Chapter 5

Unconstrained Optimization Algorithmic Methods

In this chapter we discus iterative methods to solve minimization problems. We generate a sequence of points according to a prescribed set of instructions together with a termination criteria.

5.1 One Dimensional Problems

In this section we consider the problem

$$\begin{aligned}&\min f(x)\\&\textbf{subject to}\\&x \in X \subset \mathbb{R}\end{aligned}$$

We consider this minimization problem when the local minimizaer x^* is in $(a, b) \subset X$.

5.1.1 Methods that do not use derivatives

Definition 5.1.1 ***Let $f : S \longrightarrow \mathbb{R}$ where S is a nonempty convex set in $\mathbb{R}^n$. The function f is said to be quasiconvex if for each $x_1, x_2 \in S$,***

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \max\{f(x_1), f(x_2)\}, \quad 0 < \lambda < 1.$$

The function f is said to be quasiconcave if $-f$ is quasiconvex

Definition 5.1.2 ***The function $f : X \longrightarrow \mathbb{R}$ is said to be unimodal on $[a, b] \subset X$ if there exists $x^* \in [a, b]$ at which f attains its minimum and f is nondecreasing on $[x^*, b]$ and nonincreasing on $[a, x^*]$.***
If f is decreasing on $[x^*, b]$ and increasing on $[a, x^*]$, then f is said to be strictly unimodal.

Remark 5.1.1 ***Unimodality is equivalent to quasiconvexity.***

5.1.1.1 Golden Section Method

Let $f : X \subset \mathbb{R} \longrightarrow \mathbb{R}$ be such that f is unimodal on $[a, b] \subset X$.

At iteration $k \geq 1$ assume we have $a_k, \lambda_k, \mu_k, b_k$ such that

$$a \leq a_k < \lambda_k < \mu_k < b_k \leq b$$

$$x^* \in [a_k, b_k].$$

If $f(\lambda_k) \leq f(\mu_k)$ **then**

$$\begin{cases} a_{k+1} = a_k \\ b_{k+1} = \mu_k \\ \textbf{define } \lambda_{k+1} \textbf{ and } \mu_{k+1} \textbf{ so that} \\ a_{k+1} < \lambda_{k+1} < \mu_{k+1} < b_{k+1} \end{cases}$$

else

$$a_{k+1} = \lambda_k$$

$$b_{k+1} = b_k$$

end if

In the Golden Section Method, given a_k **and** b_k **we choose** λ_k **and** μ_k **so that**

$$\begin{aligned} \frac{\lambda_k - a_k}{b_k - a_k} &= r = \frac{3-\sqrt{5}}{2} \approx 0.38197 \\ \frac{\mu_k - a_k}{b_k - a_k} &= 1 - r = \frac{\sqrt{5}-1}{2} \approx 0.61803 \end{aligned}$$

If $f(\lambda_k) \leq f(\mu_k)$ **then we set** $\mu_{k+1} = \lambda_k$ **and compute for** λ_{k+1} **from the formula**

$$\frac{\lambda_{k+1} - a_k}{\mu_k - a_k} = r.$$

Since $\mu_{k+1} = \lambda_k$ **we have**

$$\frac{\mu_{k+1} - a_k}{\mu_k - a_k} = 1 - r$$

$$\begin{aligned} \mu_{k+1} - a_k &= (1-r)(\mu_k - a_k) = (1-r)^2(b_k - a_k) \\ \lambda_k - a_k &= (1-r)^2(b_k - a_k) \\ r(b_k - a_k) &= (1-r)^2(b_k - a_k) \\ r &= (1 - r^2) \end{aligned}$$

Therefore

$$\begin{aligned} r^2 - 3r + 1 &= 0 \\ r &= \frac{3 \pm \sqrt{5}}{2} \\ r &= \frac{3 - \sqrt{5}}{2} \end{aligned}$$

Next, if $f(\lambda_k) > f(\mu_k)$ **then set** $\lambda_{k+1} = \mu_k$ **and solve for** μ_{k+1} **from the formula**

$$\frac{\mu_{k+1} - \lambda_k}{b_k - \lambda_k} = 1 - r.$$

$$\frac{\lambda_{k+1} - \lambda_k}{b_k - \lambda_k} = r$$

$$\begin{aligned} \lambda_{k+1} &= \lambda_k + r\,(b_k - \lambda_k) \\ &= \lambda_k + r\,(b_k - a_k + a_k - \lambda_k) \\ &= \lambda_k + r\,\left(b_k - a_k - (\lambda_k - a_k)\right) \\ &= \lambda_k + r\,\left(b_k - a_k - r\,(b - k - a_k)\right) \\ &= \lambda_k + r\,(b_k - a_k)(1 - r) \end{aligned}$$

Since $\lambda_{k+1} = \mu_k$,

$$\mu_k = \lambda_k + r\,(b_k - a_k)(1-r)$$
$$\mu_k - a_k = \lambda_k - a_k + r\,(b_k - a_k)(1-r)$$
$$(1-r)(b_k - a_k) = r\,(b_k - a_k) + r\,(b_k - a_k)(1-r))$$
$$1 - r = r + r\,(1-r)$$
$$r^2 - 3r + 1 = 0$$
$$r = \frac{3 \pm \sqrt{5}}{2}$$
$$r = \frac{3 - \sqrt{5}}{2}$$

Next, we note how fast we are decreasing the interval in which we are looking for the minimizer x^*.

If $f(\lambda_k) \le f(\mu_k)$ then $a_{k+1} = a_k$ and $b_{k+1} = \mu_k$. Then,

$$\frac{b_{k+1} - a_{k+1}}{b_k - a_k} = \frac{\mu_k - a_k}{b_k - a_k} = 1 - r$$

In the case $f(\lambda_k) > f(\mu_k)$, then $a_{k+1} = \lambda_k$ and $b_{k+1} = b_k$. In this cse,

$$\begin{aligned}
\frac{b_{k+1} - a_{k+1}}{b_k - a_k} &= \frac{b_k - \lambda_k}{b_k - a_k} \\
&= \frac{b_k - a_k + a_k - \lambda_k}{b_k - a_k} \\
&= \frac{b_k - a_k - (\lambda_k - a_k)}{b_k - a_k} \\
&= 1 - r
\end{aligned}$$

Thus, with each iteration we cut the interval of interest by about 62%.

5.1.1.2 The Fibonacci Search Method

This procedure is based on Fibonacci sequence defined by the recursion formula

$$F_{\nu+1} = F_\nu + F_{\nu-1}, \quad \nu = 1, 2, 3, \cdots$$
$$F_0 = F_1 = 1$$

Suppose we wish to minimize the strictly quasiconvex function f on $[a, b]$. Suppose $f(x^*) = \min\{f(x) : a \le x \le b\}$. Suppose we have four points $a \le a_k < \lambda_k < \mu_k < b_k \le b$ and $x^* \in [a_k, b_k]$. Suppose we have planned a total of n functional evaluations and λ_k, μ_k are defined by the formulas

$$\begin{aligned}
\lambda_k &= a_k + \frac{F_{n-k-1}}{F_{n-k+1}}(b_k - a_k), \quad k = 1, \cdots, n-1 \\
\mu_k &= a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k), \quad k = 1, \cdots, n-1
\end{aligned}$$

Either $f(\lambda_k) > f(\mu_k)$ or $f(\lambda_k) < f(\mu_k)$. <u>Case 1</u> suppose $f(\lambda_k) > f(\mu_k)$. In this case we set

$$\begin{aligned}
a_{k+1} &= \lambda_k \\
b_{k+1} &= b_k
\end{aligned}$$

and

$$\begin{aligned}
\frac{b_{k+1}-a_k}{b_k-a_k} &= \frac{b_k-\lambda_k}{b_k-a_k} \\
&= \frac{1}{b_k-a_k}\left(b_k-a_k-\frac{F_{n-k-1}}{F_{n-k+1}}(b_k-a_k)\right) \\
&= 1-\frac{F_{n-k-1}}{F_{n-k+1}} \\
&= \frac{F_{n-k+1}-F_{n-k-1}}{F_{n-k+1}} \\
&= \frac{F_{n-k}}{F_{n-k+1}}
\end{aligned}$$

Thus, we have a reduction of the uncertainty interval by a factor of F_{n-k}/F_{n-k+1}**.**
<u>**Case 2**</u> **Here we suppose that** $f(\lambda_k) < f(\mu_k)$**. Then, we set**

$$\begin{aligned}
a_{k+1} &= a_k \\
b_{k+1} &= \mu_k.
\end{aligned}$$

Now,

$$\begin{aligned}
\frac{b_{k+1}-a_{k+1}}{b_k-a_k} &= \frac{\mu_k-a_k}{b_k-a_k} \\
&= \frac{1}{b_k-a_k}\left(a_k+\frac{F_{n-k}}{F_{n-k+1}}(b_k-a_k)-a_k\right) \\
&= \frac{F_{n-k}}{F_{n-k+1}}.
\end{aligned}$$

Again we have a reduction by a factor of F_{n-k}/F_{n-k+1}**.**

We now go back to Case 1 and show that $\lambda_{k+1} = \mu_k$ **so that we only need to compute for** μ_{k+1} **for the next iteration.**

According to the defining formula for λ_k **and** μ_k **we have**

$$\lambda_{k+1} = a_{k+1} + \frac{F_{n-(k+1)-1}}{F_{n-(k+1)+1}}(b_{k+1}-a_{k+1}).$$

However, in Case 1, $a_{k+1} = \lambda_k$ **and** $b_{k+1} = b_k$**. Thus,**

$$\begin{aligned}
\lambda_{k+1} &= \lambda_k + \frac{F_{n-k-2}}{F_{n-k}}(b_k-\lambda_k) \\
&= \lambda_k + \frac{F_{n-k-2}}{F_{n-k}}\left(b_k-a_k-\frac{F_{n-k-1}}{F_{n-k+1}}(b_k-a_k)\right) \\
&= \lambda_k + (b_k-a_k)\left(\frac{F_{n-k-2}}{F_{n-k}}\right)\left(1-\frac{F_{n-k-1}}{F_{n-k+1}}\right) \\
&= \lambda_k + (b_k-a_k)\frac{F_{n-k-2}}{F_{n-k}}\cdot\frac{F_{n-k+1}-F_{n-k-1}}{F_{n-k+1}} \\
&= \lambda_k + (b_k-a_k)\frac{F_{n-k-2}}{F_{n-k}}\cdot\frac{F_{n-k}}{F_{n-k+1}} \\
&= \lambda_k + (b_k-a_k)\frac{F_{n-k-2}}{F_{n-k+1}} \\
&= \mu_k
\end{aligned}$$

Thus, we already have λ_{k+1} (it is μ_k!). Finally,

$$\begin{aligned}\mu_{k+1} &= a_{k+1} + \frac{F_{n-(k+1)}}{F_{n-(k+1)+1}}\,(b_{k+1}-a_{k+1}) \\ &= a_k + \frac{F_{n-k-1}}{F_{n-k}}\,(\mu_k - a_k)\end{aligned}$$

Next, we go back to Case 2 and show that already we have μ_{k+1} (it is λ_k !) and we do not need to compute to find it. We only need λ_{k+1}. According to the defining formulas for λ_k, μ_k, $k = 1, \cdot, n-1$

$$\mu_{k+1} = a_{k+1} + \frac{F_{n-(k+1)}}{F_{n-(k+1)+1}}\,(b_{k+1}-a_{k+1})$$

In Case 2, $a_{k+1} = a_k$ and $b_{k+1} = \mu_k$. Thus,

$$\begin{aligned}\mu_{k+1} &= a_k + \frac{F_{n-k-1}}{F_{n-k}}\,(\mu_k - a_k) \\ &= a_k + \frac{F_{n-k-1}}{F_{n-k}}\left(a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k) - a_k\right) \\ &= a_k + \frac{F_{n-k-1}}{F_{n-k+1}}\,(b_k - a_k) \\ &= \lambda_k\end{aligned}$$

Finally,

$$\begin{aligned}\lambda_{k+1} &= a_{k+1} + \frac{F_{n-(k+1)-1}}{F_{n-(k+1)+1}}\,(b_{k+1}-a_{k+1}) \\ &= a_k + \frac{F_{n-k-2}}{F_{n-k}}\,(\mu_k - a_k).\end{aligned}$$

How Fibonacci method reduces the interval of unceraininty

We have seen that

$$\frac{b_{k+1}-a_{k+1}}{b_k - a_k} = \frac{F_{n-k}}{F_{n-k+1}}.$$

Thus, if the initial length of uncertainity is $b-a$. Then, the first application of Fibonacci search method reduces it to

$$\frac{F_{n-1}}{F_n}\,(b-a).$$

Then, to

$$\frac{F_{n-2}}{F_{n-1}} \cdot \left[\frac{F_{n-1}}{F_n}\,(b-a)\right]$$

and then to

$$\frac{F_{n-3}}{F_{n-2}} \cdot \left[\frac{F_{n-2}}{F_{n-1}}\left[\frac{F_{n-1}}{F_n}\,(b-a)\right]\right]$$

When $k = n-1$, we get to

$$\frac{F_{n-1}}{F_n} \cdot \frac{F_{n-2}}{F_{n-1}} \cdot \frac{F_{n-3}}{F_{n-2}} \cdot \cdots \cdot \frac{F_2}{F_1} \cdot (b-a) = \frac{1}{F_n} \cdot (b-a).$$

For n large it is known that $\frac{1}{F_n} \approx (0.618)^{n-1}$. Thus, the Fibonacci method and Golden Section method are essentially equally effective.

