The Solution of Large-Scale Optimization Problems
(using MATLAB)

Thomas F. Coleman[1]
Computer Science
Cornell University
Ithaca , New York

March 29, 1999

[1]coleman@tc.cornell.edu http://www.tc.cornell.edu/~ coleman

Contents

Chapter 1

Unconstrained Minimization and Nonlinear Systems

With sparse linear algebra tools now in hand, we are ready to study numerical methods for large-scale optimization,

$$\min_x \{f(x)\}, \tag{1.1}$$

and nonlinear systems,

$$\text{solve} \quad F(x) = 0, \tag{1.2}$$

where $f : \Re^n \to \Re$ and $F : \Re^n \to \Re^n$. We are concerned with smooth differentiable mappings; specific smoothness assumptions will be made subsequently, as required.

1.1 The End Game

We begin by studying local Newton processes for problem (1.5). That is, assuming we can start (arbitrarily) close to x_*, where $F(x_*) = 0$, how can we generate iterates $\{x_k\}$ that rapidly converge to the zero x_*? This may appear to be an impractical question since in general we cannot start (arbitrarily) close to x_*, given that x_* is unknown. However, the question is crucial: ultimately we seek global methods that automatically reduce to effective (fast) local procedures once a neighbourhood of x_* is reached.

A Newton *process* for $F(x) = 0$ is described in Figure 1.1.

A linear approximation to F, at x_k, is written $F(x_k + s) = F(x_k) + B_k s$ where matrix B_k is viewed as approximating the Jacobian of F at x_k. Therefore, we can describe the Newton process (1.1) with more precision: see Figure 1.2.

The algorithm in Figure 1.2 is a *local* procedure: if the starting point x_0 is close enough to the zero x_* and B_k doesn't stray too far from $J(x_k)$, the true Jacobian of F, then the iterates $\{x_k\}$ converge to x_*. Theorem 1.1.1 formalizes this result.

Theorem 1.1.1 *Assume F is a continuously differentiable square mapping on an open domain $\mathcal{D} \subseteq \Re^n$ and the Jacobian of F, $J(x)$, is nonsingular at $x_* \in \mathcal{D}$ where $F(x_*) = 0$. There exists positive constants δ_1, δ_2 such that if*

$$\{\|x_0 - x_*\| < \delta_1, \quad \|B_k - J(x_*)\| < \delta_2, \quad \forall\ k\}$$

then $\{x_k\}$ converges (linearly) to x_*.

Guess starting point x_0, B_0; $k \leftarrow 0$.
Repeat

(i) Model F *with a linear approximation* L_k *at the current point* x_k.

(ii) Find s_k, *a zero of the model* L_k.

(iii) Update: $x_{k+1} \leftarrow x_k + s_k$, $k \leftarrow k+1$

Figure 1.1: Newton Process I

Guess starting point x_0, B_0; $k \leftarrow 0$.
Repeat

(i) Evaluate $F(x_k), B_k$; *(Define* $L_k(s) = F(x_k) + B_k s$.*)*

(ii) Solve $B_k s_k = -F(x_k)$.

(iii) Update: $x_{k+1} \leftarrow x_k + s_k$, $k \leftarrow k+1$

Figure 1.2: Newton Process II

To prove Theorem 1.1.1 (see Exercise ??), it is necessary to show that F is a *contraction* in a neighbourhood of x_*. This can be done by induction. First prove that x_1 is (sufficiently) closer to x_* by using Taylor's Theorem to show[1]

$$x_1 - x_* = B_0^{-1}(B_0 - J_*)(x_0 - x_*) + o(\|x_0 - x_*\|). \tag{1.3}$$

It follows from (1.3) that for positive δ_1, δ_2 sufficiently small,

$$\|x_1 - x_*\| \leq \kappa \cdot \|x_0 - x_*\| \tag{1.4}$$

for a positive constant $\kappa < 1$. An induction argument can now be used to establish the result.

Theorem 1.1.1 establishes that under mild assumptions a Newton process generates a sequence of points that converges to a zero of F, provided the starting point x_0 is close enough to x_* and the approximating matrices B_k are not too wild. However, merely establising *linear* convergence, without strong bounds on the linear factor κ, is usually not enough to provide support for a (locally) good optimization procedure. What further restrictions on the Newton process in Figure 1.2 can guarantee an improved convergence rate? Theorem 1.1.2 indicates that by demanding greater accuracy in the Jacobian approximations, a superlinear convergence rate can be achieved.

Basic convergence rate definitions are given in Figure 1.3.

[1]Notation: Suppose $\phi = \phi(x), \psi = \psi(x)$ are scalar functions of x and $x \to x_*$. We write $\phi = O(\psi)$ if the ratio $\frac{\phi}{\psi}$ remains bounded as $x \to x_*$. We write $\phi = o(\psi)$ if the ratio $\frac{\phi}{\psi} \to 0$ as $x \to x_*$.

Assume the sequence $\{x_k\}$ converges to x_*.

(i) The sequence $\{x_k\}$ converges to x_* linearly *if there exists an index $\bar{k}$ and a constant $c \in [0,1)$ such that*

$$\|x_{k+1} - x_*\| \leq c \cdot \|x_k - x_*\|, \quad \forall k \geq \bar{k}.$$

(ii) The sequence $\{x_k\}$ converges to x_* superlinearly *if there exists an index $\bar{k}$ and a sequence of positive scalars $\{c_k\} \to 0$, such that*

$$\|x_{k+1} - x_*\| \leq c_k \cdot \|x_k - x_*\|, \quad \forall k \geq \bar{k}.$$

(iii) The sequence $\{x_k\}$ converges to x_* quadratically *if there exists an index $\bar{k}$ and a positive scalar c such that*

$$\|x_{k+1} - x_*\| \leq c \cdot \|x_k - x_*\|^2, \; \forall k \geq \bar{k}.$$

Figure 1.3: Convergence rates

Theorem 1.1.2 *Let the assumptions of Theorem 1.1.1 hold. In addition, assume that $\{\|B_k^{-1}\|\}$ is bounded and $\{x_k\} \to x_*$. If $\lim\{\frac{\|(B_k - J_*)(x_k - x_*)\|}{\|x_k - x_*\|}\} = 0$ then $\{x_k\}$ converges to x_* at a superlinear rate, i.e.,*

$$\{\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|}\} \;\to\; 0. \tag{1.5}$$

A complete proof is requested in (§1.3, Question 3). The essence of the proof is to establish, by Taylor's Theorem,

$$x_{k+1} - x_* \;=\; B_k^{-1}(B_k - J_*)(x_k - x_*) \;+\; o(\|x_k - x_*\|). \tag{1.6}$$

By taking norms and dividing by $\|x_k - x_*\|$, it follows that

$$\frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|} \;\leq\; \|B_k^{-1}\| \cdot \frac{\|(B_k - J_*)(x_k - x_*)\|}{\|x_k - x_*\|} \;+\; \frac{o(\|x_k - x_*\|)}{\|x_k - x_*\|}. \tag{1.7}$$

The limit of the right-hand-side of (1.7) is zero and so the result follows.

Theorem 1.1.2 can be used to establish superlinear convergence of quasi-Newton methods as well as schemes that ensure $\{B_k\} \to J_*$. For example it follows that finite differencing of F to obtain Jacobian approximations will yield superlinear convergence, in principle, if the differencing parameter goes to zero. (In practise it is *not* recommended to drive the differencing parameter to zero because of the potential of disastrous truncation error.)

To obtain a quadratic convergence rate, the approximation matrices must satisfy more stringent accuracy requirements.

Definition. A continuous function $h(x)$ is *Lipschitz continuous* at point x_* if $\|h(x) - h(x_*)\| = O(\|x - x_*\|)$ for all x in a neigbourhood of x_*.

Theorem 1.1.3 *Assume the assumptions of Theorem 1.1.1 hold. Further, assume that $J(x)$ is Lipschitz continuous at x_* and $\|B_k - J_k\| \;=\; O(\|x_k - x_*\|)$. There exists a positive scalar δ_1 such that if $\|x_0 - x_*\| < \delta_1$ then $\{x_k\}$ converges quadratically to x_*.*

The proof of Theorem 1.1.3 is left as an exercise (§1.3 , Question 5).

1.1.1 Optimality conditions for unconstrained minimization.

The previous discussion on Newton-like processes for the zero finding problem (1.5) can be applied to the minimization problem (1.5) in a straightforward way. In particular, identify $F(x)$ with the gradient of f, ∇f. In this case the matrix B_k will approximate the Hessian matrix of f, $\nabla^2 f(x)$. However, it is important to note additional properties that hold at (or imply) a local minimizer.

A point x_* is a *local minimizer* of a continuous function f is there exists a positive scalar $\delta > 0$ such that if $\|x - x_*\| < \delta$ then $f(x) \leq f(x_*)$. A point x_* is a *strong local minimizer* of a continuous function f is there exists a positive scalar $\delta > 0$ such that if $\|x - x_*\| < \delta$ and $x \neq x_*$, then $f(x) < f(x_*)$. Optimality conditions are as follows:

> *First-order necessary conditions:* Assume f is continuously differentiable on an open set $\mathcal{D} \subseteq \Re^n$. If $x_* \in \mathcal{D}$ is a local minimizer of f then $\nabla f(x_*) = 0$.
>
> *Second-order necessary conditions:* Assume f is twice continuously differentiable on an open set $\mathcal{D} \subseteq \Re^n$. If $x_* \in \mathcal{D}$ is a local minimizer of f then $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*)$ is positive semi-definite.
>
> *Second-order sufficiency conditions:* Assume f is twice continuously differentiable on an open set $\mathcal{D} \subseteq \Re^n$. If $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*)$ is positive definite then x_* is a strong local minimizer.

1.2 Descent Direction Methods

Most algorithms for continuous unconstrained minimization are descent direction algorithms. Vector d is a *descent direction* for a function f at a point $\bar{x}$ if there is a positive $\bar{\alpha}$ such that

$$f(\bar{x} + \alpha d) \;<\; f(\bar{x}), \quad \forall\; 0 < \alpha < \bar{\alpha}.$$

It is easy to see that if f is continuously differentiable and if $\nabla f(\bar{x})^T d \;<\; 0$, then d is a descent direction for f at $\bar{x}$. A generic descent direction iteration is given in Figure 1.4.

