CHAPTER 3
UNCONSTRAINED OPTIMIZATION

1. Preliminaries

1.1. Introduction

In this chapter we will examine some theory for the optimization of unconstrained functions. We will assume all functions are continuous and differentiable. Although most engineering problems are constrained, much of constrained optimization theory is built upon the concepts and theory presented in this chapter.

1.2. Notation

We will use lower case italics, e.g., *x*, to represent a scalar quantity. Vectors will be represented by lower case bold, e.g., **x**, and matrices by upper case bold, e.g., **H**.

The set of *n* design variables will be represented by the *n*-dimensional vector **x**. For example, previously we considered the design variables for the Two-bar truss to be represented by scalars such as diameter, d, thickness, t, height, h; now we consider diameter to be the first element, x_1, of the vector **x**, thickness to be the second element, x_2, and so forth. Thus for any problem the set of design variables is given by **x**.

Elements of a vector are denoted by subscripts. Values of a vector at specific points are denoted by superscripts. Typically $\mathbf{x}^0$ will be the starting vector of values for the design variables. We will then move to $\mathbf{x}^1$, $\mathbf{x}^2$, until we reach the optimum, which will be **x***. A summary of notation used in this chapter is given in Table 1.

Table 1 Notation

$\mathbf{A}$	Matrix **A**	$\mathbf{x}$, $\mathbf{x}^k$, $\mathbf{x}^*$	Vector of design variables, vector at iteration k, vector at the optimum
$\mathbf{I}$	Identity matrix	$x_1, x_2 \ldots x_n$	Elements of vector **x**
$\mathbf{a}$	Column vector	$\mathbf{s}, \mathbf{s}^k$	Search direction, search direction at iteration k
$\mathbf{a}_i \quad i = 1, 2, \ldots$	Columns of **A**	$\alpha, \alpha^k, \alpha^*$	Step length, step length at iteration k, step length at minimum along search direction
$\mathbf{e}_i \quad i = 1, 2, \ldots$	Coordinate vectors (columns of **I)**	$f(\mathbf{x}),\ f(\mathbf{x}^k),\ f^k$	Objective function, objective evaluated at $\mathbf{x}^k$
$\mathbf{A}^T, \mathbf{a}^T$	transpose	$\nabla^2 f(\mathbf{x}^k), \nabla^2 f^k$ $\mathbf{H}(\mathbf{x}^k), \mathbf{H}^k$	Hessian matrix at $\mathbf{x}^k$
$\nabla f(\mathbf{x}), \nabla f(\mathbf{x}^k), \nabla f^k$	Gradient of $f(\mathbf{x})$, gradient evaluated at $\mathbf{x}^k$	$\lvert\mathbf{A}\rvert$	Determinant of **A**

$\Delta\mathbf{x}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$	Difference in $\mathbf{x}$ vectors	$\mathbf{x} \in R^n$	All vectors which are in n-dimensional Euclidean space
$\boldsymbol{\gamma}^k = \nabla f^{k+1} - \nabla f^k$	Difference in gradients at $\mathbf{x}^{k+1}$, $\mathbf{x}^k$	$\mathbf{N}^k$	Direction matrix at $\mathbf{x}^k$

1.3. Statement of Problem

The problem we are trying to solve in this chapter can be stated as,

Find $\mathbf{x}$, $\mathbf{x} \in R^n$
To Minimize $f(\mathbf{x})$

1.4. Gradient Vector

1.4.1. Definition

The gradient of $f(\mathbf{x})$ is denoted $\nabla f(\mathbf{x})$. The gradient is defined as a column vector of the first partial derivatives of $f(\mathbf{x})$:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{3.1}$$

1.4.2. Example: Gradient of a Function

Evaluate the gradient of the function $f(\mathbf{x}) = 6 - 2x_1 + x_2 + 2x_1^2 + 3x_1x_2 + x_2^2$

$$\nabla f = \begin{bmatrix} -2 + 4x_1 + 3x_2 \\ 1 + 3x_1 + 2x_2 \end{bmatrix} \text{ If evaluated at } \mathbf{x}^0 = \begin{bmatrix} -2 \\ 2 \end{bmatrix},\ \nabla f^0 = \begin{bmatrix} -4 \\ -1 \end{bmatrix}$$

A very important property of the gradient vector is that it *is orthogonal to the function contours and points in the direction of greatest increase of a function.* The negative gradient points in the direction of greatest decrease. Any vector $\mathbf{v}$ which is orthogonal to $\nabla f(\mathbf{x})$ will satisfy $\mathbf{v}^{\mathbf{T}} \nabla f(\mathbf{x}) = 0$.

1.5. Vectors That Point "Downhill" or "Uphill"

If we have some search direction **s**, then $\mathbf{s}^T\nabla f$ is proportional to the projection of **s** onto the gradient vector. We can see this better in Fig. 3.1:

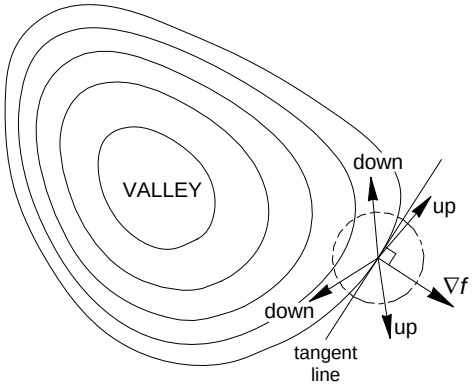

Fig. 3.1. Vectors that point uphill or downhill.

As long as $\mathbf{s}^T\nabla f > 0$, then **s** points, at least for some small distance, in a direction that increases the function (points uphill). In like manner, if $\mathbf{s}^T\nabla f < 0$, then **s** points downhill. As an example, suppose at the current point in space the gradient vector is $\nabla f(\mathbf{x}^k)^T = [6,1,-2]$. We propose to move from this point in a search direction $\mathbf{s}^T = [-1,1,0]$. Does this direction go downhill? We evaluate $\mathbf{s}^T\nabla f = \begin{bmatrix}-1,-1,0\end{bmatrix}\begin{bmatrix}6\\1\\-2\end{bmatrix} = -7$

So this direction would take us downhill, at least for a short step.

1.6. Hessian Matrix

1.6.1. Definition

The Hessian Matrix, **H**(**x**) or $\nabla^2 f(\mathbf{x})$ is defined to be the square matrix of second partial derivatives:

$$\mathbf{H}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \cdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \qquad (3.2)$$

We can also obtain the Hessian by applying the gradient operator on the gradient transpose,

$$\mathbf{H}(\mathbf{x}) = \nabla^2 f(\mathbf{x}) = \nabla(\nabla f(\mathbf{x})^T) = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \cdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

The Hessian is a symmetric matrix. The Hessian matrix gives us information about the curvature of a function, and tells us how the gradient is changing.

For simplicity, we will sometimes write $\mathbf{H}^k$ instead of $\mathbf{H}(\mathbf{x}^k)$.

1.6.2. Example: Hessian Matrix

Find the Hessian matrix for the function, $f(x) = 6 - 2x_1 + x_2 + 2x_1^2 + 3x_1x_2 + x_2^2$

$$\nabla f = \begin{bmatrix} -2 + 4x_1 + 3x_2 \\ 1 + 3x_1 + 2x_2 \end{bmatrix}, \qquad \frac{\partial^2 f}{\partial x_1^2} = 4 \quad \frac{\partial^2 f}{\partial x_1 \partial x_2} = 3$$

$$\frac{\partial^2 f}{\partial x_2 \partial x_1} = 3 \quad \frac{\partial^2 f}{\partial x_2^2} = 2$$

and the Hessian is:

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}$$

1.7. Positive and Negative Definiteness

1.7.1. Definitions

If for any vector, **x**, the following is true for a symmetric matrix **B,**

$$\begin{aligned} &\mathbf{x}^T\mathbf{B}\mathbf{x} > 0 \text{ then } B \textit{ is positive definite} \\ &\mathbf{x}^T\mathbf{B}\mathbf{x} < 0 \text{ then } B \textit{ is negative definite} \end{aligned} \tag{3.3}$$

1.7.2. Checking Positive Definiteness

The above definition is not very useful in terms of checking if a matrix is positive definite, because it would require that we examine every possible vector **x** to see if the condition given in (3.3) is true. So, how can we tell if a matrix is positive definite? There are three ways we will mention,

1. A symmetric matrix **B** is positive definite if all eigenvalues of **B** are positive.

2. A symmetric matrix is positive definite if and only if the determinant of each of its principal minor matrices is positive.

3. A $n \times n$ matrix $\mathbf{B}$ is symmetric and positive definite if and only if it can be written as $\mathbf{B} = \mathbf{L}\mathbf{L}^T$ where $\mathbf{L}$ is a lower triangular matrix with positive diagonal elements. The $\mathbf{L}$ matrix can be developed through Choleski decomposition.

The matrix we will be most interested in checking is the Hessian matrix, $\mathbf{H}(\mathbf{x})$

What does it mean for the Hessian to be positive or negative definite? If positive definite, it means curvature of the function is everywhere positive. This will be an important condition for checking if we have a minimum. If negative definite, curvature is everywhere negative. This will be a condition for verifying we have a maximum.

1.7.3. Example: Checking if a Matrix is Positive Definite Using Principal Minor Matrices

Is the matrix given below positive definite? We need to check the determinants of the principal minor matrices, found by taking the determinant of a 1x1 matrix along the diagonal, the determinant of a 2x2 matrix along the diagonal, and finally the determinant of the entire matrix. If any one of these determinants is not positive, the matrix is not positive definite.

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

$$\left|\begin{bmatrix} 2 \end{bmatrix}\right| = 2 > 0 \qquad \left|\begin{bmatrix} 2 & 3 \\ 3 & 5 \end{bmatrix}\right| = 1 > 0 \qquad \left|\begin{bmatrix} 2 & 3 & -2 \\ 3 & 5 & -1 \\ -2 & -1 & 5 \end{bmatrix}\right| = -5 < 0$$

The determinants of the first two principal minors are positive. However, because the determinant of the matrix as a whole is negative, this matrix is not positive definite.

We also note that the eigenvalues are -0.15, 4.06, 8.09. That these are not all positive also indicates the matrix is not positive definite.

1.7.4. Checking Negative Definiteness

How can we check to see if a matrix is negative definite? There are two ways we will mention,

1. A symmetric matrix **B** is negative definite if all eigenvalues of **B** are negative.

2. A symmetric matrix is negative definite if we reverse the sign of each element and the resulting matrix is positive definite.

   Note: A symmetric matrix is not negative definite if the determinant of each of its principal minor matrices is negative. Rather, in the negative definite case, the signs of the determinants alternate minus and plus, so the easiest way to check for negative definiteness using principal minor matrices is to reverse all signs and see if the resulting matrix is positive definite.

It is also possible for a matrix to be positive *semi*-definite, or negative *semi*-definite. This occurs when one or more of the determinants or eigenvalues are equal to zero, and the others are all positive (or negative, as the case may be). These are special cases we won't worry about here.

If a matrix is neither positive definite nor negative definite (nor semi-definite) then it is *indefinite*. If using principal minor matrices, note that we need to check both cases before we reach a conclusion that a matrix is indefinite.

1.7.5. Example: Checking if a Matrix is Negative Definite Using Principal Minor Matrices

Is the matrix given above negative definite? We reverse the signs and see if the resulting matrix is positive definite:

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

$$\left|\left[-2\right]\right| = -2 < 0$$

Because the first determinant is negative there is no reason to go further. We also note that the eigenvalues of the "reversed sign" matrix are not all positive.

Because this matrix is neither positive nor negative definite, it is indefinite.

1.8. Taylor Expansion

1.8.1. Definition

The Taylor expansion is an approximation to a function at a point $\mathbf{x}^k$ and can be written in vector notation as:

$$f(\mathbf{x}) = f(\mathbf{x}^k) + \left(\nabla f(\mathbf{x}^k)\right)^{\mathrm{T}} (\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^{\mathrm{T}} \nabla^2 f(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) + \ldots \qquad (3.4)$$

If we note that $\mathbf{x} - \mathbf{x}^k$ can be written as $\Delta\mathbf{x}^k$, and using notation $f(\mathbf{x}^k) = f^k$, we can write (3.4) more compactly as,

$$f^{k+1} = f^k + \left(\nabla f^k\right)^{\mathrm{T}} \Delta\mathbf{x}^k + \frac{1}{2}\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \nabla^2 f^k \Delta\mathbf{x}^k + \ldots \tag{3.5}$$

The Taylor expansion allows us to approximate any continuous function as a polynomial in terms of its derivatives at the point $\mathbf{x}^k$. We can make a linear approximation by taking the first two terms of the expansion. We can make a quadratic approximation by taking the first three terms of the expansion.