Pseudocode (Fibonacci)

- **Given** $\varepsilon > 0$**, say** $\varepsilon = 10^{-8}$
- **We seek the minimum of the quasiconvex function** f **in the iterval** (a, b)**.**
- **Choose** n **so that** $\frac{1}{F_n}(b-a) < 10^{-8}$**.**
- **Set** $a_1 = a$**,** $b_1 = b$**.**
- **Set**

$$\lambda_1 = a_1 + \frac{F_{n-2}}{F_n}\,(b_1 - a_1),$$

$$\mu_1 = a_1 + \frac{F_{n-1}}{F_n}\,(b_1 - a_1).$$

- **For** $k = 1$ **to** $n-1$ **do**

  If $f(\lambda_k) < f(\mu_k)$ **then**
  $a_{k+1} = a_k$
  $b_{k+1} = \mu_k$
  $\mu_{k+1} = \lambda_k$
  $\lambda_{k+1} = a_{k+1} + \frac{F_{n-(k+1)-1}}{F_{n-(k+1)+1}}(b_{k+1} - a_k)$
  else
  $a_{k+1} = \lambda_k$
  $b_{k+1} = b_k$
  $\lambda_{k+1} = \mu_k$
  $\mu_{k+1} = a_k + \frac{F_{n-k-1}}{F_{n-k}}(b_{k+1} - a_{k+1})$
  End if

- **End do**

$x^* = \frac{b_n - a_n}{2}$
End

5.1.2 Methods That Use Derivatives

In this section we consider one dimensional minimization problems where the function to be minimized is differentiable.

Definition 5.1.3 ***Let*** S ***be a nonempty subset of*** $\mathbb{R}^n$***. Let*** $f : S \longrightarrow \mathbb{R}$ ***be differentiable on*** S***. The function*** f ***is said to be peudoconvex if for each*** $x_1, x_2 \in S$ ***with*** $\langle \nabla f(x_1), x_2 - x_1 \rangle \geq 0$ ***we have*** $f(x_2) \geq f(x_1)$***, or equivalently, if*** $f(x_2) < f(x_1)$***, then*** $\langle \nabla f(x_1), x_2 - x_1 \rangle < 0$***. The function is said to be pseudoconcave if*** $-f$ ***is pseudoconvex.***

5.1.2.1 The Bisection Method

Let $f : X \subset \mathbb{R} \longrightarrow \mathbb{R}$ **be a differentiable function on the nonempty set** X**. Suppose** f **attains its minimum at** $x^* \in (a, b)$**. Suppose** f **is pseudoconvex on** X**, and** $f^{'}(a) < 0$**,** $f^{'}(b) > 0$**.**

At iteration k **we have two points** a_k, b_k **such that** $a \leq a_k < b_k \leq b_k$**. Suppose** $c_k = (a_k + b_k)/2$**. If** $f^{'}(c_k) \geq 0$**, then** $a_{k+1} = a_k$**, and** $b_{k+1} = c_k$**. If** $f^{'}(c_k) < 0$**, then** $a_{k+1} = c_k$**, and** $b_{k+1} = b_k$**.**

5.1.2.2 Newton's Method

Let $X \subset \mathbb{R}$ be a nonempty open set. Let $f : X \longrightarrow \mathbb{R}$ be twice continuously differentiable function on X. In Newton's method we start with an estimate x_1 of a local minimizer $x^* \in X$, and then generate a sequence $\{x_k\} \subset X$. At iterate x_k we approximate f in a neighborhood $N(x_k)$ of x_k by the following local quadratic approximation.

$$f(x) \approx \psi_k(x) = f(x_k) + f^{'}(x_k)(x - x_k) + \frac{1}{2} f^{''}(x_k)(x - x_k)^2$$

Suppose that $f^{''}(x_k) > 0$. Then, ψ_k has a minimizer at

$$x = x_k - \frac{f^{'}(x_k)}{f^{''}(x_k)}$$

Thus, we generate a sequence according to the formula

$$x_{k+1} = x_k - \frac{f^{'}(x_k)}{f^{''}(x_k)}, \quad k = 1, 2, 3, \cdots$$

This iteration need not converge, or even defined. However, if x_1 is close enough to x^* then under suitable conditions $\{x_k\}$ converges to x^* quadratically.

Theorem 5.1.1 (Kantorovich) ***Let X be a nonempty open subset of $\mathbb{R}$. Let $f : X \longrightarrow \mathbb{R}$ be twice continuously differentiable function. Suppose that there is an open set $D \subset X$ such that there is $x^* \in D$ with $f^{'}(x^*) = 0$, and $|f^{''}(x)| \geq \rho > 0$ for all $x \in D$. Further, there exists $L > 0$ such that for $x_1, x_2 \in D$ we have $|f^{''}(x_2) - f^{''}(x_1)| \leq L\,|x_2 - x_1|$. Then, there exists an $\eta > 0$ such that if $|x_1 - x^*| < \eta$ the sequence $\{x_k\}$ defined by the formula***

$$x_{k+1} = x_k - \frac{f^{'}(x_k)}{f^{''}(x_k)}, \quad k = 1, 2, 3, \cdots$$

is a well-defined sequence in D and converges to x^* quadratically.

Proof **: Choose $\eta > 0$ so that $x_1 \in D$ whenever $|x_1 - x^*| < \eta$.**

$$\begin{aligned}
x_2 - x^* &= x_1 - x^* - \frac{f^{'}(x_1)}{f^{''}(x_1)} \\
&= x_1 - x^* - \frac{f^{'}(x_1) - f^{'}(x^*)}{f^{''}(x_1)} \\
&= \frac{1}{f^{''}(x_1)} \left[f^{'}(x^*) - f^{'}(x_1) - f^{''}(x_1)(x^* - x_1) \right] \\
&= \frac{1}{f^{''}(x_1)} \int_0^1 \left[f^{''}\left(x_1 + t\,(x^* - x_1)\right) - f^{''}(x_1) \right] (x^* - x_1)dt \\
\|x_2 - x^*\| &\leq \frac{1}{|f^{''}(x_1)|} \int_0^1 L\,t\,|x^* - x_1|\,|x^* - x_1|\,dt \\
&\leq \frac{L}{\rho} \left(\int_0^1 t\,dt \right) |x^* - x_1|^2 \\
&= \frac{1}{2\rho}\,L\,|x^* - x_1|^2
\end{aligned}$$

Next, we note that

$$\begin{aligned}
|x_2 - x^*| &\leq \frac{L}{2\rho} |x^* - x_1|^2 \\
&= \frac{L}{2\rho} |x^* - x_1|\,|x^* - x_1| \\
&< \frac{L\eta}{2\rho} |x^* - x_1|
\end{aligned}$$

We choose η so that $L\eta/2\rho < 1$. Then

$$|x_2 - x^*| < \frac{L\eta}{2\rho}|x^* - x_1| < |x^* - x_1| < \eta$$

showing $x_2 \in D$. We complete the proof by induction. □

5.1.2.3 Modified Safeguarded Newton's Iteration

Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be twice continuously differentiable function. We provide below an algorithm to iteratively approximate the minimum value of f in (a,b). We have to provide a safeguard for our iteration in case $f^{''}$ is nonpositive at any point in the iteration. We also need to insure that the iterates are in (a,b). First we present the rational for the algorithm we will present.

We choose $x_1 \in (a,b)$ so that

$$f(x_1) < \min\{f(a), f(b)\}.$$

Set $y_1 = x_1$.

If $f^{''}(x_k) > 0$ Newton's iteration gives the next point x_{k+1} by the formula

$$x_{k+1} = x_k - \frac{f^{'}(x_k)}{f^{''}(x_k)}.$$

By Taylor series approximation we have

$$f(x_{k+1}) \approx f(x_k) + f^{'}(x_k)(x_{k+1} - x_k) + \frac{1}{2}(x_{k+1} - x_k)^2$$

If $x_{k+1} - x_k$ is "too big" the higher order terms contribute positive constant and we may not have $f(x_{k+1}) < f(x_k)$. Thus, we need to scale back how far away we may move from x_k. Set

$$\tilde{x}_{k+1} = x_k + \lambda\,(x_{k+1} - x_k), \quad 0 < \lambda < 1$$

The quantity λ controls how far we move away from x_k. We note that

$$\begin{aligned} f(\tilde{x}_{k+1}) &\approx f(x_k) + f^{'}(x_k)(\tilde{x}_{k+1} - x_k) + \frac{1}{2}f^{''}(x_k)(\tilde{x}_{k+1} - x_k)^2 \\ &= f(x_k) + \lambda\,(x_{k+1} - x_k)f^{'}(x_k) + \frac{1}{2}\lambda^2 f^{''}(x_k)(x_{k+1} - x_k)^2 \end{aligned}$$

Thus, we minimize the effect of the second order term by an approximate choice of λ, $0 < \lambda < 1$. In addition, a sufficiently small $\lambda > 0$ guarantees that $\tilde{x}_{k+1}$ is in (a,b). Then,

$$f^{'}(\tilde{x}_{k+1}) \approx f(x_k) + \lambda(x_{k+1} - x_k)f^{'}(x_k)$$

Now, from Newton's iteration formula

$$x_{k+1} - x_k = -\frac{f^{'}(x_k)}{f^{''}(x_k)}.$$

Thus,

$$f(\tilde{x}_{k+1}) \approx f(x_k) - \lambda\frac{f^{'}(x_k)^2}{f^{''}(x_k)} < f(x_k)$$

Thus, we move from x_k to $\tilde{x}_{k+1}$. Set $y_{k+1} = \tilde{x}_{k+1}$.

If $f^{''}(x_k) \le 0$ we proceed as follows. Set $\delta_k = -1$ if $f^{'}(x_k) \ge 0$ and $\delta_k = 1$ if $f^{'}(x_k) < 0$. Chose i the smallest positive integer so that

$$x_{k+1} = x_k - 2^{-i}f^{'}(x_k) + \delta_k\,2^{-i/2}$$

is in (a,b). Next, set

$$\tilde{x}_{k+1} = x_k + \lambda\,(x_{k+1} - x_k)$$

Note that $\tilde{x}_{k+1} \in (a,b)$ because it is a convex combination of x_k and x_{k+1}.

Then,

$$F(\tilde{x}_{k+1}) \approx f(x_k) + (\tilde{x}_{k+1} - x_k)f^{'}(x_k) + \frac{1}{2}f^{''}(x_k)(\tilde{x}_{k+1} - x_k)$$

If $0 < \lambda < 1$ is sufficiently small

$$\begin{aligned} f(\tilde{x}_{k+1}) &\approx f(x_k) + \lambda\,(x_{k+1} - x_k)f^{'}(x_k) \\ &= f(x_k) - \lambda\,2^{-i}f^{'}(x_k)^2 + \lambda\,\delta_k\,2^{(\frac{-i}{2})}f^{'}(x_k) \\ &= \begin{cases} f(x_k) - \lambda\,2^{-i}f^{'}(x_k)^2 - \lambda\,2^{(\frac{-i}{2})}f^{'}(x_k) \textbf{ if } f^{'}(x_k) \geq 0 \\ f(x_k) - \lambda\,2^{-i}f^{'}(x_k)^2 + \lambda\,2^{(\frac{-i}{2})}f^{'}(x_k) \textbf{ if } f^{'}(x_k) < 0 \end{cases} \\ &< f(x_k) \end{aligned}$$

Thus, we have achieved a decrease, and we set $y_{k+1} = \tilde{x}_{k+1}$.

We now present the safeguarded algorithm.

Begin (algorithm)

Choose $x_1 \in (a,b)$ so that $f(x_1) < \min\{f(a), f(b)\}$.

Set $y_1 = x_1$.

For $k = 1, 2, \cdots$ do

If $f^{''}(x_k) > 0$ then

$x_{k+1} = x_k - \frac{f^{'}(x_k)}{f^{''}(x_k)}$

Choose $0 < \lambda < 1$ so that

$$\begin{cases} x_k + \lambda\,(x_{k+1} - x_k) \in (a,b) \\ f\,(x_k + \lambda\,(x_{k+1} - x_k)) < f(x_k) \end{cases}$$

End Choose

Set $y_{k+1} = x_k + \lambda\,(x_{k+1}) - x_k$

Else (i.e. $f^{''}(x_k) \leq 0$)

Choose

$$\begin{cases} \delta_k = -1 \textbf{ if } f^{'}(x_k) \geq 0 \\ \delta_k = 1 \textbf{ if } f^{'}(x_k) < 0 \end{cases}$$

End Choose

Choose i to be the smallest positive integer so that

$$x_{k+1} = x_k - 2^{-i}f^{'}(x_k) + \delta_k\,2^{(\frac{-i}{2})} \in (a,b)$$

Choose $0 < \lambda < 1$ so that

$$\begin{cases} x_k + \lambda\,(x_{k+1} - x_k) \in (a,b) \\ f\,(x_k + \lambda\,(x_{k+1} - x_k)) < f(x_k) \end{cases}$$

End Choose

$y_{k+1} = x_k + \lambda\,(x_{k+1} - x_k)$

End if

End do

End (algorithm)

5.2 Multidimensional Unconstrained Problems

Let X be an open subset of $\mathbb{R}^n$, $X \neq \emptyset$. Let $f : X \longrightarrow \mathbb{R}$. We are interested in the following problem:

$$\begin{array}{ll} \textbf{minimize } f(x) & \\ \textbf{subject to} & (P) \\ x \in X & \end{array}$$

Most of the time $X = \mathbb{R}^n$.

There are many problems of the type (P) that arise in applications. However, such unconstrained problems are important in nonlinear programming due to the fact that some constrained problems are successfully solved by considering a related sequence of unconstrained problems.

