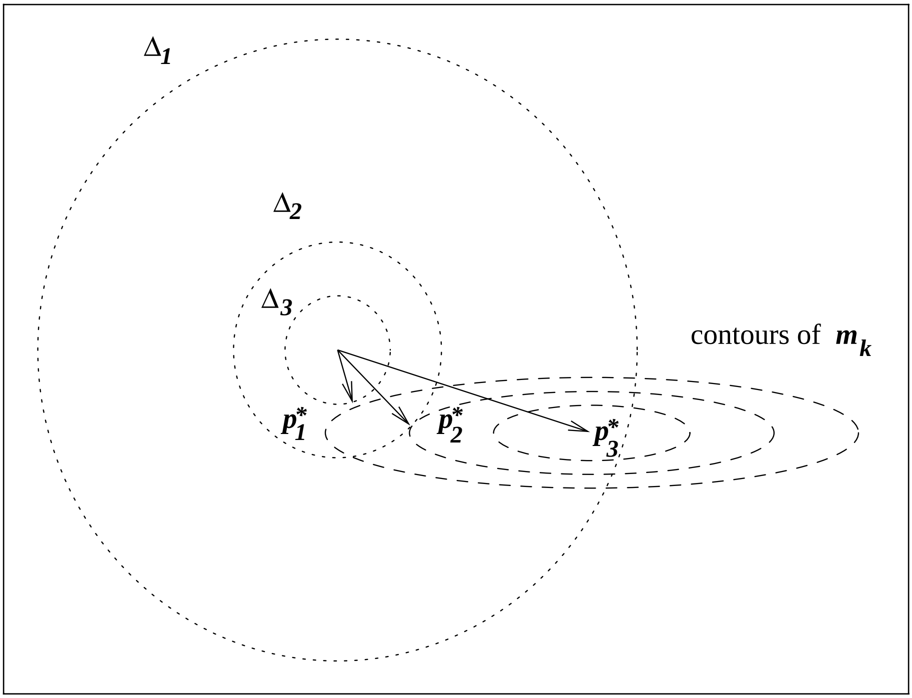


**Figure 4.4** Solution of trust-region subproblem for different radii $\Delta_1$, $\Delta_2$, $\Delta_3$.

for $\lambda$ sufficiently large that $B + \lambda I$ is positive definite (see the exercises), and seek a value $\lambda > 0$ such that

$$\|p(\lambda)\| = \Delta. \tag{4.20}$$

This problem is a one-dimensional root-finding problem in the variable $\lambda$.

To see that a value of $\lambda$ with all the desired properties exists, we appeal to the eigendecomposition of $B$ and use it to study the properties of $\|p(\lambda)\|$. Since $B$ is symmetric, there is an orthogonal matrix $Q$ and a diagonal matrix $\Lambda$ such that $B = Q\Lambda Q^T$, where

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n),$$

and $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ are the eigenvalues of $B$; see (A.46). Clearly, $B + \lambda I = Q(\Lambda + \lambda I)Q^T$, and for $\lambda \ne \lambda_j$, we have

$$p(\lambda) = -Q(\Lambda + \lambda I)^{-1} Q^T g = -\sum_{j=1}^{n} \frac{q_j^T g}{\lambda_j + \lambda} q_j, \tag{4.21}$$

where $q_j$ denotes the $j$th column of $Q$. Therefore, by orthonormality of $q_1, q_2, \ldots, q_n$, we have

$$\|p(\lambda)\|^2 = \sum_{j=1}^{n} \frac{\left(q_j^T g\right)^2}{(\lambda_j + \lambda)^2}. \tag{4.22}$$

This expression tells us a lot about $\|p(\lambda)\|$. If $\lambda > -\lambda_1$, we have $\lambda_j + \lambda > 0$ for all $j = 1, 2, \ldots, n$, and so $\|p(\lambda)\|$ is a continuous, nonincreasing function of $\lambda$ on the interval $(-\lambda_1, \infty)$. In fact, we have that

$$\lim_{\lambda \to \infty} \|p(\lambda)\| = 0. \tag{4.23}$$

Moreover, we have when $q_j^T g \neq 0$ that

$$\lim_{\lambda \to -\lambda_j} \|p(\lambda)\| = \infty. \tag{4.24}$$

These features can be seen in Figure 4.5. It is clear that the graph of $\|p(\lambda)\|$ attains the value $\Delta$ at exactly one point in the interval $(-\lambda_1, \infty)$, which is denoted by $\lambda^*$ in the figure. For the case of $B$ positive definite and $\|B^{-1}g\| \leq \Delta$, the value $\lambda = 0$ satisfies (4.19), so there

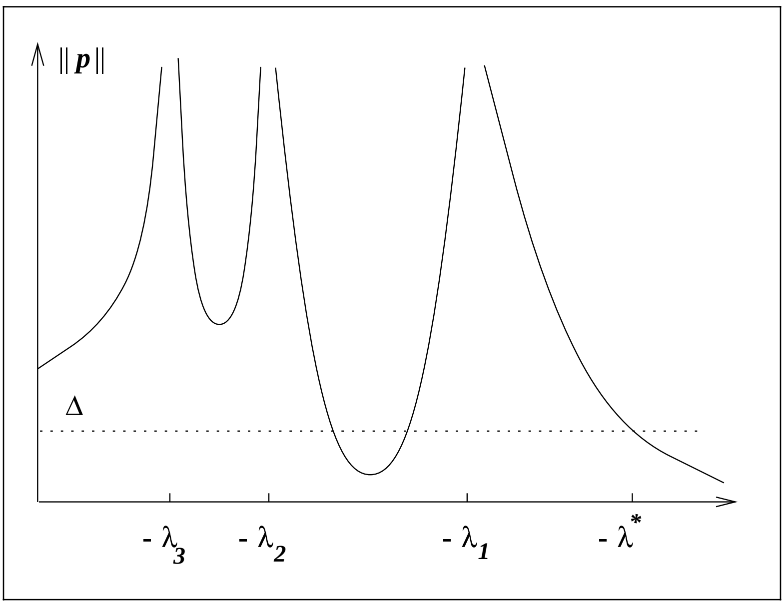


**Figure 4.5** $\|p(\lambda)\|$ as a function of $\lambda$.

is no need to carry out a search. When $B$ is positive definite but $\|B^{-1}g\| > \Delta$, there is a strictly positive value of $\lambda$ for which $\|p(\lambda)\| = \Delta$, so we search for the solution to (4.20) in the interval $(0, \infty)$.

