Contents

Chapter 2

Newton Optimization Method

2.1 Introduction

The previous chapter presented some of the basic ideas for inverting seismic traveltimes. Now we broaden our understanding of seismic inversion by discussing the theory of unconstrained optimization. In optimization theory, we seek the optimal model vector $\mathbf{x}^*$ that minimizes the cost functional (or sometimes referred to as a misfit function) given by $f(\mathbf{x})$. The misfit functional is a scalar measure of how well the predicted data (based on some assumed model) fit the observed data, and in most geophysical problems this functional is non-linear with respect to the model components.

There are two main classes of optimization methods: gradient methods and non-gradient methods. In the gradient methods, the local gradient and/or curvature of the functional are used to steer us to a new model with a smaller value of $f(\mathbf{x})$. The chief merit is that they converge quickly for well-posed quadratic misfit functions, but tend to get stuck in local minima for highly non-linear misfit functions. In contrast, the non-gradient methods (e.g., Press et al., 1992; Stoffa and Sen, 1991; Sen and Stoffa, 1991; Ma, 2001) such as Monte Carlo search and simulated annealing (Aarts and Korst, 1990), conduct global searches over model space. Global searches by random or semi-random selections

tend to avoid many local minima that plague gradient methods, but the penalty is a very slow rate of convergence.

So what do most geophysicists use, fast and locally convergent gradient methods or the slow and possibly globally convergent methods. Most choose the gradient methods, which is what this book concentrates on. In fact, it mainly discusses the *unconstrained* optimization methods where no constraints are used in minimizing the misfit function. See Gill et al. (1981) for further details about constrained optimization methods.

This chapter discusses the Newton method, which assumes that the misfit function can be expressed as a summation of second-order model parameters. It is a very effective gradient method, but is often impractical because it requires a prohibitively expensive matrix inverse. Nevertheless, it is a good strating point to illustrate many important ideas in opimization theory. The next chapter will discuss the more practical iterative gradient methods such as steepest descent, Gauss-Newton, conjugate gradient and quasi-Newton methods. These methods are commonly used in tomography to invert for the slowness field from seismic data.

2.1.1 Notation

In the following sections, it will be assumed (unless otherwise stated) that the functional $f(\mathbf{x})$ is infinitely differentiable with respect to model parameters x_i, can be approximated by a quadratic functional, and possesses a global minima. A $Nx1$ vector $\mathbf{x}$ will be denoted with bold lower case letters, an operator or matrix such as $\mathbf{L}$ by bold upper case letters, and a scalar by lower case letters such as the function $f(\mathbf{x})$. The ponderous use of functional analysis (Kreyzig, 1978) notation will be minimized, but we will occasionally have relapses such as defining a functional as a mapping from a function space to a real line. Earth models will mostly be discretized into vector space elements in R^N, so we will mostly use notation such as differentiation w/r to the model space parameters rather than the more precise variation w/r to the model space function preferred by the theory of variational calculus.

2.2 Newton-Type Methods

Newton methods explicitly invert the Hessian matrix associated with second derivatives of the misfit function. The Hessian will be introduced starting from the 1-D functional minimization problem and working our way to the multi-dimensional optimization problem.

2.2.1 1-D Newton's Method

Let $f(x)$ be a differentiable 1-D functional dependent on the scalar variable x. The goal is to use Newton's method to find the optimal $x^* = x_o + \Delta x^*$ that minimizes $f(x)$. Expanding $f(x)$ about x_o we get

$$\begin{aligned} f(x_o + \Delta x) &= f(x_o) + \frac{\partial f(x_o)}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 f(x_o)}{\partial x^2}\Delta x^2 + O(\Delta x^3), \\ &\approx f(x_o) + \frac{\partial f(x_o)}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 f(x_o)}{\partial x^2}\Delta x^2, \end{aligned} \tag{2.1}$$

where the last equation assumes a nearly quadratic functional. Recognizing that the slope of $f(x)$ is zero at $x^* = x_o + \Delta x^*$ we differentiate equation 2.1 and evaluate at x^* to give

$$\begin{aligned} \frac{\partial f(x_o + \Delta x^*)}{\partial x} &= \frac{\partial f(x)}{\partial x} + \frac{\partial^2 f(x_o)}{\partial x^2}\Delta x^*, \\ &= 0 \end{aligned} \tag{2.2}$$

or rearranging and solving for Δx^* gives

$$\Delta x^* = -\frac{\partial f(x_o)}{\partial x} / \frac{\partial^2 f(x_o)}{\partial x^2}. \tag{2.3}$$

Equation 2.3 is the 1-D Newton formula and says that the Δx^* is given by the negative slope $-\partial f/\partial x$ to curvature $\partial^2 f/\partial x^2$ ratio. If the higher order terms in equation 2.1 can not be neglected (i.e., the functional is non-quadratic) then we must iteratively use the following update formula

$$x^{(k+1)} = x^{(k)} - \frac{\partial f(x^{(k)})}{\partial x} / \frac{\partial^2 f(x^{(k)})}{\partial x^2}, \tag{2.4}$$

where k denotes the k^{th} iteration, and it is understood that the derivative terms are evaluated at $x^{(k)}$.

In general, the convergence rate of Newton's method depends on the topography of the functional, which is described by the curvature and slope terms in equation 2.4 and illustrated in Figure 2.1.

- The sufficient conditions for a local minimum (Gill et al., 1981) are that $\frac{\partial f(x^*)}{\partial x} = 0$ and that $\frac{\partial^2 f(x^*)}{\partial x^2} > 0$ so that the slope is always increasing just away from the minimum; such a minimum is designated as a strong local minimum (Gill et al., 1981). If the curvature is negative at a stationary point then x^* represents a maximum.

- If $\frac{\partial^2 f(x^*)}{\partial x^2} = 0$, then the there are many neighboring values of x that locally minimize $f(x^*)$; such a minimum is designated as a weak local minimum (Gill et al., 1981). For multidimensional problems, the equivalent condition is that the curvature matrix (i.e., Hessian) is ill-conditioned so that many models explain the same noisy data.