1.8.2. Example: Quadratic Approximation of a Transcendental Function

Suppose $f(\mathbf{x}) = 2\left(x_1\right)^{1/2} + 3\ln\left(x_2\right)$

$$\text{at } \left(\mathbf{x}^k\right)^{\mathrm{T}} = [5,4] \qquad \nabla f^T = \left[x_1^{(-1/2)}, \frac{3}{x_2}\right] \qquad \left(\nabla f^k\right)^T = \left[0.447, 0.750\right]$$

$$\frac{\partial^2 f}{\partial x_1^2} = -\frac{1}{2}x_1^{(-3/2)} \qquad \frac{\partial^2 f}{\partial x_1 \partial x_2} = 0 \qquad \frac{\partial^2 f}{\partial x_2 \partial x_1} = 0 \qquad \frac{\partial^2 f}{\partial x_2^2} = \frac{-3}{x_2^2}$$

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} -\frac{1}{2}x_1^{(-3/2)} & 0 \\ 0 & \frac{-3}{x_2^2} \end{bmatrix} \text{ at } \begin{bmatrix} 5 \\ 4 \end{bmatrix} = \begin{bmatrix} -0.045 & 0.0 \\ 0.0 & -0.188 \end{bmatrix}$$

$$f(\mathbf{x}) \approx 8.631 + [0.447, 0.750]\begin{bmatrix} x_1 - 5 \\ x_2 - 4 \end{bmatrix} + \frac{1}{2}[x_1 - 5, x_2 - 4]\begin{bmatrix} -0.045 & 0.0 \\ 0.0 & -0.188 \end{bmatrix}\begin{bmatrix} x_1 - 5 \\ x_2 - 4 \end{bmatrix}$$

If we wish, we can stop here with the equation in vector form. To see the equation in scalar form we can carry out the vector multiplications and combine similar terms:

$$f(\mathbf{x}) \approx 8.631 + 0.447x_1 - 2.235 + 0.750x_2 - 3.000 + \frac{1}{2}\left[\left(-0.045x_1 + 0.225\right), \left(-0.188x_2 + 0.752\right)\right]\begin{bmatrix} x_1 - 5 \\ x_2 - 4 \end{bmatrix}$$

$$f(\mathbf{x}) \approx 3.396 + 0.447x_1 + 0.750x_2 + \frac{1}{2}\left(-0.045x_1^2 + 0.450x_1 - 1.125 - 0.188x_2^2 + 1.504x_2 - 3.008\right)$$

$$f(\mathbf{x}) \approx 1.300 + 0.672x_1 + 1.502x_2 - 0.023x_1^2 - 0.094x_2^2$$

Evaluating and comparing this approximation to the original:

$[\mathbf{x}]^T$	Quadratic	Actual	Error
[5,4]	8.63	8.63	0.00
[5,5]	9.28	9.3	0.02
[6,4]	9.05	9.06	0.01
[7,6]	10.55	10.67	0.12
[2,1]	3.98	2.83	-1.15
[9,2]	8.19	8.08	-0.11

We notice that the further the point gets from the expansion point, the greater the error that is introduced. We also see that at the point of expansion the approximation is exact.

2. Properties and Characteristics of Quadratic Functions

A lot of optimization theory is based on optimizing quadratic functions. It is therefore helpful to investigate some of the properties of these functions.

2.1. Representation

We can represent a quadratic function three ways—as a scalar equation, a general vector equation, and as a Taylor expansion. Although these representations look different, they give exactly the same results. For example, consider the equation,

$$f(x) = 6 - 2x_1 + x_2 + 2x_1^2 + 3x_1x_2 + x_2^2 \tag{3.6}$$

This is a scalar representation of a quadratic.

As a another representation, we can write a quadratic in general vector form,

$$f(\mathbf{x}) = a + \mathbf{b}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{C}\mathbf{x} \tag{3.7}$$

By inspection, the example given in (3.6), in the form of (3.7), is:

$$f(\mathbf{x}) = 6 + [-2,\ 1]\mathbf{x} + \frac{1}{2}\mathbf{x}^T\begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}\mathbf{x} \tag{3.8}$$

where,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

We also observe that **C** in (3.7) ends up being **H**.

A third form is a Taylor representation,

$$f(\mathbf{x}) = f^k + \left(\nabla f^k\right)^{\mathrm{T}} \Delta\mathbf{x}^k + \frac{1}{2}\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \mathbf{H}\Delta\mathbf{x}^k \tag{3.9}$$

We note for (3.6), $\nabla f = \begin{bmatrix} -2 + 4x_1 + 3x_2 \\ 1 + 3x_1 + 2x_2 \end{bmatrix}$ and $\mathbf{H} = \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix}$

We will assume a point of expansion, $\mathbf{x}^k = \begin{bmatrix} -2 \\ 2 \end{bmatrix}, \nabla f^k = \begin{bmatrix} -4 \\ -1 \end{bmatrix}$. (It may not be apparent, but if we are approximating a quadratic, it doesn't matter what point of expansion we assume. The Taylor expansion will be exact.)

The example in (3.6), as a Taylor representation, becomes,

$$f(\mathbf{x}) = 12 + [-4,\ -1]\Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^T \begin{bmatrix} 4 & 3 \\ 3 & 2 \end{bmatrix} \Delta\mathbf{x} \tag{3.10}$$

where,

$$\Delta\mathbf{x} = \begin{bmatrix} x_1 + 2 \\ x_2 - 2 \end{bmatrix}$$

These three representations are equivalent. If we pick the point $\mathbf{x}^T = [1.0,\quad 2.0]$, all three representations give $f = 18$ at this point, as you can verify by substitution.

2.2. Characteristics of Quadratic Functions

It is useful to note the following characteristics of quadratic equations:

- The equations for the gradient vector of a quadratic function are linear. This makes it easy to solve for where the gradient is equal to zero.
- The Hessian for a quadratic function is a matrix of constants (so we will write as **H** or $\nabla^2 f$ instead of **H**(**x**) or $\nabla^2 f(\mathbf{x})$). Thus the curvature of a quadratic is everywhere the same.
- Excluding the cases where we have a semi-definite Hessian, quadratic functions have only one *stationary point*, i.e. only one point where the gradient is zero.
- Given the gradient and Hessian at some point $\mathbf{x}^k$, the gradient at some other point, $\mathbf{x}^{k+1}$, is given by,

$$\nabla f^{k+1} = \nabla f^k + \mathbf{H}\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) \tag{3.11}$$

This expression is developed in Section 9.1 of the Appendix by differentiating a Taylor expansion in vector form.

- Given the gradient some point $\mathbf{x}^k$, Hessian, **H**, and a search direction, **s**, the optimal step length, α^*, in the direction **s** is given by,

$$\alpha^* = -\frac{\left(\nabla f^k\right)^T \mathbf{s}}{\mathbf{s}^T \mathbf{H} \mathbf{s}} \tag{3.12}$$

This expression is derived in Section 9.2 of the Appendix.

- The best methods of optimization are *methods of conjugate directions*. A method of conjugate directions will solve for the optimum of a quadratic function of *n* variables in *n* steps, providing minimizing steps are taken in each search direction. We will learn more about these methods in sections which follow.

2.3. Examples

We start with the example,

$$f\left(\mathbf{x}\right) = 4x_1 + 2x_2 + x_1^2 - 4x_1x_2 + x_2^2 \tag{3.13}$$

Since this is a quadratic, we know it only has one stationary point. We note that the Hessian,

$$\mathbf{H} = \begin{bmatrix} 2 & -4 \\ -4 & 2 \end{bmatrix}$$

is indefinite (eigenvalues are -2.0 and 6.0). This means we should have a saddle point. The contour plots in Fig 3.2 and Fig. 3.3 confirm this.

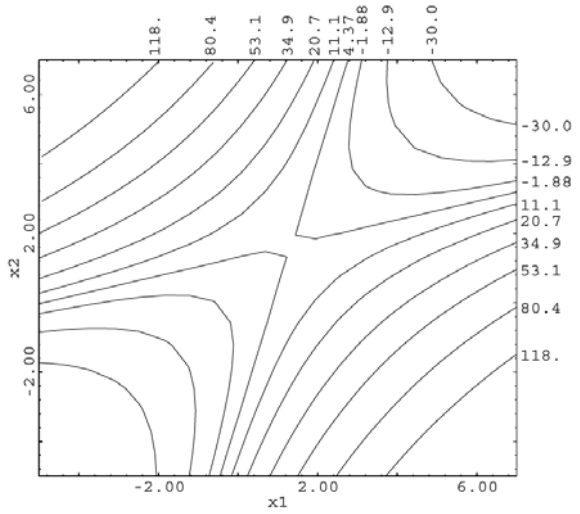

Fig. 3.2 Contour plot of Eq (3.13).

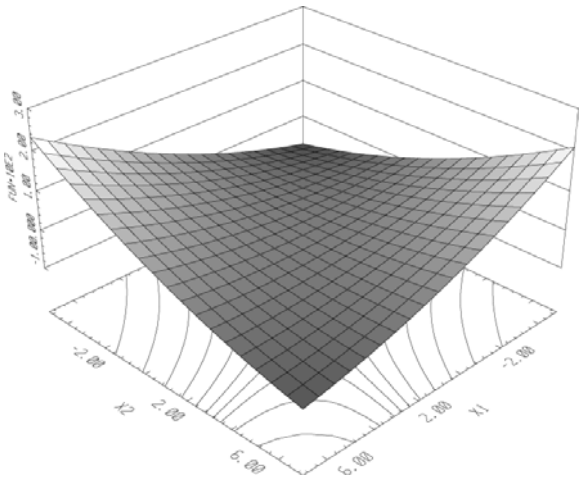

Fig. 3.3. 3D contour plot of (3.13).

We will do a second example. Suppose we have the function,

$$f(\mathbf{x}) = x_1 + 2x_2 + 4x_1^2 - x_1x_2 + 2x_2^2 \tag{3.14}$$

$$\nabla f = \begin{bmatrix} 1+8x_1 - x_2 \\ 2 - x_1 + 4x_2 \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} 8 & -1 \\ -1 & 4 \end{bmatrix}$$

By inspection, we see that the determinants of the principal minor matrices are all positive. Thus this function should have a min and look like a bowl. The contour plots follow.

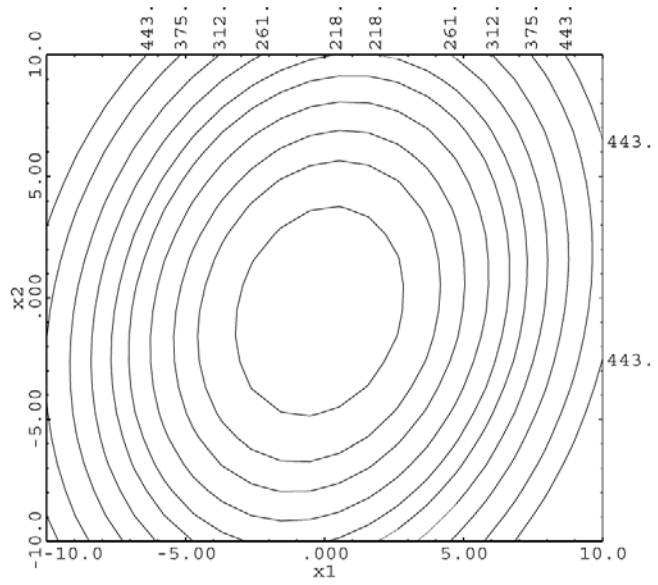

Fig. 3.4. Contour plot for (3.14)

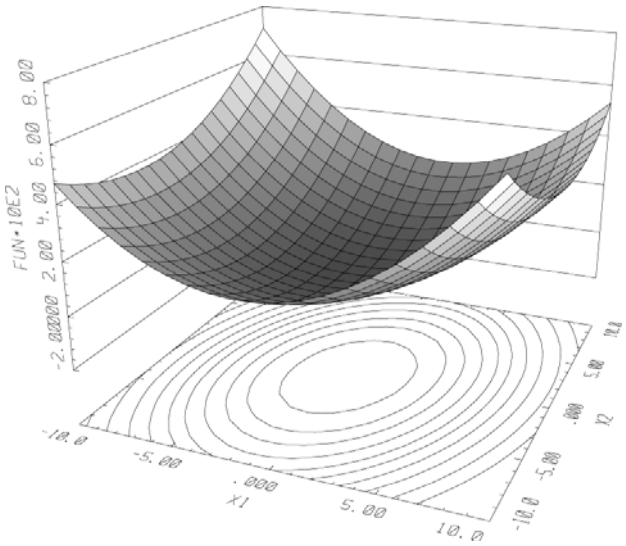

Fig. 3.5 3D contour plot for (3.14)

3. Necessary and Sufficient Conditions for an Unconstrained Optimum

With some preliminaries out of the way, we are now ready to begin discussing the theory of unconstrained optimization of differentiable functions. We start with the mathematical conditions which must hold at an unconstrained, local optimum.

3.1. Definitions

3.1.1. Necessary Conditions for an Unconstrained Optimum

The necessary conditions for an unconstrained optimum at $\mathbf{x}^*$ are,

$$\nabla f(\mathbf{x}^*) = 0 \text{ and } f(\mathbf{x}) \text{ be differentiable at } \mathbf{x}^* \qquad (3.15)$$

These conditions are necessary but not *sufficient*, inasmuch as $\nabla f(\mathbf{x}) = 0$ can apply at a max, min or a saddle point. However, if at a point $\nabla f(\mathbf{x}) \neq 0$, then that point *cannot* be an optimum.

3.1.2. Sufficient Conditions for a Minimum

The sufficient conditions include the necessary conditions but add other conditions such that when satisfied we *know* we have an optimum. For a minimum,

$$\nabla f(\mathbf{x}^*) = 0, \; f(\mathbf{x}) \text{ differentiable at } \mathbf{x}^*, \text{ plus } \nabla^2 f(\mathbf{x}^*) \text{ is positive definite.} \qquad (3.16)$$

3.1.3. Sufficient Conditions for a Maximum

For a maximum,

$\nabla f(\mathbf{x}^*) = 0$, $f(\mathbf{x})$ differentiable at $\mathbf{x}^*$, plus $\nabla^2 f(\mathbf{x}^*)$ is negative definite. (3.17)

3.2. Examples: Applying the Necessary, Sufficient Conditions

Apply the necessary and sufficient conditions to find the optimum for the quadratic function,

$$f(\mathbf{x}) = x_1^2 - 2x_1x_2 + 4x_2^2$$

Since this is a quadratic function, the partial derivatives will be linear equations. We can solve these equations directly for a point that satisfies the necessary conditions. The gradient vector is,

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - 2x_2 \\ -2x_1 + 8x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

When we solve these two equations, we have a solution, $x_1 = 0$, $x_2 = 0$--this a point where the gradient is equal to zero. This represents a minimum, a maximum, or a saddle point. At this point, the Hessian is,

$$\mathbf{H} = \begin{bmatrix} 2 & -2 \\ -2 & 8 \end{bmatrix}$$

Since this Hessian is positive definite (eigenvalues are 1.4, 8.6), this must be a minimum.

As a second example, apply the necessary and sufficient conditions to find the optimum for the quadratic function,

$$f(\mathbf{x}) = 4x_1 + 2x_2 + x_1^2 - 4x_1x_2 + x_2^2$$

As in example 1, we will solve the gradient equations directly for a point that satisfies the necessary conditions. The gradient vector is,

$$\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 - 4x_2 + 4 \\ -4x_1 + 2x_2 + 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

When we solve these two equations, we have a solution, $x_1 = 1.333$, $x_2 = 1.667$. The Hessian is,

$$\mathbf{H} = \begin{bmatrix} 2 & -4 \\ -4 & 2 \end{bmatrix}$$

The eigenvalues are -2, 6. The Hessian is indefinite. This means this is neither a max nor a min—it is a saddle point.