The three most important methods for unconstrained problems are

(i) Newton's method and variants of it

(ii) Quasi-Newton Methods

(iii) Conjugate gradient Method

A method that can be used to handle (P) in the case f is not smooth/irregular is the Nelder-Mead method.

A general algorithm for handling problem (P) is as follows:

Initial guess : x_{old}

While (Stopping criterion Not met) do

Begin

Select x_{new} so that $f(x_{new}) < f(x_{old})$

$x_{old} = x_{new}$

End

End While

5.2.1 Line Search Methods

A common way of solving problem (P) is by generating a sequence of vectors/iterates x_k, $k = 1, 2, 3, \cdots$.

We start the iteration by a judicious initial choice x_0.

At iterate x_k we seek a direction vector $d_k \neq 0$, called a search direction. Then, we restrict f on the line $\{x_k + \lambda\, d_k : \lambda \in \mathbb{R}\}$.

Set

$$\Phi_k(\lambda) = f(x_k + \lambda\, d_k).$$

Only this λ for which $x_k + \lambda\, d_k \in X$ are admissible. We try to minimize $\Phi_k(\lambda)$ as a function of λ. At the minimum we would like to have $\Phi_k(\lambda) < \Phi_k(0)$. We choose λ_k so that $\Phi_k(\lambda) < \Phi_k(0)$ and is an optimal choice of λ. This is called a line search. We will later see how the choice of d_k and λ_k affect convergence. The new iterate is now $x_{k+1} = x_k + \lambda_k\, d_k$

Definition 5.2.1 *The vector d_k in $\mathbb{R}^n$ is called a descent direction for f at iterate x_k if $d_k^T \cdot \nabla f(x_k) < 0$.*

5.2.1.1 Step Length Selection Criteria

There are several methods of choosing the step length λ_k at iteration x_k.

(i) Letting $\Phi_k(\lambda) = f(x_k+\lambda\ d_k)$ and choosing λ_k such that $\Phi_k(\lambda_k) \leq \Phi_k(\lambda)$ is called exact line search. However, this is not practical and may not necessarily contribute in a significant way in terms of solving the original problem. In addition it may require excessive computation.

(ii) λ_k is a local minimizer of $\Phi_k(\lambda)$ closest to $\lambda = 0$.

(iii) If λ_k is chosen so that

$$\Phi_k^{'}(\lambda_k) = d_k^T \cdot \nabla\ f(x_k + \lambda_k\ d_k) = 0 \textbf{ and } f(x_k + \lambda_k\ d_k) < f(x_k)$$

then we have a perfect line search.

(iv) Goldstein-Armijo Criteria:

(a) Assume $d_k^T \cdot \nabla\ f(x_k) < 0$

(b) Choose $\lambda_k > 0$ such that

(i) $f(x_{k+1}) - f(x_k) \leq \alpha\ \lambda_k\ d_k^T \cdot \nabla\ f(x_k)$

(ii) $d_k^T \cdot \nabla\ f(x_{k+1}) \geq \eta\ d_k^T \cdot \nabla\ f(x_k)$ **where** $0 < \alpha < \eta < 1$**. Typically** $\alpha = 10^{-4}$ **and** $\eta = 0.9$

(v) Modified Goldstein-Armijo Criteria:

Given $d_k^T \cdot \nabla\ f(x_k) < 0$, $0, \alpha < \eta < 1$ choose λ_k so that

(i) $f(x_{k+1}) - f(x_k) \leq \alpha\ \lambda_k\ d_k^T \cdot \nabla\ f(x_k)$

(ii) $\left|d_k^T \cdot \nabla\ f(x_{k+1})\right| \leq \eta\ \left|d_k^T \cdot \nabla\ f(x_k)\right|$

Theorem 5.2.1 (Sufficient Decrease) ***Let*** $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $f \in C^{(1)}$***. Suppose that*** f ***is bounded below. Let*** $x_1, x_2, \cdots, x_n, \cdots$ ***be generated by some minimization algorithm where***

$$x_{k+1} = x_k + \lambda_k\ d_k, \quad k = 1, 2, 3, \cdots$$

and d_k ***is such that*** $d_k^T \cdot \nabla\ f^{'}(x_k) < 0$, $k = 1, 2, 3, \cdots$ ***and*** λ_k ***is chosen to satisfy the Goldstein-Armijo Criteria. Further, suppose that*** $\nabla\ f$ ***is Lipschitz, i.e.,*** $\|\nabla\ f(x) - \nabla\ f(z)\| \leq \|x - z\|$, $x, z \in \mathbb{R}^n$***. Then***

$$\lim_{k\to\infty} \frac{d_k^T \cdot \nabla\ f(x_k)}{\|d_k\|} = 0.$$

Proof **: Condition (a) of Goldstein-Armijo Criteria gives**

$$f(x_{k+1}) - f(x_k) \leq -\alpha\ \lambda_k \sigma_k\ \|d_k\|, \quad \sigma_k = \frac{-d_k^T \cdot \nabla\ f(x_k)}{\|d_k\|}$$

Thus,

$$f(x_j) - f(x_1) \leq -\alpha \sum_{k=1}^{j-1} \lambda_k\ \sigma_k\ \|d_k\|$$

Therefore

$$\sum_{k=1}^{j-1} \lambda_k\ \sigma_k\ \|d_k\| \leq \frac{f(x_1) - f(x_j)}{\alpha}.$$

Since f is bounded below it follows that

$$\sum_{k=1}^{j-1} \lambda_k\ \sigma_k\ \|d_k\| < \infty.$$

Thus,

$$\lim_{k\to\infty} \lambda_k \ \sigma_k \ \|d_k\| = 0.$$

We now use condition (b) of Goldstein-Armijo Criteria to show that

$$\lim_{k\to\infty} \lambda_k \ \sigma_k \ \|d_k\| = 0 \textbf{ implies } \lim_{k\to\infty} \sigma_k = 0.$$

Condition (b) gives

$$\begin{aligned}
\eta \ d_k^T \cdot \ \nabla \ f(x_k) &\leq d_k^T \ \cdot \ \nabla \ f(x_{k+1}) \\
(\eta - 1) \ d_k^T \cdot \ \nabla \ f(x_k) &\leq d_k^T \ \cdot \ [\nabla \ f(x_{k+1}) - \nabla \ f(x_k)] \\
(1 - \eta) \ \cdot \ -d_k^T \cdot \ \nabla \ f(x_k) &\leq d_k^T \ \cdot \ [\nabla \ f(x_{k+1}) - \nabla \ f(x_k)] \\
(1 - \eta) \ \sigma_k \ \|d_k\| &\leq d_k^T \ \cdot \ [\nabla \ f(x_{k+1}) - \nabla \ f(x_k)] \\
&\leq \|d_k\| \ L \ \|x_{k+1} - x_k\| \\
&\leq \|d_k\| \ L \ \|d_k\|
\end{aligned}$$

Thus,

$$\begin{aligned}
\sigma_k &\leq \frac{L \ \lambda_k}{1 - \eta} \|d_k\| \\
\sigma_k^2 &\leq \frac{L}{1 - \eta} \lambda_k \ \|d_k\| \ \|\sigma_k\|
\end{aligned}$$

Since

$$\lim_{k\to\infty} \lambda_k \ \|d_k\| \ \|\sigma_k\| = 0$$

it follows that

$$\lim_{k\to\infty} \sigma_k^2 = 0. \textbf{ Thus } \lim_{k\to\infty} \sigma_k = 0.$$

□

Remark 5.2.1 ***There is a great degree of latitude in choosing*** d_k**.**

Remark 5.2.2 ***Suppose*** d_k ***is chosen to be away from being orthogonal to*** $\nabla \ f(x_k)$ ***uniformly in*** k***, i.e., suppose that there exists a*** $\delta > 0$ ***such that***

$$-\left\langle \frac{d_k}{\|d_k\|}, \frac{\nabla \ f(x_k)}{\|\nabla \ f(x_k)\|} \right\rangle \geq \delta > 0$$

Then,

$$\frac{-d_k^T \cdot \nabla \ f(x_k)}{\|d_k\| \ \|\nabla \ f(x_k)\|} \geq \delta$$

$$\frac{-d_k^T \cdot \nabla \ f(x_k)}{\|d_k\|} \geq \delta \ \|\nabla \ f(x_k)\|$$

Now, Theorem 3.2.1.1 implies that

$$\lim_{k\to\infty} \|\nabla \ f(x_k)\| = 0.$$

Remark 5.2.3 ***If an algorithm insures that***

$$\lim_{k\to\infty} \frac{-d_k^T \cdot \nabla \ f(x_k)}{\|d_k\|} = 0,$$

we say that it has a sufficient decrease in the sense of Ortega and Rheinboldt.

5.2.2 Newton's Method

Let $f : X \longrightarrow \mathbb{R}$, X an open subset of $\mathbb{R}^n$. Let $f \in C^{(2)}(X)$. Suppose we are at iterate x_k. We make a local approximation as follows:

$$f(x) \approx \psi(x) = f(x_k) + \nabla\, f(x_k)^T \cdot (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k)$$

Assume that $\nabla^2 f(x_k)$ is positive definite. Then, ψ has a unique minimizer

$$x = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla\, f(x_k).$$

In fact,

$$\nabla\psi(x) = \nabla\, f(x_k) + \nabla^2 f(x_k)(x - x_k) = 0$$

leads to

$$\begin{aligned} x - x_k &= -\left(\nabla^2 f(x_k)\right)^{-1} \nabla\, f(x_k) \\ x &= x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla\, f(x_k) \end{aligned}$$

Direct Newton Iteration Algorithm:

Initial guess x_1

For $k = 1, 2, \cdots$ **do**

$$x_{k+1} = x_k - \left(\nabla^2 f(x_k)\right)^{-1} \nabla\, f(x_k)$$

End do

We can show that $x_k \longrightarrow x^*$ quadratically provided that we start close enough to a minimizer.

Theorem 5.2.2 ***Let $f : X \longrightarrow \mathbb{R}$, X open subset of $\mathbb{R}^n$. Let $f \in C^{(2)}(X)$. Suppose that we have $x^* \in X$ such that $\nabla\, f(x^*) = 0$. Suppose $\nabla^2\, f(x^*)$ is positive definite with $\|\nabla^2\, f(x^*)^{-1}\| \leq K$. Suppose that there exists $r > 0$, $L > 0$ such that $\nabla^2\, f$ satisfies***

$$\|\nabla^2\, f(x)\nabla^2\, f(x^*)\| \leq L\|x - x^*\|, \quad x \in B(x^*, r).$$

Then, there exists $\varepsilon > 0$ such that if $x_1 \in B(x^*, \varepsilon)$, then for $\{x_k\}$ such that

$$x_{k+1} = x_k - \left[\nabla^2\, f(x_k)\right]^{-1} \nabla^2\, f(x_k)$$

we have $x_k \in B(x^*, \varepsilon)$ and

$$\|x_{k+1} - x^*\| \leq K\, L\|x_k - x^*\|^2, \quad k = 1, 2, 3, \cdots$$

Proof **: The proof of this theorem follows closely that of Theorem 3.1.2.1 and is thus omitted.**
□

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a twice continuously differentiable function. Near $x \in \mathbb{R}^n$ we have

$$f(x + p) \approx \Phi(p) = f(x) + p^T \nabla\, f(x) + \frac{1}{2} p^T \nabla^2\, f(x)\, p.$$

$$\Phi^{'}(p) = \nabla\, f(x) + \nabla^2\, f(x)\, p = 0$$

Therefore

$$p = -\left[\nabla^2\, f(x)\right]^{-1} \nabla\, f(x)$$

provided that $\left[\nabla^2\, f(x)\right]^{-1}$ exists.

Now,

$$f(x + p) \approx f(x) - \nabla\, f(x)^T \cdot \left[\nabla^2\, f(x)\right]^{-1}\; \nabla\, f(x) + \cdots.$$

we would like

$$\nabla f(x)^T \cdot \left[\nabla^2 f(x)\right]^{-1} \nabla f(x) > 0$$

We seek a diagonal matrix E with positive diagonal entries such that the lower triangular matrix $\tilde{L}$ satisfying the equation

$$\tilde{L}\tilde{L}^T = \nabla^2 f(x) + E$$

is such that

$$\sqrt{\tilde{\ell}_{ii}} \geq \delta > 0,$$

$$\tilde{L} = \left(\tilde{\ell}_{ij}\right)$$

Setting

$$D = \begin{pmatrix} \sqrt{\tilde{\ell}_{11}} & 0 \\ 0 & \sqrt{\tilde{\ell}_{nn}} \end{pmatrix}$$

Define the upper triangualar matrix $L^{'}$ by the equation

$$\tilde{L}^T = D\, L^{'}$$

Then,

$$\begin{aligned} \tilde{L}\tilde{L}^T &= L^{'T} D^T D L^{'} \\ &= L^{'T} \begin{pmatrix} \tilde{\ell}_{11} & & \\ & \ddots & \\ & & \tilde{\ell}_{nn} \end{pmatrix} L^{'} \end{aligned}$$

Thus, there is a lower triangular matrix L such that

$$\nabla^2 f(x) + E = L\, D\, L^T$$

where D is a diagonal matrix with each of the digonal entries of D bounded away from zero by a desired amount, that is, if $D = (d_{ij})$ we can arrange it so that $d_{ii} \geq \delta > 0$.

