1. Search Directions

In this chapter we again focus on the unconstrained optimization problem

$$\mathcal{P} \qquad \min_{x\in\mathbb{R}^n} f(x),$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be twice continuously differentiable, and consider the selection of search directions. The algorithms we consider are descent methods having an iteration scheme of the form

$$x^{k+1} = x^k + t_k d^k \ .$$

Line search methods for choosing a suitable stepsize t_k were the focus of our work in last chapter. We now devote our attention to choices for the search directions d^k.

All of the search directions considered in this chapter can be classified as *Newton-like* since they are all of the form

$$d^k = -H_k \nabla f(x^k)$$

for some matrix H_k. If $H_k = I$ for all k, we recover the method of steepest descent. However, in general we wish to choose H_k to be an approximation to $\nabla^2 f(x^k)^{-1}$ which gives Newton's method for optimization. The significance of Newton's method is that when it converges, it converges very fast, typically doubling the accuracy of the solution at each iteration. In order to understand this behavior, we must first learn a bit about *rate of convergence.*

1.1. **Rate of Convergence.**

In this section we focus on notions of *quotient convergence*, or Q-convergence. There are corresponding notions of *root convergence*, or R-convergence. These notions are derived from the quotient or root test for the convergence of power series. In all instances the discussion given below only refers to Q-convergence.

Let $\{x^\nu\} \subset \mathbb{R}^n$ and $\bar{x} \in \mathbb{R}^n$ be such that $\bar{x}^\nu \to \bar{x}$. We say that $\bar{x}^\nu \to \bar{x}$ at a *linear* rate if

$$\limsup_{\nu\to\infty} \frac{\|x^{\nu+1} - \bar{x}\|}{\|x^\nu - \bar{x}\|} < 1 \ .$$

The convergence is said to be *superlinear* if this limsup is 0.

The convergence is said to be *quadratic* if

$$\limsup_{\nu\to\infty} \frac{\|x^{\nu+1} - \bar{x}\|}{\|x^\nu - \bar{x}\|^2} < \infty \ .$$

For example, given $\gamma \in (0,1)$ the sequence $\{\gamma^\nu\}$ converges linearly to zero, but not superlinearly. The sequence $\{\gamma^{\nu^2}\}$ converges superlinearly to 0, but not quadratically. Finally, the sequence $\{\gamma^{2^\nu}\}$ converges quadratically to zero. Superlinear convergence is much faster than linear convergences, but quadratic convergence is much, much faster than superlinear convergence.

1.2. **Newton's Method for Solving Equations.**

Consider the following problem:

$$\mathcal{E} : \text{ Given } g : \mathbb{R}^n \to \mathbb{R}^n, \text{ find } x \in \mathbb{R}^n \text{ for which } g(x) = 0.$$

A great variety of problems can be posed in this format. It is a problem of great practical importance and various versions of this problem continue to be the focus of much ongoing scientific, mathematical, and numerical research. It is important in the context of optimization because of the first–order necessary conditions for optimality, $\nabla f(x) = 0$. A standard approach to solving optimization problems is to locate all critical points, or, equivalently, the zero set of the gradient. Newton's method for equation solving (or just, Newton's method) is designed for this purpose. When one applies Newton's method to solve for the critical points of a function, it is refered to Newton's method for optimization. Before moving on to optimization though, we first consider the standard Newton's method for equations.

Assume that the function g in $\mathcal{E}$ is continuously differentiable and that we have an approximate solution $x^0 \in \mathbb{R}^n$ to $\mathcal{E}$. We now wish to improve on this approximation. If $\overline{x}$ is a solution to $\mathcal{E}$, then

$$0 = g(\overline{x}) = g(x^0) + g'(x^0)(\overline{x} - x^0) + o\|\overline{x} - x^0\|.$$

Thus, if x^0 is "close" to $\overline{x}$, it is reasonable to suppose that the solution to the linearized system

$$0 = g(x^0) + g'(x^0)(x - x^0) \tag{1.1}$$

is even closer. This proceedure is known as Newton's method for finding the roots of the equation $g(x) = 0$. It has one obvious pitfall. Equation (1.1) may not be consistent. That is, there may not exist an x solving (1.1). In general, the set of solutions to (1.1) is either

(1) the empty set,
(2) an infinite set, or
(3) a single point.

For the sake of the present argument, we assume that (3) holds, i.e. $g'(x^0)^{-1}$ exists. Under this assumption (1.1) defines the iteration scheme,

$$x^{k+1} := x^k - [g'(x^i)]^{-1} g(x^k), \tag{1.2}$$

called the Newton iteration. The associated direction

$$d^k := -[g'(x^k)]^{-1} g(x^k). \tag{1.3}$$

is called the Newton direction. We analyze the convergence behavior of this scheme under the additional assumption that only an approximation to $g'(x^k)^{-1}$ is available. We denote this approximation by J_k. The resulting iteration scheme is

$$x^{k+1} := x^k - J_k g(x^k). \tag{1.4}$$

Methods of this type are called *Newton-Like methods.*

Theorem 1.1. *Let $g : \mathbb{R}^n \to \mathbb{R}^n$ be differentiable, $x^0 \in \mathbb{R}^n$, and $J_0 \in \mathbb{R}^{n \times n}$. Suppose that there exists $\bar{x}$, $x_0 \in \mathbb{R}^n$, and $\epsilon > 0$ with $\|x_0 - \bar{x}\| < \epsilon$ such that*

(1) *$g(\overline{x}) = 0$,*
(2) *$g'(x)^{-1}$ exists for $x \in B(\overline{x}; \epsilon) := \{x \in \mathbb{R}^n : \|x - \overline{x}\| < \epsilon\}$ with*

$$\sup\{\|g'(x)^{-1}\| : x \in B(\overline{x}; \epsilon)] \le M_1$$

(3) *g' is Lipschitz continuous on $c\ell B(\overline{x}; \epsilon)$ with Lipschitz constant L, and*
(4) *$\theta_0 := \frac{LM_1}{2}\|x^0 - \overline{x}\| + M_0 K < 1$ where $K \ge \|(g'(x^0)^{-1} - J_0)y^0\|$, $y^0 := g(x^0)/\|g(x^0)\|$, and $M_0 = \max\{\|g'(x)\| : x \in B(\overline{x}; \epsilon)\}$.*

Further suppose that iteration (1.4) is initiated at x^0 where the J_k's are chosen to satisfy one of the following conditions;

(i) $\|(g'(x^k)^{-1} - J_k)y^k\| \leq K$,
(ii) $\|(g'(x^k)^{-1} - J_k)y^k\| \leq \theta_1^k K$ *for some* $\theta_1 \in (0,1)$,
(iii) $\|(g'(x^k)^{-1} - J_k)y^k\| \leq \min\{M_2\|x^k - x^{k-1}\|, K\}$, *for some* $M_2 > 0$, *or*
(iv) $\|(g'(x^k)^{-1} - J_k)y^k\| \leq \min\{M_2\|g(x^k)\|, K\}$, *for some* $M_3 > 0$,

where for each $k = 1, 2, \ldots,$ $y^k := g(x^k)/\|g(x^k)\|$.

These hypotheses on the accuracy of the approximations J_k *yield the following conclusions about the rate of convergence of the iterates* x^k.