Different descent algorithms differ primarily in how the descent direction d_k is chosen. There is also some variation in how to implement the line search; typically, a loose tolerance is used. For example, it is popular to demand that α_k satisfy the conditions in Figure 1.5.

Condition (LS1) in Figure 1.5 demands that the average change in f over the line segment $[x_k, x_k + \alpha_k d_k]$ yield a reduction comensurate with a constant factor times the reduction predicted by the directional derivative $\nabla f(x_k)^T d_k$. It has the effect of restricting the the step size from above. Condition (LS1) requires that the directional derivative at the new point (x_{k+1}) be less negative, by a constant factor, than the directional derivative at the current point x_k. It has the effect of restricting the step size from below.

If f is continuously differentiable, $0 < \gamma < \beta < 1$, and $\nabla f(x_k)^T d_k < 0$ then there exists a range of values of α satisfying (LS1) and (LS2). A proof is given, for example, in ([25] Theorem 6.3.2).

We are now in position to state a fundamental convergence result.

(i) Construct d_k *such that* $\nabla f_k^T d_k < 0$.

(ii) Minimize along d_k *(i.e., line search):*

$$\alpha_k \approx \alpha_* \;=\; argmin\{f(x_k + \alpha d_k) : \alpha > 0\}$$

(iii) Update: $x_{k+1} \leftarrow x_k \;+\; \alpha_k d_k$

Figure 1.4: A Generic Descent Direction Iteration

(LS1) $\quad \frac{f(x_k+\alpha_k d_k)-f(x_k)}{\alpha_k} \;\leq\; \gamma \cdot \nabla f(x_k)^T d_k, 0 < \gamma < 1.$

(LS2) $\quad \nabla f(x_k + \alpha_k d_k)^T d_k \geq \beta \cdot \nabla f(x_k)^T d_k$, *for* $0 < \gamma < \beta < 1.$

Figure 1.5: Sufficient Line Search Conditions

Theorem 1.2.1 *Let* f *be continuously differentiable and bounded below. Suppose that* ∇f *is Lipschitz continuous;* $x_{k+1} = x_k + \alpha_k d_k$ *where* $\nabla f_k^T d_k < 0$. *Assume that* α_k *satisfies (LS1). Further, assume either (LS2) holds or* $\alpha_k \geq \bar{\alpha}$ *for some* $\bar{\alpha} > 0$. *Then, either* $\nabla f_k = 0$ *for some* k *or*

$$\nabla f_k^T (\frac{d_k}{\|d_k\|}) \to 0.$$

The proof of (Theorem 6.3.3, [25]) can easily be adapted to establish Theorem 1.2.1.

Remarks:

(i) Theorem 1.2.1 does not imply the convergence of $\{x_k\}$, nor does it imply the convergence of ∇f_k; however, Theorem 1.2.1 establishes a useful framework for descent direction methods.

(ii) If the angle between $-\nabla f_k$ and d_k is bounded strictly less than 90 degrees then Theorem 1.2.1 implies $\nabla f_k \to 0$. Therefore, if $\{B_k\}$ is a sequence of SPD matrices with $B_k d_k = -\nabla f_k$, and if the condition number of B_k is bounded as $k \to \infty$ then $\nabla f_k \to 0$.

(iii) The purpose of $\bar{\alpha}$ in Theorem 1.2.1 is to allow for the acceptance of unit step sizes, $\alpha = 1$, even when condition (LS2) is violated. This works because the role of (LS2) is to prevent asymptotically small step lengths; clearly this objective is also achieved if $\alpha > \bar{\alpha} > 0$.

Different Descent Directions. There are a number of popular ways to compute the downhill direction d_k in Algorithm 1.4. We list some of the main possibilities here with a few remarks concerning their suitability.

Steepest Descent: In the steepest descent method the search direction is the negative gradient, $d_k = -\nabla f(x_k)$. In general this is a poor method and is not recommended – see Question 3 in §1.3. Steepest descent can exhibit zig-zagging behaviour on (mildy) ill-conditioned problems, thus requiring many iterations to achieve modest accuracy.

Quasi-Newton Methods: Modern nonlinear optimization began with the development of quasi-Newton methods [21, 34, 10]. Quasi-Newton methods have been enormously successful, especially for small and moderate dimensions. Specifically, the positive definite secant method, or BFGS, has been a valuable tool for solving unconstrained continuous optimization problems.[2] Quasi-Newton methods such as the BFGS are less attractive for large-scale problems: an effective sparsity-preserving quasi-Newton approach has not been found, despite considerable research effort. Coleman [13] summarizes some of the sparse approaches that have been tried. One solution to the problem of preserving both sparsity and positive-definiteness is to use a limited-memory secant method, e.g., [65, 54]. In a limited memory method, positive definiteness is preserved but inverse quasi-Newton updates are stored as a finite-sequence of update vectors, avoiding the need to store an explicit Hessian approximation. Thus, preservation of sparsity is not an issue. However, information must be discarded in order to make room for new updates and so performance may be degraded.

For more background on quasi-Newton methods, see [23, 25, 33].

Nonlinear Conjugate Gradients: The method of conjugate gradents for SPD systems of linear equations, discussed in §**??**, can be adapted to solve minimization problems. For example, the Fletcher-Reeves Algorithm [35], is obtained from Algorithm CG (Figure **??**) by replacing the residual vector r with the gradient ∇f, and the computation of α with a line search on the nonlinear function f. The Fletcher-Reeves Algorithm is given in Figure 1.6.

While $\|\nabla f(x)\| > tol$

1. $d \leftarrow -\nabla f(x) + \beta d_-$
2. $\alpha_* \approx argmin\{f(x + \alpha d,\ \alpha > 0\},\ x \leftarrow x + \alpha_* d$
3. $\beta \leftarrow \frac{\nabla f(x)^T \nabla f(x)}{\nabla f(x_-)^T \nabla f(x_-)}$

return(x)

Figure 1.6: Fletcher-Reeves Algorithm

Another nonlinear conjugate gradient variation is the Polak and Ribière [67] procedure, given in Figure 1.7. Similar to Fletcher-Reeves, this method reduces to Algorithm CG when applied to a positive definite quadratic function.

Nonlinear CG methods are attractive due to their simplicity and minimal memory requirements. However, there are several difficulties. First, a relatively exact line search procedure is required

[2] The BFGS update was discovered, independently, by Broyden, Fletcher, Goldfarb, and Shanno [42, 32, 11, 74].

While $\|\nabla f(x)\| > tol$

1. $d \leftarrow -\nabla f(x) + \beta d_{-}$
2. $\alpha \leftarrow argmin\{f(x + \alpha d,\ \alpha > 0\}$, $x \leftarrow x + \alpha d$
3. $\beta \leftarrow \frac{y^T \nabla f(x)}{\nabla f(x_-)^T \nabla f(x_-)}$, *where* $y = \nabla f(x) - \nabla f(x_-)$.

return(x)

Figure 1.7: Polak and G. Ribière Algorithm

to ensure the generation of descent directions. This can be costly in terms of function evaluations and, in practise, necessitates the use of restart strategies. Second, convergence can be slow, thus requiring the use of preconditioners. In this case effective preconditioning can be a subtle thing since the underlying Hessian matrices are not computed. For more information and background on nonlinear conjugate gradient methods see [33, 1].

1.3 Exercises

1. Let $f : R^n \to R$ be twice continuously differentiable on an open convex set D containing a local minimum x_*. Prove, from first principles, that $\nabla f(x_*) = 0$ and the Hessian matrix is positive semi-definite at the solution.

2. Prove Theorem 1.1.1.

3. Prove Theorem 1.1.2.

4. Consider the following theorem (characterization of superlinear convergence). Assume $F : R^n \to R^n$ satisfies

(i) F is continuously differentiable on an open convex set D,
(ii) There exists $x_* \in D$ such that $F(x_*) = 0$ and $F'(x_*)$ is nonsingular,
(iii) $\|F'(x) - F'(x_*)\| \leq \eta(\|x - x_*\|)$ for all $x \in D$ and some $\eta \geq 0$.

Let $\{B_k\}$ be a sequence of nonsingular matrices. Suppose that for some x_0 in D the sequence

$$x_{k+1} = x_k - B_k^{-1} F(x_k), \quad k = 0, 1, ... \tag{1.8}$$

remains in D, $x_k \neq x_*$ for $k \geq 0$, and converges to x_*. Then $\{x_k\}$ converges superlinearly to x_* if and only if

$$\lim_{k \to +\infty} \frac{\|[B_k - J_*](x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0. \quad \square \tag{1.9}$$

It is interesting to note that, if we define $s_k = x_{k+1} - x_k$ and $s_k^N = -J_k^{-1} F(x_k)$, then it is possible to rephrase (1.9) without explicity referring to an iteration matrix B_k. Consider the following restatement of the superlinear characterization theorem: Let assumptions (i-iii) hold. Let $\{x_k\}$ be a sequence of points that remains in D, $x_{k+1} \neq x_k$ and $x_k \neq x_*$, for $k \geq 0$, and converges to x_*. Then $\{x_k\}$ converges superlinearly to x_* if and only if

$$\lim_{k \to +\infty} \frac{\|s_k - s_k^N\|}{\|s_k\|} = 0. \quad \square \tag{1.10}$$

Prove that the above two superlinear characterization theorems are equivalent.