Comments: As mentioned, the equations for the gradient for a quadratic function are linear, so they are easy to solve. Obviously we don't usually have a quadratic objective, so the equations are usually not linear. Often we will use the necessary conditions to *check* a point to see if we are at an optimum. Some algorithms, however, solve for an optimum by solving directly where the gradient is equal to zero. Sequential Quadratic Programming (SQP) is this type of algorithm.

Other algorithms search for the optimum by taking downhill steps and continuing until they can go no further. The GRG (Generalized Reduced Gradient) algorithm is an example of this type of algorithm. In the next section we will study one of the simplest unconstrained algorithms that steps downhill: steepest descent.

4. Steepest Descent with a Quadratic Line Search

4.1. Description

One of the simplest unconstrained optimization methods is steepest descent. Given an initial starting point, the algorithm moves downhill until it can go no further.

The search can be broken down into stages. For any algorithm, at each stage (or iteration) we must determine two things:

1. What should the search direction be?
2. How far should we go in that direction?

Answer to question 1: For the method of steepest descent, the search direction is $-\nabla f(\mathbf{x})$

Answer to question 2: A *line search* is performed. "Line" in this case means we search along a direction vector. The line search strategy presented here, bracketing the function with quadratic fit, is one of many that have been proposed, and is one of the most common.

General Approach for each step:
Given some starting point, $\mathbf{x}^k$, we wish to determine,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \mathbf{s} \tag{3.18}$$

where **s** is the search direction vector, usually normalized, and α is the step length, a scalar.

We will step in direction **s** with increasing values of α until the function starts to get worse. Then we will curve fit the data with a parabola, and step to the minimum of the parabola.

4.2. Example: Steepest Descent with Line Search

Min $f(\mathbf{x}) = x_1^2 - 2x_1x_2 + 4x_2^2$ $\qquad f^0 = 19$

starting at $\mathbf{x}^0 = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$ $\qquad -\nabla f^0 = \begin{bmatrix} 8 \\ -14 \end{bmatrix}$ $\quad \mathbf{s}^0 = \begin{bmatrix} 8 \\ -14 \end{bmatrix}$

normalized $\mathbf{s}^0 = \begin{bmatrix} 0.50 \\ -0.86 \end{bmatrix}$ $\qquad \mathbf{x}^1 = \begin{bmatrix} -3 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 0.50 \\ -0.86 \end{bmatrix}$

We will find α^*, which is the optimal step length, by trial and error.

Guess $\alpha^* = .4$ for step number 1:

Line Search Step	α	$\mathbf{x}^1 = \mathbf{x}^0 + \alpha \mathbf{s}^0$	$f(\mathbf{x})$
1	0.4	$\mathbf{x}^1 = \begin{bmatrix} -3.0 \\ 1.0 \end{bmatrix} + .4 \begin{bmatrix} 0.50 \\ -0.86 \end{bmatrix} = \begin{bmatrix} -2.80 \\ 0.66 \end{bmatrix}$	13.3

We see that the function has decreased; we decide to double the step length and continue doubling until the function begins to increase:

Line Search Step	α	$\mathbf{x}^1 = \mathbf{x}^0 + \alpha \mathbf{s}^0$	$f(\mathbf{x})$
2	0.8	$\mathbf{x}^1 = \begin{bmatrix} -3.0 \\ 1.0 \end{bmatrix} + .8 \begin{bmatrix} 0.50 \\ -0.86 \end{bmatrix} = \begin{bmatrix} -2.60 \\ 0.31 \end{bmatrix}$	8.75
3	1.6	$\mathbf{x}^1 = \begin{bmatrix} -3.0 \\ 1.0 \end{bmatrix} + 1.6 \begin{bmatrix} 0.50 \\ -0.86 \end{bmatrix} = \begin{bmatrix} -2.20 \\ -0.38 \end{bmatrix}$	3.74
4	3.2	$\mathbf{x}^1 = \begin{bmatrix} -3.0 \\ 1.0 \end{bmatrix} + 3.2 \begin{bmatrix} 0.50 \\ -0.86 \end{bmatrix} = \begin{bmatrix} -1.40 \\ -1.75 \end{bmatrix}$	9.31

The objective function has started to increase; therefore we have gone too far.

We will cut the change in the last step by half:

5	2.4	$\mathbf{x}^1 = \begin{bmatrix} -3.0 \\ 1.0 \end{bmatrix} + 2.4 \begin{bmatrix} 0.50 \\ -0.86 \end{bmatrix} = \begin{bmatrix} -1.80 \\ -1.06 \end{bmatrix}$	3.91

A graph of our progress is shown in Fig. 3.6:

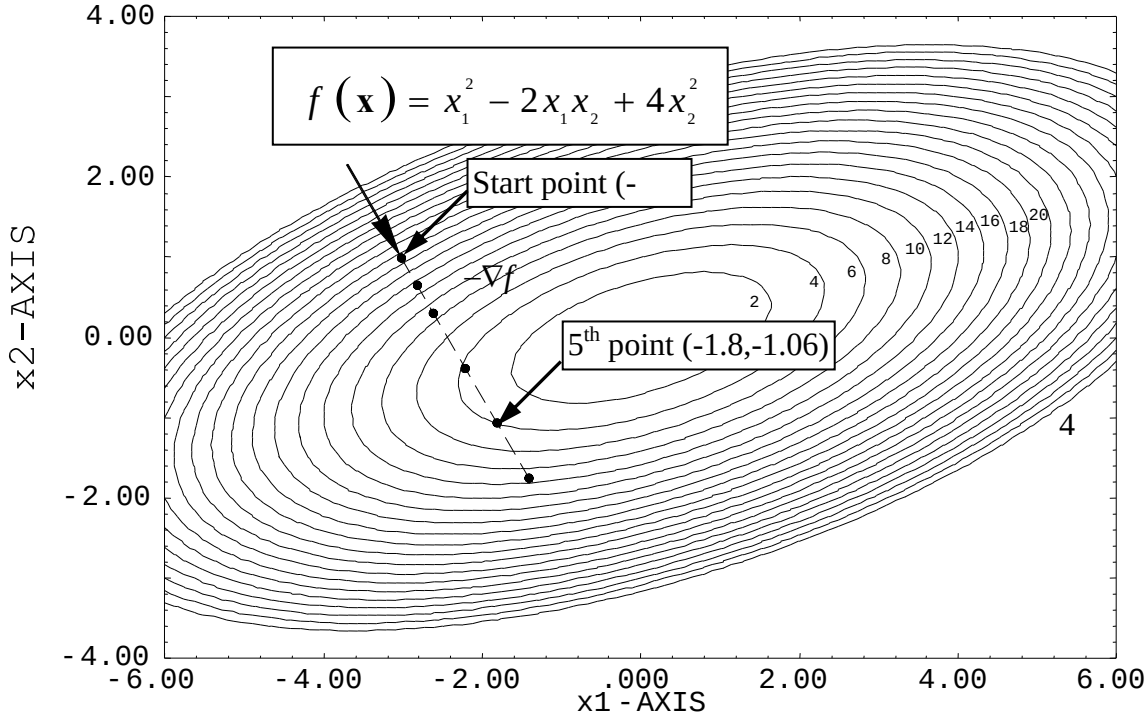

Fig. 3.6 Progress in the line search shown on a contour plot.

If we plot the objective value as a function of step length as shown in Fig 3.7:

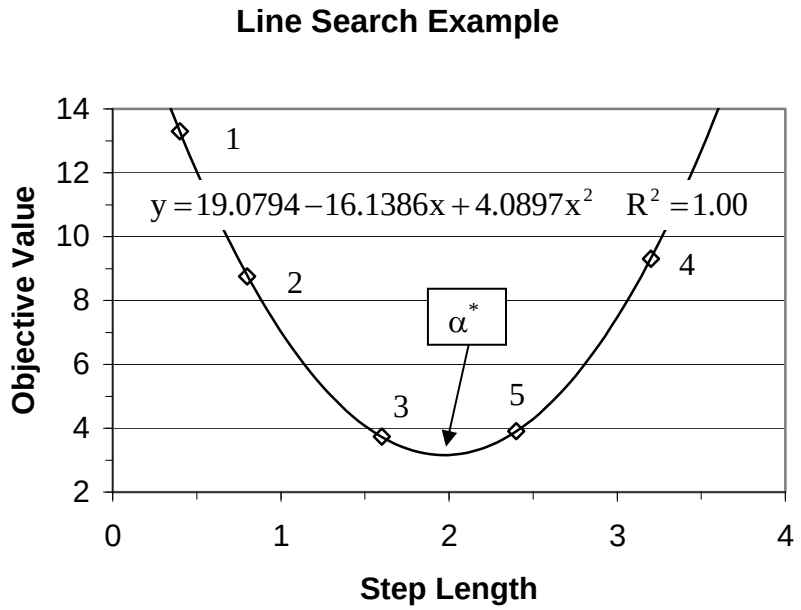

Fig. 3.7 The objective value vs. step length for the line search.

We see that the data plot up to be a parabola. We would like to estimate the minimum of this curve. We will curve fit points 2, 5, 3. These points are equally spaced and bracket the minimum.

$$\|2 \quad \|3 \quad \|5$$

Renumbering these points as α_1, α_2, α_3 the minimum of the parabola is given by

$$\alpha^* = \alpha_2 + \frac{\Delta\alpha\left[f(\alpha_1) - f(\alpha_3)\right]}{2\left[f(\alpha_1) - 2f(\alpha_2) + f(\alpha_3)\right]}$$

$$\alpha^* = 1.60 + \frac{(0.8)\left[8.75 - 3.91\right]}{2\left[8.75 - 2(3.74) + 3.91\right]} \qquad (3.19)$$

$$\alpha^* = 1.97$$

where $f(\mathbf{x}) = 3.2$

When we step back, after the function has become worse, we have four points to choose from (points 2, 3, 5, 4). How do we know which three to pick to make sure we don't lose the bracket on the minimum? The rule is this: take the point with the lowest function value (point 3) and the two points to either side (points 2 and 5).

In summary, the line search consists of stepping along the search direction until the minimum of the function in this direction is bracketed, fitting three points which bracket the minimum with a parabola, and calculating the minimum of the parabola. If necessary the parabolic fit can be carried out several times until the change in the minimum is very small (although the α are then no longer equally spaced, so the following formula must be used):

$$\alpha^* = \frac{f(\alpha_1)\left(\alpha_2^2 - \alpha_3^2\right) + f(\alpha_2)\left(\alpha_3^2 - \alpha_1^2\right) + f(\alpha_3)\left(\alpha_1^2 - \alpha_2^2\right)}{2\left[f(\alpha_1)(\alpha_2 - \alpha_3) + f(\alpha_2)(\alpha_3 - \alpha_1) + f(\alpha_3)(\alpha_1 - \alpha_2)\right]} \qquad (3.20)$$

Each sequence of obtaining the gradient and moving along the negative gradient direction until a minimum is found (i.e. executing a line search) is called an *iteration.* The algorithm consists of executing iterations until the norm of the gradient drops below a specified tolerance, indicating the necessary conditions have been met.

As shown in Fig. 3.7, at α^*, $\frac{df}{d\alpha} = 0$. The process of determining α^* will be referred to as *taking a minimizing step,* or, *executing an exact line search.*

4.3. Pros and Cons of Steepest Descent

Steepest descent has several advantages. It usually makes good progress when far from the optimum (in the above example the objective decreased from 19 to 3 in the first iteration), and it is very simple to implement. It always goes downhill. It is also guaranteed to converge to a local optimum if enough steps are taken.

However, if the function to be minimized is *eccentric*, convergence of steepest descent can be very slow, as indicated by the following theorem from Luenberger.[1]

THEOREM. Convergence of Steepest Descent. For a quadratic function, if we take enough steps, the method of steepest descent converges to the unique minimum point $\mathbf{x}^*$ of f. If we define the error in the objective function at the current value of $\mathbf{x}$ as,

$$E(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^*\right)^{\mathrm{T}} \mathbf{H}\left(\mathbf{x}-\mathbf{x}^*\right) \tag{3.21}$$

there holds at every step k,

$$E\left(\mathbf{x}^{k+1}\right) \le \left(\frac{A-a}{A+a}\right)^2 E\left(\mathbf{x}^k\right)$$

where (3.22)

$A = \text{Largest eigenvalue of } \mathbf{H}$

$a = \text{Smallest eigenvalue of } \mathbf{H}$

Thus if A=50 and a=1, we have that the error at the k+1 step is only guaranteed to be less than the error at the k step by,

$$E^{k+1} \le \left(\frac{49}{51}\right)^2 E^k$$

and thus the error may be reduced very slowly.

"Roughly speaking, the above theorem says that the convergence rate of steepest descent is slowed as the contours of f become more eccentric. If $a = A$, corresponding to circular contours, convergence occurs in a single step. Note, however, that even if $n-1$ of the n eigenvalues are equal and the remaining one is a great distance from these, convergence will be slow, and hence a single abnormal eigenvalue can destroy the effectiveness of steepest descent."