(a) *If (i) holds, then* $x^k \to \overline{x}$ *linearly.*
(b) *If (ii) holds, then* $x^k \to \overline{x}$ *superlinearly.*
(c) *If (iii) holds, then* $x^k \to \overline{x}$ *two step quadratically.*
(d) *If (iv) holds, then* $x^k \to \overline{x}$ *quadratically.*

Proof. We begin by establishing the basic inequalities

$$\|x^{k+1} - \overline{x}\| \leq \frac{LM_1}{2}\|x^k - \overline{x}\|^2 + \|(g'(x^k)^{-1} - J_k)g(x^k)\|, \tag{1.5}$$

and

$$\|x^{k+1} - \overline{x}\| \leq \theta_0\|x^k - \overline{x}\| \tag{1.6}$$

and the inclusion

$$x^{k+1} \in B(\bar{x}; \epsilon) \tag{1.7}$$

by induction on k. For $k = 0$ we have

$$\begin{aligned} x^1 - \overline{x} &= x^0 - \overline{x} - g'(x^0)^{-1}g(x^0) + [g'(x^0)^{-1} - J_0]g(x^0) \\ &= g'(x^0)^{-1}[g(\overline{x}) - (g(x^0) + g'(x^0)(\overline{x} - x^0))] \\ &\qquad + [g'(x^0)^{-1} - J_0]g(x^0), \end{aligned}$$

since $g'(x^0)^{-1}$ exists by the hypotheses. Consequently, the hypothese (1)–(4) plus the quadratic bound lemma imply that

$$\begin{aligned} \|x^{k+1} - \overline{x}\| &\leq \|g'(x^0)^{-1}\|\|g(\overline{x}) - (g(x^0) + g'(x^0)(\overline{x} - x^0))\| \\ &\qquad + \|(g'(x^0)^{-1} - J_0)g(x^0)\| \\ &\leq \frac{M_1L}{2}\|x^0 - \overline{x}\|^2 + K\|g(x^0) - g(\overline{x})\| \\ &\leq \frac{M_1L}{2}\|x^0 - \overline{x}\|^2 + M_0K\|x^0 - \overline{x}\| \\ &\leq \theta_0\|x^0 - \overline{x}\| < \epsilon, \end{aligned}$$

whereby (1.5) – (1.6) are established for $k = 0$.

Next suppose that (1.5) – (1.6) hold for $k = 0, 1, \ldots, s - 1$. We show that (1.5) – (1.6) hold at $k = s$. Since $x^s \in B(\overline{x}, \epsilon)$, hypotheses (2)–(4) hold at x^s, one can proceed exactly as in the case $k = 0$ to obtain (1.5). Now if any one of (i)–(iv) holds, then (i) holds. Thus, by

(1.5), we find that

$$\begin{aligned}\|x^{s+1} - \overline{x}\| &\leq \tfrac{M_1 L}{2}\|x^s - \overline{x}\|^2 + \|(g'(x^s)^{-1} - J_s)g(x^s)\| \\ &\leq [\tfrac{M_1 L}{2}\theta_0^s\|x^0 - \overline{x}\| + M_0 K]\|x^s - \overline{x}\| \\ &\leq [\tfrac{M_1 L}{2}\|x^0 - \overline{x}\| + M_0 K]\|x^s - \overline{x}\| \\ &= \theta_0\|x^s - \overline{x}\|.\end{aligned}$$

Hence $\|x^{s+1} - \overline{x}\| \leq \theta_0\|x^s - \overline{x}\| \leq \theta_0\epsilon < \epsilon$ and so $x^{s+1} \in B(\overline{x}, \epsilon)$. We now proceed to establish (a)–(d).

(a) This clearly holds since the induction above established that

$$\|x^{k+1} - \overline{x}\| \leq \theta_0\|x^k - \overline{x}\|.$$

(b) From (1.5), we have

$$\begin{aligned}\|x^{k+1} - \overline{x}\| &\leq \frac{LM_1}{2}\|x^k - \overline{x}\|^2 + \|(g'(x^k)^{-1} - J_k)g(x^k)\| \\ &\leq \frac{LM_1}{2}\|x^k - \overline{x}\|^2 + \theta_1^k K\|g(x^k)\| \\ &\leq [\frac{LM_1}{2}\theta_0^k\|x^0 - \overline{x}\| + \theta_1^k M_0 K]\|x^k - \bar{x}\|\end{aligned}$$

Hence $x^k \to \overline{x}$ superlinearly.

(c) From (1.5) and the fact that $x^k \to \bar{x}$, we eventually have

$$\begin{aligned}\|x^{k+1} - \overline{x}\| &\leq \frac{LM_1}{2}\|x^k - \overline{x}\|^2 + \|(g'(x^k)^{-1} - J_k)g(x^k)\| \\ &\leq \frac{LM_1}{2}\|x^k - \overline{x}\|^2 + M_2\|x^k - x^{k-1}\|\|g(x^k)\| \\ &\leq [\frac{LM_1}{2}\|x^k - \overline{x}\| + M_0 M_2[\|x^{k-1} - \bar{x}\| + \|x^k - \bar{x}\|]]\|x^k - \bar{x}\| \\ &\leq [\frac{LM_1}{2}\theta_0\|x^{k-1} - \overline{x}\| + M_0 M_2(1 + \theta_0)\|x^{k-1} - \overline{x}\|] \\ &\qquad \times\theta_0\|x^{k-1} - \overline{x}\| \\ &= [\frac{LM_1}{2}\theta_0 + M_0 M_2(1 + \theta_0)]\theta_0\|x^{k-1} - \overline{x}\|^2.\end{aligned}$$

Hence $x^k \to \overline{x}$ two step quadratically.

(d) Again by (1.5) and the fact that $x^k \to \bar{x}$, we eventually have

$$\begin{aligned} \|x^{k+1} - \bar{x}\| &\leq \frac{LM_1}{2}\|x^k - \bar{x}\|^2 + \|(g'(x^k)^{-1} - J_k)g(x^k)\| \\ &\leq \frac{LM_1}{2}\|x^k - \bar{x}\|^2 + M_2\|g(x^k)\|^2 \\ &\leq [\frac{LM_1}{2} + M_2M_0^2]\|x^k - \bar{x}\|^2 \ . \end{aligned}$$

□

Note that the conditions required for the approximations to the Jacobian matrices $g'(x^k)^{-1}$ given in (i)–(ii) do not imply that $J_k \to g'(\bar{x})^{-1}$. The stronger conditions

$(i)'$ $\|g'(x^k)^{-1} - J_k\| \leq \|g'(x^0)^{-1} - J_0\|$,
$(ii)'$ $\|g'(x^{k+1})^{-1} - J_{k+1}\| \leq \theta_1\|g'(x^k)^{-1} - J_k\|$ for some $\theta_1 \in (0,1)$,
$(iii)'$ $\|g'(x^k)^{-1} - J_k\| \leq \min\{M_2\|x^{k+1} - x^k\|, \|g'(x^0)^{-1} - J_0\|\}$ for some $M_2 > 0$, or
$(iv)'$ $g'(x^k)^{-1} = J_k$,

which imply the conditions (i) through (iv) of Theorem 1.1 respectively, all imply the convergence of the inverse Jacobian approximates to $g'(\bar{x})^{-1}$. Clearly the conditions $(i)'$–$(iv)'$ are not as desirable since they require a great deal more expense and care in the construction of the inverse Jacobian approximates.

1.3. **Newton's Method for Minimization.** In this section we translate the results of previous section in the context of minimization. Here the underlying problem is

$$\mathcal{P} \qquad \min_{x \in \mathbb{R}^n} f(x) \ .$$

The Newton-like iterations are of the form

$$x^{k+1} = x^k - H_k\nabla f(x^k),$$

where H_k is an approximation to the inverse of the Hessian matrix $\nabla^2 f(x^k)$.

Theorem 1.2. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be twice differentiable, $x^0 \in \mathbb{R}^n$, and $H_0 \in \mathbb{R}^{n\times n}$. Suppose that*

(1) *there exists $\overline{x} \in \mathbb{R}^n$ and $\epsilon > \|x^0 - \bar{x}\|$ such that $f(\overline{x}) \leq f(x)$ whenever $\|x - \bar{x}\| \leq \epsilon$,*
(2) *there is a $\delta > 0$ such that $\delta\|z\|_2^2 \leq z^T\nabla^2 f(x)z$ for all $x \in B(\overline{x}, \epsilon)$,*
(3) *$\nabla^2 f$ is Lipschitz continuous on $clB(\overline{x}; \epsilon)$ with Lipschitz constant L, and*
(4) *$\theta_0 := \frac{L}{2\delta}\|x^0 - \overline{x}\| + M_0K < 1$ where $M_0 > 0$ satisfies $z^T\nabla^2 f(x)z \leq M_0\|z\|_2^2$ for all $x \in B(\overline{x}, \epsilon)$ and $K \geq \|(\nabla^2 f(x^0)^{-1} - H_0)y^0\|$ with $y^0 = \nabla f(x^0)/norm\nabla f(x^0)$.*

Further, suppose that the iteration

$$x^{k+1} := x^k - H_k\nabla f(x^k) \tag{1.8}$$

is initiated at x^0 where the H_k's are chosen to satisfy one of the following conditions:

(i) $\|(\nabla^2 f(x^k)^{-1} - H_k)y^k\| \leq K$,
(ii) $\|(\nabla^2 f(x^k)^{-1} - H_k)y^k\| \leq \theta_1^k K$ *for some $\theta_1 \in (0,1)$,*
(iii) $\|(\nabla^2 f(x^k)^{-1} - H_k)y^k\| \leq \min\{M_2\|x^k - x^{k-1}\|, K\}$, *for some $M_2 > 0$, or*
(iv) $\|(\nabla^2 f(x^k)^{-1} - H_k)y^k\| \leq \min\{M_2\|\nabla f(x^k)\|, K\}$, *for some $M_3 > 0$,*

where for each $k = 1, 2, \ldots$ $y^k := \nabla f(x^k)/\|\nabla f(x^k)\|$.