5. Prove Theorem 1.1.3.

6. Let $q(x)$ be the quadratic form,

$$q(x) = c^T x + \frac{1}{2} x^T H x$$

where H is symmetric positive definite. The steepest algorithm for minimizing $q(x)$ can be stated:

> *Guess* $x_0 \in \Re^n$, *set* $k = 0$.
>
> *while* ($\|\nabla q(x_k)\| > tol$) *and* ($k <itbound$)
>
>   *Set* $d_k = -\nabla q(x_k)$
>   *Set* $x_{k+1} = x_k + \alpha_k^* d_k$, *where* $\alpha_k^* = argmin\{q(x_k + \alpha d_k) : \alpha > 0\}$.
>   *Set* $k = k + 1$

Figure 1.8: Steepest Descent on a Quadratic

Implement this method in MATLAB and try it on two 2-dimensional examples:

Problem 1: tol $= 10^{-10}$, itbound $= 100$, cond$(H) = 10$.

Problem 2: tol $= 10^{-10}$, itbound $= 100$, cond$(H) = 100$.

In each case choose $x_1 = (2.9, 2)^T$. For your test problem you may choose $c = 0$ and create H by first choosing the diagonal matrix of eigenvalues D and then choosing a random **orthogonal** matrix Q; set $H = Q^T D Q$. Plot the contours of your quadratic functions (use the contour statement) and plot the trajectory of the x-iterates on top of the contour map (for each problem). Repeat this experiment except replace the optimal step (in each iteration) by a near optimal step. Specifically, use

$$\alpha_k = (1 + .01 \cdot (2 \cdot rand - 1)) \cdot \alpha_k^*.$$

Comment on your results; keep in mind the more general situation where the function to be minimized is not quadratic, and determination of an optimal step size is not feasible.

7. M-file tbroy.m contains a 900-dimensional nonlinear test problem to be minimized. The assignment

$$[\mathsf{val,g}] = \mathsf{tbroy(x)}$$

evaluates the the function at $x \in \Re^n$. Implement steepest descent, Fletcher-Reeves, and the Polak and Ribière methods, using the MATLAB 1-D minimization function fmin for the line search, and try your implementations on test function tbroy. Start at $x_i = -1, i = 1 : n$ and terminate when either the norm of the gradient is less than 10^{-6} or the number of iterations exceeds 100. Compare the three approaches in terms of running time and number of function/gradient evaluations required. Next, try the more difficult problem,

$$[\mathsf{val,g}] = \mathsf{tbroy2(x,A)}$$

where A is a symmetric matrix available in MAT-file ps3.mat.

8. The symmetric secant update is given by $B_+ = B + C$ where C solves

$$\min\{\|C\|_F;\ Cs = r, C = C^T\} \tag{1.11}$$

and $r = y - Bs$. The solution is

$$C = \frac{rs^T + sr^T}{s^T s} - (s^T r)\frac{ss^T}{(s^T s)^2}. \tag{1.12}$$

This update can be derived by considering (1.11) and introducing an orthogonal matrix Q such that $Q^T[s, r]$ is an upper triangular n-by-2 matrix. Complete this derivation.

9. The symmetric secant update (1.12) can be expressed as $B_+ = B + C$ where

$$C = (s, r)\begin{pmatrix} a & b \\ b & 0 \end{pmatrix}\begin{pmatrix} s^T \\ r^T \end{pmatrix}$$

and a, b can be derived from (1.12). Consider the slightly more general form for C:

$$C = (s, r)\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} s^T \\ r^T \end{pmatrix}$$

and look at the question of defining a, b, c such that $Cs = r$ is satisfied and C is *positive semi-definite.*

10. Show that both the BFGS update

$$B_+ \;=\; B + \frac{yy^T}{s^T y} - \frac{Bss^T B}{s^T Bs}$$

and the DFP update

$$B_+ \;=\; B + \frac{y - Bs)y^T + y(y - Bs)^T}{y^T s} - \frac{[s^T(y - Bs)]yy^T}{(y^T s)^2}$$

can be written $B_+ = B + C$, where

$$C = (Bs, y)\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} (Bs)^T \\ y^T \end{pmatrix} \tag{1.13}$$

for suitable a, b, c. (The vector $y = \nabla f(x+s) - \nabla f(x)$.) Under the assumption that $s^T y > 0$ both these updates guarantee that B_+ is symmetric positive definite (provided B is positive definite), and guarantee satisfaction of the quasi-Newton equation $B_+ s = y$.

11. Consider updates of the form (1.13) and examine the question of determining a, b, c such that $Cs = r$ and B_+ is symmetric positive definite under the assumptions that B is symmetric positive definite and $s^T y > 0$.

12*. A line search procedure can achieve (guaranteed) convergence to a second-order point. Suppose that $x_{k+1} = x_k + \alpha_k d_k$ where α_k is chosen to satisfy

$$f(x_{k+1}) < f(x_k) + \sigma_l(\alpha_k g_k^T d_k + \frac{1}{2}\alpha_k^2 \min(d_k^T H_k d_k, 0)) \tag{1.14}$$

and

$$f(x_{k+1}) > f(x_k) + \sigma_u(\alpha_k g_k^T d_k + \frac{1}{2}\alpha_k^2 \min(d_k^T H_k d_k, 0)), \tag{1.15}$$

for $0 < \sigma_l < \frac{1}{2} < \sigma_u < 1$. The effect of condition (1.14) is to restrict the step size from being too large by demanding that the decrease in f be large enough relative to the quadratic prediction; condition (1.15) can be interpreted as restricting the step length from being relatively too small. Assume that $f \in C^2$, f is bounded below.

(i) Suppose that either $d_k^T g_k < 0$ or $d_k^T g_k = 0$ and $d_k^T H_k d_k < 0$. Show that there exists an interval (α_l, α_u) with $0 < \alpha_l < \alpha_u$, depending on k, such that conditions (1.14, 1.15) are satisfied for any point $\alpha \in (\alpha_l, \alpha_u)$.

(ii) Assume that $d_k^T g_k < 0 \ \forall k$ and the angle between d_k and g_k is bounded away from 90 degrees. Assume that α_k is chosen such that line search conditions (1.14, 1.15) hold. Show that every accumulation point is a first-order point, i.e., $\{\|g_k\|\} \longrightarrow 0$.

(iii) Assume that d_k has the property that if $g_k \neq 0$ then $d_k^T g_k < 0$ and the angle between d_k and g_k is bounded away from 90 degrees; if $\lambda_{min}(H_k) < 0$ then $d_k^T H_k d_k \leq \tau \cdot \lambda_{min}(H_k) \cdot \|d_k\|_2^2$ for $0 < \tau < 1$. Assume the line search conditions (1.14, 1.15) hold. Show that every accumulation point satisfies first and second-order necessary conditions.

(iv) In addition to the assumptions above, assume that if H_k is positive definite then d_k is the Newton direction; assume that an accumulation point $\bar{x}$ satisfies the second-order sufficiency conditions. Assume the line search conditions (1.14, 1.15) hold; moreover, assume that a unit step size is chosen, i.e, $\alpha_k = 1$, if $\alpha_k = 1$ satisfies the line search conditions (1.14,1.15). Prove that $\{x_k\} \rightarrow \bar{x}$ and the convergence rate is quadratic.

1.4 Minimization by Trust Region

Trust region methods (or restricted step methods) represent a powerful approach to unconstrained optimization, aggressively researched over the last decade. Trust region methods possess stronger theoretical convergence properties than line search methods; moreover, in practise trust region methods usually require fewer iterations, sometimes dramatically fewer. On the other hand the linear algebra cost (per iteration) is higher for a trust region method: therefore, (pure) trust region methods are cost-effective on highly nonlinear problems with expensive function evaluations. Recent advances in adapting trust region methods to the large-scale setting have proven quite successful.

We begin this section with background and motivation of trust region methods. Subsequently, we will consider the efficient use of trust region ideas tailored to large-scale problems.

Taylor's Theorem underpins most unconstrained minimization methods, and this is certainly true for trust region methods. If f is twice continuously differentiable then at $x_k \in \Re^n$,

$$f(x_k + s) = f(x_k) + \nabla f_k^T s + \frac{1}{2} s^T \nabla^2 f_k s + o(\|s\|^2). \tag{1.16}$$

Therefore the change in f is well-represented by $\nabla f_k^T s + \frac{1}{2} s^T \nabla^2 f_k s$ in a sufficiently small neighbourhood of $\bar{x}$. This leads immediately to the notion of the *trust region subproblem* (TRS) and a *trial step*. The trust region subproblem at x_k is

$$\min\{\nabla f_k^T s + \frac{1}{2} s^T \nabla^2 f_k s : \quad \|s\| \leq \Delta_k\}. \tag{1.17}$$

A *trial step* is generated by TRS at iteration k: clearly for Δ_k sufficiently small $f(x_k + s_k) < f(x_k)$.

Remarks:

(i) A major point is that TRS is well-posed regardless of the inertia of $\nabla^2 f_k$. There is no need to replace $\nabla^2 f_k$ with a positive definite approximation.

(ii) If Δ_k is large enough and $\nabla^2 f_k$ is positive definite then the Newton step, $s_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k$, is the TRS solution.

(iii) If the smallest eigenvalue is negative, i.e., $\lambda_{min}(\nabla^2 f_k) < 0$, then the TRS solution s_k satisfies $\|s_k\| = \Delta_k$. (To see this suppose the solution s_k is interior. But then it follows that s_k is an unconstrained local minimizer of an indefinite quadratic function, an impossibility.)

(iv) If $\lambda_{min}(\nabla^2 f_k) = 0$ then there *exists* a solution to TRS on the boundary. The solution may not be unique.

(v) We have not yet specified a norm $\| \cdot \|$ to use in TRS. In fact the strong trust region convergence results, e.g., Theorem 1.4.1, are independent of norm. However, the 2-norm, sometimes used with an additional scaling matrix, i.e., $\|D_k s\|_2 \leq \Delta_k$, admits efficient *global* solutions. Other norms such as the 1-norm or ∞-norm are sometimes used; however, efficient algorithms to generate global minimizers do not exist for such norms and local TRS solutions (typically obtained by an iterative procedure) yield weaker convergence results.

The Trust Region Method is outlined in Figure 1.9. The constants satisfy $0 < \tau_1 < \tau_2 < 1$ and $\gamma_1 > 1$, $\gamma_2 > 1$. Fletcher [33] chooses $\tau_1 = \frac{1}{4}$, $\tau_2 = \frac{3}{4}$, $\gamma_1 = 4$, and $\gamma_2 = 2$.