For the case of $B$ indefinite and $q_1^T g \neq 0$, (4.23) and (4.24) guarantee that we can find a solution in the interval $(-\lambda_1, \infty)$. We could use the root-finding Newton's method (see the Appendix) to find the value of $\lambda > \lambda_1$ that solves

$$\phi_1(\lambda) = \|p(\lambda)\| - \Delta = 0. \tag{4.25}$$

The disadvantage of this approach can be seen by considering the form of $\|p(\lambda)\|$ when $\lambda$ is greater than, but close to, $-\lambda_1$. We then have

$$\phi_1(\lambda) \approx \frac{C_1}{\lambda + \lambda_1} + C_2,$$

where $C_1 > 0$ and $C_2$ are constants. For these values of $\lambda$ the function is highly nonlinear, and therefore the root-finding Newton's method will be unreliable or slow. Better results will be obtained if we reformulate the problem (4.25) so that it is nearly linear near the optimal $\lambda$. By defining

$$\phi_2(\lambda) = \frac{1}{\Delta} - \frac{1}{\|p(\lambda)\|},$$

we see that for $\lambda$ slightly greater than $-\lambda_1$ we have from (4.22) that

$$\phi_2(\lambda) \approx \frac{1}{\Delta} - \frac{\lambda + \lambda_1}{C_3}$$

for some $C_3 > 0$. Hence $\phi_2$ is nearly linear in the range we consider, and the root-finding Newton's method will perform well, provided that it maintains $\lambda > -\lambda_1$ (see Figure 4.6). The root-finding Newton's method applied to $\phi_2$ generates a sequence of iterates $\lambda^{(\ell)}$ by setting

$$\lambda^{(\ell+1)} = \lambda^{(\ell)} - \frac{\phi_2\left(\lambda^{(\ell)}\right)}{\phi_2'\left(\lambda^{(\ell)}\right)}. \tag{4.26}$$

After some elementary manipulation (see the exercises), this updating formula can be implemented in the following practical way.

**Algorithm 4.4** (Exact Trust Region).

Given $\lambda^{(0)}$, $\Delta > 0$:

**for** $\ell = 0, 1, 2, \ldots$

Factor $B + \lambda^{(\ell)} I = R^T R$;

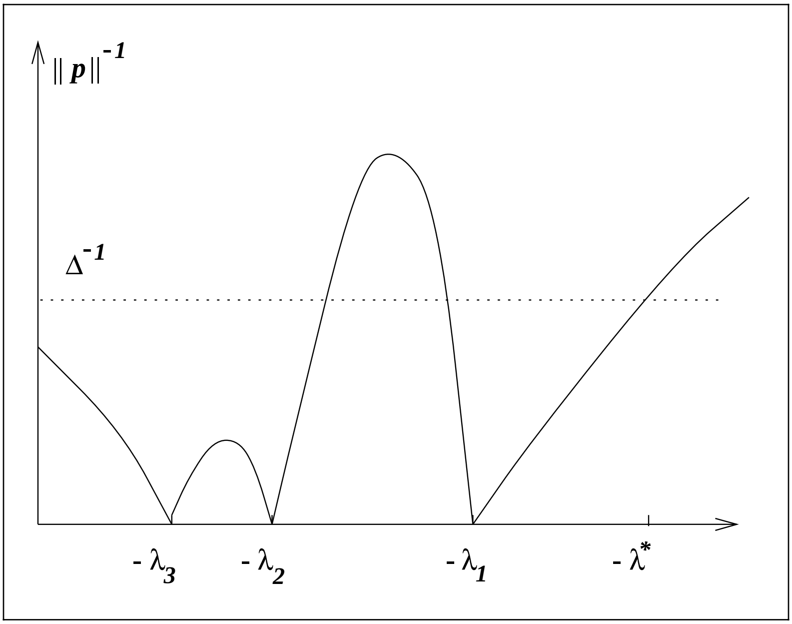


**Figure 4.6** $1/\|p(\lambda)\|$ as a function of $\lambda$.

Solve $R^T R p_\ell = -g$, $R^T q_\ell = p_\ell$;
Set

$$\lambda^{(\ell+1)} = \lambda^{(\ell)} + \left(\frac{\|p_\ell\|}{\|q_\ell\|}\right)^2 \left(\frac{\|p_\ell\| - \Delta}{\Delta}\right); \tag{4.27}$$

**end (for).**

Safeguards must be added to this algorithm to make it practical; for instance, when $\lambda^{(\ell)} < -\lambda_1$, the Cholesky factorization $B + \lambda^{(\ell)} I = R^T R$ will not exist. A slightly enhanced version of this algorithm does, however, converge to a solution of (4.20) in most cases.

The main work in each iteration of this method is, of course, the Cholesky factorization. Practical versions of this algorithm do not iterate until convergence to the optimal $\lambda$ is obtained with high accuracy, but are content with an approximate solution that can be obtained in two or three iterations.

### THE HARD CASE

Recall that in the discussion above, we assumed that $q_1^T g \neq 0$ in dealing with the case of indefinite $B$. In fact, the approach described above can be applied even when the most

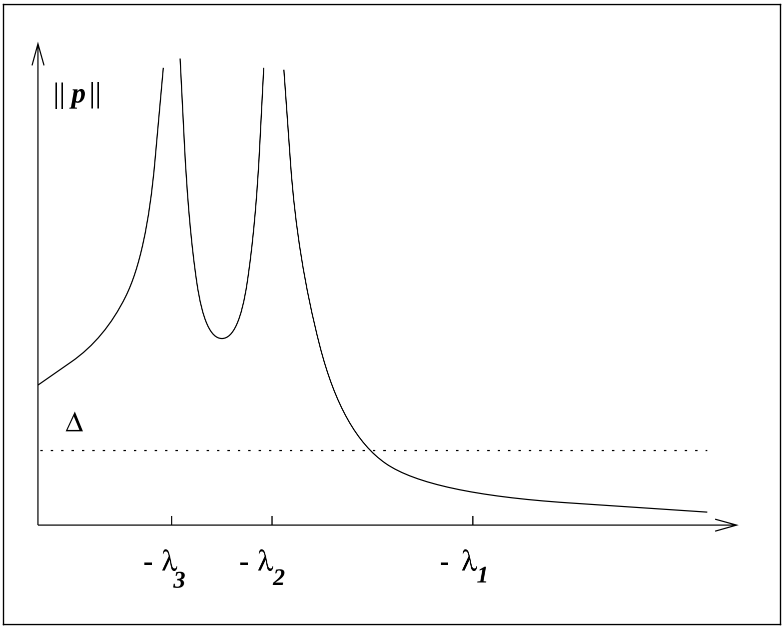


**Figure 4.7** The hard case: $\|p(\lambda)\| < \Delta$ for all $\lambda \in (-\lambda_1, \infty)$.

negative eigenvalue is a multiple eigenvalue (that is, $0 > \lambda_1 = \lambda_2 = \cdots$), provided that $q_j^T g \neq 0$ for at least one of the indices $j$ for which $\lambda_j = \lambda_1$. When this condition does not hold, the situation becomes a little complicated, because the limit (4.24) does not hold for $\lambda_j = \lambda_1$ and so there may not be a value $\lambda \in (-\lambda_1, \infty)$ such that $\|p(\lambda)\| = \Delta$ (see Figure 4.7). Moré and Sorensen [170] refer to this case as the *hard case*. At first glance, it is not clear how $p$ and $\lambda$ can be chosen to satisfy (4.19) in the hard case. Clearly, our root-finding technique will not work, since there is no solution for $\lambda$ in the open interval $(-\lambda_1, \infty)$. But Theorem 4.3 assures us that the right value of $\lambda$ lies in the interval $[-\lambda_1, \infty)$, so there is only one possibility: $\lambda = -\lambda_1$. To find $p$, it is not enough to delete the terms for which $\lambda_j = \lambda_1$ from the formula (4.21) and set

$$p = \sum_{j:\lambda_j \neq \lambda_1} \frac{q_j^T g}{\lambda_j + \lambda} q_j.$$

