16

Least Squares Approaches to Diffusion Tensor Estimation

CHENG GUAN KOAY

Summary Box

- Different parametrizations of the diffusion tensor lead to different diffusion tensor representations, e.g., Cholesky, Log-Cholesky, Euler, and Log-Euler representations.
- Cholesky, Log-Cholesky, and Log-Euler representations may be used in diffusion tensor estimation to enforce positive definiteness of the diffusion tensor estimate.
- There are several approaches to estimating the diffusion tensor, e.g., maximum likelihood, least squares, robust, and Bayesian methods.
- Within the class of least squares methods, the connection between them can be established by working with the error terms in each of the objective functions.
- Several methods for performing unconstrained function minimization or nonlinear least squares estimation include the steepest descent method, the Gauss-Newton method, Newton's method, the Levenberg method, the Marquardt method, and the modified full Newton method.
- Other estimation methods are available that are less sensitive to outliers or more appropriate for the Rician-distributed measured signals when the signal-to-noise ratio is low.
- "Seek simplicity and distrust it." —A. N. Whitehead.

Diffusion tensor imaging (DTI) (Basser et al., 1994b) can be viewed as a flexible but coherent data-analytic "pipeline" ranging from signal generation and detection to diffusion tensor estimation (Basser et al., 1994a), tractography (Conturo et al., 1999; Mori et al., 1999), and beyond. In this chapter, we will explore a particular segment of this pipeline—diffusion tensor estimation.

The general subject matter of diffusion tensor estimation is much broader and more difficult to unify than the subset of the various least squares approaches to diffusion tensor estimation. There are several reasons for the lack of unity in the subject matter: 1) there are many approaches to parametric estimation, e.g., the maximum likelihood (Andersson, 2007, 2008; Fillard et al., 2007), the least squares (Basser et al., 1994a; Jones and Basser, 2004; Salvador et al., 2005; Koay et al., 2006a, 2006b), the Bayesian (Andersson, 2007, 2008; Fillard et al., 2007), and the robust (Mangin et al., 2002; Chang et al., 2005) approaches;

16-Jones-Chap-16.indd 272 8/3/2010 9:17:41 AM

2) each of these estimation methods may incorporate a priori information such as intravoxel or intervoxel (spatial) regularization (Wang et al., 2004; Koay et al., 2006b; Andersson, 2007; Fillard et al., 2007) or positive definiteness constraint (Wang et al., 2004; Koay et al., 2006a, 2006b; Andersson, 2007, 2008) to improve the accuracy and precision of the estimate; and (3) additional layers of complexity may be added to these methods by incorporating specific information from other segments of the pipeline to the estimation process (Andersson and Skare, 2002; Koay, 2005; Aksoy et al., 2008). For these reasons, current literature on diffusion tensor estimation may seem somewhat noncohesive. Amid this potpourri of possible combinations of techniques and approaches, there is a certain unity within each *class* of estimation methods (the maximum likelihood, the least squares, the Bayesian, and the robust methods). The unity within the class of least squares methods of estimation is particularly simple and illuminating (Koay et al., 2006b).

The aim of this chapter is, therefore, to provide a straightforward introduction to this class of estimation in diffusion tensor imaging with a special emphasis on the theoretical and algorithmic connections between the various least squares methods through their respective objective functions and the higher-order derivatives of these objective functions. In addition to least squares approaches to diffusion tensor estimation, we will also consider the notion of a diffusion tensor representation (Koay et al., 2007, 2008). This concept is a unifying idea in diffusion tensor imaging, especially in diffusion tensor estimation (Wang et al., 2004; Koay et al., 2006a, 2006b) and error propagation or uncertainty assessment (Hext, 1963; Anderson, 2001; Chang et al., 2007; Koay et al., 2007, 2008).

Diffusion Tensor Representations

We start with the notion of a diffusion tensor representation (Koay et al., 2007). For purposes of definition, a matrix, $\boldsymbol{D}$, is *symmetric* if $\mathbf{D} = \mathbf{D}^T$, where the superscript T denotes matrix transposition. An element of $\Re^{3\times3}$, $\boldsymbol{D}$, is a *positive definite* matrix if $\boldsymbol{x}^T \boldsymbol{D}\boldsymbol{x} > \boldsymbol{0}$ for all nonzero $\boldsymbol{x}$ in $\Re^3$ or if all eigenvalues of $\boldsymbol{D}$ are positive. The diffusion tensor, $\boldsymbol{D}(\boldsymbol{\beta})$, in the ordinary representation, is a 3×3 symmetric positive definite matrix defined by

$$\boldsymbol{D}(\boldsymbol{\beta}) \equiv \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xy} & D_{yz} & D_{zz} \end{pmatrix}, \qquad [16.1]$$

where $\boldsymbol{\beta} = [D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{yz}, D_{xz}]^T$ and the superscript T denotes matrix or vector transposition. $\boldsymbol{D}(\boldsymbol{\beta})$ is symmetric, and so a full characterization of $\boldsymbol{D}(\boldsymbol{\beta})$ requires only six distinct elements, i.e., $\{D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{yz}, D_{xz}\}$. These six elements form the minimal set of parameters.

A symmetric positive definite matrix has many factorizations (Golub and Van Loan, 1996; Pinheiro and Bates, 1996), each with its own minimal set of parameters. Therefore, the ordinary diffusion tensor representation can be parametrized by many different minimal sets of parameters, a process which naturally leads to different diffusion tensor representations.

It is useful to distinguish three different types of mapping among diffusion tensor representations: the global, the local, and the locally linear mappings. A mapping between two distinct diffusion tensor representations is considered *global* if the parameters of the first diffusion tensor representation can be expressed as functions of the parameters of the second diffusion tensor representation and vice versa. If the mapping cannot be expressed globally but can only be given at each point in the parameter space, then it is a *local* mapping. When neither the global mapping nor the local mapping is available, the *locally linear* mapping may be used. These three types of mapping are illustrated below with a few concrete diffusion tensor representations.

To help the reader navigate from one representation and its mappings to another, we present here a schematic diagram showing the relationships between different diffusion tensor representations through coordinate transformations (see Fig. 16.1).

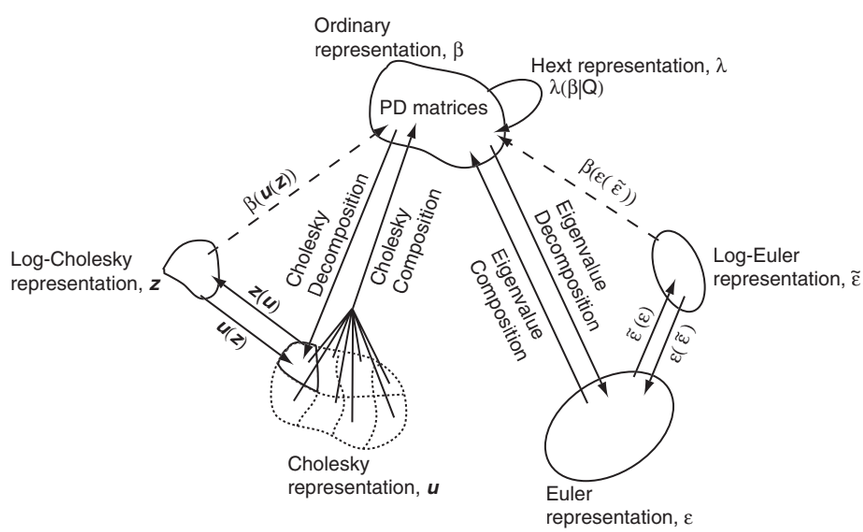

Figure 16.1 Diffusion tensor representations and their mappings. PD, positive definite.

Cholesky Representation

The basic idea of the Cholesky representation is to parametrize the elements of the diffusion tensor in the ordinary representation as functions of the factors in the Cholesky decomposition (Golub and Van Loan, 1996). Specifically, the diffusion tensor, $\boldsymbol{D}(\boldsymbol{u})$, in the Cholesky representation, is defined by

$$\boldsymbol{D}(\boldsymbol{u}) = \boldsymbol{U}^T\boldsymbol{U}, \qquad [16.2]$$

where $\boldsymbol{U}$ is an upper triangular matrix with nonzero diagonal elements given by

$$\boldsymbol{U} = \begin{pmatrix} u_1 & u_4 & u_6 \\ 0 & u_2 & u_5 \\ 0 & 0 & u_3 \end{pmatrix}, \qquad [16.3]$$

and $\boldsymbol{u} = [u_1, u_2, u_3, u_4, u_5, u_6]^T$. In other words, each component of $\boldsymbol{\beta} = [D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{yz}, D_{xz}]^T$ can be expressed as a function of $\boldsymbol{u} = [u_1, u_2, u_3, u_4, u_5, u_6]^T$:

$$\boldsymbol{\beta}(\boldsymbol{u}) \equiv \begin{pmatrix} \beta_1 & (\boldsymbol{u}) \\ \beta_2 & (\boldsymbol{u}) \\ \beta_3 & (\boldsymbol{u}) \\ \beta_4 & (\boldsymbol{u}) \\ \beta_5 & (\boldsymbol{u}) \\ \beta_6 & (\boldsymbol{u}) \end{pmatrix} \equiv \begin{pmatrix} D_{xx} & (\boldsymbol{u}) \\ D_{yy} & (\boldsymbol{u}) \\ D_{zz} & (\boldsymbol{u}) \\ D_{xy} & (\boldsymbol{u}) \\ D_{yz} & (\boldsymbol{u}) \\ D_{xz} & (\boldsymbol{u}) \end{pmatrix} = \begin{pmatrix} u_1^2 \\ u_2^2 + u_4^2 \\ u_3^2 + u_5^2 + u_6^2 \\ u_1 u_4 \\ u_2 u_5 + u_4 u_6 \\ u_1 u_6 \end{pmatrix}. \qquad [16.4]$$

Note that the mapping from $\boldsymbol{u}$ to $\boldsymbol{\beta}$ is a *global* mapping. It is also a many-to-one mapping because it maps eight distinct $\boldsymbol{u}'s$ to the same $\boldsymbol{\beta}$. These eight different $\boldsymbol{u}'s$ are:

$$\begin{aligned}
\boldsymbol{u}_1 &= [u_1, u_2, u_3, u_4, u_5, u_6]^T, \\
\boldsymbol{u}_2 &= [-u_1, -u_2, -u_3, -u_4, -u_5, -u_6]^T, \\
\boldsymbol{u}_3 &= [-u_1, -u_2, u_3, -u_4, -u_5, -u_6]^T, \\
\boldsymbol{u}_4 &= [-u_1, u_2, -u_3, -u_4, u_5, -u_6]^T, \\
\boldsymbol{u}_5 &= [u_1, -u_2, -u_3, u_4, -u_5, u_6]^T, \\
\boldsymbol{u}_6 &= [-u_1, u_2, u_3, -u_4, u_5, -u_6]^T, \\
\boldsymbol{u}_7 &= [u_1, -u_2, u_3, u_4, -u_5, u_6]^T, \text{ and} \\
\boldsymbol{u}_8 &= [u_1, u_2, -u_3, u_4, u_5, u_6]^T.
\end{aligned} \qquad [16.5]$$

In a constrained estimation of the diffusion tensor, which we will discuss later, the minimization of a least squares objective function is carried out by searching for an optimal point in the space of $\boldsymbol{u}$ rather than $\boldsymbol{\beta}$. The implication of the many-to-one property of the mapping from $\boldsymbol{u}$ to $\boldsymbol{\beta}$ is that the optimal point in the space of $\boldsymbol{u}$ is not unique. However, the uniqueness of $\boldsymbol{u}$ or $\boldsymbol{U}$ is not important here as long as we have the optimal point $\boldsymbol{\beta}$ from any one of the eight optimal points in the space of $\boldsymbol{u}$. In cases when the uniqueness of $\boldsymbol{u}$ or $\boldsymbol{U}$ is important, e.g., if one is interested in performing classification on tensors through the factors in the Cholesky decomposition, a unique $\boldsymbol{u}$ or $\boldsymbol{U}$ can be obtained by constraining the diagonal elements of $\boldsymbol{U}$ to be positive. This procedure, the Log-Cholesky representation, is discussed in the next section.