Convergence results for TRM are strong. We state a representative convergence result in Theorem 1.4.1; see Fletcher[33] for a proof.

Theorem 1.4.1 *Assume $\{x_k\}$ is generated by TRM (Figure 1.9) and $\{x_k\} \in R \subset \Re^n$ where R is a bounded region. Further, assume f is twice continuously differentiable on R and $\{\|\nabla^2 f(x)\|\}$ is bounded on R. Then there exists an accumulation point $\bar{x}$ satisfying second order necessary conditions for optimality (1.1.1).*

For $k = 1, \ldots$

Solve TRS $\rightarrow s_k$, $qpval_k \leftarrow \nabla f_k^T s_k + \frac{1}{2} s_k^T \nabla^2 f_k s_k$

$newval_k \leftarrow f(x_k + s_k)$

$ratio_k \leftarrow \frac{newval_k - f_k}{qpval_k}$

Adjust Δ_k:

If $ratio_k < \tau_1$, $\Delta_{k+1} \leftarrow \frac{\|s_k\|}{\gamma_1}$
elseif $ratio_k > \tau_2$ *and* $\|s_k\| = \Delta_k$, $\Delta_{k+1} \leftarrow \gamma_2 \cdot \Delta_k$
else $\Delta_{k+1} \leftarrow \Delta_k$
endif

Update x:

If $ratio_k \leq 0$, $x_{k+1} \leftarrow x_k$
else $x_{k+1} \leftarrow x_k + s_k$, **endif**

Figure 1.9: Trust Region Method (TRM)

Theorem 1.4.1 is stronger than convergence results for standard line search algorithms in that there is mention of the *second order* optimality conditions. Indeed if one of the accumulation points identified in Theorem 1.4.1 satisfies second-order sufficiency then the entire sequence converges to such a point, and the (local) rate of convergence is quadratic [33].

Theorem 1.4.1 is independent of the norm used in TRS. However, the proof of Theorem 1.4.1 depends on the determination of a global solution to TRS and the Euclidean norm (2-norm) is most useful for that purpose.

Note that $qpval_k$ is negative and so $ratio_k$ is positive if and only if $f(x_k+s_k) < f(x_k)$. Therefore, TRM updates x_k with $x_k + s_k$ whenever s_k provides decrease if f. In general $ratio_k$ is a measure of how well the TRS solution predicts the real change in f: the match is perfect if $ratio_k = 1$. If $ratio_k > 1$ then the true change is even better than predicted - a situation TRM rewards with an increase in Δ for the next round.

1.4.1 Analysis of the Trust Region Subproblem

The trust region subproblem is minimization of a quadratic function subject to a restriction on the size of the solution:

$$\min\{g^T s + \frac{1}{2} s^T H s : \ \|s\| \leq \Delta\}. \tag{1.18}$$

We begin with a useful characterization of the global solution to (1.18) when the norm used is the 2-norm, i.e., $\|s\| = \|s\|_2$.

Theorem 1.4.2 *Vector s solves (1.18) with $\|\cdot\| = \|\cdot\|_2$, if and only if there exists a scalar $\lambda \geq 0$ such that*

$$(H + \lambda I)s = -g, \tag{1.19}$$

$$(H + \lambda I) \text{ is positive semidefinite,} \tag{1.20}$$

$$\|s\| \le \Delta, \quad and \tag{1.21}$$

$$(\|s\| - \Delta)\lambda = 0. \tag{1.22}$$

Sorensen [75] gives the following simple proof. If (s_*, λ_*) satisfy conditions (1.19) - (1.20) then s_* is the unconstrained minimizer of the quadratic function $g^T s + \frac{1}{2} s^T (H + \lambda_* I)s$. From this it follows that

$$g^T s_* + \frac{1}{2} s_*^T H s_* \le g^T s + \frac{1}{2} s^T H s + \frac{\lambda_*}{2}(\|s\|_2^2 - \|s_*\|_2^2). \tag{1.23}$$

Since the last term in (1.23) is non-positive for all feasible s, by (1.21, 1.22), it follows that s_* is a global solution to (1.18). On the other hand, if s_* solves (1.18) the either s_* is *strictly* feasible or it is not. In the former case the result is trivial; in the latter, conditions (1.19), (1.21), and (1.22) follow from local necessary conditions for optimality. To establish (1.20) note that since s_* is the global solution to (1.18) it follows that for all feasible s,

$$g^T s + \frac{1}{2} s^T H s \ge g^T s_* + \frac{1}{2} s_*^T H s_*. \tag{1.24}$$

Substituting $-(H + \lambda_* I)s_*$ for g in (1.24) yields

$$\frac{1}{2}(s - s_*)^T (H + \lambda_* I)(s - s_*) \ge 0, \tag{1.25}$$

which implies that $H + \lambda_* I$ is positive semi-definite.

The utility of this result is best revealed after diagonalization. Suppose $H = V \Lambda V^T$ where the columns of V are the orthonormal eigenvectors of H and

$$\Lambda = diag(\lambda_1, ..., \lambda_n), \quad \lambda_1 \le \lambda_2 \le ... \le \lambda_n. \tag{1.26}$$

Obviously then $H + \lambda I \;=\; V(\Lambda + \lambda I)V^T$ and so (1.19) is equivalent to

$$(\Lambda + \lambda I)\bar{s} = \alpha \stackrel{def}{=} -V^T g, \quad s \leftarrow V\bar{s}. \tag{1.27}$$

By (1.20), $\lambda \ge -\lambda_1$, and so all vectors s satisfying (1.19, 1.20) are of the form illustrated in Figure 1.10.

$$\boxed{s = \sum_{\{i:\lambda_i + \lambda > 0\}} (\frac{\alpha_i}{\lambda_i + \lambda}) v_i \;+\; \sum_{\{i:\lambda_i + \lambda = 0\}} \beta_i v_i. \qquad (1.28)}$$

Figure 1.10: TRS Solutions

The vector β can sometimes help with respect to satisfying (1.22). A basis for an algorithmic approach to this problem is to assume the form given in (1.28) and strive to satisfy (1.21, 1.22) by choosing λ, and in some cases β, appropriately (β plays a role only if the optimal value $\lambda_* = -\lambda_1$)

The situation where (1.21, 1.22) can be satisfied with $\|s\| < \Delta$ and $\lambda = 0$ is easily dispensed with (first half of Case 1 below). Therefore, the primary task, assuming form (1.28), is to determine λ, and sometimes β, to satisfy

$$\|s\| - \Delta = 0. \tag{1.29}$$

We divide our approach into three possibilities. Define $\mathcal{I} = \{i : \lambda_i = \lambda_1\}$.

Case 1: $\lambda_1 > 0$. In this case either the Newton step is within the sphere $\|s\|_2 \le \Delta$ or it is not. If $\|H^{-1}g\| \le \Delta$ then the optimal solution is $s_* = -H^{-1}g,\ \lambda_* = 0$. Otherwise, for $\lambda \ge 0$ define $s(\lambda) = \sum_{i=1}^{n}(\frac{\alpha_i}{\lambda_i+\lambda})v_i$, $\|s(\lambda)\| \to 0$ as $\lambda \to \infty$.

Case 2: $\lambda_1 \le 0$ and for some $i \in \mathcal{I}, \alpha_i \ne 0$. For $\lambda > -\lambda_1$, define $s(\lambda) = \sum_{i=1}^{n}(\frac{\alpha_i}{\lambda_i+\lambda})v_i$. Obviously $\|s(\lambda)\| \to \infty$ as $\lambda \to -\lambda_1^+$, and $\|s(\lambda)\| \to 0$ as $\lambda \to \infty$. Moreover, $\|s(\lambda)\|$ is convex; therefore, $\|s(\lambda)\|$ intersects Δ in exactly one place for $\lambda > -\lambda_1$.

Case 3: $\lambda_1 \le 0$ and $\alpha_i = 0 \forall i \in \mathcal{I}$. Let $s_1(\lambda) = \sum_{i \notin \mathcal{I}}(\frac{\alpha_i}{\lambda_i+\lambda})v_i$ for $\lambda > -\lambda_1$. There are now two possibilites. If $\|s_1(-\lambda_1)\| \ge \Delta$ then there is a solution to (1.29) to the right of $-\lambda_1$, $s(\lambda) = s_1(\lambda)$. Otherwise, $s = s_1(-\lambda_1) + \sum_{i \in \mathcal{I}} \beta_i v_i$ with β chosen to ensure (1.29), and $\lambda_* = -\lambda_1$. Note that if $|\mathcal{I}| > 1$ then the nullspace component of s may not be unique.

The Reciprocal Secular Equation. In theory we can build an algorithm on the remarks given above. However it is better, numerically, to replace condition (1.29) with

$$rsec(\lambda) \stackrel{def}{=} \frac{1}{\Delta} - \frac{1}{\|s(\lambda)\|} = 0. \tag{1.30}$$

Equation (1.30) is more linear in shape than equation (1.29); therefore, equation (1.30) is more amenable to solution via Newton's method.

Considering the definition of $rsec(\lambda)$,

$$rsec(\lambda) = \frac{1}{\Delta} - \frac{1}{[(\frac{\alpha_1}{\lambda_1+\lambda})^2 + (\frac{\alpha_2}{\lambda_2+\lambda})^2 + \dots + (\frac{\alpha_n}{\lambda_n+\lambda})^2]^{\frac{1}{2}}}, \tag{1.31}$$

it is easy to verify the following:

(i) $rsec(\lambda)$ is convex on $(-\lambda_1, \infty]$ and $\lim_{\lambda\to\infty} rsec(\lambda) = -\infty$.

(ii) $lim_{\lambda\to-\lambda_1^+} rsec(\lambda)$ is finite.

(iii) If $\alpha_i \ne 0$, for some $i \in \mathcal{I}$, then $rsec(-\lambda_1) \stackrel{def}{=} \lim_{\lambda\to-\lambda_1^+} rsec(\lambda) = \frac{1}{\Delta}$. Obviously a single zero of $rsec$ exits to the right of $-\lambda_1$ in this case.