Instead, we note that $(B - \lambda_1 I)$ is singular, so there is a vector $z$ such that $\|z\| = 1$ and $(B - \lambda_1 I)z = 0$. In fact, $z$ is an eigenvector of $B$ corresponding to the eigenvalue $\lambda_1$, so by orthogonality of $Q$ we have $q_j^T z = 0$ for $\lambda_j \neq \lambda_1$. It follows from this property that if we set

$$p = \sum_{j:\lambda_j \neq \lambda_1} \frac{q_j^T g}{\lambda_j + \lambda} q_j + \tau z \tag{4.28}$$

for any scalar $\tau$, we have

$$\|p\|^2 = \sum_{j:\lambda_j \neq \lambda_1} \frac{\left(q_j^T g\right)^2}{(\lambda_j + \lambda)^2} + \tau^2,$$

so it is always possible to choose $\tau$ to ensure that $\|p\| = \Delta$. It is easy to check that (4.19) holds for this choice of $p$ and $\lambda = -\lambda_1$.

### PROOF OF THEOREM 4.3

We now give a formal proof of Theorem 4.3, the result that characterizes the exact solution of (4.9). The proof relies on the following technical lemma, which deals with the unconstrained minimizers of quadratics and is particularly interesting in the case where the Hessian is positive semidefinite.

**Lemma 4.4.**

*Let $m$ be the quadratic function defined by*

$$m(p) = g^T p + \tfrac{1}{2} p^T B p, \tag{4.29}$$

*where $B$ is any symmetric matrix. Then*

**(i)** *$m$ attains a minimum if and only if $B$ is positive semidefinite and $g$ is in the range of $B$;*

**(ii)** *$m$ has a unique minimizer if and only if $B$ is positive definite;*

**(iii)** *if $B$ is positive semidefinite, then every $p$ satisfying $Bp = -g$ is a global minimizer of $m$.*

PROOF. We prove each of the three claims in turn.

(i) We start by proving the "if" part. Since $g$ is in the range of $B$, there is a $p$ with $Bp = -g$. For all $w \in R^n$, we have

$$\begin{aligned} m(p + w) &= g^T (p + w) + \tfrac{1}{2}(p + w)^T B(p + w) \\ &= (g^T p + \tfrac{1}{2} p^T Bp) + g^T w + (Bp)^T w + \tfrac{1}{2} w^T Bw \\ &= m(p) + \tfrac{1}{2} w^T Bw \\ &\geq m(p), \end{aligned}$$

since $B$ is positive semidefinite. Hence $p$ is a minimum of $m$.

For the "only if" part, let $p$ be a minimizer of $m$. Since $\nabla m(p) = Bp + g = 0$, we have that $g$ is in the range of $B$. Also, we have $\nabla^2 m(p) = B$ positive semidefinite, giving the result.

(ii) For the "if" part, the same argument as in (i) suffices with the additional point that $w^T B w > 0$ whenever $w \neq 0$. For the "only if" part, we proceed as in (i) to deduce that $B$ is positive semidefinite. If $B$ is not positive definite, there is a vector $w \neq 0$ such that $Bw = 0$. Hence from the logic above we have $m(p + w) = m(p)$, so the minimizer is not unique, giving a contradiction.

(iii) Follows from the proof of (i). □

To illustrate case (i), suppose that

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

which has eigenvalues $0, 1, 2$ and is therefore singular. If $g$ is any vector whose second component is zero, then $g$ will be in the range of $B$, and the quadratic will attain a minimum. But if the second element in $g$ is nonzero, we can decrease $m(\cdot)$ indefinitely by moving along the direction $\alpha(0, -g_2, 0)^T$ as $\alpha \uparrow \infty$.

We are now in a position to take account of the trust-region bound $\|p\| \leq \Delta$ and hence prove Theorem 4.3.

PROOF. (Theorem 4.3)

Assume first that there is $\lambda \geq 0$ such that the conditions (4.19) are satisfied. Lemma 4.4(iii) implies that $p^*$ is a global minimum of the quadratic function

$$\hat{m}(p) = g^T p + \tfrac{1}{2} p^T (B + \lambda I) p = m(p) + \frac{\lambda}{2} p^T p. \tag{4.30}$$

Since $\hat{m}(p) \geq \hat{m}(p^*)$, we have

$$m(p) \geq m(p^*) + \frac{\lambda}{2}((p^*)^T p^* - p^T p). \tag{4.31}$$

Because $\lambda(\Delta - \|p^*\|) = 0$ and therefore $\lambda(\Delta^2 - (p^*)^T p^*) = 0$, we have

$$m(p) \geq m(p^*) + \frac{\lambda}{2}(\Delta^2 - p^T p).$$

Hence, from $\lambda \geq 0$, we have $m(p) \geq m(p^*)$ for all $p$ with $\|p\| \leq \Delta$. Therefore, $p^*$ is a global minimizer of (4.18).

For the converse, we assume that $p^*$ is a global solution of (4.18) and show that there is a $\lambda \geq 0$ that satisfies (4.19).

In the case $\|p^*\| < \Delta$, $p^*$ is an unconstrained minimizer of $m$, and so

$$\nabla m(p^*) = Bp^* + g = 0, \qquad \nabla^2 m(p^*) = B \text{ positive semidefinite},$$

and so the properties (4.19) hold for $\lambda = 0$.

Assume for the remainder of the proof that $\|p^*\| = \Delta$. Then (4.19b) is immediately satisfied, and $p^*$ also solves the constrained problem

$$\min m(p) \quad \text{subject to } \|p\| = \Delta.$$

By applying optimality conditions for constrained optimization to this problem (see (12.30)), we find that there is a $\lambda$ such that the Lagrangian function defined by

$$\mathcal{L}(p, \lambda) = m(p) + \frac{\lambda}{2}(p^T p - \Delta^2)$$

has a stationary point at $p^*$. By setting $\nabla_p \mathcal{L}(p^*, \lambda)$ to zero, we obtain

$$Bp^* + g + \lambda p^* = 0 \quad \Rightarrow (B + \lambda I)p^* = -g, \tag{4.32}$$

so that (4.19a) holds. Since $m(p) \ge m(p^*)$ for any $p$ with $p^T p = (p^*)^T p^* = \Delta^2$, we have for such vectors $p$ that

$$m(p) \ge m(p^*) + \frac{\lambda}{2}\left((p^*)^T p^* - p^T p\right).$$

If we substitute the expression for $g$ from (4.32) into this expression, we obtain after some rearrangement that

$$\tfrac{1}{2}(p - p^*)^T (B + \lambda I)(p - p^*) \ge 0. \tag{4.33}$$

Since the set of directions

$$\left\{ w \,:\, w = \pm \frac{p - p^*}{\|p - p^*\|}, \text{ for some } p \text{ with } \|p\| = \Delta \right\}$$

is dense on the unit sphere, (4.33) suffices to prove (4.19c).