For completeness, here we provide the mapping from $\boldsymbol{u}$ to $\boldsymbol{\beta}$—the Cholesky decomposition, which is also a global mapping:

$$\boldsymbol{u}(\boldsymbol{\beta}) \equiv \begin{pmatrix} u_1(\boldsymbol{\beta}) \\ u_2(\boldsymbol{\beta}) \\ u_3(\boldsymbol{\beta}) \\ u_4(\boldsymbol{\beta}) \\ u_5(\boldsymbol{\beta}) \\ u_6(\boldsymbol{\beta}) \end{pmatrix} = \begin{pmatrix} \det(\boldsymbol{D}_1)^{1/2} \\ (\det(\boldsymbol{D}_2)/\det(\boldsymbol{D}_1))^{1/2} \\ (\det(\boldsymbol{D}_3)/\det(\boldsymbol{D}_2))^{1/2} \\ D_{xy}/\det(\boldsymbol{D}_1)^{1/2} \\ \dfrac{D_{xy}D_{yz} - D_{xy}D_{xz}}{\det(\boldsymbol{D}_1)^{1/2}\det(\boldsymbol{D}_2)^{1/2}} \\ D_{xz}/\det(\boldsymbol{D}_1)^{1/2} \end{pmatrix}, \qquad [16.6]$$

where $\boldsymbol{D}_1 = D_{xx}$, $\boldsymbol{D}_2 = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix}$, $\boldsymbol{D}_3 = \boldsymbol{D}(\boldsymbol{\beta})$, and det(.) is the matrix determinant.

Log-Cholesky Representation

The basic idea of enforcing uniqueness of $\boldsymbol{u}$ or $\boldsymbol{U}$ discussed in the previous section is straightforward. A simple example illustrates this idea: suppose we have a routine (or "black-box") that can perform only unconstrained minimization of a single-variable function and our interest is in finding a positive number, $\hat{x}$, such that the function, $f(\hat{x})$, is at a minimum. We may assume further that $f(x)$ is ill defined or problematic whenever x is negative. It is clear that the minimization routine cannot be used directly on this function because the function may be evaluated at points that belong to the negative axis, which may cause the minimization routine to crash.

This conundrum can be resolved by applying a simple reparametrization on x, i.e., $y = \ln(x)$ (hence, the "Log" in the title of this subsection) and by applying the minimization routine on a new function, $g(y)$, defined by $g(y) \equiv f(e^y)$. Therefore, if $\hat{y}$ is a point at which the new function g is minimum, then $\hat{x} = \exp(\hat{y})$ will always be positive.

It should be clear from the above example that a unique $\boldsymbol{u}$ or $\boldsymbol{U}$ can be found by applying the same technique of reparametrization based on the logarithm function on the diagonal elements of $\boldsymbol{U}$, i.e., $z \equiv [z_1, z_2, z_3, z_4, z_5, z_6]^T \equiv [\ln u_1, \ln u_2, \ln u_3, u_4, u_5, u_6]^T$. Specifically, the diffusion tensor, $\boldsymbol{D}(z)$, in the Log-Cholesky representation, is defined as

$$\boldsymbol{D}(z) = \boldsymbol{Z}^T\boldsymbol{Z}, \qquad [16.7]$$

where $\boldsymbol{Z}$ is an upper triangular matrix given by

$$\boldsymbol{Z} = \begin{pmatrix} e^{Z_1} & z_4 & z_6 \\ 0 & e^{Z_2} & z_5 \\ 0 & 0 & e^{Z_3} \end{pmatrix}. \qquad [16.8]$$

Similar to the previous section, each component of $\boldsymbol{\beta} = [D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{yz}, D_{xz}]^T$ can be expressed as a function of $\boldsymbol{z} = [z_1, z_2, z_3, z_4, z_5, z_6]^T$:

$$\boldsymbol{\beta}(z) \equiv \begin{pmatrix} D_{xx} & (z) \\ D_{yy} & (z) \\ D_{zz} & (z) \\ D_{xy} & (z) \\ D_{yz} & (z) \\ D_{xz} & (z) \end{pmatrix} = \begin{pmatrix} e^{2z_1} \\ e^{2z_2} + z_4^2 \\ e^{2z_3} + z_5^2 + z_6^2 \\ e^{z_1} z_4 \\ e^{z_2} z_5 + z_4 z_6 \\ e^{z_1} z_6 \end{pmatrix} \qquad [16.9]$$

The Log-Cholesky representation was presented here for completeness; however, it will not be needed for the diffusion tensor estimation or error propagation that follows. As mentioned earlier, the Log-Cholesky representation will be essential if one is interested in performing classification on tensors through the factors in the Cholesky decomposition because the Log-Cholesky representation is unique.

Euler Representation

The Euler representation is based on another well-known matrix decomposition, the eigenvalue decomposition of a symmetric matrix. That is, the diffusion tensor, $\boldsymbol{D}(\varepsilon)$, in the Euler representation, is defined by

$$\boldsymbol{D}(\varepsilon) = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T, \qquad [16.10]$$

where $\varepsilon \equiv [\lambda_1, \lambda_2, \lambda_3, \theta, \phi, \psi]^T$ and $\boldsymbol{Q}$ is an orthogonal matrix parametrized by the Euler angles:

$$\boldsymbol{Q} \equiv \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix} \equiv \boldsymbol{R}_z(\phi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_z(\psi), \qquad [16.11]$$

and $\boldsymbol{\Lambda}$ is a diagonal matrix parametrized by three eigenvalues:

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}. \qquad [16.12]$$

A matrix, Q, is *orthogonal* if $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$, where $\boldsymbol{I}$ denotes the identity matrix.

Note that the rotation matrices, $\boldsymbol{R}_x(\Omega)$, $\boldsymbol{R}_y(\Omega)$, and $\boldsymbol{R}_x(\Omega)$, represent rotations through the angle Ω around the x, y, and z axes, respectively, and are defined below:

$$\boldsymbol{R}_x(\Omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\Omega) & -\sin(\Omega) \\ 0 & \sin(\Omega) & \cos(\Omega) \end{pmatrix},$$

$$\boldsymbol{R}_y(\Omega) = \begin{pmatrix} \cos(\Omega) & 0 & \sin(\Omega) \\ 0 & 1 & 0 \\ -\sin(\Omega) & 0 & \cos(\Omega) \end{pmatrix},$$

and

$$\boldsymbol{R}_z(\Omega) = \begin{pmatrix} \cos(\Omega) & -\sin(\Omega) & 0 \\ \sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \qquad [16.13]$$

The explicit expression of each component of $\boldsymbol{\beta} = [D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{yz}, D_{xz}]^T$ as a function of $\varepsilon \equiv [\lambda_1, \lambda_2, \lambda_3, \theta, \phi, \psi]^T$ is slightly more complicated, but it is still tractable:

$$\boldsymbol{\beta}(\boldsymbol{\varepsilon}) \equiv \begin{pmatrix} D_{xx} & (\varepsilon) \\ D_{yy} & (\varepsilon) \\ D_{zz} & (\varepsilon) \\ D_{xy} & (\varepsilon) \\ D_{yz} & (\varepsilon) \\ D_{xz} & (\varepsilon) \end{pmatrix} = \begin{pmatrix} Q_{11}^2\lambda_1 + Q_{12}^2\lambda_2 + Q_{13}^2\lambda_3 \\ Q_{21}^2\lambda_1 + Q_{22}^2\lambda_2 + Q_{23}^2\lambda_3 \\ Q_{31}^2\lambda_1 + Q_{32}^2\lambda_2 + Q_{33}^2\lambda_3 \\ Q_{11}Q_{21}\lambda_1 + Q_{12}Q_{22}\lambda_2 + Q_{13}Q_{23}\lambda_3 \\ Q_{21}Q_{31}\lambda_1 + Q_{22}Q_{32}\lambda_2 + Q_{23}Q_{33}\lambda_3 \\ Q_{11}Q_{31}\lambda_1 + Q_{12}Q_{32}\lambda_2 + Q_{13}Q_{33}\lambda_3 \end{pmatrix}. \qquad [16.14]$$

Note that the components of $\boldsymbol{Q}$ are functions of the Euler angles, $\{\theta, \phi, \psi\}$, defined in Eq. [16.11].

Although the Euler representation has been used in DTI error propagation, in its current form it cannot be used in a constrained estimation of the diffusion tensor because the eigenvalues, $\{\lambda_1, \lambda_2, \lambda_3\}$, are not constrained to be positive. It should be clear by now that a simple reparametrization through the logarithm function can be performed on the eigenvalues to ensure that these values are positive. This process is the Log-Euler representation, discussed in the next section.

In addition, the mapping from ε to $\boldsymbol{\beta}$ is a global and many-to-one mapping because there are (infinitely) many Euler angles and therefore (infinitely) many ε that can be mapped to the same $\boldsymbol{\beta}$. This mapping can be made one-to-one by restricting the Euler angles to certain ranges, i.e., $\{0 \le \theta < \pi, -\pi \le \phi < \pi, -\pi \le \psi < \pi\}$. In this case, we cannot use the logarithm function to parametrize the Euler angles. Instead, we need a different parametrization known as the *logistic transformation* (Bates and Watts, 1988). Without loss of generality, the problem of parametrizing the Euler angles is related to constructing a map that takes a finite interval, say from $a \le \theta < b$, to an infinite interval, i.e., $-\infty \le t < \infty$. To construct such a map, the logarithm function and only two different positive terms are needed, i.e., $\theta - a$ and $b - \theta$. It is clear that the ratio of $\theta - a$ to $b - \theta$ ranges from 0 to ∞. To extend the range to $-\infty$, we use the logarithm function. Thus, we have the following parametrization:

$$t = \ln\left(\frac{\theta - a}{b - \theta}\right), \qquad [16.15]$$

or equivalently,

$$\begin{aligned} \theta &= \frac{a + b\exp(t)}{1 + \exp(t)}, \\ &= \frac{a\exp(-t) + b}{1 + \exp(-t)}, \\ &= \frac{a + a\exp(-t) + b - a}{1 + \exp(-t)}, \\ &= a + \frac{b - a}{1 + \exp(-t)}. \end{aligned} \qquad [16.16]$$

Note that the various expressions shown in Eq. [16.16] are mathematically equivalent, but the last expression is the one most suitable for numerical computation.

Having discussed how to transform the mapping from ε to $\boldsymbol{\beta}$ into a one-to-one mapping through Eq. [16.16], we should point out that its inverse mapping, which takes $\boldsymbol{\beta}$ to ε, is a local mapping because at each new point $\boldsymbol{\beta}$ a cubic equation must be solved or a symmetric matrix diagonalization must be carried out to obtain a new point ε. Nevertheless, this mapping is useful in DTI error propagation (Koay et al., 2007), but it is a one-to-many mapping unless the range of each of the Euler angles is restricted. Here, we outline a procedure for finding unique Euler angles from an orthogonal matrix $\boldsymbol{Q}$ defined in Eq. [16.11].

First, we compute $\theta = \cos^{-1}(Q_{33})$. Then, if $\theta \neq 0$, we have $\phi = \tan^{-1}(Q_{23}, Q_{13})$ and $\psi = \tan^{-1}(Q_{32}, -Q_{31})$; otherwise, we can set either ϕ or ψ to zero. Here, we use $\psi = 0$, so that we have $\phi = \tan^{-1}(-Q_{12}, Q_{12})$. The function $\tan^{-1}(y, x)$ returns the angle (or phase) whose range is from $-\pi$ to π by computing an arc

tangent of the ratio of the ordinate coordinate, y, to the abscissa coordinate, x. (In Java programming language, $\tan^{-1}(y,x)$ is given by $atan2(y,x)$).