(iv) If $\alpha_i = 0\ \forall i \in \mathcal{I}$, then $rsec(-\lambda_1) \stackrel{def}{=} \lim_{\lambda\to-\lambda_1^+} rsec(\lambda) = \frac{1}{\Delta} - \frac{1}{[\sum_{i\notin\mathcal{I}}(\frac{\alpha_i}{\lambda_i-\lambda_1})^2]^{\frac{1}{2}}}$.

Algorithmic and Numerical Concerns. We assume that a solution to (1.18) is sought and we compute full eigenvalue information, $H = V\Lambda V^T$.

A method using $H = V\Lambda V^T$ appears to be straightforward.

Case 1: $\lambda_1 > 0$. If $\|H^{-1}g\| \le \Delta$ then the Newton step, $s = -H^{-1}g$, is the solution with $\lambda_* = 0$. If $\|H^{-1}g\| > \Delta$ then we can determine the zero of $rsec(\lambda)$, $\lambda > 0$.

Case 2: $\lambda_1 \le 0$ $\alpha_i \neq 0$ for some $i \in \mathcal{I}$. Then $\lim_{\lambda \to -\lambda_1^+} rsec(\lambda) = \frac{1}{\Delta}$ and $rsec$ admits a single solution to the right of $-\lambda_1$.

Case 3: $\lambda_1 \le 0$ *and* $\forall i \in \mathcal{I}, \alpha_i = 0$. Let $s = s_1(\lambda) + \beta v_1$ where $s_1(\lambda) = \sum_{i \notin \mathcal{I}} (\frac{\alpha_i}{\lambda_i + \lambda}) v_i$. If $\|s_1(-\lambda_1)\| \ge \Delta$ then there is a zero of $rsec(\lambda)$ to the right of $-\lambda_1$ with $s(\lambda) = s_1(\lambda)$. Otherwise, a solution to (1.18) is given by (1.28), $s = s_1(-\lambda_1) + \sum_{i \in \mathcal{I}} \beta_i v_i$, with $\sum \beta_i^2 = \Delta^2 - \|s_1(-\lambda_1)\|^2$, and $\lambda_* = -\lambda_1$.

Unfortunately, the situation is not so clean from a numerical point of view: the separation between the second and third cases is not sharp. In particular, if α_1 is small the equation $rsec(\lambda) = 0$ is very ill-conditioned for λ near $-\lambda_1$ and it can be quite difficult (nigh impossible) to compute λ such that $rsec(\lambda)$ is small. This situation is illustrated in Figure 1.11. In this problem $\lambda_{min}(H) = -1$ and therefore we are interested in a zero of $rsec$ to the right of $\lambda = 1$. The three plots in Figure 1.11 correspond to different values of α_1, the coefficient corresponding to the smallest eigenvalue. The solid graph corresponds to $\alpha_1 = .1$, the "dash-dot" graph corresponds to $\alpha = .01$, and the "dash-dash" graph corresponds $\alpha = .001$. Notice the extreme ill-conditioning of $rsec$ near $\lambda = 1$, as α_1 gets small. The limiting graph, as $\alpha_1 \to 0$, is (almost) a straight line on the interval shown with function value $rsec = -1$ at $\lambda = 1$. Notice that if $\alpha_1 = 0$ then $rsec(\lambda)$ does not have a zero on $[1, \infty)$.

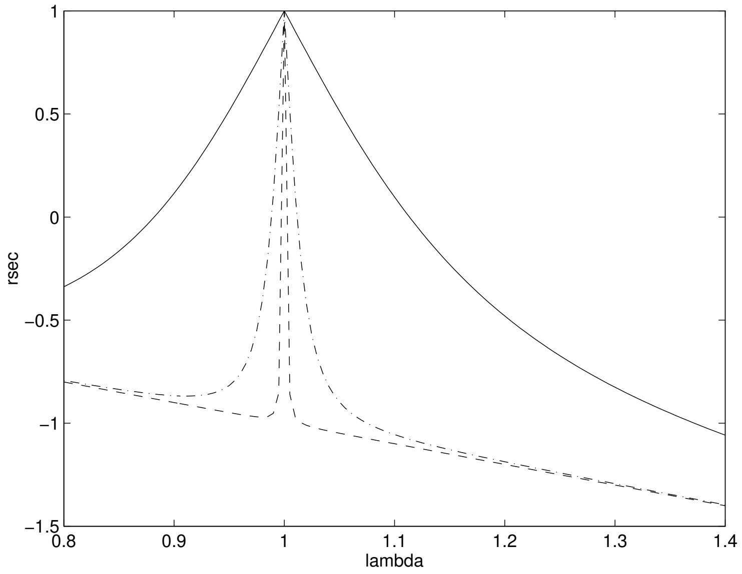

Figure 1.11: The Reciprocal Secular Equation (rsec)

This extreme ill-conditioning is due to the following "disagreement".
Assume $\mathcal{I} = \{1\}$, for simplicity, and note that, if $\alpha_1 = 0$,

$$rsec(-\lambda_1) \stackrel{def}{=} \lim_{\lambda \to -\lambda_1^+} rsec(\lambda) = \frac{1}{\Delta} - \frac{1}{[(\frac{\alpha_2}{\lambda_2 - \lambda_1})^2 + \ldots + (\frac{\alpha_n}{\lambda_n - \lambda_1})^2]^{\frac{1}{2}}}. \tag{1.32}$$

On the other hand, if $\alpha_1 \neq 0$, $rsec(-\lambda_1) \stackrel{def}{=} \lim_{\lambda \to -\lambda_1^+} rsec(\lambda) = \frac{1}{\Delta}$ which is not, in general, the limiting value of (1.32). Therefore nearby problems ($\alpha_1 = 0$ .vs. $\alpha_1 = \epsilon$) can yield very different solutions to (1.30), and this is the cause of the ill-conditioning of (1.30). One solution to this ill-conditioning problem is to first *attempt* to find a solution to $rsec(\lambda) = 0, \lambda \geq -\lambda_1$. However, if $|rsec(\bar{\lambda})|$ is not small, where $\bar{\lambda}$ is the computed "zero" returned by the zero-finder, then set $\lambda_* = -\lambda_1$ and compute a solution to (1.18) via (1.28).

This strategy works because the solution to (1.30) with α_i small for $i \in \mathcal{I}$ is close to a solution of (1.18) with the corresponding α_i at zero. To see this, initially assume that $\mathcal{I} = \{1\}$; that is, $\lambda_1 < \lambda_2$.

Now first consider the case where $\alpha_1 = 0$. Then the solution to (1.18) is:

$$s = \sum_{i>1} (\frac{\alpha_i}{\lambda_i - \lambda_1}) v_i + \beta v_1, \tag{1.33}$$

where $\beta^2 = \Delta^2 - \sum_{i>1}(\frac{\alpha_i}{\lambda_i - \lambda_1})^2$ (we assume $\Delta^2 < \sum_{i>1}(\frac{\alpha_i}{\lambda_i - \lambda_1})^2$; otherwise, the result is obvious). Define $s_1 = \sum_{i>1}(\frac{\alpha_i}{\lambda_i - \lambda_1})v_i$; hence, the solution to (1.18) can be written $s = s_1 + \beta v_1$.

Next consider $\alpha_1 = \epsilon$, a small number. We can write the solution to (1.18) as

$$s(\lambda) = \sum_{i>1} (\frac{\alpha_i}{\lambda_i + \lambda}) v_i + (\frac{\alpha_1}{\lambda_1 + \lambda}) v_1. \tag{1.34}$$

But if $\alpha_1 \to 0$, then $\lambda \to -\lambda_1$ (to keep $\|s(\lambda)\| = \Delta$). But if $\lambda \to -\lambda_1$, then $\sum_{i>1}(\frac{\alpha_i}{\lambda_i + \lambda})v_i \to s_1$, which implies

$$(\frac{\alpha_1}{\lambda_1 + \lambda})^2 \to \beta^2, \ as\ \alpha_1 \to 0. \tag{1.35}$$

Therefore, the solution to (1.18) with $\alpha_1 = \epsilon$ is near to the solution to (1.18) with $\alpha_1 = 0$.

In general, if $|\mathcal{I}| > 1$ a solution to (1.18), with some components α_i near to zero, is near to a solution of (1.18) with those components set to zero. In this case, where several coefficients α_i equal to zero, $i \in \mathcal{I}$, problem (1.18) does not enjoy a unique solution [see (1.28)] but the range space component is unique.

For further information and analysis of TRS, see [36, 62, 73, 12, 15, 75]. A trust region solver, using the eigen-decomposition as discussed above, is available to you:

$$\mathsf{s = trust(g,H,delta).}$$

1.4.2 The Numerical Solution of TRS

The primary task in the solution of TRS is to solve the reciprocal secular equation (1.30). The analysis in §1.4.1 indicates how this can be done given the eigen-decomposition of H. However, except for small problems, this is not a practical solution.

An alternative approach, which allows for approximate solutions to (1.30), is to return to equation (1.19) and define $s(\lambda) = -(H + \lambda I)^{-1} g$, for $\lambda > -\lambda_1$. A one-dimensional Newton process can be applied to (1.30) where each evaluation of $s(\lambda)$ involves two steps:

(i) Factor $H + \lambda I \ = \ LL^T$,

(ii) Solve $(LL^T)s \ = \ -g$.

If $H + \lambda I$ is close to singular, i.e., λ approaches $-\lambda_1$ from the right, then as indicated in §1.4.1, an (approximate) eigenvector corresponding to the smallest eigenvalue may be required. Moré and

Sorensen and [62] suggest using the current factorization, $H + \lambda I = LL^T$, and the Linpack [26] condition number estimation technique to compute an approximate eigenvector. Other approximation schemes are possible.