These hypotheses on the accuracy of the approximations H_k yield the following conclusions about the rate of convergence of the iterates x^k.

(a) *If (i) holds, then $x^k \to \overline{x}$ linearly.*
(b) *If (ii) holds, then $x^k \to \overline{x}$ superlinearly.*
(c) *If (iii) holds, then $x^\epsilon \to \overline{x}$ two step quadratically.*
(d) *If (iv) holds, then $x^k \to \overline{k}$ quadradically.*

In order to more fully understand the convergence behavior described in the above result a careful study of the role of the controling parameters L, M_0, and M_1 needs to be made. Although we do not attempt this study, we do make a few observations. First observe that since L is a Lipschitz constant for $\nabla^2 f$ it represents a bound on the third–order behavior of f. Thus the assumptions for convergence make implicit demands on the third derivative. Next, the constant δ in the context of minimization represents a local uniform lower bound on the eigenvalues of $\nabla^2 f$. That is, f behaves locally as if it were a *strongly convex function* (see exercises) with modulus δ. Finally, M_0 can be interpreted as a local Lipschitz constant for ∇f and only plays a role when $\nabla^2 f$ is approximated inexactly by H_k's.

We now consider the performance differences between the method of steepest descent and Newton's method on a simple one dimensional problem. For this we consider the function $f(x) = x^2 + e^x$. Clearly, f is a strongly convex function with

$$\begin{array}{rcl} f(x) & = & x^2 + e^x \\ f'(x) & = & 2x + e^x \\ f''(x) & = & 2 + e^x \ > \ 2 \\ f'''(x) & = & e^x. \end{array}$$

If we apply the steepest descent algorithm with backtracking ($\gamma = \alpha,\ c = 0.01$) initiated at $x^0 = 1$, we get the following table

k	x^k	$f(x^k)$	$f'(x^k)$	s
0	1	.37182818	4.7182818	0
1	0	1	1	0
2	$-.5$	.8565307	-0.3934693	1
3	$-.25$	.8413008	0.2788008	2
4	$-.375$	.8279143	$-.0627107$	3
5	$-.34075$	.8273473	.0297367	5
6	$-.356375$	.8272131	$-.01254$	6
7	$-.3485625$	.8271976	.0085768	7
8	$-.3524688$	.8271848	$-.001987$	8
9	$-.3514922$	.8271841	.0006528	10
10	$-.3517364$	.827184	$-.0000072$	12

Let us now apply Newton's method from the same starting point taking a unit step at each iteration. This time we get

x	$f'(x)$
1	4.7182818
0	1
$-1/3$	.0498646
$-.3516893$	.00012
$-.3517337$	.00000000064

and one more iteration give $|f'(x^5)| \leq 10^{-20}$. This is a stunning improvement in performance and shows why one always uses Newton's method (or an approximation to it) whenever possible.

Our next objective is to develop numerically viable methods for approximating Jacobians and Hessians in Newton-like methods. We begin with a brief excursion into numerical linear algebra.

1.4. **Numerical Linear Algebra.**

1.4.1. *The LU Factorization.*

Recall from linear algebra that Gaussian elimination is a method for solving linear systems of the form

$$Ax = b,$$

where $A \in \mathbb{R}^{m\times n}$ and $b\mathrm{Ran}(A)$. In this method one first forms the augmented system

$$[A\,|b]$$

and then uses the three elementary row operations to put this system into row echelon form (or upper triangular form). A solution x is then obtained by back substitution, or back solving, starting with the component x_n. We now show how the process of bringing a matrix to upper triangular form can be performed by left matrix multiplication.

The key step in Gaussian elimination is to transform a vector of the form

$$\begin{bmatrix} a \\ \alpha \\ b \end{bmatrix},$$

where $a \in \mathbb{R}^k$, $0 \neq \alpha \in \mathbb{R}$, and $b \in \mathbb{R}^{n-k-1}$, into one of the form

$$\begin{bmatrix} a \\ \alpha \\ 0 \end{bmatrix}.$$

This can be accomplished by left matrix multiplication as follows:

$$\begin{bmatrix} I_{k\times k} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\alpha^{-1}b & I_{(n-k-1)\times(n-k-1)} \end{bmatrix} \begin{bmatrix} a \\ \alpha \\ b \end{bmatrix} = \begin{bmatrix} a \\ \alpha \\ 0 \end{bmatrix}.$$

The matrix

$$\begin{bmatrix} I_{k\times k} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\alpha^{-1}b & I_{(n-k-1)\times(n-k-1)} \end{bmatrix}$$

is called a Gaussian elimination matrix. This matrix is invertible with inverse

$$\begin{bmatrix} I_{k\times k} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \alpha^{-1}b & I_{(n-k-1)\times(n-k-1)} \end{bmatrix}.$$

We now use this basic idea to show how a matrix can be put into upper triangular form.

Suppose

$$A = \begin{bmatrix} a_1 & v_1^T \\ u_1 & \widetilde{A}_1 \end{bmatrix} \in \mathbb{C}^{n\times m},$$

with $0 \neq a_1 \in \mathbb{C}$, $u_1 \in \mathbb{C}^{m-1}$, $v_1 \in \mathbb{C}^{n-1}$, and $\widetilde{A}_1 \in \mathbb{C}^{(m-1)\times(n-1)}$. Then using the first row to zero out u_1 amounts to left multiplication of the matrix A by the matrix

$$\begin{bmatrix} 1 & 0 \\ -\frac{u_1}{a_1} & I \end{bmatrix}$$

to get

$$(*) \qquad \begin{bmatrix} 1 & 0 \\ -\frac{u_1}{a_1} & I \end{bmatrix} \begin{bmatrix} a_1 & v_1^T \\ u_1 & \widetilde{A}_1 \end{bmatrix} \in \mathbb{C}^{n\times m} = \begin{bmatrix} a_1 & v_1^T \\ 0 & A_1 \end{bmatrix},$$

where

$$A_1 = \widetilde{A}_1 - u_1 v_1^T / a_1 .$$

Define

$$L_1 = \begin{bmatrix} 1 & 0 \\ \frac{u_1}{a_1} & I \end{bmatrix} \in \mathbb{C}^{m\times m} \quad \text{and} \quad U_1 = \begin{bmatrix} a_1 & v_1^T \\ 0 & A_1 \end{bmatrix} \in \mathbb{C}^{m\times n} .$$

and observe that

$$L_1^{-1} = \begin{bmatrix} 1 & 0 \\ -\frac{u_1}{a_1} & I \end{bmatrix} .$$

Hence (*) becomes

$$L_1^{-1}A = U_1, \text{ or equivalently, } \quad A = L_1 U_1 .$$

Note that L_1 is *unit* lower triangular (ones on the mail diagonal) and U_1 is block upper-triangular with one 1×1 block and one $(m-1)\times(n-1)$ block on the block diagonal. The multipliers are usually denoted

$$u/a = [\mu_{21},\ \mu_{31},\ \ldots,\ \mu_{m1}]^T .$$

If the $(1,1)$ entry of A_1 is not 0, we can apply the same procedure to A_1: if

$$A_1 = \begin{bmatrix} a_2 & v_2^T \\ u_2 & \widetilde{A}_2 \end{bmatrix} \in \mathbb{C}^{(m-1)\times(n-1)}$$