The above theorem is based on a quadratic function. If we have a quadratic, and we do rotation and translation of the axes, we can eliminate all of the linear and cross product terms. We then have only the pure second order terms left. The eigenvalues of the resulting Hessian are equal to twice the coefficients of the pure second order terms. Thus the function,

$$f = x_1^2 + x_2^2$$

would have equal eigenvalues of (2, 2) and would represent the circular contours as mentioned above, shown in Fig. 3.8. Steepest descent would converge in one step. Conversely the function,

[1] Luenberger and Ye, *Linear and Nonlinear Programming, Third Edition*, 2008

$$f = 50x_1^2 + x_2^2$$

has eigenvalues of (100, 2). The contours would be highly eccentric and convergence of steepest descent would be very slow. A contour plot of this function is given in Fig 3.9,

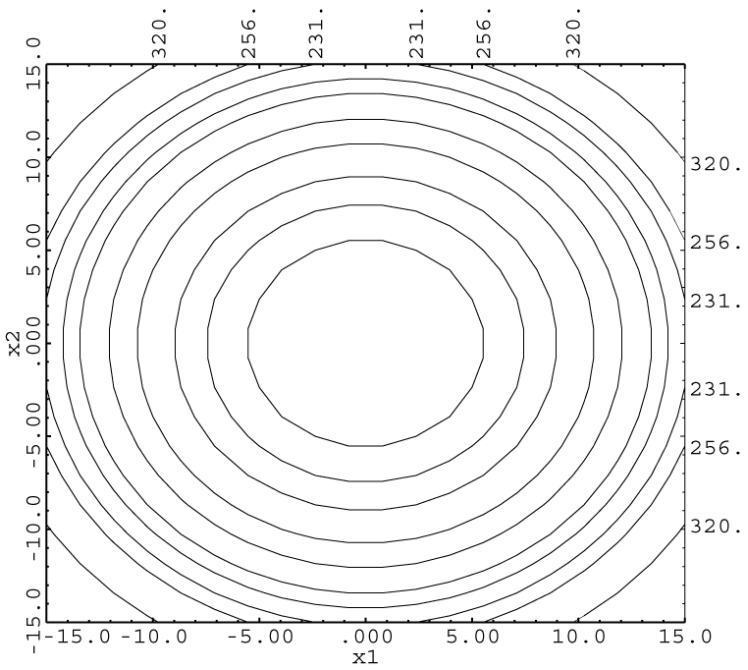

Fig. 3.8. Contours of the function, $f = x_1^2 + x_2^2$.

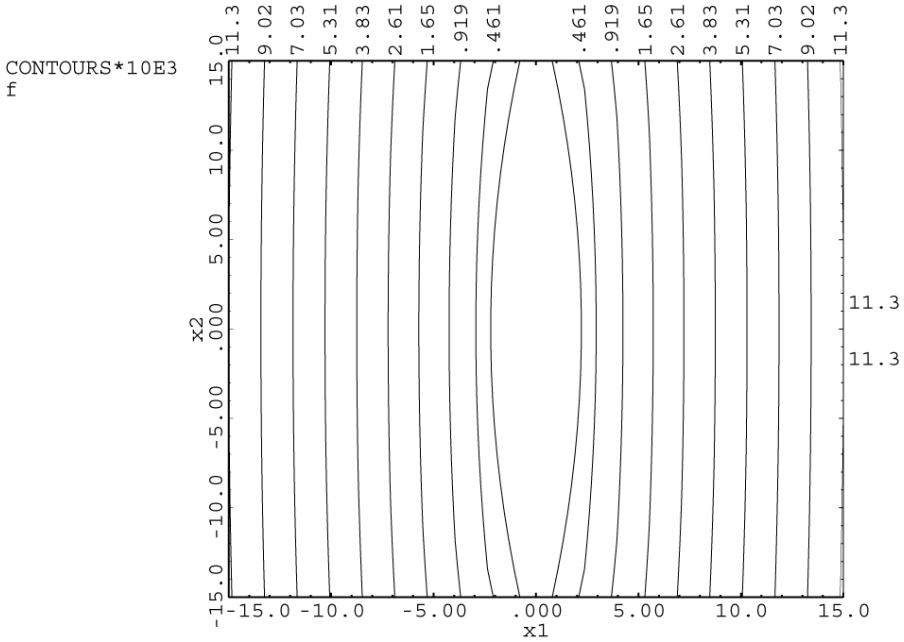

Fig. 3.9. Contours of the function, $f = 50x_1^2 + x_2^2$. Notice how the contours have been "stretched" out.

5. The Directional Derivative

It is sometimes useful to calculate $\frac{df}{d\alpha}$ along some search direction **s.** From the chain rule for differentiation,

$$\frac{df}{d\alpha} = \sum\left(\frac{\partial f}{\partial x_i}\right)\left(\frac{dx_i}{d\alpha}\right)$$

Noting that $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha\mathbf{s}$, or, for a single element of vector **x**, $x_i^{k+1} = x_i^k + \alpha s_i^k$, we have $\frac{dx_i}{d\alpha} = s_i$, so

$$\frac{df}{d\alpha} = \sum\left(\frac{\partial f}{\partial x_i}\right)\left(\frac{dx_i}{d\alpha}\right) = \sum\left(\frac{\partial f}{\partial x_i}\right)s_i = \nabla f^{\mathrm{T}}\mathbf{s} \tag{3.23}$$

As an example, we will find the directional derivative, $\frac{df}{d\alpha}$, for the problem given in Section 4.2 above, at α=0. From (3.23): $\frac{df}{d\alpha} = \nabla f^{\mathrm{T}}\mathbf{s} = \begin{bmatrix}-8 & 14\end{bmatrix}\begin{bmatrix}0.5\\-0.86\end{bmatrix} = -16.04$

This gives us the change in the function for a small step in the search direction, i.e.,

$$\Delta f \approx \frac{df}{d\alpha}\Delta\alpha \tag{3.24}$$

If $\Delta\alpha = 0.01$, the predicted change is 0.1604. The actual change in the function is 0.1599.

Equation (3.23) is the same equation for checking if a direction goes downhill, given in Section 1.4. Before we just looked at the sign; if negative we knew we were going downhill. Now we see that the value has meaning as well: it represents the expected change in the function for a small step. If, for example, the value of $\left.\frac{df}{d\alpha}\right|_{\alpha=0}$ is less than some epsilon, we could terminate the line search, because the predicted change in the objective function is below a minimum threshold.

Another important value of $\frac{df}{d\alpha}$ occurs at α^*. If we locate the minimum exactly, then

$$\left.\frac{df}{d\alpha}\right|_{\alpha=\alpha^*} = \left(\nabla f^{k+1}\right)^{\mathrm{T}}\mathbf{s}^k = 0 \tag{3.25}$$

As we have seen in examples, when we take a minimizing step we stop where the search direction is tangent to the contours of the function. Thus the gradient at this new point is orthogonal to the previous search direction.

6. Newton's Method

6.1. Derivation

Another classical method we will briefly study is called Newton's method. It simply makes a quadratic approximation to a function at the current point and solves for where the necessary conditions (to the approximation) are satisfied. Starting with a Taylor series:

$$f^{k+1} = f^k + \left(\nabla f^k\right)^{\mathrm{T}} \Delta\mathbf{x}^k + \frac{1}{2}\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \mathbf{H}^k \Delta\mathbf{x}^k \tag{3.26}$$

Since the gradient and Hessian are evaluated at *k*, they are just a vector and matrix of constants. Taking the gradient (Section 9.1),

$$\nabla f^{k+1} = \nabla f^k + \mathbf{H}^k \Delta\mathbf{x}^k$$

and setting $\nabla f^{k+1} = 0$, we have,

$$\mathbf{H}^k \Delta\mathbf{x}^k = -\nabla f^k$$

Solving for $\Delta\mathbf{x}$:

$$\Delta\mathbf{x}^k = -\left(\mathbf{H}^k\right)^{-1} \nabla f^k \tag{3.27}$$

Note that we have solved for a vector, i.e. $\Delta\mathbf{x}$, which has both a step length and direction.

6.2. Example: Newton's Method

We wish to optimize the function, $f(\mathbf{x}) = x_1^2 - 2x_1x_2 + 4x_2^2$ from the point $\mathbf{x}^0 = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$.

At this point $\nabla f^0 = \begin{bmatrix} -8 \\ 14 \end{bmatrix}$ and the Hessian is, $\mathbf{H} = \begin{bmatrix} 2 & -2 \\ -2 & 8 \end{bmatrix}$. The Hessian inverse is given by: $\begin{bmatrix} 0.6667 & 0.16667 \\ 0.16667 & 0.16667 \end{bmatrix}$. Thus $\Delta\mathbf{x} = -\begin{bmatrix} 0.6667 & 0.16667 \\ 0.16667 & 0.16667 \end{bmatrix}\begin{bmatrix} -8 \\ 14 \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}$

So, $\mathbf{x}^1 = \mathbf{x}^0 + \Delta\mathbf{x} = \begin{bmatrix} -3 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

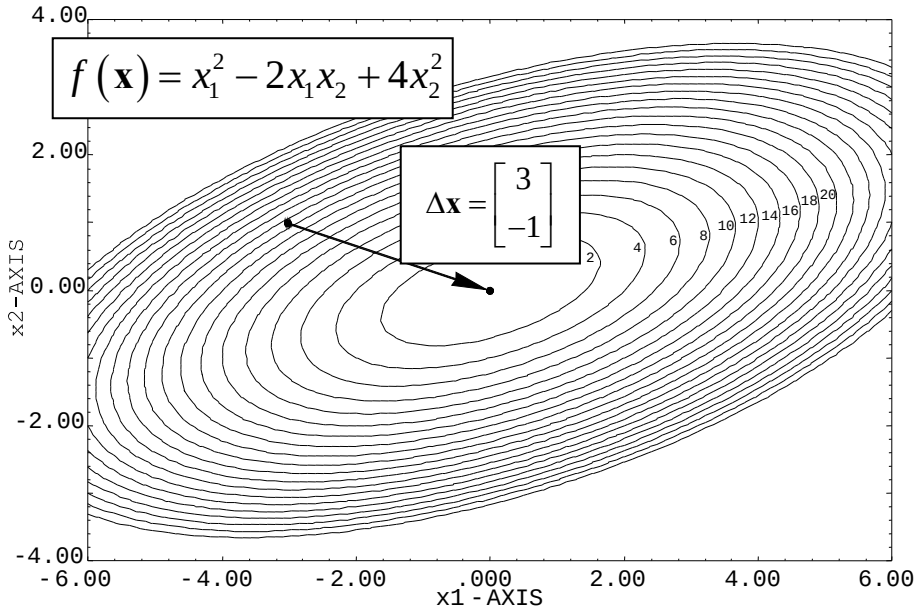

Fig. 3.10. The operation of Newton's method.

6.3. Pros and Cons of Newton's Method

We can see in the above example Newton's method solved the problem in one step. This is true in general: Newton's method will drive to the stationary point of a quadratic in one step. On a non-quadratic, if we are near an optimum, Newton's method will drive very close to the optimum in one step.

However we should note some drawbacks. First, it requires second derivatives. Normally we compute derivatives numerically, and computing second derivatives is computationally expensive, on the order of n^2 function evaluations, where *n* is the number of design variables.

The derivation of Newton's method solved for where the gradient is equal to zero. The gradient is equal to zero at a min, a max or a saddle, and nothing in the method differentiates between these. Thus Newton's method can *diverge*, or fail to go downhill (indeed, not only not go downhill, but go to a maximum!). This is obviously a serious drawback.

7. Quasi-Newton Methods

7.1. Introduction

Let's summarize the pros and cons of Newton's method and Steepest Descent:

	Pros	Cons
Steepest Descent	Always goes downhill Always converges Simple to implement	Slow on eccentric functions
Newton's Method	Solves quadratic in one step. Very fast when close to optimum on non quadratic.	Requires second derivatives, Can diverge

We want to develop a method that starts out like steepest descent and gradually becomes Newton's method, doesn't need second derivatives, doesn't have trouble with eccentric functions and doesn't diverge! Fortunately such methods exist. They combine the good aspects of steepest descent and Newton's method without the drawbacks. These methods are called *quasi-Newton* methods or sometimes *variable metric* methods.

In general we will define our search direction by the expression

$$\mathbf{s} = -\mathbf{N}\nabla f(\mathbf{x}) \tag{3.28}$$

where **N** will be called the "direction matrix."

If $\mathbf{N} = \mathbf{I}$, then $\mathbf{s} = -\nabla f(\mathbf{x}) \rightarrow$ Steepest Descent

If $\mathbf{N} = \mathbf{H}^{-1}$, then $\mathbf{s} = -\mathbf{H}^{-1}\nabla f(\mathbf{x}) \rightarrow$ Newton's Method

If **N** is always positive definite, then **s** always points downhill. To show this, our criterion for moving downhill is:

$$\mathbf{s}^{\mathrm{T}}\nabla f < 0$$

Or,

$$\nabla f^{\mathrm{T}}\mathbf{s} < 0 \tag{3.29}$$

Substituting (3.28) into (3.29):

$$-\left(\nabla f^{\mathrm{T}}\mathbf{N}\nabla f\right) < 0 \tag{3.30}$$

Since **N** is positive definite, we know that any vector which pre-multiplies **N** and post-multiplies **N** will result in a positive scalar. Thus the quantity within the parentheses is always positive; with the negative sign it becomes always negative, and therefore always goes downhill.

7.2. A Rank One Hessian Inverse Update

7.2.1. Development

In this section we will develop one of the simplest updates, called a "rank one" update because the correction to the direction matrix, **N**, is a rank one matrix (i.e., it only has one independent row or column). We first start with some preliminaries.

Starting with a Taylor series:

$$f^{k+1} = f^{k} + \left(\nabla f^{k}\right)^{\mathrm{T}} \Delta\mathbf{x}^{k} + \frac{1}{2}\left(\Delta\mathbf{x}^{k}\right)^{\mathrm{T}} \mathbf{H}\, \Delta\mathbf{x}^{k} \tag{3.31}$$

where $\Delta\mathbf{x}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$

the gradient is given by,

$$\nabla f^{k+1} = \nabla f^k + \mathbf{H}\Delta\mathbf{x}^k \tag{3.32}$$

and defining:

$$\boldsymbol{\gamma}^k = \nabla f^{k+1} - \nabla f^k \tag{3.33}$$

we have,

$$\boldsymbol{\gamma}^k = \mathbf{H}\Delta\mathbf{x}^k \quad or \quad \mathbf{H}^{-1}\boldsymbol{\gamma}^k = \Delta\mathbf{x}^k \tag{3.34}$$

Equation (3.34) is very important: it shows that for a quadratic function, the inverse of the Hessian matrix ($\mathbf{H}^{-1}$) maps differences in the gradients to differences in **x**. The relationship expressed by (3.34) is called the *Newton condition*.