As noted above, ϕ or ψ can be set to zero when $\theta = 0$. This is possible because, when $\theta = 0$, the rotation matrix $\boldsymbol{Q}$ is reduced to

$$\begin{pmatrix} \cos(\phi+\psi) & -\sin(\phi+\psi) & 0 \\ \sin(\phi+\psi) & \cos(\phi+\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad [16.17]$$

Thus, ϕ and ψ cannot be uniquely determined and one of them can be set to zero.

Log-Euler Representation

The Log-Euler representation is related to the matrix decomposition via a matrix logarithm. This representation is used so that the eigenvalues of the Euler representation will be positive, which occurs when these eigenvalues are reparametrized with the logarithm function. Therefore, the diffusion tensor, $\boldsymbol{D}(\tilde{\varepsilon})$, in the Log-Euler representation, is defined by

$$\boldsymbol{D}(\tilde{\varepsilon}) = \boldsymbol{Q}\tilde{\Lambda}\boldsymbol{Q}^T, \quad [16.18]$$

where $\tilde{\varepsilon} \equiv [\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3, \theta, \phi, \psi]^T$ and $\boldsymbol{Q}$ is the same orthogonal matrix defined in the previous section. Note that $\tilde{\Lambda}$ is a diagonal matrix parametrized by three eigenvalues:

$$\tilde{\Lambda} \equiv \begin{pmatrix} \exp(\tilde{\lambda}_1) & 0 & 0 \\ 0 & \exp(\tilde{\lambda}_2) & 0 \\ 0 & 0 & \exp(\tilde{\lambda}_3) \end{pmatrix}, \quad [16.19]$$

Additionally, $\{\lambda_1, \lambda_2, \lambda_3\}$ of the previous section and $\{\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3\}$ are related through $\tilde{\lambda}_1 = \ln\lambda_1$, $\tilde{\lambda}_2 = \ln\lambda_2$, and $\tilde{\lambda}_3 = \ln\lambda_3$

We have mentioned in previous work (Koay et al., 2007) that the Euler representation is not used in constrained estimation of the diffusion tensor because its computational cost per iteration is higher than that of the Cholesky representation. Specifically, the number of essential operations needed to obtain the Euler representation and the Cholesky representation from the ordinary representation are about $32n^3$ (using the method of cyclic Jacobi) and $n^3/6$, respectively, e.g., see Schwarz (1989). Nevertheless, Andersson (2007, 2008) has used this representation to carry out constrained estimation of the diffusion tensor.

As discussed in the previous section on reparametrizing the Euler angles of the Euler representation, the same technique can be applied to the Euler angles of the Log-Euler representation using Eqs. [16.15] and [16.16].

Hext Representation

The representation Hext (1963) used is related to the Euler representation but is used for quite different reasons. This representation is useful in error propagation, but not in diffusion tensor estimation, because it is generally not a diffusion tensor representation. In fact, it is a representation of a rotated diffusion tensor, Λ, expressed in terms of the original diffusion tensor $\boldsymbol{D}(\boldsymbol{\beta})$ and a rotation matrix, $\boldsymbol{Q}$, as given by

$$\Lambda = \begin{pmatrix} \lambda_1 & \lambda_4 & \lambda_6 \\ \lambda_4 & \lambda_2 & \lambda_5 \\ \lambda_6 & \lambda_5 & \lambda_3 \end{pmatrix} = \boldsymbol{Q}^T\boldsymbol{D}\boldsymbol{Q}. \quad [16.20]$$

In general, the off-diagonal elements of Λ are nonzero except when each column vector of $\boldsymbol{Q}$ is an eigenvector of $\boldsymbol{D}$; Anderson (2001) used this special case to compute the covariance between two eigenvalues.

Equation [16.20] can be rewritten in vector form as

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{pmatrix} = \boldsymbol{P} \begin{pmatrix} D_{xx} \\ D_{yy} \\ D_{zz} \\ D_{xy} \\ D_{yz} \\ D_{xz} \end{pmatrix}, \quad [16.21]$$

where the matrix, $\boldsymbol{P}$, is given by

$$\boldsymbol{P} = \begin{pmatrix} Q_{11}^2 & Q_{21}^2 & Q_{31}^2 & 2Q_{11}Q_{21} & 2Q_{21}Q_{31} & 2Q_{11}Q_{31} \\ Q_{12}^2 & Q_{22}^2 & Q_{32}^2 & 2Q_{12}Q_{22} & 2Q_{22}Q_{32} & 2Q_{12}Q_{32} \\ Q_{13}^2 & Q_{23}^2 & Q_{33}^2 & 2Q_{13}Q_{23} & 2Q_{23}Q_{33} & 2Q_{13}Q_{33} \\ Q_{11}Q_{12} & Q_{21}Q_{22} & Q_{31}Q_{32} & Q_{11}Q_{22}+Q_{21}Q_{12} & Q_{21}Q_{32}+Q_{31}Q_{22} & Q_{11}Q_{32}+Q_{31}Q_{12} \\ Q_{12}Q_{13} & Q_{22}Q_{23} & Q_{32}Q_{33} & Q_{12}Q_{23}+Q_{22}Q_{13} & Q_{22}Q_{33}+Q_{32}Q_{23} & Q_{12}Q_{33}+Q_{32}Q_{13} \\ Q_{11}Q_{13} & Q_{21}Q_{23} & Q_{31}Q_{33} & Q_{11}Q_{23}+Q_{21}Q_{13} & Q_{21}Q_{33}+Q_{31}Q_{23} & Q_{11}Q_{33}+Q_{31}Q_{13} \end{pmatrix}. \quad [16.22]$$

The inverse of $\boldsymbol{P}$, derived readily from Eq. [16.20], is

$$\boldsymbol{P}^{-1} = \begin{pmatrix} Q_{11}^2 & Q_{12}^2 & Q_{13}^2 & 2Q_{11}Q_{12} & 2Q_{12}Q_{13} & 2Q_{11}Q_{13} \\ Q_{21}^2 & Q_{22}^2 & Q_{23}^2 & 2Q_{21}Q_{22} & 2Q_{22}Q_{23} & 2Q_{21}Q_{23} \\ Q_{31}^2 & Q_{32}^2 & Q_{33}^2 & 2Q_{31}Q_{32} & 2Q_{32}Q_{33} & 2Q_{31}Q_{33} \\ Q_{11}Q_{21} & Q_{12}Q_{22} & Q_{13}Q_{22} & Q_{11}Q_{22}+Q_{12}Q_{21} & Q_{12}Q_{23}+Q_{13}Q_{22} & Q_{11}Q_{23}+Q_{13}Q_{21} \\ Q_{21}Q_{31} & Q_{22}Q_{32} & Q_{23}Q_{33} & Q_{21}Q_{32}+Q_{22}Q_{31} & Q_{22}Q_{33}+Q_{23}Q_{32} & Q_{21}Q_{33}+Q_{23}Q_{31} \\ Q_{11}Q_{31} & Q_{12}Q_{32} & Q_{13}Q_{33} & Q_{11}Q_{32}+Q_{12}Q_{31} & Q_{12}Q_{33}+Q_{13}Q_{32} & Q_{11}Q_{33}+Q_{13}Q_{31} \end{pmatrix}. \quad [16.23]$$

In general, $\boldsymbol{P}$ is not an orthogonal matrix, i.e., $\boldsymbol{P}^{-1} \neq \boldsymbol{P}^T$, except when $\boldsymbol{Q}$, and therefore $\boldsymbol{P}$, are identity matrices. However, another formulation of $\beta = [D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{yz}, D_{xz}]^T$, which is $\beta_{new} = [D_{xx}, D_{yy}, D_{zz}, \sqrt{2}D_{xy}, \sqrt{2}D_{yz}, \sqrt{2}D_{xz}]^T$, can lead to an orthogonal matrix with similar structure to $\boldsymbol{P}$ (see Koay, 2009). Such a reformulation of the ordinary representation will not affect diffusion tensor estimation but may greatly facilitate the statistical analysis of experimental designs and material symmetry (Moakher, 2006; Moakher and Norris, 2006).

Whether the mapping of $\boldsymbol{\beta}$ to $\lambda = [\lambda_1, \cdots, \lambda_6]^T$ is local or global depends on the nature of the rotation matrix, $\boldsymbol{Q}$. It is a local mapping if each column vector of $\boldsymbol{Q}$ is an eigenvector of $\boldsymbol{D}$ because matrix diagonalization will have to be carried out for each $\boldsymbol{\beta}$ or $\boldsymbol{D}$. It is a global mapping if each component of the rotation matrix, $\boldsymbol{Q}$, is a function of some set of parameters, e.g., the Euler angles. When each column vector of $\boldsymbol{Q}$ is an eigenvector of $\boldsymbol{D}$, only the eigenvalues of $\boldsymbol{D}$ are in λ. Therefore, this representation may be thought of as a partial representation of the diffusion tensor.

Locally Linear Map

The concept of a locally linear map is very useful in diffusion tensor estimation as well as error propagation. A locally linear map is related to the differential of a vector function of several variables. Thus, we define a locally linear map at $\boldsymbol{u}_0$ between two

different representations from $\boldsymbol{u}$ to $\boldsymbol{\beta}$ as the differential of $\boldsymbol{\beta}$ at $\boldsymbol{u}_0$, which can be given by

$$d\boldsymbol{\beta}(\boldsymbol{u}_0) = \boldsymbol{J}_u(\boldsymbol{\beta}(\boldsymbol{u}))\big|_{\boldsymbol{u}=\boldsymbol{u}_0} d\boldsymbol{u}, \qquad [16.24]$$

where each component of the Jacobian matrix, $\boldsymbol{J}_u(\boldsymbol{\beta})$, is expressed as $[\boldsymbol{J}_u(\boldsymbol{\beta})]_{ij} = \partial\beta_i / \partial u_j$. Based on the definition in Eq. [16.24], it is should be clear that a locally linear map can be derived from its corresponding global and local mappings if these mappings exist. Thus, the mappings among different diffusion tensor representations discussed above also have corresponding locally linear maps.

Sometimes neither global nor local mapping exists, so the Jacobian matrix cannot be readily constructed. For example, suppose we are interested in the differential of the major eigenvector of the diffusion tensor as a vector function of the components of the ordinary representation of the diffusion tensor, which is useful in understanding how uncertainty or error propagates from diffusion-weighted signals to the eigenvector, i.e.,

$$d\boldsymbol{q}_1 = \boldsymbol{J}_\beta(\boldsymbol{q}_1) d\boldsymbol{\beta}. \qquad [16.25]$$

The first step would be to compute the Jacobian matrix, $\boldsymbol{J}_\beta(\boldsymbol{q}_1)$. However, $\boldsymbol{J}_\beta(\boldsymbol{q}_1)$ depends on $\boldsymbol{q}_1(\boldsymbol{\beta})$, which cannot be readily expressed.

Instead of relying on $\boldsymbol{q}_1(\boldsymbol{\beta})$, we can try to derive Eq. [16.25] directly. Based on a simple reformulation of the matrix perturbation method (Koay et al., 2008), we have shown that the Jacobian matrix, $\boldsymbol{J}_\beta(\boldsymbol{q}_1)$, can be constructed and that the cone of uncertainty (Basser, 1997; Jones, 2003) of the major eigenvector is elliptical (Koay et al., 2008; see also Lazar and Alexander, 2003; Jeong et al., 2005; Lazar et al., 2005; Jeong and Anderson, 2008). The reformulated matrix perturbation method as described in Koay et al. (2008) is not the only way to construct the elliptical cone of uncertainty. Another approach is based on the error propagation framework (Koay et al., 2007). Interested readers should consult Koay et al. (2007, 2008) for further information on this topic, as the reformulation would take us too far afield here.