with $a_2 \neq 0$, letting

$$\widetilde{L}_2 = \begin{bmatrix} I & 0 \\ \frac{u_2}{a_2} & I \end{bmatrix} \in \mathbb{C}^{(m-1)\times(m-1)},$$

and forming

$$\widetilde{L}_2^{-1}A_1 = \begin{bmatrix} 1 & 0 \\ -\frac{u_1}{a_2} & I \end{bmatrix}\begin{bmatrix} a_2 & v_2^T \\ u_2 & \widetilde{A}_1 \end{bmatrix} = \begin{bmatrix} a_2 & v_2^T \\ 0 & A_2 \end{bmatrix} \equiv \widetilde{U}_2 \in \mathbb{C}^{(m-1)\times(n-1)},$$

where $A_2 \in \mathbb{C}^{(m-2)\times(n-2)}$. This process amounts to using the second row to zero out elements of the second column below the diagonal. Setting

$$L_2 = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{L}_2 \end{bmatrix} \quad \text{and} \quad U_2 = \begin{bmatrix} a & v^T \\ 0 & \widetilde{U}_2 \end{bmatrix},$$

we have

$$L_2^{-1}L_1^{-1}A = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{L}_2^{-1} \end{bmatrix}\begin{bmatrix} a & v^T \\ 0 & A_1 \end{bmatrix} = U_2,$$

or equivalently,

$$A = L_2 L_1 U_2.$$

Here U_2 is block upper triangular with two 1×1 blocks and one $(m-2)\times(n-2)$ block on the diagonal, and again L_2 is unit lower triangular. We can continue in this fashion at most $\tilde{m}-1$ times, where

$$\tilde{m} = \min\{m, n\}.$$

If we *can* proceed $\tilde{m}-1$ times, then

$$L_{\tilde{m}-1}^{-1}\cdots L_2^{-1}L_1^{-1}A = U_{\tilde{m}-1} = U$$

is upper triangular provided that along the way that the $(1,1)$ entries of

$$A,\ A_1,\ A_2,\ \ldots,\ A_{\tilde{m}-2}$$

are nonzero so the process can continue. Define

$$L = (L_{\tilde{m}-1}^{-1}\cdots L_1^{-1})^{-1} = L_1 L_2 \cdots L_{\tilde{m}-1}.$$

The matrix L is square unit lower triangular, and so is invertable. Moreover, $A = LU$, where the matrix U is the so called *row echelon form* of A. In general, a matrix $T \in \mathbb{C}^{m\times n}$ is said to be in row echelon form if for each $i = 1, \ldots, m-1$ the first non-zero entry in the $(i+1)^{\text{st}}$ row lies to the right of the first non-zero row in the i^{th} row.

Let us now suppose that $m = n$ and $A \in \mathbb{C}^{n\times n}$ is invertible. Writing $A = LU$ as a product of a unit lower triangular matrix $L \in \mathbb{C}^{n\times n}$ (necessarily invertible) and an upper triangular matrix $U \in \mathbb{C}^{n\times n}$ (also nessecarily invertible in this case) is called the *LU factorization* of A.

Remarks

(1) If $A \in \mathbb{C}^{n\times n}$ is invertible and has an LU factorization, it is unique.

(2) One can show that $A \in \mathbb{C}^{n\times n}$ has an LU factorization iff for $1 \le j \le n$, the upper left $j \times j$ principal submatrix

$$\begin{bmatrix} a_{11} & \cdots & a_{ij} \\ \vdots & & \\ a_{j1} & \cdots & a_{jj} \end{bmatrix}$$

is invertible.

(3) Not every invertible $A \in \mathbb{C}^{n\times n}$ has an LU-factorization.

Example: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Typically, one must permute the rows of A to move nonzero entries to the appropriate spot for the elimination to proceed. Recall that a permutation matrix $P \in \mathbb{C}^{n\times n}$ is the identity I with its rows (or columns) permuted: so

$$P \in \mathbb{R}^{n\times n} \text{ is orthogonal, and } P^{-1} = P^T.$$

Permuting the rows of A amounts to left multiplication by a permutation matrix P^T; then $P^T A$ has an LU factorization, so $A = PLU$ (called the PLU factorization of A).

(4) Fact: Every invertible $A \in \mathbb{C}^{n\times n}$ has a (not necessarily unique) PLU factorization.

(5) The LU factorization can be used to solve linear systems $Ax = b$ (where $A = LU \in \mathbb{C}^{n\times n}$ is invertible). The system $Ly = b$ can be solved by forward substitution (1$^{\text{st}}$ equation gives x_1, etc.), and $Ux = y$ can be solved by back-substitution (n^{th} equation gives x_n, etc.), giving the solution to? $Ax = LUx = b$.

Example: We now use the procedure outlined above to compute the LU factorization of the matrix

$$A = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 4 & 2 \\ -1 & 1 & 3 \end{bmatrix}.$$

$$\begin{aligned} L_1^{-1}A &= \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 2 & 4 & 2 \\ -1 & 1 & 3 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 2 & 5 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} L_2^{-1}L_1^{-1}A &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 2 & 5 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 8 \end{bmatrix} \end{aligned}$$

We now have

$$U = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & -3 \\ 0 & 0 & 8 \end{bmatrix},$$

and

$$L = L_1L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix}.$$

1.4.2. *The Cholesky Factorization.* We now consider the application of the LU factorization to a symmetric positive definite matrix, but with a twist. Suppose the $n \times n$ matrix H is symmetric and we have performed the first step in the procedure for computing the LU factorization of H so that

$$L_1^{-1}H = U_1 \ .$$

Clearly, U_1 is no-longer symmetric (assuming L_1 is not the identity matrix). To recover symmetry we could multiply U_1 on the right by the upper triangular matrix L_1^{-T} so that

$$L_1^{-1}HL_1^{-T} = U_1L_1^{-T} = H_1 \ .$$

We claim that H_1 necessarily has the form

$$H_1 = \begin{bmatrix} h_{(1,1)} & 0 \\ 0 & \hat{H}_1 \end{bmatrix},$$

where $h_{(1,1)}$ is the $(1,1)$ element of H and $\hat{H}_1$ is an $(n-1)\times(n-1)$ symmetric matrix. For example, consider the matrix

$$H = \begin{bmatrix} 1 & 2 & -1 \\ 2 & 5 & 1 \\ -1 & 1 & 3 \end{bmatrix}.$$

In this case, we get

$$\begin{aligned} L_1^{-1}HL_1^{-T} &= \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 2 & 5 & 1 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 3 & 2 \end{bmatrix}. \end{aligned}$$

If we now continue this process with the added feature of multiplying on the right by L_j^{-T} as we proceed, we obtain

$$L^{-1}HL^{-T} = D,$$

or equivalently,

$$H = LDL^T,$$

where L is a unit lower triangular matrix and D is a diagonal matrix. Note that the entries on the diagonal of D are not necessarily the eigenvalues of H since the transformation $L^{-1}HL^{-T}$ is not a similarity transformation.

Observe that if it is further assumed that H is positive definite, then the diagonal entries of D are necessarity all positive and the factorization $H = LDL^T$ can always be obtained, i.e. no zero pivots can arise in computing the LU factorization (see exercises).

Let us apply this approach by continuing the computation of the LU factorization for the matrix given above. Thus far we have

$$\begin{aligned} L_1^{-1}HL_1^{-T} &= \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 2 & 5 & 1 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 3 & 2 \end{bmatrix}. \end{aligned}$$

Next

$$\begin{aligned} L_2^{-1}L_1^{-1}HL_1^{-T}L_2^{-T} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 3 \\ 0 & 3 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -7 \end{bmatrix}, \end{aligned}$$

giving the desired factorization

$$H = LDL^T = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -7 \end{bmatrix} \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that this implies that the matrix H is not positive definite.

