Chapter 5

Part II

Analytical Calculation of the b Matrix in Diffusion Imaging

James Mattiello, Peter J. Basser, and Denis Le Bihan

The clinical ability to measure molecular diffusion noninvasively and *in vivo* using magnetic resonance imaging (MRI) has generated great interest (1). In tissues, the effective diffusivity depends upon tissue anatomy, microstructure, and gradient pulse parameters. Diffusion MR spectroscopy (2–4), 2DFT spin-echo imaging (5–7), and echo-planar imaging (EPI) (8–11), of *in vivo* isotropic media, to estimate the apparent diffusion constant (ADC) one must first calculate a b factor from the gradient pulse sequences (see Chapters 1 and 2). The scalar effective diffusivity also depends strongly on the direction of diffusion-sensitizing gradients with respect to the grain or fiber direction of the sample. This dependence has been observed in both diffusion NMR spectroscopy and imaging studies of skeletal muscle (12) and brain white matter (13). In these heterogeneous, macroscopically anisotropic media, diffusive transport is characterized by an effective diffusion tensor, D_{eff}, not an effective scalar diffusivity (14–18). Analogously, in diffusion tensor imaging (DTI), to estimate the diffusion tensor in each voxel, one must first calculate the nine elements of its coefficient matrix (the b matrix) from the gradient pulse sequences (19,20).

In anisotropic media, imaging and diffusion gradients appear to interact with one another, producing "cross terms" that are contained in the b matrix (21,22). If these cross-terms are unaccounted for, then the estimate of the diffusion tensor will be corrupted (see Chapter 5, Part I). In isotropic media, in principle, these cross-terms only exist between magnetic field gradient pulses applied along the same direction, however, in anisotropic media (17,18), additional cross-terms arise between magnetic field gradient pulses applied in orthogonal directions. We cannot assume *a priori* that off-diagonal elements of the effective diffusion tensor vanish (19).

Using imaging gradient parameters derived from pulse sequences, one can calculate numerically or analytically all the elements of the b matrix which contain the cross-terms between gradient pulses. Using these b matrix values, one can estimate the effective diffusion tensors and construct diffusion ellipsoid images as described by Basser et al. (19,20,23). We have previously presented data using these method to assess the quality of the DTI protocol for isotropic media (22,24). Here we present the analytical expressions to calculate the b matrix in 2DFT and EPI pulse sequences, as well as the Mathematica computer code which allows one to calculate the b matrix.

THEORY

General Expression for the b Matrix

To calculate an expression for the b matrix, we begin with the solution to the Bloch equations for a 90°–180° spin-echo pulsed-gradient NMR experiment with diffusion gradients (25), as Stejskal used it to describe free diffusion in an anisotropic medium (18). The imaging pulse sequence, $G(t)$, is defined as row vector:

$$G(t) = \{G_x(t), G_y(t), G_z(t)\} \qquad [1]$$

and functions of $G(t)$:

J. Mattiello and P. J. Basser: Biomedical Engineering and Instrumentation Program, National Center for Research Resources, National Institutes of Health, Bethesda, Maryland 20892.

D. Le Bihan: Department of Diagnostic Radiology, Warren G. Magnuson Clinical Center, National Institutes of Health, Bethesda, Maryland 20892.

$$F(t) = \int_0^t G(t')\,dt' \qquad [2]$$

and

$$f = F\left(\frac{1}{2}\mathrm{TE}\right) \qquad [3]$$

With the pulse sequence program from which all the imaging gradient parameters are derived from satisfied the condition:

$$F(\mathrm{TE}) = \int_0^{\mathrm{TE}} G(t')dt' = 0 \qquad [4]$$

The symmetric b matrix, b, as described by Basser et al. (19,20,23) is given by the formula:

$$b = \gamma^2 \int_0^{\mathrm{TE}} (F(t) - 2\xi(t)f)(F(t) - 2\xi(t)f)^{\mathrm{T}}\,dt \qquad [5]$$

Above γ is the gyromagnetic ratio, TE is the echo time, $\xi(t) = 0$ when $t < \mathrm{TE}$, $\xi(t) = 1$ when $t \geq \mathrm{TE}$.

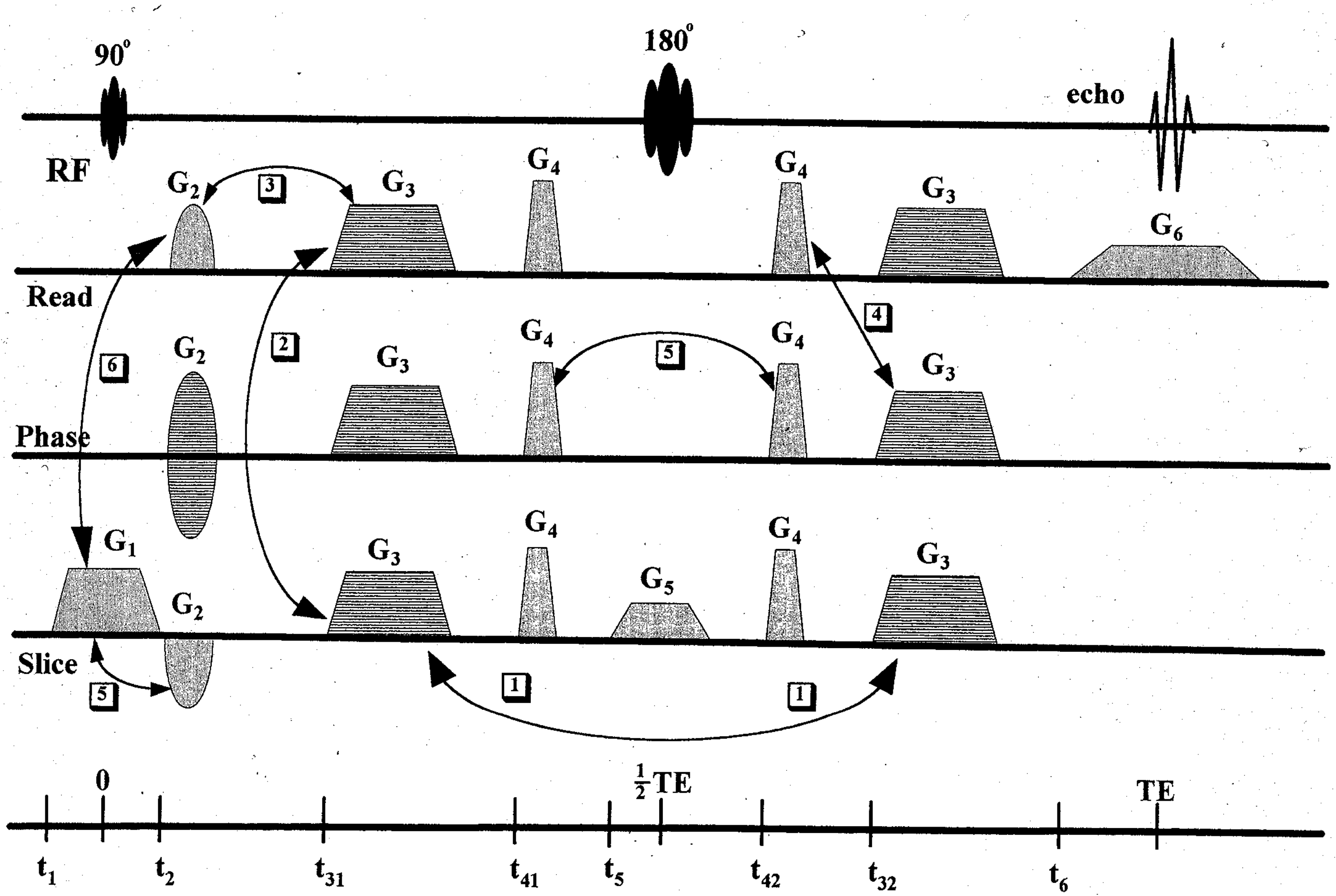

FIG. 3A: The numbered boxes (1), indicate interactions which may exist between gradients that exist in a typical 2DFT spin-echo imaging pulse sequence. They are between: 1) diffusion gradients in the same direction (i.e., read & read, phase & phase, or slice & slice); 2) diffusion gradients in different directions (i.e., read & phase, read & slice, or phase & slice); 3) diffusion gradients and imaging in the same direction (i.e., read-dephase & read diffusion gradients, or 90° slice selection gradients & slice diffusion gradients); 4) diffusion gradients and imaging in different directions (i.e., read-dephase & slice diffusion gradients, or phase crusher gradients & slice diffusion gradients); 5) imaging gradients in the same direction (i.e., read crusher gradients, 90° slice refocusing gradient & 90° slice selection gradient, or read-dephase gradient & readout gradients); and 6) imaging gradients in different directions (i.e., read crusher gradients & slice crusher gradients, read-dephase gradient & 180° slice selection gradient). Only interactions 1, 3, and 5 had been previously considered in diffusion imaging. The imaging and diffusion gradient intensities are defined as: G_1 = *(Gsl)*, a 90° slice-selection gradient; G_2 = *(Grdp, Gpe,* or *Gsrf)*, the read-dephasing, phase-encode, or slice-refocusing gradients, respectively; G_3 = *(Gdr, Gdp,* or *Gds)*, the diffusion gradients in the read, phase, and slice directions, respectively; G_4 = *(Gcr, Gcp,* or *Gcs)*, the crusher gradients in the read, phase, and slice directions, respectively; G_5 = *(Gsl)*, a 180° slice-selection gradient; G_6 = *(Gro)*, the readout gradient.

In DTI, the effective diffusion tensor is estimated from the measured spin-echo, using the formula (19,20,23):

$$ln\left(\frac{S(b)}{S(0)}\right) = -\gamma^2 \int_0^{TE} (F(t)-2\xi(t)f)\, D(F(t) - 2\xi(t)f)^T\, dt \quad [6]$$

This equation can be simplified into the expression:

$$ln\left(\frac{S(b)}{S(0)}\right) = -\sum_{i=1}^{3}\sum_{j=1}^{3} b_{ij}D_{ij} \quad [7]$$

in which b_{ij} is a component of the b matrix, b, D_{ij} is a component of the effective diffusion tensor, D, $S(0)$ is the echo intensity with no diffusion gradient applied, and $S(b)$ is the echo intensity for a particular gradient sequence.

For a particular pulse sequence, the b matrix weights the relative contribution of the various components of the diffusion tensor to the measured NMR signal. The b matrix consists of the sum of pairwise interactions between gradient pulses. Figure 3A and B show, respectively, a 2DFT spin-echo and an EPI pulse sequences with some of these pairwise interactions be-

