Chapter 11

Numerical methods

Most numerical gradient-based methods require an initial design $\underset{\sim}{x}_0$, a search direction $\underset{\sim}{d}_0$, and a step size α_0. In this way, the improved design can be defined as $\underset{\sim}{x}_1 = \underset{\sim}{x}_0 + \alpha_0 \underset{\sim}{d}_0$. Iterativelly, these algorithms may be written as

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \alpha_k \underset{\sim}{d}_k. \tag{11.1}$$

Gradient-based numerical methos are characterized by the different ways they determine the search direction vector $\underset{\sim}{d})k$. The step size α_k is determined by one-dimensional optimization methods, also referred to as *line search methods*.

11.1 Steepest descent method

11.1.1 Definition

This method was presented by Cauchy in 1847. In the original definition, a current design $\underset{\sim}{x}_k$ is driven to an improved design $\underset{\sim}{x}_{k+1}$ such that

$$f(\underset{\sim}{x}_{k+1}) < f(\underset{\sim}{x}_k); \tag{11.2}$$

this is,

$$f(\underset{\sim}{x}_k + \alpha_k \underset{\sim}{d}_k) < f(\underset{\sim}{x}_k). \tag{11.3}$$

Using Taylor series expansion about $\underset{\sim}{x}_k$, the objective function f can be written as

$$f(\underset{\sim}{x}) = f(\underset{\sim}{x}_k) + \nabla f(\underset{\sim}{x}_k)^{\mathrm{T}} (\underset{\sim}{x} - \underset{\sim}{x}_k) + R_1(\underset{\sim}{x}). \tag{11.4}$$

Evaluating in $\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \alpha_k \underset{\sim}{d}_k$, one obtains that

$$f(\underset{\sim}{x}_{k+1}) = f(\underset{\sim}{x}_k) + \nabla f(\underset{\sim}{x}_k)^{\mathrm{T}} (\alpha_k \underset{\sim}{d}_k) + R_1(\alpha_k^2). \tag{11.5}$$

For α_k sufficiently small, the residual term can be disregarded. Since $f(\underset{\sim}{x}_{k+1}) - f(\underset{\sim}{x}_k) < 0$, see (11.2), one

observes that

$$\alpha_k \nabla f(\underset{\sim}{x}_k)^{\mathrm{T}} \underset{\sim}{d}_k < 0. \tag{11.6}$$

For any $\alpha_k > 0$,

$$\nabla f(\underset{\sim}{x}_k)^{\mathrm{T}} \underset{\sim}{d}_k < 0, \tag{11.7}$$

where $\underset{\sim}{d}_k$ is a *descent direction* and (11.7) is referred to as *descent condition*. If

$$\underset{\sim}{d}_k = -\nabla f(\underset{\sim}{x}_k), \tag{11.8}$$

then

$$\nabla f(\underset{\sim}{x}_k)^{\mathrm{T}} \underset{\sim}{d}_k = -||\nabla f(\underset{\sim}{x}_k)||^2 < 0. \tag{11.9}$$

The search direction in (11.8) is referred to as *steepest descent direction*.

11.1.2 Significance

In $\mathbb{R}^n$ the gradient $\nabla f(\underset{\sim}{x}_k)$ is a normal vector to the tangent hyperplane of the function isovalue at $\underset{\sim}{x}_k$. The gradient vector is aligned with the direction of maximum growth steepest ascent of the function f at $\underset{\sim}{x}_k$.

Example. Consider the function

$$f(\underset{\sim}{x}) = 2x_1^2 + x_2^2.$$

Determine if the search direction $\underset{\sim}{d}_0 = (0.5, 0.8)^{\mathrm{T}}$ is a descent direction in $\underset{\sim}{x}_0 = (1, 2)^{\mathrm{T}}$. Determine the steepest descent direction in this point.

The gradient of the function is

$$\nabla f(\underset{\sim}{x}) = \begin{pmatrix} 4x_1 \\ 2x_2 \end{pmatrix}.$$

Evaluating at $\underset{\sim}{x}_0 = (1, 2)^{\mathrm{T}}$ yields

$$\nabla f(\underset{\sim}{x}_0) = \begin{pmatrix} 4 \\ 4 \end{pmatrix};$$

therefore, $\underset{\sim}{d}_s = -(4, 4)^{\mathrm{T}}$ is the steepest descent direction of f at $\underset{\sim}{x}_0$. The descent condition for $\underset{\sim}{d}_0 = (0.5, 0.8)^{\mathrm{T}}$ can be verified with the sign of the directional derivative of f_0 along $\underset{\sim}{d}_0$. This is

$$\begin{pmatrix} 4 & 4 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.8 \end{pmatrix} = 5.2 \not< 0,$$

then, $\underset{\sim}{d}_0$ is not a descent direction. ∎

11.1.3 Line search

Once the search direction $\underset{\sim}{d}_k$ in the point $\underset{\sim}{x}_k$ is identified, the algorithm has to identify the step size α_k. This problem can be stated as,

$$\min_{\alpha} \qquad f(\alpha) = f(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k). \tag{11.10}$$

The necessary and sufficient condition is obtained from

$$\frac{\mathrm{d}f(\alpha)}{\mathrm{d}\alpha} = \frac{\mathrm{d}(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)}{\mathrm{d}\alpha} = 0. \tag{11.11}$$

This is

$$\frac{\mathrm{d}f(\underset{\sim}{x}_{k+1})}{\mathrm{d}\alpha} = \frac{\partial f(\underset{\sim}{x}_{k+1})}{\partial \underset{\sim}{x}} \frac{\mathrm{d}\underset{\sim}{x}_{k+1}}{\mathrm{d}\alpha} = \nabla f(\underset{\sim}{x}_{k+1})^{\mathrm{T}} \underset{\sim}{d}_k = 0. \tag{11.12}$$

In other words, the directional derivative of $f(\underset{\sim}{x}_{k+1})$ along $\underset{\sim}{d}_k$ must be zero. Using (11.8), one can observe that

$$\underset{\sim}{d}_{k+1} = -\nabla f(\underset{\sim}{x}_{k+1}), \tag{11.13}$$

therefore,

$$\frac{\mathrm{d}f(\underset{\sim}{x}_{k+1})}{\mathrm{d}\alpha} = -\underset{\sim}{d}_{k+1}{}^{\mathrm{T}} \underset{\sim}{d}_k = 0. \tag{11.14}$$

Interestingly, since two consecutive steepest descent directions are orthogonal, this method approaches the minimum in a "zig-zag" shape.

11.1.4 Convergence criteria

The necessary condition for optimality is satisfied in a particular point $\underset{\sim}{x}_k$ if

$$||\nabla f(\underset{\sim}{x}_k)|| \leq \varepsilon_G, \tag{11.15}$$

where ε_G is a tolerance on the gradient supplied by the user. Notice that the gradient can vanish at any stationary point (maximum, minimum, or saddle point). However, the chances to find a maximum or a saddle point with a steepest descent algorithm are remote.

The algorithm should also check successive reduction in the function value, for example,

$$|f(\underset{\sim}{x}_{k+1} - f(\underset{\sim}{x}_k)| \leq \varepsilon_A + \varepsilon_R |f(\underset{\sim}{x}_k)|, \tag{11.16}$$

where ε_A is the absolute tolerance on the change in the function value while ε_R is the relative tolerance. Suggested values might be $\varepsilon_G = 10^{-4}$, $\varepsilon_A = 10^{-6}$, and $\varepsilon_R = 10^{-2}$.

11.1.5 Algorithm

Step 1. Estimate a reasonable initial point $\underset{\sim}{x}_0$ and convergence parameters ε_A, ε_G, and ε_R, where ε_A is the absolute tolerance, ε_G and ε_R tolerances for the gradient and the function.

Step 2. Determine $\nabla f(\underset{\sim}{x}_k)$. Stop if

$$||\nabla f(\underset{\sim}{x}_k)|| \leq \varepsilon_G.$$

Otherwise, define the search direction $\underset{\sim}{d}_k = -\nabla f(\underset{\sim}{x}_k)$.