We make one final comment on the positive definite case. When H is symmetric and positive definite, an LU factorization always exists and we can use it to obtain a factorization of the form $H = LDL^T$, where L is unit lower triangular and D is diagonal with positive diagonal entries. If $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ with each $d_i > 0$, the $D^{1/2} = \mathrm{diag}(\sqrt{d_1}, \sqrt{d_2}, \ldots, \sqrt{d_n})$. Hence we can write

$$H = LDL^T = LD^{1/2}D^{1/2}L^T = \hat{L}\hat{L}^T,$$

where $\hat{L} = LD^{1/2}$ is a non-singular lower triangular matrix. The factorization $H = \hat{L}\hat{L}^T$ where $\hat{L}$ is a non-singular lower triangula matrix is called the *Cholesky factorization* of H. In this regard, the process of computing a Chlesky factorization is an effective means for determining is a symmetric matrix is positive definite.

1.4.3. *The QR Factorization.* Recall the Gram-Schmidt orthogonalization process for a sequence of linearly independent vectors $a_1, \ldots, a_n \in \mathbb{R}^n$. Define $q_1, \ldots, q_n$ inductively, as follows: set

$$p_1 = a_1, \qquad q_1 = p_1/\|p_1\|,$$

$$p_j = a_j - \sum_{i=1}^{j-1} \langle a_j, q_j \rangle \, q_i \qquad \text{for} \quad 2 \le j \le n, \qquad \text{and}$$

$$q_j = p_j/\|p_j\| \; .$$

For $1 \le j \le n$,

$$q_j \in \text{Span}\{a_1, \dots, a_j\},$$

so each $p_j \ne 0$ by the lin. indep. of $\{a_1, \dots, a_n\}$. Thus each q_j is well-defined. We have $\{q_1, \dots, q_n\}$ is an orthonormal basis for $\text{Span}\{a_1, \dots, a_n\}$. Also

$$a_k \in \text{Span}\{q_1, \dots, q_k\} \qquad 1 \le k \le n,$$

so $\{q_1, \dots, q_k\}$ is an orthonormal basis of $\text{Span}\{a_1, \dots, a_k\}$.

Define

$$r_{jj} = \|p_j\| \quad \text{and} \quad r_{ij} = \langle a_j, q_i \rangle \quad \text{for} \quad 1 \le i < j \le n,$$

we have:

$$\begin{aligned} a_1 &= r_{11}\, q_1, \\ a_2 &= r_{12}\, q_1 + r_{22}\, q_2, \\ a_3 &= r_{13}\, q_1 + r_{23}\, q_2 + r_{33}\, q_3, \\ &\vdots \\ a_n &= \sum_{i=1}^{n} r_{in}\, q_i. \end{aligned}$$

Set

$$A = [a_1 \; a_2 \; \dots \; a_n] \, , \quad R = [r_{ij}], \quad \text{and} \quad Q = [q_1 \; q_2 \; \dots \; q_n] \;\; ,$$

where $r_{ij} = 0, \; i > j$. Then

$$A = QR \; ,$$

where Q is unitary and R is upper triangular.

Remarks

(1) If $a_1, a_2, \cdots$ is a linearly independent sequence, apply Gram-Schmidt to obtain an orthonormal sequence $q_1, q_2, \dots$ such that $\{q_1, \dots, q_k\}$ is an orthonormal basis for $\text{Span}\{a_1, \dots, a_k\}$, $k \ge 1$.

(2) If the a_j's are linearly dependent, for some value(s) of k,

$$a_k \in \text{Span}\{a_1, \dots, a_{k-1}\}, \quad \text{so} \quad p_k = 0.$$

The process can be modified by setting $r_{kk} = 0$, not defining a new q_k for this iteration and proceeding. We end up with orthogonal q_j's. Then for $k \ge 1$, the vectors $\{q_1, \dots, q_k\}$ form an orthonormal basis for $\text{Span}\{a_1, \dots, a_{\ell+k}\}$ where ℓ is the number of $r_{jj} = 0$. Again we obtain $A = QR$, but now Q may not be square.

(3) The classical Gram-Schmidt algorithm as described does not behave well computationally. This is due to the accumulation of round-off error. The computed q_j's are not orthogonal: $\langle q_j, q_k \rangle$ is small for $j \neq k$ with j near k, but not so small for $j \ll k$ or $j \gg k$.

An alternate version, "Modified Gram-Schmidt," is equivalent in exact arithmetic, but behaves better numerically. In the following "pseudo-codes," p denotes a temporary storage vector used to accumulate the sums defining the p_j's.

```
Classic Gram-Schmidt
For  $j = 1, \cdots, n$ do
    $p := a_j$
    For $i = 1, \ldots, j-1$ do
        $r_{ij} = \langle a_j, q_i \rangle$
        $p := p - r_{ij} q_i$
    $r_{jj} := \|p\|$
    $q_j := p / r_{jj}$
```

```
Modified Gram-Schmidt
For  $j = 1, \ldots, n$ do
    $p := a_j$
    For $i = 1, \ldots, j-1$ do
        $r_{ij} = \langle p, q_i \rangle$
        $p := p - r_{ij} q_i$
    $r_{jj} = \|p\|$
    $q_j := p / r_{jj}$
```

The only difference is in the computation of r_{ij}: in Modified Gram-Schmidt, we orthogonalize the accumulated partial sum for p_j against each q_i successively.

Theorem 1.3. *Suppose $A \in \mathbb{R}^{m \times n}$ with $m \geq n$. Then*

$$\exists \quad \text{unitary} \quad Q \in \mathbb{R}^{m \times m} \quad \text{upper triangular} \quad R \in \mathbb{R}^{m \times n}$$

for which

$$A = QR.$$

If $\widetilde{Q} \in \mathbb{R}^{m \times n}$ denotes the first n columns of Q and $\widetilde{R} \in \mathbb{R}^{n \times n}$ denotes the first n rows of R, then

$$A = QR = [\widetilde{Q} \; *] \begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix} = \widetilde{Q}\widetilde{R}.$$

Moreover

(a) *We may choose an R to have nonnegative diagonal entries.*

(b) *If A is of full rank, then we can choose R with positive diagonal entries, in which case the condensed factorization $A = \widetilde{Q}\widetilde{R}$ is unique (and thus if $m = n$, the factorization $A = QR$ is unique since then $Q = \widetilde{Q}$ and $R = \widetilde{R}$).*

Proof. If A has full rank, apply the Gram-Schmidt. Define

$$\widetilde{Q} = [q_1, \ldots, q_n] \in \mathbb{R}^{m \times n} \quad \text{and} \quad \widetilde{R} = [r_{ij}] \in \mathbb{R}^{n \times n}$$

as above, so

$$A = \widetilde{Q}\widetilde{R}.$$

Extend $\{q_1, \ldots, q_n\}$ to an orthonormal basis $\{q_1, \ldots, q_m\}$ of $\mathbb{R}^m$, and set

$$Q = [q_1, \ldots, q_m] \quad \text{and} \quad R = \begin{bmatrix} \widetilde{R} \\ 0 \end{bmatrix} \in \mathbb{C}^{m\times n}, \text{ so } A = QR.$$

As $r_{jj} > 0$ in the G-S process, we have (b). Uniqueness follows by induction passing through the G-S process again, noting that at each step we have no choice. □

Remarks

(1) In practice, there are more efficient and better computationally behaved ways of calculating the Q and R factors. The idea is to create zeros below the diagonal (successively in columns $1, 2, \ldots$) as in Gaussian Elimination, except we now use Householder transformations (which are unitary) instead of the unit lower triangular matrices L_j.

(2) A QR factorization is also possible when $m < n$.

$$A = Q[R_1 \; R_2] \; ,$$

where $Q \in \mathbb{C}^{m\times m}$ is unitary and $R_1 \in \mathbb{C}^{m\times m}$ is upper triangular.

Every $A \in \mathbb{R}^{m\times n}$ has a QR-factorization, even when $m < n$. Indeed, if

$$\text{rank}(A) = k,$$

there always exist

$$\begin{array}{rl} Q \in \mathbb{R}^{m\times k} & \text{with orthonormal columns,} \\ R \in \mathbb{R}^{k\times n} & \text{full rank upper triangular,} \end{array}$$

and a permutation matrix $P \in \mathbb{R}^{n\times n}$ such that

$$(*) \qquad AP = QR.$$

Moreover, if A has rank n (so $m \geq n$), then $R \in \mathbb{R}^{n\times n}$ is nonsingular. On the other hand, if $m < n$, then

$$R = [R_1 \; R_2],$$

where $R_1 \in \mathbb{R}^{k\times k}$ is nonsingular. Finally, if $A \in \mathbb{R}^{m\times n}$, then the same facts hold, but now both Q and R can be chosen to be real matrices.