- Global minimum where $f(x^*) < f(x)$ for all x.

Typical seismic optimization problems are characterized by the pathological cases shown in Figure 2.1: flat topography that leads to non-unique solutions and local minima that trap a gradient method into the wrong stationary point.

A MATLAB script which implements Newton's method for the 1-D function $f(x) = x^4 + x^2 - 3x$ is given below (courtesy of Maike Buddensiek).

```
% 1-D Newton Method to find zeros of a function

% plot functions
x = [-10:0.1:20];
f = abs(x.^4 + x.^2 - 3*x);
f1prime = 4*x.^3 + 2*x - 3;
f2prime = 12*x.^2 + 2;
```

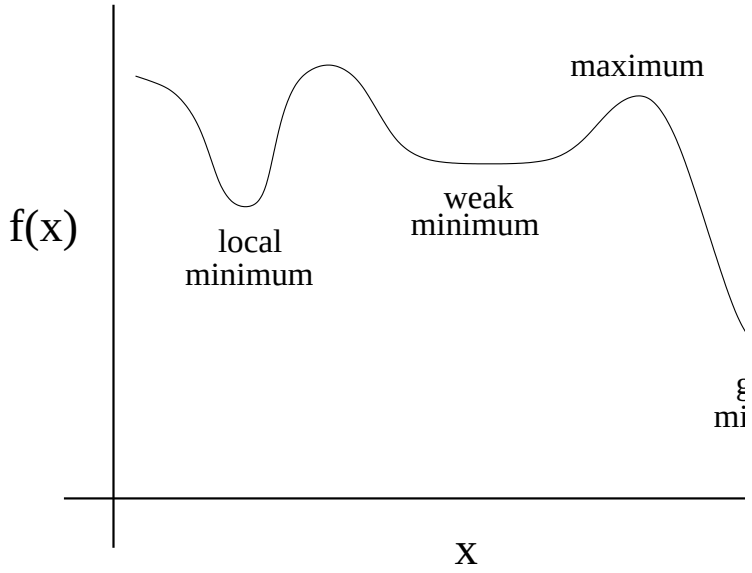

Figure 2.1: Functional with pathological cases: local minima, weak minima surrounded by flat topography, strong global minimum, and a maximum.

```
plot(x,f,'blue',x,f1prime,'green',x,f2prime,'red');
zoom; grid on;

% Start Newton 1-D to find zeros

tolerance = 0.01;
deltax = tolerance + 0.1; % to make sure, while starts
xinit = 01;  % guess

i = 0;
max = 10; % maximum number of iterations, in case it doesn't converge

xact = xinit;
while abs(deltax) > tolerance,
        i = i+1; f = xact.^4 + xact.^2 - 3*xact;
        f1prime = 4*xact.^3 + 2*xact - 3;
        f2prime = 12*xact.^2 + 2;

        deltax = - f/f1prime; xnew = xact + deltax;
        table = [deltax, xnew];
        fprintf('%6.2f %12.8f\n', table);

        xact = xnew;
        if i == max break; end
end
```

2.2.2 2-D Newton's Method

Let $f(x, y)$ be a differentiable functional dependent on the scalar variables x and y. The goal is to use Newton's method to find the optimal $\mathbf{x}^* = \mathbf{x_o} + \mathbf{\Delta x}^*$ that minimizes $f(\mathbf{x})$, where $\mathbf{x} = (x, y)^T$. Expanding $f(\mathbf{x}_o + \Delta\mathbf{x})$ about $\mathbf{x}_o$ we get

$$f(\mathbf{x}_o + \Delta\mathbf{x}) \approx f(\mathbf{x}_o) + \frac{\partial f(\mathbf{x}_o)}{\partial x}\Delta x + \frac{\partial f(\mathbf{x}_o)}{\partial y}\Delta y + \frac{1}{2}\frac{\partial^2 f(\mathbf{x}_o)}{\partial x^2}\Delta x^2$$

$$+ \quad \frac{1}{2}\frac{\partial^2 f(\ \mathbf{x}_o)}{\partial y^2}\Delta y^2 + \frac{\partial^2 f(\ \mathbf{x}_o)}{\partial y \partial x}\Delta y \Delta x, \tag{2.5}$$

where cubic and higher order terms have been neglected. Similar to the 1-D case, the Newton formula is found by setting the 2-D gradient at $\mathbf{x}^*$ equal to zero, (i.e., $\Delta f(\ \mathbf{x} + \Delta\ \mathbf{x}^*) = 0$) and solving the resulting 2x2 set of equations for $\Delta\ \mathbf{x}^*$. Explicitly,

$$\begin{aligned} \frac{\partial f(\ \mathbf{x}_o + \Delta \mathbf{x}^*)}{\partial x} &= \frac{\partial f(\ \mathbf{x}_o)}{\partial x} + \frac{\partial^2 f(\ \mathbf{x}_o)}{\partial x^2}\Delta x^* + \frac{\partial^2 f(\ \mathbf{x}_o)}{\partial y \partial x}\Delta y^*, \\ &= 0. \end{aligned} \tag{2.6}$$

$$\begin{aligned} \frac{\partial f(\ \mathbf{x} + \Delta \mathbf{x}^*)}{\partial y} &= \frac{\partial f(\ \mathbf{x}_o)}{\partial y} + \frac{\partial^2 f(\ \mathbf{x}_o)}{\partial y^2}\Delta y^* + \frac{\partial^2 f(\ \mathbf{x}_o)}{\partial y \partial x}\Delta x^*, \\ &= 0. \end{aligned} \tag{2.7}$$

Equations 2.6 and 2.7 can be rearranged and written in matrix notation

$$\mathbf{H}\boldsymbol{\Delta}\mathbf{x}^* = -\mathbf{g}, \tag{2.8}$$

where the Hessian matrix $\mathbf{H}$ and gradient vector $\mathbf{g}$ are given by

$$\mathbf{H} = \begin{vmatrix} \dfrac{\partial^2 f}{\partial x^2} & \dfrac{\partial^2 f}{\partial x \partial y} \\ \dfrac{\partial^2 f}{\partial x \partial y} & \dfrac{\partial^2 f}{\partial y^2} \end{vmatrix}, \tag{2.9}$$

$$\mathbf{g} = \begin{vmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{vmatrix}. \tag{2.10}$$