Step 3. Obtain α_k minimizing $f(\alpha) = f(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)$, for $\alpha > 0$. Update $\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \alpha_k \underset{\sim}{d}_k$.

Step 4. Evaluate $f(\underset{\sim}{x}_{k+1})$. Stop if

$$|f(\underset{\sim}{x}_{k+1}) - f(\underset{\sim}{x}_k)| \leq \varepsilon_A + \varepsilon_R |f(\underset{\sim}{x}_k)|$$

is satisfied in two consecutive iterations. Otherwise, update $k = k + 1$ and $\underset{\sim}{x}_k = \underset{\sim}{x}_{k+1}$ and go to Step 2.

Example. Minimize the function

$$f(\underset{\sim}{x}) = x_1^2 + x_2^2 - 2x_1 x_2$$

using the steepest descent method from the point $\underset{\sim}{x}_0 = (1, 0)^{\mathrm{T}}$.

The search direction is $\underset{\sim}{d}_0 = -(-2, 2)^{\mathrm{T}}$. The step size minimizes $f(\alpha) = 18\alpha^2 - 8\alpha + 1$, this is, $\alpha_0 = 0.25$. The improved design is $\underset{\sim}{x}_1 = (0.5, 0.5)^{\mathrm{T}}$. In this point, the gradient of the function is $\underset{\sim}{0}$ which satisfies the optimality condition and the convergence criterion. Therefore, this is the optimum point. ∎

The steepest descent method is simple and robust to optimization. However, it has some drawbacks.

1. Even though this method is convergent, it is also slow even with positive definite quadratic forms.
2. The information from previous iterations is not used in the next steps.
3. In practice, the function is greatly reduced in the first iterations but it decreases mores slowly as the iterations continue.
4. The steepest descent direction makes sense from a local prospective (current point) but it can be improved in a more global sense.

11.1.6 Scaling

The order of convergence in this method is close to linear, even with some quadratic functions. That means

$$\lim_{k \to \infty} \frac{||\underset{\sim}{x}_{k+1} - \underset{\sim}{x}^*||}{||\underset{\sim}{x}_k - \underset{\sim}{x}^*||^p} < \infty, \tag{11.17}$$

where $p \to 1$. Using scaling techniques on the design variables it is possible to improve the order of convergence. A quadratic function can be scaled such that its Hessian is the identity matrix. In this case, the algorithm find the minimum in one iteration and the order of convergence is ∞.

Example. Minimize the function

$$f(\underset{\sim}{x}) = 25x_1^2 + x_2^2$$

using the steepest descent method form the point $\underset{\sim}{x}_0 = (1, 1)^{\mathrm{T}}$.

The gradient and the steepest descent direction is determined in every iteration. The line search is performed using a one-dimensional technique (e.g., golden section, polynomial approximations). Using the function `fminbnd` in MATLAB, this can be done as

```
function [xopt,fopt,aopt,ngradf]=afun(x0)
d0=-gfun('fun',x0);
[aopt,fopt] = fminbnd(@(a) fun(x0,d0,a),0,10);
xopt=x0+d0*aopt;
ngradf=norm(d0);

function f=fun(x0,d0,a)
if nargin==1
    d0=zeros(size(x0));
    a=0;
end
x=x0+a*d0;
f=25*x(1)^2+x(2)^2;
```

On the work space,

```
>> [x,f,a,n]=afun([1;1])
x =
   -0.0010
    0.9600
f =
    0.9215
a =
    0.0200
n =
   50.0400
```

Table 11.1 shows the iterative process.

Analytically, the Hessian is given by

$$\nabla^2 f(\underset{\sim}{x}) = \begin{pmatrix} 50 & 0 \\ 0 & 2 \end{pmatrix}.$$

Table 11.1: Steepest descent, $f(x) = 25x_1^2 + x_2^2$

k	x_1	x_2	$f(\underset{\sim}{x})$	α	$\|\|\nabla f(\underset{\sim}{x})\|\|$
001	+1.0000E+00	+1.0000E+00	+2.6000E+01	+2.0033E-02	+5.0040E+01
002	-1.6444E-03	+9.5992E-01	+9.2154E-02	+4.7898E-02	+1.9216E+00
$\vdots$					
111	-2.3545E-06	+1.3752E-03	+1.8916E-02		+2.7531E-03

Introducing new variables $\hat{\underset{\sim}{x}} = (\hat{x}_1, \hat{x}_2)^{\mathrm{T}}$ such that

$$\underset{\sim}{x} = \underset{\approx}{D}\hat{\underset{\sim}{x}}, \tag{11.18}$$

where

$$\underset{\approx}{D} = \begin{pmatrix} \frac{1}{\sqrt{50}} & 0 \\ 0 & \frac{1}{\sqrt{2}} \end{pmatrix},$$

then $x_1 = \hat{x}_1/\sqrt{50}$, $x_2 = \hat{x}_2/\sqrt{2}$ and

$$f(\hat{\underset{\sim}{x}}) = \frac{1}{2}(\hat{x}_1^2 + \hat{x}_2^2).$$

The minimum of $f(\hat{\underset{\sim}{x}})$ can be located in one iteration using the steepest descent method. The point that minimizes f is $\hat{\underset{\sim}{x}}^* = (0, 0)^{\mathrm{T}}$ in the scaled space. Then,

$$x_1^* = \hat{x}_1^*/\sqrt{50} = 0$$

and

$$x_2^* = \hat{x}_2^*/\sqrt{2} = 0.$$

∎

Example. Minimize the function

$$f(\underset{\sim}{x}) = 6x_1^2 - 6x_1x_2 + 2x_2^2 - 5x_1 + 4x_2 + 2$$

using the steepest descent method from the point $\underset{\sim}{x}_0 = (-1, -2)^{\mathrm{T}}$. Perform an appropriate scaling to improve the convergence rate of this method.

The Hessian of the function is given by

$$\nabla^2 f(\underset{\sim}{x}) = \begin{pmatrix} 12 & -6 \\ -6 & 4 \end{pmatrix}.$$

The eigenvalues are $\lambda_1 = 0.7889$ and $\lambda_2 = 15.211$ with eigenvectors $\underset{\sim}{e}_1 = (0.4718, 0.8817)^{\mathrm{T}}$ and $\underset{\sim}{e}_2 =$

$(-0.8817, 0.4718)^{\mathrm{T}}$. The scaling is defined as

$$\underset{\sim}{x} = \underset{\approx}{Q}\hat{\underset{\sim}{x}}, \tag{11.19}$$

where $\underset{\approx}{Q} = (\underset{\sim}{e}_1, \underset{\sim}{e}_2)$, this is

$$\underset{\approx}{Q} = \begin{pmatrix} 0.4718 & -0.8817 \\ 0.8817 & 0.4718 \end{pmatrix}.$$

Note that the Hessian of the scaled function is not the identity matrix yet. To accomplish this condition one needs a new change of variables

$$\hat{\underset{\sim}{x}} = \underset{\approx}{D}\hat{\hat{\underset{\sim}{x}}}, \tag{11.20}$$

where

$$\underset{\approx}{D} = \begin{pmatrix} \frac{1}{\sqrt{0.7889}} & 0 \\ 0 & \frac{1}{\sqrt{15.211}} \end{pmatrix}.$$

The Hessian of the new function $\hat{\hat{\underset{\sim}{x}}}$ is, in fact, the identity matrix. Using the steepest descent method one reaches the optimum in one single iteration. The minimum is $\hat{\hat{\underset{\sim}{x}}}^* = (-1.3158, -1.6142)^{\mathrm{T}}$. Applying the linear transformations, the minimum in the original space is

$$\underset{\sim}{x} = \underset{\approx}{Q}\underset{\approx}{D}\hat{\hat{\underset{\sim}{x}}}, \tag{11.21}$$

and the solution is $\underset{\sim}{x}^* = (-1/3, -2/3)^{\mathrm{T}}$. ∎

11.2 Conjugate gradient method

11.2.1 Definition

The conjugate gradient method was presented by Fletcher & Reeves (1964) as a quadratically convergent gradient method for locating an unconstrained local minimum of a function of several variables. With this method it is possible to locate the minimum of a quadratic function of n variables in n iterations. Consider the problem of minimizing a quadratic function

$$\min_{\underset{\sim}{x}} \qquad f(\underset{\sim}{x}) = \tfrac{1}{2}\underset{\sim}{x}^{\mathrm{T}}\underset{\approx}{A}\underset{\sim}{x} + \underset{\sim}{c}^{\mathrm{T}}\underset{\sim}{x} + b. \tag{11.22}$$