The diffusion tensor representations presented here are logically equivalent when the underlying diffusion tensor is positive definite, but they are not all equally useful or significant. In particular, the ordinary, the Cholesky, and the Log-Euler diffusion tensor representations have been used in diffusion tensor estimation; and the Cholesky, the Hext, and the Euler representations have been used in error propagation (e.g., see Anderson, 2001; Chang et al., 2007; Koay et al., 2007; or Chapter 18 by Pierpaoli in this book). We should note that the parametrization used in the Log-Euclidean formalism (Fillard et al., 2007) is exactly the same as the Log-Euler representation. In summary, it is the particular application that makes one representation preferable to another.

Least Squares Estimations of the Diffusion Tensor

In this section, we shall focus on various least squares methods for estimating the diffusion tensor. In general, there are six least squares methods of estimation: the linear least squares (LLS), the weighted linear least squares (WLLS), the nonlinear least squares (NLS) methods, and their corresponding constrained counterparts, denoted as CLLS, CWLLS, and CNLS, respectively. The positive definite constraint is used in the CLLS, CWLLS, and CNLS methods to ensure that all eigenvalues of the diffusion tensor estimate are positive.

In general, solving constrained estimation or optimization problems usually requires sophisticated nonlinear programming (Mangasarian, 1969) or constrained optimization (Nocedal and Wright, 1999) routines. Fortunately, these sophisticated routines are not needed to solve the constrained estimation problem in DTI because the positive definite constraint can be automatically satisfied through the simple reparametrizations of the diffusion tensor in the Cholesky, Log-Cholesky, or Log-Euler representations described earlier. The main advantage of these representations is that any routine of unconstrained optimization can be used to solve the constrained estimation problem (see Fig. 16.2).

Least Squares Objective Functions

The measured signal, s, in a DT-MRI experiment in a single voxel is modeled as an orientionally dependent exponential function (Stejskal and Tanner, 1965; Basser et al., 1994a, 1994b) (e.g., see Chapter 6 by Basser and Özarslan):

$$s = s_0 \exp(-b\boldsymbol{g}^T \boldsymbol{D}\boldsymbol{g}), \qquad [16.26]$$

where the measured signal, s, depends on the diffusion-encoding gradient vector (or direction), $\boldsymbol{g}$, of unit length, the diffusion weighting (or b-value), b, the reference signal, s_0, and the diffusion tensor, $\boldsymbol{D}$. The symbol T denotes the matrix or vector transposition. The diffusion tensor is assumed to be a symmetric 3×3 matrix. Note that there are only two unknown entities in Eq. (16.26)—the reference signal and the diffusion tensor—therefore, there are seven parameters in our estimation problem.

Although seven sampled signals based on at least six noncollinear gradient directions at the same nonzero b-value and at least one sampled reference signal (at zero b-value) are sufficient to produce an estimate of the diffusion tensor, it is preferable to have more sampled signals through an increase in either the number of gradient directions or the number of b-values (e.g., see Chapters 6 and 13 in this book).

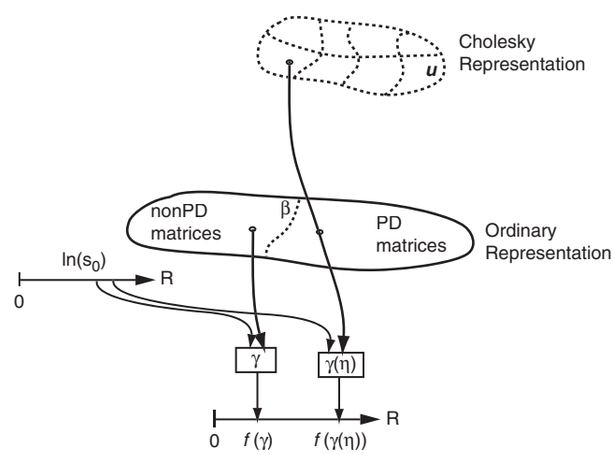

Figure 16.2 Automatic constraint satisfaction via Cholesky representation. Constrained estimation is used in DTI to get an estimate in a different parameter space, for example, the one associated with the Cholesky representation. Using this different space results in an estimated diffusion tensor that is always positive definite (PD). The function f is any least squares objective function defined in this chapter and used for estimating the diffusion tensor. R, the real axis.

To facilitate later discussion, we list all the least squares objective functions here—the LLS, WLLS, NLS, CLLS, WLLS, and CNLS objective functions are defined respectively as follows:

$$f_{LLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m}(y_i - \sum_{j=1}^{7} W_{ij}\gamma_j)^2 = \tfrac{1}{2}\sum_{i=1}^{m} F_i^2, \quad [16.27]$$

$$f_{WLLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m}\omega_i^2(y_i - \sum_{j=1}^{7} W_{ij}\gamma_j)^2 = \tfrac{1}{2}\sum_{i=1}^{m}\omega_i^2 F_i^2, \quad [16.28]$$

$$f_{NLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m}(s_i - \exp[\sum_{j=1}^{7} W_{ij}\gamma_j])^2 = \tfrac{1}{2}\sum_{i=1}^{m}(s_i - \hat{s}_i(\gamma))^2 = \tfrac{1}{2}\sum_{i=1}^{m} r_i^2, \quad [16.29]$$

$$f_{CLLS}(\eta) = f_{LLS}(\gamma(\eta)), \quad [16.30]$$

$$f_{CWLLS}(\eta) = f_{WLLS}(\gamma(\eta)), \text{ and} \quad [16.31]$$

$$f_{CNLS}(\eta) = f_{NLS}(\gamma(\eta)), \quad [16.32]$$

where $\boldsymbol{\gamma} = \left[\ln(s_0), D_{xx}, D_{yy}, D_{zz}, D_{xy}, D_{yz}, D_{xz}\right]^T$, and η is any of the following three parameter vectors depending on the diffusion tensor representation used:

$\eta_C \equiv [\ln(s_0), u_1, u_2, u_3, u_4, u_5, u_6]^T$ for the Cholesky representation, or
$\eta_{LC} \equiv [\ln(s_0), z_1, z_2, z_3, z_4, z_5, z_6]^T$ for the Log-Cholesky representation,
or
$\eta_{LE} \equiv [\ln(s_0), \tilde{\varepsilon}_1, \tilde{\varepsilon}_2, \tilde{\varepsilon}_3, \tilde{\varepsilon}_4, \tilde{\varepsilon}_5, \tilde{\varepsilon}_6]^T$ for the Log-Euler representation.

Note that the expressions for $\boldsymbol{\gamma}(\eta_C)$, $\boldsymbol{\gamma}(\eta_{LC})$, and $\boldsymbol{\gamma}(\eta_{LE})$ can be gleaned from the section on diffusion tensor representations. The symbols used above are as follows:

$\hat{s}_i(\boldsymbol{\gamma}) = \exp[\sum_{j=1}^{7} W_{ij}\gamma_j]$ is the diffusion-weighted function evaluated at $\boldsymbol{\gamma}$,

ω_i is the weight for the WLLS objective function,

$F_i = y_i - \sum_{i=1}^{7} W_{ij}\gamma_j$ is the error term for the LLS objective function,

$\omega_i F_i$ is the error term for the WLLS objective function,

$r_i(\boldsymbol{\gamma}) = s_i - \hat{s}_i(\gamma)$ is the error term for the NLS objective function, and

$y_i = \ln(s_i)$.

Finally, the design matrix is defined as

$$\boldsymbol{W} = \begin{pmatrix} 1 & -b_1 g_{1x}^2 & -b_1 g_{1y}^2 & -b_1 g_{1z}^2 & -2b_1 g_{1x} g_{1z} & -2b_1 g_{1y} g_{1z} & -2b_1 g_{1x} g_{1z} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & -b_m g_{mx}^2 & -b_m g_{my}^2 & -b_m g_{mz}^2 & -2b_m g_{mx} g_{my} & -2b_m g_{my} g_{mz} & -2b_m g_{mx} g_{mz} \end{pmatrix}. \quad [16.33]$$

For the sake of notational consistency with respect to our previous work (Koay et al., 2006b), we will use ρ to denote the parameter vector η_C so that

$$\rho \equiv [\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6, \rho_7]^T \equiv [\ln(s_0), u_1, u_2, u_3, u_4, u_5, u_6]^T. \quad [16.34]$$

Theoretical Connections between the Least Squares Methods through Their Objective Functions

Here we present a simple, general, and useful technique for establishing the connection between various least squares methods through their objective functions (Koay et al., 2006b). Without loss of generality, we will work with the unconstrained objective functions.

The basic approach of this technique is to begin with the NLS objective function and arrive at the LLS objective function via the WLLS objective function. Certain assumptions are needed to arrive at the WLLS and LLS objective functions; these assumptions are analyzed after the calculations have been completed to see if they are reasonable or at least tolerable for the problem under consideration. That is, if we use the LLS objective function without performing the deduction, we may not be aware of implicit assumptions we have made or conditions we have unknowingly imposed. We should also note that this technique can be used to derive the proper least squares weights without resorting to any error propagation technique, which are usually more complicated than this technique of deduction (e.g., see our recent application of this technique in finding the proper least squares weights in the least squares estimation of the longitudinal relaxation time [Chang et al., 2008]).

The NLS objective function in Eq. [16.29] is

$$f_{NLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m}(s_i - \exp[\sum_{j=1}^{7} W_{ij}\gamma_j])^2. \quad [16.35].$$

Based on Eq. [16.35], we will express the NLS objective function in two different but equivalent forms:

$$\begin{aligned} f_{NLS}(\boldsymbol{\gamma}) &= \tfrac{1}{2}\sum_{i=1}^{m}(s_i - \exp[\sum_{j=1}^{7} W_{ij}\gamma_j])^2, \\ &= \tfrac{1}{2}\sum_{i=1}^{m} s_i^2(1 - s_i^{-1}\exp[\sum_{j=1}^{7} W_{ij}\gamma_j])^2, \text{factoring } s_i \text{ out} \\ &= \tfrac{1}{2}\sum_{i=1}^{m} s_i^2(1 - \exp[-(\ln(s_i) - \sum_{j=1}^{7} W_{ij}\gamma_j)])^2, \\ &= \tfrac{1}{2}\sum_{i=1}^{m} s_i^2(1 - \exp[-F_i])^2 - \sum_{=1}^{7} W_{ij}\gamma. \end{aligned} \quad [16.36]$$

and

$$\begin{aligned} f_{NLS}(\boldsymbol{\gamma}) &= \tfrac{1}{2}\sum_{i=1}^{m}(s_i - \exp[\sum_{j=1}^{7} W_{ij}\gamma_j])^2 \\ &= \tfrac{1}{2}\sum_{i=1}^{m}\hat{s}_i^2(s_i\exp[-\sum_{j=1}^{7} W_{ij}\gamma_j] - 1)^2 \\ &= \tfrac{1}{2}\sum_{i=1}^{m}\hat{s}_i^2(\exp[\ln(s_i) - \sum_{j=1}^{7} W_{ij}\gamma_j] - 1)^2 \\ &= \tfrac{1}{2}\sum_{i=1}^{m}\hat{s}_i^2(\exp[+F_i] - 1)^2. \end{aligned} \quad [16.37]$$

Therefore, we have

$$f_{NLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m} s_i^2(1 - \exp[-F_i])^2 = \tfrac{1}{2}\sum_{i=1}^{m}\hat{s}_i^2(\exp[+F_i] - 1)^2. \quad [16.38]$$

Equation [16.38] is useful because it contains the error term, F_i, which belongs to both the WLLS and LLS objective functions. Once we have the WLLS or LLS error term in sight, we are one step closer to establishing the connection between the least squares objective functions mentioned above.