This approach is feasible for moderate sized problems; however, it is not economical for most large-scale problems. First, a direct factorization is required (to estimate the condition of $H + \lambda I$ and possibly to determine an approximate eigenvector). Second, several factorizations of matrix $H + \lambda I$ are required per iteration (depending on the number of iterations required to obtain an approximate solution to (1.30)). This is costly for large dimension. Sorensen **??** proposes a method, more suitable for large-scale problems, based on an Arnoldi/Lanczos process.

1.4.3 Heuristic approximations to TRS

The Dogleg Algorithm of Powell [70] can be viewed as a heuristic approximation to TRS. Motivation for this approach is based on observing the TRS solution trajectory as a function of Δ (when H is positive definite). In particular, the solution path, starting at the origin, is initially tangential to the negative gradient direction and then curves to the Newton step. See Figure 1.12.

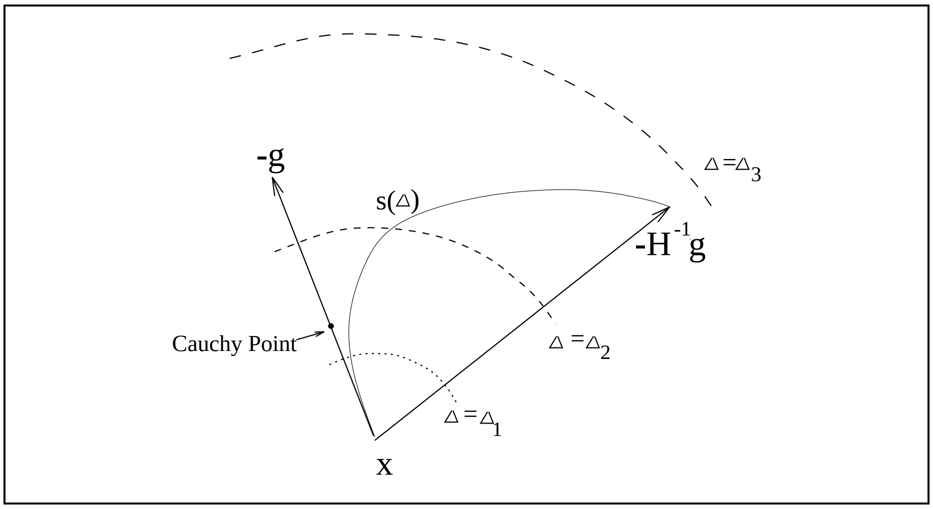

Figure 1.12: TRS Solution Trajectory

An approximation to this path is the two-segment dogleg path: Let x_{cp} be the *Cauchy Point*, i.e., the point that minimizes the quadratic function $q(s) = g^T s + \frac{1}{2} s^T H s$ in the negative gradient direction $d = -g$. The second segment of this two-legged path connects x_{cp} to the Newton step, $s = -H^{-1}g$. The dogleg idea is to minimizes $q(s)$ along this two-legged path, subject to satisfying $\|s\| \leq \Delta$. Given $s = -H^{-1}g$ this procedure is easy to implement, requiring the computation of the Cauchy point and the point of intersection of the dogleg path with the sphere $\|s\| = \Delta$. Figure 1.13 illustrates a dogleg path.

The Dogleg Algorithm has the advantage that only a single factorization is required each iteration. The disadvantage is that the indefinite case is poorly motivated: what does the dogleg path mean when H is indefinite? The Dogleg Algorithm makes no effective use of negative curvature: when H is indefinite the Dogleg solution has little relationship with the solution to (1.18).

Steihaug's Dogleg Generalization. Steihaug [76] suggests the following heuristic approximation to

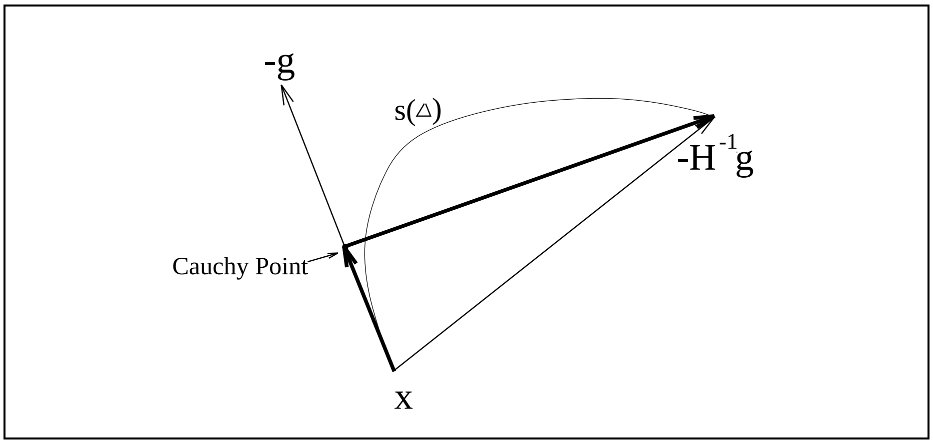

Figure 1.13: The Dogleg Path

TRS. Let C be a SPD preconditioner for matrix H. Consider applying Algorithm PCG, Figure **??**, to the system $Hs = -g$. As noted in (§**??**, Question 3.), the current PCG solution is monitonically increasing in the C-norm. Therefore if we use the C-norm in TRS (1.18), Algorithm PCG can be modified to terminate when the current PCG solution crosses Δ. The Generalized Dogleg Algorithm is given in Figure 1.14.

The Generalized Dogleg Algorithm is an appealing way to solve large-scale minimization problems. It is simple to implement and represents a natural union of the PCG Algorithm for linear systems and the Dogleg Algorithm. Under reasonable assumptions superlinear convergence can be achieved [76]. However, the strong convergence results of the trust region method, e.g., Theorem 1.4.1, are no longer valid since there is no guarantee on the quality of the solution produced vis-a-vis (1.18). At best convergence to a first-order point can be guaranteed.

A Subspace Approach. An alternative way to adapt the trust region approach to the large-scale setting is proposed in [12]; subsequently Branch, Coleman, and Li [9] introduced preconditioned conjugate gradients into this format (and extended the methodology to handle box contraints).

The main idea is to define a subspace $\mathcal{S}_k$ and replace TRS in the trust region method (Figure 1.9), with

$$\min_s\{s^T\nabla f_k + \frac{1}{2}s^T\nabla^2 f_k s: \ \|s\|_2 \le \Delta_k, \ s \in \mathcal{S}_k\}. \tag{1.36}$$

Clearly if the dimension of $\mathcal{S}_k$ is small enough then (1.36) can be solved exactly, applying the ideas discussed in §1.4.1 to the eigen-decomposition of the reduction of H onto $\mathcal{S}_k$. How large does $\mathcal{S}_k$ have to be to ensure the strong convergence results of trust region methods, e.g., Theorem 1.4.1? The answer is encouraging: Byrd, Schnabel, and Shultz [12] show that, in principle, it is possible to choose $\mathcal{S}_k$ with $|\mathcal{S}_k| = 2$.

Theorem 1.4.3 *Let $\mathcal{S}_k =< Y_k >$, where Y_k is a a matrix with orthonormal columns. Let ϵ, τ, be constants: $\epsilon > 0$, $0 < \tau < 1$. Assume $\{x_k\} \in R \subset \Re^n$, a bounded region, and $f \in C^2(R)$. Further, assume $\mathcal{S}_k$ is a subspace satisfying*

(i) $\nabla f_k \in \mathcal{S}_k$

While $\|z\| > tol$

1. $d \leftarrow z + \beta d_-$, $\gamma \leftarrow d^T H d$
2. **if** $\gamma \leq 0$

    compute τ *s.t.* $\|s + \tau d\|_C = \Delta$

    $s \leftarrow s + \tau d$

    return(s, d)

   else $\alpha \leftarrow \frac{r^T z}{\gamma}$

    if $\|s + \alpha d\|_C > \Delta$

     compute τ *s.t.* $\|s + \tau d\|_C = \Delta$

     $s \leftarrow s + \tau d$

     return(s, d)

    else $s \leftarrow s + \alpha d$, $r \leftarrow r - \alpha H d$ **endif**

   endif
3. *Solve* $Cz = r$
4. $\beta \leftarrow \frac{r^T z}{r_-^T z_-}$

return(s)

Figure 1.14: Generalized Dogleg Algorithm

(ii) $\lambda_{min}(\nabla^2 f_k) < 0 \Longrightarrow \lambda_{min}(Y_k^T \nabla^2 f_k Y_k) \leq \max\{-\epsilon, \tau\lambda_{min}(\nabla^2 f_k)\}$.

Then, there exists an accumulation point of the subspace trust region method, $\bar{x}$*, satisfying first and second-order necessary optimality conditions.*

Further, assume that that if $\lambda_{min}(\nabla^2 f_k) > 0$ *then* $\mathcal{S}_k$ *contains the Newton step* $-\nabla^2 f_k^{-1} \nabla f_k$*. If we also assume that an accumulation point* $\bar{x}$ *(referred to above) also satisfies second-order sufficiency conditions, then* $\{x_k\} \to \bar{x}$ *and the convergence rate is quadratic.*

It is difficult to ensure condition (ii) in Theorem 1.4.3 in practise. The implementation in MATLAB M-file sfmin uses Algorithm PCG (Figure **??**) applied to the Newton system, $\nabla^2 f_k d = -\nabla f_k$. The output from sfmin is either an approximate Newton step, $d_k = d_k^{Newt}$, or a direction of nonpositive curvature of matrix $\nabla^2 f_k$, $d_k = d_k^{neg}$. Subspace $\mathcal{S}_k$ is then defined by the space spanned by the output of PCG and the gradient of f: $\mathcal{S}_k =< d_k, \nabla f_k >$. Therefore sfmin does not *guarantee* that sufficient negative curvature is found; however, if sufficient negative curvature is detected by Algorithm PCG, then the strong convergence property of Theorem 1.4.3 will hold for sfmin.

The Trust Region Subspace Subproblem approach (TRSS) is an effective way to solve large-scale optimization problems. It bears some similarity to the Generalized Dogleg Algorithm proposed by Steihaug (discussed above); however, numerical results reported in [9] indicate that TRSS is more efficient. In general TRSS does a better job of picking up negative curvature.