Solving for $\Delta\mathbf{x}^*$ in equation 2.8 gives

$$\Delta\mathbf{x}^* = -\mathbf{H}^{-1}\mathbf{g}, \tag{2.11}$$

for the optimal move. For higher-order quadratic functions the iterative formula

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{H}^{-1}\mathbf{g}^{(k)}, \tag{2.12}$$

is used to update the solution. Here $\nabla f(\ \mathbf{x}^{(k)}) = (\partial f/\partial x, \partial f/\partial y)^T = \mathbf{g^{(k)}}$ so that the outer product $\nabla\nabla^T$ applied to $f(x)$ is exactly the Hessian matrix given in equation 2.7. Also, equation 2.5 can be expressed in the compact notation

$$f(\ \mathbf{x}) \ = \ f(\ \mathbf{x}_o) + \mathbf{g}^T \cdot \Delta\ \mathbf{x} + 1/2\Delta\ \mathbf{x}^T \cdot \mathbf{H} \cdot \Delta\ \mathbf{x}. \qquad (2.13)$$

$\hat{r}^T \cdot H \cdot \hat{r}$ **= Curvature.** The Hessian matrix contains information about the curvature or type of bumps associated with $f(\mathbf{x})$. In fact, the functional's curvature at $\mathbf{x}$ along the unit vector direction $\hat{\mathbf{r}}$ is given by $\hat{\mathbf{r}}^{\mathbf{T}} \cdot \mathbf{H} \cdot \hat{\mathbf{r}}$. To show this assume that $\mathbf{x}$ is parameterized by the scalar α so that

$$\mathbf{x} \ = \ \mathbf{x_0} + \alpha\hat{\mathbf{r}}, \qquad (2.14)$$

for some fixed $\mathbf{x_0}$ and the unit vector $\hat{\mathbf{r}}$. Therefore, the curvature of $f(\mathbf{x})$ along the direction $\hat{\mathbf{r}}$ is given by $d^2f(\mathbf{x})/d\alpha^2$, which can be obtained by inserting equation 2.14 into equation 2.13 and differentiating twice w/r to α to get

$$\frac{d^2f(\mathbf{x})}{d\alpha^2} \ = \ \hat{\mathbf{r}}^T \cdot H \cdot \hat{\mathbf{r}}. \qquad (2.15)$$

Hessian Eigenvalues Determine Geometry. The sign and magnitude of the eigenvalues λ_1 and λ_2 of H determine the shape of $f(\ \mathbf{x})$. This can be shown by replacing $\mathbf{x}_o + \Delta\mathbf{x}^*$ by $\mathbf{x}^* + \alpha\mathbf{e_1} + \beta\mathbf{e_2}$, where $\mathbf{e_i}$ is the i^{th} orthonormal eigenvector of the symmetric matrix $\mathbf{H}$, and α and β are scalars. Expanding $f(\ \mathbf{x})$ about the minimum point $\mathbf{x}^*$ so that equation 2.13 becomes

$$\begin{aligned} f(\mathbf{x}^* + \alpha\mathbf{e_1} + \beta\mathbf{e_2}) \ &= \ f(\mathbf{x}^*) + \alpha^{\mathbf{2}}\mathbf{e_1}^{\mathbf{T}} \cdot \mathbf{H} \cdot \mathbf{e_1} + \beta^{\mathbf{2}}\mathbf{e_2}^{\mathbf{T}} \cdot \mathbf{H} \cdot \mathbf{e_2}, \\ &= \ f(\mathbf{x}^*) + \lambda_{\mathbf{1}}\alpha^{\mathbf{2}}\mathbf{e_1}^{\mathbf{T}} \cdot \mathbf{e_1} + \lambda_{\mathbf{2}}\beta^{\mathbf{2}}\mathbf{e_2}^{\mathbf{T}} \cdot \mathbf{e_2}, \\ &= \ f(\mathbf{x}^*) + \lambda_{\mathbf{1}}\alpha^{\mathbf{2}} + \lambda_{\mathbf{2}}\beta^{\mathbf{2}}, \end{aligned} \qquad (2.16)$$

where the gradient term $\mathbf{g^T} \cdot \mathbf{\Delta}\ \mathbf{x}$ is dropped because it is equal to zero at the minimum point $\mathbf{x}^*$. Since $\mathbf{H}$ is a symmetric matrix then the

eigenvalues are real, so that for positive definite $\mathbf{H}$ (i.e., $\mathbf{H}$ has only positive eigenvalues) any move along an eigenvector direction from $\mathbf{x}^*$ will increase the value of the functional. Hence, $f(\mathbf{x})$ describes a bowl-like surface around the minimum point $\mathbf{x}^*$ (see left column of plots in Figure 2.2). Large positive eigenvalues suggest that small changes in position lead to large changes in $f(\mathbf{x})$ so that the bowl has steeply curving sides; conversely, small positive eigenvalues suggest a bowl with gently curving sides.

If the Hessian is negative definite (i.e., $\mathbf{H}$ has only positive eigenvalues) then equation 2.16 says that any move along an eigenvector direction will decrease the functional, i.e., $f(\mathbf{x})$ describes an inverted bowl about the *maximal* point $\mathbf{x}^*$. If the Hessian is indefinite (both positive and negative eigenvalues) then a move along one eigenvector direction will decrease the functional value while a move along the other eigenvector direction will increase the functional value. This latter surface describes the saddle shown in Figure 2.2f.

Exercises

1. Find the zeros of $f(x) = x^4 + x^2 - 3x$ using the 1-D Newton method and the MATLAB script given in the previous section. Test the convergence rate sensitivity to different starting points.

2. Write a 2-D Newton MATLAB script that solves the minimization of the Rosenbrock function $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$. Plot out the Rosenbrock function in an x-y plot and comment about why the curved contours indicate a functional with a higher order than second-degree. Show that the convergence rate strongly depends on the starting point.