Equation [16.38] has a symmetry that is particularly relevant when the error term, F_i, is small. We can explore Eq. [16.38] by

assuming that $|F_i| << 1$. If we take the first-order Taylor expansion of $\exp[-F_i] \cong 1 - F_i$ and of $\exp[+F_i] \cong 1 + F_i$, it is clear that the two equations on the right-hand side of Eq. [16.38] can be reduced to $\frac{1}{2}\sum_{i=1}^{m} s_i^2 F_i^2$ and $\frac{1}{2}\sum_{i=1}^{m} \hat{s}_i^2 F_i^2$, respectively. Because these two expressions are deduced from the same expression in Eq. [16.35] with the same approximation, i.e., $|F_i| << 1$, they should be equal. It is clear that these expressions are related to the WLLS objective function. Therefore, we have

$$f_{WLLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m} s_i^2 F_i^2 = \tfrac{1}{2}\sum_{i=1}^{m} \hat{s}_i^2 F_i^2. \quad [16.39]$$

Notably, Eq. [16.39] shows that the least squares weights, s_i and $\hat{s}_i$, are equally appropriate when the error, F_i, is small; this is exactly the symmetry alluded to earlier. Thus, the observed diffusion-weighted signals may be used as the least squares weights for the WLLS method. Note that previous approaches (Basser et al., 1994a; Salvador et al., 2005) have arrived at the same least squares weights for the WLLS method, but they are less straightforward than the approach presented here.

If we assume the least squares weights, s_i, in Eq. [16.39] are approximately equal to some constant, then it should be clear that the WLLS objective function reduces to the LLS objective function when the constant is set to unity. Thus,

$$f_{LLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m} F_i^2 \quad [16.40]$$

The assumption that all of the least squares weights, s_i, in Eq. [16.39] are equal to a constant regardless of the b-value or the gradient direction on which the measured signals, s_i, depend, however, is highly restrictive and physically implausible. This assumption is not valid even when the diffusion tensor is perfectly isotropic if different b-values are used in the acquisition and the estimation.

Note that the same assumptions made here to establish the theoretical connection between the least squares objective functions will be needed in the next section when we deal with the higher-order structures, e.g., the gradient and the Hessian, of the least squares objective functions.

Theoretical Connections between the Least Squares Methods through the Higher Order Structures of Their Objective Functions

First we need to define a few terms and notations used throughout this section. The gradient vector and the Hessian matrix of a function, e.g., the NLS objective function $f_{NLS}(\boldsymbol{\gamma})$, are defined respectively by

$$\nabla f_{NLS}(\boldsymbol{\gamma}) \equiv [\partial f / \partial \gamma_1, \cdots, \partial f / \partial \gamma_7]^T, \quad [16.41]$$

and

$$\nabla^2 f_{NLS}(\boldsymbol{\gamma}) \equiv \begin{pmatrix} \frac{\partial^2 f}{\partial \gamma_1^2} & \cdots & \frac{\partial^2 f}{\partial \gamma_1 \partial \gamma_7} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \gamma_7 \partial \gamma_1} & \cdots & \frac{\partial^2 f}{\partial \gamma_7^2} \end{pmatrix}. \quad [16.42]$$

Further, the Jacobian matrix $\boldsymbol{\gamma}$ with respect to $\boldsymbol{\rho}$ is given by

$$\boldsymbol{J}_\rho(\boldsymbol{\gamma}) \equiv \begin{pmatrix} \frac{\partial \gamma_1}{\partial \rho_1} & \cdots & \frac{\partial \gamma_1}{\partial \rho_7} \\ \vdots & \ddots & \vdots \\ \frac{\partial \gamma_7}{\partial \rho_1} & \cdots & \frac{\partial \gamma_7}{\partial \rho_7} \end{pmatrix}. \quad [16.43]$$

To arrive at the explicit expressions for Eqs. [16.41–16.42], we begin with the NLS objective function again:

$$f_{NLS}(\boldsymbol{\gamma}) = \tfrac{1}{2}\sum_{i=1}^{m} (s_i - \exp[\sum_{j=1}^{7} W_{ij}\gamma_j])^2$$
$$= \tfrac{1}{2}\sum_{i=1}^{m} (s_i - \hat{s}_i(\boldsymbol{\gamma}))^2 = \tfrac{1}{2}\boldsymbol{r}^T\boldsymbol{r}, \quad [16.44]$$

where $\boldsymbol{r}$ is a vector containing all the NLS error terms, $r_i \equiv s_i - \hat{s}_i(\gamma)$. Then the gradient vector of $f_{NLS}(\boldsymbol{\gamma})$ may be expressed as the product of $\boldsymbol{J}_\gamma^T(\boldsymbol{r})$ and $\boldsymbol{r}$:

$$\nabla f_{NLS}(\boldsymbol{\gamma}) = \boldsymbol{J}_\gamma^T(\boldsymbol{r}) \cdot \boldsymbol{r} = -\boldsymbol{W}^T \hat{\boldsymbol{S}}\boldsymbol{r}, \quad [16.45]$$

and the Hessian matrix of $f_{NLS}(\boldsymbol{\gamma})$ can be given by

$$\nabla^2 f_{NLS}(\boldsymbol{\gamma}) = \boldsymbol{J}_\gamma^T(\boldsymbol{r})\boldsymbol{J}_\gamma(\boldsymbol{r}) + \sum_{i=1}^{m} r_i \nabla^2 r_i = \boldsymbol{W}^T(\hat{\boldsymbol{S}}^2 - \boldsymbol{R}\hat{\boldsymbol{S}})\boldsymbol{W}. \quad [16.46]$$

Note that $\boldsymbol{R} \equiv \boldsymbol{S} - \hat{\boldsymbol{S}}$, and both $\boldsymbol{S}$ and $\hat{\boldsymbol{S}}$ are diagonal matrices where their diagonal elements are the s_i's, and the $\hat{s}_i(\boldsymbol{\gamma})$'s, respectively.

The gradient vector and Hessian matrix of the WLLS objective function may be obtained on the basis of the assumption made in the previous section, i.e., $\boldsymbol{S} \approx \hat{\boldsymbol{S}}$, which implies that $\boldsymbol{R} \approx \boldsymbol{0}$. Under this assumption, the NLS error term, $r_i \equiv s_i - \hat{s}_i(\gamma)$, is turned into the WLLS error term, $\omega_i F_i = s_i F_i$, with $F_i = y_i - \sum_{j=1}^{7} W_{ij}\gamma_j$. Therefore, the gradient vector and Hessian matrix of the WLLS objective function are given by

$$\nabla f_{WLLS}(\boldsymbol{\gamma}) = -\boldsymbol{W}^T\boldsymbol{S}^2(\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\gamma}), \text{ and} \quad [16.47]$$

$$\nabla^2 f_{WLLS}(\boldsymbol{\gamma}) = \boldsymbol{W}^T\boldsymbol{S}^2\boldsymbol{W}. \quad [16.48]$$

Similarly, the higher-order structures of the LLS objective function may be obtained from Eqs. [16.47 and 16.48] by setting $\boldsymbol{S}$ to the identity matrix:

$$\nabla f_{LLS}(\boldsymbol{\gamma}) = -\boldsymbol{W}^T(\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\gamma}) \quad [16.49]$$

$$\nabla^2 f_{LLS}(\boldsymbol{\gamma}) = \boldsymbol{W}^T\boldsymbol{W} \quad [16.50]$$

It should be obvious how the approach presented above may also be used to derive the higher-order structures associated with the CWLLS and CLLS objective functions from the CNLS objective function. For completeness, we present here the structures associated with the CNLS objective function:

$$\nabla^2 f_{CNLS}(\boldsymbol{\rho}) = -\boldsymbol{J}_\rho^T(\boldsymbol{\gamma})\boldsymbol{W}^T(\hat{\boldsymbol{S}}^2 - \boldsymbol{R}\hat{\boldsymbol{S}})\boldsymbol{W}\boldsymbol{J}_\rho(\boldsymbol{\gamma}) + \sum_{i=1}^{m} r_i \hat{s}_i \boldsymbol{P}_i, \text{ and} \quad [16.51]$$

$$\nabla f_{CNLS}(\boldsymbol{\rho}) = -\boldsymbol{J}_\rho^T(\gamma)\boldsymbol{W}^T\hat{\boldsymbol{S}}\boldsymbol{r}, \quad [16.52]$$

where

$$\boldsymbol{J}_\rho(\boldsymbol{\gamma}) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2\rho_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\rho_3 & 0 & 2\rho_5 & 0 & 0 \\ 0 & 0 & 0 & 2\rho_4 & 0 & 2\rho_6 & 2\rho_7 \\ 0 & \rho_5 & 0 & 0 & \rho_2 & 0 & 0 \\ 0 & 0 & \rho_6 & 0 & 0 & \rho_3 & 0 \\ 0 & \rho_7 & 0 & 0 & 0 & 0 & \rho_2 \end{pmatrix}, \text{ and} \quad [16.53]$$

and

$$\boldsymbol{P}_i = (-1)\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2W_{i2} & 0 & 0 & W_{i5} & 0 & W_{i7} \\ 0 & 0 & 2W_{i3} & 0 & 0 & W_{i6} & 0 \\ 0 & 0 & 0 & 2W_{i4} & 0 & 0 & 0 \\ 0 & W_{i5} & 0 & 0 & 2W_{i3} & 0 & W_{i6} \\ 0 & 0 & W_{i6} & 0 & 0 & 2W_{i4} & 0 \\ 0 & W_{i7} & 0 & 0 & W_{i6} & 0 & 2W_{i4} \end{pmatrix}. \quad [16.54]$$

A Brief Introduction to Methods of Function Minimization

Here we introduce the unconstrained minimization of a scalar-valued function of several variables (or a vector), $f(\boldsymbol{\gamma})$. Finding the minimum point of an f usually entails a process of successive approximation. At each iteration, the iterative process begins with a current estimate, $\boldsymbol{\gamma}$, and tries to find a point within the vicinity of $\boldsymbol{\gamma}$ that has lower functional value than that of the current estimate. The value of the function, f, evaluated at a displacement vector, $\boldsymbol{\delta}$, away from $\boldsymbol{\gamma}$ can be approximated through the Taylor expansion of f around $\boldsymbol{\gamma}$, that is:

$$f(\boldsymbol{\gamma}+\boldsymbol{\delta}) \approx m(\boldsymbol{\delta}|\boldsymbol{\gamma}) + \ldots, \quad [16.55]$$

where the ellipsis indicates higher-order terms and $m(\boldsymbol{\delta} \mid \boldsymbol{\gamma})$ is given by

$$m(\boldsymbol{\delta} \mid \boldsymbol{\gamma}) = f(\boldsymbol{\gamma}) + [\nabla f(\boldsymbol{\gamma})]^T \boldsymbol{\delta} + \tfrac{1}{2}\boldsymbol{\delta}^T \nabla^2 f \boldsymbol{\delta}. \quad [16.56]$$

The iterative process in unconstrained minimization usually boils down to finding an optimal vector, $\boldsymbol{\delta}$, that minimizes $m(\boldsymbol{\delta} \mid \boldsymbol{\gamma})$ at each iteration because $m(\boldsymbol{\delta} \mid \boldsymbol{\gamma})$ may not be a good approximation to $f(\boldsymbol{\gamma} \mid \boldsymbol{\delta})$ unless the Hessian matrix of f is a constant matrix. Setting the gradient of $m(\boldsymbol{\delta} \mid \boldsymbol{\gamma})$ to zero vector, we have

$$\nabla m(\boldsymbol{\delta} \mid \boldsymbol{\gamma}) = \nabla f(\boldsymbol{\gamma}) + \nabla^2 f \boldsymbol{\delta} = \boldsymbol{0}, \text{ or} \quad [16.57]$$

$$\boldsymbol{\delta} = -[\nabla^2 f]^{-1} \nabla f(\boldsymbol{\gamma}). \quad [16.58]$$

Finding the displacement vector (or the search step vector) $\boldsymbol{\delta}$ through Eq. [16.58] is generally known as Newton's method. If the Hessian matrix in Eq. [16.58] is replaced by the identity matrix, then it is known as the steepest descent method.