Now, in

$$f(x+p) \approx f(x) + p^T \nabla f(x) + \frac{1}{2} p^T \nabla^2 f(x)\, p,$$

if we choose p so that

$$L\, D\, L^T\, p = -\nabla f(x),$$

then

$$\begin{aligned} f(x+p) &\approx f(x) - p^T\, L\, D\, L^T\, p + \frac{1}{2} p^T\, \nabla^2 f(x)\, p \\ &= f(x) - (L^T\, p)^T\, D\, L^T\, p + \frac{1}{2} p^T\, \nabla^2 f(x)\, p \\ &= f(x) - (L^T\, p)^T\, D\, L^T\, p + \frac{1}{2} p^T\, (L\, D\, L^T - E)\, p \\ &= f(x) - \frac{1}{2} (L^T\, p)^T\, D\, L^T\, p - \frac{1}{2} p^T\, E\, p < 0 \end{aligned}$$

5.2.2.1 Modified Newton Algorithm With Line Search (Levenberg - Marquardt Method)

In the process of minimizing a function of $n-$ variables f that is twice continuously differentiable function by Newton's iteration we may encounter a problem in that the Hessian $H(x_k)$ at an iterate x_k may be singular, or $d_k = -H(x_k)^{-1} \nabla f(x_k)$ may not be a descent direction. To safeguard against these possibilities we add to $H(x_k)$ an appropriate diagonal matrix E_k with positive diagonal entries as in the above discussion so that

$$H(x_k) + E_k = L_k D_k L_k^T$$

is positive definite. A new iterate x_{k+1} can now be found from the equation

$$L_k D_k L_k^T (x_{k+1} - x_k) = -\nabla\, f(x_k)$$

This method of getting the new iteration x_{k+1} is known as Levenberg - Marquardt method. We have now the following algorithm:

pick initial guess x_0

pick tolerance $\varepsilon > 0$

If $\|\nabla\, f(x_k)\| < \varepsilon$

stop

else

Find a lower triangular matrix L_k and a diagonal matrix D_k with

positive diagonal entries such that $L_k D_k L_k^T$ is positive definite.

Solve

$$\begin{aligned} L_k D_k L_k^T\, p &= -\nabla\, f(x_k) \\ p_k &= p \end{aligned}$$

Perform a line search to pick λ_k, $0 < \lambda_k < 1$ and

Set

$$x_{k+1} = x_k + \lambda_k\, p_k$$

End If

5.2.2.2 Cauchy's Method of Steepest descent

This is the simplest Newton-type method. However, it is slow. Suppose $f : X \subset \mathbb{R}^n \longrightarrow \mathbb{R}$, $f \in C^{(1)}(X)$, where X is an open subset of $\mathbb{R}^n$. Let x_k be the current iterate. We approximate f in a neighborhood $N(x_k)$ of x_k in X as follows.

$$f(x) \approx \psi_k(x) = f(x_k) + \nabla\, f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T (x - x_k)$$

We note that

$$\begin{aligned} \psi_k(x_k) &= f(x_k) \\ \nabla\psi(x_k) &= \nabla\, f(x_k) \\ \nabla^2\psi_k(x_k) &= I \\ \nabla\psi_k(x) &= 0 \text{ if } x = x_k - \nabla\, f(x_k) \end{aligned}$$

Thus, ψ_k is minimized at $x = x_k - \nabla\, f(x_k)$.

In the method of steepest descent

$$d_k = -\nabla\, f(x_k)$$

Algorithm (Steepest Descent)

Initial guess $x_1 \in \mathbb{R}^n$

For $k = 1, 2, \cdots$ **(until norm of** ∇f **is less than** $\varepsilon > 0$

do

$d_k = -\nabla f(x_k)$

Use Goldstin - Armijo criteria to select λ_k **and**

Set $x_{k+1} = x_k + \lambda_k d_k$

End do

Let Q **be a positive definite symmetric matrix. Consider the problem of solving**

$$\begin{aligned} &\textbf{min} \quad f(x) \\ &f(x) = \frac{1}{2}x^T Q x - c^T x \end{aligned}$$

by steepest descent method.

The steepest descent direction at iteration k **is given by**

$$\begin{aligned} d_k &= -\nabla f(x_k) \\ &= -(Qx_k - c) \end{aligned}$$

The next iterate x_{k+1} **is given by the formula**

$$x_{k+1} = x_k + \lambda_k \, d_k$$

where λ_k **is determined by a line search. Suppose we use exact line search. Then** λ_k **is chosen to be the minimizer of** $\lambda \longrightarrow f(x_k + \lambda \, d_k)$ **and thus**

$$\lambda_k = -\frac{\nabla f(x_k)^T \, d_k}{d_k^T Q \, d_k}$$

Theorem 5.2.2.1 ***Assume*** $\{x_k\}$ ***is the sequece of approximate solutions obtained when the steepest descent method is applied to the quadratic function*** $f(x) = \frac{1}{2}x^T \, Q \, x - c^T \, x$***, and where an exact line search is used. Then, for any*** x_0 ***the method converges to the unique minimizer*** x^* ***of*** f***, and further more,***

$$f(x_{k+1}) - f(x^*) \leq \left[\frac{K(Q)-1}{K(Q)+1}\right]^2 (f(x_k) - f(x^*))$$

where $K(Q)$ ***is the condition number of the matrix*** Q***.***

Proof **: Since** $x^* = Q^{-1}c$**, that is** $c = Q\, x^*$**, we have**

$$\begin{aligned} f(x_k) - f(x^*) &= \frac{1}{2}\left(x_k^T Q x_k - c^T x_k\right) - \left(\frac{1}{2}x^{*T} Q x^* - c^T x^*\right) \\ &= \frac{1}{2}\left(x_k^T Q x_k - (Qx^*)^T x_k\right) - \left(\frac{1}{2}x^{*T} Q x^* - (Qx^*)^T x^*\right) \\ &= \frac{1}{2}(x_k - x^*)^T \, Q \, (x_k - x^*) \end{aligned}$$

Determine the next iterate $x_{k+1} = x_k + \alpha_k \, d_k$**,** $d_k = x_k - \alpha_k \, \nabla \, f(x_k)$ **where** α_k **is determined by exact line search. That is,**

$$\alpha_k = -\nabla \, f(x_k)^T \frac{d_k}{d_k^T \, Q \, d_k}, \qquad d_k = -\nabla \, f(x_k)$$

Setting

$$E(x) = \frac{1}{2}(x - x^*)^T(x - x^*)$$

we can verify that

$$E(x_{k+1}) = \left[1 - \frac{\left(\nabla f(x_k)^T \nabla f(x_k)\right)^2}{\left(\nabla f(x_k)^T\ Q\ \nabla f(x_k)\right)\left(\nabla f(x_k)^T\ Q^{-1}\ \nabla f(x_k)\right)}\right]\ E(x_k)$$

Next, for any vector $y \neq 0$

$$\frac{(y^T y)^2}{(y^T\ Q\ y)(y^T\ Q^{-1}\ y)} \geq 1 - \left[\frac{K(Q) - 1}{k(Q) + 1}\right]^2$$

Thus,

$$E(x_{k+1}) \leq 1 - \left[\frac{K(Q) - 1}{k(Q) + 1}\right]^2\ E(x_k).$$

That is,

$$f(x_{k+1}) - f(x^*) \leq 1 - \left[\frac{K(Q) - 1}{k(Q) + 1}\right]^2\ (f(x_k) - f(x^*))$$

□

Remark 5.2.4 ***The steepest descent method does not require second derivatives. We do not need to solve a linear system for descent directions. In addition matrix storage is not required. However, the method is slow as is shown in Theorem 3.2.2.1.***

Example 5.2.1

min $f(x)$

where

$$f(x) = \frac{1}{2}x^T\ Q\ x + c^T x$$

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 25 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad c = \begin{pmatrix} 1 \\ -1 \\ 2 \\ -1 \\ 1 \end{pmatrix}$$

Approximately how many iterations are required to come within 10^{-8} **of the minimum ?**

5.2.3 Newton's Method with a Model Trust Region

In this section we will describe another modification of Newton's method. All norms described in this section are ℓ_2**−norms. Let** $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ **and** $f \in C^{(2)}$**. Suppose that we are at iterae** x_k**. For**

$$s \in S = \{s \in \mathbb{R}^n : \|s\| \leq r_k, \quad r_k > 0\} \tag{5.1}$$

we approximate $f(x_k + s)$ **by**

$$f(x_k + s) \approx f(x_k) + \psi_k(s) \tag{5.2}$$

where

$$\psi_k(s) = [\nabla\ f(x_k)]^T\ s + \frac{1}{2}s^T \nabla^2\ f(x_k)s. \tag{5.3}$$

The set S in (5.1) is called the *trust region* and r_k is called the trust region radius. We compute the step s_k so that s_k solves the following problem:

$$\begin{array}{l} \min \psi_k(s) \\ \textbf{subject to} \\ \|s\| \leq r_k \end{array} \tag{5.4}$$

and then if $\psi_k(s_k)$ compares "well" with $f(x_k + s_k) - f(x_k)$ we let $x_{k+1} = x_k + s_k$. Otherwise we reduce r_k and repeat the process.

In practice (5.4) is only solved approximately. One set of criteria for such an approximate solution s_k is

$$\begin{array}{rcl} \psi_k(s_k) - \psi_k^* & \leq & \sigma\,(2-\sigma)|\psi_k^*| \\ \|s_k\| & \leq & (1+\sigma)\,r_k \end{array} \tag{5.5}$$

where $\sigma \in (0,1)$ is a fixed constant and ψ_k^* is the exact value. We next present the algorithm.

Algorithm (Newton's Method with a Trust Region)

(Initialization): $x_1 \in \mathbb{R}^n$, $r_1 > 0$, $\sigma \in (0,1)$, $0 < \gamma_1 < \gamma_2 < 1 < \gamma_3$, $0 < \mu < \eta < 1$.

For $k = 1, 2, \cdots$ (until some convergence criteria is satisfied) do

Compute $\nabla\, f(x_k)$ and $\nabla^2\, f(x_k)$

Solve (5.4) to get s_k that satisfies (5.5) (or any other set of criteria available).

Compute the ratio ρ_k which is the ratio of the actual reduction $f(x_k + s_k) - f(x_k)$

and the predicted reduction $\psi_k(s_k)$.

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{\psi_k(s_k)}$$

If $\rho_k < \mu$ (i.e. we don't accept the potential reduction) then

$r_k = r \in [\gamma_1\, r_k, \gamma_2\, r_k]$

go to Step 2;

Else

$x_{k+1} = x_k + s_k$

End If

If $\rho_k < \eta$ then

$r_{k+1} = r \in [\gamma_2\, r_k, r_k]$

Else

$r_{k+1} = r \in [r_k, \gamma_3\, r_k]$

End If

End do

Theorem 5.2.3 ***Suppose that $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $f \in C^{(2)}$ and that the level set $L(x_1) = \{x : f(x) \leq f(x_1)\}$ is compact. Let the sequence $\{x_k\}$ be generated using the algorithm for Newton's method with trust region. Then either the algorithm terminates with some $x_\ell \in L(x_1)$ with $\nabla\, f(x_\ell) = 0$ and $\nabla^2\, f(x_\ell)$ positive semidefinite, or it generates a sequence $\{x_k\}$ with limit point $x^* \in L(x_1)$ with $\nabla\, f(x^*) = 0$ and $\nabla^2\, f(x^*)$ positive semidefinite.***

Proof **: The proof is left as an exercice** □

5.2.4 Conjugate Gradient Method

The conjugate gradient method is designed to solve the equation

$$A\,x = b \tag{5.6}$$

where A is a symmetric positive definite matrix. Solving (5.6) in the case A is symmetric and positive definite is equivalent to the problem

$$\begin{aligned} \min \quad & f(x) \\ f(x) = & \frac{1}{2} x^T\,A\,x - b^T\,x \end{aligned}$$

we note that $\nabla\,f(x) = A\,x - b$ and thus $\nabla\,f(x) = 0$ if $A\,x = b$. Also $\nabla^2\,f(x) = A$ assuring us that, in fact, the solution of $A\,x = b$ is the unique minimizer of $f(x)$.

The conjugate gradient method gets its name from the fact that it generates a set of vectors d_i that are conjugate with respect to the matrix A. That is,

$$d_i^T\,A\,d_j = 0, \qquad i \neq j$$

Suppose the vectors $d_1, d_2, \cdots, d_m$ are known. Let

$$y = \alpha_1\,d_1 + \alpha_2\,d_2 + \cdots + \alpha_m\,d_m.$$

Then

$$\begin{aligned} f(y) &= \frac{1}{2}\left(\sum \alpha_i\,d_i^T\right)\,A\,\left(\sum \alpha_j\,d_j^T\right) - b^T\,\sum \alpha_j\,d_j \\ &= \frac{1}{2}\sum\left(\alpha_i^2\,d_i^T\,A\,d_i - \alpha_i\,b^T\,d_i\right) \end{aligned}$$

Then,

$$\begin{aligned} \min_y f(y) &= \frac{1}{2}\min_{\alpha_1,\cdots,\alpha_m}\sum_i\left(\alpha_i^2\,d_i^T\,A\,d_i - \alpha_i\,b^T\,d_i\right) \\ &= \frac{1}{2}\sum_i \min_{\alpha_i}\left(\alpha_i^2\,d_i^T\,A\,d_i - \alpha_i\,b^T\,d_i\right) \end{aligned}$$

Thus, we need

$$\begin{aligned} \alpha_i\,d_i^T\,A\,d_i - b^T\,d_i &= 0 \\ \alpha_i &= \frac{b^T\,d_i}{d_i^T\,A\,d_i} \end{aligned} \tag{5.7}$$

The formula for α_i corresponds to an exact line search along the direction d_i.

The conjugaate gradient method iteratively determines a set of conjugate vectors $\{d_i\}$ and corresponding coefficients $\{\alpha_i\}$.