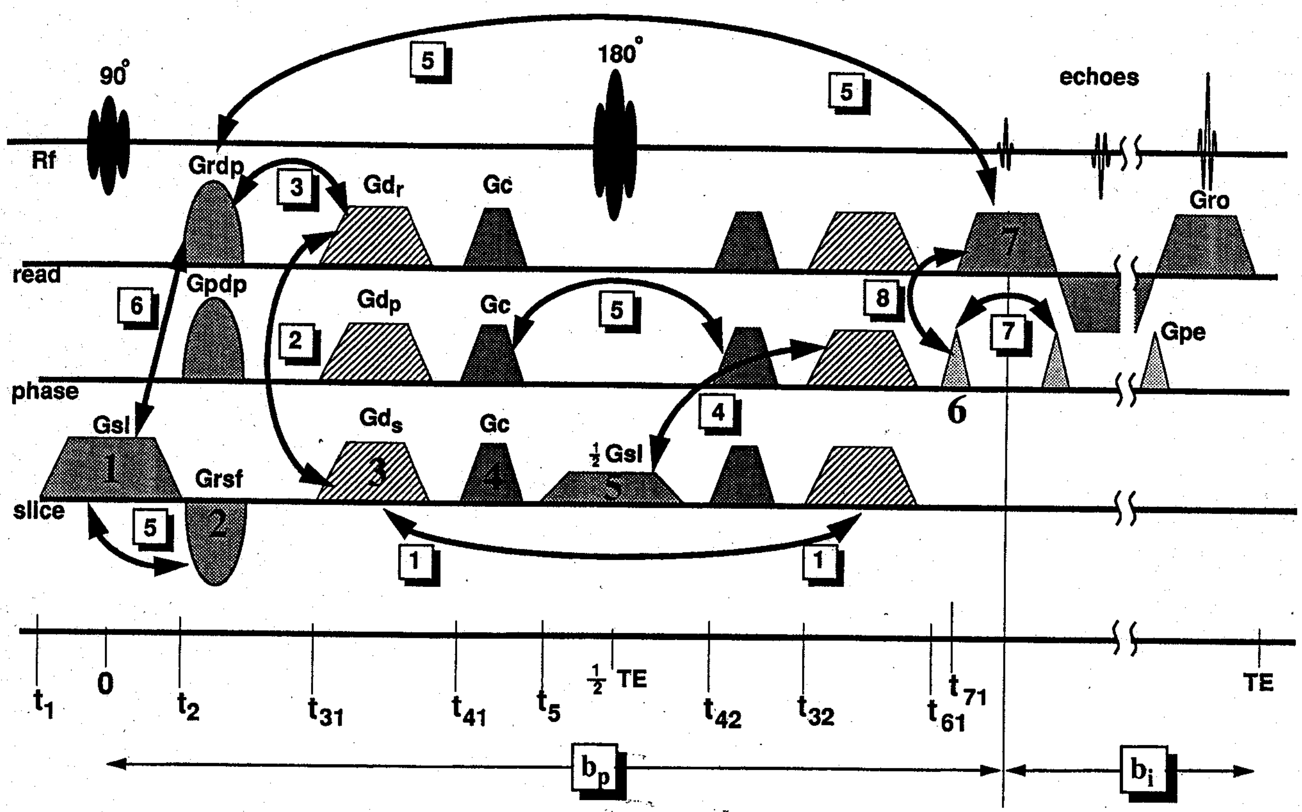

FIG. 3B: The numbered boxes (1), indicate interactions which may exist between gradients that exist in a typical EPI pulse sequence. They are between: 1) diffusion gradients in the same direction (i.e., read & read, phase & phase, or slice & slice); 2) diffusion gradients in different directions (i.e., read & phase, read & slice, or phase & slice); 3) diffusion gradients and imaging in the same direction (i.e., read-dephase & read diffusion gradients, or 90° slice selection gradients & slice diffusion gradients); 4) diffusion gradients and imaging in different directions (i.e., read-dephase & slice diffusion gradients, or phase crusher gradients & slice diffusion gradients); 5) imaging gradients in the same direction (i.e., read crusher gradients, 90° slice refocusing gradient & 90° slice selection gradient, or read-dephase gradient & read-out gradients); 6) imaging gradients in different directions (i.e., read crusher gradients & slice crusher gradients, read-dephase gradient & 180° slice selection gradient); 7) series of imaging gradients in the same direction (i.e., the phase-encode gradient pulses); and 8) series of imaging gradients in different directions (i.e., readout & phase-encode gradients). Only interactions 1, 3, and 5 had been previously considered in diffusion imaging. The imaging and diffusion gradient intensities are defined as: G_1 = *(Gsl)*, a 90° slice-selection gradient; G_2 = *(Grdp, Gpdp,* or *Gsrf)*, the read-dephasing, phase-dephasing, or slice-refocusing gradients, respectively; G_3 = *(Gdr, Gdp,* or *Gds)*, the diffusion gradients in the read, phase, and slice directions, respectively; G_4 = *(Gcr, Gcp,* or *Gcs)*, the crusher gradients in the read, phase, and slice directions, respectively; G_5 = *(Gsl)*, a 180° slice-selection gradient; G_6 = *(Gpe)*, the phase-encode gradients; and G_{7m} = *(Gro)*, the readout gradients.

tween gradients. Each term in the b matrix is the product of two gradient amplitudes (G/mm²) and a timing parameter (s³).

One approximation used is to calculate the b matrix at the center of the k-space. As the b matrix depends on all the gradient pulses "seen" by the spins at a given time t, each point of the k-space should be associated with a specific b matrix given by the expression

$$S(x, y, z) = \int \int S_0(x, y, z) \times [e^{i(xk_x+yk_y+zk_z)} e^{-b(k_x,k_y,k_z)D(x,y,z)}] dk_x dk_y dk_z \quad [8]$$

where $S_0(x, y, z)$ is the signal with T_1, T_2, and proton density contribution, and $k_i = \gamma F_i$, $(i = x, y, z)$. Considering that diffusion contrast is given by low spatial frequencies, the b matrix is usually calculated at the center of the k-space, Eq. [8] reduces to the expression;

$$S(x, y, z) = e^{-b(0,0,0)D(x,y,z)} \int \int S_0(x, y, z) \times [e^{i(xk_x+yk_y+zk_z)}] dk_x dk_y dk_z \quad [9]$$

In the case of a conventional 2DFT spin-echo pulse sequence, this means that the b matrix is calculated in the absence of phase-encoding gradients at the top of the echo in the center of the readout gradient. With EPI, this is no longer possible, since multiple phase-encode gradient pulses are dispersed throughout the sequence. Furthermore, the use of very large readout gradient pulses during the collection of the echoes may result in an incorrect estimate of the b matrix if only the center of k-space is used. For EPI we integrate up to the top of each echo for each readout gradient.

Appendix A contains the Mathematica computer code used to the calculate analytical expression for the b matrix from a given gradient pulse sequence. In cal-

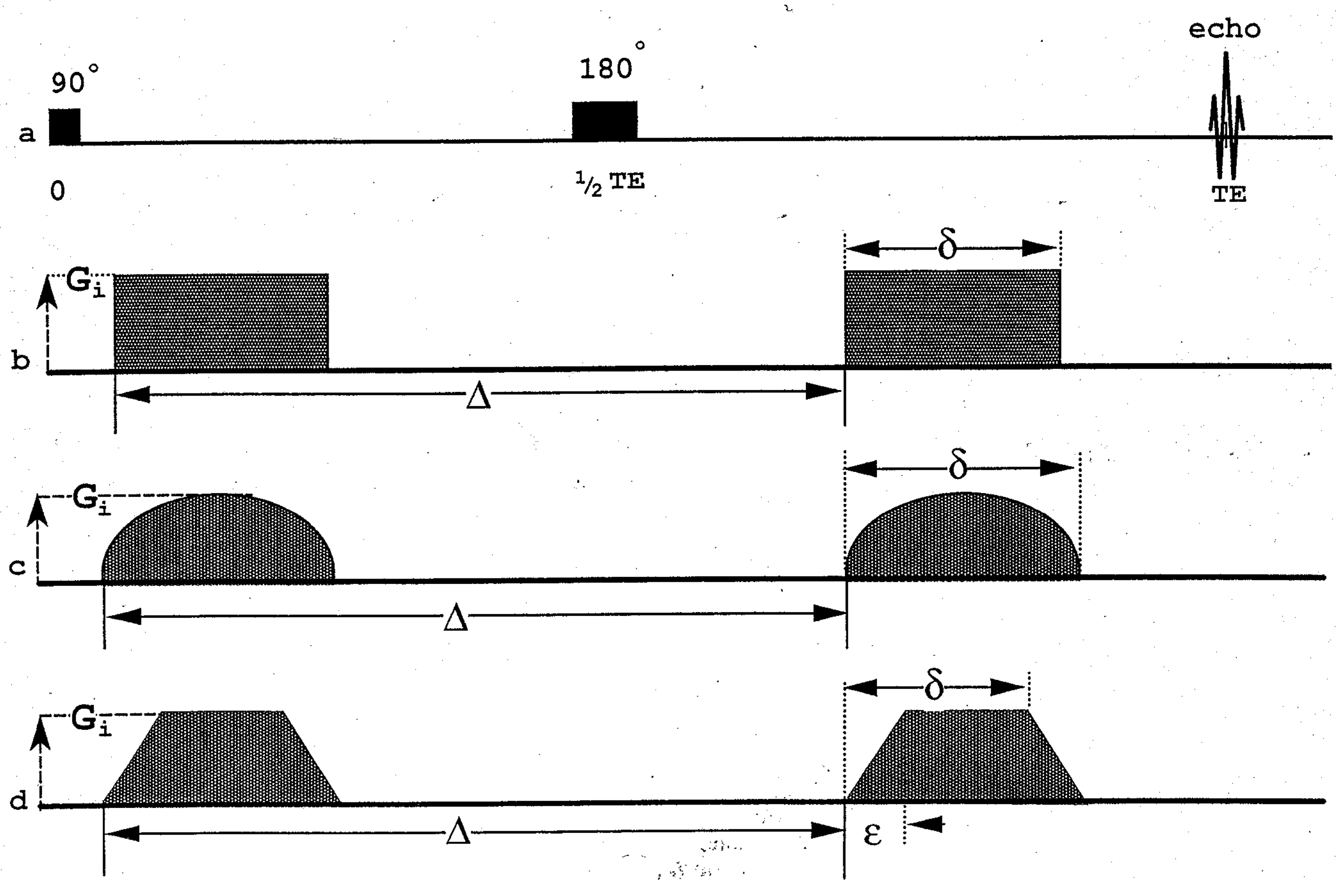

FIG. 4. Spectroscopic gradient pulse sequences for spin-echo experiment. **A:** The 90° and 180° rf pulses, and the echo signal are indicated. G_i is the diffusion gradient strength along the i^{th} coordinate direction. **B:** Rectangular diffusion gradients along one coordinate axis, the *x*-, *y*-, *z*-direction, where δ is the pulse duration and Δ is the time between the onset of the first and second gradient pulses. **C:** Sinusoidal diffusion gradients along one coordinate axis, the *x*-, *y*-, *z*-direction, where δ is the pulse duration and Δ is the time between the onset of the first and second gradient pulses. **D:** Trapezoidal diffusion gradients along one coordinate axis, the *x*-, *y*-, *z*-direction, where δ is the time between the onset of a trapezoidal pulse and the end of its plateau, ϵ is the rise time of the trapezoidal ramp, and Δ is the time between the onset of the first and second gradient pulses.

culating the analytical expression for the b matrix using Eq. [5], one could write $G(t)$ as the entire gradient pulse sequence or as two gradient pulses form the pulse sequence, for example, the read-dephase and the diffusion gradients. By doing so, the calculated b matrix expression will only show the interaction of each gradient pulse and between the two gradient pulses. In this way, one is able to examine all the possible interaction associated with all the different gradient pulses in the sequence.

General Spectroscopic b Matrix Expressions

Analytical expressions relating the echo intensity and the apparent diffusion tensor are presented for spectroscopic pulse sequences. Figure 4A show the rf pulses and the echo, G_i and G_j are the maximum field gradients along the i^{th} and j^{th} coordinate directions, respectively, then for a spin-echo sequence in the presence of a constant gradient, the b matrix is:

$$b_{ij} = \frac{1}{12}\gamma^2 G_i G_j (\mathrm{TE}^3) \qquad [10]$$

For a pair of rectangular gradient pulses, with duration δ separated by a time interval Δ as illustrated in Fig. 4B, the b matrix is:

$$b_{ij} = \gamma^2 G_i G_j \left(\delta^2\left(\Delta - \frac{1}{3}\delta\right)\right) \qquad [11]$$

With a pair of sinusoidal gradient pulses, with duration δ separated by a time interval Δ as illustrated in Fig. 4C, the b matrix is:

$$b_{ij} = \frac{4}{\pi^2}\gamma^2 G_i G_j \left(\delta^2\left(\Delta - \frac{1}{4}\delta\right)\right) \qquad [12]$$

TABLE 1. *Tau table for 2DFT spin-echo pulse sequence with trapezoidal shaped diffusion and crusher gradients*

$k =$ / $l =$	1 $\Delta_1 = \mathrm{TE}$, $\delta_1 = \mathrm{sinct} + \epsilon$,	2 $\Delta_2 = \mathrm{TE} - t_2$, $\delta_2 = \mathrm{idpt}$,	3 $\Delta_3 = t_{32} - t_{31}$, $\delta_3 = \mathrm{dift} + \epsilon$,	4 $\Delta_4 = t_{42} - t_{41}$, $\delta_4 = \mathrm{crut} + \epsilon$,	5 $\delta_5 = \delta_1$,	6 $\delta_6 = \mathrm{grot} + \epsilon$,
1	$\frac{1}{4}\left[\delta_1^2\left(\Delta_1 - \frac{1}{3}\delta_1\right) + \frac{1}{30}\epsilon^3 - \frac{1}{6}\delta_1\epsilon^2\right]$					
2	$\frac{1}{\pi}\delta_1\delta_2\left(\Delta_2 - \frac{1}{2}\delta_2\right)$	$\frac{4}{\pi 2}\delta_2^2\left(\Delta_2 - \frac{5}{8}\delta_2\right)$				
3	$\frac{1}{2}\delta_1\delta_3\Delta_3$	$\frac{2}{\pi}\delta_2\delta_3\Delta_3$	$\left[\delta_3^2\left(\Delta_3 - \frac{1}{3}\delta_3\right) + \frac{1}{30}\epsilon^3 - \frac{1}{6}\delta_3\epsilon^2\right]$			
4	$\frac{1}{2}\delta_1\delta_4\Delta_4$	$\frac{2}{\pi}\delta_2\delta_4\Delta_4$	$\delta_3\delta_4\Delta_4$	$\left[\delta_4^2\left(\Delta_4 - \frac{1}{3}\delta_4\right) + \frac{1}{30}\epsilon^2 - \frac{1}{6}\delta_4\epsilon^2\right]$		
5	$\frac{1}{8}\delta_1\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$	$\frac{1}{2\pi}\delta_2\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$	$\frac{1}{4}\delta_3\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$	$\frac{1}{4}\delta_4\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$	$\frac{1}{2}\left(\frac{1}{6}\delta_5^3 + \frac{1}{30}\epsilon^3\right)$	
6	$-\frac{1}{16}\delta_1\left(\delta_6^2 + \frac{1}{3}\epsilon^2\right)$	$-\frac{1}{4\pi}\delta_2\left(\delta_6^2 + \frac{1}{3}\epsilon^2\right)$	0	0	0	$\frac{1}{4}\left(\frac{1}{6}\delta_6^3 + \frac{1}{30}\epsilon^3\right)$