Assume that $\underset{\approx}{A}$ is symmetric ($\underset{\approx}{A}^{\mathrm{T}} = \underset{\approx}{A}$) and positive definite ($\underset{\sim}{x}^{\mathrm{T}}\underset{\approx}{A}\underset{\sim}{x} > 0$, $\forall \underset{\sim}{x} \in \mathbb{R}^n, \underset{\sim}{x} \neq \underset{\sim}{0}$). Then, two search directions, $\underset{\sim}{d}_i$ and $\underset{\sim}{d}_j$, are conjugate with respect to $\underset{\approx}{A}$ if

$$\underset{\sim}{d}_i{}^{\mathrm{T}}\underset{\approx}{A}\underset{\sim}{d}_j = 0, \quad i \neq j. \tag{11.23}$$

Consider an initial design $\underset{\sim}{x}_0$ and a set of conjugate directions $\underset{\sim}{d}_0, \underset{\sim}{d}_1 \ldots, \underset{\sim}{d}_{n-1}$. Minimizing $f(\underset{\sim}{x})$ along $\underset{\sim}{d}_0$ one obtains $\underset{\sim}{x}_1$. From the point $\underset{\sim}{x}_1$, $f(\underset{\sim}{x})$ is minimized along $\underset{\sim}{d}_1$ to obtain $\underset{\sim}{x}_2$. The process continues until one reaches $\underset{\sim}{x}_n$ along $\underset{\sim}{d}_{n-1}$. The point $\underset{\sim}{x}_n$ minimizes (11.22). Denoting

$$\nabla f_k = \nabla f(\underset{\sim}{x}_k), \tag{11.24}$$

then

$$\nabla f_k = \underset{\sim}{x}_k{}^{\mathrm{T}} \underset{\approx}{A} + \underset{\sim}{c}. \tag{11.25}$$

If the descent direction $\underset{\sim}{d}_k$ is known, then

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \alpha_k \underset{\sim}{d}_k, \tag{11.26}$$

where α_k is the solution that minimizes $f(\alpha)$. This is

$$\min_{\alpha} \qquad f(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k), \tag{11.27}$$

where

$$f(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k) = \frac{1}{2}(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)^{\mathrm{T}} \underset{\approx}{A} (\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k) + \underset{\sim}{c}^{\mathrm{T}} (\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k). \tag{11.28}$$

The derivative with respect to α can be expressed as

$$\frac{\mathrm{d}f(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)}{\mathrm{d}\alpha} = \frac{1}{2}(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k + \frac{1}{2}\left[\underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} (\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)\right]^{\mathrm{T}} + \underset{\sim}{c}^{\mathrm{T}} \underset{\sim}{d}_k \tag{11.29}$$

or

$$\frac{\mathrm{d}f(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)}{\mathrm{d}\alpha} = (\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k + \underset{\sim}{c}^{\mathrm{T}} \underset{\sim}{d}_k. \tag{11.30}$$

The necessary (and sufficient) condition for optimality is satisfied when (11.30) is equal to zero. This condition can be written as

$$\underset{\sim}{x}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k + \alpha \underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k + \underset{\sim}{c}^{\mathrm{T}} \underset{\sim}{d}_k = 0, \tag{11.31}$$

this is

$$\alpha \underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k = -\underset{\sim}{x}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k - \underset{\sim}{c}^{\mathrm{T}} \underset{\sim}{d}_k. \tag{11.32}$$

Comparing (11.32) with (11.25) and using the symmetry of $\underset{\approx}{A}$ yields

$$\alpha \underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k = -\underset{\sim}{d}_k{}^{\mathrm{T}} \nabla f_k. \tag{11.33}$$

Finally, solving for α allows

$$\alpha_k = -\frac{\underset{\sim}{d}_k{}^{\mathrm{T}} \nabla f_k}{\underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k}. \tag{11.34}$$

In this method the search direction $\underset{\sim}{d}_{k+1}$ has the form

$$\underset{\sim}{d}_{k+1} = -\nabla f_{k+1} + \beta_k \underset{\sim}{d}_k, \tag{11.35}$$

where $\beta_k \underset{\sim}{d}_k$ represents the deflection of the search direction with respect to the steepest descent direction. If two consecutive search directions are conjugate with respect to $\underset{\approx}{A}$ then

$$\underset{\sim}{d}_{k+1}{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k = 0. \tag{11.36}$$

Using (11.34), the conjugate condition (11.36) can be written as

$$(-\nabla f_{k+1} + \beta_k \underset{\sim}{d}_k)^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k = 0, \tag{11.37}$$

this is

$$-\nabla f_{k+1}{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k + \beta_k \underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k = 0. \tag{11.38}$$

Solving for β_k,

$$\beta_k = \frac{\nabla f_{k+1}{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k}{\underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k}. \tag{11.39}$$

From (11.26) one observes that

$$\underset{\sim}{d}_k = \frac{\underset{\sim}{x}_{k+1} - \underset{\sim}{x}_k}{\alpha_k}.$$

Multiplying by $\underset{\approx}{A}$ one obtains

$$\underset{\approx}{A} \underset{\sim}{d}_k = \frac{\underset{\approx}{A} \underset{\sim}{x}_{k+1} - \underset{\approx}{A} \underset{\sim}{x}_k}{\alpha_k}. \tag{11.40}$$

Using (11.25),

$$\nabla f_k = \underset{\approx}{A} \underset{\sim}{x}_k + \underset{\sim}{c}$$

and

$$\nabla f_{k+1} = \underset{\approx}{A} \underset{\sim}{x}_{k+1} + \underset{\sim}{c},$$

then (11.40) can be written as

$$\underset{\approx}{A} \underset{\sim}{d}_k = \frac{\nabla f_{k+1} - \nabla f_k}{\alpha_k}. \tag{11.41}$$

Replacing (11.41) into (11.39) yields

$$\beta_k = \frac{\nabla f_{k+1}{}^{\mathrm{T}} (\nabla f_{k+1} - \nabla f_k)}{\alpha_k \underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k}. \tag{11.42}$$

From (11.35) one observes that

$$\underset{\sim}{d}_k = -\nabla f_k + \beta_{k-1} \underset{\sim}{d}_{k-1}. \tag{11.43}$$

Multiplying (11.43) by ∇f_k,

$$\underset{\sim}{d}_k{}^{\mathrm{T}}\nabla f_k = -\nabla f_k{}^{\mathrm{T}}\nabla f_k + \beta_{k-1}\underset{\sim}{d}_{k-1}{}^{\mathrm{T}}\nabla f_k. \tag{11.44}$$

The condition $\mathrm{d}f/\mathrm{d}\alpha = 0$ make $\underset{\sim}{d}_k$ and ∇f_{k+1} orthogonal, this is

$$\underset{\sim}{d}_k{}^{\mathrm{T}}\nabla f_{k+1} = 0, \tag{11.45}$$

or

$$\underset{\sim}{d}_{k-1}{}^{\mathrm{T}}\nabla f_k = 0.$$

In this way, (11.44) can be expressed as

$$\underset{\sim}{d}_k{}^{\mathrm{T}}\nabla f_k = -\nabla f_k{}^{\mathrm{T}}\nabla f_k. \tag{11.46}$$

In consequence, (11.34) can be written as

$$\alpha_k = \frac{\nabla f_k{}^{\mathrm{T}}\nabla f_k}{\underset{\sim}{d}_k{}^{\mathrm{T}}\underset{\approx}{A}\underset{\sim}{d}_k}. \tag{11.47}$$

Substituting (11.47) into (11.39) yields

$$\beta_k = \frac{\nabla f_{k+1}{}^{\mathrm{T}}(\nabla f_{k+1} - \nabla f_k)}{\nabla f_k{}^{\mathrm{T}}\nabla f_k}. \tag{11.48}$$

This expression is used by the *Polak-Ribiere* algorithm. On the other hand, if one considers

$$\nabla f_{k+1}{}^{\mathrm{T}}\nabla f_k = 0, \tag{11.49}$$

then

$$\beta_k = \frac{\nabla f_{k+1}{}^{\mathrm{T}}\nabla f_{k+1}}{\nabla f_k{}^{\mathrm{T}}\nabla f_k}. \tag{11.50}$$

This expression is used by the *Fletcher-Reeves* algorithm.