It remains to show that $\lambda \ge 0$. Because (4.19a) and (4.19c) are satisfied by $p^*$, we have from Lemma 4.4(i) that $p^*$ minimizes $\hat{m}$, so (4.31) holds. Suppose that there are only negative values of $\lambda$ that satisfy (4.19a) and (4.19c). Then we have from (4.31) that $m(p) \ge m(p^*)$ whenever $\|p\| \ge \|p^*\| = \Delta$. Since we already know that $p^*$ minimizes $m$ for $\|p\| \le \Delta$, it follows that $m$ is in fact a global, unconstrained minimizer of $m$. From Lemma 4.4(i) it follows that $Bp = -g$ and $B$ is positive semidefinite. Therefore conditions (4.19a) and

(4.19c) are satisfied by $\lambda = 0$, which contradicts our assumption that only negative values of $\lambda$ can satisfy the conditions. We conclude that $\lambda \geq 0$, completing the proof. □

## 4.3 GLOBAL CONVERGENCE

### REDUCTION OBTAINED BY THE CAUCHY POINT

In the preceding discussion of algorithms for approximately solving the trust-region subproblem, we have repeatedly emphasized that global convergence depends on the approximate solution obtaining at least as much decrease in the model function $m$ as the Cauchy point. In fact, a fixed fraction of the Cauchy decrease suffices, as we show in the next few pages. We start by obtaining an estimate of the decrease in $m$ achieved by the Cauchy point, and then use this estimate to prove that the sequence of gradients $\{\nabla f_k\}$ generated by Algorithm 4.1 either has an accumulation point at zero or else converges to zero, depending on whether we choose the parameter $\eta$ to be zero or strictly positive in Algorithm 4.1. Finally, we state a convergence result for the version of Algorithm 4.1 that uses the nearly exact solutions calculated by Algorithm 4.4 above.

We start by proving that the dogleg and two-dimensional subspace minimization algorithms and Algorithm 4.3 produce approximate solutions $p_k$ of the subproblem (4.3) that satisfy the estimate

$$m_k(0) - m_k(p_k) \geq c_1 \|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right), \tag{4.34}$$

for some constant $c_1 \in (0, 1]$. The presence of an alternative given by the minimum in (4.34) is typical of trust-region methods and arises because of the trust-region bound. The usefulness of this estimate will become clear in the following two sections. For now, we note that when $\Delta_k$ is the minimum value in (4.34), the condition is slightly reminiscent of the first Wolfe condition: The desired reduction in the model is proportional to the gradient and the size of the step.

We show now that the Cauchy point $p_k^{\mathrm{C}}$ satisfies (4.34), with $c_1 = \frac{1}{2}$.

**Lemma 4.5.**

*The Cauchy point $p_k^{\mathrm{C}}$ satisfies (4.34) with $c_1 = \frac{1}{2}$, that is,*

$$m_k(0) - m_k(p_k^{\mathrm{C}}) \geq \tfrac{1}{2} \|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right). \tag{4.35}$$

PROOF. We consider first the case of $\nabla f_k^T B_k \nabla f_k \leq 0$. Here, we have

$$m_k(p_k^{\mathrm{C}}) - m_k(0) = m_k(\Delta_k \nabla f_k / \|\nabla f_k\|)$$

$$
\begin{aligned}
&= -\frac{\Delta_k}{\|\nabla f_k\|}\|\nabla f_k\|^2 + \tfrac{1}{2}\frac{\Delta_k^2}{\|\nabla f_k\|^2}\nabla f_k^T B_k \nabla f_k \\
&\le -\Delta_k \|\nabla f_k\| \\
&\le -\|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right),
\end{aligned}
$$

and so (4.35) certainly holds.

For the next case, consider $\nabla f_k^T B_k \nabla f_k > 0$ and

$$
\frac{\|\nabla f_k\|^3}{\Delta_k \nabla f_k^T B_k \nabla f_k} \le 1. \tag{4.36}
$$

We then have $\tau = \|\nabla f_k\|^3 / \left(\Delta_k \nabla f_k^T B_k \nabla f_k\right)$, and so

$$
\begin{aligned}
m_k(p_k^{\mathrm{C}}) - m_k(0) &= -\frac{\|\nabla f_k\|^4}{\nabla f_k^T B_k \nabla f_k} + \tfrac{1}{2}\nabla f_k^T B_k \nabla f_k \frac{\|\nabla f_k\|^4}{(\nabla f_k^T B_k \nabla f_k)^2} \\
&= -\tfrac{1}{2}\frac{\|\nabla f_k\|^4}{\nabla f_k^T B_k \nabla f_k} \\
&\le -\tfrac{1}{2}\frac{\|\nabla f_k\|^4}{\|B_k\|\|\nabla f_k\|^2} \\
&= -\tfrac{1}{2}\frac{\|\nabla f_k\|^2}{\|B_k\|} \\
&\le -\tfrac{1}{2}\|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right),
\end{aligned}
$$

so (4.35) holds here too.

In the remaining case, (4.36) does not hold, and therefore

$$
\nabla f_k^T B_k \nabla f_k < \frac{\|\nabla f_k\|^3}{\Delta_k}. \tag{4.37}
$$

From the definition of $p_k^{\mathrm{C}}$, we have $\tau = 1$, so using this fact together with (4.37), we obtain

$$
\begin{aligned}
m_k(p_k^{\mathrm{C}}) - m_k(0) &= -\frac{\Delta_k}{\|\nabla f_k\|}\|\nabla f_k\|^2 + \frac{1}{2}\frac{\Delta_k^2}{\|\nabla f_k\|^2}\nabla f_k^T B_k \nabla f_k \\
&\le -\Delta_k\|\nabla f_k\| + \frac{1}{2}\frac{\Delta_k^2}{\|\nabla f_k\|^2}\frac{\|\nabla f_k\|^3}{\Delta_k} \\
&= -\tfrac{1}{2}\Delta_k\|\nabla f_k\| \\
&\le -\tfrac{1}{2}\|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right),
\end{aligned}
$$

yielding the desired result (4.35) once more. $\square$

To satisfy (4.34), our approximate solution $p_k$ has only to achieve a reduction that is at least some fixed fraction $c_2$ of the reduction achieved by the Cauchy point. We state the observation formally as a theorem.