The imaging parameters used to describe trapezoidal diffusion and crusher gradients for a 2DFT pulse sequence (the functional form of the gradient pulse, e.g., its shape, duration, and magnitude) are shown in Fig. 3A. The timing parameters used in the analytical expression for the b matrix derived from a 2DFT pulse sequence, are given in this table. The abbreviations used are as follows: δ_k is the k^{th} gradient pulse duration (k, 1 through 6), i.e., either the time between the initial rise of a trapezoidal pulse and the end of its plateau, or the pulse duration for sinusoidal gradient pulse shapes; and (ϵ) is the rise time for the trapezoidal ramp for preparation gradients. The times at which the gradient pulses turn on are given by t_i: $-t_2$ for G_1; t_2 for G_2; t_{31} for the first diffusion gradient G_3; t_{41} for the first crusher gradient G_4; t_5 for G_5; t_{32} for the second diffusion gradient; t_{42} for the second crusher gradient; t_6 for G_6 readout gradient. The time between the k^{th} gradient pulses in the pulse sequence Δ_k, are given in the table. Note that $\tau_{kl} = \tau_{lk}$. For sinusoidal diffusion and crusher gradients (G_3 and G_4) in the pulse sequence some of these parameters are different:

$$\tau_{13} = \frac{1}{\pi}\delta_1\delta_3\Delta_3,\ \tau_{14} = \frac{1}{\pi}\delta_1\delta_4\Delta_4,\ \tau_{23} = \frac{4}{\pi^2}\delta_2\delta_3\Delta_3,\ \tau_{24} = \frac{4}{\pi^2}\delta_2\delta_4\Delta_4,\ \tau_{33} = \frac{4}{\pi^2}\delta_3^2\left(\Delta_3 - \frac{1}{4}\delta_3\right),$$

$$\tau_{34} = \frac{4}{\pi_2}\delta_3\delta_4\Delta_4,\ \tau_{35} = \frac{1}{2\pi}\delta_3\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right),\ \tau_{44} = \frac{4}{\pi^2}\delta_4^2\left(\Delta_4 - \frac{1}{4}\delta_4\right),\ \tau_{45} = \frac{1}{2\pi}\delta_4\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right).$$

TABLE 2. *Tau table for EPI pulse sequence with*

$k =$ $l =$	1 $\Delta_1 = \text{TE}$, $\delta_1 = \text{sinct} + \epsilon$,	2 $\Delta_2 = \text{TE} - t_2$, $\delta_2 = \text{idpt}$,	3 $\Delta_3 = t_{32} - t_{31}$, $\delta_3 = \text{dift}$,
1	$\frac{1}{4}\left[\delta_1^2\left(\Delta_1 - \frac{1}{3}\delta_1\right) + \frac{1}{30}\epsilon^3 - \frac{1}{6}\delta_1\epsilon^2\right]$		
2	$\frac{1}{\pi}\delta_1\delta_2\left(\Delta_2 - \frac{1}{2}\delta_2\right)$	$\frac{4}{\pi^2}\delta_2^2\left(\Delta_2 - \frac{5}{8}\delta_2\right)$	
3	$\frac{1}{\pi}\delta_1\delta_3\Delta_3$	$\frac{4}{\pi^2}\delta_2\delta_3\Delta_3$	$\frac{4}{\pi^2}\delta_3^2\left(\Delta_3 - \frac{1}{4}\delta_3\right)$
4	$\frac{1}{\pi}\delta_1\delta_4\Delta_4$	$\frac{4}{\pi^2}\delta_2\delta_4\Delta_4$	$\frac{4}{\pi^2}\delta_3\delta_4\Delta_4$
5	$\frac{1}{8}\delta_1\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$	$\frac{1}{2\pi}\delta_2\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$	$\frac{1}{2\pi}\delta_3\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$
6_m	$-\frac{1}{2}\epsilon'\delta_1 \sum_{m=1}^{\frac{res}{2}} (\Delta_{6m} - \epsilon')$	$-\frac{2}{\pi}\epsilon'\delta_2 \sum_{m=1}^{\frac{res}{2}} (\Delta_{6m} - \epsilon')$	0
7_1	$-\frac{1}{4}\delta_1\left[\delta_7\left(\Delta_{71} - \frac{1}{4}\delta_7\right) + \frac{1}{12}\epsilon'^2 - \frac{1}{2}\delta_7\epsilon'\right]$	$-\frac{1}{\pi}\delta_2\left[\delta_7\left(\Delta_{71} - \frac{1}{4}\delta_7\right) + \frac{1}{12}\epsilon'^2 - \frac{1}{2}\delta_7\epsilon'\right]$	0

The imaging parameters used to describe trapezoidal diffusion and crusher gradients for an EPI pulse sequence (the functional form of the gradient pulse, e.g., its shape, duration, and magnitude) are shown in Fig. 3B. The timing parameters used in the analytical expression for the b matrix derived from an EPI pulse sequence, are given in this table. The abbreviations used are as follows: δ_k is the k^{th} gradient pulse duration (k, 1 through 7), i.e., either the time between the initial rise of a trapezoidal pulse and the end of its plateau, or the pulse duration for sinusoidal gradient pulse shapes; and (ϵ) is the rise time for the trapezoidal ramp for preparation gradients, (ϵ') is the rise time for the trapezoidal ramp for the readout and phase-encode gradients. The times at which the gradient pulses turn on are given by t_i: $-t_2$ for G_1; t_2 for G_2; t_{31} for the first diffusion gradient G_3; t_{41} for the first crusher gradient G_4; t_5 for G_5; t_{32} for the second diffusion gradient; t_{42} for the second crusher gradient; t_{6m} for G_{6m} the different phase readout gradients; and t_{7m} for

Finally, for a pair of symmetric trapezoidal pulses like those shown in Fig. 4D, where δ is the time between the initial rise of a trapezoidal pulse and the end of its plateau, Δ is the time between the initial rise of the first and second gradient pulses, ϵ is the rise time of the trapezoidal ramp, the b matrix is (19):

$$b_{ij} = \gamma^2 G_i G_j \left(\delta^2\left(\Delta - \frac{1}{3}\delta\right) + \frac{1}{30}\epsilon^3 - \frac{1}{6}\delta\epsilon^2\right) \quad [13]$$

These formulae reduce to the familiar one-dimensional expressions for isotropic media (26–29).

For imaging applications, the analytical expressions for the b matrix become too tedious to evaluate by hand. We have used a symbolic manipulation program, similar to that employed by Price and Kuchel (30), to derive analytical expressions for the b matrix for DTI.

In evaluating Eq. [5], we have included localization, crusher, and diffusion gradients in the pulse sequence, which are all known to affect the echo intensity (6).

General 2DFT Spin-Echo b Matrix Expression

We synthesized a generalized imaging gradient pulse sequence, $G(t)$, from a library of individual sinusoidal and trapezoidal pulses. The parameters, such as gradient pulse shape, intensity, and duration, used to calculate the analytical expression for the elements of the b matrix, are derived from a 2DFT spin-echo pulse sequence program as shown in Fig. 1A. From Eq. [5] these b matrix elements are given by:

sinusoidal shaped diffusion and crusher gradients

4	5	6_m	7_1	7_m
$\Delta_4 = t_{42} - t_{41}$, $\delta_4 = \text{crut}$,	$\delta_5 = \delta_1$,	$\Delta_{6m} = \text{TE} - t_{6m}$, $t_{6m} = t_6 + (\delta_7 + \epsilon')(m - 1)$, $m = 1, \ldots, \frac{res}{2}$.	$\Delta_{71} = \text{TE} - t_{71}$, $\delta_7 = \text{grot} + \epsilon'$,	
$\frac{4}{\pi^2}\delta_4^2\left(\Delta_4 - \frac{1}{4}\delta_4\right)$				
$\frac{1}{2\pi}\delta_4\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right)$	$\frac{1}{2}\left(\frac{1}{6}\delta_5^3 + \frac{1}{30}\epsilon^3\right)$			
0	0	$\epsilon'^2 \sum_{m=1}^{\frac{res}{2}} \left\{\left(2m - 1\right)\Delta_{6m} - \left(m\frac{67}{30} - 1\right)\epsilon'\right\}$		
0	0	$\frac{1}{4}\epsilon'\left(\delta_7\Delta_{71} - \frac{1}{60}\epsilon'^2\right)$		
0	0	$\frac{1}{4}\left[\delta_7^2\left(\Delta_{71} - \frac{1}{3}\delta_7\right) + \frac{1}{30}\epsilon'^3 - \frac{1}{2}\delta_7^2\epsilon'\right]$		
			7_{m+1}	$\left(\frac{res}{2} - 1\right)\left[\frac{1}{12}\delta_7^3 + \frac{1}{60}\epsilon'^3 + \frac{1}{4}\delta_7^2\epsilon' - \frac{1}{12}\delta_7\epsilon'^2\right]$

G_{7m} the different readout gradients. The time between the k^{th} gradient pulses in the pulse sequence Δ_k, are given in the table. Note that $\tau_{kl} = \tau_{lk}$, and res is the number of phase-encode steps (i.e., the resolution). For sinusoidal diffusion and crusher gradients (G_3 and G_4) in the pulse sequence some of these parameters are different:

$$\tau_{13} = \frac{1}{\pi}\delta_1\delta_3\Delta_3,\ \tau_{14} = \frac{1}{\pi}\delta_1\delta_4\Delta_4,\ \tau_{23} = \frac{4}{\pi^2}\delta_2\delta_3\Delta_3,\ \tau_{24} = \frac{4}{\pi^2}\delta_2\delta_4\Delta_4,\ \tau_{33} = \frac{4}{\pi^2}\delta_3^2\left(\Delta_3 - \frac{1}{4}\delta_3\right),\ \tau_{34} = \frac{4}{\pi_2}\delta_3\delta_4\Delta_4,$$

$$\tau_{35} = \frac{1}{2\pi}\delta_3\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right),\ \tau_{44} = \frac{4}{\pi^2}\delta_4^2\left(\Delta_4 - \frac{1}{4}\delta_4\right),\ \tau_{45} = \frac{1}{2\pi}\delta_4\left(\delta_5^2 + \frac{1}{3}\epsilon^2\right).$$

$$\begin{aligned} b_{ij} = \gamma^2\{&(G_{1i}G_{1j})\tau_{11} + (G_{1i}G_{2j} + G_{2i}G_{1j})\tau_{12} + (G_{1i}G_{3j} \\ &+ G_{3i}G_{1j})\tau_{13} + (G_{1i}G_{4j} + G_{4i}G_{1j})\tau_{14} + (G_{1i}G_{5j} \\ &+ G_{5i}G_{1j})\tau_{15} + (G_{1i}G_{6j} + G_{6i}G_{1j})\tau_{16} + (G_{2i}G_{2j})\tau_{22} \\ &+ (G_{2i}G_{3j} + G_{3i}G_{2j})\tau_{23} + (G_{2i}G_{4j} + G_{4i}G_{2j})\tau_{24} \\ &+ (G_{2i}G_5 + G_{5i}G_{2j})\tau_{25} + (G_{2i}G_{6j} + G_{6i}G_{2j})\tau_{26} \\ &+ (G_{3i}G_{3j})\tau_{33} + (G_{3i}G_{4j} + G_{4i}G_{3j})\tau_{34} + (G_{3i}G_{5j} \\ &+ G_{5i}G_{3j})\tau_{35} + (G_{3i}G_{6j} + G_{6i}G_{3j})\tau_{36} + (G_{4i}G_{4j})\tau_{44} \\ &+ (G_{4i}G_{5j} + G_{5i}G_{4j})\tau_{45} + (G_{4i}G_{6j} + G_{6i}G_{4j})\tau_{46} \\ &+ (G_{5i}G_{5j})\tau_{55} + (G_{5i}G_{6j} + G_{6i}G_{5j})\tau_{56} \\ &+ (G_{6i}G_{6j})\tau_{66}\} \end{aligned} \qquad [14]$$

In this expression for the b matrix, G_{kl} are the gradient intensities. The first index, k, indicates the type of gradient pulse (e.g., slice-selection, phase-encode, etc.). The second index, l, indicates the coordinate direction in which that pulse is applied (i.e., the read, phase, or slice direction). The τ_{kl}, are combinations of time intervals and pulse parameters and are represented as the tau parameters in Table 1 and Appendix B. When the imaging parameters are specified in Eq. [14], one can calculate all the elements of the b matrix.