Note that if $\nabla^2 f$ is positive definite, then so is $[\nabla^2 f]^{-1}$. This correspondence can be demonstrated through the eigenvalue decomposition of $\nabla^2 f$. Under this condition, $\boldsymbol{\delta}$ will always point in the direction of decreasing functional value because $[\nabla f(\boldsymbol{\gamma})]^T \boldsymbol{\delta} = -[\nabla f(\boldsymbol{\gamma})]^T [\nabla^2 f]^{-1} \nabla f(\boldsymbol{\gamma}) < 0$, i.e., the angle between $\nabla f(\boldsymbol{\gamma})$ and $\boldsymbol{\delta}$ is obtuse. We should also note that $\nabla m(\boldsymbol{\delta} \mid \boldsymbol{\gamma})$ is the gradient vector of $m(\boldsymbol{\delta} \mid \boldsymbol{\gamma})$ with respect to the parameter vector, $\boldsymbol{\delta}$.

Although the steepest-descent method requires determining whether $\nabla^2 f$ is positive definite, this method is less informed about the local landscape of f, which is encapsulated in $\nabla^2 f$. Therefore, the steepest-descent method may run into a common problem of zigzagging through a long, narrow valley (see p. 421 in Press et al., 1992).

In practice, the Hessian matrix of f evaluated at $\boldsymbol{\gamma}$ may not be positive definite, as may be the case when the parameter vector, $\boldsymbol{\gamma}$, is too far from the minimum point. If, at the end of the iterative process, $\nabla^2 f$ is still not positive definite, then the tensor model may not be appropriate, or the data may be too noisy or contain some outliers.

Several commonly used methods for finding the search step vector, $\boldsymbol{\delta}$, for a least squares objective function, e.g., the NLS objective function, can be given concisely as

$$\boldsymbol{\delta} = -\boldsymbol{H}^{-1} \nabla f_{NLS}(\boldsymbol{\gamma}), \quad [16.59]$$

where $\boldsymbol{H}$ is the generalized Hessian matrix, which takes on different expressions depending on the method used to find the search step vector:

1. Steepest-descent method: $\boldsymbol{H}_{SD} = \boldsymbol{I}$ [16.60]
2. Gauss-Newton method: $\boldsymbol{H}_{GN}(\gamma) = \boldsymbol{J}_\gamma^T(\boldsymbol{r})\boldsymbol{J}_\gamma(\boldsymbol{r})$ [16.61]
3. Levenberg method: $\boldsymbol{H}_L(\gamma,\lambda) = \boldsymbol{J}_\gamma^T(\boldsymbol{r})\boldsymbol{J}_\gamma(\boldsymbol{r}) + \lambda \boldsymbol{I}$ [16.62]
4. Marquardt method: $\boldsymbol{H}_M(\gamma,\lambda) = \boldsymbol{J}_\gamma^T(\boldsymbol{r})\boldsymbol{J}_\gamma(\boldsymbol{r}) + \lambda \boldsymbol{D}$ [16.63]
5. Newton's method: $\boldsymbol{H}_N(\gamma) = \nabla^2 f_{NLS}(\gamma)$ [16.64]
6. Modified full Newton method: $\boldsymbol{H}_{MFN}(\gamma,\lambda) = \nabla^2 f_{NLS}(\gamma) + \lambda \boldsymbol{M}$ [16.65]

In the Marquardt method, the matrix, $\boldsymbol{D}$, is a diagonal matrix with its diagonal elements equal to the diagonal elements of $\boldsymbol{J}_\gamma^T(\boldsymbol{r})\boldsymbol{J}_\gamma(\boldsymbol{r})$. In the modified full Newton method, the matrix, $\boldsymbol{M}$, can either be the identity matrix as in the Levenberg method (Levenberg, 1944) or the matrix, $\boldsymbol{D}$, as in the Marquardt method (Marquardt, 1963). The non-negative parameter, λ, which appears in three different places in the above expression, is known as the Levenberg-Marquardt parameter. All of the above methods for finding the search step vector can be extended to the CNLS objective function by replacing f_{NLS} with f_{CNLS}, and $\boldsymbol{J}_\gamma^T(\boldsymbol{r})\boldsymbol{J}_\gamma(\boldsymbol{r})$ with $\boldsymbol{J}_\rho^T(\boldsymbol{\gamma})\boldsymbol{J}_\gamma^T(\boldsymbol{r})\boldsymbol{J}_\gamma(\boldsymbol{r})\boldsymbol{J}_\rho(\boldsymbol{\gamma})$ in Eqs. [16.59, 16.61–16.65].

Before introducing the modified full Newton algorithm for finding the NLS estimate of the diffusion tensor, we will touch on the LLS and WLLS estimations with two examples.

Example 1: LLS estimation

The most straightforward approach to finding the LLS estimate is to use this equation

$$y = \boldsymbol{W}\boldsymbol{\gamma} \quad [16.66]$$

By multiplying $\boldsymbol{W}^T$ on both sides of Eq. [16.66] and inverting the resulting factor $\boldsymbol{W}^T\boldsymbol{W}$, we have

$$\hat{\gamma}_{LLS} = [\boldsymbol{W}^T\boldsymbol{W}]^{-1}\boldsymbol{W}^T \quad [16.67]$$

Another approach to finding the LLS estimate is through Eq. [16.49]. By setting the gradient of the LLS objective function to the null vector and solving for $\boldsymbol{\gamma}$, we are led to the same LLS estimate. In practice, the LLS estimate is obtained by the singular value decomposition or QR decomposition of $\boldsymbol{W}$ (e.g., see Björck, 1996).

Example 2: WLLS Estimation

Based on Eq. [16.47], we can set the gradient vector to the null vector and solve for the WLLS estimate:

$$\nabla f_{WLLS}(\boldsymbol{\gamma}) = -\boldsymbol{W}^T \boldsymbol{S}^2 (\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\gamma}) = \boldsymbol{0}, \quad [16.68]$$

$$\hat{\boldsymbol{\gamma}}_{WLLS} = (\boldsymbol{W}^T \boldsymbol{S}^2 \boldsymbol{W})^{-1} (\boldsymbol{S}\boldsymbol{W})^T \boldsymbol{S}\boldsymbol{y}, \text{ or} \quad [16.69]$$

$$\hat{\boldsymbol{\gamma}}_{WLLS} = \left(\boldsymbol{M}^T \boldsymbol{M}\right)^{-1} \boldsymbol{M}^T \boldsymbol{t}, \quad [16.70]$$

where $\boldsymbol{M} \equiv \boldsymbol{S}\boldsymbol{W}$ and $\boldsymbol{t} \equiv \boldsymbol{S}\boldsymbol{y}$. Note that Eq. [16.69] and Eq. [16.70] are equivalent but Eq. [16.70] is more convenient in practice because the same routine can be used to find the LLS and WLLS estimates.

The modified full Newton method for finding the NLS estimate of the diffusion tensor will be shown here. Later, specific modifications needed for finding CNLS estimate will be discussed. The MFN algorithm can be summarized as follows.

Initially, the WLLS estimate is used as the input parameter vector for the NLS estimation, i.e., $\gamma_0 = \gamma_{WLLS}$, and the Levenberg-Marquardt parameter is set to zero, $\lambda_0 = 0$. Let ε_1 and ε_2 be some small positive numbers, and let *flag = true*, $f_- = f_{NLS}(\boldsymbol{\gamma}_0)$, and $k_{\max}$ be the maximum number of iterations. The *flag* is needed to avoid unnecessary evaluation of the gradient vector and the Hessian matrix of the objective function. Therefore, if the gradient vector and the Hessian matrix have to be updated at some point, $\boldsymbol{\gamma}$, then the *flag* will be set to *true*, which is the case at the beginning. The algorithm returns the NLS estimate, $\hat{\boldsymbol{\gamma}}_{NLS}$.

At the k th iteration:

1. If ($k > k_{\max}$) **exit**
2. If (*flag == true*) update $\boldsymbol{H}_{MFN}(\boldsymbol{\gamma}_k, \boldsymbol{\lambda}_k)$ and $\nabla f_{NLS}(\boldsymbol{\gamma}_\kappa)$
3. Solve for $\boldsymbol{\delta}_k$ in $\boldsymbol{H}_{MFN}(\boldsymbol{\gamma}_k, \boldsymbol{\lambda}_k)\boldsymbol{\delta}_k = -\nabla f_{NLS}(\boldsymbol{\gamma}_k)$
4. $f_+ = f_{NLS}(\boldsymbol{\gamma}_k + \boldsymbol{\delta}_k)$
5. If ($|f_+ - f_-| < \varepsilon_1$ AND $0 \le -\boldsymbol{\delta}_k^T \nabla f_{NLS}(\boldsymbol{\gamma}_k) < \varepsilon_2$){

   if ($f_+ < f_-$) $\hat{\boldsymbol{\gamma}}_{NLS} = \boldsymbol{\gamma}_k + \boldsymbol{\delta}_k$ else $\hat{\boldsymbol{\gamma}}_{NLS} = \boldsymbol{\gamma}_k$

   exit

   }
6. If ($f_+ < f_-$) {

   $\lambda_{k+1} = 0.1 \times \lambda_k$

   $\gamma_{k+1} = \gamma_k + \boldsymbol{\delta}_k$

   flag = true

   $f_- = f_+$

   } else {

   if ($\lambda == 0$) $\lambda_{k+1} = 0.0001$ else $\lambda_{k+1} = 10.0 \times \lambda_\kappa$

   $\boldsymbol{\gamma}_{k+1} = \boldsymbol{\gamma}_k$

   flag = false

   }
7. Repeat the steps above.

To use the same algorithm above for the CNLS estimation, a minor modification is needed. The diffusion tensor estimate within the WLLS estimate must be positive definite before the Cholesky decomposition can be applied to map the diffusion tensor estimate to the Cholesky representation. If the diffusion tensor estimate is not positive definite, we can make it positive definite through either the modified Cholesky decomposition (Gill et al., 1981; Nocedal and Wright, 1999), which is a more efficient approach, or the eigenvalue decomposition of a symmetric matrix, which is a slightly more tortuous approach (e.g., see p. 426 of Golub and Van Loan, 1996). Specifically, the approach through the eigenvalue decomposition entails making all the eigenvalues positive, combining all the factors from the decomposition to form a new positive definite diffusion tensor, and finally decomposing the new diffusion tensor again via the Cholesy decomposition to obtain the elements of the Cholesky representation. It is clear that this approach is less efficient.

We should note that many criteria of convergence have been proposed (e.g., Moré, 1978; Bates and Watts, 1988), and those presented here in the modified full Newton method algorithm are chosen for their simplicity. Interested readers are encouraged to explore other criteria suggested in Moré (1978).

Monte Carlo Simulations

Earlier, in the section Theoretical Connections between the Least Squares Methods through Their Objective Functions, we demonstrated through a simple theoretical argument that the LLS method is the least appropriate method for estimating the diffusion tensor. Here, we use Monte Carlo simulations to provide a more tangible difference between the LLS method and other least squares methods.