**Theorem 4.6.**

*Let $p_k$ be any vector such that $\|p_k\| \le \Delta_k$ and $m_k(0) - m_k(p_k) \ge c_2 \left(m_k(0) - m_k(p_k^{\mathrm{C}})\right)$. Then $p_k$ satisfies (4.34) with $c_1 = c_2/2$. In particular, if $p_k$ is the exact solution $p_k^*$ of (4.3), then it satisfies (4.34) with $c_1 = \frac{1}{2}$.*

PROOF. Since $\|p_k\| \le \Delta_k$, we have from (4.35) that

$$m_k(0) - m_k(p_k) \ge c_2 \left(m_k(0) - m_k(p_k^{\mathrm{C}})\right) \ge \tfrac{1}{2} c_2 \|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right),$$

giving the result. $\square$

Note that the dogleg and two-dimensional subspace minimization algorithms and Algorithm 4.3 all satisfy (4.34) with $c_1 = \frac{1}{2}$, because they all produce approximate solutions $p_k$ for which $m_k(p_k) \le m_k(p_k^{\mathrm{C}})$.

### CONVERGENCE TO STATIONARY POINTS

Global convergence results for trust-region methods come in two varieties, depending on whether we set the parameter $\eta$ in Algorithm 4.1 to zero or to some small positive value. When $\eta = 0$ (that is, the step is taken whenever it produces a lower value of $f$), we can show that the sequence of gradients $\{\nabla f_k\}$ has a limit point at zero. For the more stringent acceptance test with $\eta > 0$, which requires the actual decrease in $f$ to be at least some small fraction of the predicted decrease, we have the stronger result that $\nabla f_k \to 0$.

In this section we prove the global convergence results for both cases. We assume throughout that the approximate Hessians $B_k$ are uniformly bounded in norm, and that the level set

$$\{x \mid f(x) \le f(x_0)\} \tag{4.38}$$

is bounded. For generality, we also allow the length of the approximate solution $p_k$ of (4.3) to exceed the trust-region bound, provided that it stays within some fixed multiple of the bound; that is,

$$\|p_k\| \le \gamma \Delta_k, \qquad \text{for some constant } \gamma \ge 1. \tag{4.39}$$

The first result deals with the case of $\eta = 0$.

**Theorem 4.7.**

*Let $\eta = 0$ in Algorithm 4.1. Suppose that $\|B_k\| \le \beta$ for some constant $\beta$, that $f$ is continuously differentiable and bounded below on the level set (4.38), and that all approximate solutions of (4.3) satisfy the inequalities (4.34) and (4.39), for some positive constants $c_1$ and $\gamma$. We then have*

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0. \tag{4.40}$$

PROOF. We first perform some technical manipulation with the ratio $\rho_k$ from (4.4). We have

$$\begin{aligned}|\rho_k - 1| &= \left|\frac{(f(x_k) - f(x_k + p_k)) - (m_k(0) - m_k(p_k))}{m_k(0) - m_k(p_k)}\right| \\ &= \left|\frac{m_k(p_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}\right|.\end{aligned}$$

Since from Taylor's theorem (Theorem 2.1) we have that

$$f(x_k + p_k) = f(x_k) + \nabla f(x_k)^T p_k + \int_0^1 [\nabla f(x_k + tp_k) - \nabla f(x_k)]^T p_k \, dt,$$

it follows from the definition (4.1) of $m_k$ that

$$\begin{aligned}|m_k(p_k) - f(x_k + p_k)| &= |\tfrac{1}{2} p_k^T B_k p_k - \int_0^1 [\nabla f(x_k + tp_k) - \nabla f(x_k)]^T p_k \, dt| \\ &\le (\beta/2)\|p_k\|^2 + C_4(p_k)\|p_k\|,\end{aligned} \tag{4.41}$$

where we can make the scalar $C_4(p_k)$ arbitrarily small by restricting the size of $p_k$.

Suppose for contradiction that there is $\epsilon > 0$ and a positive index $K$ such that

$$\|\nabla f_k\| \ge \epsilon \qquad \text{for all } k \ge K. \tag{4.42}$$

From (4.34), we have for $k \ge K$ that

$$m_k(0) - m_k(p_k) \ge c_1 \|\nabla f_k\| \min\left(\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right) \ge c_1 \epsilon \min\left(\Delta_k, \frac{\epsilon}{\beta}\right). \tag{4.43}$$

Using (4.43), (4.41), and the bound (4.39), we have

$$|\rho_k - 1| \le \frac{\gamma \Delta_k (\beta\gamma\Delta_k/2 + C_4(p_k))}{2c_1\epsilon \min(\Delta_k, \epsilon/\beta)}. \tag{4.44}$$

We now derive a bound on the right-hand-side that holds for all sufficiently small values of $\Delta_k$, that is, for all $\Delta_k \le \bar{\Delta}$, where $\bar{\Delta}$ is to be determined. By choosing $\bar{\Delta}$ to be small enough and noting that $\|p_k\| \le \gamma\Delta_k \le \gamma\bar{\Delta}$, we can ensure that the term in parentheses in the numerator of (4.44) satisfies the bound

$$\beta\gamma\Delta_k/2 + C_4(p_k) \le \frac{c_1\epsilon}{2\gamma}. \tag{4.45}$$

By choosing $\bar{\Delta}$ even smaller, if necessary, to ensure that $\Delta_k \le \bar{\Delta} \le \epsilon/\beta$, we have from (4.44) that

$$|\rho_k - 1| \le \frac{\gamma\Delta_k c_1\epsilon/(2\gamma)}{2c_1\epsilon\Delta_k} = \frac{1}{4}.$$

Therefore, $\rho_k > \frac{3}{4}$, and so by the workings of Algorithm 4.1, we have $\Delta_{k+1} \ge \Delta_k$ whenever $\Delta_k$ falls below the threshold $\bar{\Delta}$. It follows that reduction of $\Delta_k$ $\left(\text{by a factor of } \frac{1}{4}\right)$ can occur in our algorithm only if

$$\Delta_k \ge \bar{\Delta},$$

and therefore we conclude that

$$\Delta_k \ge \min\left(\Delta_K, \bar{\Delta}/4\right) \qquad \text{for all } k \ge K. \tag{4.46}$$

Suppose now that there is an infinite subsequence $\mathcal{K}$ such that $\rho_k \ge \frac{1}{4}$ for $k \in \mathcal{K}$. If $k \in \mathcal{K}$ and $k \ge K$, we have from (4.43) that

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= f(x_k) - f(x_k + p_k) \\ &\ge \tfrac{1}{4}\left[m_k(0) - m_k(p_k)\right] \\ &\ge \tfrac{1}{4}c_1\epsilon\min(\Delta_k, \epsilon/\beta). \end{aligned}$$

Since $f$ is bounded below, it follows from this inequality that

$$\lim_{k\in\mathcal{K},\, k\to\infty} \Delta_k = 0,$$

contradicting (4.46). Hence no such infinite subsequence $\mathcal{K}$ can exist, and we must have $\rho_k < \frac{1}{4}$ for all $k$ sufficiently large. In this case, $\Delta_k$ will eventually be reduced by a factor of $\frac{1}{4}$ at every iteration, and we have $\lim_{k\to\infty}\Delta_k = 0$, which again contradicts (4.46). Hence, our original assertion (4.42) must be false, giving (4.40). □

Our second global convergence result, for the case $\eta > 0$, borrows much of the analysis from the proof above. Our approach here follows that of Schultz, Schnabel, and Byrd [226].