QR-Factorization and Orthogonal Projections

Let $A \in \mathbb{R}^{m\times n}$ have condensed QR-factorization

$$A \;=\; \widetilde{Q}\widetilde{R} \; .$$

Then by construction the columns of $\widetilde{Q}$ form an orthonormal basis for the range of A. Hence $P = \widetilde{Q}\widetilde{Q}^T$ is the orthogonal projector onto the range of A. Similarly, if the condensed QR-factorization of A^T is

$$A^T \;=\; \widetilde{Q}_1\widetilde{R}_1 \; ,$$

then

$$P_1 = \widetilde{Q}_1\widetilde{Q}_1^T$$

is the orthogonal projector onto $\text{Ran}(A^T) = \ker(A)^\perp$, and so

$$I - \widetilde{Q}_1\widetilde{Q}_1^T$$

is the orthogonal projector onto $\ker(A)$.

The QR-factorization can be computed using either Givens rotations or Householder reflections. Although, the approach via rotations is arguably more stable numerically, it is more difficult to describe so we only illustrate the approach using Householder reflections.

QR using Householder Reflections

Given $w \in \mathbb{R}^n$ we can associate the matrix

$$U = I - 2\frac{ww^T}{w^Tw}$$

which reflects $\mathbb{R}^n$ across the hyperplane $\mathrm{Span}\{w\}^\perp$. The matrix U is call the Householder reflection across this hyperplane.

Given a pair of vectors x and y with

$$\|x\|_2 = \|y\|_2, \quad \text{and} \quad x \neq y,$$

there is a Householder reflection such that $y = Ux$:

$$U = I - 2\frac{(x-y)(x-y)^T}{(x-y)^T(x-y)}.$$

Proof.

$$\begin{aligned} Ux &= x - 2(x-y)\frac{\|x\|^2 - y^Tx}{\|x\|^2 - 2y^Tx + \|y\|^2} \\ &= x - 2(x-y)\frac{\|x\|^2 - y^Tx}{2(\|x\|^2 - y^Tx)} \\ &= y \end{aligned}$$

since $\|x\| = \|y\|$. □

QR using Householder Reflections

We now describe the basic *deflation* step in the QR-factorization.

$$A_0 = \begin{bmatrix} \alpha_0 & a_0^T \\ b_0 & A_0 \end{bmatrix}.$$

Set

$$\nu_0 = \left\| \begin{pmatrix} \alpha_0 \\ b_0 \end{pmatrix} \right\|_2.$$

Let H_0 be the Householder transformation that maps

$$\begin{pmatrix}\alpha_0\\ b_0^T\end{pmatrix} \;\mapsto\; \nu_0\; e_1 \quad :$$

$$H_0 = I - 2\frac{ww^T}{w^Tw} \qquad \text{where} \qquad w = \begin{pmatrix}\alpha_0\\ b_0\end{pmatrix} - \nu_0 e_1 = \begin{pmatrix}\alpha_0 - \nu_0\\ b_0\end{pmatrix}.$$

Thus,

$$H_0 A = \begin{bmatrix}\nu_0 & a_1^T\\ 0 & A_1\end{bmatrix}.$$

A problem occurs if $\nu_0 = 0$ or

$$\begin{pmatrix}\alpha_0\\ b_0\end{pmatrix} \;=\; 0\ .$$

In this case, permute the offending column to the right bringing in the column of greatest magnitude. Now repeat with A_1.

If the above method if implemented by always permuting the column of greatest magnitude into the current pivot column, then

$$AP = QR$$

gives a QR-factorization with the diagonal entries of R nonnegative and listed in the order of descending magnitude.

1.5. **Matrix Secant Methods.** Let us return to the problem of finding $\overline{x} \in \mathbb{R}^n$ such that $g(\overline{x}) = 0$ where $g : \mathbb{R}^n \to \mathbb{R}^n$ is C^1. In this section we consider Newton-Like methods of a special type. Recall that in a Newton-Like method the iteration scheme takes the form

$$x^{k+1} := x^k - J_k g(x^k), \tag{1.9}$$

where J_k is meant to approximate the inverse of $g'(x^k)$. In the one dimensional case, a method proposed by the Babylonians 3700 years ago is of particular significance. Today we call it the *secant method*:

$$J_k = \frac{x^{k-1} - x^k}{g(x^{k-1}) - g(x^k)}. \tag{1.10}$$

With this approximation one has

$$g'(x^k)^{-1} - J_k = \frac{g(x^{k-1}) - [g(x^k) + g'(x^k)(x^{k-1} - x^k)]}{g'(x^k)[g(x^{k-1}) - g(x^k)]}.$$

Also, near a point x^* at which $g'(x^*) \neq 0$ there exists an $\alpha > 0$ such that

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

Consequently, by the Quadratic Bound Lemma,

$$\|g'(x^k)^{-1} - J_k\| \le \frac{\frac{L}{2}\|x^{k-1} - x^k\|^2}{\alpha\|g'(x^k)\|\|x^{k-1} - x^k\|} \le K\|x^{k-1} - x^k\|$$

for some constant $K > 0$ whenever x^k and x^{k-1} are sufficiently close to x^*. Therefore, by Theorem 1.1, the secant method is locally two step quadratically convergent to a non–singular solution of the equation $g(x) = 0$. An additional advantage of this approach is that no extra function evaluations are required to obtain the approximation J_k.

1.5.1. *Matrix Secant Methods for Equations.* Unfortunately, the secant approximation (1.10) is meaningless if the dimension n is greater than 1 since division by vectors is undefined. But this can be rectified by simply writing

$$J_k(g(x^{k-1}) - g(x^k)) = x^{k-1} - x^k. \tag{1.11}$$

Equation (1.11) is called the *Quasi-Newton equation* (QNE), or *matrix secant equation* (MSE), at x^k and it determines J_k along an n dimensional manifold in $\mathbb{R}^{n\times n}$. Equation (1.11) is not enough to uniquely determine J_k since (1.11) is n linear equations in n^2 unknowns. To nail down a specific J_k further conditions on the update J_k must be given. In order to determine sensible conditions, let us consider an overall iteration scheme based on (1.9). At every iteration we have (x^k, J_k) and compute x^{k+1} by (1.9). Then J_{k+1} is constructed to satisfy (1.11). If $B_k = J_k^{-1}$ is close to $g'(x^k)$ and x^{k+1} is close to x^k, then J_{k+1} should be chosen not only to satisfy (1.11) but also to be as "close" to J_k as possible. In what sense should we mean "close" here? In order to facilitate the computations it is reasonable to mean "algebraically" close in the sense that J_{k+1} (or $B_{k+1} = J_{k+1}^{-1}$) is only a rank 1 modification of J_k (respectively, $B_k = J_k^{-1}$). Since it may happen that $g'(x^k)$ is not invertible, we first see what happens when we use this idea to approximate B_{k+1}. Since we are assuming that B_{k+1} is a rank 1 modification to B_k, there are vectors $u, v \in \mathbb{R}^n$ such that

$$B_{k+1} = B_k + uv^T. \tag{1.12}$$

We now use the matrix secant equation (MSE) to derive conditions on the choice of u and v. In this setting, the MSE becomes

$$B_{k+1}s^k = y^k,$$

where

$$s^k := x^{k+1} - x^k \qquad \text{and} \qquad y^k := g(x^{k-1}) - g(x^k)\ .$$

Multiplying (1.12) by s^k gives

$$y^k = B_{k+1}s^k = B_k s^k + uv^T s^k\ .$$

Hence, if $v^T s^k \neq 0$, we obtain

$$u = \frac{y^k - B_k s^k}{v^T s^k}$$

and

$$B_{k+1} = B_k + \frac{(y^k - B_k s^k)v^T}{v^T s^k}. \tag{1.13}$$

Equation (1.13) determines a whole class of rank one updates that satisfy the MSE where one is allowed to choose $v \in \mathbb{R}^n$ as long as $v^T s^k \neq 0$. If $s^k \neq 0$, then an obvious choice for v

is s^k yielding the update

$$B_{k+1} = B_k = \frac{(y^k - B_k s^k){s^k}^T}{{s^k}^T s^k}. \tag{1.14}$$

This is known as Broyden's update. It turns out that the Broyden update is also analytically close.