First, a design matrix, **W**, based on a set of 30 gradient directions and 4 b-values (0, 500, 1000, and 1500 s/mm^2), was chosen for this comparison. Second, a simulated human brain data set as described in Koay et al. (2006b) was used for comparing the mean squared error of each least squares method in estimating the fractional anisotropy (FA) of the diffusion tensor estimate. At each voxel, a uniform level of signal-to-noise ratio (SNR) of 10 was used. Further, the signal used in determining the level of SNR was the non–diffusion-weighted signal. Finally, 2000 sets of the noisy diffusion-weighted signals were generated and each set of noisy diffusion-weighted signals was used to estimate the diffusion tensor through different least squares methods. Therefore, there were 2000 FA estimates at each voxel for each least squares method. These FA estimates were used to compute the mean squared error in estimating the FA (see Figs. 16.3 and 16.4).

16.5 Concluding Remarks

Diffusion tensor estimation is a fascinating blend of geometry, optimization, statistics, computation, and heuristics. We hope that this chapter has prepared interested readers to initiate their own exploration of this and other related areas—visualizing the landscape of a diffusion tensor objective function in different representations through the eyes of Gauss, Newton, Levenberg, and

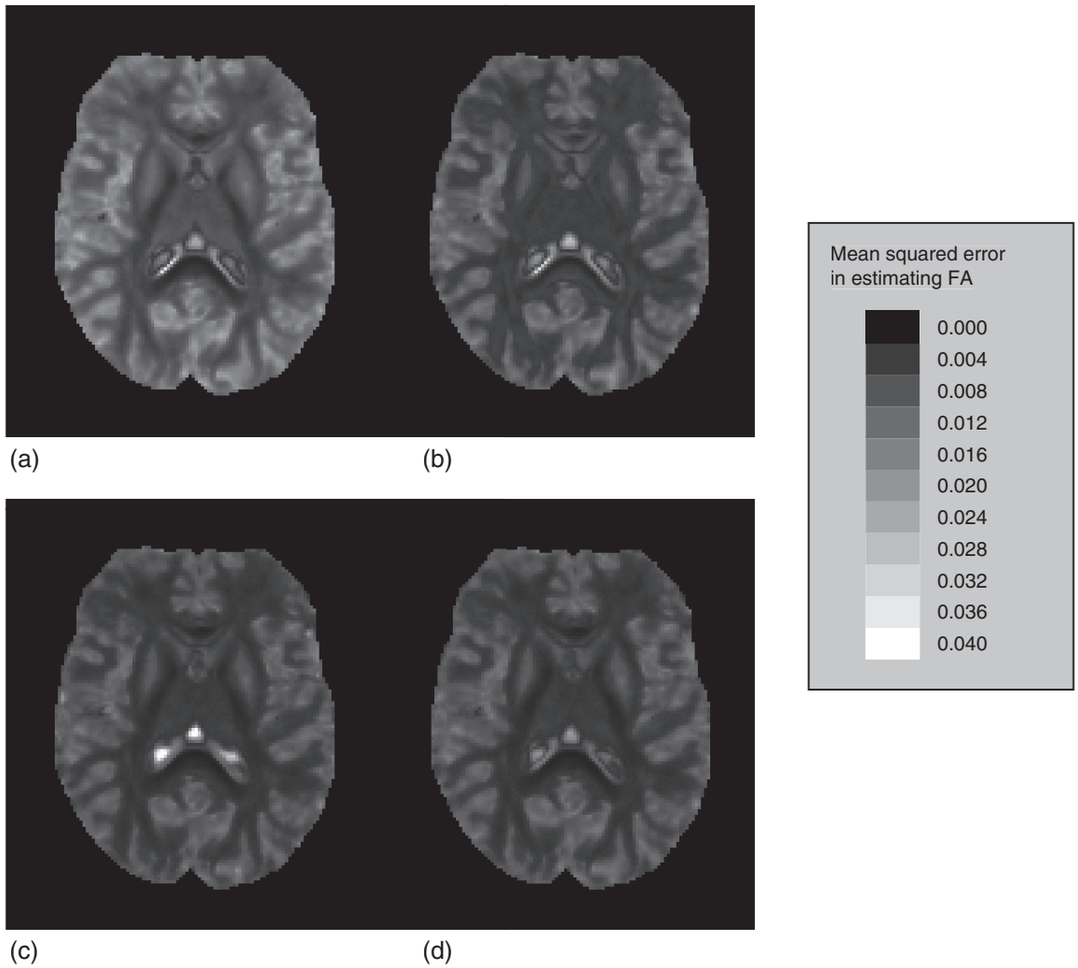

Figure 16.3 Mean squared error (MSE) of **a)** LLS, **b)** WLLS, **c)** NLS, and **d)** CNLS in estimating fractional anisotropy (FA).

Marquardt, or tracking the landscape by standing on the shoulders of these giants.

We also hope that this chapter will help readers appreciate and understand recent literature in related areas of research (e.g., Jones et al., 1999; Skare et al., 2000; Anderson, 2001, 2007, 2008; Andersson and Skare, 2002; Mangin et al., 2002; Basser and Pajevic, 2003, 2007; Jones, 2003, 2004; Pajevic and Basser, 2003; Jones and Basser, 2004; Poonawalla and Zhou, 2004; Wu et al., 2004; Chang et al., 2005, 2007; Jones and Pierpaoli, 2005; Koay et al., 2006a, 2006b, 2007, 2008; Fillard et al., 2007; Landman et al., 2007; see also Chapters 13, 16, 17, and 18 in this book). Readers should have no trouble understanding our recent works in Koay et al. (2006a, 2006b, 2007, 2008) because much of material presented here was an elaboration, extension, or refinement of these works. In two of these works, the CNLS method was found to have the lowest mean squared error in estimating various tensor-derived quantities compared with other least squares methods considered under the condition that the main source of variability is due to thermal noise. In cases where other sources of noise or contamination are more dominant than the thermal noise, it may be helpful to perform diagnostic or exploratory analysis on the data or identify sources of contamination or outliers. Other approaches that are less sensitive to outliers than the least squares methods are collectively known as the *robust* methods (e.g., Press et al., 1992, and references therein; Mangin et al., 2002; Chang et al., 2005; and Chapter 16 in this book).

Another important issue in diffusion tensor estimation involves the characterization of the variability in magnitude MR signals (Henkelman, 1985). In phased-array MRI systems (Roemer et al., 1990), each pixel contains an array of complex signals. The magnitude MR signals based on the sum-of-squares algorithm (Roemer et al., 1990) follow a non-central chi distribution (Constantinides et al., 1997; Koay and Basser, 2006), a special case of which is the Rician distribution (Rice, 1944; Bernstein et al., 1989; Gudbjartsson and Patz, 1995). Note that the magnitude MR signals obtained from a single receiver coil follow the Rician distribution. The least squares methods are not appropriate when the SNR is very low; the effect of the noise floor and its manifestation on tensor-derived quantities has been explored in detail by Jones and Basser (2004). At low-SNR regime, the accuracy of the tensor estimate can be improved by incorporating the Rician distribution in the estimation of the diffusion tensor

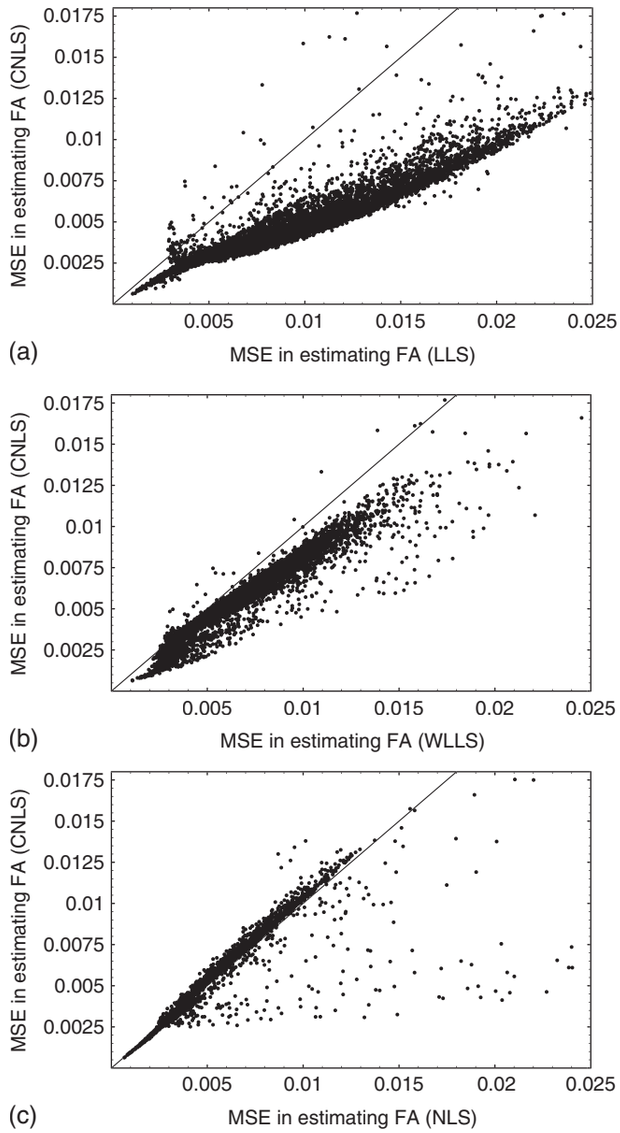

Figure 16.4 Scatter plot of the mean squared errors (MSE) of **a)** LLS, **b)** WLLS, and **c)** NLS against those of CNLS in estimating FA with data based on those of Figure 16.3. The solid lines are straight lines with unit slope.

(see Fillard et al., 2007, and Andersson, 2007, 2008). Interested readers are also encouraged to explore our recent framework for breaking the noise floor by transforming Rician- or non-central chi-distributed signals to Gaussian-distributed signals (Koay et al., 2009a, 2009b).

We should note that the diffusion tensor model may not be appropriate in some cases where there are crossing fibers in the volume element (see Chapters 27 and 28 in this book) or when the data can be modeled by a scalar diffusion (see Tuch et al., 1999, 2002, 2003; Frank, 2001, 2002; Özarslan and Mareci, 2003; Tournier et al., 2004; Tuch, 2004; Anderson, 2005; Özarslan et al., 2006; Descoteaux et al., 2007; Wu and Alexander, 2007). We close with a reminder to consider A. N. Whitehead's words, "Seek simplicity and distrust it."

Acknowledgments

This work is dedicated to my delta (Elie) and epsilon (Felix). I am profoundly grateful to my parents, Pauline Toh and Helen for everything; to Peter Basser, Carlo Pierpaoli, Beth Meyerand, and Andy Alexander for their continued support and encouragement; and to Derek K. Jones for the invitation to pen this chapter on diffusion tensor estimation. I offer many thanks to Douglas M. Bates for his insightful comments on my previous works, which have helped shaped some of the materials presented here; to Pierre Fillard for specific discussion related to his work on tensor estimation; and to Peter Basser and Liz Salak for reading and editing this article. This work was supported by the intramural research program of the Eunice Kennedy Shriver National Institute of Child Health and Human Development. The software related to the author's work is available at http://sites.google.com/site/hispeedpackets/.

References

Aksoy M, Liu C, Moseley ME, Bammer R (2008). Single-step nonlinear diffusion tensor estimation in the presence of microscopic and macroscopic motion. *Magn Reson Med* 59:1138–1150.

Anderson AW (2001). Theoretical analysis of the effects of noise on diffusion tensor imaging. *Magn Reson Med* 46:1174–1188.

Anderson AW (2005). Measurement of fiber orientation distributions using high angular resolution diffusion imaging. *Magn Reson Med* 54: 1194–1206.

Andersson JLR (2007). Maximum likelihood estimation of diffusion parameters with a Rician noise model. Paper presented at Proceedings of the International Society of Magnetic Resonance in Medicine (Berlin).