General Echo-planar Imaging b Matrix Expression

We synthesized a generalized echo-planar imaging gradient pulse sequence, $G(t)$, from a library of individual sinusoidal and trapezoidal pulses. The method used to consider all the pulses for the EPI sequence, including all the readout and phase-encode pulses, were described in our earlier paper (24). The parameters, such as gradient pulse shape, intensity, and duration, used to derive the analytical expression for the elements of the b matrix, are derived from an EPI pulse sequence program as shown in Fig. 3B. From Eq. [5] these b matrix elements are given by:

$$\begin{aligned} b_{ij}=\gamma^2\{&(G_{1i}G_{1j})\tau_{11}+(G_{1i}G_{2j}+G_{2i}G_{1j})\tau_{12}+(G_{1i}G_{3j} \\ &+G_{3i}G_{1j})\tau_{13}+(G_{1i}G_{4j}+G_{4i}G_{1j})\tau_{14}+(G_{1i}G_{5j} \\ &+G_{5i}G_{1j})\tau_{15}+(G_{1i}G_{6mj}+G_{6mi}G_{1j})\tau_{16m} \\ &+(G_{1i}G_{71j}+G_{71i}G_{1j})\tau_{171}+(G_{2i}G_{2j})\tau_{22} \\ &+(G_{2i}G_{3j}+G_{3i}G_{2j})\tau_{23}+(G_{2i}G_{4j}+G_{4i}G_{2j})\tau_{24} \\ &+(G_{2i}G_5+G_{5i}G_{2j})\tau_{25}+(G_{2i}G_{6mj}+G_{6mi}G_{2j})\tau_{26m} \\ &+(G_{2i}G_{71j}+G_{71i}G_{2j})\tau_{271}+(G_{3i}G_{3j})\tau_{33} \\ &+(G_{3i}G_{4j}+G_{4i}G_{3j})\tau_{34}+(G_{3i}G_{5j}+G_{5i}G_{3j})\tau_{35} \\ &+(G_{3i}G_{6mj}+G_{6mi}G_{3j})\tau_{36m}+(G_{3i}G_{71j} \\ &+G_{71i}G_{1j})\tau_{371}+(G_{4i}G_{4j})\tau_{44}+(G_{4i}G_{5j} \\ &+G_{5i}G_{4j})\tau_{45}+(G_{4i}G_{6mj}+G_{6mi}G_{4j})\tau_{46m} \\ &+(G_{4i}G_{71j}+G_{71i}G_{4j})\tau_{471}+(G_{5i}G_{5j})\tau_{55}+(G_{5i}G_{6mj} \\ &+G_{6mi}G_{5j})\tau_{56m}+(G_{5i}G_{71j}+G_{71i}G_{5j})\tau_{571} \\ &+(G_{6mi}G_{6mj})\tau_{6m6m}+(G_{6i}G_{71j}+G_{71i}G_{6j})\tau_{671} \\ &+(G_{71i}G_{71j})\tau_{7171}+(G_{7mi}G_{7(m+1)j} \\ &+G_{7(m+1)i}G_{7mj})\tau_{7m7(m+1)}\} \end{aligned} \qquad [15]$$

Above, G_{kl} are the gradient intensities. The first index, k, indicates the type of gradient pulse (e.g., slice-selection, phase-encode, etc.). The second index, l, indicates the coordinate direction in which that pulse is applied (i.e., the read, phase, or slice direction), the index m indicates the gradient number, i.e., 1st, 2nd, 3rd, . . . , $r/2$th, where r is the number of phase-encode steps, given by the spatial resolution. The parameters τ_{kl}, are combinations of time intervals and pulse parameters and are represented as the tau parameters in Table 2 and Appendix C. When the imaging gradient parameters from an EPI pulse sequence are specified in Eq. [15], one can calculate the elements of the b matrix.

Expressions for b Matrix

The plane of the image to be acquired specifies the read, phase, and slice directions with respect to the x-, y-, and z-coordinate directions. By specifying the plane, we define the read, phase, and slice directions with the laboratory x-, y-, z-coordinate reference frame. For $i = j$, the gradient pulses lie in the same coordinate direction (e.g., in the read direction), while for $i \neq j$, the gradient pulses lie along different coordinate directions (e.g., the read and phase directions). Some of the gradient pulses may be set to zero (i.e., they are not used in a particular coordinate direction). For example, G_1 and G_5 the slice-selection gradients are set to zero for the read direction. In some cases it is useful to calculate the diagonal and off-diagonal elements of the b matrix separately, Eqs. [14 and 15] allow for the accurate and quick determination of these elements for either a 2DFT spin-echo imaging or an EPI pulse sequence.

In an isotropic medium, in principle, we need only consider the diagonal elements of the b matrix. In this case Eq. [7] reduces to a linear relationship between the logarithm of the echo attenuation, $ln(S(b)/S(0))$, and each diagonal component D (i.e., $-bD = (b_{xx} + b_{yy} + b_{zz})D_{\text{eff}}$) (19,20,23). When we consider diffusion in an anisotropic medium, we must replace the scalar diffusion coefficient with a diffusion tensor, and the scalar b factor by the b matrix as in Eq. [7]. The diagonal elements (D_{rr}, D_{pp}, and D_{ss}) and their respective b matrix (b_{rr}, b_{pp}, and b_{ss}) had been considered in the context of isotropic diffusion spectroscopy (18,27,28,30) and imaging (5,30,31); recently, so have the remaining six off-diagonal elements of D (D_{rp}, D_{rs}, D_{ps}, D_{pr}, D_{sr}, and D_{sp}) and their respective b matrix elements (b_{rp}, b_{rs}, b_{ps}, b_{pr}, b_{sr}, and b_{sp}) (19). There should be no confusion between the off-diagonal elements of the b matrix and "cross terms" (21) (groups of terms that may appear within an individual b matrix). Since the b matrix is symmetric (i.e., $b_{rp} = b_{pr}$, $b_{ps} = b_{sp}$, and $b_{rs} = b_{rs}$) only three of the six off-diagonal elements of the b matrix need to be determined.

Elements of the 2DFT Spin-echo b Matrix

Equation [14] allows one to calculate the diagonal and off-diagonal elements of the b matrix for a 2DFT spin-echo pulse sequence, shown in Fig. 3A, in which the diffusion gradients may be applied in any direction. The τ_{ij}, tau parameters are given in the Mathematica computer code in Appendix B, which allows one to calculate the b matrix values from a 2DFT spin-echo pulse sequence.

For the read direction, $G_1 = 0$; $G_2 = Grdp$; $G_3 = Gdr$; $G_4 = Gcr$; $G_5 = 0$; and $G_6 = Gro$, the corresponding read-read diagonal element of the b matrix, reduces to:

$$b_{rr}=\gamma^2\{Grdp^2\cdot\tau_{22}+2Gdr\cdot Grdp\cdot\tau_{23}+2Gcr\cdot Grdp\cdot\tau_{24}+2Gro\cdot Grdp\cdot\tau_{26}+Gdr^2\cdot\tau_{33}+2Gcr\cdot Gdr\cdot\tau_{34}+Gcr^2\cdot\tau_{44}+Gro^2\cdot\tau_{66}\} \quad [16]$$

In the phase direction $G_1 = 0$; $G_2 = Gpe$; $G_3 = Gdp$; $G_4 = Gcp$; $G_5 = 0$; and $G_6 = 0$, the corresponding diagonal element of the b matrix, reduces to:

$$b_{pp}=\gamma^2\{Gpe^2\cdot\tau_{22}+2Gdp\cdot Gpe\cdot\tau_{23}+2Gcp\cdot Gpe\cdot\tau_{24}+\mathrm{Gdp}^2\cdot\tau_{33}+2Gdp\cdot Gcp\cdot\tau_{34}+Gcp^2\cdot\tau_{44}\} \quad [17a]$$

Here, b_{pp} varies as Gpe is incremented. However, most b matrix calculations assume that $Gpe = 0$, corresponding to the center line of Fourier space in the phase-encode direction (i.e., at the top of the echo) as described above in Eq. [9]. With this condition, the expression further reduces to:

$$b_{pp} = \gamma^2\{Gdp^2\cdot\tau_{33}+2Gdp\cdot Gcp\cdot\tau_{34}+Gcp^2\cdot\tau_{44}\} \quad [17b]$$

In the slice direction, $G_1 = Gsl$; $G_2 = Grsf$; $G_3 = Gds$; $G_4 = Gcs$; $G_5 = Gsl$; and $G_6 = 0$, the corresponding diagonal element of the b matrix, reduces to:

$$b_{ss}=\gamma^2\left\{Gsl^2\cdot\left(\tau_{11}+\tau_{15}+\frac{1}{4}\tau_{55}\right)+2Gsl\cdot Gsrf\cdot\left(\tau_{12}+\frac{1}{2}\tau_{25}\right)+2Gsl\cdot Gds\cdot\times\left(\tau_{13}+\frac{1}{2}\tau_{35}\right)+2Gsl\cdot Gcs\cdot\left(\tau_{14}+\frac{1}{2}\tau_{45}\right)+Gsrf^2\cdot\tau_{22}+2Gds\cdot Gsrf\cdot\tau_{23}+2Gcs\cdot Gsrf\cdot\tau_{24}+Gds^2\cdot\tau_{33}+2Gds\cdot Gcs\cdot\tau_{34}+Gcs^2\cdot\tau_{44}\right\} \quad [18]$$

For the read-phase interaction, the corresponding read-phase off-diagonal element of the b matrix, reduces to:

$$b_{rp} = \gamma^2\{Gpe\cdot Grdp\cdot\tau_{22} + (Gdp\cdot Grdp + Gdr\cdot Gpe)\cdot\tau_{23}+(Gcp\cdot Grdp + Gcr\cdot Gpe)\cdot\tau_{24} + Gpe\cdot Gro\cdot\tau_{26}+Gdr\cdot Gdp\cdot\tau_{33} + (Gcp\cdot Gdr + Gcr\cdot Gdp)\cdot\tau_{34}+Gcr\cdot Gcp\cdot\tau_{44}\} \quad [19]$$

For the read-slice interaction, the corresponding read-slice off-diagonal element of the b matrix, reduces to:

$$b_{rs}=\gamma^2\left\{Gsl\cdot Grdp\cdot\left(\tau_{12}+\frac{1}{2}\tau_{25}\right)+Gsl\cdot Gdr\cdot\left(\tau_{13}+\frac{1}{2}\tau_{35}\right)+Gsl\cdot Gcr\cdot\left(\tau_{14}+\frac{1}{2}\tau_{45}\right)+Gro\cdot Gsl\cdot\tau_{16}+Grdp\cdot Gsrf\cdot\tau_{22}+(Gdr\cdot Gsrf+Gds\cdot Grdp)\cdot\tau_{23}+(Gcr\cdot Gsrf+Gcs\cdot Grdp)\cdot\tau_{24}+Gro\cdot Gsrf\cdot\tau_{26}+Gds\cdot Gdr\cdot\tau_{33}+(Gds\cdot Gcr+Gdr\cdot Gcs)\cdot\tau_{34}+Gsc\cdot Gcr\cdot\tau_{44}\right\} \quad [20]$$

For the phase-slice interaction, the corresponding phase-slice off-diagonal element of the b matrix, reduces to:

$$b_{ps}=\gamma^2\left\{Gsl\cdot Gpe\cdot\left(\tau_{12}+\frac{1}{2}\tau_{25}\right)+Gsl\cdot Gdp\cdot\left(\tau_{13}+\frac{1}{2}\tau_{35}\right)+Gsl\cdot Gcp\cdot\left(\tau_{14}+\frac{1}{2}\tau_{45}\right)+Gpe\cdot Gsrf\cdot\tau_{22}+(Gds\cdot Gpe+Gdp\cdot Gsrf)\cdot\tau_{23}+(Gcp\cdot Gsrf+Gcs\cdot Gpe)\cdot\tau_{24}+Gds\cdot Gdp\cdot\tau_{33}+(Gds\cdot Gcp+Gdp\cdot Gcs)\cdot\tau_{34}+Gcp\cdot Gcs\cdot\tau_{44}\right\} \quad [21]$$

Here, also, Gpe is usually set to zero (the center line of Fourier space). These equations allow us to calculate the elements of the b matrix for a 2DFT spin-echo imaging pulse sequence, shown in Fig. 3A, in which the diffusion gradients are applied in any of the three imaging directions. When all the values from the imaging pulse sequence are inserted into these b matrix equations, one is able to calculate all the b matrix elements and use them to estimate the effective diffusion tensor for each voxel.