Theorem 1.4. *Let $A \in \mathbb{R}^{n\times n}$, s, $y \in \mathbb{R}^n$, $s \neq 0$. Then for any matrix norms $\|\cdot\|$ and $|||\cdot|||$ such that*

$$\|AB\| \leq \|A\|\, |||B|||$$

and

$$|||\frac{vv^T}{v^Tv}||| \leq 1,$$

the solution to

$$\min\{\|B - A\| : Bs = y\} \tag{1.15}$$

is

$$A_+ = A + \frac{(y - As)s^T}{s^Ts}. \tag{1.16}$$

In particular, (1.16) solves (1.15) when $\|\cdot\|$ is the ℓ_2 matrix norm, and (1.16) solves (1.15) uniquely when $\|\cdot\|$ is the Frobenius norm.

Proof. Let $B \in \{B \in \mathbb{R}^{n\times n} : Bs = y\}$, then

$$\begin{aligned} \|A_+ - A\| &= \|\frac{(y - As)s^T}{s^Ts}\| = \|(B - A)\frac{ss^T}{s^Ts}\| \\ &\leq \|B - A\|\, |||\frac{ss^T}{s^Ts}||| \leq \|B - A\|. \end{aligned}$$

Note that if $|||\cdot||| = \|\cdot\|_2$, then

$$\begin{aligned} \|\frac{vv^T}{v^Tv}\|_2 &= \sup\{\|\frac{vv^T}{v^Tv}x\|_2 : \|x\|_2 = 1\} \\ &= \sup\{\sqrt{\frac{(v^Tx)^2}{\|v\|^2}} : \|x\|_2 = 1\} \\ &= 1, \end{aligned}$$

so that the conclusion of the result is not vacuous. For uniqueness observe that the Frobenius norm is strictly convex and $\|A \cdot B\|_F \leq \|A\|_F \|B\|_2$. □

Therefore, the Broyden update (1.14) is both algebraically and analytically close to B_k. These properties indicate that it should perform well in practice and indeed it does.

Algorithm: Broyden's Method

Initialization: $x^0 \in \mathbb{R}^n$, $B_0 \in \mathbb{R}^{n\times n}$

Having (x^k, B_k) compute (x^{k+1}, B_{x+1}) as follows:

Solve $B_k s^k = -g(x^k)$ for s^k and set

$$\begin{aligned} x^{k+1} : &= x^k + s^k \\ y^k : &= g(x^k) - g(x^{k+1}) \\ B_{k+1} : &= B_k + \frac{(y^k - B_k s^k){s^k}^T}{{s^k}^T s^k}. \end{aligned}$$

Let us now see if we can compute $J_k = B_k^{-1}$ so that we can write the step computation as $s^k = -J_k g(x^k)$ avoiding the need to solve an equation. For this we employ the following important lemma.

Lemma 1.1. (Sherman-Morrison-Woodbury*) Suppose* $A \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times k}$, $V \in \mathbb{R}^{n\times k}$ *are such that both* A^{-1} *and* $(I + V^T A^{-1} U)^{-1}$ *exist, then*

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1}$$

The above lemma verifies that if $B_k^{-1} = J_k$ exists and ${s^k}^T J_k y^k = {s^k}^T B_k^{-1} y^k \neq 0$, then
(1.17)

$$J_{k+1} = [B_k + \frac{(y^k - B_k s^k){s^k}^T}{{s^k}^T s^k}]^{-1} = B_k^{-1} + \frac{(s^k - B_k^{-1} y^k){s^k}^T B_k^{-1}}{{s^k}^T B_k^{-1} y} = J_k + \frac{(s^k - J_k y^k){s^k}^T J_k}{{s^k}^T J_k y}.$$

In this case, it is possible to directly update the inverses J_k. But this process can become numerically unstable if $|{s^k}^T J_k y^k|$ is small. Therefore, in practise, the Broyden update is usually implemented in *forward* mode described by the algorithm for Broyden's method above.

Although we do not pause to establish the convergence rates here, we do give the following result due to Dennis and Moré (1974).

Theorem 1.5. *Let* $g : \mathbb{R}^n \to \mathbb{R}^n$ *be continuously differentiable in an open convex set* $D \subset \mathbb{R}^n$. *Assume that there exists* $x^* \in \mathbb{R}^n$ *and* $r, \beta > 0$ *such that* $x^* + r\mathbb{B} \subset D$, $g(x^*) = 0$, $g'(x^*)^{-1}$ *exists with* $\|g'(x^*)^{-1}\| \le \beta$, *and* g' *is Lipschitz continuous on* $x^* + r\mathbb{B}$ *with Lipschitz constant* $\gamma > 0$. *Then there exist positive constants* ϵ *and* δ *such that if* $\|x^0 - x^*\|_2 \le \epsilon$ *and* $\|B_0 - g'(x^0)\| \le \delta$, *then the sequence* $\{x^k\}$ *generated by the iteration*

$$\left[\begin{array}{lll} x^{k+1} & := & x^k + s^k \text{ where } s^k \text{ solves } 0 = g(x^k) + B_k s \\ B_{k+1} & := & B_k + \frac{(y^k - B_k s^k)s_k^T}{s_k^T s^k} \text{ where } y^k = g(x^{k+1}) - g(x^k) \end{array} \right.$$

is well-defined with $x^k \to x^*$ *superlinearly.*

1.5.2. *Matrix Secant Methods for Minimization.* Let us now see how we can mimic these matrix secant ideas in the context of optimization where the underlying problem is one of minimization:

$$\mathcal{P} : \underset{x\in\mathbb{R}^n}{\text{minimize}} \ f(x)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is C^2. How can we modify and/or extend the matrix secant methods for equations to the setting of minimization where one wishes to solve the equation $\nabla f(x) = 0$. In this context the MSE (matrix secant equation) becomes

$$s^k = H_{k+1} y^k$$

where $s^k := x^{k+1} - x^k$ and

$$y^k := \nabla f(x^{k+1}) - \nabla f(x^k).$$

In this context the matrix H_k is intended to be an approximation to the inverse of the Hessian matrix $\nabla^2 f(x^k)$. Writing $M_k = H_k^{-1}$, a straightforward application of Broyden's method gives the update

$$M_{k+1} = M_k + \frac{(y^k - M_k s^k){s^k}^T}{{s^k}^T s^k}.$$

However, this is unsatisfactory for two reasons:

(1) Since M_k approximates $\nabla^2 f(x^k)$ it must be symmetric.

(2) Since we are minimizing, then M_k must be positive definite to insure that $s^k = -M_k^{-1}\nabla f(x^k)$ is a direction of descent for f at x^k.

To address problem 1 above, one could return to equation (1.13) an find an update that preserves symmetry. Such an update is uniquely obtained by setting

$$v = (y^k - M_k s^k).$$

This is called the symmetric rank 1 update or SR1. Although this update can on occasion exhibit problems with numerical stability, it has recently received a great deal of renewed interest. The stability problems occur whenever

$$v^T s^k = -\nabla f(x^{k+1})^T M_k^{-1} \nabla f(x^k) \tag{1.18}$$

has small magnitude.