11.2.2 Algorithm

Step 1. Estimate the initial design $\underset{\sim}{x}_0$. Select a convergence parameter ε. The first search direction is defined as the steepest descent direction,

$$\underset{\sim}{d}_0 = -\nabla f_0.$$

If $||\nabla f_0|| < \varepsilon$, then convergence is obtained and algorithm cannot improve the current design, otherwise go to Step 5.

Step 2. If $||\nabla f_k|| < \varepsilon$ then convergence is obtained and the algorithm cannot improve the current design, otherwise go to the next Step.

Step 3. Determine the deflection factor using Polak-Ribiere equation (11.48) or Fletcher-Reeves equation (11.50). In the second case, the deflection factor can be expressed as

$$\beta_k = \frac{\nabla f_{k+1}{}^{\mathrm{T}} \nabla f_{k+1}}{\nabla f_k{}^{\mathrm{T}} \nabla f_k}$$

Step 4. Update the search direction

$$\underset{\sim}{d}_{k+1} = -\nabla f_{k+1} + \beta_k \underset{\sim}{d}_k.$$

Step 5. Determine the step size by performing a line search. In a quadratic function, the step size is given by

$$\alpha_k = \frac{\nabla f_k{}^{\mathrm{T}} \nabla f_k}{\underset{\sim}{d}_k{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_k}.$$

If the function is not quadratic, α is determined by minimizing $f(\alpha)$.

Step 6. Update the design

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \alpha_k \underset{\sim}{d}_k$$

and go to Step 2.

If the minimum is not found after $n + 1$ iterations, it is recommended to restart the algorithm using the steepest descent direction.

Example. Consider the minimization of the function

$$f(\underset{\sim}{x}) = x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1x_2 + 2x_2x_3$$

from the initial design $\underset{\sim}{x}_0 = (2, 4, 10)^{\mathrm{T}}$. Using the conjugate gradient method, how many iterations would it take to find the minimum? Perform that many iterations and compare your numerical solution with the analytical solution.

Since this is a quadratic problem, we can use the Hessian to update the step size. The algorithm should converge in three iterations. The gradient of the function is given by

$$\nabla f(\underset{\sim}{x}) = \begin{pmatrix} 2x_1 + 2x_2 \\ 2x_1 + 4x_2 + 2x_3 \\ 2x_2 + 4x_3 \end{pmatrix}$$

and the Hessian is

$$\underset{\approx}{A} = \nabla^2 f(\underset{\sim}{x}) = \begin{pmatrix} 2 & 2 & 0 \\ 2 & 4 & 2 \\ 0 & 2 & 4 \end{pmatrix}$$

Iteration 1. The first improved point is $\underset{\sim}{x}_1 = \underset{\sim}{x}_0 + \alpha_0 \underset{\sim}{d}_0$ where

$$\underset{\sim}{d}_0 = -\nabla f_0 = \begin{pmatrix} -12 \\ -40 \\ -48 \end{pmatrix}$$

Since f is quadratic then the step size is

$$\alpha_0 = \frac{\nabla f_0{}^{\mathrm{T}} \nabla f_0}{\underset{\sim}{d}_0{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_0} = \frac{253}{1594} = 0.1587.$$

Then,

$$\underset{\sim}{x}_1 = \underset{\sim}{x}_0 + \alpha_0 \underset{\sim}{d}_0 = \begin{pmatrix} 0.0954 \\ -2.3488 \\ 2.3814 \end{pmatrix}$$

Iteration 2. The second improved point is $\underset{\sim}{x}_2 = \underset{\sim}{x}_1 + \alpha_1 \underset{\sim}{d}_1$ where

$$\underset{\sim}{d}_1 = -\nabla f_1 + \beta_0 \underset{\sim}{d}_0$$

Using Fletcher-Reeves equation

$$\beta_0 = \frac{\nabla f_1{}^{\mathrm{T}} \nabla f_1}{\nabla f_0{}^{\mathrm{T}} \nabla f_0},$$

where

$$\nabla f_1 = \begin{pmatrix} -4.5069 \\ -4.4417 \\ 4.8281 \end{pmatrix};$$

then

$$\beta_0 = 0.01565$$

and

$$\underset{\sim}{d}_1 = \begin{pmatrix} 4.3191 \\ 3.81566 \\ -5.5793 \end{pmatrix}.$$

Now, as before

$$\alpha_1 = \frac{\nabla f_1{}^{\mathrm{T}} \nabla f_1}{\underset{\sim}{d}_1{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_1} = 0.3155,$$

and

$$\underset{\sim}{x}_2 = \underset{\sim}{x}_1 + \alpha_1 \underset{\sim}{d}_1 = \begin{pmatrix} 1.4578 \\ -1.1452 \\ 0.6214 \end{pmatrix}$$

Iteration 3. The third (hopefully last) design is determine in the same way:

$$\nabla f_2 = \begin{pmatrix} 0.62534 \\ -0.422104 \\ 0.195419 \end{pmatrix},$$

$$\beta_1 = \frac{\nabla f_2{}^{\mathrm{T}} \nabla f_2}{\nabla f_1{}^{\mathrm{T}} \nabla f_1} = 0.0096,$$

$$\underset{\sim}{d}_2 = -\nabla f_2 + \beta_1 \underset{\sim}{d}_1 = \begin{pmatrix} -0.5839 \\ 0.4587 \\ -0.2489 \end{pmatrix},$$

$$\alpha_2 = \frac{\nabla f_2{}^{\mathrm{T}} \nabla f_2}{\underset{\sim}{d}_2{}^{\mathrm{T}} \underset{\approx}{A} \underset{\sim}{d}_2} = 2.4966,$$

and

$$\underset{\sim}{x}^* = \underset{\sim}{x}_3 = \underset{\sim}{x}_1 + \alpha_1 \underset{\sim}{d}_1 = \begin{pmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \end{pmatrix}.$$

The minimum value of the function is $f(\underset{\sim}{x}^*) = 0$.

Notice that the convergence criterion is satisfied, this is

$$\nabla f_3 = \begin{pmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \end{pmatrix}.$$

The reader can easily prove that this corresponds to the analytical solution. ∎

The conjugate gradient method is in essence the steepest descent with a deflected search direction. This simple modification does not require a considerably amount of computational work; however, it substantially improves the rate of convergence.

11.3 Newton's method

11.3.1 Definition

When second-order derivatives are available, the objective function can be better approximated. In this way, better search directions can be obtained and the convergence rate can be also improved. Newton's method makes use of the Hessian of the objective function and has quadratic convergence. When the method is applied to a quadratic function that is positive definite, it locates the minimum in one iteration.