Let

$$\begin{aligned} r_i &= b - A\,x_i; \quad \textbf{residual at iteration } x_i \\ r_0 &= b - A\,x_0 \\ d_{-1} &= 0 \\ \beta_0 &= 0 \\ x_0 &= 0 \end{aligned}$$

Then,

$$\begin{aligned} d_0 &= r_0 + \beta_0\, d_{-1} = r_0 = b - A\, x_0 \\ \alpha_0 &= \frac{r_0^T\, r_0}{d_0^T\, A\, d_0} \\ x_1 &= x_0 + \alpha_0\, d_0 \\ r_1 &= r_0 - \alpha_0\, d_0 \end{aligned}$$

Conjugate Gradient Algorithm

Set $x_0 = 0$, $r_0 = b - A\, x_0$, $d_{-1} = 0$, $\beta_0 = 0$

$d_0 = r_0$

$\alpha_0 = \left(r_0^T\, r_0\right) / \left(d_0^T\, A\, d_0\right)$

$x_1 = x_0 + \alpha_0\, d_0$

$r_1 = r_0 - \alpha_0\, d_0$

Specify convergence tolerance $\varepsilon > 0$

For $i = 0, 1, 2, \cdots$ **do**

  If $\|r_i\| < \varepsilon$ **Stop**

  Else

    If $i > 0$ **then**

      $\beta_i = \frac{r_i^T\, r_i}{r_{i-1}^T\, r_i}$

      $d_i = r_i + \beta_i\, d_{i-1}$

      $\alpha_i = \frac{r_i^T\, r_i}{d_i^T\, A\, d_i}$

      $x_{i+1} = x_i + \alpha_i\, d_i$

      $r_{i+1} = r_i - \alpha_i\, A\, d_i$

    End If

  End If

End do

The conjugate gradient algorithm generates a set of orthogonal vectors $\{r_i\}$ and conjugate vectors $\{d_i\}$. We can also choose $x_0 \neq 0$ in the algorithm. The conjugate gradient method does not require the matrix A explicitly. It only requires the computation of the matrix vector product $A\, d_i$.

Theorem 5.2.4 ***Assume that the vectors $\{d_i\}$ and $\{r_i\}$ are defined by the formulas for the conjugate gradient method. Then,***

$$r_i^T\, r_j = 0, \quad r_i^T\, d_j = 0, \quad \textbf{\textit{and}} \quad d_i^T\, A\, d_j = 0, \qquad i > j.$$

Proof **: We prove the theorem by induction. First we verify $r_1^T\, r_0 = r_1^T\, d_0 = d_1^T\, A\, d_0$.**

$$\begin{aligned} r_1^T\, r_0 = (r_0 - \alpha_0\, A\, d_0)^T\, r_0 &= r_0^T\, r_0 - \alpha_0\, d_0^T\, A\, r_0 \\ &= r_0^T\, r_0 - \frac{r_0^T\, r_0}{d_0^T\, A\, d_0} d_0^T\, A\, r_0 \end{aligned}$$

Since $d_0 = r_0 + \beta_0\, d_{-1} = r_0$ we have

$$r_0^T\, r_0 - \frac{r_0^T\, r_0}{d_0^T\, A\, d_0} d_0^T\, A\, r_0 = r_0^T\, r_0 - \frac{r_0^T\, r_0}{d_0^T\, A\, d_0} d_0^T\, A\, d_0 = 0.$$

Next,

$$r_1^T\, d_0 = r_1^T\, r_0 = 0$$

Finally we verify that

$$d_1^T\, A\, d_0 = 0.$$

We have

$$r_1 = r_0 - \alpha_0\, A\, d_0, \qquad d_1 = r_1 + \beta_1\, r_0$$

Thus,

$$A\, d_0 = \frac{1}{\alpha_0}\, (r_0 - r_1)$$

Now,

$$\begin{aligned} d_1^T\, A\, d_0 &= (r_1 + \beta_1\, r_0)^T \frac{1}{\alpha_0} (r_0 - r_1) \\ &= \frac{1}{\alpha_0} (r_1^T\, r_0 + \beta_1\, r_0^T\, r_0 - r_1^T\, r_1 - \beta_1\, r_0^T\, r_1) \\ &= \frac{1}{\alpha_0} (0 + \beta_1\, r_0^T\, r_0 - r_1^T\, r_1 - \beta_1 \cdot 0) \\ &= \frac{1}{\alpha_0} (\beta_1\, r_0^T\, r_0 - r_1^T\, r_1) \\ &= \frac{1}{\alpha_0} \left(\frac{r_1^T\, r_1}{r_0^T\, r_0} (r_0^T\, r_0) - r_1^T\, r_1 \right) \\ &= 0 \end{aligned}$$

We now proceed by induction. We assume that the theorem is valid for case i and verify its validty for case $i+1$. Case $i = 1$ has already been verified to be true. In what follows we assume $i > 1$.

(1) $r_{i+1}^T\, r_j = (r_i - \alpha_i\, A\, d_i)^T\, r_j = r_i^T\, r_j - \alpha_i\, d_i^T\, A\, r_j$**. If $j < i$, by the induction hypothesis**

$$r_i^T\, r_j = 0, \qquad d_i^T\, A\, r_j = 0.$$

Therefore $r_{i+1}^T\, r_j = 0$ if $j < i$. It remains to prove $r_{i+1}^T\, r_i = 0$.

$$\begin{aligned} r_{i+1}^T\, r_i &= (r_i - \alpha_i\, A\, d_i)^T\, r_i = r_i^T\, r_i - \alpha_i\, d_i^T\, A\, r_i \\ &= r_i^T\, r_i - \alpha_i\, (d_i^T\, A)\, (d_i - \beta_i\, d_{i-1}) \\ &= r_i^T\, r_i - \alpha_i\, d_i^T\, A\, d_i + \alpha_i\, \beta_i\, d_i^T\, A\, d_{i-1} \\ &= r_i^T\, r_i - \alpha_i\, d_i^T\, A\, d_i + \alpha_i\, \beta_i\, \cdot\, 0 \end{aligned}$$

Since by induction hypothesis $d_i^T\, A\, d_{i-1} = 0$. Since

$$\alpha_i = \frac{r_i^T\, r_i}{d_i^T\, A\, d_i}$$

it follows that

$$r_i^T\ r_i - \alpha_i\ d_i^T\ A\ d_i = 0$$

Thus,

$$r_{i+1}^T r_i = 0.$$

Thus, we have shown $r_{i+1}^T\ r_j = 0$ if $j < i+1$.

(2) Next, we verify $r_{i+1}^T\ d_j = 0$ for $j < i+1$.

$$\begin{aligned} r_{i+1}^T\ d_j &= r_{i+1}^T\ (r_j + \beta_j\ d_{j-1}) \\ &= r_{i+1}^T\ r_j + \beta_j\ r_{i+1}^T\ d_{j-1} \end{aligned}$$

In (1) above we have already seen that $r_{i+1}^T\ r_j = 0$. Thus

$$\begin{aligned} r_{i+1}^T\ d_j &= \beta_j\ r_{i+1}^T\ d_{j-1} \\ &= \beta_j\ (r_i - \alpha_i\ A\ d_i)^T\ d_{j-1} \\ &= \beta_j\ r_i^T\ d_{j-1} - \alpha_i\ \beta_j\ d_i^T\ A\ d_{j-1} \end{aligned}$$

By the induction hypothesis

$$\begin{aligned} r_i^T\ d_{j-1} &= 0 \\ d_i^T\ A\ d_{j-1} = 0 \end{aligned}$$

Thus,

$$r_{i+1}^T\ d_j = 0, \qquad j < i+1.$$

(3) Finally we prove that $d_{i+1}^T\ A\ d_j = 0$ for $j < i+1$ and the induction will be complete.

$$\begin{aligned} d_{i+1}^T\ A\ d_j &= (r_{i+1} + \beta_{i+1}\ d_i)^T\ A\ d_j \\ &= (r_{i+1}^T + \beta_{i+1}\ d_i^T)\ A\ d_j \\ &= r_{i+1}^T\ A\ d_j + \beta_{i+1}\ d_i^T\ A\ d_j \end{aligned}$$

By the induction hypothesis $d_i^T\ A\ d_j = 0$ if $j < i$. Assuming $j < i$ we would like to show that

$$r_{i+1}^T\ A\ d_j = 0$$

Now,

$$\begin{aligned} r_{i+1}^T\ A\ d_j &= r_{i+1}^T\ \frac{1}{\alpha_j}(r_{j+1} - r_j) \\ &= \frac{1}{\alpha_j} r_{i+1}^T\ r_{j+1} - \frac{1}{\alpha_j}\ r_{i+1}^T\ r_j \end{aligned}$$

Since $j < i$, we have shown in (1) that $r_{i+1}^T\ r_j = 0$. If $j < i$ then $j+1 \le i$, and by (1) again $r_{i+1}^T\ r_{j+1} = 0$. Thus,

$$d_{i+1}^T\ A\ d_j = 0 \qquad \textbf{if } j < i.$$

Next we show that $d_{i+1}^T\ A\ d_i = 0$.

$$\begin{aligned} d_{i+1}^T\ A\ d_i &= (r_{i+1} + \beta_{i+1}\ d_i)\ \left(\alpha_i^{-1}(r_i - r_{i+1})\right) \\ &= \alpha_i^{-1}\left(r_{i+1}^T\ r_i + \beta_{i+1}\ d_i^T\ r_i - r_{i+1}^T\ r_{i+1} - \beta_{i+1}\ d_i^T\ r_{i+1}\right) \\ &= \alpha_i^{-1}\left(\beta_{i+1}\ d_i^T\ r_i - r_{i+1}^T\ r_{i+1}\right) \quad \textbf{since} \quad d_i^T\ r_{i+1} = 0 \textbf{ and } r_{i+1}^T\ r_i = 0 \\ &= \alpha_i^{-1}\left(\beta_{i+1}(r_i + \beta_i\ d_{i-1})^T\ r_i - r_{i+1}^T\ r_{i+1}\right) \\ &= \alpha_i^{-1}\left(\beta_{i+1}\ r_i^T\ r_i + \beta_{i+1}\ \beta_i\ d_{i-1}^T\ r_i - r_{i+1}^T\ r_{i+1}\right) \quad \textbf{since } d_{i-1}^T\ r_i = 0 \\ &= \alpha_i^{-1}\left(\beta_{i+1}\ r_i^T\ r_i - r_{i+1}^T\ r_{i+1}\right) \\ &= \alpha_i^{-1}\left(\frac{r_{i+1}^T\ r_{i+1}}{r_i^T\ r_i}\ r_i^T\ r_i - r_{i+1}^T\ r_{i+1}\right) \\ &= 0 \end{aligned}$$

□

5.2.5 Nonlinear Conjugate Gradient Method

The conjugate gradient method solves the problem

$$\begin{aligned} &\mathbf{min} \quad f(x) \\ &f(x) = \frac{1}{2} x^T\, A\, x - b^T\, x \end{aligned}$$

by computing a conjugate direction d_i **using the residual** $r_i = b - A\, x_i\, (= -\nabla\, f(x_i))$**. In adopting the method to more general nonlinear function the computation of** α_i **in the conjugate gradient algorithm is replaced by a line search.**

Residuals

1. $r_0 = b - A\, x_0$
2. $r_1 = r_0 - \alpha_0\, A\, d_0 = b - A\, x_0 - \alpha_0\, A\, d_0 = b - A\, (x_0 + \alpha_0 \partial_0) = b - A\, x_1$
3. **Verify** $r_i = b - A\, x_i$

If $i = 1$ **the conclusion is true.**

Assume case i **to be true.**

Then,

$$\begin{aligned} r_{i+1} = r_i - \alpha_i\, A\, d_i &= b - A\, x_i - \alpha_i\, A\, d_i \\ &= b - A\, (x_i + \alpha_i\, d_i) \\ &= b - A\, x_{i+1} \end{aligned}$$

Thus case $i + 1$ **is also true.**

By induction $r_i = b - A\, x_i$

On the definition of α_i **in the conjugate gradient algorithm,**

$$\alpha_0 = \frac{r_0^T\, r_0}{d_0^T\, A\, d_0} \qquad r_0 = b - A\, x_0 = b \quad \mathbf{if} \quad x_0 = 0. \quad \mathbf{Also} \quad d_0 = r_0.$$

Thus,

$$\alpha_0 = \frac{b^T\, d_0}{d_0^T\, A\, d_0}$$

We have

$$\alpha_i = \frac{r_i^T\, r_i}{d_i^T\, A\, d_i}.$$

We would like to verify that $r_i^T\, r_i = b^T\, d_i$**. We have already seen** $r_0^T\, r_0 = b^T\, d_0$**. In what follows assume that** $i \geq 1$**.**

$$\begin{aligned} r_i^T\, r_i &= r_i^T\, (d_i - \beta_i\, d_{i-1}) \\ &= r_i^T\, d_i - \beta_i\, r_i^T\, d_{i-1} \\ &= r_i^T\, d_i \qquad \mathbf{since} \quad r_i^T\, d_{i-1} = 0 \\ &= (b_A\, x_i)^T\, d_i = b^T\, d_i - (A\, x_i)^T\, d_i. \end{aligned}$$