Elements of the Echo-planar Imaging b Matrix

In calculating analytical expressions for the b matrix for a spin-echo EPI pulse sequence shown in Fig. 3B, we have included all gradient pulses that typically arise. The preparation period of an EPI pulse sequence is similar to a 2DFT spin-echo pulse sequence for which the interactions between these gradients have already been derived (24). The image period of an EPI consists of a series of readout and phase-encode gradient pulses. The signal generated by an EPI sequence consists of a series of echoes, each occurring at the center of its corresponding readout gradient. EPI's particularities come from the remaining interactions between the phase-encode gradient pulses and the readout gradient pulses. In the read and slice selection directions spins are totally refocused during the first readout gradient pulse, similar to the 2DFT spin-echo pulse sequence. In these directions the b matrix contribution of the preparation (b_p) and imaging (b_i) periods may be added, ($b = b_p + b_i$), illustrated in Fig. 3B. This additive contribution is not applicable to the phase-encode direction since spins remain out of focus.

Equation [15] allows one to calculate the diagonal and off-diagonal elements of the b matrix for an EPI pulse sequence, shown in Fig. 3B, in which the diffusion gradients may be applied in any direction. The τ_{ij}, tau, parameters are given in the Mathematica computer code in Appendix C, which allows one to calculate the b matrix values from an EPI pulse sequence.

For the read direction, $G_1 = 0$; $G_2 = Grdp$; $G_3 = Gdr$; $G_4 = Gcr$; $G_5 = 0$; and $G_6 = 0$; $G_7 = Gro$, the corresponding read-read diagonal element of the b matrix, reduces to:

$$b_{rr} = \gamma^2\{Grdp^2\cdot\tau_{22} + 2Gdr\cdot Grdp\cdot\tau_{23} + 2Gcr\cdot Grdp\cdot\tau_{24} + 2Gro\cdot Grdp\cdot\tau_{271} + Gdr^2\cdot\tau_{33} + 2Gcr\cdot Gdr\cdot\tau_{34} + Gcr^2\cdot\tau_{44} + Gro^2 \cdot(\tau_{7171} + \tau_{7m7(m+1)})\} \quad [22]$$

In the phase direction, $G_1 = 0$; $G_2 = Gpdp$; $G_3 = Gdp$; $G_4 = Gcp$; $G_5 = 0$; and $G_6 = Gpe$; $G_7 = 0$, the corresponding phase-phase diagonal element of the b matrix, reduces to:

$$b_{pp} = \gamma^2\{Gpdp^2\cdot\tau_{22} + 2Gdp\cdot Gpdp\cdot\tau_{23} + 2Gcp\cdot Gpdp\cdot\tau_{24} + 2Gpe\cdot Gpdp\cdot\tau_{26m} + Gdp^2\cdot\tau_{33} + 2Gdp\cdot Gcp\cdot\tau_{34} + Gcp^2\cdot\tau_{44} + Gpe^2\cdot\tau_{6m6m}\} \quad [23]$$

In the slice direction, $G_1 = Gsl$; $G_2 = Gsrf$; $G_3 = Gds$; $G_4 = Gcs$; $G_5 = Gsl$; and $G_6 = 0$; $G_7 = 0$, the corresponding slice-slice diagonal element of the b matrix, reduces to:

$$b_{ss} = \gamma^2\left\{Gsl^2\cdot\left(\tau_{11} + \tau_{15} + \frac{1}{4}\tau_{55}\right) + 2Gsl\cdot Gsrf \cdot\left(\tau_{12} + \frac{1}{2}\tau_{25}\right) + 2Gsl\cdot Gds\cdot\left(\tau_{13} + \frac{1}{2}\tau_{35}\right) + 2Gsl\cdot Gcs\cdot\left(\tau_{14} + \frac{1}{2}\tau_{45}\right) + Gsrf^2\tau_{22} + 2Gds\cdot Gsrf\cdot\tau_{23} + 2Gcs\cdot Gsrf\cdot\tau_{24} + Gds^2\cdot\tau_{33} + 2Gds\cdot Gcs\cdot\tau_{34} + Gcs^2\cdot\tau_{44}\right\} \quad [24]$$

For the read-phase interaction, the corresponding read-phase off-diagonal element of the b matrix, reduces to:

$$b_{rp} = \gamma^2\{Gpdp\cdot Grdp\cdot\tau_{22} + (Gdp\cdot Grdp + Gdr\cdot Gpdp)\cdot\tau_{23} + (Gcp\cdot Grdp + Gcr\cdot Gpdp)\cdot\tau_{24} + Grdp\cdot Gpe\cdot\tau_{26m} + Gpdp\cdot Gro\cdot\tau_{271} + Gdr\cdot Gdp\cdot\tau_{33} + (Gcp\cdot Gdr + Gcr\cdot Gdp)\cdot\tau_{34} + Gcr\cdot Gcp\cdot\tau_{44} + Gro\cdot Gpe\cdot\tau_{6m71}\} \quad [25]$$

For the read-slice interaction, the corresponding read-slice off-diagonal element of the b matrix, reduces to:

$$b_{rs} = \gamma^2\left\{Gsl\cdot Grdp\cdot\left(\tau_{12} + \frac{1}{2}\tau_{25}\right) + Gsl\cdot Gdr \cdot\left(\tau_{13} + \frac{1}{2}\tau_{35}\right) + Gsl\cdot Gcr\cdot\left(\tau_{14} + \frac{1}{2}\tau_{45}\right) + Gro\cdot Gsl\cdot\tau_{171} + Grdp\cdot Gsrf\cdot\tau_{22} + (Gdr\cdot Gsrf + Gds\cdot Grdp)\cdot\tau_{23} + (Gcr\cdot Gsrf + Gcs\cdot Grdp)\cdot\tau_{24} + Gro\cdot Gsrf\cdot\tau_{271} + Gds\cdot Gdr\cdot\tau_{33} + (Gds\cdot Gcr + Gdr\cdot Gcs)\cdot\tau_{34} + Gsc\cdot Gcr\cdot\tau_{44}\right\} \quad [26]$$

For the phase-slice interaction, the corresponding phase-slice off-diagonal element of the b matrix, reduces to:

$$b_{ps} = \gamma^2\left\{Gsl\cdot Gpdp\cdot\left(\tau_{12} + \frac{1}{2}\tau_{25}\right) + Gsl\cdot Gdp \cdot\left(\tau_{13} + \frac{1}{2}\tau_{35}\right) + Gsl\cdot Gcp\cdot\left(\tau_{14} + \frac{1}{2}\tau_{45}\right) + Gsl\cdot Gpe\cdot\tau_{16m} + Gpdp\cdot Gsrf\cdot\tau_{22} + (Gds\cdot Gpdp\right.$$

$$+ Gdp \cdot Gsrf) \cdot \tau_{23} + (Gcp \cdot Gsrf + Gcs \cdot Gpdp) \cdot \tau_{24} + Gpe \cdot Gsrf \cdot \tau_{26m} + Gds \cdot Gdp \cdot \tau_{33} + (Gds \cdot Gcp + Gdp \cdot Gcs) \cdot \tau_{34} + Gcp \cdot Gcs \cdot \tau_{44} \Big\} \quad [27]$$

When all the values from the imaging pulse sequence are inserted into these b matrix equations, one is able to calculate all the b matrix elements and use them to estimate the effective diffusion tensor for each voxel.

RESULTS

Equations [16–21 and 22–26] demonstrate how the b matrix varies as a function of the diffusion gradient strength, when the diffusion gradients, *Gdr, Gdp,* and *Gds* in (G/mm), are varied for each image, the *b* is given in (s/mm^2). In Eqs. [16–21 and 22–27] the terms which are independent of the diffusion gradients are due to the interaction of the imaging gradient only, the linear terms in the diffusion gradients are due to the interaction between the imaging gradients and the diffusion gradients, while the quadratic terms in the diffusion gradient (the "Stejskal and Tanner" terms) are due only to the interaction between the diffusion gradients. The "cross terms" in Eqs. [16–21 and 22–27] are shown in Fig. 3A and B, are those terms that do not depend on the square of the diffusion gradient, which can be significant, particularly for small diffusion gradient values.

Using the Mathematica computer code in Appendices B and C, one can calculate the b matrix values from the imaging parameters. The appropriate tau parameters and gradient intensities are for a 2DFT spin-echo imaging and EPI pulse sequences. The method described by Basser et al. (19,20,23) was used to estimate an effective diffusion tensor from diffusion-weighted images, from which diffusion tensor images were generated. Our earlier papers (22,24), demonstrated the accuracy of these analytical b matrix expressions used to estimate the diffusion tensors. Diffusion ellipsoid images were constructed from the diffusion tensor for each voxel as described by Basser et al. (19,20,23). Figures 5 and 6 show the diffusion ellipsoid images for a region of interest for a water phantom, for both a multi-sliced 2DFT and an EPI pulse sequence.

CONCLUSION

In order to achieve the largest value of the b matrix for a given pulse sequence, one should choose the orientation of the diffusion and crusher gradients in the same orientation as the other imaging gradients along that coordinate direction. That is, if the readout gradients are positive or negative, the crusher and diffusion gradients should be positive or negative, respectively. Likewise, if the 90° and 180° slice-selection and crusher gradients are positive or negative, the diffusion gradient in the slice direction should be positive or negative, respectively. Otherwise, the interactions among the imaging, crusher, and diffusion gradients would reduce the magnitude of the b value.

There should be no confusion between the off-diagonal elements of the b matrix and cross-terms, interaction between imaging and diffusion gradients. The contribution of the cross-terms in the calculation of the b matrix can be significant for small values of the diffusion gradients. Our calculations confirm that ignoring cross-terms can introduce significant errors into the estimate of the diffusion coefficient, which can be eliminated entirely by using analytical expressions or numerically evaluated b matrices (22,24). These interactions between gradients are pronounced even when the diffusion gradients are zero. Even then the components of the b matrix are significant and off-diagonal components can be as large as the diagonal components. Whether the diffusion gradients are zero or not, the components of the b matrix are significant and off-diagonal components can be as large as the diagonal components. When the diffusion gradient is zero the b matrix may still have small nonzero values. A diffusion gradient is applied in only one direction (such as the read), all diagonal and off-diagonal b matrix elements become significantly different from the zero. For the case where two diffusion gradient directions are nonzero (read and slice), these diagonal b matrix elements and all of the off-diagonal elements are significantly different from the zero b matrix values. When all diffusion gradients are nonzero, all of the b matrix elements are significantly different from the zero b matrix values. Using only a scalar b factor, which ignores these off-diagonal elements, results in an incorrect estimate of the diffusion coefficient. These off-diagonal components are needed to estimate the components of the effective diffusion tensor, and to perform diffusion tensor imaging (DTI) as described by Basser et al. (19,20,23).

These proposed analytical expressions of the b matrix permit accurate and efficient determination of the diagonal and off-diagonal elements of the effective diffusion tensor in NMR diffusion spectroscopy and imaging. These analytical expressions to calculate the b matrix can be inserted into the image file headers and the b matrix values can be calculated automatically. The imaging data from the water, showed that the diffusion coefficient in each of the three coordinate directions were similar, within allowable experimental error. Our work confirms that ignoring "cross terms" can introduce significant errors into the estimate of the diffusion coefficient. Using analytical expressions or numerically evaluated b matrix values eliminates this source of error (22,25).

Isotropic effective diffusion tensor images are demonstrated for the water phantom using these analytically calculated b matrices. Our previous work demonstrated the components of the effective diffusion tensor images for a water phantom. The diagonal components in this image have a bright uniform intensity, indicating that most of the signal is due to the diagonal components of the effective diffusion tensor. The diagonal elements of the estimated effective diffusion tensor are the same, within experimental error, and to the expected value of the diffusion coefficient for water. The off-diagonal components are at the level of the background signal, indicating their nonsignificant contribution to the effective diffusion tensor. Since the effective diffusion coefficient in each of the three main coordinate directions were similar and within allowable experimental error, the diffusion ellipsoids shown in Figs. 5 and 6 resulted in spheres. If the diagonal elements were not similar to one another, and off-diagonal components were significant, these spheres would appear more prolate and tilted off the main axis. The nonspherical ellipsoids at the edge of the multi-slice 2DFT water images are due to an artifact resulting from local gradient present at the interference between the plastic bottle and the water. These local gradients are not known and were not accounted for in the calculation of the b matrix, resulting in the nonspherical ellipsoids. Although b matrices can be calculated, calibration is still required. Control experiments, for instance in water at known temperatures should demonstrate that proper calibration of the instrument is obtained for correct estimation of the effective diffusion tensor. These control studies test whether the diagonal and off-diagonal elements of the b matrix are calculated correctly from the imaging pulse sequences, and whether the estimation of the effective diffusion is being performed properly.