**Theorem 4.8.**

*Let $\eta \in \left(0, \frac{1}{4}\right)$ in Algorithm 4.1 . Suppose that $\|B_k\| \le \beta$ for some constant $\beta$, that $f$ is Lipschitz continuously differentiable and bounded below on the level set (4.38), and that all approximate solutions $p_k$ of (4.3) satisfy the inequalities (4.34) and (4.39) for some positive constants $c_1$ and $\gamma$. We then have*

$$\lim_{k\to\infty} \nabla f_k = 0. \tag{4.47}$$

PROOF. Consider any index $m$ such that $\nabla f_m \ne 0$. If we use $\beta_1$ to denote the Lipschitz constant for $\nabla f$ on the level set (4.38), we have

$$\|\nabla f(x) - \nabla f_m\| \le \beta_1 \|x - x_m\|,$$

for all $x$ in the level set. Hence, by defining the scalars

$$\epsilon = \tfrac{1}{2}\|\nabla f_m\|, \qquad R = \frac{\|\nabla f_m\|}{2\beta_1} = \frac{\epsilon}{\beta_1},$$

and the ball

$$\mathcal{B}(x_m, R) = \{x \mid \|x - x_m\| \le R\},$$

we have

$$x \in \mathcal{B}(x_m, R) \;\Rightarrow\; \|\nabla f(x)\| \ge \|\nabla f_m\| - \|\nabla f(x) - \nabla f_m\| \ge \tfrac{1}{2}\|\nabla f_m\| = \epsilon.$$

If the entire sequence $\{x_k\}_{k\ge m}$ stays inside the ball $\mathcal{B}(x_m, R)$, we would have $\|\nabla f_k\| \ge \epsilon > 0$ for all $k \ge m$. The reasoning in the proof of Theorem 4.7 can be used to show that this scenario does not occur. Therefore, the sequence $\{x_k\}_{k\ge m}$ eventually leaves $\mathcal{B}(x_m, R)$.

Let the index $l \ge m$ be such that $x_{l+1}$ is the first iterate after $x_m$ outside $\mathcal{B}(x_m, R)$. Since $\|\nabla f_k\| \ge \epsilon$ for $k = m, m+1, \ldots, l$, we can use (4.43) to write

$$\begin{aligned} f(x_m) - f(x_{l+1}) &= \sum_{k=m}^{l} f(x_k) - f(x_{k+1}) \\ &\ge \sum\nolimits_{k=m, x_k \ne x_{k+1}}^{l} \eta[m_k(0) - m_k(p_k)] \\ &\ge \sum\nolimits_{k=m, x_k \ne x_{k+1}}^{l} \eta c_1 \epsilon \min\left(\Delta_k, \frac{\epsilon}{\beta}\right), \end{aligned}$$

where we have limited the sum to the iterations $k$ for which $x_k \ne x_{k+1}$, that is, those iterations on which a step was actually taken. If $\Delta_k \le \epsilon/\beta$ for all $k = m, m+1, \ldots, l$, we have

$$f(x_m) - f(x_{l+1}) \ge \eta c_1 \epsilon \sum_{k=m, x_k \ne x_{k+1}}^{l} \Delta_k \ge \eta c_1 \epsilon R = \eta c_1 \epsilon^2 \frac{1}{\beta_1}. \tag{4.48}$$

Otherwise, we have $\Delta_k > \epsilon/\beta$ for some $k = m, m+1, \dots, l$, and so

$$f(x_m) - f(x_{l+1}) \geq \eta c_1 \epsilon \frac{\epsilon}{\beta}. \tag{4.49}$$

Since the sequence $\{f(x_k)\}_{k=0}^{\infty}$ is decreasing and bounded below, we have that

$$f(x_k) \downarrow f^* \tag{4.50}$$

for some $f^* > -\infty$. Therefore, using (4.48) and (4.49), we can write

$$\begin{aligned} f(x_m) - f^* &\geq f(x_m) - f(x_{l+1}) \\ &\geq \eta c_1 \epsilon^2 \min\left(\frac{1}{\beta}, \frac{1}{\beta_1}\right) = \frac{1}{4}\eta c_1 \min\left(\frac{1}{\beta}, \frac{1}{\beta_1}\right) \|\nabla f_m\|^2. \end{aligned}$$

By rearranging this expression, we obtain

$$\|\nabla f_m\|^2 \leq \left(\frac{1}{4}\eta c_1 \min\left(\frac{1}{\beta}, \frac{1}{\beta_1}\right)\right)^{-1} (f(x_m) - f^*),$$

so from (4.50) we conclude that $\nabla f_m \to 0$, giving the result. □

## CONVERGENCE OF ALGORITHMS BASED ON NEARLY EXACT SOLUTIONS

As we noted in the discussion of Algorithm 4.4, the loop to determine the optimal values of $\lambda$ and $p$ for the subproblem (4.9) does not iterate until high accuracy is achieved. Instead, it is terminated after two or three iterations with a fairly loose approximation to the true solution. The inexactness in this approximate solution is, however, measured in a different way from the dogleg and subspace minimization algorithms and Algorithm 4.3, and this difference affects the nature of the global convergence results that can be proved.

Moré and Sorensen [170] describe a safeguarded version of the root-finding Newton method that adds features for handling the hard case. Its termination criteria ensure that their approximate solution $p$ satisfies the conditions

$$m(0) - m(p) \geq c_1(m(0) - m(p^*)), \tag{4.51a}$$

$$\|p\| \leq \gamma\Delta \tag{4.51b}$$

(where $p^*$ is the exact solution of (4.3)), for some constants $c_1 \in (0, 1]$ and $\gamma > 0$. The condition (4.51a) ensures that the approximate solution achieves a significant fraction of the maximum decrease possible in the model function $m$. Of course, it is not necessary to know $p^*$ to enforce this condition; it follows from practical termination criteria. One major difference between (4.51) and the earlier criterion (4.34) is that (4.51) makes better use of the second-order part of $m(\cdot)$, that is, the $p^T Bp$ term. This difference is illustrated by the case in which $g = 0$ while $B$ has negative eigenvalues, indicating that the current iterate $x_k$ is a saddle point. Here, the right-hand-side of (4.34) is zero (indeed, the algorithms we described earlier would terminate at such a point). The right-hand-side of (4.51) is positive, indicating that decrease in the model function is still possible, so it forces the algorithm to move away from $x_k$.

The close attention that near-exact algorithms pay to the second-order term is warranted only if this term closely reflects the actual behavior of the function $f$—in fact, the trust-region Newton method, for which $B = \nabla^2 f(x)$, is the only case that has been treated in the literature. For purposes of global convergence analysis, the use of the exact Hessian allows us to say more about the limit points of the algorithm than merely that they are stationary points. In fact, second-order necessary conditions (Theorem 2.3) are satisfied at the limit points. The following result establishes this claim.