2.2.3 Multidimensional Newton's Method

Let $f(\mathbf{x})$ be a differentiable functional dependent on the N-dimensional vector $\mathbf{x}$. The 2-D Newton's method is easily extended to the N-dimensional case by expanding $f(\mathbf{x})$ about $\mathbf{x}_o$ in an N-dimensional Taylor's series

$$f(\mathbf{x}_o + \Delta\,\mathbf{x}) \approx f(\mathbf{x}_o) + \mathbf{g}^T \cdot \mathbf{\Delta}\,\mathbf{x} + \frac{1}{2}\mathbf{\Delta}\,\mathbf{x}^\mathbf{T} \cdot \mathbf{H} \cdot \mathbf{\Delta}\,\mathbf{x}, \qquad (2.17)$$

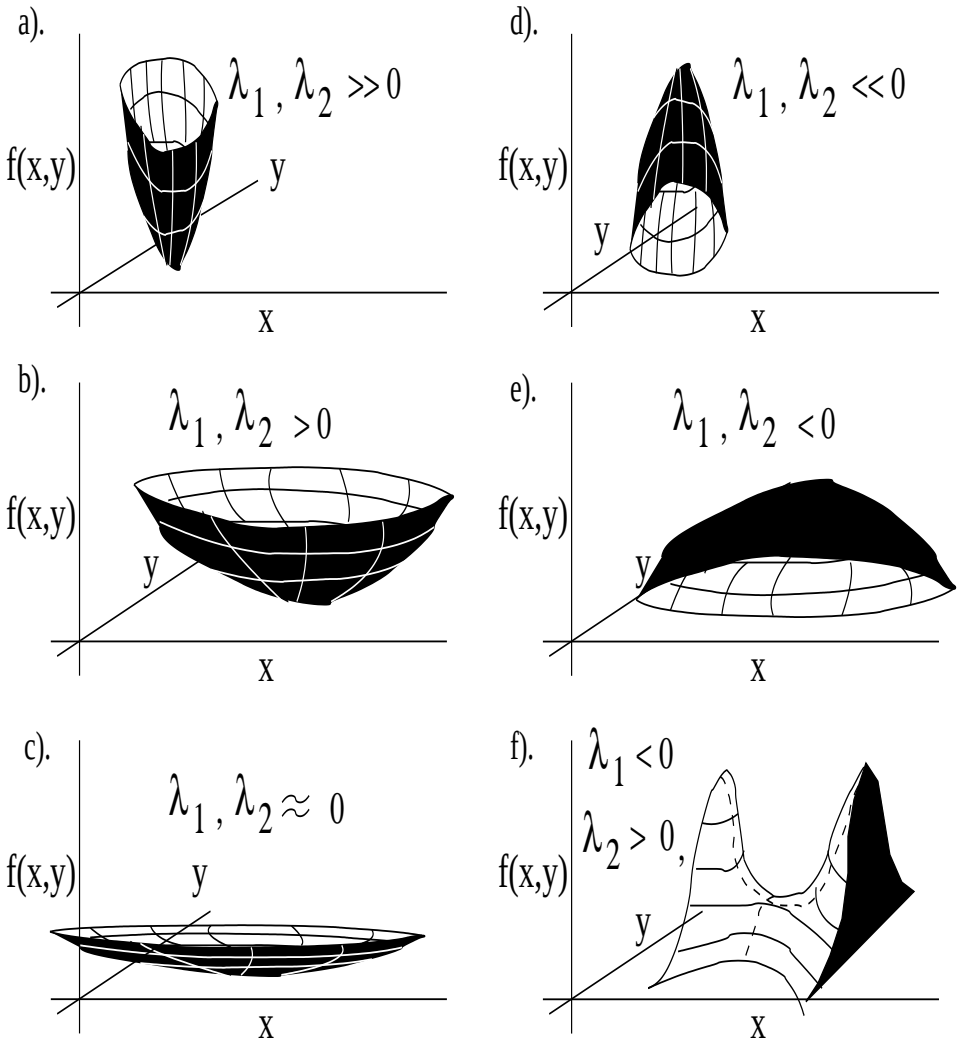

Figure 2.2: Plots of functionals with different curvatures. Left column of figures are associated with positive eigenvalues while the right column corresponds to examples with negative eigenvalues.

where the cubic and higher order terms have been dropped. As before, the gradient of $f(\mathbf{x}_o + \Delta\,\mathbf{x})$ is set equal to zero at $\mathbf{x}_o + \Delta\,\mathbf{x}^*$ so that $\Delta\,\mathbf{x}^*$ can be iteratively solved by equation 2.12. Except now, $\mathbf{H}$ is an NxN Hessian with elements $H_{ij} = \partial^2 f/\partial x_i \partial x_j$ and the Nx1 gradient vector $\mathbf{g}$ has elements $g_i = \partial f/\partial x_i$.

In many geophysics problems, the misfit functional is highly non-linear so that many Newton searches are needed. Indeed, a step of unity along the Newton direction may not reduce the misfit function (Gill et al., 1981). Thus the step length denoted by the scalar value α is used to give the formula:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha \mathbf{H}^{-1}_{(k)} \mathbf{g}^{(k)}, \tag{2.18}$$

where the k subscript in the Hessian indicates updating of the Hessian components after each iteration. Equation 2.18 describes the non-linear Newton method with line search (Fletcher, 1987). Line search methods that determine the optimal value of α will be discussed in a later chapter.

2.2.4 Quasi-Newton Condition

It is well known that a finite-difference approximation to the second derivative of a function is proportional to the difference between the first derivatives at neighboring points:

$$\partial^2 f(x)/\partial x^2\; dx \;\;\approx\;\; \partial f(x+dx)/\partial x - \partial f(x)/\partial x, \tag{2.19}$$

where dx is the spatial increment between the evaluation points. This formula is a special case of the quasi-Newton condition, which relates the Hessian (second derivatives of quadratic misfit function) to the difference between the misfit gradient at neighboring points.