Newton's method relay on a quadratic approximation of the function about a current design $\underset{\sim}{x}_k$,

$$f_Q(\underset{\sim}{x}) = f(\underset{\sim}{x}_k) + \nabla f(\underset{\sim}{x}_k)^{\mathrm{T}}(\underset{\sim}{x} - \underset{\sim}{x}_k) + \frac{1}{2}(\underset{\sim}{x} - \underset{\sim}{x}_k)^{\mathrm{T}}\nabla^2 f(\underset{\sim}{x}_k)(\underset{\sim}{x} - \underset{\sim}{x}_k) \tag{11.51}$$

and its gradient

$$\nabla f_Q(\underset{\sim}{x}) = \nabla f(\underset{\sim}{x}_k) + \nabla^2 f(\underset{\sim}{x}_k)(\underset{\sim}{x} - \underset{\sim}{x}_k). \tag{11.52}$$

The minimum of f_Q satisfies that $\nabla f_Q(\underset{\sim}{x}^*_Q) = \underset{\sim}{0}$, this is

$$\nabla f_Q(\underset{\sim}{x}^*_Q) = \nabla f(\underset{\sim}{x}_k) + \nabla^2 f(\underset{\sim}{x}_k)(\underbrace{\underset{\sim}{x}^*_Q - \underset{\sim}{x}_k}_{\underset{\sim}{d}_k}) = \underset{\sim}{0}, \tag{11.53}$$

then

$$\nabla^2 f(\underset{\sim}{x}_k)\underset{\sim}{d}_k = -\nabla f(\underset{\sim}{x}_k) \tag{11.54}$$

Assuming that $\nabla^2 f(\underset{\sim}{x}_k)$ is not singular, and therefore invertible, one observes that

$$\underset{\sim}{d}_k = -\left[\nabla^2 f(\underset{\sim}{x}_k)\right]^{-1} \nabla f(\underset{\sim}{x}_k). \tag{11.55}$$

The descent condition (11.7) is satisfied if

$$-\nabla f_k{}^{\mathrm{T}}[\nabla^2 f_k]^{-1}\nabla f_k < 0, \tag{11.56}$$

which is accomplished if and only if $\nabla^2 f(\underset{\sim}{x}_k)$ is *positive definite*. Defining the search direction as

$$\underset{\sim}{d}_k = \underset{\sim}{x}_{k+1} - \underset{\sim}{x}_k \tag{11.57}$$

the updating rule of the Newton's method can be expressed as

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k - [\nabla^2 f(\underset{\sim}{x}_k)]^{-1}\nabla f(\underset{\sim}{x}_k). \tag{11.58}$$

Some drawbacks of the Newton's method include:

1. Each iteration in the Newton's method requires the Hessian of the objective function. In some func-

tions that might be impossible for certain points; however, when it is possible, the number of function calls increases substantially with respect to the previous first-order methods. For instance, if $f : \mathbb{R}^n \to \mathbb{R}$, then the algorithm requires $n(n+1)/2$ second-order derivatives.

2. The Hessian might be singular (or close to singular) in some points which makes impossible to determine a search direction.

3. The method does not use information collected in previous iterations.

4. The convergence of this method is not guaranteed.

11.3.2 Modified Newton's methods

When the function is highly nonlinear, then a quadratic approximation may not be good enough. In fact, it might happen that $f(\underset{\sim}{x}_{k+1}) \not\leq f(\underset{\sim}{x}_k)$ and convergence cannot be achieved. A modified Newton's method incorporates a line search to find a step size α_k over the search direction $\underset{\sim}{d}_k$. This approach is referred to as *modified Newton's method*.

Marquardt (1964) suggested a modification on Newton's method by proposing a new way to find search directions. In this approach, Marquardt combines the advantages of the Newton's method with the steepest descent direction method. The proposed search direction is defined as

$$\underset{\sim}{d}_k = -(\nabla^2 f(\underset{\sim}{x}_k) + \gamma \underset{\sim}{I})^{-1} \nabla f(\underset{\sim}{x}_k), \tag{11.59}$$

where $\underset{\sim}{I}$ is the identity matrix and γ is a weighting value. When γ is big, then the direction corresponds to the steepest descent direction. When γ is small, then it corresponds to the Newton's direction. This approach is referred to as *Marquardt's modification*.

11.3.3 Algorithm

The algorithm for the modified Newton's method can be described as follows.

Step 1. Estimate an initial design $\underset{\sim}{x}_0$ and a convergence parameter ε.

Step 2. Determine $\nabla f(\underset{\sim}{x}_k)$. If $||\nabla f(\underset{\sim}{x}_k)|| < \varepsilon$, then convergence is obtained and algorithm cannot improve the current design, otherwise go to the next Step.

Step 3. Determine the Hessian $\nabla^2 f(\underset{\sim}{x}_k)$.

Step 4. Determine the search direction

$$\underset{\sim}{d}_k = -\left[\nabla^2 f(\underset{\sim}{x}_k)\right]^{-1} \nabla f(\underset{\sim}{x}_k).$$

Step 5. Update the design

$$\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \alpha_k \underset{\sim}{d}_k,$$

where α_k is the minimum of the function $f(\underset{\sim}{x}_k + \alpha \underset{\sim}{d}_k)$, then go to Step 2.

Example. Given the function

$$f(\underset{\sim}{x}) = 10x_1^4 - 20x_1x_2 + 10x_2^2 + x_1^2 - 2x_1 + 5,$$

perform two iterations using the modified Newton's method from the point $\underset{\sim}{x}_0 = (-1, 3)^{\mathrm{T}}$.

This is not a quadratic function, therefore one cannot expect to find the minimum in one single iteration. The gradient of the function if given by

$$\nabla f(\underset{\sim}{x}) = \begin{pmatrix} -2 + 2x_1 + 40x_1^3 - 20x_2 \\ -20x_1 + 20x_2 \end{pmatrix}$$

and the Hessian is given by

$$\nabla^2 f(\underset{\sim}{x}) = \begin{pmatrix} 2 + 120x_1^2 & -20 \\ -20, 20 & \end{pmatrix}.$$

Iteration 1. The first improved point is $\underset{\sim}{x}_1 = \underset{\sim}{x}_0 + \alpha_0 \underset{\sim}{d}_0$ where

$$\underset{\sim}{d}_0 = -\left[\nabla^2 f(\underset{\sim}{x}_0)\right]^{-1} \nabla f(\underset{\sim}{x}_0) = \begin{pmatrix} 0.2353 \\ -3.7647 \end{pmatrix}$$

Minimizing $f(\alpha) = f(\underset{\sim}{x}_0 + \alpha \underset{\sim}{d}_0)$

$$\alpha_0 = 1.0045$$

then,

$$\underset{\sim}{x}_1 = \underset{\sim}{x}_0 + \alpha_0 \underset{\sim}{d}_0 = \begin{pmatrix} -0.7637 \\ -0.7815 \end{pmatrix}$$

Iteration 2. Following the same procedure,

$$\underset{\sim}{d}_1 = -\left[\nabla^2 f(\underset{\sim}{x}_1)\right]^{-1} \nabla f(\underset{\sim}{x}_1) = \begin{pmatrix} 0.1167 \\ 0.1346 \end{pmatrix}$$

$$\alpha_1 = 1.3424$$

$$\underset{\sim}{x}_2 = \underset{\sim}{x}_1 + \alpha_1 \underset{\sim}{d}_1 = \begin{pmatrix} -0.6070 \\ -0.6008 \end{pmatrix}$$

The exact solution is given by $\underset{\sim}{x}^* = (0.7207, 0.7207)^{\mathrm{T}}$ and $f(\underset{\sim}{x}^*) = 1.5818$.

11.4 Quasi-Newton methods

Quasi-Newton method's approximate the Hessian using first-order derivatives. For positive definite quadratic functions of n variables, these method achieve convergence in n iterations. If the minimum is not found after $n+1$, then the algorithm restarts. In these methods the search direction is given by

$$\underset{\sim}{d}_k = -\underset{\approx}{H}_k \nabla f(\underset{\sim}{x}_k), \tag{11.60}$$

where $\underset{\approx}{H}_k$ is an approximation of $[\nabla^2 f(\underset{\sim}{x}_k)]^{-1}$. Initially, this approximation is the identity matrix, $\underset{\approx}{H}_0 = \underset{\approx}{I}$.