**Theorem 4.9.**

*Suppose Algorithm 4.1 is applied with $B_k = \nabla^2 f(x_k)$, constant $\eta$ in the* open *interval $\left(0, \frac{1}{4}\right)$, and the approximate solution $p_k$ at each iteration satisfying (4.51) for some fixed $\gamma > 0$. Then* $\lim_{k \to \infty} \|\nabla f_k\| = 0$.

*If, in addition, the level set $\{x \mid f(x) \le f(x_0)\}$ is compact, then either the algorithm terminates at a point $x_k$ at which the second-order necessary conditions (Theorem 2.3) for a local minimum hold, or else $\{x_k\}$ has a limit point $x^*$ in the level set at which the necessary conditions hold.*

We omit the proof, which can be found in Moré and Sorensen [170, Section 4].

## 4.4 OTHER ENHANCEMENTS

### SCALING

As we noted in Chapter 2, optimization problems are often posed with poor scaling—the objective function $f$ is highly sensitive to small changes in certain components of the vector $x$ and relatively insensitive to changes in other components. Topologically, a symptom of poor scaling is that the minimizer $x^*$ lies in a narrow valley, so that the contours of the objective $f(\cdot)$ near $x^*$ tend towards highly eccentric ellipses. Algorithms can perform poorly unless they compensate for poor scaling; see Figure 2.7 for an illustration of the poor performance of the steepest descent approach.

Recalling our definition of a trust region—a region around the current iterate within which the model $m_k(\cdot)$ is an adequate representation of the true objective $f(\cdot)$—it is easy to see that a *spherical* trust region is not appropriate to the case of poorly scaled functions. We can trust our model $m_k$ to be reasonably accurate only for short distances along the highly sensitive directions, while it is reliable for longer distances along the less sensitive directions. Since the shape of our trust region should be such that our confidence in the model is more or less the same at all points on the boundary of the region, we are led naturally to consider *elliptical* trust regions in which the axes are short in the sensitive directions and longer in the less sensitive directions. Elliptical trust regions can be defined by

$$\|Dp\| \le \Delta, \tag{4.52}$$

where $D$ is a diagonal matrix with positive diagonal elements, yielding the following scaled trust-region subproblem:

$$\min_{p\in \mathbb{R}^n} m_k(p) \stackrel{\text{def}}{=} f_k + \nabla f_k^T p + \tfrac{1}{2} p^T B_k p \qquad \text{s.t. } \|Dp\| \le \Delta_k. \tag{4.53}$$

When $f(x)$ is highly sensitive to the value of the $i$th component $x_i$, we set the corresponding diagonal element $d_{ii}$ of $D$ to be large. The value of $d_{ii}$ will be closer to zero for less-sensitive components $x_i$.

Information to construct the scaling matrix $D$ can be derived reliably from the second derivatives $\partial^2 f/\partial x_i^2$. We can allow $D$ to change from iteration to iteration, as long as all diagonal elements $d_{ii}$ stay inside some predetermined range $[d_{\text{lo}}, d_{\text{hi}}]$, where $0 < d_{\text{lo}} \le d_{\text{hi}} < \infty$. Of course, we do not need $D$ to be a *precise* reflection of the scaling of the problem, so it is not necessary to devise elaborate heuristics or to spend a lot of computation to get it just right.

All algorithms discussed in this chapter can be modified for the case of elliptical trust regions, and the convergence theory continues to hold, with numerous superficial modifications. The Cauchy point calculation procedure (Algorithm 4.2), for instance, changes only in the specifications of the trust region in (4.5) and (4.6). We obtain the following generalized version.

**Algorithm 4.5** (Generalized Cauchy Point Calculation).
Find the vector $p_k^{\text{s}}$ that solves

$$p_k^{\text{s}} = \arg\min_{p\in \mathbb{R}^n} f_k + \nabla f_k^T p \qquad \text{s.t. } \|Dp\| \le \Delta_k; \tag{4.54}$$

Calculate the scalar $\tau_k > 0$ that minimizes $m_k(\tau p_k^{\text{s}})$ subject to satisfying the trust-region bound, that is,

$$\tau_k = \arg\min_{\tau>0} m_k(\tau p_k^{\text{s}}) \qquad \text{s.t. } \|\tau D p_k^{\text{s}}\| \le \Delta_k; \tag{4.55}$$

$$p_k^{\text{C}} = \tau_k p_k^{\text{s}}.$$

For this scaled version, we find that

$$p_k^{\text{s}} = -\frac{\Delta_k}{\|D^{-1}\nabla f_k\|} D^{-2}\nabla f_k, \tag{4.56}$$

and that the step length $\tau_k$ is obtained from the following modification of (4.8):

$$\tau_k = \begin{cases} 1 & \text{if } \nabla f_k^T D^{-2} B_k D^{-2} \nabla f_k \le 0 \\ \min\left( \dfrac{\|D^{-1}\nabla f_k\|^3}{\Delta_k \nabla f_k^T D^{-2} B_k D^{-2} \nabla f_k}, 1 \right) & \text{otherwise.} \end{cases} \tag{4.57}$$

(The details are left as an exercise.)

A simpler alternative for adjusting the definition of the Cauchy point and the various algorithms of this chapter to allow for the elliptical trust region is simply to rescale the variables $p$ in the subproblem (4.53) so that the trust region is spherical in the scaled variables. By defining

$$\tilde{p} \stackrel{\text{def}}{=} Dp,$$

and by substituting into (4.53), we obtain

$$\min_{\tilde{p} \in \mathbb{R}^n} \tilde{m}_k(\tilde{p}) \stackrel{\text{def}}{=} f_k + \nabla f_k^T D^{-1} \tilde{p} + \tfrac{1}{2} \tilde{p}^T D^{-1} B_k D^{-1} \tilde{p} \qquad \text{s.t. } \|\tilde{p}\| \le \Delta_k.$$

The theory and algorithms can now be derived in the usual way by substituting $\tilde{p}$ for $p$, $D^{-1}\nabla f_k$ for $\nabla f_k$, $D^{-1} B_k D^{-1}$ for $B_k$, and so on.

### NON-EUCLIDEAN TRUST REGIONS

Trust regions may also be defined in terms of norms other than the Euclidean norm. For instance, we may have

$$\|p\|_1 \le \Delta_k \qquad \text{or} \qquad \|p\|_\infty \le \Delta_k,$$

or their scaled counterparts

$$\|Dp\|_1 \le \Delta_k \qquad \text{or} \qquad \|Dp\|_\infty \le \Delta_k,$$

where $D$ is a positive diagonal matrix as before. Norms such as these offer no obvious advantages for unconstrained optimization, but they may be useful for constrained problems. For instance, for the bound-constrained problem

$$\min_{x \in \mathbb{R}^n} f(x), \qquad \text{subject to } x \ge 0,$$

the trust-region subproblem may take the form

$$\min_{p \in \mathbb{R}^n} m_k(p) = f_k + \nabla f_k^T p + \tfrac{1}{2} p^T B_k p \qquad \text{s.t. } x_k + p \ge 0, \|p\| \le \Delta_k. \tag{4.58}$$

When the trust region is defined by a Euclidean norm, the feasible region for (4.58) consists of the intersection of a sphere and the nonnegative orthant—an awkward object, geometrically speaking. When the $\infty$-norm is used, however, the feasible region is simply the rectangular box defined by

$$x_k + p \geq 0, \qquad p \geq -\Delta_k e, \qquad p \leq \Delta_k e,$$

where $e = (1, 1, \ldots, 1)^T$, so the solution of the subproblem is easily calculated by standard techniques for quadratic programming.