These proposed analytical expressions for the diagonal and off-diagonal components of the b matrix permit

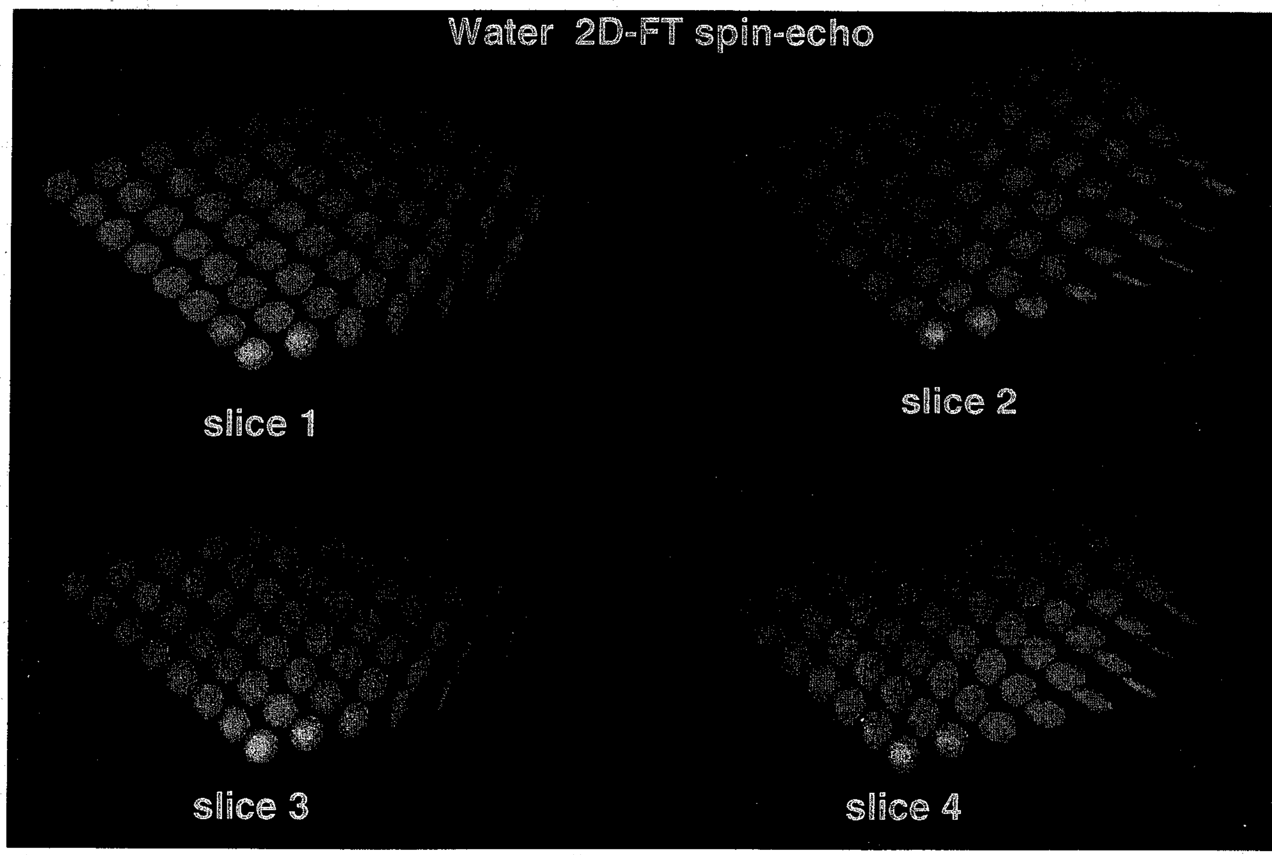

FIG. 5. An array of ellipsoids from a multi-sliced 2DFT spin-echo image, this phantom was a water-filled plastic bottle, placed perpendicular to the main magnetic field. The spherical ellipsoids are consistent with water being "isotropic," indicating that correct b matrix values were calculated. The nonspherical ellipsoids at the edge of the array is an artifact resulting from local gradient present at the interface between the plastic bottle and the water. These local gradients are not known and were not accounted for in the calculation of the b matrix, resulting in the nonspherical ellipsoids.

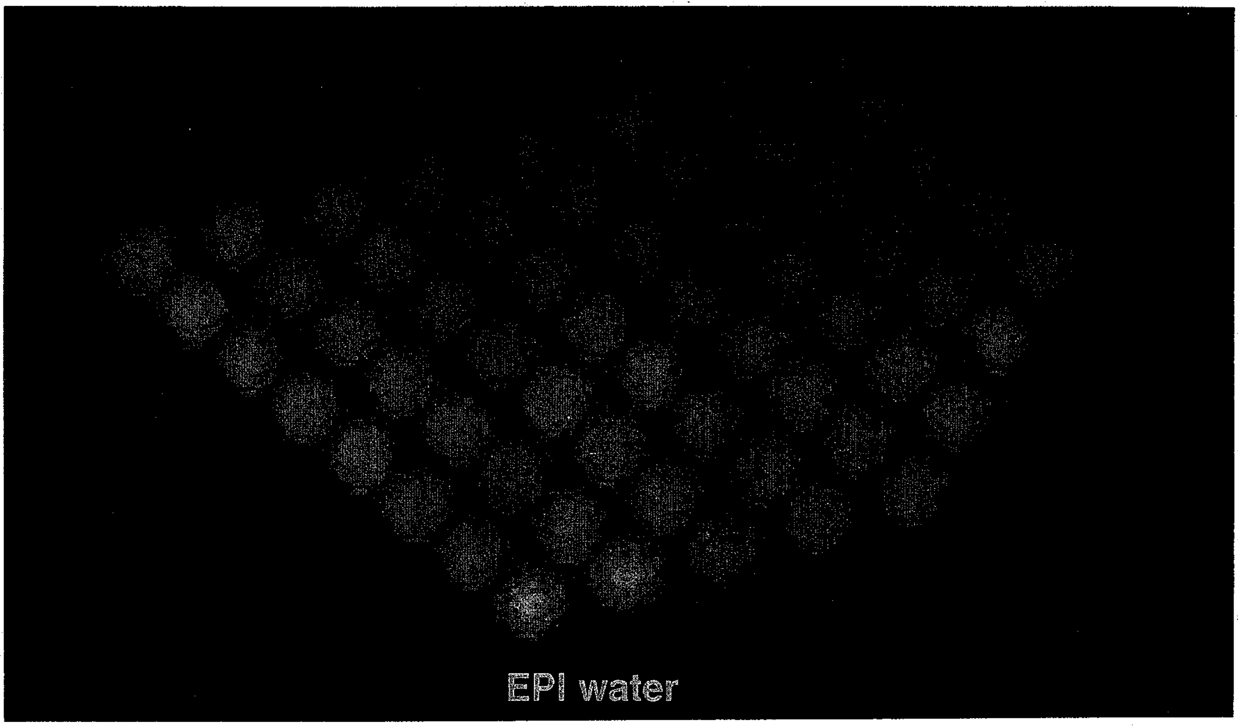

FIG. 6. An array of ellipsoids from a water-filled glass sphere, generated from an EPI pulse sequence. The spherical ellipsoids are consistent with water being "isotropic," indicating that correct b matrix values were calculated.

accurate and efficient determination of the diagonal and off-diagonal elements of the effective diffusion tensor in diffusion tensor imaging (DTI). These expressions allow one to take into account all possible interactions between gradients and to calculate the b matrix accurately, especially for small diffusion gradient strengths.

ACKNOWLEDGMENTS

This work was performed at the NIH *In Vivo* NMR Center. We thank Alan Olson, Peter Jezzard, Robert Turner, Geoff Sobering, and Scott Chesnick for their technical support. We wish to thank Barry Bowman and Brad Roth for editing of the chapter.

REFERENCES

1. Le Bihan D, Turner R, Douek P, Patronas N. Diffusion MR imaging: Clinical applications. *Am J Roentgenol* 1992;159:591.
2. Moonen CTW, von Kienlin M, van Zijl PCM, Cohen J, Gillen J, Daly P, Wolf G. Comparison of single-shot localization methods (STEAM and PRESS) for *in vivo* proton NMR spectroscopy. *NMR in Biomedicine* 1989;2:201.
3. Moonen CTW, van Zijl PCM, Le Bihan D, DesPres D. *In Vivo* NMR diffusion spectroscopy: ^{31}P application to phosphorus metabolites in muscle. *Magn Reson Med* 1990;13(3):467.
4. Posse S, Cuenod CA, Le Bihan D. *In Vivo* ^{1}H diffusion spectroscopy in human brain. In: *Abstracts of the proceedings of the eleventh annual meeting of the Society for Magnetic Resonance in Medicine, Berlin.* Berkeley: Society for Magnetic Resonance in Medicine; 1992:2141.
5. Taylor DG, Bushell MC. The spatial mapping of translational diffusion coefficients by the NMR imaging technique. *Phys Med Biol* 1985;30:345.
6. Le Bihan D, Breton E, Lallemond D, Grenier P, Cabanis E, Laval-Jeantet M. MR imaging of intravoxel incoherent motions: Application to diffusion and perfusion in neurologic disorders. *Radiology* 1986;161:401.
7. Merboldt KD, Hänicke W, Frahm J. Self-diffusion NMR imaging using stimulated echoes. *J Magn Reson* 1985;64:479.
8. Stehling MK, Turner R, Mansfield P. Echo-planar imaging: magnetic resonance imaging in a fraction of a second. *Science* 1991; 254:43.
9. Avram HE, Cooks LE. Effect of self-diffusion on echo planar imaging. In: *Abstracts of the proceedings of the seventh annual meeting of the Society for Magnetic Resonance in Medicine, San Francisco.* Berkeley: Society for Magnetic Resonance in Medicine; 1988:980.
10. Turner R, Le Bihan D. Single-shot diffusion imaging at 2.0 Tesla. *J Magn Reson* 1990;86:445.
11. Turner R, Le Bihan D, Maier J, Vavrek R, Hedges LK, Pekar J. Echo-planar imaging of intravoxel incoherent motion. *Radiology* 1990;177:407.
12. Cleveland GG, Chang DG, Hazlewood CF, Rorschach HE. Nuclear magnetic resonance measurement of skeletal muscle. *Biophys J* 1976;16:1043.
13. Moseley ME, Cohen Y, Kucharczyk J, Mintorovitch J, Asgari HS, Wendland MF, Tsuruda J, Norman D. Diffusion-weighted MR imaging of anisotropic water diffusion in cat central nervous system. *Radiology* 1990;176:439.
14. DeGroot SR, Mazur P. *Non-equilibrium thermodynamics*. New York: Dover; 1984.
15. Casimir HBG. On Onsager principles of microscopic reversibility. *Review of Modern Physics* 1945;17:343.
16. Onsager L. Reciprocal relation in irreversible process. Part I. *Physical Review* 1931;37:405.

17. Onsager L. Reciprocal relation in irreversible process. Part II. *Physical Review* 1931;38:2265.
18. Stejskal EO. Use of spin echoes in a pulsed magnetic-field gradient to study anisotropic, restricted diffusion and flow. *Journal of Chemical Physics* 1965;43:3597.
19. Basser PJ, Mattiello J, Le Bihan D. Estimation of the effective self-diffusion tensor from the NMR spin echo. *J Magn Reson* 1994;103:247–254.
20. Basser PJ, Mattiello J, Le Bihan D. MR diffusion tensor spectroscopy and imaging. *Biophys J* 1994;66:259.
21. Neeman M, Freyer JP, Sillerud IO. Pulsed-gradient spin-echo diffusion studies in NMR imaging. Effects of the imaging gradients on the determination of diffusion coefficients. *J Magn Reson* 1990;90:303.
22. Mattiello J, Basser PJ, Le Bihan D. Analytical expressions for the b matrix in MR diffusion imaging and spectroscopy. *J Magn Reson* 1994;108:131–141.
23. Basser PJ, Mattiello J, Le Bihan D. MR imaging of fiber-tract direction and diffusion in anisotropic tissues. In: *Abstracts of the proceedings of the twelfth annual meeting of the Society of Magnetic Resonance in Medicine, New York.* Berkeley: Society of Magnetic Resonance in Medicine; 1993:585.
24. Mattiello J, Basser PJ, Le Bihan D. Analytical expression for the gradient b-factor in NMR diffusion imaging. In: *Abstracts of the proceedings of the twelfth annual meeting of the Society of Magnetic Resonance in Medicine, New York.* Berkeley: Society Magnetic Resonance in Medicine; 1993:582.
25. Torrey HC. Bloch equations with diffusion terms. *Physical Review* 1956;104:563.
26. Hahn EL. Spin echoes. *Physical Review* 1950;80:580.
27. Carr HY, Purcell EM. Effects of diffusion and free precession in nuclear magnetic resonance experiments. *Physical Review* 1954; 94:930.
28. Simpson JH, Carr HY. Diffusion and nuclear spin relaxation in water. *Physical Review* 1958;111:1201.
29. Stejskal EO, Tanner JE. Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient. *Journal of Chemical Physics* 1965;42:288.
30. Price WS, Kuchel PW. Effect of nonrectangular field gradient pulses in the Stejskal and Tanner (diffusion) pulse sequence. *J Magn Reson* 1991;94:133.
31. Le Bihan D. Molecular diffusion nuclear magnetic resonance imaging. *Magn Reson Q* 1991;7:1.