We will make the direction matrix satisfy this relationship. However, since we can only calculate $\boldsymbol{\gamma}^k$ and $\Delta\mathbf{x}^k$ after the line search, we will make

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^k = \Delta\mathbf{x}^k \tag{3.35}$$

This expression is sometimes called the *quasi-Newton condition*. It is "quasi" in that it involves $k+1$ for **N** instead of k. Equation (3.35) involves more unknowns (the elements of $\mathbf{N}^{k+1}$) than equations, so how do we solve for $\mathbf{N}^{k+1}$?

One of the simplest possibilities is:

$$\mathbf{N}^{k+1} = \mathbf{N}^k + a\mathbf{u}\mathbf{u}^{\mathrm{T}} \tag{3.36}$$

Where we will "update" the direction matrix with a correction which is of the form $a\mathbf{u}\mathbf{u}^{\mathrm{T}}$, which is a rank one symmetric matrix.

If we substitute (3.36) into (3.35), we have,

$$\mathbf{N}^k\boldsymbol{\gamma}^k + a\mathbf{u}\mathbf{u}^{\mathrm{T}}\boldsymbol{\gamma}^k = \Delta\mathbf{x}^k \tag{3.37}$$

or

$$a\mathbf{u}\underbrace{\mathbf{u}^{\mathrm{T}}\boldsymbol{\gamma}^k}_{scalar} = \left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right) \tag{3.38}$$

Noting that $\mathbf{u}^{\mathrm{T}}\boldsymbol{\gamma}^k$ is a scalar, then **u** must be proportional to $\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)$. Since any change in length can be absorbed by a, we will set

$$\mathbf{u} = \left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right) \tag{3.39}$$

Substituting (3.39) into (3.38):

$$a\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)\underbrace{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k}_{scalar} = \left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right) \tag{3.40}$$

For this to be true,

$$a\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k = 1$$

so

$$a = \frac{1}{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k} \tag{3.41}$$

Substituting (3.41) and (3.39) into (3.36) gives the expression we need:

$$\mathbf{N}^{k+1} = \mathbf{N}^k + \frac{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}}{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k} \tag{3.42}$$

Equation (3.42) allows us to get a new direction matrix in terms of the previous matrix and the difference in **x** and the gradient. We then use this to get a new search direction according to (3.28).

7.2.2. Example: Rank One Hessian Inverse Update

We wish to minimize the function $f(\mathbf{x}) = x_1^2 - 2x_1x_2 + 4x_2^2$

starting from $\mathbf{x}^0 = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$ $\nabla f^0 = \begin{bmatrix} -8 \\ 14 \end{bmatrix}$

We let $\mathbf{N}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ so the search direction is

$$\mathbf{s}^0 = -\mathbf{N}\nabla f^0 = -\nabla f^0$$

We normalize the search direction to be: $\mathbf{s}^0 = \begin{bmatrix} 0.496 \\ -0.868 \end{bmatrix}$

We execute a line search in this direction (using, for example, a quadratic fit) and stop at

$$\mathbf{x}^1 = \begin{bmatrix} -2.030 \\ -0.698 \end{bmatrix} \quad \nabla f^1 = \begin{bmatrix} -2.664 \\ -1.522 \end{bmatrix}$$

Then $$\Delta\mathbf{x}^0 = \mathbf{x}^1 - \mathbf{x}^0 = \begin{bmatrix} -2.030 \\ -0.698 \end{bmatrix} - \begin{bmatrix} -3.000 \\ 1.000 \end{bmatrix} = \begin{bmatrix} 0.970 \\ -1.698 \end{bmatrix}$$

$$\boldsymbol{\gamma}^0 = \nabla f^1 - \nabla f^0 = \begin{bmatrix} -2.664 \\ -1.522 \end{bmatrix} - \begin{bmatrix} -8.000 \\ 14.000 \end{bmatrix} = \begin{bmatrix} 5.336 \\ -15.522 \end{bmatrix}$$

and $$\Delta\mathbf{x}^0 - \mathbf{N}^0\boldsymbol{\gamma}^0 = \begin{bmatrix} 0.970 \\ -1.698 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 5.336 \\ -15.522 \end{bmatrix} = \begin{bmatrix} -4.366 \\ 13.824 \end{bmatrix}$$

$$a\mathbf{u}\mathbf{u}^T = \frac{\left(\Delta\mathbf{x}^0 - \mathbf{N}^0\boldsymbol{\gamma}^0\right)\left(\Delta\mathbf{x}^0 - \mathbf{N}^0\boldsymbol{\gamma}^0\right)^T}{\left(\Delta\mathbf{x}^0 - \mathbf{N}^0\boldsymbol{\gamma}^0\right)^T\boldsymbol{\gamma}^0} = \frac{\begin{bmatrix} -4.366 \\ 13.824 \end{bmatrix}\begin{bmatrix} -4.366 & 13.824 \end{bmatrix}}{\begin{bmatrix} -4.366 & 13.824 \end{bmatrix}\begin{bmatrix} 5.336 \\ -15.522 \end{bmatrix}}$$

$$= \frac{\begin{bmatrix} 19.062 & -60.364 \\ -60.364 & 191.158 \end{bmatrix}}{-237.932}$$

$$= \begin{bmatrix} -0.080 & 0.254 \\ 0.254 & -0.803 \end{bmatrix}$$

$$\mathbf{N}^1 = \mathbf{N}^0 + a\mathbf{u}\mathbf{u}^T$$

$$\mathbf{N}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -0.080 & 0.254 \\ 0.254 & -0.803 \end{bmatrix}$$

$$\mathbf{N}^1 = \begin{bmatrix} 0.920 & 0.254 \\ 0.254 & 0.197 \end{bmatrix}$$

New search direction:

$$\mathbf{s}^1 = -\begin{bmatrix} 0.920 & 0.254 \\ 0.254 & 0.197 \end{bmatrix}\begin{bmatrix} -2.664 \\ -1.522 \end{bmatrix}$$

$$= \begin{bmatrix} 2.837 \\ 0.975 \end{bmatrix}$$

When we step in this direction, using again a line search, we arrive at the optimum

$$\mathbf{x}^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \nabla f\left(\mathbf{x}^2\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

At this point we are done. However, if we update the direction matrix one more time, we find it has become the inverse Hessian.

$$\Delta\mathbf{x}^1 = \mathbf{x}^2 - \mathbf{x}^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -2.030 \\ -0.698 \end{bmatrix} = \begin{bmatrix} 2.030 \\ 0.698 \end{bmatrix}$$

$$\boldsymbol{\gamma}^1 = \nabla f^2 - \nabla f^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -2.664 \\ -1.524 \end{bmatrix} = \begin{bmatrix} 2.664 \\ 1.524 \end{bmatrix}$$

$$\left(\Delta\mathbf{x}^1 - \mathbf{N}^1\boldsymbol{\gamma}^1\right) = \begin{bmatrix} 2.030 \\ 0.698 \end{bmatrix} - \begin{bmatrix} 0.920 & 0.254 \\ 0.254 & 0.197 \end{bmatrix}\begin{bmatrix} 2.664 \\ 1.524 \end{bmatrix}$$

$$= \begin{bmatrix} 2.030 \\ 0.698 \end{bmatrix} - \begin{bmatrix} 2.838 \\ 0.977 \end{bmatrix} = \begin{bmatrix} -0.808 \\ -0.279 \end{bmatrix}$$

$$a\mathbf{u}\mathbf{u}^{\mathrm{T}} = \frac{\left(\Delta\mathbf{x}^1 - \mathbf{N}^1\boldsymbol{\gamma}^1\right)\left(\Delta\mathbf{x}^1 - \mathbf{N}^1\boldsymbol{\gamma}^1\right)^{\mathrm{T}}}{\left(\Delta\mathbf{x}^1 - \mathbf{N}^1\boldsymbol{\gamma}^1\right)^{\mathrm{T}}\boldsymbol{\gamma}^1} = \frac{\begin{bmatrix} -0.808 \\ -0.279 \end{bmatrix}\begin{bmatrix} -0.808 & -0.279 \end{bmatrix}}{\begin{bmatrix} -0.808 & -0.279 \end{bmatrix}\begin{bmatrix} 2.664 \\ 1.524 \end{bmatrix}} = \begin{bmatrix} -0.253 & -0.088 \\ -0.088 & -0.030 \end{bmatrix}$$

$$\mathbf{N}^2 = \mathbf{N}^1 + a\mathbf{u}\mathbf{u}^{\mathrm{T}} = \begin{bmatrix} 0.920 & 0.254 \\ 0.254 & 0.197 \end{bmatrix} + \begin{bmatrix} -0.253 & -0.088 \\ -0.088 & -0.030 \end{bmatrix} = \begin{bmatrix} 0.667 & 0.166 \\ 0.166 & 0.167 \end{bmatrix} = \mathbf{H}^{(-1)}$$

7.2.3. The Hereditary Property

The hereditary property is an important property of all update methods. The hereditary property states that not only will $\mathbf{N}^{k+1}$ satisfy (3.35) , but

$$\begin{aligned} \mathbf{N}^{k+1}\boldsymbol{\gamma}^k &= \Delta\mathbf{x}^k \\ \mathbf{N}^{k+1}\boldsymbol{\gamma}^{k-1} &= \Delta\mathbf{x}^{k-1} \\ \mathbf{N}^{k+1}\boldsymbol{\gamma}^{k-2} &= \Delta\mathbf{x}^{k-2} \\ \mathbf{N}^{k+1}\boldsymbol{\gamma}^{k-n+1} &= \Delta\mathbf{x}^{k-n+1} \end{aligned} \qquad (3.43)$$

where *n* is the number of variables. That is, (3.35) is not only satisfied for the current step, but for the *last n-1 steps.* Why is this significant? Let's write this relationship of (3.43) as follows:

$$\mathbf{N}^{k+1}\left[\boldsymbol{\gamma}^k, \boldsymbol{\gamma}^{k-1}, \boldsymbol{\gamma}^{k-2} \ \ldots \ \boldsymbol{\gamma}^{k-n+1}\right] = \left[\Delta\mathbf{x}^k, \Delta\mathbf{x}^{k-1}, \Delta\mathbf{x}^{k-2}, \ldots, \Delta\mathbf{x}^{k-n+1}\right]$$

Let the matrix defined by the columns of $\boldsymbol{\gamma}$ be denoted by **G**, and the matrix defined by columns of $\Delta\mathbf{x}$ be denoted by $\Delta\mathbf{X}$. Then,

$$\mathbf{N}^{k+1}\mathbf{G} = \Delta\mathbf{X}$$

If $\boldsymbol{\gamma}^k \ldots \boldsymbol{\gamma}^{k-n+1}$ are independent, and if we have *n* vectors, i.e. **G** is a square matrix, then the inverse for **G** exists and is unique and

$$\mathbf{N}^{k+1} = \Delta\mathbf{X}\mathbf{G}^{-1} \tag{3.44}$$

is uniquely defined.

Since the Hessian inverse satisfies (3.44) for a quadratic function, then we have the important result that, *after n updates the direction matrix becomes the Hessian inverse for a quadratic function*. This implies the quasi-Newton method will solve a quadratic in no more than *n+1* steps. The proof that our rank one update has the hereditary property is given in the next section.

7.2.4. Proof of the Hereditary Property for the Rank One Update

THEOREM. Let **H** be a constant symmetric matrix and suppose that $\Delta\mathbf{x}^0, \Delta\mathbf{x}^1, \ldots, \Delta\mathbf{x}^k$ and $\boldsymbol{\gamma}^0, \boldsymbol{\gamma}^1, \ldots, \boldsymbol{\gamma}^k$ are given vectors, where $\boldsymbol{\gamma}^i = \mathbf{H}\Delta\mathbf{x}^i, \quad i = 0,1,2,\ldots,k$, where $k < n$. Starting with any initial symmetric matrix $\mathbf{N}^0$, let

$$\mathbf{N}^{k+1} = \mathbf{N}^k + \frac{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}}{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k} \tag{3.45}$$

then

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \Delta\mathbf{x}^i \quad \text{for} \quad i \le k \tag{3.46}$$

PROOF. The proof is by induction. We will show that if (3.46) holds for previous direction matrix, it holds for the current direction matrix. We know that at the current point, *k*, the following is true,

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^k = \Delta\mathbf{x}^k \tag{3.47}$$

because we enforced this condition when we developed the update. Now, *suppose* it is true that,

$$\mathbf{N}^k\boldsymbol{\gamma}^i = \Delta\mathbf{x}^i \quad \text{for} \quad i \le k\text{-}1 \tag{3.48}$$

i.e. that the hereditary property holds for the *previous* direction matrix. We can post multiply (3.45) by $\boldsymbol{\gamma}^i$, giving,

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \mathbf{N}^k\boldsymbol{\gamma}^i + \frac{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^i}{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k} \tag{3.49}$$

To simplify things, let $\mathbf{y}^k = \dfrac{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)}{\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k}$ so that we can write (3.49) as,

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \mathbf{N}^k\boldsymbol{\gamma}^i + \mathbf{y}^k\left(\Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^i \tag{3.50}$$

We can distribute the transpose on the last term, and distribute the post multiplication $\boldsymbol{\gamma}^i$ to give (Note: Recall that when you take the transpose inside a product, the order of the product is reversed; also because **N** is symmetric, $\mathbf{N}^T = \mathbf{N}$ thus: $\left(\mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^i = \left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\mathbf{N}^k\boldsymbol{\gamma}^i$),

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \mathbf{N}^k\boldsymbol{\gamma}^i + \mathbf{y}^k\left[\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^i - \left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\mathbf{N}^k\boldsymbol{\gamma}^i\right] \tag{3.51}$$

Since we have assumed (3.48) is true, we can replace $\mathbf{N}^k\boldsymbol{\gamma}^i$ with $\Delta\mathbf{x}^i$:

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \Delta\mathbf{x}^i + \mathbf{y}^k\left[\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^i - \left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\Delta\mathbf{x}^i\right] \tag{3.52}$$

Now we examine the term in brackets. We note that,

$$\left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\Delta\mathbf{x}^i = \left(\mathbf{H}\Delta\mathbf{x}^k\right)^{\mathrm{T}}\Delta\mathbf{x}^i = \left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\mathbf{H}\Delta\mathbf{x}^i = \left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^i \tag{3.53}$$

So the term in brackets in (3.52) vanishes, giving,

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \Delta\mathbf{x}^i \quad \textit{for } i \le k \tag{3.54}$$

Thus, if the hereditary property holds for the previous direction matrix, it holds for the current direction matrix. When $k = 0$, condition (3.47) is all that is needed to have the hereditary property for the first update, $\mathbf{N}^1$. The second update, $\mathbf{N}^2$, will then have the hereditary property since $\mathbf{N}^1$ does, and so on.