We are going to show that $(A\,x_i)^T\,d_i = 0$**.**

$$\begin{aligned}
(A\,x_i)^T\,d_i &= x_i^T\,A\,d_i \\
&= (x_{i-1} + \alpha_i\,d_{i-1})^T\,A\,d_i \\
&= x_{i-1}^T\,A\,d_i + \alpha_i\,d_{i-1}^T\,A\,d_i \\
&= x_{i-1}^T\,A\,d_i \qquad \textbf{since} \quad d_{i-1}^T\,A\,d_i = 0. \\
&= (x_{i-2} + \alpha_{i-2}\,d_{i-2})^T\,A\,d_i \\
&= x_{i-2}^T\,A\,d_i \\
&= \vdots \\
&= x_0^T\,A\,d_i = 0 \qquad \textbf{since} \quad x_0 = 0.
\end{aligned}$$

Nonlinear Conjugate Gradient Method (Algorithm)

Initial guess x_0

$r_0 = -\nabla\,f(x_0)$**,** $\quad d_{-1} = 0$**,** $\quad \beta_0 = 0$**,** $\quad d_0 = r_0$

$x_1 = x_0 + \alpha\,d_0$ **where** α_0 **is chosen by a line search**

Specify convergence tolerance $\varepsilon > 0$

For $i = 0, 1, 2, \cdots$ **do**

If $\|\nabla\,f(x_0)\| < \varepsilon$ **Stop**

If $i > 0$

Set

$$\beta_i = \frac{\nabla\,f(x_i)^T\,\nabla\,f(x_i)}{\nabla\,f(x_{i-1})^T\,\nabla\,f(x_{i-1})}$$

Set $d_i = -\nabla\,f(x_i) + \beta_i\,d_{i-1}$

Set $x_{i+1} = x_i + \lambda_i\,d_i$ **where** λ_i **is chosen by a line search**

End if

End do

5.2.6 Expanding Subspace Property

Let

$$f(x) = \frac{1}{2}x^T\,A\,x - b^T\,x$$

where A **is an** $n \times n$ **symmetric matrix. Let** $d_1, d_2, \cdots, d_n$ **be** $A-$**conjugate (i.e.** $d_i^T\,A\,d_j = 0$ **for** $i \neq j$**) Let** x_1 **be an arbitrary point in** $\mathbb{R}^n$**. For** $k = 1, 2, \cdots, n$ **let** λ_k **be a solution to the problem**

$$\min\,\{f(x_k + \lambda\,d_k) : \lambda \in \mathbb{R}\}\,.$$

Then, we have for $k = 1, 2, \cdots, n$

(i) $\nabla\,f(x_{k+1})^T\,d_j = 0, \quad j = 1, \cdots, k$

(ii) $\nabla\,f(x_1)^T\,d_k = \nabla\,f(x_k)^T\,d_k$

(iii) $f(x_{k+1} = \min\{f(x) : x \in x_1 + L\}$ **where**

$L = \{\alpha_1\,d_1 + \cdots + \alpha_k\,d_k | \alpha_1, \cdots, \alpha_k \in \mathbb{R}\}$

(i) Since $f(x_j + \lambda\, d_j)$, as a function of λ, attains its minimum at $\lambda = \lambda_j$ it follows that

$$\nabla\, f(x_j + \lambda_j\, d_j)^T\, d_j = 0$$

That is,

$$\nabla\, f(x_{j+1})^T\, d_j = 0$$

Thus,

$$\nabla\, f(x_{k+1})^T\, d_j = 0 \qquad \textbf{if} \quad j = k.$$

We have

$$\begin{aligned} \nabla\, f(x_{k+1}) &= A\, x_{k+1} - b \\ &= A\, (x_k + \lambda_k\, d_k) - b \\ &= A\, x_k - b + \lambda_k\, A\, d_k \\ &= \nabla\, f(x_k) + \lambda_k\, A\, d_k \end{aligned}$$

Now,

$$\nabla\, f(x_{k+1})^T\, d_{k-1} = \nabla\, f(x_k)^T\, d_{k-1} + \lambda_k\, d_k^T\, A\, d_{k-1} = 0$$

Since the vectors are $A-$conjugate $d_k^T\, A\, d_{k-1} = 0$. The fact that $\nabla\, f(x_k)^T\, d_{k-1} = 0$ follows since

$$\frac{d}{d\,\lambda} f(x_{k-1} + \lambda\, d_k) = 0 \quad \textbf{if} \quad \lambda = \lambda_k.$$

Thus,

$$\nabla\, f(x_{k+1})^T\, d_{k-1} = 0.$$

Next,

$$\begin{aligned} \nabla\, f(x_{k+1}) &= \nabla\, f(x_k) + \lambda_k\, A\, d_k \\ &= \nabla\, f(x_{k-1}) + \lambda_{k-1}\, A\, d_{k-1} + \lambda_k\, A\, d_k \end{aligned}$$

Thus,

$$\nabla\, f(x_{k+1})^T\, d_{k-2} = \nabla\, f(x_{k-1})^T\, d_{k-2} + \lambda_{k-1}\, d_{k-1}^T\, A\, d_{k-2} + \lambda_k\, d_k^T\, A\, d_{k-2} = 0$$

since

$$\nabla\, f(x_{k-1})^T\, d_{k-2} = 0, \quad d_{k-1}^T\, A\, d_{k-2} = 0, \quad \textbf{and} \quad d_k^T\, A\, d_{k-2} = 0.$$

Thus, we can use induction to verify that (i) is true.

(ii) $\nabla\, f(x_1)^T\, d_k = \nabla\, f(x_k)^T\, d_k$ is true if $k = 1$.
If $k \geq 2$,

$$\begin{aligned} \nabla\, f(x_k) &= A\, x_k - b \\ &= A\, (x_{k-1} + \lambda_{k-1}\, d_{k-1}) - b \\ &= A\, x_{k-1} - b + \lambda_{k-1}\, A\, d_{k-1} \\ &= \nabla\, f(x_{k-1}) + \lambda_{k-1}\, A\, d_{k-1} \\ &= \nabla\, f(x_{k-2}) + \lambda_{k-2}\, A\, d_{k-2} + \lambda_{k-1}\, A\, d_{k-1} \\ &= \nabla\, f(x_1) + \sum_{j=1}^{k} \lambda_j\, A\, d_j \end{aligned}$$

Now,

$$\begin{aligned} \nabla\, f(x_k)^T\, d_k &= \nabla\, f(x_1)^T\, d_k + \sum_{j=1}^{k-1} \lambda_j\, d_j^T\, A\, d_k \\ &= \nabla\, f(x_1)^T\, d_k \end{aligned}$$

as each of the terms in the sum is zero by conjugacy.

(iii) To verify the last statement we note that

$$f(x_1 + \alpha_1\, d_1 + \cdots + \alpha_k\, d_k) = f(x_1) + \sum_{j=1}^{k} \alpha_j\, d_j^T\, \nabla\, f(x_1) + \frac{1}{2} \sum_{j=1}^{k} \alpha_j^2\, d_j^T\, A\, d_j$$

Using (ii) we can replace $d_j^T\, \nabla\, f(x_1)$ **by** $d_j^T\, \nabla\, f(x_j)$ **in the second sum so that**

$$\begin{aligned} f(x_1 + \alpha_1\, d_1 + \cdots + \alpha_k\, d_k) &= f(x_1) + \sum_{j=1}^{k} \alpha_j\, d_j^T \nabla\, f(x_j) + \sum_{j=1}^{k} \alpha_j^2\, d_j^T\, A\, d_j \\ &= f(x_1) + \sum_{j=1}^{k} \left[\alpha_j\, d_j^T \nabla\, f(x_j) + \alpha_j^2\, d_j^T\, A\, d_j\right] \end{aligned}$$

Since

$$f(x_j + \lambda_j\, d_j) \leq f(x_j + \lambda\, d_j), \quad \lambda \in \mathbb{R}$$

we have

$$f(x_j + \lambda_j\, d_j) \leq f(x_j + \alpha_i\, d_j)$$

Thus,

$$f(x_j) + \lambda_j\, \nabla\, f(x_j)^T\, d_j + \frac{1}{2}\lambda_j^2\, d_j^T\, A\, d_j \leq f(x_j) + \alpha_j\, \nabla\, f(x_j)^T\, d_j + \frac{1}{2}\alpha_j^2\, d_j^T\, A\, d_j$$

That is,

$$\lambda_j\, \nabla\, f(x_j)^T\, d_j + \frac{1}{2}\lambda_j^2\, d_j^T\, A\, d_j \leq \alpha_j\, \nabla\, f(x_j)^T\, d_j + \frac{1}{2}\alpha_j^2\, d_j^T\, A\, d_j$$

Thus,

$$\begin{aligned} f(x_1 + \alpha_1\, d_1 + \cdots + \alpha_k\, d_k) &= f(x_1) + \sum_{j=1}^{k} \left[\alpha_j\, d_j^T \nabla\, f(x_j) + \alpha_j^2\, d_j^T\, A\, d_j\right] \\ &\geq f(x_1) + \sum_{j=1}^{k} \left[\lambda_j\, d_j^T \nabla\, f(x_j) + \lambda_j^2\, d_j^T\, A\, d_j\right] \\ &= f(x_1 + \lambda_1\, d_1 + \cdots + \lambda_k\, d_k) \\ &= f(x_{k+1}) \end{aligned}$$

Note that

$$d_k^T\, A\, x_k = d_k^T\, A\, (x_{k-1} + \lambda_{k-1}\, d_{k-1}) = d_k^T\, A\, x_{k-1} + \lambda_{k-1}\, d_k^T\, A\, d_{k-1} = d_k^T\, A\, x_{k-1}$$

since

$$x_{k+1} = x_k + \lambda_k\, d_k, \quad k = 1, 2, \cdots$$

We note that if $k \geq 2$

$$\begin{aligned} x_{k+1} &= x_k + \lambda_k\, d_k \\ &= (x_{k-1} + \lambda_{k-1}\, d_{k-1}) + \lambda_k\, d_k \\ &= x_{k-1} + \lambda_{k-1}\, d_{k-1} + \lambda_k\, d_k \\ &\vdots \\ &= x_1 + \lambda_1\, d_1 + \lambda_2\, d_2 + \cdots + \lambda_k\, d_k \end{aligned}$$

In (2.3.2) α_i **is given by the formula**

$$\alpha_i = \frac{b^T\, d_i}{d_i^T\, A\, d_i}$$

In the conjugate gradient algorithm α_i is given by the formula

$$\alpha_i = \frac{r_i^T\ r_i}{d_i^T\ A\ d_i}$$

In fact,

$$\frac{b^T\ d_i}{d_i^T\ A\ d_i} = \frac{r_i^T\ r_i}{d_i^T\ A\ d_i}$$

Note that $\lambda \longrightarrow f(x_k + \lambda\ d_k)$ is minimized when $\lambda = \lambda_k$. That is $d_k^T\ \nabla\ f(x_k + \lambda\ d_k) = 0$.

$$\begin{aligned} d_k^T\ \nabla\ f(x_k + \lambda\ d_k) &= d_k^T \left[A(x_k + \lambda\ d_k) - b^T\ d_k\right] \\ &= d_k^T \left[A\ x_k + \lambda\ d_k^T\ A\ d_k - b^T\ d_k\right] \end{aligned}$$

Therefore

$$\lambda = \frac{b^T\ d_k}{d_k^T\ A\ d_k}$$

where we have used the fact that $d_k^T\ A\ x_k = 0$.

Thus, from the expanding subspace property we see that the conjugate gradient algorithm should converge to the minimum point in at most n iterations regardless of our starting point x_0.

5.3 Quasi-Newton Methods

Let $f : X \longrightarrow \mathbb{R}$ where X is an open subset of $\mathbb{R}^n$ that is nonempty. Let f be twice continuously differentiable on X.

Again we are interested in minimizing the function on X. The methods we discuss in this section are called quasi-Newton methods and are generalizations of a method for one dimensional problems called the secant method. The secant method for one dimension employs the approximation

$$f^{''}(x_k) \approx \frac{f^{'}(x_k) - f^{'}(x_{k-1})}{x_k - x_{k-1}}$$

in the formula for Newton's method

$$x_{k+1} = x_k - \frac{f^{'}(x_k)}{f^{''}(x_k)}$$

resulting in the formula

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f^{'}(x_k)$$

Under appropriate assumptions the secant method converges superlinearly with rate $r = (1+\sqrt{5})/2$. (Samuel D. Conte and Carl W. deBool, Elementary Numerical Analysis: An Algorithm Approach, McGraw-Hill, New York, 1980).

To generalize the secant method for one dimensional problems to multidimensional problems we start with

$$f^{''}(x_k)(x_k - x_{k+1}) \approx f^{'}(x_k) - f^{'}(x_{k-1})$$

and the multidimensional version will be

$$\nabla^2\ f(x_k)(x_k - x_{k-1}) \approx \nabla\ f(x_k) - \nabla\ f(x_{k-1})$$

We now seek H_k, an approximation to the Hessian $\nabla^2\ f(x_k)$, and write

$$H_k\ (x_k - x_{k-1}) \approx \nabla\ f(x_k) - \nabla\ f(x_{k-1}).$$

In the case

$$f(x) = \frac{1}{2} x^T \, Q \, x + c^T \, x, \qquad H_k = Q.$$

As usual to minimize f we generate a sequence of iterates $x_1, x_2, \cdots, x_k, \cdots$. Suppose we are at iterate x_k. Let J_k be a nonsingular $n \times n-$matrix. Consider the affine scaling

$$S_k(w) = x_k + J_k(w)$$

where w is in a neighborhood $N_\delta(0)$ of zero.