11.4.1 Davidon-Fletcher-Powell (DFP) method

This method was proposed by Davidon (1959) and modified by Fletcher & Powell (1963). In the DFP method, the approximation of the inverse of the Hessian is given by

$$\underset{\approx}{H}_{k+1} = \underset{\approx}{H}_k - \frac{\underset{\approx}{H}_k \underset{\sim}{y}_k \underset{\sim}{y}_k^{\mathrm{T}} \underset{\approx}{H}_k}{\underset{\sim}{y}_k^{\mathrm{T}} \underset{\approx}{H}_k \underset{\sim}{y}_k} + \frac{\underset{\sim}{s}_k \underset{\sim}{s}_k^{\mathrm{T}}}{\underset{\sim}{s}_k^{\mathrm{T}} \underset{\sim}{y}_k}, \tag{11.61}$$

where $\underset{\sim}{s}_k = \underset{\sim}{x}_{k+1} - \underset{\sim}{x}_k$ represents the change in the design and $\underset{\sim}{y}_k = \nabla f(\underset{\sim}{x}_{k+1}) - \nabla f(\underset{\sim}{x}_k)$ represents the change in the gradient.

Example. Minimize the function

$$f(\underset{\sim}{x}) = 5x_1^5 + 2x_1 x_2 + x_2^2 + 7$$

from $\underset{\sim}{x}_0 = (1,2)^{\mathrm{T}}$. Use $\varepsilon = 0.001$ as convergence parameter.

$\underset{\approx}{H}_0 = \underset{\approx}{I}$ then

$$\underset{\sim}{d}_0 = -\nabla f_0 = \begin{pmatrix} 14 \\ 6 \end{pmatrix}.$$

The norm $||\nabla f_0|| = 15.232 > \varepsilon$, then there is no convergence and the algorithm continues. The updated design is given by

$$\underset{\sim}{x}_1 = \underset{\sim}{x}_0 + \alpha_0 \underset{\sim}{d}_0$$

where $\alpha_0 = 0.0988$. Por lo tanto

$$\underset{\sim}{x}_1 = \begin{pmatrix} -1.046 \\ 2.042 \end{pmatrix}.$$

In the second iteration, $||\nabla f_1|| = 2.29 > \varepsilon$, therefore the is no convergence. The algorithm determines

$\underset{\sim}{s}_0 = \underset{\sim}{x}_1 - \underset{\sim}{x}_0 = (-1.386, -0.593)^{\mathrm{T}}$ and $\underset{\sim}{y}_0 = \nabla f_1 - \nabla f_0 = (-15.046, -3.958)^{\mathrm{T}}$. Then,

$$\underset{\approx}{H}_1 = \begin{pmatrix} 0.148 & -0.211 \\ -0.211 & 0.950 \end{pmatrix}.$$

The new search direction is

$$\underset{\sim}{d}_1 = \begin{pmatrix} 0.586 \\ -1.719 \end{pmatrix}.$$

The updated design is

$$\underset{\sim}{x}_2 = \underset{\sim}{x}_1 + \alpha_1 \underset{\sim}{d}_1$$

where $\alpha_1 = 0.776$. Therefore

$$\underset{\sim}{x}_2 = \begin{pmatrix} 0.069 \\ 0.073 \end{pmatrix}.$$

∎

11.4.2 Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

In the BFGS, the approximation of the inverse of the Hessian is given by

$$\underset{\approx}{H}_{k+1} = \underset{\approx}{H}_k - \frac{\underset{\sim}{s}_k \underset{\sim}{y}_k{}^{\mathrm{T}} \underset{\approx}{H}_k + \underset{\approx}{H}_k \underset{\sim}{y}_k \underset{\sim}{s}_k{}^{\mathrm{T}}}{\underset{\sim}{s}_k{}^{\mathrm{T}} \underset{\sim}{y}_k} + \left(1 + \frac{\underset{\sim}{y}_k{}^{\mathrm{T}} \underset{\approx}{H}_k \underset{\sim}{y}_k}{\underset{\sim}{s}_k{}^{\mathrm{T}} \underset{\sim}{y}_k}\right) \frac{\underset{\sim}{s}_k \underset{\sim}{s}_k{}^{\mathrm{T}}}{\underset{\sim}{s}_k{}^{\mathrm{T}} \underset{\sim}{y}_k}, \tag{11.62}$$

where $\underset{\sim}{s}_k = \underset{\sim}{x}_{k+1} - \underset{\sim}{x}_k$ represents the change in the design variables and $\underset{\sim}{y}_k = \nabla f_{k+1} - \nabla f_k$ represents the change in the gradient.

Example. Minimize the function

$$f(\underset{\sim}{x}) = 5x_1^5 + 2x_1x_2 + x_2^2 + 7$$

from $\underset{\sim}{x}_0 = (1, 2)^{\mathrm{T}}$. Use $\varepsilon = 0.001$ as convergence parameter.

The first iteration is just like in the previous example, this is $\underset{\approx}{H}_0 = \underset{\approx}{I}$ and

$$\underset{\sim}{x}_1 = \begin{pmatrix} -1.046 \\ 2.042 \end{pmatrix}.$$

In the second iteration one obtains that

$$\underset{\approx}{H}_1 = \begin{pmatrix} 1.770 & -7.506 \\ -7.506 & 33.750 \end{pmatrix}$$

and the new search direction is

$$\underset{\sim}{d}_1 = \begin{pmatrix} 17.20 \\ -76.77 \end{pmatrix}.$$

The updated design is given by

$$\underset{\sim}{x}_2 = \underset{\sim}{x}_1 + \alpha_1 \underset{\sim}{d}_1$$

where $\alpha_1 = 0.0185$. Therefore

$$\underset{\sim}{x}_2 = \begin{pmatrix} -0.0688 \\ -0.0098 \end{pmatrix}.$$

∎

11.5 Trust regions methods

11.5.1 Definition

The basic idea under this methods is to approximate the objective function f with a simpler expression f_A that reasonably reflects its behavior in the neighborhood of a current design $\underset{\sim}{x}_k$. This neighborhood is referred to as the trust region. If f is highly nonlinear the approximation f_A will be valid in a *trust region* Ω_k around $\underset{\sim}{x}_k$. This region can be described as

$$\Omega_k = \{\underset{\sim}{x} : ||\underset{\sim}{x} - \underset{\sim}{x}_k||_\infty \leq h_k\}, \tag{11.63}$$

where h_k is the size of the trust region, which is dynamically adjusted. The sub-optimization problem can be stated as

$$\begin{aligned} \min_{\underset{\sim}{d}} \quad & f_A(\underset{\sim}{d}) \\ \text{s.a.} \quad & -h_k \leq d_i \leq h_k, \qquad i = 1, \ldots, n. \end{aligned} \tag{11.64}$$

The solution of this problem is $\underset{\sim}{d}_k$. Notice that these methods do not require line search as the size of the trust region accounts for that.

If the approximation is linear, then the sub-optimization problem to locate $\underset{\sim}{d}_k$ within Ω_k can be described as

$$\begin{aligned} \min_{\underset{\sim}{d}} \quad & f_L(\underset{\sim}{d}) = f(\underset{\sim}{x}_k) + \nabla f(\underset{\sim}{x}_k)^{\mathrm{T}} \underset{\sim}{d} \\ \text{s.a.} \quad & -h_k \leq d_i \leq h_k, \qquad i = 1, \ldots, n. \end{aligned} \tag{11.65}$$

This type of problems with linear objective and linear constraints is known as *linear programming* (LP) problem.

If the approximation is quadratic, then the sub-optimization problem is

$$\begin{array}{ll} \min\limits_{\underset{\sim}{d}} & f_Q(\underset{\sim}{d}) = f(\underset{\sim}{x}_k) + \nabla f(\underset{\sim}{x}_k)^{\mathrm{T}}\underset{\sim}{d} + \frac{1}{2}\underset{\sim}{d}^{\mathrm{T}}\nabla^2 f(\underset{\sim}{x}_k)\underset{\sim}{d} \\ \text{s.a.} & -h_k \leq d_i \leq h_k, \qquad i = 1, \ldots, n. \end{array} \tag{11.66}$$

This type of problems with a quadratic objective function and linear constraints is known as *quadratic programming* (QP) problem.