### NOTES AND REFERENCES

The influential paper of Powell [199] proves a result like Theorem 4.7 for the case of $\eta = 0$, where the algorithm takes a step whenever it decreases the function value. Powell uses a weaker assumption than ours on the matrices $\|B\|$, but his analysis is more complicated. Moré [167] summarizes developments in algorithms and software before 1982, paying particular attention to the importance of using a scaled trust-region norm. Much of the material in this chapter on methods that use nearly exact solutions to the subproblem (4.3) is drawn from the paper of Moré and Sorensen [170].

Byrd, Schnabel, and Schultz [226], [39] provide a general theory for inexact trust-region methods; they introduce the idea of two-dimensional subspace minimization and also focus on proper handling of the case of indefinite $B$ to ensure stronger local convergence results than Theorems 4.7 and 4.8. Dennis and Schnabel [70] survey trust-region methods as part of their overview of unconstrained optimization, providing pointers to many important developments in the literature.

## EXERCISES

**4.1** Let $f(x) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2$. At $x = (0, -1)$ draw the contour lines of the quadratic model (4.1) assuming that $B$ is the Hessian of $f$. Draw the family of solutions of (4.3) as the trust region radius varies from $\Delta = 0$ to $\Delta = 2$. Repeat this at $x = (0, 0.5)$.

**4.2** Write a program that implements the dogleg method. Choose $B_k$ to be the exact Hessian. Apply it to solve Rosenbrock's function (2.23). Experiment with the update rule for the trust region by changing the constants in Algorithm 4.1, or by designing your own rules.

**4.3** Program the trust region method based on Algorithm 4.3. Choose $B_k$ to be the exact Hessian, and use it to minimize the function

$$\min \ f(x) = \sum_{i=1}^{n} \left[ (1 - x_{2i-1})^2 + 10(x_{2i} - x_{2i-1}^2)^2 \right]$$

with $n = 10$. Experiment with the starting point and the stopping test for the CG iteration. Repeat the computation with $n = 50$.

Your program should indicate, at every iteration, whether Algorithm 4.3 encountered negative curvature, reached the trust region boundary, or met the stopping test.

✐ **4.4** Theorem 4.7 shows that the sequence $\{\|g\|\}$ has an accumulation point at zero. Show that if the iterates $x$ stay in a bounded set $\mathcal{B}$, then there is a limit point $x_\infty$ of the sequence $\{x_k\}$ such that $\nabla f(x_\infty) = 0$.

✐ **4.5** Show that $\tau_k$ defined by (4.8) does indeed identify the minimizer of $m_k$ along the direction $-\nabla f_k$.

✐ **4.6** The Cauchy–Schwarz inequality states that for any vectors $u$ and $v$, we have

$$|u^T v|^2 \le (u^T u)(v^T v),$$

with equality only when $u$ and $v$ are parallel. When $B$ is positive definite, use this inequality to show that

$$\gamma \stackrel{\text{def}}{=} \frac{\|g\|^4}{(g^T Bg)(g^T B^{-1} g)} \le 1,$$

with equality only if $g$ and $Bg$ (and $B^{-1}g$) are parallel.

✐ **4.7** When $B$ is positive definite, the *double-dogleg method* constructs a path with three line segments from the origin to the full step. The four points that define the path are

- the origin;
- the unconstrained Cauchy step $p^{\mathrm{C}} = -(g^T g)/(g^T Bg)g$;
- a fraction of the full step $\bar{\gamma} p^{\mathrm{B}} = -\bar{\gamma} B^{-1} g$, for some $\bar{\gamma} \in (\gamma, 1]$, where $\gamma$ is defined in the previous question; and
- the full step $p^{\mathrm{B}} = -B^{-1} g$.

Show that $\|p\|$ increases monotonically along this path.

(Note: The double-dogleg method, as discussed in Dennis and Schnabel [69, Section 6.4.2], was for some time thought to be superior to the standard dogleg method, but later testing has not shown much difference in performance.)

✐ **4.8** Show that (4.26) and (4.27) are equivalent. Hints: Note that

$$\frac{d}{d\lambda}\left(\frac{1}{\|p(\lambda)\|}\right) = \frac{d}{d\lambda}\left(\|p(\lambda)\|^2\right)^{-1/2} = -\frac{1}{2}\left(\|p(\lambda)\|^2\right)^{-3/2}\frac{d}{d\lambda}\|p(\lambda)\|^2,$$

$$\frac{d}{d\lambda}\|p(\lambda)\|^2 = -2\sum_{j=1}^{n}\frac{(q_j^T g)^2}{(\lambda_j + \lambda)^3}$$

(from (4.22)), and

$$\|q\|^2 = \|R^{-T}p\|^2 = p^T(B+\lambda I)^{-1}p = \sum_{j=1}^{n} \frac{(q_j^T g)^2}{(\lambda_j+\lambda)^3}.$$

**4.9** Derive the solution of the two-dimensional subspace minimization problem in the case where $B$ is positive definite.

**4.10** Show that if $B$ is any symmetric matrix, then there exists $\lambda \geq 0$ such that $B+\lambda I$ is positive definite.

**4.11** Verify that the definitions (4.56) for $p_k^{\mathrm{s}}$ and (4.57) for $\tau_k$ are valid for the Cauchy point in the case of an elliptical trust region. (Hint: Using the theory of Chapter 12, we can show that the solution of (4.54) satisfies $\nabla f_k + \alpha D^2 p_k^{\mathrm{s}} = 0$ for some scalar $\alpha \geq 0$.)

**4.12** The following example shows that the reduction in the model function $m$ achieved by the two-dimensional minimization strategy can be much smaller than that achieved by the exact solution of (4.9).

In (4.9), set

$$g = \left(-\frac{1}{\epsilon}, -1, -\epsilon^2\right)^T,$$

where $\epsilon$ is a small positive number. Set

$$B = \mathrm{diag}\left(\frac{1}{\epsilon^3}, 1, \epsilon^3\right), \quad \Delta = 0.5.$$

Show that the solution of (4.9) has components $\left(O(\epsilon), \frac{1}{2}+O(\epsilon), O(\epsilon)\right)^T$ and that the reduction in the model $m$ is $\frac{3}{8}+O(\epsilon)$. For the two-dimensional minimization strategy, show that the solution is a multiple of $B^{-1}g$ and that the reduction in $m$ is $O(\epsilon)$.

# Chapter 5