Quasi-Newton Condition

$$\mathbf{H} \cdot (\,\mathbf{x}^{(k+1)} - \,\mathbf{x}^{(k)}) \;\;=\;\; \mathbf{g}^{(k+1)} - \,\mathbf{g}^{(k)}. \tag{2.20}$$

The QN condition can easily be proved by forming the gradient of equation 2.17 at $\mathbf{x}_o + \Delta\,\mathbf{x}$:

$$\nabla f(\,\mathbf{x_o} + \mathbf{\Delta}\,\mathbf{x}) \;\;=\;\; \mathbf{g}(\,\mathbf{x}_o) + \mathbf{H} \cdot \Delta\,\mathbf{x}. \tag{2.21}$$

Defining $\nabla f(\ \mathbf{x_o} + \mathbf{\Delta}\ \mathbf{x}) = \mathbf{g}^{(k+1)}$, $\mathbf{g}(\ \mathbf{x}_o) = \mathbf{g}^{(k)}$ and $\Delta\ \mathbf{x} =\ \mathbf{x}^{(k+1)} -\ \mathbf{x}^{(k)}$ we get the QN condition in equation 2.20. Note the similarity between the 2nd-order difference equation and its multideimensional generalization to the QN formula 2.20. We will denote $\mathbf{H} \cdot (\ \mathbf{x}^{(k+1)} -\ \mathbf{x}^{(k)})$ as the unnormalized curvature vector along the direction $\ \mathbf{x}^{(k+1)} -\ \mathbf{x}^{(k)}$ because the curvature of the functional along this direction is given by $(\ \hat{\mathbf{x}}^{(k+1)} -\ \hat{\mathbf{x}}^{(k)})^T \cdot \mathbf{H} \cdot (\ \hat{\mathbf{x}}^{(k+1)} -\ \hat{\mathbf{x}}^{(k)})$ (see equation 2.15). In a following chapter, we will use the QN property to derive a practical generalization of the steepest descent method, known as the conjugate gradient method.

2.2.5 Gauss-Newton Method

The Gauss-Newton method was discussed in the first chapter in the context of minimizing the sum of squared traveltime residuals. For the general case of M non-linear equations defined as

$$\begin{aligned} r_i(\ \mathbf{x}) &= \mathbf{L}_i\ \mathbf{x} - t_i \\ &\approx 0 \qquad i\epsilon[1, 2, ...M] \end{aligned} \tag{2.22}$$

where $r_i(\ \mathbf{x})$ is known as the data residual, the difference between the ith predicted data given by $\mathbf{L}_i\ \mathbf{x}$ and the ith observed data t_i; $\mathbf{L}_i$ is the modeling operator that acts on the model vector $\ \mathbf{x}$. The goal is to find the model vector $\ \mathbf{x}$ that minimizes the residuals in some sense. Thus, a cost functional ϵ is defined as the pth power of the l^p norm (Kreyszig, 1978) of the residual

$$\begin{aligned} \epsilon &= ||\mathbf{r}||^p, \\ &= \sum_{i=1}^{N} |r_i|^p, \end{aligned} \tag{2.23}$$

and for $n = 2$ is called the l^2 norm or least squares solution if we find the optimal $\ \mathbf{x}$ that minimizes

$$\epsilon = 1/2 \sum_{i=1} r_i^2, \tag{2.24}$$

where the 1/2 factor was included for convenience. This misfit function is a non-negative functional that is non-linearly related to the model parameters $\mathbf{x}$. Thus, the Newton iterative formula in equation 2.12 can be used to find the optimal solution.

Explicitly, the jth component of the gradient is given by

$$\begin{aligned} g_j &= \partial\epsilon/\partial x_j \\ &= 2\sum_{i=1} r_i(\ \mathbf{x})\partial r_i/\partial x_j, \end{aligned} \tag{2.25}$$

where the components $\partial r_i/\partial x_j$ make up the elements of what is known as the Jacobian matrix; the Jacobian elements are composed of the 1st derivatives of each residual. The Hessian is obtained by taking second derivatives of the misfit function, i.e.,

$$\begin{aligned} H_{jk} &= \partial^2\epsilon/\partial x_j\partial x_k \\ &= 2\sum_{i=1}^{N}[r_i(\ \mathbf{x})\partial^2 r_i/\partial x_k\partial x_j + \partial r_i/\partial x_k\ \partial r_i/\partial x_j]. \end{aligned} \tag{2.26}$$

More compactly, the above equation can be represented in matrix-vector notation as

$$\mathbf{H} = \sum_{i=1}^{M}\mathbf{T}_i r_i\ +\ \mathbf{L}^T\ \mathbf{L}, \tag{2.27}$$

where the Jacobian is given by $L_{ij} = \partial r_i/\partial x_j$ and the third-rank tensor $[T_i]_{jk} = \partial^2 r_i/\partial x_k\partial x_j$. Substituting the above Hessian into the Newton formula gives

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [\sum_{i=1}^{M}\mathbf{T}_i r_i\ +\ \mathbf{L}^T\ \mathbf{L}]^{-1}\mathbf{g}^{(k)}, \tag{2.28}$$

and is called the large residual Gauss-Newton method.

Small Residual GN

For small residuals where $\mathbf{r}$ is small then the third-rank tensor is neglected to give the small-residual Gauss-Newton method:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - [\mathbf{L}^T\ \mathbf{L}]^{-1}\mathbf{g}^{(k)}, \tag{2.29}$$

where $\mathbf{g} = \mathbf{L}^T\mathbf{r}$.

The small residual GN method is almost universally used by all tomographers, mainly because the second derivatives do not need to be explicitly computed. Only the Jacobian matrix is computed, which can be used to form the approximation to the Hessian $\mathbf{L}^T\mathbf{L}$.