11.5.2 Reliability index

The reliability index is defined as

$$r_k = \frac{f(\underset{\sim}{x}_k) - f(\underset{\sim}{x}_k + \underset{\sim}{d}_k)}{f(\underset{\sim}{x}_k) - f_A(\underset{\sim}{x}_k + \underset{\sim}{d}_k)} \tag{11.67}$$

where f is the objective function to minimize and f_A is its approximation. If the numerator and the denominator are greater than zero and $r_k \approx 1$, then it is said that there is a good agreement between the function the function and its approximation. On the hand, if the numerator is negative, then r_k is also negative, which is not desired. In that case, the new point $\underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \underset{\sim}{d}_k$ does not decrease the function so the size of the trust region h_k has to be adjusted according to the value of r_k. A heuristic updating rule is as follows:

$$\begin{array}{lcl} \text{If } r_k < 0.25 & \Rightarrow & h_{k+1} = \frac{||\underset{\sim}{d}_k||_\infty}{4} \\ \text{If } r_k > 0.75 \text{ y } h_k = ||\underset{\sim}{d}_k||_\infty & \Rightarrow & h_{k+1} = 2h_k \\ \text{Otherwise} & \Rightarrow & h_{k+1} = h_k \end{array} \tag{11.68}$$

11.5.3 Algorithm

Step 1. Estimate $\underset{\sim}{x}_0$, $h_0 = 1$ and a convergence parameter ε.

Step 2. Determine $\nabla f(\underset{\sim}{x}_k)$ and $\nabla^2 f(\underset{\sim}{x}_k)$. If $||\nabla f(\underset{\sim}{x}_k)|| \leq \varepsilon$ then convergence has been achieved and the minimum has been found. Otherwise, continue to the next Step.

Step 3. Obtain $\underset{\sim}{d}_k$ solving the sub-optimization problem (11.64).

Step 4. Evaluar $f(\underset{\sim}{x}_{k+1})$ and obtain r_k according to (11.67).

Step 5. Update the size of the trust region h_{k+1} according to (11.68).

Step 6. Determine the new point $\underset{\sim}{x}_{k+1}$ according to the following rule:

$$\begin{array}{lcl} \text{If } r_k \leq 0 & \Rightarrow & \underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k \text{ and go to Step 3} \\ \text{Otherwise} & \Rightarrow & \underset{\sim}{x}_{k+1} = \underset{\sim}{x}_k + \underset{\sim}{d}_k \text{ and go to Step 2} \end{array}$$

11.6 Least square error

Consider the system of linear equations

$$\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} = \underset{\sim}{b}. \tag{11.69}$$

where the matrix $\underset{\sim}{\underset{\sim}{A}} \in \mathbb{R}^{m\times n}$ applies the vector $\underset{\sim}{x} \in \mathbb{R}^n$ into $\underset{\sim}{b} \in \mathbb{R}^m$. When the system is inconsistent, this is $\mathrm{rank}(\underset{\sim}{\underset{\sim}{A}}) < \mathrm{rank}(\underset{\sim}{\underset{\sim}{A}}, \underset{\sim}{b})$, it does not have a solution. However, one can attempt to provide an approximate point that minimizes $||\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} - \underset{\sim}{b}||$. This is,

$$\min_{\underset{\sim}{x}} \qquad f(\underset{\sim}{x}) = (\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} - \underset{\sim}{b})^{\mathrm{T}}(\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} - \underset{\sim}{b}) \tag{11.70}$$

The necessary condition for optimality can be expressed as $\nabla f(\underset{\sim}{x}^*) = \underset{\sim}{0}$, where

$$\begin{aligned} \nabla f(\underset{\sim}{x}) &= \underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} - \underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{b} + \underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} - \underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{b} \\ &= 2\underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} - 2\underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{b}. \end{aligned}$$

In the optimal point,

$$\underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x}^* = \underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{b}. \tag{11.71}$$

Solving for $\underset{\sim}{x}^*$ yields

$$\underset{\sim}{x}^* = (\underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{\underset{\sim}{A}})^{-1}\underset{\sim}{\underset{\sim}{A}}^{\mathrm{T}}\underset{\sim}{b}. \tag{11.72}$$

This solution is referred to as *least square error*. Notice that this is a convex problem, therefore $\underset{\sim}{x}^*$ is a global minimum.

Example. Solve for $\underset{\sim}{x}$ in the following system of linear equations,

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Since $\mathrm{rank}(\underset{\sim}{\underset{\sim}{A}}) = 2$ and $\mathrm{rank}(\underset{\sim}{\underset{\sim}{A}}, \underset{\sim}{b}) = 3$ the system is inconsistent and does not have a solution. However, the solution that minimizes the quadratic error is obtained using (11.72), this is

$$\begin{pmatrix} x_1^* \\ x_2^* \end{pmatrix} = \left[\begin{pmatrix} 1 & 3 & -1 \\ 2 & 4 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ -1 & 1 \end{pmatrix} \right]^{-1} \begin{pmatrix} 1 & 3 & -1 \\ 2 & 4 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tag{11.73}$$

$$= \begin{pmatrix} -0.4516 \\ 0.6129 \end{pmatrix}. \tag{11.74}$$

∎

One of the most important applications of this method is the identification of the coefficients of a function that better matches a cloud of points.

Example. An experiment yields the data in the following table.

t	x
0.00	1.001
0.10	1.089
0.23	1.240
0.70	1.604
0.90	1.738
1.50	2.020
2.65	1.412
3.00	1.241

Find the least square best fit coefficients a, b, and c if the assumed functional form is

(a) $x = a + bt + ct^2$,

(b) $x = a + b\sin(t) + c\sin(2t)$.

Which best fit estimate has the smallest least square error?

(a) The system of linear equations can be expressed as

$$\begin{pmatrix} 1.000 & 0.000 & 0.000 \\ 1.000 & 0.100 & 0.010 \\ 1.000 & 0.230 & 0.053 \\ 1.000 & 0.700 & 0.490 \\ 1.000 & 0.900 & 0.810 \\ 1.000 & 1.500 & 2.250 \\ 1.000 & 2.650 & 7.023 \\ 1.000 & 3.000 & 9.000 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1.001 \\ 1.089 \\ 1.240 \\ 1.604 \\ 1.738 \\ 2.020 \\ 1.412 \\ 1.241 \end{pmatrix}.$$

Using (11.72), the solution can be expressed as

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0.982 \\ 1.205 \\ -0.379 \end{pmatrix}.$$

The function is $x(t) = 0.982 + 1.205t - 0.379t^2$. The quadratic error is 0.0224.

(b) In this case, the system can be expressed as

$$\begin{pmatrix} 1.000 & 0.000 & 0.000 \\ 1.000 & 0.100 & 0.199 \\ 1.000 & 0.228 & 0.444 \\ 1.000 & 0.644 & 0.985 \\ 1.000 & 0.783 & 0.974 \\ 1.000 & 0.997 & 0.141 \\ 1.000 & 0.472 & -0.832 \\ 1.000 & 0.141 & -0.279 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1.001 \\ 1.089 \\ 1.240 \\ 1.604 \\ 1.738 \\ 2.020 \\ 1.412 \\ 1.241 \end{pmatrix}.$$

Using (11.72), the solution can be expressed as

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1.017 \\ 0.960 \\ -0.011 \end{pmatrix}.$$

The function is $x(t) = 1.017 + 0.960\sin(t) - 0.011\sin(2t)$. The quadratic error is 0.0157, therefore, it is a better approximation. ∎

In some occasions, the data are weighted according to their reliability. In this case, one makes use of a weighting matrix $\underset{\sim}{\underset{\sim}{W}}$, such that

$$\underset{\sim}{\underset{\sim}{W}}\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x} = \underset{\sim}{b}. \tag{11.75}$$

In this case, the optimum solution is given by

$$(\underset{\sim}{\underset{\sim}{W}}\underset{\sim}{\underset{\sim}{A}})^{\mathrm{T}}\underset{\sim}{\underset{\sim}{W}}\underset{\sim}{\underset{\sim}{A}}\underset{\sim}{x}^* = (\underset{\sim}{\underset{\sim}{W}}\underset{\sim}{\underset{\sim}{A}})^{\mathrm{T}}\underset{\sim}{b} \tag{11.76}$$

$$\underset{\sim}{x}^* = [(\underset{\sim}{\underset{\sim}{W}}\underset{\sim}{\underset{\sim}{A}})^{\mathrm{T}}\underset{\sim}{\underset{\sim}{W}}\underset{\sim}{\underset{\sim}{A}}]^{-1}(\underset{\sim}{\underset{\sim}{W}}\underset{\sim}{\underset{\sim}{A}})^{\mathrm{T}}\underset{\sim}{b}. \tag{11.77}$$

This method is referred to as *weighted least square error*.