Let

$$\begin{aligned} \varphi_k(w) &= f(x_k + J_k \, w) \quad (= f \circ S_k(w)) \\ \varphi_k^{'}(w) &= J_k^T \, \nabla \, f(x_k + J_k \, w) \end{aligned}$$

If we approximate $\varphi_k(w)$ as

$$\varphi_k(w) \approx \psi_k(w) = \varphi_k(0) + \varphi_k^{'}(0)^T \, w + \frac{1}{2} w^T \, w$$

then,

$$\begin{aligned} \psi_k^{'}(w) &= \varphi_k^{'}(0) + w \\ &= J_k^T \, \nabla \, f(x_k) + w \end{aligned}$$

and $\psi_k(w)$ is minimized if

$$w = -J_k^T \, \nabla \, f(x_k)$$

Set

$$\bar{v}_k = -\varphi_k^{'}(0) = -J_k^T \, \nabla \, f(x_k).$$

We would like

A.

$$\begin{aligned} S_{k+1}(-\bar{v}_k) &= x_k, \qquad \textbf{that is} \\ x_{k+1} - J_{k+1} \, \bar{v}_k &= x_k \\ J_{k+1} \, \bar{v}_k &= x_{k+1} - x_k \end{aligned}$$

B.

$$\begin{aligned} \varphi_{k+1}^{'}(-\bar{v}_k) &= \psi_{k+1}^{'}(-\bar{v}_k), \qquad \textbf{that is} \\ J_{k+1}^T \, f^{'}(x_{k+1} - J_{k+1} \, \bar{v}_k) &= J_{k+1}^T \, f^{'}(x_{k+1}) - \bar{v}_k \\ \textbf{using (A)} & \\ J_{k+1}^T \, f^{'}(x_k) &= J_{k+1}^T \, f^{'}(x_{k+1}) - \bar{v}_k \\ \textbf{and} & \\ J_{k+1}^T \left(f^{'}(x_{k+1}) - f^{'}(x_k) \right) &= \bar{v}_k \end{aligned}$$

Henceforth we write s_k for $x_{k+1} - x_k$ and y_k for $f^{'}(X_{k+1}) - f^{'}(x_k)$. From (A) and (B) above we have

$$\begin{aligned} J_{k+1} \, \bar{v}_k &= s_k \\ J_{k+1}^T \, y_k &= v_k \\ s_k^T \, y_k &= \bar{v}_k^T \, J_{k+1}^T \, y_k = \bar{v}_k^T \, v_k \\ J_{k+1} \, J_{k+1}^T \, y_k &= s_k \end{aligned}$$

From (B) we have

$$J_{k+1}^T \left(f^{'}(x_{k+1}) - f^{'}(x_k)\right) = \bar{v}_k,$$

Then,

$$J_{k+1}\; J_{k+1}^T \left(f^{'}(x_{k+1}) - f^{'}(x_k)\right) = \bar{v}_k = x_{k+1} - x_k$$

Thus, we expect $J_{k+1}\; J_{k+1}^T$ to be an approximation to the inverse of the Hessian of f at x_k, that is $[\nabla^2\; f(x_k)]^{-1}$.

In Newton's method the descent direction d_k satisfies the equation

$$\begin{aligned} \nabla^2\; f(x_k)\; d_k &= -\nabla\; f(x_k) \\ d_k &= -[\nabla^2\; f(x_k)]^{-1}\; \nabla\; f(x_k) \end{aligned}$$

If we approximate $[\nabla^2\; f(x_k)]^{-1}$ by $J_k\; J_k^T$ we have

$$d_k = -J_k\; J_k^T\; \nabla\; f(x_k)$$

and

$$d_k^T\; \nabla\; f(x_k) = -\nabla\; f(x_k)^T\; J_k\; J_k^T\; \nabla\; f(x_k) < 0 \quad \textbf{if} \quad \nabla\; f(x_k) \neq 0.$$

Factored Quasi-Newton Algorithm

x_0 **initial guess**

$J_0 = I$

$d_0 = -J_0\; J_0^T\; \nabla\; f(x_0) = -\nabla\; f(x_0)$

$x_1 = x_0 + \lambda_0\; d_0$

where λ_0 is determined by a line search, for example, Goldstein-Armijo

For $k = 1, 2, \cdots$ **(until convergence) do**

$d_k = -J_k\; J_k^T\; \nabla\; f(x_k)$

$x_{k+1} = x_k + \lambda_k\; d_k$ **(Determine λ_k by a line search)**

$s_k = \lambda_k\; d_k$

$y_k = \nabla\; f(x_{k+1}) - \nabla\; f(x_k)$

Choose $\bar{v}_k$ in $\mathbb{R}^n$ so that $\bar{v}_k^T\; \bar{v}_k = s_k^T y_k$

Choose J_{k+1} so that

$$\begin{aligned} J_{k+1}\; \bar{v}_k &= s_k \\ J_{k+1}^T\; y_k &= \bar{v}_k \end{aligned}$$

End do.

Lemma 5.3.1 ***Let $s, y \in \mathbb{R}^n$ with $s^T\; y > 0$. Then, there is a symmetric positive definite $n \times n$-matrix H_+ satisfying $H_+\; y = s$ if and only if there is a vector $\bar{v}$ in $\mathbb{R}^n$, and a nonsingular matrix J_+ such that $J_+\; v = s$, $J_+\; y = \bar{v}$ and $\bar{v}^T\; \bar{v} = s^T\; y$.***

Proof **:** Suppose there exists a nonsingular $n \times n$-matrix J_+ such that $J_+\ v = s$ and $J_+^T\ y = \bar{v}$. Then, $J_+\ J_+^T\ y = s$. Since $J_+\ J_+^T$ is positive definite we can set $H_+ = J_+\ J_+^T$. Conversely, suppose that there exists a symmetric positive definite $n \times n$-matrix H_+ so that $H_+\ y = s$. Then, $J_+\ J_+^T\ y = s$ where J_+ is the Cholesky factor of H_+. Letting $\bar{v} = J_+^T\ y$ we see that $J_+\ \bar{v} = s$. Furthermore, $s^T\ y = \bar{v}^T\ J_+^T\ y = \bar{v}^T\ \bar{v} > 0$. □

An important factored class of quasi-Newton updates is given by

$$J_{k+1} = J_k + \frac{(s_k - J_k\ \bar{v}_k)(\bar{v}_k - J_k^T\ y_k)}{\bar{v}_k\ - J_k^T\ y_k)^T\ \bar{v}_k}$$

where $\bar{v}_k$ is chosen so that $\bar{v}_k^T\ \bar{v}_k = s_k^T\ y_k$ and $\left(\bar{v}_k\ - J_k^T\ y_k\right)^T\ \bar{v}_k \neq 0$.

If we set

$$\bar{v}_k = \sqrt{\frac{s_k^T\ y_k}{y_k^T\ J_k\ J_k^T\ y_k}}\ J_k^T\ y_k$$

we get the DFP (Davidon(1957) - Fletcher and Powell (1963)) update.

If we set

$$\bar{v}_k = \sqrt{\frac{y_k^T\ s_k}{s_k^T\ L_k\ L_k^T\ s_k}}\ L_k^T\ s_k$$

we get the BFGS (Broyden-Fletcher-Goldfarb-Shano) update.

Remark 5.3.1 *In the update formula it is easily verified that $J_{k+1}^T\ y_k = \bar{v}_k$, $J_{k+1}\ \bar{v}_k = s_k$ and that J_{k+1} is positive definite if J_k is.*

The unfactored DFP update is given by

$$H_{k+1} = H_k + \frac{s_k\ s_k^T}{s_k^T\ y_k} - \frac{H_k\ y_k\ y_k^T\ H_k}{y_k^T\ H_k\ y_k}$$

The unfactored BFGS update is given by

$$H_{k+1} = H_k + \frac{(s_k - H_k\ y_k)\ s_k^T + s_k\ (s_k - H_k\ y_k)^T}{y_k^T\ s_k} - \frac{(s_k - H - k\ y_k)^T\ y_k\ s_k\ s_k^T}{(y_k^T\ s_k)^2}$$

Remark 5.3.2 *In the unfactored update we can verify that $H_{k+1}\ y_k = s_k$ and that H_{k+1} is positive definite.*

Theorem 5.3.1 *Let Q be an $n \times n$-symmetric positive definite matrix. Consider the problem of minimizing the function $f(x) = \frac{1}{2}x^T\ Q\ x + c^T\ x$. Suppose the problem is solved using the DFP updates starting with an initial point x_1 and a symmetric positive definite matrix H_1. Suppose the step length λ_k are determined by minimizing the function $\lambda \longrightarrow f(x_k + \lambda\ d_k)$ and $\nabla\ f(x_j) \neq 0$ for each j, then the descent directions $d_1, \cdots, d_n$ are Q-conjugate and $H_{n+1} = Q^{-1}$. Furthermore, x_{n+1} is an optimal solution to this problem.*

Proof **:** For $1 \leq j \leq n$ we must show

(1) $H_{j+1}\ Q\ s_k = s_k, \quad 1 \leq k \leq j$

(2) $d_i^T\ Q\ d_k = 0, \quad i \neq k, i \leq j, k \leq j$

(3) $d_1, d_2, \cdots, d_j$ are linearly independent.

$$Q\ s_1 = Q(\lambda_1\ d_1) = Q(x_2 - x_1) = Q(x_2) + c - (q(x_1) + c) = \nabla\ f(x_2) - \nabla\ f(x_1) = y_1$$

$$H_2 = H_1 + \frac{s_1\ s_1^T}{s_1^T\ y_1} - \frac{H_1\ y_1\ y_1^T\ H_1}{y_1^T\ H_1\ y_1}$$

Therefore

$$H_2\ Q\ s_1 = \left(H_1 + \frac{s_1\ s_1^T}{s_1^T\ y_1} - \frac{H_1\ y_1\ y_1^T\ H_1}{y_1 6T\ H_1\ y_1}\right)\ y_1 = s_1.$$

Thus, (1) is proved if $j = 1$**.**

Next,

$$\lambda_1\ H_2\ Q\ d_1 = H_2\ Q\ \lambda_1\ d_1 = H_2\ Q\ s_1 = s_1 = \lambda_1\ d_1$$

Therefore

$$H_2\ Q\ d_1 = d_1.$$

Now,

$$\begin{aligned} d_1^T\ Q\ d_2 &= -d_1^T\ Q\ H_2\ \nabla\ f(x_2) \\ &= -\nabla\ f(x_2)^T\ H_2\ Q\ d_1 \\ &= -\nabla\ f(x_2)^T\ d_1 \end{aligned}$$

From section 3.2.3.2 $\nabla\ f(x_2)^T\ d_1 = 0$**. Thus,** $d_1^T\ Q\ d_2 = 0$ **and (2) is satisfied if** $j = 2$**.**

Suppose $j \geq 2$ **and we have verified that**

(i) $H_j\ Q\ s_i = s_i$**,** $\quad i < j$

(ii) $d_i^T\ Q\ d_j = 0$**,** $\quad i < j$

We will show that

(i) $H_{j+1}\ Q\ s_i = s_i$**,** $\quad i < j+1$

(ii) $d_i^T\ Q\ d_{j+1} = 0$**,** $\quad i < j+1$

We have

$$Q\ s_j = Q(\lambda_j\ d_j) = Q(x_{j+1} - x_j) = \nabla\ f(x_{j+1}) - \nabla\ f(x_j) = y_j$$

using the update formula

$$H_{j+1}\ Q\ s_j = \left(H_j + \frac{s_j\ s_j^T}{s_j^T\ y_j} - \frac{H_j\ y_j\ y_j^T\ H_j}{y_j^T\ H_j\ y_j}\right)\ y_j = s_j.$$

Suppose $i < j$**. Then,**

$$\begin{aligned} H_{j+1}\ Q\ s_i &= \left(H_j + \frac{s_j\ s_j^T}{s_j^T\ y_j} - \frac{H_j\ y_j\ y_j^T\ H_j}{y_j^T\ H_j\ y_j}\right)\ Q\ s_i = s_j \\ &= H_j\ Q\sigma_i + \frac{s_j\ s_j^T\ Q\sigma_i}{s_j^T\ y_j} - \frac{H_j\ y_j\ y_j^T\ H_j\ Q\sigma_i}{y_j^T\ H_j\ y_j} \end{aligned}$$

$$s_j^T\ Q\ s_i = \lambda_j\ \lambda_i\ d_j^T\ d_i = \lambda_j\ \lambda_i \cdot 0 = 0$$

$$y_j^T\ H_j\ Q\ s_i = y_j^T\ s_i = (Q\ s_j)^T\ s_i = s_j^T\ Q\ s_i = \lambda_j\ \lambda_i\ d_j^T\ Q\ d_i = \lambda_j\ \lambda_i\ \cdot\ 0 = 0.$$

Thus,

$$H_{j+1}\ Q\ si = H_j\ Q\ s_i, \quad i < j.$$

By the induction hypothesis $H_j\ Q\ si = s_i, \quad i < j$**.**

Thus,

$$H_{j+1}\ Q\ si = s_i, \quad i < j.$$

We have already shown above that $H_{j+1}\ Q\ s_j = s_j$**.**

Thus,

$$H_{j+1}\ Q\ s_i = s_i, \quad i < j+1$$

and the induction is complete in this case.