The MATLAB M-file, x = sfmin(fun,xstart,options) is available to you: see the Appendix §?? for more detail.

1.5 Nonlinear systems.

Nonlinear least-squares problems represent the most common class of nonlinear minimization problems. The general methods discussed in §1.4 can be applied to this particular case; however, nonlinear least-squares have distinctive structure and exploiting this structure is often cost effective.

If F is a vector mapping, $F : \Re^n \to \Re^m$, with component i of F denoted by f_i, then the nonlinear least-squares problem is:

$$\min f(x) \stackrel{def}{=} \frac{1}{2}\|F(x)\|_2^2 = \frac{1}{2}\sum_{i=1}^{m} f_i(x)^2. \tag{1.37}$$

The gradient of f is $\nabla f(x) = J^T F(x)$, where $J = J(x)$ is the m-by-n Jacobian of F, and the Hessian of f is:

$$\nabla^2 f(x) = J(x)^T J(x) \ + \ \frac{1}{2}\sum_{i=1}^{m} f_i(x)\nabla^2 f_i(x). \tag{1.38}$$

Typically $m \geq n$. Square nonlinear systems of equations,

$$\textit{solve} \quad F(x) = 0,$$

where $F : \Re^n \to \Re^n$, are often solved using nonlinear least-squares technology. This is not a fail-safe approach in that a local minimizer of $f(x) \stackrel{def}{=} \frac{1}{2}\|F(x)\|_2^2$ is not necessarily a zero of F; however, a zero of F is certainly a global minimizer of f.

Equation (1.38) tells the structure story: If the residuals, $f_i(x)$, are small at the solution $x = x_*$, then we can ignore the last term in (1.38) and approximate $\nabla^2 f(x)$ with $J(x)^T J(x)$. Notice that this Hessian approximation involves only the use of first derivatives. Moreover, provided $\|F(x_*)\|$ is small enough, Theorem 1.1.1 supports the use of this approximation in a local procedure, i.e., the Gauss-Newton Algorithm given in Figure 1.15.

Guess starting point x_0; $k \leftarrow 0$.
Repeat

(i) Evaluate $F(x_k), J_k$;

(ii) Solve $J_k s_k \stackrel{\text{l.s.}}{=} -F(x_k)$.

(iii) Update: $x_{k+1} \leftarrow x_k \ + \ s_k, \ k \leftarrow k+1$

Figure 1.15: The Gauss-Newton Algorithm

By Theorem 1.1.1, the iterates generated by the Gauss-Newton Algorithm will converge linearly to x_* provided x_0 is sufficiently close to x_* and $\|F(x_*)\|$ is small enough. Moreover, the linear contraction factor in Theorem 1.1.1 is proportional to $\|F(x_*)\|$.

The trust region model is a natural way to globalize the Gauss-Newton Algorithm. Algorithm TRM, Figure 1.9, can be used with the Trust Region Subproblem (TRS) defined by:

$$\min\{\|J_k s + F_k\|_2 : \quad \|s\| \leq \Delta_k\}. \tag{1.39}$$

The TRS analysis given in §1.4.1 is applicable here: replace H in (1.18) with $J^T J$, and g with $J^T F$. Indeed, the analysis simplifies here since the matrix $H = J^T J$ is always positive semi-definite.

Following the analysis in §1.4.1, the solution to (1.39) will involve solving

$$\begin{pmatrix} J_k \\ \sqrt{\lambda} I \end{pmatrix} s \overset{\text{l.s.}}{=} \begin{pmatrix} -F_k \\ 0 \end{pmatrix}, \tag{1.40}$$

for different values of λ (for fixed index k). If (1.39) is solved using orthogonal transformations then, assuming J_k is a dense matrix, the cost is $O(mn^2)$ flops. However, if the upper triangular factor of of J_k is already computed, then the cost of solving (1.39) for a subsequent value of λ is $O(n^3)$, [60]. This is important because many nonlinear least-squares problem have $m >> n$. The Minpack [61] framework for solving TRSS is given in Figure 1.16.

1. *Determine the unconstrained solution* s_u*:* $\min \|Js - F\|_2$.
2. **If** $\|s_u\| \leq \Delta$, $s_* \leftarrow s_u$, $\lambda_* \leftarrow 0$
   else *solve* $\min\{\|Js + F\|_2 : \quad \|s\| = \Delta\} \rightarrow (\lambda_*, s_*)$.

Figure 1.16: Minpack Approach to TRSS

1.5.1 Large Sparse Nonlinear Least-Squares.

Adaptation of the trust region approach to the large-scale nonlinear least-squares setting is similar to the general situation described in §1.4.3. For example, a dogleg approximation is possible (see Figures 1.12, 1.13). In this case the Newton step is the least-squares solution, $J_k s_k^{Newt} \overset{\text{l.s.}}{=} -F_k$, and the gradient is $\nabla f_k = J_k^T F_k$. In the square case, i.e., $m = n$, the Newton step simplifies to the linear system $J_k s_k^{Newt} = -F_k$. The Minpack approach to nonlinear systems of equations follows these lines with an additional approximation: The Jacobian matrix is approximated by a secant update [10] and the gradient direction is replaced with $B_k^T F_k$. Clearly this approach could be extended to the large sparse realm using both sparse LU techniques (see §**??**) and possibly sparsity-preserving secant updates [72, 57].

The approaches based on linearized conjugate gradients discussed in §1.4.3 can also be tailored to this setting, applying Algorithm PCG (Figure **??**) to the normal system

$$J_k^T J_k s = -J_k^T F_k. \tag{1.41}$$

Of course it is not necessary to explicitly form the Hessian approximation, $\tilde{H}_k = J_k^T J_k$, since PCG requires only matrix-vector products with $\tilde{H}_k$. On the other hand, an effective preconditioning strategy is very important since the conditioning of $\tilde{H}_k$ will affect the convergence behaviour of PCG and the conditioning number of $\tilde{H}_k$ is the square of the condition number of J_k.

The subspace approach discussed in §1.4.3 can be adapted, in a straightforward way, to the nonlinear least-squares setting. In this case the 2-dimensional subspace $\mathcal{S}_k$ is defined

$$\mathcal{S}_k = < J_k^T F_k, d_k^{PCG} >,$$

where d_k^{PCG} is an approximate solution to (1.41), obtained by Algorithm PCG (Figure **??**).

1.6 Exercises

1. Let A be an m-by-n matrix of rank $r \le n \le m$. For $\lambda \ge 0$ define

$$x(\lambda) = (A^T A + \lambda I)^{-1} A^T b.$$

Let x_* be the least squares solution of *minimum norm* to the overdetermined system $Ax \overset{\text{l.s.}}{=} b$. Prove that $x(\lambda) \to x_*$ as $\lambda \to 0^+$.

2. Consider the "dogleg" method for locating a zero of a system of nonlinear equations, $F(x) = 0$, where $F : \Re^n \to \Re^n$ (see §1.5). So at each iteration k a "trial step" is computed by solving

$$\min_s \{\|B_k s + F_k\|_2^2 : \|s\|_2 \le \Delta_k, s \in \mathcal{P}_k\}$$

where $\mathcal{P}_k$ is the "dogleg path" and B_k is a nonsingular approximation to the Jacobian of F. Prove that

(i) $\|B_k s + F_k\|_2^2$ is strictly decreasing along $\mathcal{P}_k$

(ii) $\|s\|_2$ is strictly increasing along $\mathcal{P}_k$.

3*. A subspace trust region method determines a trial step by solving the problem:

$$\min_s \{s^T g_k + \frac{1}{2} s^T H_k s : \ \|s\|_2 \le \Delta_k, \ s \in \mathcal{S}_k\} \tag{1.42}$$

where $\mathcal{S}_k$ is a subspace of $\mathcal{R}^n$. Let s_k be a global solution to (1.42). Consider the subspace trust region method discussed in §1.4.3.

i) Prove the following:

Theorem: Let $\mathcal{S}_k = < Y_k >$, where Y_k is an orthonormal matrix. Let ϵ, τ, be constants: $\epsilon > 0$, $0 < \tau < 1$. Assume $\{x_k\} \in R \subset \Re^n$, a bounded region, and $f \in C^2(R)$. Further, assume $\mathcal{S}_k$ is a subspac e satisfying

1. $g_k \in \mathcal{S}_k$
2. $\lambda_{min}(H_k) < 0 \Longrightarrow \lambda_{min}(Y_k^T H_k Y_k) \le \max\{-\epsilon, \tau \lambda_{min}(H_k)\}$.

Then, there exists an accumulation point of the subspace trust region method, $\bar{x}$, satisfying first and second-order necessary optimality conditions.

---

ii) Assume that $\mathcal{S}_k$ has the additional property that if $\lambda_{min}(H_k) > 0$ then $-H_k^{-1}g_k \in \mathcal{S}_k$. If we further assume that an accumulation point $\bar{x}$ (referred to above) also satisfies second-order sufficiency conditions, show that $\{x_k\} \to \bar{x}$ and the convergence rate is quadratic.

Bibliography

[1] L. Adams and J. Nazareth, *Linear and Nonlinear Conjugate Gradient-Related Methods*, SIAM, 1996.

[2] N. Alexandrov and M. Y. Hussani, *Multidisciplinary Design Optimization. Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization*, SIAM, 1997.

[3] O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.

[4] L. Biegler, T. Coleman, A. Conn, and F. Santosa, *Large-scale Optimization with Applications. Part II: Optimal Design and Control*, vol. 93 of The IMA Volumes in Mathematics and Its Applications, Springer, 1997.

[5] G. Birkhoff and C. R. De Boor, *Piecewise polynomial interpolation and approximation*, in Approximation of Functions, H. L. Garabedian, ed., Elsevier (New York), 1965, pp. 164–190.

[6] Å. Bjorck, *Stability analysis of the method of semi-normal equations for linear least squares problems*, Tech. Rep. CMA-R26-85, Centre for Mathematical Analysis, Australian National University, 1985.

[7] ——, *Numerical Methods for Least Squares Problems*, SIAM, 1996.