Chapter 5—Appendix A

THE MATHEMATICA COMPUTER CODE USED TO DERIVE THE B MATRIX EQUATION FOR DIFFUSION AND IMAGING GRADIENT SEQUENCES

```
(* Filename = Calc_b_exp.m *)
  << ComplexExpand.m
  << DiracDelta.m

(* define wave shapes *)
halfSin[t_,amp_,tst_,tdur_] = amp * ((UnitStep[t-tst] - UnitStep[t-(tst+tdur)]) *
                               Sin[Pi*(t-tst)/tdur] )

Square[t_,amp_,tst_,tdur_] = amp * ( UnitStep[t-tst] - UnitStep[t-(tst+tdur)] )

RampU[t_,amp_,tst_,tdur_] = amp * ( (UnitStep[t-tst] - UnitStep[t-(tst+tdur)]) *
                               ((t-tst)/tdur))

RampD[t_,amp_,tst_,tdur_] = amp * ( (UnitStep[t-tst] - UnitStep[t-(tst+tdur)]) *
                               (((tst+tdur)-t)/tdur) )

Trap[t_,amp_,tst_,trt_,tdur_] = RampU[t,amp,tst,trt] +
                                    Square[t,amp,(tst + trt),tdur] +
                                    RampD[t,amp,(tst + trt + tdur),trt]

(* derived from the pulse sequence program see Fig. 3A for definition of parameters Grdp = read
dephase, Gpe = phase encode or Gpdp = phase dephase, Grsf = slice refocusing, Gdi = diffusion
in the ith direction, Gci = crusher in the ith direction, Gro = read-out, gradient pulses,
assumes trapezoidal diffusion and crusher gradient *)

(* these values are only used to evaluate the unitstep function time in usec *)
time1 = {te -> 85710.,rtr -> 200.,rt -> 500., sinct -> 1750., idpt -> 2000.,
       dift -> 10000. , difd -> 5000., crut -> 2000.0, crud -> 1000.,
       grot -> 240., res -> 64, sinctl -> 1750., sinct2 -> 1750. } ;

(* calculate some time parameters *)
time2 ={ deld -> dift + 2 rt + difd, delc -> crut + 2 rt + crud,
      tel -> (te/2. ) - ( idpt + 6 rt + sinct + dift + difd + crud + crut ) }/. time1

(* start time for pulse gradients *)
time 3 ={ t1  -> 0.,
       t2  -> sinct / 2 + rt ,
       t31 -> sinct / 2 + rt + idpt + tel ,
       t41 -> sinct / 2 + rt + idpt + tel + deld ,
       t5  -> sinct / 2 + rt + idpt + tel + deld + delc ,
       t42 -> 3 sinct / 2 + 3 rt + idpt + tel + deld + delc + crud ,
       t32 -> 3 sinct / 2 + 3 rt + idpt + tel + deld + 2 delc + difd ,
       t61 -> 3 sinct / 2 + 3 rt + idpt + tel + 2 deld + 2 delc + 1000. ,
       t71 -> 3 sinct / 2 + 3 rt + idpt + tel + 2 deld + 2 delc + 1000. + rtr,
       t72 -> 3 sinct / 2 + 3 rt + idpt + tel + 2 deld + 2 delc + 1000. + 3 rtr + grot,
       t73 -> 3 sinct / 2 + 3 rt + idpt + tel + 2 deld + 2 delc + 1000. + 5 rtr + 2 grot,
       t74 -> 3 sinct / 2 + 3 rt + idpt + tel + 2 deld + 2 delc + 1000. + 7 rtr + 3 grot, } /. time2 /.time1

(* define simple spectroscopic pulses *)
(* a pair of square diffusion gradient pulse *)
Gsq[t_] = Square[t, Gd, t31, dift ] + Square[t, Gd, t32, dift] ;

(* a pair of sinusoidal diffusion gradient pulse *)
Gsin[t_] = halfSin[t, Gd, t31, dift ] +halfSin[t, Gd, t32, dift] ;

(* a pair of trapezoidal diffusion gradient pulse *)
Gtrap[t_] =Trap[t, Gd, t31, rt, dift] +Trap[t, Gd, t32, rt, dift] ;
```

```
(* define the entire gradient pulse sequence for 2DFT spin-echo *)
Gread[t_]= halfSin[t, Grdp, t2, idpt] +
            Trap[t, Gd, t31, rt, dift] +
            Trap[t, Gc, t41, rt, crut] +
            Trap[t, Gc, t42, rt, crut] +
            Trap[t, Gd, t32, rt, dift] +
            Trap[t, Gro, t71,tr,grot] ;

Gphase[t_]= halfSin[t, Gpe, t2, idpt] +
            Trap[t, Gd, t31, rt, dift] +
            Trap[t, Gc, t41, rt, crut] +
            Trap[t, Gc, t42, rt, crut] +
            Trap[t, Gd, t32, rt, dift] ;

Gslice[t_]- Square[t, Gsl, 0, sinct/2] +
            RampD[t,Gsl,sinct/2,rt] +
            halfSin[t, Gsrf, t2, idpt] +
            Trap[t, Gd, t31, rt, dift] +
            Trap[t, Gc, t41, rt, crut] +
            Trap[t, Gs1/2, t5 , rt, sinct] +
            Trap[t, Gc, t42, rt, crut] +
            Trap[t, Gd, t32, rt, dift] ;

(* define the entire gradient pulse sequence for EPI *)

Gread[t_]= halfSin[t, Grdp, t2, idpt] +
            Trap[t, Gd, t31, rt, dift] +
            Trap[t, Gc, t41, rt, crut] +
            Trap[t, Gc, t42, rt, crut] +
            Trap[t, Gd, t32, rt, dift] +
            Trap[t, Gro, t71,rtr,grot] +
            Trap[t, -Gro, t72,rtr,grot] +
            Trap[t, Gro, t73,rtr,grot] ;

Gphase[t_]= halfSin[t, Gpdp, t2, idpt] +
            Trap[t, Gd, t31, rt, dift] +
            Trap[t, Gc, t41, rt, crut] +
            Trap[t, Gc, t42, rt, crut] +
            Trap[t, Gd, t32, rt, dift] +
            RampU[t,Gpe,t71 - rtr,rtr] +
            RampD[t,Gpe,t71 ,rtr] +
            RampU[t,Gpe,t72 - rtr,rtr] +
            RampD[t,Gpe,t72 ,rtr] +
            RampU[t,Gpe,t73 - rtr,rtr] +
            RampD[t,Gpe,t73 +rtr,rtr] ;

Gslice[t_]= Square[t, Gsl, 0, sinct/2] +
            RampD[t,Gsl,sinct/2,rt] +
            halfSin[t, Gsrf, t2, idpt] +
            Trap[t, Gd, t31, rt, dift] +
            Trap[t, Gc, t41, rt, crut] +
            RampU[t, Gs1/2, t5, rt] +
            Square[t, Gs1/2, t5 + rt, sinct] +
            RampD[t,Gs1/2, t5 + sinct + rt,rt] +
            Trap[t, Gc, t42, rt, crut] +
            Trap[t, Gd, t32, rt, dift] ;

(* or write pulses as pair of gradients *)
G12[t_]= Square[t, Gsl, 0, sinct/2] +
            RampD[t, Gsl, sinct/2, rt] +
            halfSin[t, Gpdp, t2, idpt] ;

(* diffusion and crusher are the same *)
G13[t_]= Square[t, Gs1, 0, sinct/2] +
            RampD[t, Gsl, sinct/2, rt] +
            Trap[t, Gd, t31, rt, dift] +
            Trap[t, Gd, t32, rt, dift] ;
```

```
G15[t_]= Square[t, Gsl1, 0, sinct1/2] +
         RampD[t, Gsl1, sinct1/2, rt] +
         Trap[t, Gs12/2, t5, rt, sinct2] ;

G16[t_]= Square[t, Gsl, 0, sinct/2] +
         RampD[t, Gsl, sinct/2, rt] +
         RampU[t, Gpe, t61, rtr] +
         RampD[t, Gpe, t61 + rtr, rtr] ;

G17[t_]= Square[t, Gsl, 0, sinct/2] +
         RampD[t, Gsl, sinct/2, rt] +
         Trap[t, Gro, t71, rtr, grot] ;

G23[t_]= halfSin[t, Gpdp, t2, idpt] +
         Trap[t, Gd, t31, rt, dift] +
         Trap[t, Gd, t32, rt, dift] ;

G25[t_]= halfSin[t, Gpdp, t2, idpt] +
         Trap[t, Gs12, t5, rt, sinct] ;

G26[t_]= halfSin[t, Gpdp, t2, idpt] +
         RampU[t, Gpe, t61, rtr] +
         RampD[t, Gpe, t61 + rtr, rtr] ;

G27[t_]= halfSin[t, Gpdp, t2, idpt] +
         Trap[t, Gro, t71, rtr, grot] ;

G35[t_]= Trap[t, Gd, t31, rt, dift] +
         Trap[t, Gs12, t4, rt, sinct] +
         Trap[t, Gd, t32, rt, dift] ;

G33[t_]= Trap[t, Gd, t31, rt, dift] +
         Trap[t, Gd, t32, rt, dift] ;

G36[t_]= Trap[t, Gd, t31, rt, dift] +
         Trap[t, Gd, t32, rt, dift] +
         RampU[t, Gpe, t61, rtr] +
         RampD[t, Gpe, t61 + rtr, rtr] ;

G37[t_]= Trap[t, Gd, t31, rt, dift] +
         Trap[t, Gd, t32, rt, dift] +
         Trap[t, Gro, t71, rtr, grot] ;

G56[t_]= Trap[t, Gs12, t5, rt, sinct] +
         RampU[t, Gpe, t61, rtr] +
         RampD[t, Gpe, t61 + rtr, rtr] ;

G57[t_]= Trap[t, Gs12, t5, rt, sinct] +
         Trap[t, Gro, t71, rtr, grot] ;

(* for multiple pulses *)
(* use FG[t_] = Integrate[G61[t]+ G62[t] + G63[t] + G64[t] ,t] ; *)
(* to check out the series *)
G61[t_] = RampU[t, Gpe, t71 - rtr, rtr] + RampD[t, Gpe, t71, rtr] ;
G62[t_] = RampU[t, Gpe, t72 - rtr, rtr] + RamD[t, Gpe, t72 , rtr] ;
G63[t_] = RampU[t, Gpe, t73 - rtr, rtr] + RampD[t, Gpe, t73 , rtr] ;
G64[t_] = RampU[t, Gpe, t74 - rtr, rtr] + RampD[t, Gpe, t74 , rtr] ;

G71[t_] = RampU[t,Gro,t71,rtr]+Square[t,Gro,t71+rtr,grot/2] ;

G7172[t_]= Square[t, Gro, t72 - rtr - grot/2, grot/2] +
         RampD[t, Gro, t72 - rtr , rtr] +
         RampU[t, -Gro, t72, rtr] +
         Square[t, -Gro, t72 + rtr, grot/2] ;
```

```
G7273[t_]= Square[t, -Gro, t73 - rtr - grot/2, grot/2] +
             RampD[t, -Gro, t73 - rtr , rtr] +
             RampU[t, Gro, t73 , rtr] +
             Square[t, Gro, t73 + rtr, grot/2] ;

(* the main program starts here *)
(* define evaluate unistep function *)
eus[z_] = If[(z /. time3 /. time2 /. time1 ) < 0 , 0 , 1 ]

(* evaluate b matrix expression *)
(* define FG[t] for the above gradient pulses, for example *)
FG[t_] = Integrate[Gsq[t] ,t] ;

f1 = FG[t] /. t -> te/2 f = Simplify[ f1 /. UnitStep[x:__] -> eus[x] ]

(* the UnitStep[x:__] -> eus[x] ] will evaluate the unitstep, without *)
(*evaluating the other analytical parameters *)

b3v = Simplify[ 4 * f^2 * (te/2) ]

(* b2 value this takes some time to calculate *)
b2i = Integrate[FG[t], {t,te/2,te}] ;
b2 = Simplify[ b2i /. UnitStep[x:__] -> eus[x] ] ;
b2v = Simplify[- 4 * f * b2]

(* b1 value , this takes a long time to calculate*)
b1[t_] = FG[t]^2 ; b1i = Integrate[b1[t], {t, 0,te}] ;
b1v = Simplify[ b1i /. UnitStep[x:__] -> eus[x] ]

(* calculate final expression *)
b = ExpandAll[b1v+b2v+b3v]

(* save it as a file *)
Write["b_23.m'',%]

(* basic algebra finally gives the analytical expressions; tau values x gradient amplitudes *)
End[]
```