7.3. Conjugacy

7.3.1. Definition

Quasi-Newton methods are also methods of *conjugate directions.* A set of search directions, $\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^k$ are said to be *conjugate* with respect to a square, symmetric matrix, **H**, if,

$$\left(\mathbf{s}^k\right)^T\mathbf{H}\mathbf{s}^i = 0 \quad \text{for all } i \ne k \tag{3.55}$$

A set of conjugate directions possesses an important property: If minimizing line searches are used along each conjugate direction, a method of conjugate directions is guaranteed to minimize a quadratic function of *n* variables in at most *n* steps. Himmelblau indicates the excellent convergence properties of quasi-Newton methods on general functions may be due more to their conjugate direction properties than to their ability to approximate the Hessian inverse.[2] Because of the importance of conjugate directions, we will prove two results here.

PROPOSITION. If **H** is positive definite and the set of non-zero vectors $\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{n-1}$ are conjugate to **H,** then these vectors are linearly independent.

PROOF. Suppose we have constants, α^i, $i = 0, 2, ..., n-1$ such that

$$\alpha^0 \mathbf{s}^0 + \alpha^1 \mathbf{s}^1 + ... + \alpha^k \mathbf{s}^k + ... + \alpha^{n-1} \mathbf{s}^{n-1} = \mathbf{0} \tag{3.56}$$

Now we multiply each term by $\left(\mathbf{s}^k\right)^T \mathbf{H}$:

$$\alpha^0 \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^0}_{=0} + \alpha^1 \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^1}_{=0} + ... + \alpha^k \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^k}_{positive} + ... + \alpha^{n-1} \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^{n-1}}_{=0} = \mathbf{0} \tag{3.57}$$

From conjugacy, all of the terms except $\alpha^k \left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^k$ are zero. Since **H** is positive definite, then the only way for this remaining term to be zero is for α^k to be zero. In this way we can show that for (3.57) to be satisfied all the α coefficients must be zero. This is the definition of linear independence.

7.3.2. Conjugate Direction Theorem

We will now show that a method of conjugate directions will solve a quadratic function in *n* steps, if minimizing steps are taken.

THEOREM. Let $\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{n-1}$ be a set of non-zero **H** conjugate vectors, with **H** a positive definite matrix. For the function,

$$f^{k+1} = f^k + \left(\nabla f^k\right)^{\mathrm{T}} \left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) + \frac{1}{2}\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right)^{\mathrm{T}} \mathbf{H} \left(\mathbf{x}^{k+1} - \mathbf{x}^k\right) \tag{3.58}$$

the sequence,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k \tag{3.59}$$

with,

$$\alpha^k = -\frac{\left(\nabla f^k\right)^T \mathbf{s}^k}{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^k} \tag{3.60}$$

$$\nabla f^{k+1} = \nabla f^k + \mathbf{H}\left(\mathbf{x}^{k+1} - \mathbf{x}^k\right)$$

[2] Himmelblau, *Applied Nonlinear Programming*, p. 112.

converges to the unique solution, $\mathbf{H}(\mathbf{x}^* - \mathbf{x}^k) = -\nabla f^k$, after *n* steps, that is $\mathbf{x}^n = \mathbf{x}^*$.

PROOF. Based on (3.59) above we note that,

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha^0 \mathbf{s}^0$$

Likewise for $\mathbf{x}^2$:

$$\mathbf{x}^2 = \mathbf{x}^1 + \alpha^1 \mathbf{s}^1 = \mathbf{x}^0 + \alpha^0 \mathbf{s}^0 + \alpha^1 \mathbf{s}^1$$

Or, in general

$$\left(\mathbf{x}^k - \mathbf{x}^0\right) = \alpha^0 \mathbf{s}^0 + \alpha^1 \mathbf{s}^1 + \ldots + \alpha^{k-1} \mathbf{s}^{k-1} \tag{3.61}$$

After *n* steps, we can write the optimum (assuming the directions are independent, which we just showed) as,

$$\left(\mathbf{x}^* - \mathbf{x}^0\right) = \alpha^0 \mathbf{s}^0 + \alpha^1 \mathbf{s}^1 + \ldots + \alpha^k \mathbf{s}^k + \ldots + \alpha^{n-1} \mathbf{s}^{n-1} \tag{3.62}$$

Multiplying both sides of (3.62) by $\left(\mathbf{s}^k\right)^T \mathbf{H}$, we have,

$$\left(\mathbf{s}^k\right)^T \mathbf{H}\left(\mathbf{x}^* - \mathbf{x}^0\right) = \alpha^0 \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^0}_{=0} + \alpha^1 \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^1}_{=0} + \ldots + \alpha^k \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^k}_{positive} + \ldots + \alpha^{n-1} \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^{n-1}}_{=0}$$

Solving for α^k:

$$\alpha^k = \frac{\left(\mathbf{s}^k\right)^T \mathbf{H}(\mathbf{x}^* - \mathbf{x}^0)}{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^k} \tag{3.63}$$

Unfortunately (3.63) is in terms of **x***, which we presumably don't know. However, if we multiply (3.61) by $\left(\mathbf{s}^k\right)^T \mathbf{H}$, we have,

$$\left(\mathbf{s}^k\right)^T \mathbf{H}\left(\mathbf{x}^k - \mathbf{x}^0\right) = \alpha^0 \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^0}_{=0} + \alpha^1 \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^1}_{=0} + \ldots + \alpha^{k-1} \underbrace{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^{k-1}}_{=0} \tag{3.64}$$

which gives,

$$\left(\mathbf{s}^k\right)^T \mathbf{H}(\mathbf{x}^k - \mathbf{x}^0) = 0 \tag{3.65}$$

Substituting this result into (3.63), we have

$$\alpha^k = \frac{\left(\mathbf{s}^k\right)^T \mathbf{H}(\mathbf{x}^* - \mathbf{x}^k)}{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^k} \tag{3.66}$$

Noting that $\mathbf{H}(\mathbf{x}^* - \mathbf{x}^k) = -\nabla f^k$ is the solution to (3.58), we can solve for the α^k as,

$$\alpha^k = -\frac{\left(\nabla f^k\right)^T \mathbf{s}^k}{\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^k}$$

which is identical with (3.60).

We notice that (3.60) is the same as the minimizing step we derived in Section 9.2. Thus the conjugate direction theorem relies on taking minimizing steps.

7.3.3. Examples

We stated earlier that quasi-Newton methods are also methods of conjugate directions. Thus for the example given in Section 7.3, we should have,

$$\left(\mathbf{s}^0\right)^T \mathbf{H}\mathbf{s}^1 = 0$$

Substituting the search directions and Hessian of that problem,

$$\begin{bmatrix} 0.496 & -0.868 \end{bmatrix} \begin{bmatrix} 2. & -2. \\ -2. & 8. \end{bmatrix} \begin{bmatrix} 2.837 \\ 0.975 \end{bmatrix} = 0.0017$$

Within the round-off of the data, we see this is verified.

In the previous problem we only had two search directions. Let's look at a problem where we have three search directions so we have more conjugate relationships to examine. We will consider the problem, Min $f = 2x_1 + x_1^2 + 4x_2 + 4x_2^2 + 8x_3 + 2x_3^2$.

Starting from $\mathbf{x}^0 = \begin{bmatrix} 2 \\ 3 \\ 3 \end{bmatrix}$ $\nabla f^0 = \begin{bmatrix} 6 \\ 28 \\ 20 \end{bmatrix}$ $\mathbf{s}^0 = \begin{bmatrix} -0.172 \\ -0.802 \\ -0.573 \end{bmatrix}$

We execute a line search in the direction of steepest descent (normalized as $\mathbf{s}^0$ above), stop at α^* and determine the new point and gradient. We calculate the new search direction using our rank 1 update,

$$\mathbf{x}^1 = \begin{bmatrix} 1.079 \\ -1.300 \\ -0.072 \end{bmatrix} \quad \nabla f^1 = \begin{bmatrix} 4.157 \\ -6.401 \\ 7.714 \end{bmatrix} \quad \mathbf{s}^1 = \begin{bmatrix} -4.251 \\ 3.320 \\ -8.657 \end{bmatrix}$$

We go through this cycle again,

$$\mathbf{x}^2 = \begin{bmatrix} 0.019 \\ -0.473 \\ -2.229 \end{bmatrix} \quad \nabla f^2 = \begin{bmatrix} 2.038 \\ 0.218 \\ -0.917 \end{bmatrix} \quad \mathbf{s}^2 = \begin{bmatrix} -2.107 \\ -0.056 \\ 0.474 \end{bmatrix}$$

After stepping in the above direction we arrive at the optimum,

$$\mathbf{x}^3 = \begin{bmatrix} -1.000 \\ -0.500 \\ -2.000 \end{bmatrix} \quad \nabla f^3 = \begin{bmatrix} 0.000 \\ 0.000 \\ 0.000 \end{bmatrix}$$

Since we have used a method of conjugate directions, $\mathbf{s}^2$ should be conjugate to $\mathbf{s}^1$ and $\mathbf{s}^0$. We will check this:

$$\left(\mathbf{s}^0\right)^T \mathbf{H}\mathbf{s}^2 = \begin{bmatrix} -0.172 & -0.802 & -0.573 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} -2.107 \\ -0.056 \\ 0.474 \end{bmatrix} = -0.0023$$

$$\left(\mathbf{s}^1\right)^T \mathbf{H}\mathbf{s}^2 = \begin{bmatrix} -4.251 & 3.320 & -8.657 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} -2.107 \\ -0.056 \\ 0.474 \end{bmatrix} = 0.0127$$

7.3.4. Some Insight into Conjugacy

As we did in section 4.3, we will define the "error" in the objective at the current value of **x** as,

$$E(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x} - \mathbf{x}^*\right)^{\mathrm{T}} \mathbf{H}\left(\mathbf{x} - \mathbf{x}^*\right)$$

We can rewrite this expression as,

$$E(\boldsymbol{\alpha}) = \frac{1}{2}\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right)\mathbf{S}^T\mathbf{H}\mathbf{S}\left(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\right) \tag{3.67}$$

Where **S** is a matrix with columns, $\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^{n-1}$. If the **s** vectors are conjugate then (3.67) reduces to,

$$E(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=0}^{n=1}(\alpha^i - \alpha^*)^2 d^i$$

where $d^i = \left(\mathbf{s}^i\right)^T \mathbf{H}\mathbf{s}^{(i)}$. $E(\boldsymbol{\alpha})$ can then be minimized by choosing $\alpha^i = \alpha^*$, i.e., by making exact line searches. Quoting Fletcher,[3] "Thus conjugacy implies a diagonalizing transformation $\mathbf{S}^T\mathbf{H}\mathbf{S}$ of $\mathbf{H}$ to a new coordinate system, $\boldsymbol{\alpha}$, in which the variables are decoupled. A conjugate direction method is then the alternating variables method applied in this new coordinate system." The "alternating variables" method referred to is just a method where the optimum is found with respect to one variable, holding the rest constant, and then a second variable, etc. Usually such a scheme would not work well. Conjugate directions are such that the α^i's are decoupled so it does work here.

As we show in Section 9.3, another result of conjugacy is that at the *k+1* step,

$$\left(\nabla f^{k+1}\right)^T \mathbf{s}^i = 0 \quad \text{for all} \quad i \le k \tag{3.68}$$

Equation (3.68) indicates 1) that the current gradient is orthogonal to all the past search directions, and 2) at the current point we have zero slope with respect to all past search directions, i.e.,

$$\frac{\partial f}{\partial \alpha^i} = 0 \quad \text{for all} \quad i \le k$$

meaning we have minimized the function in the "subspace" of the previous directions. As an example, for the three variable function of Section 7.5, ∇f^2 should be orthogonal to $\mathbf{s}^0$ and $\mathbf{s}^1$:

$$\left(\nabla f^2\right)^T \mathbf{s}^0 = \begin{bmatrix} 2.038 & 0.218 & -0.917 \end{bmatrix} \begin{bmatrix} -0.172 \\ -0.802 \\ -0.573 \end{bmatrix} = 0.0007$$

$$\left(\nabla f^2\right)^T \mathbf{s}^1 = \begin{bmatrix} 2.038 & 0.218 & -0.917 \end{bmatrix} \begin{bmatrix} -4.251 \\ 3.320 \\ -8.657 \end{bmatrix} = -0.0013$$

7.4. Rank 2 Updates

7.4.1. The DFP Method

Although the rank one update does have the hereditary property (and is a method of conjugate directions), it does not guarantee that at each stage the direction matrix, **N**, is positive definite. It is important that the update remain positive definite because this insures the search

[3] R. Fletcher, *Practical Methods of Optimization, Second Edition,* 1987, pg. 26.

direction will always go downhill. It has been shown that (3.42) is the only rank one update which satisfies the quasi-Newton condition. For more flexibility, rank 2 updates have been proposed. These are of the form,

$$\mathbf{N}^{k+1} = \mathbf{N}^k + a\mathbf{u}\mathbf{u}^{\mathrm{T}} + b\mathbf{v}\mathbf{v}^{\mathrm{T}} \tag{3.69}$$

If we substitute this into the quasi-Newton condition,

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^k = \Delta\mathbf{x}^k \tag{3.70}$$

we have,

$$\mathbf{N}^k\boldsymbol{\gamma}^k + a\mathbf{u}\mathbf{u}^{\mathrm{T}}\boldsymbol{\gamma}^k + b\mathbf{v}\mathbf{v}^{\mathrm{T}}\boldsymbol{\gamma}^k = \Delta\mathbf{x}^k \tag{3.71}$$

There are a number of possible choices for **u** and **v**. One choice is to try,

$$\mathbf{u} = \Delta\mathbf{x}^k, \quad \mathbf{v} = \mathbf{N}^k\boldsymbol{\gamma}^k \tag{3.72}$$

Substituting (3.72) into (3.71),

$$\mathbf{N}^k\boldsymbol{\gamma}^k + a\Delta\mathbf{x}^k\underbrace{\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k}_{scalar} + b\mathbf{N}^k\boldsymbol{\gamma}^k\underbrace{\left(\mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k}_{scalar} = \Delta\mathbf{x}^k \tag{3.73}$$

In (3.73) we note that the dot products result in scalars. If we choose *a* and *b* such that,

$$a\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k = 1 \text{ and } b\left(\mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k = -1 \tag{3.74}$$

Equation (3.71) becomes,

$$\mathbf{N}^k\boldsymbol{\gamma}^k + \Delta\mathbf{x}^k - \mathbf{N}^k\boldsymbol{\gamma}^k = \Delta\mathbf{x}^k \tag{3.75}$$

and is satisfied.