The application of least square error is not limited to linear systems. In fact, one can find minimize the sum of squares deviations between a set of given values and predicted values. In any case, the least square error problem is given by

$$\min_{\underset{\sim}{x}} \qquad f(\underset{\sim}{x}) = \textstyle\sum_i (g_i - p_i(\underset{\sim}{x}))^2 \tag{11.78}$$

where g_i represent the given data and p_i are the predicted data. To illustrate this concept, let us consider the following example.

Example. An experiment yields the data in the following table.
Find the least square best fit coefficients a, b, and c if the assumed functional form is

(a) $f(\underset{\sim}{x}) = ax_1 + bx_2 + c$,

f	x_1	x_2
2.2	5.0	10.0
9.5	3.0	1.0
23.6	0.6	0.6
74.3	0.1	2.0
6.3	3.0	1.8

(b) $f(\underset{\sim}{x}) = ax_1^b x_2^c$.

(a) In this case we have a system of linear equations given by

$$\begin{pmatrix} 5.0 & 10.0 & 1.0 \\ 3.0 & 1.0 & 1.0 \\ 0.6 & 0.6 & 1.0 \\ 0.1 & 2.0 & 1.0 \\ 3.0 & 1.8 & 1.0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 2.2 \\ 9.5 \\ 23.6 \\ 74.3 \\ 6.3 \end{pmatrix}$$

Using (11.72), the solution can be expressed as

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} -18.17 \\ 4.34 \\ 52.35 \end{pmatrix}.$$

The least square error is 707.2.

(b) In this case, the function is nonlinear and (11.72) cannot be used. In this case, the general approach in (11.78) is utilized. The optimization problem is given by

$$\min_{a,b,c} \quad (2.2 - a5.0^b 10.0^c)^2 + \cdots + (6.3 - a3.0^b 1.8^c)^2$$

The solution of this nonlinear, unconstrained problem is given by

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 15.72 \\ -0.72 \\ -0.14 \end{pmatrix}.$$

The least square error is 8.08 which indicates that it is a better fit than the previous functional. ∎

Chapter 12

Problems

12.1 Exercises

1. Consider the following functions:

   - $f(\underset{\sim}{x}) = (x_1 - 1)^2 + (x_2^2 - 1)^2$
   - $f(\underset{\sim}{x}) = x_1 + \frac{1}{x_1 x_1} + x_2$
   - $f(\underset{\sim}{x}) = \frac{x_1 + x_2}{1 + x_1^2 + x_2^2 + x_1 x_2}$
   - $f(\underset{\sim}{x}) = (x_1 - 1)^2 + x_1 x_3 + x_2 x_3$

   Determine (a) all stationary points and (b) check if they are strict local minima using the sufficient conditions.

2. Determine if the given direction at the point is that of descent for the following functions (show all the calculations). In each case, determine the steepest descent direction.

   - $f(\underset{\sim}{x}) = 10(x_2 - x_1^2)^2 + (1 - x_1)^2$; $\underset{\sim}{d} = (162, -40)^{\mathrm{T}}$ at $\underset{\sim}{x} = (2, 2)^{\mathrm{T}}$.
   - $f(\underset{\sim}{x}) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2$; $\underset{\sim}{d} = (2, -2, 2, -2)^{\mathrm{T}}$ at $\underset{\sim}{x} = (2, 1, 4, 3)^{\mathrm{T}}$.
   - $f(\underset{\sim}{x}) = \sum_{i=1}^{i=4}[(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2]$; $\underset{\sim}{d} = (-1, 1, 2, 1, -1)^{\mathrm{T}}$ at $\underset{\sim}{x} = (-1, 1, 2, 1, -1)^{\mathrm{T}}$.
   - $f(\underset{\sim}{x}) = x_1 \sin(x_2 x_3^2)$; $\underset{\sim}{d} = (1, -1, -1)^{\mathrm{T}}$ at $\underset{\sim}{x} = (1, 2, 1)^{\mathrm{T}}$.
   - $f(\underset{\sim}{x}) = \log_{x_1}(x_2)$; $\underset{\sim}{d} = (-1, 2)^{\mathrm{T}}$ at $\underset{\sim}{x} = (10, 10)^{\mathrm{T}}$.

3. Consider the following functions, search directions, and current points:

   - $f(\underset{\sim}{x}) = 4x_1^2 + x_2^2 - 4x_1 x_2$; $\underset{\sim}{d} = (-1, 2)^{\mathrm{T}}$ at $\underset{\sim}{x} = (-1, 0)^{\mathrm{T}}$
   - $f(\underset{\sim}{x}) = x_1^2 + x_2^2 + x_3^2$; $\underset{\sim}{d} = (-2, -4, 2)^{\mathrm{T}}$ at $\underset{\sim}{x} = (1, 2, -1)^{\mathrm{T}}$
   - $f(\underset{\sim}{x}) = (x_1 - 1)^2 + (x_2 - 2)^2 + (x_3 - 3)^2 + (x_4 - 4)^2$; $\underset{\sim}{d} = (-2, 2, -2, 2)^{\mathrm{T}}$ at $\underset{\sim}{x} = (2, 1, 4, 3)$.

In each case, (a) derive the function of one variable (line search function) and (b) determine optimum step size (show all calculations).

4. Consider the function $f(\underset{\sim}{x}) = x_1 x_2^2$ at the point $\underset{\sim}{x}_0 = (1, 2)^{\mathrm{T}}$. (a) Obtain the expression quadratic approximation f_Q and make a plot of f and f_Q. (b) Analytically, obtain the minimum of f_Q. This is equivalent to one iterations of Newton's method. (c) Improve this minimum by performing a line search. This is equivalent to one iteration of a modified Newton's method.

5. For the function $f(\underset{\sim}{x}) = x_1 x_2$, determine the expression for $f(\alpha)$ along the line $x_1 = x_2$ and along the line joining $(0, 1)^{\mathrm{T}}$ to $(1, 0)^{\mathrm{T}}$.

6. Perform two iterations (by hand) using steepest descent, Fletcher-Reeves, Newtons's, and BFGS for the following functions:

   - $f(\underset{\sim}{x}) = (x_1 + x_2 - 6)^2) + (2x_1 + x_2 - 5)^2$ at $\underset{\sim}{x}_0 = (0, 0)^{\mathrm{T}}$.
   - $f(\underset{\sim}{x}) = (x_1 - 1)^2 + 2x_2^2 + 2x_3^2 + 2x_1 x_2 + 2x_2 x_3$ at $\underset{\sim}{x}_0 = (0, 0, 0)^{\mathrm{T}}$.

7. A cardboard box is to be designed to have a volume of 1 m^3. Determine the optimal values of length, width, and height to minimize the amount of cardboard material. Hint: The problem can be formulated as unconstrained in terms of two design variables.

12.2 Mini-projects

1. Develop a MATLAB function to solve unconstrained optimization problems using:

   - Steepest descent method.
   - Fletcher-Reeves.
   - Polak-Ribiere.
   - Modified Newton's method (line search).
   - BFGS
   - DFP
   - Trust region method with linear approximation.

   Makes sure that your algorithm works for two and three dimensional problems.