Next, we show that

$$d_i^T\ Q\ d_{j+1} = 0, \quad i < j+1.$$

We have

$$d_i^T\ Q\ d_{j+1} = -d_i^T\ Q\ H_{j+1}\ \nabla\ f(x_{j+1})$$

We just showed that

$$H_{j+1}\ Q\ s_i = s_i, \quad i < j+1.$$

Thus, since $s_i = \lambda_i\ d_i$**,**

$$\begin{aligned} H_{j+1}\ Q\ \lambda_i\ d_i &= \alpha_i\ d_i \\ H_{j+1}\ Q\ d_i &= d_i \\ d_i^T\ Q\ H_{j+1} &= d_i^T \end{aligned}$$

Now,

$$\begin{aligned} d_i^T\ Q\ d_{j+1} &= -d_i^T\ Q\ H_{j+1}\ \nabla\ f(x_{j+1}) \\ &= -d_i^T\ \nabla\ f(x_{j+1}) \\ &= 0 \quad (\textbf{See Section 3.2.3.2}) \end{aligned}$$

Finally, suppose $d_1, d_2, \cdots, d_j$**,** $j < n$ **are linearly independent. Then,** $d_1, d_2, \cdots, d_{j+1}$ **are also linearly independent.**
Suppose $\alpha_1\ d_1 + \alpha_2\ d_2 + \cdots + \alpha_{j+1}\ d_{j+1} = 0$**. We will show that** $\alpha_1 = \alpha_2 = \cdots = \alpha_{j+1} = 0$**.**

$$d_{j+1}\ Q\ (\alpha_1\ d_1 + \alpha_2\ d_2 + \cdots + \alpha_{j+1}\ d_{j+1}) = 0$$

Since $d_{j+1}^T\ Q\ d_i = 0, \quad i < j+1$ **it follows that**

$$\alpha_{j+1}\ d_{j+1}^T\ Q\ d_{j+1} = 0$$

Therefore $\alpha_{j+1} = 0$**.**
The fact that $d_1, d_2, \cdots, d_j$ **are linearly independent gives** $\alpha_1 = \cdots = \alpha_j = 0$**.**

Finally if the algorithm does not terminate before $j = n$**, then** $d_1, d_2, \cdots, d_n$ **are conjugate with respect to** Q**, and by the expanding subspace problem** $x_{n+1} = x^*$**, the optimal solution.**

If we define D **the matrix whose columns are** $d_1, d_2, \cdots, d_n$ **then** D **is nonsingular as** $d_1, d_2, \cdots, d_n$ **are linearly independent, and from (1) of the theorem**

$$H_{n+1}\ Q\ d_i = d_i, \qquad i < n+1$$

Thus,

$$\begin{aligned} H_{n+1}\ Q\ D &= D \\ H_{n+1}\ Q &= I \\ H_{n+1} &= Q^{-1} \end{aligned}$$

□

Theorem 5.3.2 (Powell - 1971) ***Let*** $f : \mathbb{R}^n \longrightarrow \mathbb{R}$***,*** $f \in C^{(2)}$ ***be a uniformly convex function (***$\|\nabla^2\ f(x)\| \geq \varepsilon > 0$***). Suppose the algorithm below is applied to minimize*** f***. Then*** $\{x_k\}$ ***converges to*** x^****, the minimizer of*** f***. Moreover, if on the set***

$$L(x_1) = \{x : f(x) \leq f(x_1)\}$$

the Lipschitz condition $\|f''(x) - f''(y) \leq L\ \|x - y\|$ ***is satisfied, then*** $x_k \longrightarrow x^*$ ***q-superlinearly.***

For $k = 1, 2, \cdots$ ***(while*** $\nabla\ f(x_k) \neq 0$***) do***

$d_k = -H_k \nabla f(x_k)$

$\lambda_k = -\frac{d_k^T \nabla f(x_k)}{d_k^T H_k d_k}$

$s_k = \lambda_k d_k$

$x_{k+1} = x_k + \lambda_k d_k$

$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}$$

End do

5.4 The Simplex Method/Nelder Mead Method

The basic idea in this method is to compare the values of the objective function at the $n+1$ points of an initially chosen simplex and creating a succession of new simplexes gradually moving towards the optimum value of the objective function. The methd was originally given by Spendley, Hext, and Himsworth (1962) and later developed by Nelder and Mead (1965).

The simplex method consists of three basic operations on the simplex as we move from one simplex to the next.

5.4.1 Reflection

If x_h is the vertex corresponding to the highest value of the objective function among the vertices of a simplex, we may expect the point x_r obtained by reflecting the point x_h in the opposite face to have the smallest value (see Figure 2.5.1). Mathematically,

$$x_r = (1+\alpha)\, x_0 - \alpha\, x_h, \quad \alpha > 0$$

where α is the distance between x_r and x_0 divided by the distance between x_h and x_0.

$$\begin{aligned} f(x_h) &= \max_{1 \le i \le n+1} f(x_i) \\ x_0 &= \frac{1}{n} \sum_{i=1\ i \ne h}^{n+1} x_i \end{aligned}$$

If

$$f(x_\ell) = \min_{1 \le i \le n+1} f(x_i) < f(x_r) < f(x_h),$$

then replace x_h by x_r and we have a new simplex.

In the case $f(x_r) = f(x_h)$ (Figure 3.2.5.3) we may remove the second worst vertex, or deform our simplex, or whatever else is appropriate.

5.4.2 Expansion

If a reflection gives a new point x_r where $f(x_r) < f(x_\ell)$, $f(x_\ell) = \min\{f(x_i) : 1 \le i \le n+1\}$, then the reflection has produced a new minimum. One expects further reduction by pushing x_r further to get a new point x_e (see Figure 2.5.4). That is,

$$x_e = \gamma\, x_r + (1-\gamma)\, x_0,$$

where γ is the expansion coefficient defined as

$$\gamma = \frac{\textit{distance between } x_e \textit{ and } x_0}{\textit{distance between } x_r \textit{ and } x_0} > 1.$$

In practice one may take $\gamma = 2$. If $f(x_e) < f(x_\ell)$, then replace x_h by x_e, and restart the process of reflection.

If, on the other hand, $f(x_e) > f(x_\ell)$, then the expansion is not successful and replace x_h by x_r, and start the reflection process.

5.4.3 Contraction

If the reflection process gives a point x_r for which $f(x_r) > f(x_i)$ for all $i \ne h$ and $f(x_r) > f(x_h)$, then replace x_h by x_r. In this case we contract the simplex as follows:

$$x_c = \beta\, x_h + (1-\beta)\, x_0 \qquad 0 \le \beta \le 1 \quad (\textbf{contraction coefficient})$$

$$\beta = \frac{\textit{distance between } x_c \textit{ and } x_0}{\textit{distance between } x_h \textit{ and } x_0}$$

If $f(x_r) > f(x_h)$ do contraction as in Figure 5 and Figure 6 above. If the contraction produces a point x_c for which $f(x_c) < \min\ \{f(x_h), f(x_r)\}$ replace $\{x_1, x_2, \cdots, x_{n+1}\}$ by the simplex $\{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{n+1}\}$ where x_h is replaced by x_c. Here contraction is prefered than reflection. If on the otherhand, $f(x_c) > \min\{f(x_h), f(x_r)\}$, then the above contraction process has not been fruitful, and replace x_i by $(x_i + x_\ell)/2$, where

$$f(x_\ell) = \min_{1 \le i \le n+1} f(x_i).$$

and start over the reflection process.

The method is declared to have converged if

$$Q = \left\{ \sum_{i=1}^{n+1} \frac{(f(x_i) - f(x_0))^2}{n+1} \right\}^{\frac{1}{2}} \le \varepsilon.$$

Example 5.4.1

$$\min\ f(x_1, x_2) = x_1 - x_2 + 2\, x_1^2 + 2\, x_1\, x_2$$

Take the points defining the initial simplex as

$$x_1 = \begin{pmatrix} 4.0 \\ 4.0 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 5.0 \\ 4.0 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 4.0 \\ 5.0 \end{pmatrix}, \quad \alpha = 2, \quad \beta = \frac{1}{2}, \quad \gamma = 2.$$

For convergence, take $\varepsilon = 0.2$.

<u>**Iteration 1**</u>

$$\begin{aligned} f(x_1) &= 80.0 \\ f(x_2) &= 107.0 \\ f(x_3) &= 96.0 \end{aligned}$$

Therefore

$$x_h = \begin{pmatrix} 5.0 \\ 4.0 \end{pmatrix}, \quad x_\ell = x_1 = \begin{pmatrix} 4.0 \\ 4.0 \end{pmatrix}, \quad f(x_\ell) = 80.$$

Centroid

$$x_0 = \frac{1}{2}(x_1 + x_3) = \begin{pmatrix} 4.0 \\ 4.5 \end{pmatrix}, \quad f(x_0) = 87.75$$

Reflection point x_r:

$$x_r = 2\,x_0 - x_h = \begin{pmatrix} 3.0 \\ 5.0 \end{pmatrix}$$

$$f(x_r) = 71.0$$

Since $f(x_r) < f(x_\ell)$, do expansion.

$$x_e = 2\,x_r - x_0 = \begin{pmatrix} 2.0 \\ 5.5 \end{pmatrix}$$

$$f(x_e) = 56.75 < f(x_\ell) = 80$$

Expansion is successful. Therefore replace x_h by x_e New simplex is $\{x_1, x_e, x_3\}$. In the new simplex $x_2 = x_e$.
Test convergence

$$Q = \left\{ \frac{(80 - 87.75)^2 + (56.75 - 87.75)^2 + (96 - 87.75)^2}{3} \right\}^{\frac{1}{2}} = 19.06 > \varepsilon = 0.02.$$

Go to the next iteration.

Iteration 2

$$\begin{aligned} f(x_1) &= 80.0 \\ f(x_2) &= f(x_e) = 56.75 \\ f(x_3) &= 96.0 \end{aligned}$$

Therefore

$$x_h = x_3, \quad x_\ell = x_2 = \begin{pmatrix} 2.0 \\ 5.5 \end{pmatrix}$$

Centroid

$$x_0 = \frac{1}{2}(x_1 + x_2) = \begin{pmatrix} 3.0 \\ 4.75 \end{pmatrix}, \quad f(x_0) = 67.31$$

Reflection point x_r:

$$x_r = 2\,x_0 - x_h = \begin{pmatrix} 2.0 \\ 4.5 \end{pmatrix}$$

$$f(x_r) = 43.75$$

Since $f(x_r) < f(x_\ell)$, do expansion.

$$x_e = 2\,x_r - x_0 = \begin{pmatrix} 1.0 \\ 4.25 \end{pmatrix}$$

$$f(x_e) = 25.3125 < f(x_\ell) = 80$$

Expansion is successful. Therefore replace x_h by x_e New simplex is $\{x_1, x_2, x_e\}$. In the new simplex $x_3 = x_e$.

$$x_1 = \begin{pmatrix} 4.0 \\ 4.0 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 2.0 \\ 5.5 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 1.0 \\ 4.25 \end{pmatrix}$$

Testing for convergence, in this case too convergence criterion is not met. ($Q = 26.1$).

<u>Iteration 3</u>

Start with

$$x_1 = \begin{pmatrix} 4.0 \\ 4.0 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 2.0 \\ 5.5 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 1.0 \\ 4.25 \end{pmatrix}$$

and do the reflection process. Again expansion is possible leading to a new simplex

$$x_1 = \begin{pmatrix} -3.5 \\ 6.625 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 2.0 \\ 5.5 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 1.0 \\ 4.25 \end{pmatrix}$$

Do reflection process on

$$x_1 = \begin{pmatrix} -3.5 \\ 6.625 \end{pmatrix}, \quad x_2 = \begin{pmatrix} 2.0 \\ 5.5 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 1.0 \\ 4.25 \end{pmatrix}$$

Obtain $x_r = 2\,x_h - x_0$, $f(x_r) < f(x_\ell)$
Expand $x_e = 2\,x_r - x_0$.
However, $f(x_e) > f(x_\ell)$. Expansion failed.
Stick with

$$x_r = \begin{pmatrix} -4.5 \\ 5.375 \end{pmatrix}$$

New simplex

$$x_1 = \begin{pmatrix} -3.5 \\ 6.625 \end{pmatrix}, \quad x_2 = x_r = \begin{pmatrix} -4.5 \\ 5.375 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 1.0 \\ 4.25 \end{pmatrix}$$

Convergence criterion still not met ($Q = 7.5$). Do more iteration.

<u>Iteration 5</u>

$$x_1 = \begin{pmatrix} -3.5 \\ 6.625 \end{pmatrix}, \quad x_2 = x_r = \begin{pmatrix} -4.5 \\ 5.375 \end{pmatrix}, \quad x_3 = \begin{pmatrix} 1.0 \\ 4.25 \end{pmatrix}$$

$$f(x_1) = 11.89, \quad f(x_2) = 11.14, f(x_3) = 25.3125$$

Centroid

$$x_0 = \frac{1}{2}(x_1 + x_2) = \begin{pmatrix} -4.0 \\ 6.0 \end{pmatrix}, \quad f(x_0) = 10.0.$$

$$x_r = 2\, x_0 - x_h = \begin{pmatrix} -9.0 \\ 7.75 \end{pmatrix}, \quad f(x_r) = 65.8125$$

$$f(x_r) > f(x_\ell) = f(x_2) = 11.14$$

Also,

$$f(x_r) > f(x_h) = f(x_3) = 25.3125$$

Therefore do contraction

$$x_c = \frac{1}{2}x_h + \frac{1}{2}x_0 = \begin{pmatrix} -1.5 \\ 5.125 \end{pmatrix}, \quad (\beta = \frac{1}{2}), \quad f(x_c) = 8.75$$

Since $f(x_c) < f(x_h)$ the new simplex is

$$x_1 = \begin{pmatrix} -3.5 \\ 6.625 \end{pmatrix}, \quad x_2 = \begin{pmatrix} -4.5 \\ 5.375 \end{pmatrix}, \quad x_3 = x_c = \begin{pmatrix} -1.5 \\ 5.125 \end{pmatrix}$$

Convergence criterion is still not met. $Q = 1.466$. However, we are getting closer to meeting the convergence criterion. We do more iteration. We can't do worse!.