[8] J. Borggaard, J. Burns, E. Cliff, and S. Schreck, *Computational Methods for Optimal Design and Control. Proceedings of the AFOSR Workshop on Optimal Design and Control, Arington Virginia*, Progress in Systems and Control Theory, Birkha´user, Boston, 1997.

[9] M. A. Branch, T. F. Coleman, and Y. Li, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, Tech. Rep. CTS95TR217 (to appear in SISC), ACRI/Theory Center, Cornell University, 1995.

[10] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Mathematics of Computation, 19 (1965), pp. 577–593.

[11] ——, *The convergence of a class of double-rank minimization algorithms*, J.I.M.A., 6 (1970), pp. 222–236.

[12] R. H. Byrd, R. B. Schnabel, and G. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Mathematical Programming, 40 (1988), pp. 247–263.

[13] T. F. Coleman, *Large Sparse Numerical Optimization*, Lecture Notes in Computer Science, Volume 165, Springer-Verlag, 1984.

[14] T. F. Coleman, A. Edenbrandt, and J. R. Gilbert, *Predicting fill for sparse orthogonal factorization*, J. Assoc. Comput. Mach., 33 (1986), pp. 517–532.

[15] T. F. Coleman and C. Hempel, *Computing a trust region step for a penalty function*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 180–201.

[16] T. F. Coleman, Y. Li, and A. Verma, *Reconstructing the unknown local volatility function*, Tech. Rep. TR 98-1706, Computer Science Department, Cornell University, 1998, (*To Appear in J. of Computational Finance*).

[17] T. F. Coleman and P. E. Plassmann, *Solution of nonlinear least-squares problems on a multiprocessor*, in Parallel Computing 1988, Shell Conference Proceedings, G. van Zee and J. van de Vorst, eds., Springer-Verlag, Lecture Notes in Computer Science, 384, 1989.

[18] T. F. Coleman, F. Santosa, and A. Verma, *Semi-automatic differentiation*, in Computational Methods for Optimal Design and Control. proceedings of the AFOSR Workshop on Optimal Design and Control, Arington Virginia, J. Borggaard, J. Burns, E. Cliff, and S. Schreck, eds., Progress in Systems and Control Theory, Birkha´user, Boston, 1997.

[19] T. F. Coleman and C. Sun, *Parallel orthogonal factorizations of large sparse matrices on distributed-memory multiprocessors*, in Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing, SIAM, Philadelphia, 1993, pp. 457–461.

[20] G. Crippen and T. Havel, *Distance Geometry and Molecular Conformation*, John Wiley and Sons, 1988.

[21] W. C. Davidon, *Variable metric methods for minimization*, Tech. Rep. ANL-5990, Argonne National Laboratory, 1959.

[22] C. de Boor, *Spline Toolbox for use with MATLAB*, The Math Works Inc., 1997.

[23] J. E. Dennis and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Review, 19 (1977), pp. 46–89.

[24] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization*, Prentice-Hall, 1983.

[25] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, no. 16 in Classics in Applied Mathematics, SIAM, Philadelphia, Penn., 1996.

[26] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK Users' Guide*, SIAM Publications, 1979.

[27] I. S. Duff, A. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Clarendon Press, Oxford UK, 1986.

[28] A. L. Dulmage and N. Mendelsohn, *Coverings of bipartite graphs*, Canadian Journal of Mathematics, 10 (1958), pp. 517–534.

[29] ——, *A structure theory of bipartite graphs of finite exterior dimension*, Trans. Royal Society Can. Sec. III, 53 (1959), pp. 1–13.

[30] ——, *Two algorithms for bipartite graphs*, J. Soc. Ind. Appl. Math., 11 (1963), pp. 183–194.

[31] D. J. Evans, *Preconditioning Methods, Theory and Applications*, Gordon and breach Science Publishers, 1983.

[32] R. Fletcher, *A new approach to variable metric algorithms*, Computer Journal, 13 (1970), pp. 317–322.

[33] ——, *Practical Methods of Optimization*, John Wiley, New York, 2nd ed., 1987.

[34] R. Fletcher and M. J. D. Powell, *A rapidly convergent descent method for minimization*, Computer Journal, 6 (1963), pp. 163–168.

[35] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Computer Journal, 6 (1964), pp. 163–168.

[36] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 186–197.

[37] J. A. George and M. T. Heath, *Solution of sparse least squares problems using Givens rotations*, SIAM J. Numer. Anal., 34 (1980), pp. 69–83.

[38] J. A. George and J. W. H. Liu, *A fast implementation of the minimum degree algorithm using quotient graphs*, ACM Trans. on Math. Software, 3 (1980), pp. 337–358.

[39] J. A. George and J. W.-H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, 1981.

[40] J. A. George and J. W. H. Liu, *The evolution of the minimum degree ordering algorithm*, SIAM Review, 31 (1989), pp. 1–19.

[41] J. R. Gilbert, C. Moler, and R. Schreiber, *Sparse matrices in Matlab: Design and implementation*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 333–356.

[42] D. Goldfarb, *A family of variable metric methods derived by variational means*, Mathematics of Computation, 24 (1970), pp. 23–26.

[43] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1989.

[44] L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic Press, Computer Science and Applied Mathematics, 1981.

[45] T. F. Havel, I. D. Kuntz, and G. M. Crippen, *The theory and practice of distance geometry*, Bulletin of Mathemtical Biology, 45(5) (1983), pp. 665–720.

[46] B. A. Hendrickson, *The molecule problem: Determining conformation from pairwise distances*, Cornell University Ph.D thesis, Computer Science, 1991.

[47] ——, *The molecule problem: Exploiting structure in global optimization*, SIAM Journal on Optimization, 5 (1995), pp. 835–857.

[48] R. W. Hockney and J. W. Eastwood, *Computer Simulations Using Particles*, Adam Hilger, Bristol, 1988.

[49] J. Hull, *Options, Futures, and Other Derivatives, 3rd Edition*, Printice-Hall, 1997.

[50] R. Jarrow and S. Turnbull, *Derivative Securities*, South-Western Publishing, 1996.

[51] R. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images*, Kluwer Academic Publishers, 1991.

[52] R. Lagnado and S. Osher, *Reconciling differences*, Risk, 10 (1997), pp. 79–83.

[53] Y. Li and F. Santosa, *A computational algorithm for minimizain total variation in image restoration*, IEEE Transactions on Image Processing, 5, No. 6 (1996), pp. 987–995.

[54] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, tech. rep., Dept. Electrical Engineering and Computer Science, Northwestern University, 1988.

[55] J. W. H. Liu, *Modification of the minimum degree algorithm by multiple elimination*, ACM Trans. on Math. Software, 11 (1985), pp. 141–153.

[56] ——, *The multifrontal method for sparse matrix solution: Theory and practice*, SIAM Review, (1992), pp. 82–109.

[57] E. Marwil, *Convergence results for Schubert's method for solving sparse nonlinear equations*, SIAM J. Numer. Anal., 16 (1979), pp. 588–604.

[58] P. Matstoms, *Sparse QR factorization in MATLAB*, ACM Trans. on Math. Software, 20 (1994), pp. 136–159.

[59] R. Merton, *The theory of rational option pricing*, Bell Journal of Economics and Management Science, 4 (1973), pp. 141–183.

[60] J. J. Moré, *The Levenberg-Marquardt algorithm: Implementation and theory*, in Numerical Analysis, G. Watson, ed., Springer-Verlag, 1977, pp. 105–116.

[61] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, *User guide for minpack-1*, Tech. Rep. ANL-80-74, Argonne National Laboratory, 1980.

[62] J. J. Moré and D. Sorensen, *Computing a trust region step*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 553–572.

[63] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM Journal on Optimization, 1 (1991), pp. 93–113.

[64] J. J. Moré and Z. Wu, *ϵ-optimal solutions to distance geometry problems via global continuation*, Preprint MCS–P520–0595, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., May 1994.

[65] S. G. Nash and J. Nocedal, *A numerical study of the limited memory BFGS method and the truncated Newton method for large scale optimization*, SIAM Journal on Optimization, 1 (1991), pp. 358–372.

[66] G. Peters and J. Wilkinson, *The least-squares problem and pseudo-inverses*, Computer Journal, 13 (1970), pp. 309–316.

[67] E. Polak and G. Ribière, *Note sur la convergence de methodes de directions conjugées*, Rev. Francaise Informat. Recherche Operationelle, 16 (1960), pp. 35–43.

[68] S. B. Pope, *PDF methods for turbulent reactive flows*, Progress in Energy and Combustion Science, 11 (1985), pp. 119–192.

[69] A. Pothen and C.-J. Fan, *Computing the block triangular form of a sparse matrix*, ACM Transactions on Mathematical Software, 16 (1990), pp. 303–324.

[70] M. J. D. Powell, *A new algorithm for unconstrained minimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, 1970, pp. 31–66.

[71] E. Rothberg and B. Hendrickson, *Sparse matrix ordering methods for parallel sparse Cholesky factorization*, Tech. Rep. TR 96-0475, Sandia National Laboratories, 1996.

[72] L. Schubert, *Modification of a quasi-Newton method for nonlinear equations with a sparse jacobian*, Math. Comp., 24 (1970), pp. 27–30.

[73] G. A. Schultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22(1) (1985), pp. 47–67.

[74] D. F. Shanno, *Conditioning of quasi-newton methods for function minimization*, Math. Comp., 24 (1970), pp. 647–657.

[75] D. Sorensen, *Trust region methods for unconstrained optimization*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[76] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[77] P. Wilmott, S. Howison, and J. Dewynne, *The Mathenatics of Financial Derivatives*, Cambridge University Press, 1997.

[78] S. J. Wright, *Modified Cholesky factorizations in interior-point algorithms for linear programming*, Tech. Rep. MCS-P600-0596, Mathematics and Computer Science Division, Argonne National Laboratory, 1996.

[79] M. Yannakakis, *Computing minimum fill-in is NP-complete*, SIAM J. Alg. & Disc. Meth., 1 (1981), pp. 77–79.