2.2.6 Non-Linear GN

If the Hessian depends on the model values $\mathbf{x}$ then the data and model are non-linearly related. Thus one iteration of the GN method will not minimize the misfit function. Instead, an initial model $\mathbf{x}^{(0)}$ is specified and the non-linear GN method is employed:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha[\sum_{i=1} T_i r_i + \mathbf{L^T}\ \mathbf{L}]^{-1}\ \mathbf{g}^{(k)}, \tag{2.30}$$

where the superscript k denotes the kth iteration and α denotes a scalar step length (to be discussed in a later chapter). If the initial model is close to the actual model then this method yields useful results after a few iterations. However, many seismic imaging problems are plagued by many *local* minima that tend to trap iterative solution methods. Later chapters will address this still bothersome issue.

2.2.7 Reweighted Least Squares

Some data residuals may be less reliable than others, and so it seems plausible to downweight these "noisy" data with a weighting factor w_i such that $0 < w_i < 1$. In this case the sum of the squared residuals become

$$\epsilon = 1/2\sum_{i=1} w_i r_i^2, \tag{2.31}$$

and the Gauss-Newton solution is known as a weighted least squares solution. Here the gradient in equation 2.25 becomes

$$g_j = 2\sum_{i=1} w_i r_i(\ \mathbf{x})\partial r_i/\partial x_j, \tag{2.32}$$

and the small-residual Hessian is

$$H_{jk} = 2\sum_{i=1}[\partial r_i/\partial x_k \; w_i \partial r_i/\partial x_j]. \tag{2.33}$$

The *reweighted* least squares algorithm is obtained by setting $w_i = 1/(|r_i| + \lambda)$ (where λ is a small positive stabilizing factor), and r_i is the residual from the previous iteration. In this way data points with large residuals are likely to be excluded in influencing the outcome of the optimization. It can be shown (Nolet, 1987) that the reweighted least squares method can be obtained by minimization of the l_1 norm misfit function.

2.3 Cures for Ill-Conditioning

If $\mathbf{H}$ is ill-conditioned then a small move along one of the eigenvector axes can lead to enormous and questionable changes in the model. For example, in the 2-D case the gradient vector $\mathbf{g}$ can be expanded in terms of the eigenvectors $\mathbf{e_1}$ and $\mathbf{e_2}$ of $\mathbf{H}$ to yield $\mathbf{g} = g_1\mathbf{e_1} + g_2\mathbf{e_2}$, where $\mathbf{He_i} = \lambda_\mathbf{i}\mathbf{e_i}$ and λ_i is the i^{th} eigenvalue. Plugging $\mathbf{g}$ into equation 2.11 gives

$$\Delta\mathbf{x}^* = -(g_1/\lambda_1)\mathbf{e_1} - (g_2/\lambda_2)\mathbf{e_2}. \tag{2.34}$$

If the i^{th} eigenvalue is almost zero then $|\Delta x_i| = |g_i/\lambda_i| >> 0$, resulting in a large unstable move along the i^{th} eigenvector axis; the move is considered unstable because a small amount of data noise will greatly change the solution. This is similar to the model being a discontinuous function of the data. Equivalently, the functional's topography along this eigendirection will appear to be a long nearly flat valley similar to that shown in Figure 2.2c. In this case, many different models $\mathbf{x}$'s along the valley floor can give almost the same value of the functional.

2.3.1 Regularization

To remedy this ill-conditioning problem, we resort to a regularization method, the simplest being the Levenberg-Marquardt method. An alternative remedy to increase convergence is preconditioing, which will be discussed in the next section.

In the Levenberg-Marquardt regularization method, the NxN $\mathbf{H}$ matrix in equation 2.5 is replaced by $(\mathbf{H} + \lambda\mathbf{I})$, so that we have the modified Newton formula

$$[\mathbf{H} + \lambda\mathbf{I}]\Delta\ \mathbf{x} = -\mathbf{g}, \tag{2.35}$$

where $\mathbf{I}$ is an NxN identity matrix and λ is some small positive scalar. The eigenvectors for $(\mathbf{H}+\lambda\mathbf{I})$ are the same as for $\mathbf{H}$, but the eigenvalues become $\lambda_i + \lambda$. Therefore the unstable move $\Delta x_i \rightarrow g_i/(\lambda_i + \lambda)$ is damped to a more reasonable value.

The Levenberg-Marquardt method approaches the Newton method as $\lambda \rightarrow 0$, i.e.,

$$\lim_{\lambda\rightarrow 0}(\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{g} = -\mathbf{H^{-1}}\mathbf{g}. \tag{2.36}$$

On the other hand, if $\lambda \rightarrow large$ then the Levenberg-Marquardt method approaches the gradient or steepest descent method, i.e.,

$$\lim_{\lambda\rightarrow large}(\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{g} = -\mathbf{g}/\lambda. \tag{2.37}$$

In this last case, the gradient move is along the steepest descent direction or the direction perpendicular to the contour's tangent. If the contours are round then the steepest descent direction points to the bullseye.

These two limiting cases are shown in Figure 2.3 where the Levenberg-Marquardt direction is between the steepest descent and Newton directions (Lines and Treitel, 1988). In practice, the value of λ is set to be large (about 1.0 percent of the largest diagonal value of $\mathbf{H}$) for the initial iteration, and is then gradually reduced as the iterations proceed until the average residual is about the same as the expected data error. Large values of the damping parameter tend to suppress the high frequency components of the inverted slowness model.

The Levenberg-Marquardt method is a special case of the regularized Newton method with small residuals. In the regularized Newton method, a regularization (i.e., stabilizing or penalty function) functional $p(\ \mathbf{x})$ is appended to the misfit functional in equation 2.17 to give

$$f(\ \mathbf{x}) \ \ \rightarrow \ \ f(\ \mathbf{x}) + \lambda\ p(\ \mathbf{x}), \tag{2.38}$$