We now approach the question of how to update M_k in a way that addresses both the issue of symmetry and positive definiteness while still using the Broyden updating ideas. Given a symmetric positive definite matrix M and two vectors s and y, our goal is to find a symmetric positive definite matrix $\bar{M}$ such that $\bar{M}s = y$. Since M is symmertic and positive definite, there is a non-singular $n \times n$ matrix L such that $M = LL^T$. Indeed, L can be chosent to be lower triangular Cholesky factor of M. If $\overline{M}$ is also symmetric and positive definite then there is a matrix $J \in \mathbb{R}^{n\times n}$ such that $\overline{M} = JJ^T$. The MSE implies that if

$$J^T s = v \tag{1.19}$$

then

$$Jv = y. \tag{1.20}$$

Let us apply the Broyden update technique to (1.20), J, and L. That is, suppose that

$$J = L + \frac{(y - Lv)v^T}{v^T v}. \tag{1.21}$$

Then by (1.19)

$$v = J^T s = L^T s + \frac{v(y - Lv)^T s}{v^T v}. \tag{1.22}$$

This expression implies that v must have the form

$$v = \alpha L^T s$$

for some $\alpha \in \mathbb{R}$. Substituting this back into (1.22) we get

$$\alpha L^T s = L^T s + \frac{\alpha L^T s (y - \alpha L L^T s)^T s}{\alpha^2 s^T L L^T s}.$$

Hence

$$\alpha^2 = \left[\frac{s^T y}{s^T M s}\right]. \tag{1.23}$$

Consequently, such a matrix J satisfying (1.22) exists only if $s^T y > 0$ in which case

$$J = L + \frac{(y - \alpha M s) s^T L}{\alpha s^T M s},$$

with

$$\alpha = \left[\frac{s^T y}{s^T M s}\right]^{1/2},$$

yielding

$$\overline{M} = M + \frac{y y^T}{y^T s} - \frac{M s s^T M}{s^T M s}. \tag{1.24}$$

Moreover, the Cholesky factorization for $\overline{M}$ can be obtained directly from the matrices J. Specifically, if the QR factorization of J^T is $J^T = QR$, we can set $\overline{L} = R$ yielding

$$\overline{M} = J J^T = R^T Q^T Q R = \overline{L}\overline{L}^T.$$

We have shown that beginning with a symmetric positive definite matrix M_k we can obtain a symmetric and positive definite update M_{k+1} that satisfies the MSE $M_{k+1} s_k = y_k$ by applying the formula (1.24) whenever ${s^k}^T y^k > 0$. We must now address the question of how to choose x^{k+1} so that ${s^k}^T y^k > 0$. Recall that

$$y = y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$$

and

$$s^k = -t_k M_k^{-1} \nabla f(x^k),$$

where t_k is the stepsize. Hence

$$\begin{aligned} {y^k}^T s^k &= \nabla f(x^{k+1})^T s^k - \nabla f(x^k)^T s^k \\ &= t_k (\nabla f(x^k + t_k d_k)^T d^k - \nabla f(x^k)^T d^k)\ , \end{aligned}$$

where $d^k := -M_k^{-1} \nabla f(x^k)$. Now since M_k is positive definite the direction d^k is a descent direction for f at x^k and so $t_k > 0$. Therefore, to insure that ${s^k}^T y^k > 0$ we need only show that $t_k > 0$ can be choosen so that

$$\nabla f(x^k + t_k d^k)^T d^k \geq \beta \nabla f(x^k)^T d^k \tag{1.25}$$

for some $\beta \in (0,1)$ since in this case

$$\nabla f(x^k + t_k d_k)^T d^k - \nabla f(x^k)^T d^k \geq (\beta - 1) \nabla f(x^k)^T d^k > 0.$$

But this precisely the second condition in the weak Wolfe conditions with $\beta = c_2$.

The update (1.24) is called the BFGS (Broyden-Fletcher-Goldfarb-Shanno) update and is currently considered the best available matrix secant type update for minimization. Observe in (1.24) that if both $\overline{M}$ and M are positive definite, then they are both invertible. The Sherman-Morrison-Woodbury formula shows that the inverse is given by

$$\overline{M}^{-1} = M^{-1} + \frac{(s - M^{-1}y)s^T + s(s - M^{-1}y)^T}{y^T s} - \frac{(s - M^{-1}y)^T y s s^T}{(y^T s)^2}. \tag{1.26}$$

Thus the corresponding inverse updating scheme for the BFGS update is

$$H_{k+1} = H_k + \frac{(s^k - H_k y^k){s^k}^T + s^k(s^k - H_k y^k)^T}{{y^k}^T s^k} - \frac{(s^k - H_k y^k)^T y^k s^k {s^k}^T}{({y^k}^T s^k)^2}.$$

This is the form of the update that is most commonly used as it avoids the need to solve the equation $M_k d^k = -\nabla f(x^k)$ for the search direction d^k. Instead, the search direction is obtained directly with a matrix multiply, $d^k = -H_k \nabla f(x^k)$. However, one still needs to be careful to avoid numerical instability. This is typically avoided by being careful to the formation of H_{k+1} from H_k. The advised process is as follows.

BFGS Updating

$$\begin{aligned}
\sigma &:= \sqrt{{s^k}^T y^k} \\
\hat{s}^k &:= s^k/\sigma \\
\hat{y}^k &:= y^k/\sigma \\
H_{k+1} &:= H_k + (\hat{s}^k - H_k\hat{y}^k)(\hat{s}^k)^T + \hat{s}^k(\hat{s}^k - H_k\hat{y}^k)^T - (\hat{s}^k - H_k\hat{y}^k)^T\hat{y}^k\hat{s}^k(\hat{s}^k)^T
\end{aligned}$$

Exercises

(1) Let $\gamma \in (0,1)$.
  (a) Show that the sequence $\{\gamma^\nu\}$ converges linearly to zero, but not superlinearly.
  (b) Show that the sequence $\{\gamma^{\nu^2}\}$ converges superlinearly to 0, but not quadratically.
  (c) Finally, show that the sequence $\{\gamma^{2^\nu}\}$ converges quadratically to zero.

(2) Apply Newton's method to the function $f(x) = x^2 + \sin x$ with initial point $x^0 = 2\pi$.

(3) If $H \in \mathbb{R}^{n\times n}$ is symmetric and positive definite show that there is a diagonal matrix D having positive entries and a unit lower triangular matrix L such that $H = LDL^T$.

(4) Compute the Cholesky factorization of the matrix

$$\begin{bmatrix} 1 & 1 & -1 \\ 1 & 5 & 1 \\ -1 & 1 & 4 \end{bmatrix}.$$

(5) Apply the secant method to the function $f(x) = x^2 + e^x$ with starting points $x_{-1} = 1 - 10^{-3}$ and $x_0 = 1$. How many ierations does it take to obtain a point for which $|f'(x^k)| \leq 10^-8$? Compare this performance with that of Newton's method and the method of steepest descent.

(6) Verify formulae (1.18) and (1.23).

(7) Verify formulae (1.17), (1.24), and (1.26).

(8) Prove the Sherman-Morrison-Woodbury Lemma (Lemma 1.1).

(9) Show that every rank 1 matrix W in $\mathbb{R}^{n\times n}$ can be written in the form $W = uv^T$ for some choice of vectors $u, v \in \mathbb{R}^n$.

(10) Show that every $n \times n$ positive semi-definite matrix M can be written in the form $M = VV^T$ for some $n \times n$ matrix V having the same rank as M. Also, show that V can be chosen to be an $n \times k$ matrix where $k = \text{rank}(M)$.

(11) * The function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be strongly convex if there is a $\delta > 0$ such that

$$f(x + \lambda(y - x)) \leq f(x) + \lambda[f(y) - f(x)] - \frac{\delta}{2}\lambda(1-\lambda)\|y - x\|^2$$

for every $x, y \in \mathbb{R}^n$ and $\lambda \subset [0,1]$. The parameter δ is called the modulus of strong convexity.

Prove the following characterization theorem for strongly convex functions.

Theorem 1.6. *(Characterizations of strongly convex functions) Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable. Then the following statements are equivalent.*
(a) *f is strongly convex with modulus δ.*
(b) *$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\delta}{2}\|y - x\|^2$ for all $x, \in \mathbb{R}^n$.*
(c) *$(\nabla f(x) - \nabla f(y))^T(x - y) \geq \delta\|y - x\|^2$ for all $x, y \in \mathbb{R}^n$.*

If it is further assumed that f is twice continuously differentiable on $\mathbb{R}^n$, then the above conditions are also equivalent to the following statement:

$$\delta\|u\|_2^2 \leq u^T\nabla^2 f(x)u \quad \forall\ u \in \mathbb{R}^n.$$

That is, the spectrum of the Hessian of f is uniformly bounded below by δ on $\mathbb{R}^n$.