Andersson JLR (2008). Maximum a posteriori estimation of diffusion tensor parameters using a Rician noise model: why, how and but. *Neuroimage* 42:1340–1356.

Andersson JLR, Skare S (2002). A model-based method for retrospective correction of geometric distortions in diffusion-weighted EPI. *Neuroimage* 16:177–199.

Basser PJ (1997). Quantifying errors in fiber-tract direction and diffusion tensor field maps resulting from MR noise. Paper presented at Proceedings of the International Society of Magnetic Resonance in Medicine.

Basser PJ, Mattiello J, Le Bihan D (1994a). Estimation of the effective self-diffusion tensor from the NMR spin echo. *J Magn Reson B* 103: 247–254.

Basser PJ, Mattiello J, Le Bihan D. (1994b). MR diffusion tensor spectroscopy and imaging. *Biophys J* 66:259–267.

Basser PJ, Pajevic S (2003). A normal distribution for tensor-valued random variables: applications to diffusion tensor MRI. *IEEE Trans Med Imaging* 22:785–794.

Basser PJ, Pajevic S (2007). Spectral decomposition of a 4th-order covariance tensor: applications to diffusion tensor MRI. *Signal Proc* 87:220–236.

Bates DM, Watts DG (1988). *Nonlinear Regression Analysis and Its Applications*. New York: John Wiley & Sons.

Bernstein MA, Thomasson DM, Perman WH (1989). Improved detectability in low signal-to-noise ratio magnetic resonance images by means of a phase-corrected real reconstruction. *Med Phys* 16:813–817.

Björck Å (1996). *Numerical Methods for Least Squares Methods*. Philadelphia: Society for Industrial and Applied Mathematics.

Chang L-C, Jones DK, Pierpaoli C (2005). RESTORE: robust estimation of tensors by outlier rejection. *Magn Reson Med* 53:1088–1095.

Chang L-C, Koay CG, Basser PJ, Pierpaoli C (2008). Linear least-squares method for unbiased estimation of T1 from SPGR signals. *Magn Reson Med* 60:496–501.

Chang L-C, Koay CG, Pierpaoli C, Basser PJ (2007). Variance of estimated DTI-derived parameters via first-order perturbation methods. *Magn Reson Med* 57:141–149.

Constantinides CD, Atalar E, McVeigh ER (1997). Signal-to-noise measurements in magnitude images from NMR phased arrays. *Magn Reson Med* 38:852–857.

Conturo TE, Lori NF, Cull TS, Akbudak E, Snyder AZ, Shimony JS, McKinstry RC, Burton H, Raichle ME (1999). Tracking neuronal fiber pathways in the living human brain. *Proc Natl Acad Sci U S A* 96:10422–10427.

Descoteaux M, Angelino E, Fitzgibbons S, Deriche R (2007). Regularized, fast, and robust analytical Q-ball imaging. *Magn Reson Med* 58:497–510.

Fillard P, Pennec X, Arsigny V, Ayache N (2007). Clinical DT-MRI estimation, smoothing, and fiber tracking with Log-Euclidean metrics. *IEEE Trans Med Imaging* 26:1472–1482.

Frank LR (2001). Anisotropy in high angular resolution diffusion-weighted MRI. *Magn Reson Med* 45:935–939.

Frank LR (2002). Characterization of anisotropy in high angular resolution diffusion-weighted MRI. *Magn Reson Med* 47:1083–1099.
Gill P, Murray W, Wright MH (1981). *Practical Optimization*. New York: Academic Press.
Golub GH, Van Loan CF (1996). *Matrix Computation*, 3rd ed. Baltimore: Johns Hopkins University Press.
Gudbjartsson H, Patz S (1995). The Rician distribution of noisy MRI data. *Magn Reson Med* 34:910–916.
Henkelman RM (1985). Measurement of signal intensities in the presence of noise in MR images. *Med Phys* 12:232–233.
Hext GR (1963). The estimation of second-order tensors, with related tests and designs. *Biometrika* 50:353–373.
Jeong H-K, Anderson AW (2008). Characterizing fiber directional uncertainty in diffusion tensor MRI. *Magn Reson Med* 60:1408–1421.
Jeong H-K, Lu Y, Ding Z, Anderson AW (2005). Characterizing cone of uncertainty in diffusion tensor MRI. Paper presented at Proceedings of the International Society of Magnetic Resonance in Medicine.
Jones DK (2003). Determining and visualizing uncertainty in estimates of fiber orientation from diffusion tensor MRI. *Magn Reson Med* 49:7–12.
Jones DK (2004). The effect of gradient sampling schemes on measures derived from diffusion tensor MRI: a Monte Carlo study. *Magn Reson Med* 51:807–815.
Jones DK, Basser PJ (2004). Squashing peanuts and smashing pumpkins: how noise distorts diffusion-weighted MR data. *Magn Reson Med* 52:979–993.
Jones DK, Horsfield MA, Simmons A (1999). Optimal strategies for measuring diffusion in anisotropic systems by magnetic resonance imaging. *Magn Reson Med* 42:515–525.
Jones DK, Pierpaoli C (2005). Confidence mapping in diffusion tensor magnetic resonance imaging tractography using a bootstrap approach. *Magn Reson Med* 53:1143–1149.
Koay CG (2005). Advances in data analysis of diffusion tensor imaging. Doctoral dissertation, University of Wisconsin, Madison.
Koay CG (2009). On the six-dimensional orthogonal tensor representation of the rotation in three dimensions: a simplified approach. *Mech Mater* 41:951–953.
Koay CG, Basser PJ (2006). Analytically exact correction scheme for signal extraction from noisy magnitude MR signals. *J Magn Reson* 179: 317–322.
Koay CG, Carew JD, Alexander AL, Basser PJ, Meyerand ME (2006a). Investigation of anomalous estimates of tensor-derived quantities in diffusion tensor imaging. *Magn Reson Med* 55:930–936.
Koay CG, Chang L-C, Carew JD, Pierpaoli C, Basser PJ (2006b). A unifying theoretical and algorithmic framework for least squares methods of estimation in diffusion tensor imaging. *J Magn Reson* 182:115–125.
Koay CG, Chang L-C, Pierpaoli C, Basser PJ (2007). Error propagation framework for diffusion tensor imaging via diffusion tensor representations. *IEEE Trans Med Imaging* 26:1017–1034.
Koay CG, Nevo U, Chang L-C, Pierpaoli C, Basser PJ (2008). The elliptical cone of uncertainty and its normalized measures in diffusion tensor imaging. *IEEE Trans Med Imaging* 27:834–846.
Koay CG, Özarslan E, Basser PJ (2009a). A signal transformational framework for breaking the noise floor and its applications in MRI. *J Magn Reson* 197:108–119.
Koay CG, Özarslan E, Pierpaoli C (2009b). Probabilistic identification and estimation noise (PIESNO): a self-consistent approach and its applications in MRI. *J Magn Reson* 199:94–103.
Landman BA, Farrell JAD, Jones CK, Smith SA, Prince JL, Mori S (2007). Effects of diffusion weighting schemes on the reproducibility of DTI-derived fractional anisotropy, mean diffusivity, and principal eigenvector measurements at 1.5T. *Neuroimage* 36:1123–1138.
Lazar M, Alexander AL (2003). An error analysis of white matter tractography methods: synthetic diffusion tensor field simulations. *Neuroimage* 20:1140–1153.
Lazar M, Lee JH, Alexander AL (2005). Axial asymmetry of water diffusion in brain white matter. *Magn Reson Med* 54:860–867.
Levenberg K (1944). A method for the solution of certain nonlinear problems in least squares. *Q J Appl Math* 2:164–168.
Mangasarian OL (1969). *Nonlinear Programming*. New York: McGraw-Hill.
Mangin JF, Poupon C, Clark C, Le Bihan D, Bloch I (2002). Distortion correction and robust tensor estimation for MR diffusion imaging. *Med Image Anal* 6:191–198.
Marquardt DW (1963). An algorithm for the estimation of nonlinear parameters. *SIAM J Appl Math* 11:431–441.
Moakher M (2006). On the averaging of symmetric positive-definite tensors. *J Elast* 82:273–296.
Moakher M, Norris A (2006). The closest elastic tensor of arbitrary symmetry to an elasticity tensor of lower symmetry. *J Elast* 85:215–263.
Moré JJ (1978). The Levenberg-Marquardt algorithm: implementation and theory. In Watson GA (Ed.), *Lecture Notes in Mathematics, No. 630–Numerical Analysis*. Dundee: Springer-Verlag, pp. 105–116.
Mori S, Crain BJ, Chacko VP, Van Zijl PCM (1999). Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Ann Neurol* 45:265–269.
Nocedal J, Wright SJ (1999). *Numerical Optimization*. New York: Springer-Verlag.
Özarslan E, Mareci TH (2003). Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magn Reson Med* 50:955–965.
Özarslan E, Shepherd TM, Vemuri BC, Blackband SJ, Mareci TH (2006). Resolution of complex tissue microarchitecture using the diffusion orientation transform (DOT). *Neuroimage* 31:1086–1103.
Pajevic S, Basser PJ (2003). Parametric and non-parametric statistical analysis of DT-MRI data. *J Magn Reson* 161:1–16.
Pinheiro JE, Bates DM (1996). Unconstrained parametrizations for variance-covariance matrices. *Stat Comput* 6:289–296.
Poonawalla AH, Zhou XJ (2004). Analytical error propagation in diffusion anisotropy calculations. *J Magn Reson Imaging* 19:489–498.
Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1992). *Numerical Recipes in C*. New York: Cambridge University Press.
Rice SO (1944). Mathematical analysis of random noise. *Bell System Technical Journal* Vols: 23 and 24.
Roemer PB, Edelstein WA, Hayes CE, Souza SP, Mueller OM (1990). The NMR phased array. *Magn Reson Med* 16:192–225.
Salvador R, Peña A, Menon DK, Carpenter TA, Pickard JD, Bullmore ET (2005). Formal characterization and extension of the linearized diffusion tensor model. *Hum Brain Mapp* 24:144–155.
Schwarz HR (1989). *Numerical Analysis: A Comprehensive Introduction*. New York: John Wiley & Sons.
Skare S, Li T-Q, Nordell B, Ingvar M (2000). Noise considerations in the determination of diffusion tensor anisotropy. *Magn Reson Imaging* 18:659–669.
Stejskal EO, Tanner JE (1965). Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient. *J Chem Phys* 42: 288–292.
Tournier J-D, Calamante F, Gadian DG, Connelly A (2004). Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. *Neuroimage* 23:1176–1185.
Tuch DS (2004). Q-ball imaging. *Magn Reson Med* 52:1358–1372.
Tuch DS, Reese TG, Wiegell MR, Makris N, Belliveau JW, Wedeen VJ (2002). High angular resolution diffusion imaging reveals intravoxel white matter fiber heterogeneity. *Magn Reson Med* 48:577–582.
Tuch DS, Reese TG, Wiegell MR, Van JW (2003). Diffusion MRI of complex neural architecture. *Neuron* 40:885–895.
Tuch DS, Weisskoff RM, Belliveau JW, Wedeen VJ. (1999). High angular resolution diffusion imaging of the human brain. Paper presented at the 7th Annual Meeting of ISMRM, Philadelphia.
Wang Z, Vemuri BC, Chen Y, Mareci TH (2004). A constrained variational principle for direct estimation and smoothing of the diffusion tensor field from complex DWI. *IEEE Trans Med Imaging* 23:930–939.
Wu Y-C, Alexander AL (2007). Hybrid diffusion imaging. *Neuroimage* 366:17–629.
Wu Y-C, Field AS, Chung MK, Badie B, Alexander AL (2004). Quantitative analysis of diffusion tensor orientation: theoretical framework. *Magn Reson Med* 52:1146–1155.