Combining (3.74), (3.72) and (3.69), the update is,

$$\mathbf{N}^{k+1} = \mathbf{N}^k + \frac{\Delta\mathbf{x}^k\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}}{\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k} - \frac{\mathbf{N}^k\boldsymbol{\gamma}^k\left(\mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}}{\left(\mathbf{N}^k\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k} \tag{3.76}$$

Or, with some rearranging, as it is more commonly given,

$$\mathbf{N}^{k+1} = \mathbf{N}^k + \frac{\Delta\mathbf{x}^k\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}}{\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}\boldsymbol{\gamma}^k} - \frac{\mathbf{N}^k\boldsymbol{\gamma}^k\left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\mathbf{N}^k}{\left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}\mathbf{N}^k\boldsymbol{\gamma}^k} \tag{3.77}$$

Davidon[4] was the first one to propose this update. Fletcher and Powell further developed his method;[5] thus this method came to be known as the Davidon-Fletcher-Powell (DFP) update. This update has the following properties,

For quadratic functions:

1. it has the hereditary property; after *n* updates, $\mathbf{N}^n = \mathbf{H}^{-1}$.
2. it is a method of conjugate directions and therefore terminates after at most n steps.

For general functions (including quadratics):

3. the direction matrix **N** remains positive definite if we do exact line searches. This guarantees the search direction points downhill at every step. This property is proved in the next section.

7.4.2. Proof the DFP Update Stays Positive Definite

THEOREM. If $\left(\Delta\mathbf{x}^k\right)^T \boldsymbol{\gamma} > 0$ for all steps of the algorithm, and if we start with any symmetric, positive definite matrix, $\mathbf{N}^0$, then the DFP update preserves the positive definiteness of $\mathbf{N}^k$ for all k.

PROOF. The proof is inductive. We will show that if $\mathbf{N}^k$ is positive definite, $\mathbf{N}^{k+1}$ is also. From the definition of positive definiteness,

$$\mathbf{z}^{\mathrm{T}}\mathbf{N}^{k+1}\mathbf{z} > 0 \quad \text{for all } \mathbf{z} \neq 0$$

For simplicity we will drop the superscript *k* on the update terms. From (3.66),

$$\mathbf{z}^{\mathrm{T}}\mathbf{N}^{k+1}\mathbf{z} = \underbrace{\mathbf{z}^{\mathrm{T}}\mathbf{N}^{k}\mathbf{z}}_{term\,1} + \underbrace{\mathbf{z}^{\mathrm{T}}\left(\frac{\Delta\mathbf{x}\Delta\mathbf{x}^{\mathrm{T}}}{\Delta\mathbf{x}^{\mathrm{T}}\boldsymbol{\gamma}}\right)\mathbf{z}}_{term\,2} - \underbrace{\mathbf{z}^{\mathrm{T}}\left(\frac{\mathbf{N}\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{N}}{\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{N}\boldsymbol{\gamma}}\right)\mathbf{z}}_{term\,3} \tag{3.78}$$

We need to show that all the terms on the right hand side are positive. We will focus for a moment on the first and third terms on the right hand side. Noting that **N** can be written as $\mathbf{N} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ via Choleski decomposition, and if we substitute $\mathbf{a} = \mathbf{L}^{\mathrm{T}}\mathbf{z}$, $\mathbf{a}^{\mathrm{T}} = \mathbf{z}^{\mathrm{T}}\mathbf{L}$, $\mathbf{b} = \mathbf{L}^{\mathrm{T}}\boldsymbol{\gamma}$, $\mathbf{b}^{\mathrm{T}} = \boldsymbol{\gamma}^{\mathrm{T}}\mathbf{L}$ the first and third terms are,

$$\mathbf{z}^{\mathrm{T}}\mathbf{N}\mathbf{z} - \mathbf{z}^{\mathrm{T}}\left(\frac{\mathbf{N}\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{N}}{\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{N}\boldsymbol{\gamma}}\right)\mathbf{z} = \mathbf{a}^{\mathrm{T}}\mathbf{a} - \frac{\left(\mathbf{a}^{\mathrm{T}}\mathbf{b}\right)^2}{\mathbf{b}^{\mathrm{T}}\mathbf{b}} \tag{3.79}$$

The Cauchy-Schwarz inequality states that for any two vectors, **x** and **y**,

[4] W. C. Davidon, *USAEC Doc*. ANL-5990 (rev.) Nov. 1959
[5] R. Fletcher and M. J. D. Powell, *Computer J.* 6: 163, 1963

$$\mathbf{x}^T\mathbf{x} \geq \frac{\left(\mathbf{x}^T\mathbf{y}\right)^2}{\mathbf{y}^T\mathbf{y}} \qquad \text{thus} \qquad \mathbf{a}^T\mathbf{a} - \frac{\left(\mathbf{a}^T\mathbf{b}\right)^2}{\mathbf{b}^T\mathbf{b}} \geq 0 \tag{3.80}$$

So the first and third terms of (3.78) are positive. Now we need to show this for the second term,

$$\mathbf{z}^T\left(\frac{\Delta\mathbf{x}\Delta\mathbf{x}^T}{\Delta\mathbf{x}^T\boldsymbol{\gamma}}\right)\mathbf{z} = \frac{\mathbf{z}^T\Delta\mathbf{x}\Delta\mathbf{x}^T\mathbf{z}}{\Delta\mathbf{x}^T\boldsymbol{\gamma}} = \frac{\left(\mathbf{z}^T\Delta\mathbf{x}\right)^2}{\Delta\mathbf{x}^T\boldsymbol{\gamma}} \tag{3.81}$$

The numerator of the right-most expression is obviously positive. The denominator can be written,

$$\Delta\mathbf{x}^T\boldsymbol{\gamma} = \left(\Delta\mathbf{x}^k\right)^T\nabla f^{k+1} - \left(\Delta\mathbf{x}^k\right)^T\nabla f^k = \underbrace{\alpha\left(\mathbf{s}^k\right)^T\nabla f^{k+1}}_{term\,1} - \underbrace{\alpha\left(\mathbf{s}^k\right)^T\nabla f^k}_{term\,2} \tag{3.82}$$

The second term in (3.82), $\left(\mathbf{s}^k\right)^T\nabla f^k$, is negative if the search direction goes downhill which it does if $\mathbf{N}^k$ is positive definite, and with the minus sign is therefore positive. The first term in (3.82), $\alpha\left(\mathbf{s}^k\right)^T\nabla f^{k+1}$, can be positive or negative; however, it is zero if we are at α^*; thus the entire expression in (3.82) is positive if we take a minimizing step, α^*.

We have now shown that all three terms of (3.78) are positive if we take a minimizing step. Thus, if $\mathbf{N}^k$ is positive definite, $\mathbf{N}^{k+1}$ is positive definite, etc.

7.4.3. DFP Update: Closing Remarks

The DFP update was popular for many years. As mentioned, we need to take a minimizing step to insure **N** stays positive definite. Recall that we find α^* using a parabolic fit; on non-quadratics there is usually some error here. We can reduce the error by refitting the parabola several times as we obtain more points in the region of α^*. However, this requires more function evaluations. The DFP method is more sensitive to errors in α^* than the BFGS update, described in the next section, and can degrade if α^* is not accurate.

7.5. The Broyden Fletcher Goldfarb Shanno (BFGS) Update

The current "best" update is known as the Broyden, Fletcher, Goldfarb, Shanno or "BFGS" update, suggested by all four authors independently in 1970. It is also a rank 2 update. It has the same properties as the DFP update but is less sensitive to errors in α^*. This means we can be "sloppy" in our line search when we are far away from the optimum and the method still works well. This update is,

$$\mathbf{N}^{k+1} = \mathbf{N}^k + \left(1 + \frac{\left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}} \mathbf{N}^k \boldsymbol{\gamma}^k}{\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \boldsymbol{\gamma}^k}\right)\left(\frac{\Delta\mathbf{x}^k \left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}}{\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \boldsymbol{\gamma}^k}\right) - \frac{\Delta\mathbf{x}^k \left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}} \mathbf{N}^k + \mathbf{N}^k \boldsymbol{\gamma}^k \left(\Delta\mathbf{x}^k\right)^{\mathrm{T}}}{\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \boldsymbol{\gamma}^k} \qquad (3.83)$$

This update is currently considered to be the best update for use in optimization. It is the update inside OptdesX, Excel and many other optimization packages.

7.6. Comments About Quasi-Newton Methods

The quasi-Newton methods explained here combine the advantages of steepest descent and Newton's method without the disadvantages. They start out as steepest descent, which works well far from the optimum, and gradually become Newton's method, which works well near the optimum. They do this without requiring the evaluation of second derivatives. By insuring the update is positive definite, the search direction will always go downhill.

Note that these methods use information the previous methods "threw away." Quasi-Newton methods use differences in gradients and differences in **x** to estimate second derivatives according to (3.34). This allows information from previous steps to correct (or update) the current step.

As mentioned, quasi-Newton methods are also methods of conjugate directions. This is shown in Section 9.4.

7.7. Hessian Updates Vs. Hessian Inverse Updates

All of the updates we have presented so far are updates for the Hessian Inverse. We can easily develop updates for the Hessian itself, as will be required for the SQP algorithm, starting from the condition

$$\boldsymbol{\gamma}^k = \mathbf{H}^k \Delta\mathbf{x}^k \qquad (3.84)$$

instead of $\left(\mathbf{H}^{-1}\right)^k \boldsymbol{\gamma}^k = \Delta\mathbf{x}^k$ which we used before. The BFGS Hessian approximation (Equation (3.83) is the Hessian inverse approximation) is given by,

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{\boldsymbol{\gamma}^k \left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}}}{\left(\boldsymbol{\gamma}^k\right)^{\mathrm{T}} \Delta\mathbf{x}^k} - \frac{\mathbf{H}^k \Delta\mathbf{x}^k \left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \mathbf{H}^k}{\left(\Delta\mathbf{x}^k\right)^{\mathrm{T}} \mathbf{H}^k \Delta\mathbf{x}^k} \qquad (3.85)$$

You will note that this looks a lot like the DFP Hessian inverse update but with **H** interchanged with **N** and **γ** interchanged with Δ**x**. In fact these two formulas are said to be *complementary* to each other.

8. The Conjugate Gradient Method

8.1. Definition

There is one more method we will learn, called the *conjugate gradient* method. We will present the results for this method primarily because it is an algorithm used in Microsoft Excel.

The conjugate gradient method is built upon steepest descent, except a correction factor is added to the search direction. The correction makes this method a method of conjugate directions. For the conjugate direction method, the search direction is given by,

$$\mathbf{s}^{k+1} = -\nabla f^{k+1} + \beta^{\mathrm{k}}\mathbf{s}^{\mathrm{k}} \tag{3.86}$$

Where β^{k}, a scalar, is given by

$$\beta^{k} = \frac{\left(\nabla f^{k+1}\right)^{\mathrm{T}} \nabla f^{k+1}}{\left(\nabla f^{k}\right)^{\mathrm{T}} \nabla f^{k}} \tag{3.87}$$

8.2. Example: Conjuage Gradient Method

We will optimize our usual function, $f = x_1^2 - 2x_1x_2 + 4x_2^2$

starting from $\mathbf{x}^0 = \begin{bmatrix} -3 \\ 1 \end{bmatrix} \quad \nabla f^0 = \begin{bmatrix} -8 \\ 14 \end{bmatrix}$

We take a minimizing step in the negative gradient direction and stop at

$$\mathbf{x}^1 = \begin{bmatrix} -2.03 \\ -0.7 \end{bmatrix} \quad \nabla f^1 = \begin{bmatrix} -2.664 \\ -1.522 \end{bmatrix}$$

Now we calculate β^0 as

$$\beta^0 = \frac{\left(\nabla f^1\right)^{\mathrm{T}} \nabla f^1}{\left(\nabla f^0\right)^{\mathrm{T}} \nabla f^0} = \frac{\begin{bmatrix} -2.664 & -1.522 \end{bmatrix}\begin{bmatrix} -2.664 \\ -1.522 \end{bmatrix}}{\begin{bmatrix} -8 & 14 \end{bmatrix}\begin{bmatrix} -8 \\ 14 \end{bmatrix}} = \frac{9.413}{260} = 0.0362$$

We calculate the new search direction as,

$$\mathbf{s}^1 = -\nabla f^1 + \beta \mathbf{s}^0 = -\begin{bmatrix} -2.664 \\ -1.522 \end{bmatrix} + 0.0362 \begin{bmatrix} 8 \\ -14 \end{bmatrix} = \begin{bmatrix} 2.954 \\ 1.015 \end{bmatrix}$$

when we step in this direction, we arrive at the optimum, $\mathbf{x}^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \nabla f^2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

The main advantage of the conjugate gradient method, as compared to quasi-Newton methods, is computation and storage. The conjugate gradient method only requires that we store the last search direction and last gradient, instead of a full matrix. Thus this method is a good one to use for large problems (say with 500 variables).