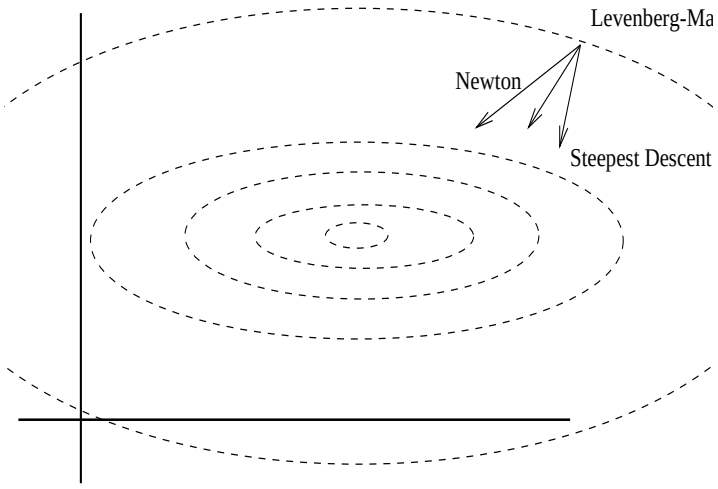

Figure 2.3: Contours associated with the $f(\mathbf{x})$ functional and the vectors associated with the Newton, Levenberg-Marquardt and steepest descent vectors.

where $\lambda > 0$ is a scalar sometimes known as the damping constant. The penalty function $p(\mathbf{x})$ is designed to penalize the solution if the iterations wander too far from where we think the model should be. For example, deciding that the solution length should be small is enforced by setting $p(\mathbf{x}) = 1/2||\mathbf{x}||^2$; the consequent Newton formula is equal to the Levenberg-Marquardt equation 2.35. In later chapters we will discuss other terms that can be used to regularize the solution, including difference terms that penalize rough solutions, e.g., $\sum_{i,j}[(x_{ij} - x_{i+1j})^2 + (x_{ij} - x_{ij+1})^2]$, or entropy terms that penalize "unsimple" models (Buck and Macauly, 1994).

If solutions are found for decreasing values of the damping constant, then this is sometimes known as Tikhonov regularization (Groetsch, 1994). Mathematically, we are replacing an ill-posed problem with a well-posed problem whose solution becomes *closer* to that of the original problem as $\lambda \rightarrow 0$. The stabilizing functional is used to guide the solution to be near some a priori estimate of the model.

As an example, if $p(\mathbf{x})$ is set equal to $1/2\Delta\ \mathbf{x}^T \cdot \mathbf{I} \cdot \Delta\ \mathbf{x}$ then it can easily be shown that the resulting Newton formula is equal to the Levenberg-Marquardt formula in equation 2.3. The regularization term

$\lambda\Delta\ \mathbf{x}^T\ \mathbf{I}\ \Delta\ \mathbf{x}$ prefers the model "closest" to the starting point $\Delta\ \mathbf{x} = 0$, while the $f(\ \mathbf{x})$ term prefers the model that satisfies the equation $\mathbf{H}\Delta\ \mathbf{x} = -\mathbf{g}$. The size of the damping parameter λ decides which one of these conflicting demands is emphasized. We will discuss more general regularization operators in later chapters.

Exercises

1. Prove that the Newton method for the functional $\epsilon = f(\ \mathbf{x}) + 0.5\lambda\Delta\ \mathbf{x}^T \cdot \mathbf{I} \cdot \Delta\ \mathbf{x}$ results in the Levenberg-Marquardt formula.

2. Show that a negative value for the damping parameter can lead to a damped Hessian with zero eigenvalues. Assume an undamped Hessian that is SPD.

3. Find the Newton formula for the Rosenbrock function by employing a regularization term that prefers to be close to some a priori model $\mathbf{x}_{apriori}$.

4. Show that the gradient minimization of the l^1 norm misfit function $\epsilon = \sum_i |\ \mathbf{L}_i\ \mathbf{x} - t_i|$ leads to the reweighted least squares method.

5. Are preconditioning and regularization commutative? That is, is preconditioning followed by regularization in an iterative Newton method the same as regularization followed by preconditioning?

Choosing a damping parameter

How do you choose the smallest value of the damping parameter λ? Trial and error usually seems to work, where tests with synthetic data having realistic noise are used to determine smallest value of λ. The discrepancy principle of Morozov (Groetsch, 1993) assigns a value for the damping parameter λ, inverts for the model $\mathbf{x}^{est}$ from the synthetic data $\mathbf{d}$, and from the estimated model generate the estimated data $\mathbf{d}^{est}$. Plot λ vs $||\ \mathbf{d}^{est} - \mathbf{d}||$ and choose the damping parameter associated with the value $||\ \mathbf{d}^{est} - \mathbf{d}||$ that is equal to the estimated residual norm in the actual data (see left plot in Figure 2.4). A more expensive modification of this approach is to experimentally choose λ

that minimizes the data residual (above the estimated data residual threshold) for each iteration in a non-linear iterative method (Constable et al., 1987).

An alternative approach is to plot the length of the model vector vs the length of residual vector, and choose the point on the curve nearest the origin (see right plot in Figure 2.4). In this context, larger model vectors are roughly equivalent to larger model variance. The point nearest the origin is considered to be the optimal tradeoff between increasing data residual and decreasing model vector length. Small values of λ lead to small data residuals but at the expense of large model vectors; conversely, large values of λ lead to small model vectors (or small model variance) but large data residuals (Treitel and Lines, 1982; Jackson, 1972). This procedure is equivalent to using the λ on the *elbow* of the L-shaped curve (Calvetti et al., 1999; Lawson and Hanson, 1972; Hansen, 1992; Hansen and O'Leary, 1993). But there are limitations as pointed out by Hanke (1996).

Two other methods deserve some mention. The first is the so-called singular value truncation method (or sometimes called truncated spectral factorization) which expands the solution in terms of the weighted singular value eigenvectors, similar to equation 2.34, except the summation is truncated at singular values less than a small threshold value (Menke, 1984). Instead of a sharp truncation, smoothly attenuating these low eigenvalue contributions produces better results (Calvetti et al., 2002). Unfortunately, solving for the singular value decomposition is impractical for many realistic tomography problems. The other regularization method is that of the Generalized Cross Validation (GCV) method (Golub et al., 1979), which can be effectively used for large-scale problems (Golub and von Matt, U., 1997). However, this method does not appear to have been tested with geophysical problems to date.

2.3.2 Preconditioning

Preconditioning is a preprocessing step which massages the Hessian so that the original elliptical-like contours in Figure 2.3 become more rounded. In this case, even a steepest descent method will converge in a few steps. The trick is to find an easily computable matrix $\mathbf{B}$ such

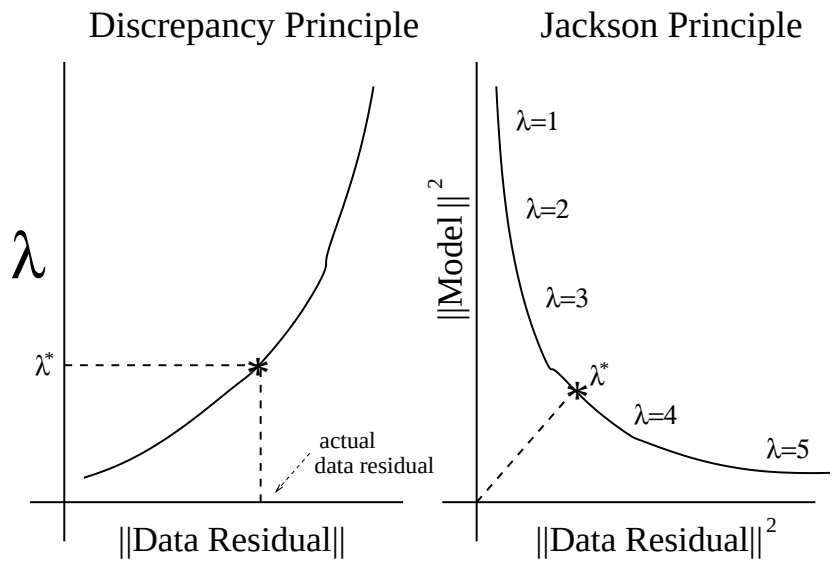

Figure 2.4: Schematic plots illustrating how to choose the optimal damping parameter λ^* using the (left) discrepancy and (right) Jackson principles. The discrepancy principle chooses the λ associated with the length of the actual data residual, while the Jackson principle might choose the point on the curve nearest the origin in a plot of squared lengths of model vector vs residual vector. Honoring the discrepancy principle insures that the model is not severely fitted to the noise.

that $\mathbf{B} \approx \mathbf{H}^{-1}$. Applying $\mathbf{B}$ to equation 2.8 yields

$$\mathbf{BH}\Delta\mathbf{x}^* = -\mathbf{Bg}, \tag{2.39}$$

so that, e.g., the steepest descent formula in equation **??** becomes

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha\mathbf{Bg}, \tag{2.40}$$

which is known as a preconditioned steepest descent method. Obviously, $\mathbf{BH}$ has eigenvalues that are nearly the same value so that the matrix is now well conditioned.

A simple example might be an NxN diagonal matrix where $B_{ii} = 1/H_{ii}$. A more accurate inverse approximation is by employing a incomplete Cholesky factorization of $\mathbf{H}$ into a product of upper and lower triangular matrices, and solve for the approximate inverse of $\mathbf{H}$ by back substitution (Nocedal and Wright, 1999). See Gill et al. (1981) for further details, and Claerbout (2001) for an inexpensive approximation to inverses by the helical transformation. Fomel (2000) discusses novel methods for both data and model regularization.

How is preconditioning related to regularization (see equation 2.38), which also attempts to stabilize an ill-conditioned set of equations? The main difference is that regularization guides the iterated solution to where our biased notion of where the solution ought to be. If the global minima is characterized by a long valley, regularization will take us to that part of the valley that, e.g., is nearest to the smallest solution norm or the smoothest solution. Preconditioning is not so explicitly prejudiced and takes us into the center of the valley no matter how gentle the slope. This might not be a good idea if the data are noisy so that the center of the valley is not near the actual solution. For this reason, I believe regularization is more powerful than preconditioning, particularly if our biases are based on ground truth such as well logs. When possible, my preference is to use both preconditioning and regularization to solve seismic imaging problems.

2.4 Summary

The theory of unconstrained Newton optimization is given for solving non-linear and linear equations. The first step is to form a misfit

function and expand it in a 2nd-order Taylor series about $\mathbf{x}_o$:

$$f(\mathbf{x}_o + \Delta\mathbf{x}) = f(\mathbf{x}_o) + \boldsymbol{\Delta} f(\mathbf{x}_o) \cdot \Delta\mathbf{x} + 1/2\ \Delta\mathbf{x}^T \cdot \mathbf{H} \cdot \Delta\mathbf{x}. \tag{2.41}$$

The 1st-order coefficient in the perturbation parameter $\Delta\mathbf{x}$ is the gradient $\mathbf{g}$ of the misfit function, while the second-order term contains the symmetric Hessian $\mathbf{H}$. For nice convergence properties we assume that the Hessian is positive definite, otherwise there will be non-uniqueness problems (nearly zero-eigenvalues) or poor convergence problems (negative eigenvalues).

The product $\Delta\hat{\mathbf{x}}^T \cdot \mathbf{H} \cdot \Delta\hat{\mathbf{x}}$ is the curvature of the misfit function along the $\Delta\hat{\mathbf{x}}$ direction and governs the convergence rate of the gradient optimization methods. Newton's method provides a search direction that points to the bullseye of a quadratic functional. If the problem is highly non-linear then Newton's method is iteratively used to seek out the local bullseyes defined by the quadratic functional at each iterate.

Major problems in seismic gradient optimization include the following.

- Misfit functions are characterized by many local minima in practical tomography problems. A partial remedy is a multigrid regularization where coarse models are first determined to explain the data, and then the models are iteratively refined with smaller grid spacing (Nemeth et al., 1997). Apparently, coarse grid spacing leads to smoother misfit functions so that the initial iterations avoid the many local minima present in a misfit function parameterized on a finer grid. Sometimes, a different misfit function can be used to avoid the local minima problems.

- Incomplete data due to limited source-receiver coverage, leading to long narrow misfit valleys and non-uniqueness in the solution. Equivalently, many models nearly explain the same data. A partial remedy is to incorporate more data and other types of data into the misfit function, e.g., include both transmission and reflection traveltimes in traveltime tomography (Nemeth et al., 1997).