Although both conjugate gradient and quasi-Newton methods will optimize quadratic functions in *n* steps, on real problems quasi-Newton methods are better. Further, small errors can build up in the conjugate gradient method so some researchers recommend *restarting* the algorithm periodically (such as every *n* steps) to be steepest descent.

9. Appendix

9.1. The Gradient of a Quadratic Function in Vector Form

We define the coordinate vector to be,

$$\mathbf{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{A single 1 in the } i^{th} \text{ position} \tag{3.88}$$

We note that $\nabla x_i = \mathbf{e}_i$ so

$$\begin{aligned} \nabla \mathbf{x}^{\mathrm{T}} &= \left[\nabla x_1,\ \nabla x_2\ , \ldots,\ \nabla x_n \right] \\ &= \left[\mathbf{e}_1,\ \mathbf{e}_2\ , \ldots,\ \mathbf{e}_n \right] = \mathbf{I} \end{aligned} \tag{3.89}$$

Suppose we have a linear function:

$$f\left(\mathbf{x}\right) = a + \mathbf{b}^{\mathrm{T}}\mathbf{x}$$

then $\nabla f\left(\mathbf{x}\right) = \nabla\left(a + \ \mathbf{b}^{\mathrm{T}}\mathbf{x}\right) = \underbrace{\nabla a}_{term\,1} + \underbrace{\nabla\left(\mathbf{b}^{\mathrm{T}}\mathbf{x}\right)}_{term\,2}$

For the first term, since *a* is a constant, $\nabla a = 0$. Looking at the second term, from the rule for differentiation of a product,

$$\nabla\left(\mathbf{b}^{\mathrm{T}}\mathbf{x}\right) = \left(\nabla \mathbf{b}^{\mathrm{T}}\right)\mathbf{x} + \left(\nabla \mathbf{x}^{\mathrm{T}}\right)\mathbf{b}$$

but $\nabla \mathbf{b}^{\mathrm{T}} = \mathbf{0}^{\mathrm{T}}$ and $\nabla \mathbf{x}^{\mathrm{T}} = \mathbf{I}$

Thus
$$\begin{aligned}\nabla f(\mathbf{x}) &= \nabla a + \nabla\left(\mathbf{b}^{\mathrm{T}}\mathbf{x}\right) \\ &= 0 + \left(\nabla \mathbf{b}^{\mathrm{T}}\right)\mathbf{x} + \left(\nabla \mathbf{x}^{\mathrm{T}}\right)\mathbf{b} \\ &= 0 + 0 + \mathbf{I}\mathbf{b} \\ &= \mathbf{b}\end{aligned} \tag{3.90}$$

Now suppose we have a quadratic function of the form:

$$q(\mathbf{x}) = a + \mathbf{b}^{\mathrm{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x} \tag{3.91}$$

We wish to evaluate the gradient in vector form. We will do this term by term,

$$\nabla q(\mathbf{x}) = \nabla a + \nabla\left(\mathbf{b}^{\mathrm{T}}\mathbf{x}\right) + \frac{1}{2}\nabla\left(\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}\right)$$

Applying the results from a linear function,

$$\begin{aligned}\nabla q(\mathbf{x}) &= \nabla a + \nabla\left(\mathbf{b}^{\mathrm{T}}\mathbf{x}\right) + \frac{1}{2}\nabla\left(\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}\right) \\ &= 0 + \mathbf{b} + \frac{1}{2}\nabla\left(\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}\right)\end{aligned}$$

So we only need to evaluate the term, $\frac{1}{2}\nabla\left(\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}\right)$. If we split this into two vectors, i.e. $\mathbf{u} = \mathbf{x},\ \mathbf{v} = \mathbf{H}\mathbf{x}$, then

$$\nabla\left(\mathbf{x}^{\mathrm{T}}\mathbf{H}\mathbf{x}\right) = \left(\nabla \mathbf{x}^{\mathrm{T}}\right)\mathbf{v} + \left(\nabla \mathbf{v}^{\mathrm{T}}\right)\mathbf{x}$$

We know $\left(\nabla \mathbf{x}^{\mathrm{T}}\right)\mathbf{v} = \mathbf{I}\mathbf{H}\mathbf{x} = \mathbf{H}\mathbf{x}$, so we must only evaluate $\left(\nabla \mathbf{v}^{\mathrm{T}}\right)\mathbf{x} = \left(\nabla\left(\mathbf{H}\mathbf{x}\right)^{\mathrm{T}}\right)\mathbf{x}$. We can write,

$$\left(\mathbf{H}\mathbf{x}\right)^{\mathrm{T}} = [\mathbf{h}_{r1}^{\mathrm{T}}\mathbf{x}\ ,\ \mathbf{h}_{r2}^{\mathrm{T}}\mathbf{x}\ ,\ldots,\ \mathbf{h}_{rn}^{\mathrm{T}}\mathbf{x}]$$

where $\mathbf{h}_{r1}^{\mathrm{T}}$ represents the first row of $\mathbf{H}$, $\mathbf{h}_{r2}^{\mathrm{T}}$ represents the second row, and so forth. Applying the gradient operator,

$$\nabla\left(\mathbf{H}\mathbf{x}\right)^{\mathrm{T}} = \left[\nabla\left(\mathbf{h}_{r1}^{\mathrm{T}}\mathbf{x}\right)\ ,\ \nabla\left(\mathbf{h}_{r2}^{\mathrm{T}}\mathbf{x}\right)\ ,\ldots,\ \nabla\left(\mathbf{h}_{rn}^{\mathrm{T}}\mathbf{x}\right)\right]$$

From the previous result for $\nabla \mathbf{b}^{\mathrm{T}}\mathbf{x}$, we know that $\nabla \mathbf{h}_{ri}^{\mathrm{T}}\mathbf{x} = \mathbf{h}_{ri}$ since $\mathbf{h}_{ri}$ is a vector constant. Therefore,

$$\nabla\left(\mathbf{Hx}\right)^{\mathrm{T}} = \left[\mathbf{h}_{r1},\ \mathbf{h}_{r2}\ ,\ldots,\ \mathbf{h}_{rn}\right]$$
$$= \mathbf{H}^{\mathrm{T}}$$

Returning now to the gradient of the expression, $q\left(\mathbf{x}\right) = a + \mathbf{b}^{\mathrm{T}}\mathbf{x} + \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Hx}$

$$\nabla q\left(\mathbf{x}\right) = \nabla a + \nabla\left(\mathbf{b}^{\mathrm{T}}\mathbf{x}\right) + \nabla\left(\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{Hx}\right)$$
$$= 0 + \mathbf{b} + \frac{1}{2}\left\{\nabla\left(\mathbf{x}^{\mathrm{T}}\right)\mathbf{H} + \nabla\left(\mathbf{Hx}\right)^{\mathrm{T}}\right\}\mathbf{x}$$
$$= \mathbf{b} + \frac{1}{2}\left(\mathbf{H} + \mathbf{H}^{\mathrm{T}}\right)\mathbf{x}$$
$$= \mathbf{b} + \mathbf{Hx} \tag{3.92}$$

If the quadratic we are approximating is a Taylor expansion,

$$f^{k+1} = f^{k} + \left(\nabla f^{k}\right)^{\mathrm{T}}\Delta\mathbf{x}^{k} + \frac{1}{2}\left(\Delta\mathbf{x}^{k}\right)^{\mathrm{T}}\mathbf{H}^{k}\Delta\mathbf{x}^{k}$$

Then (3.92) is:

$$\nabla f^{k+1} = \nabla f^{k} + \mathbf{H}^{k}\Delta\mathbf{x}^{k} \tag{3.93}$$

9.2. Optimal Step Length for Quadratic Function

In this section we will derive (3.12). If we start with a Taylor expansion,

$$f^{k+1} = f^{k} + \left(\nabla f^{k}\right)^{\mathrm{T}}\Delta\mathbf{x}^{k} + \frac{1}{2}\left(\Delta\mathbf{x}^{k}\right)^{\mathrm{T}}\mathbf{H}\Delta\mathbf{x}^{k} \tag{3.94}$$

When we do a line search,

$$\Delta\mathbf{x}^{k} = \alpha\mathbf{s} \tag{3.95}$$

Substituting (3.95) into (3.94) gives

$$f^{k+1} = f^{k} + \left(\nabla f^{k}\right)^{\mathrm{T}}\alpha\mathbf{s} + \frac{1}{2}\left(\alpha\mathbf{s}\right)^{\mathrm{T}}\mathbf{H}\alpha\mathbf{s}$$

If we take the derivative of this expression with respect to α (a scalar),

$$\frac{df^{k+1}}{d\alpha} = \left(\nabla f^{k}\right)^{\mathrm{T}}\mathbf{s} + \alpha\mathbf{s}^{\mathrm{T}}\mathbf{Hs} \tag{3.96}$$

Setting the derivative equal to zero and solving for α gives:

$$\alpha^* = -\frac{\left(\nabla f^k\right)^{\mathrm{T}} \mathbf{s}}{\mathbf{s}^T \mathbf{H} \mathbf{s}} \tag{3.97}$$

9.3. Proof that a Method of Conjugate Directions Minimizes the Current Subspace

THEOREM. A conjugate direction method is such that each $\mathbf{x}^{k+1}$ is the minimizer in the subspace generated by $\mathbf{x}^0$ and the directions, $\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^k$, i.e. $\mathbf{x}^{k+1} = \mathbf{x}^0 + \sum \alpha^i \mathbf{s}^i \quad i = 0,1,...,k$.

We wish to show that,

$$\left(\nabla f^{k+1}\right)^T \mathbf{s}^i = 0 \quad \text{for all } i \le k \tag{3.98}$$

which indicates that we have zero slope along any search direction in the subspace generated by $\mathbf{x}^0$ and the search directions $\mathbf{s}^0, \mathbf{s}^1, ..., \mathbf{s}^k$, i.e.,

$$\frac{\partial f}{\partial \alpha^i} = 0 \quad \text{for all } i \le k$$

PROOF. The proof by induction. Given the usual expression for the gradient of a Taylor expansion,

$$\nabla f^{k+1} = \nabla f^k + \mathbf{H} \Delta \mathbf{x}^k$$

Which we will write as,

$$\nabla f^{k+1} = \nabla f^k + \alpha \mathbf{H} \mathbf{s}^k \tag{3.99}$$

If we multiply both sides by $\mathbf{s}^k$

$$\left(\mathbf{s}^k\right)^T \nabla f^{k+1} = \left(\mathbf{s}^k\right)^T \nabla f^k + \alpha \left(\mathbf{s}^k\right)^T \mathbf{H} \mathbf{s}^k = 0$$

By definition of α^k this is true for $i=k$. For $i<k$,

$$\left(\mathbf{s}^i\right)^T \nabla f^{k+1} = \underbrace{\left(\mathbf{s}^i\right)^T \nabla f^k}_{term\,1} + \underbrace{\alpha \left(\mathbf{s}^i\right)^T \mathbf{H} \mathbf{s}^k}_{term\,2}$$

Term 1 vanishes by the induction hypothesis, while term 2 vanishes from the definition of conjugate directions.

9.4. Proof that an Update with the Hereditary Property is Also a Method of Conjugate Directions

THEOREM. An update with the hereditary property and exact line searches is a method of conjugate directions and therefore terminates after $m \le n$ iterations on a quadratic function.

We assume that the hereditary property holds for $k = 1, 2, ..., m$

$$\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \Delta\mathbf{x}^i \quad \text{for all } i \le k \tag{3.100}$$

We need to show that conjugacy holds as well,

$$\left(\mathbf{s}^k\right)^T \mathbf{H}\mathbf{s}^i = 0 \quad \text{for all } i \le k-1 \tag{3.101}$$

The proof is by induction. We will show that if $\mathbf{s}^k$ is conjugate then $\mathbf{s}^{k+1}$ is as well, i.e.,

$$\left(\mathbf{s}^{k+1}\right)^T \mathbf{H}\mathbf{s}^i = 0 \quad \text{for all } i \le k \tag{3.102}$$

We note that

$$\mathbf{s}^{k+1} = -\mathbf{N}^{k+1}\nabla f^{k+1} \tag{3.103}$$

by definition of the quasi-Newton method. Or taking the transpose,

$$\left(\mathbf{s}^{k+1}\right)^T = -\left(\nabla f^{k+1}\right)^T \mathbf{N}^{k+1} \tag{3.104}$$

Substituting (3.104) into (3.102);

$$\left(\mathbf{s}^{k+1}\right)^T \mathbf{H}\mathbf{s}^i = -\left(\nabla f^{k+1}\right)^T \mathbf{N}^{k+1}\mathbf{H}\mathbf{s}^i \quad \text{for all } i \le k \tag{3.105}$$

Also,

$$\mathbf{H}\mathbf{s}^i = \frac{\mathbf{H}\Delta\mathbf{x}^i}{\alpha^i} = \frac{\boldsymbol{\gamma}^i}{\alpha^i}$$

so (3.105) becomes,

$$\left(\mathbf{s}^{k+1}\right)^T \mathbf{H}\mathbf{s}^i = -\frac{\left(\nabla f^{k+1}\right)^T \mathbf{N}^{k+1}\boldsymbol{\gamma}^i}{\alpha^i} \quad \text{for all } i \le k \tag{3.106}$$

From the hereditary property we have $\mathbf{N}^{k+1}\boldsymbol{\gamma}^i = \Delta\mathbf{x}^i \quad i \le k$, so (3.106) can be written,

$$\left(\mathbf{s}^{k+1}\right)^T \mathbf{H}\mathbf{s}^i = -\left[\frac{\left(\nabla f^{k+1}\right)^T \Delta \mathbf{x}^i}{\alpha^i}\right] = 0 \quad \text{for all } i \le k$$

The term in brackets is zero for all values of $i = 1, 2, ..., k-1$ from the assumption the previous search direction was conjugate which implies (3.98). It is zero for $i = k$ from the definition of α^*. Thus if we have conjugate directions at *k*, and the hereditary property holds, we have conjugate directions at *k+1*.

10. References

For more information on unconstrained optimization and in particular Hessian updates, see:

R. Fletcher, *Practical Methods of Optimization, Second Edition*, Wiley, 1987.

D. Luenberger, and Y. Ye, *Linear and Nonlinear Programming, Third Edition*, 2008.