Chapter 5—Appendix B

THIS MATHEMATICA COMPUTER CODE HAS THE ANALYTICAL EXPRESSIONS FOR THE B MATRIX, IT IS USED TO CALCULATE THE VALUES FOR THE B MATRIX GIVEN THE IMAGING PARAMETERS AND THE DIFFUSION GRADIENT STRENGTHS FOR AN 2DFT PULSE SEQUENCE

```
(* filename = 2DFT_B_matrix.m *)
(* insert the pulse parameters, i.e., pulse duration and delays in usec, see Fig. 3A for
for definition of parameters, assumes trapezoidal diffusion and crusher gradient *)

time1 ={ te -> 40000., sinct -> 2000., rt -> 200., pet -> 2000., dift -> 4000.,
      difd -> 4000., atde -> 6414.5, crut -> 2000., crud -> 2000.) ;

dtime = (2 * rt + dift + difd) /. time1
ctime = (2 * rt + crut + crud) /. time1
tel = te/2 - ((sinct + 2 rt) + pet + dtime + ctime) /. time1
te2 = te/2 - (sinct/2 + 2 rt + dtime + ctime + atde/2) /. time1

(* time the pulse turn on during sequence in usec *)
time2 ={ t2 -> sinct / 2 + rt ,
       t31 -> sinct / 2 + rt + pet + te1 ,
       t41 -> sinct / 2 + rt + pet + te1 + dtime ,
       t5  -> sinct / 2 + rt + pet + te1 + dtime + ctime ,
       t42 -> 3 sinct / 2 + 3 rt + pet + te1 + dtime + ctime + crud ,
       t32 -> 3 sinct / 2 + 3 rt + pet + te1 + dtime + 2 ctime + difd ,
       t6  -> 3 sinct / 2 + 3 rt + pet + te1 + te2 + 2 dtime + 2 ctime ) ;

time3 = time2 /. time1

gamma = 26751

deltas = {dl -> sinct + rt,
       d2 -> pet ,
       d3 -> dift + rt ,
       d4 -> crut + rt ,
       d5 -> sinct + rt ,
       d6 -> atde + rt ) /. time1

bigdeltas = { Bd2 -> te - t2, Bd3 -> t32 - t31 , Bd4 -> t42 - t41
            } /. time3 /. time1

taus ={
       tau11 -> (d1^2 * ( te - (d1 /3)) + (rt^3 /30) - ((d1 * rt^2)/6))/4,
       tau12 -> (d2 * d1 *( Bd2 - (d2 /2)) ) / Pi,
       tau13 -> ( d1 * d3 * Bd3 )/2,
       tau14 -> ( d1 * d4 * Bd4 )/2,
       tau15 -> (d1 * ( d5^2 + (rt^2)/3 ) ) /8,
       tau16 -> - (d1 * ( d6^2 + (rt^2)/3 ) ) /16,
       tau22 -> 4*( d2^2 * ( Bd2 - (5/8) * d2) ) / ( Pi^2 ),
       tau23 -> 2 * ( d2 * d3 * Bd3 ) / Pi,
       tau24 -> 2 * ( d2 * d4 * Bd4 ) / Pi,
       tau25 -> (d2 * ( d5^2 + (rt^2)/3 ) ) /(2 * Pi),
       tau26 -> - (d2 * ( d6^2 + (rt^2)/3 ) ) /(4 * Pi),
       tau33 -> (d3^2 *(Bd3 - (d3 /3) ) + (rt^3 /30) - ((d3 * rt^2) /6)),
       tau34 -> (d3 * d4 * Bd4 ),
       tau35 -> (d3 * (d5^2 + (rt^2)/3))/4,
       tau36 -> 0.0,
       tau44 -> (d4^2 *(Bd4 - (d4 /3)) + (rt^3 /30) - ((d4 * rt^2) /6)) ,
       tau45 -> (d4 * (d5^2 + (rt^2)/3))/4,
       tau46 -> 0.0 ,
       tau55 -> ( d5^3/6 + (rt^3)/30)/2,
       tau56 -> 0.0,
```

```
      tau66 -> (d6^3/6 + (rt^3)/30)/4
                                              } /. deltas /. bigdeltas /. time1

(* 10^-18 needed to convert usec to sec *)
(* Gdi (i = read, phase, or slice) -> 0 to max G/mm *)
(* watch direction (+/-) *)

Brr = N[Expand[gamma^2 * 10^-18 ( Grdp^2 tau22 + 2 Gdr Grdp tau23 +
       2 Gcr Grdp tau24 + Gdr^2 tau33 + Gcr^2 tau44 +
       2 Gcr Gdr tau34 + Gro^2 tau66 + 2 Gro Grdp tau26 ) /. taus ]]

Bpp = N[Expand[gamma^2 * 10^-18 ( Gpe^2 tau22 + 2 Gdp Gpe tau23 +
       2 Gcp Gpe tau24 + Gdp^2 tau33 + Gcp^2 tau44 + 2 Gcp Gdp tau34 ) /. taus ]]

Bss = N[Expand[gamma^2 * 10^-18 ( Gsl^2 (tau11 + tau15 + tau55/4) +
       2 Gsl Gsrf (tau12 + tau25/2) + 2 Gsl Gds (tau13 + tau35/2) +
       2 Gsl Gcs (tau14 + tau45/2) + Gsrf^2 tau22 + 2 Gds Gsrf tau23 +
       2 Gcs Gsrf tau24 + Gds^2 tau33 + Gcs^2 tau44 + 2 Gcs Gds tau34 ) /. taus ]]

Brp = N[Expand[gamma^2 * 10^-18 ( Gpe Grdp tau22 + Gdr Gdp tau33 +
       Gcp Gcr tau44 + (Gpe Gdr + Grdp Gdp) tau23 +
       (Gpe Gcr + Grdp Gcp) tau24 + Gpe Gro tau26 +
       (Gdp Gcr + Gdr Gcp ) tau34 ) /. taus ]]

Bps = N[Expand[gamma^2 * 10^-18 ( Gsl Gpe (tau12 + tau25 /2 ) +
       Gsl Gdp (tau13 + tau35 /2 ) + Gsl Gcp (tau14 + tau45 /2 ) +
       Gpe Gsrf tau22 + (Gpe Gds + Gsrf Gdp) tau23 +
       (Gpe Gcs + Gsrf Gcp) tau24 + Gdp Gds tau33 +
       (Gds Gcp + Gdp Gcs) tau34 + Gcs Gcp tau44 ) /. taus ]]

Brs = N[Expand[gamma^2 * 10^-18 (Gsl Grdp (tau12 + tau25/2) +
       Gsl Gdr (tau13 + tau35/2) + Gsl Gcr (tau14 + tau45/2) +
       Gsl Gro tau16 + Gsrf Grdp tau22 + (Gsrf Gdr + Grdp Gds) tau23 +
       (Gsrf Gcr + Grdp Gcs) tau24 + Gsrf Gro tau26 + Gdr Gds tau33 +
       (Gdr Gcs + Gds Gcr) tau34 + Gcs Gcr tau44 ) /. taus ]]

(* the pulse strength in Gauss/mm, note direction (+/-) *)
(* Gauss/mm , for an 2DFT spin-echo imaging pulse sequence *)
subamp = { Gsl ->- 0.281, Gsrf -> 0.248, Gpe ->0, Grdp -> 0.129, Gro -> 0.489,
                  Gcr -> 0.20, Gcp ->a 0.2, Gcs -> -0.20 } ;

Brr /. subamp /. Gdr -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
Bpp /. subamp /. Gdp -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
Bss /. subamp /. Gds -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
Brp /. subamp /. Gdr -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
       % /. Gdp -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
Bps /. subamp /. Gdp -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
       % /. Gds -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
Brs /. subamp /. Gdr -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }
       % /. Gds -> { 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 }

(* use these values to construct the b-matrix
  [ Brr , Brp , Brs ]
  [ Brp , Bpp , Bps ]
  [ Brs , Bps , Bss ] *)

End[]
```

Chapter 5—Appendix C

THIS MATHEMATICA COMPUTER CODE HAS THE ANALYTICAL EXPRESSIONS FOR THE B MATRIX, IT IS USED TO CALCULATE THE VALUES FOR THE B MATRIX GIVEN THE IMAGING PARAMETERS AND THE DIFFUSION GRADIENT STRENGTHS FOR AN EPI PULSE SEQUENCE

```
(* filename = epi_B_trap.m *)
(* insert the pulse parameters, i.e., pulse duration and delays in usec, see Fig. 3B for
definition of parameters, assumes trapezoidal diffusion and crusher gradient *)

time1 = {te -> 85710.,rtr -> 200.,rt -> 500., sinct -> 1750., idpt -> 2000.,
      dift -> 10000. , difd -> 5000., crut -> 2000.0, crud -> 1000.,
      grot -> 240. , res -> 64 } ;

time2 ={d1 -> sinct + rt,d2 -> idpt, d3 -> dift + rt, d4 -> crut + rt,
       d5 -> sinct + rt, d6 -> rtr , d7 -> grot + rtr,
      deld -> dift + 2 rt + difd, delc -> crut + 2 rt + crud,
      te1 -> (te/2. ) - ( idpt + 6 rt + sinct + dift + difd + crud + crut )
                                                                  } /. time1

gamma = 26751

(* start time for pulse gradients *)
time3 = { t1 -> 0.,
        t2  -> sinct / 2 + rt ,
        t31 -> sinct / 2 + rt + idpt + te1 ,
        t41 -> sinct / 2 + rt + idpt + te1 + deld ,
        t5  -> sinct / 2 + rt + idpt + te1 + deld + delc ,
        t42 -> 3 sinct / 2 + 3 rt + idpt + te1 + deld + delc + crud ,
        t32 -> 3 sinct / 2 + 3 rt + idpt + te1 + deld + 2 delc + difd ,
        t61 -> 3 sinct / 2 + 3 rt + idpt + te1 + 2 deld + 2 delc + 1000. ,
        t71 -> 3 sinct / 2 + 3 rt + idpt + te1 + 2 deld + 2 delc + 1000. + rtr
                                                                  } /. time 2 /. time1

(* evaluate some parameters to define arrays and summations *)
evres = res/2 /. time1
evd7 = d7 + rtr /. time2
evt61 = t61 /. time3
evte = te /. time1
evrtr = rtr /. time1
Array[t6,1,evres]
Array[Bd6,1,evres]
Do [t6[i] = evt61 + evd7 * (i-1) , { i,1,evres}]
Do [Bd6[i] = evte - t6[i] , {i,evres}]

Deltas = { Bd1 -> te - t1, Bd2 -> te - t2, Bd3 -> t32 - t31 ,
             Bd4 -> t42 - t41, Bd7 -> te - t71 } /. time3 /.time2 /. time1

tausum1 = Sum[( Bd6[i] - evrtr) , {i,1,evres}]
tausum2 = Sum[( (2*i-1) * Bd6[i] - (i*67/30 - 1) evrtr ), {i,1,evres}]

taus ={
      tau11 -> ( d1^2 * ( Bd1 - d1 / 3 ) + rt^3 / 30 - ( d1 * rt^2 ) / 6 ) / 4,
      tau12 -> ( d2 * d1 * (Bd2 - d2 / 2 )) / Pi,
      tau13 -> ( d1 * d3 * (Bd3 )) / 2,
      tau14 -> ( d1 * d4 * ( Bd4 )) / 2,
      tau15 -> (d1 * ( d5^2 + rt^2 / 3 )) / 8,
      tau16 -> - rtr * d1 * tausum1 / 2,
      tau17 -> - d1 * (d7 * ( Bd7 - d7 / 4 ) + rtr^2 / 12 - ( d7 * rtr ) / 2 ) / 4,
      tau22 -> 4 * d2^2 * ( Bd2 - 5/8. d2 ) / Pi^2 ,
      tau23 -> 2 * d2 * d3 * ( Bd3 ) / Pi,
      tau24 -> 2 * d2 * d4 * ( Bd4 ) / Pi,
```

```
        tau25 -> d2 * ( d5^2 + rt^2 / 3 ) / ( 2 * Pi ),
        tau26 -> - 2 * rtr * d2 * tausum1 / Pi,
        tau27 -> - d2 * ( d7 * ( Bd7 - d7 / 4 ) + rtr^2 / 12 - ( d7 * rtr ) / 2 ) / Pi ,
        tau33 -> d3^2 * ( Bd3 - d3 / 3 ) + rt^3 / 30 - ( d3 * rt^2 ) / 6 ,
        tau34 -> d3 * d4 * Bd4 ,
        tau35 -> ( d3 * ( d5^2 + rt^2/3 )) / 4,
        tau36 -> 0.0,
        tau37 -> 0.0,
        tau44 -> d4^2 * ( Bd4 - d4 / 3 ) + rt^3 / 30 - ( d4 * rt^2 ) / 6 ,
        tau45 -> ( d3 * ( d5^2 + rt^2/3 )) / 4,
        tau46 -> 0.0,
        tau47 -> 0.0,
        tau55 -> d5^3/ 12. + rt^3 / 60.,
        tau56 -> 0.0 ,
        tau57 -> 0.0 ,
        tau66 -> rtr^2 * tausum2 ,
        tau67 -> rtr * ( d7 * Bd7 - rtr^2 / 60 ) / 4,
        tau77 -> ( d7^2 * ( Bd7 - d7 / 3 ) + rtr^3 / 30 - ( d7^2 * rtr ) / 2 ) / 4 ,
        tau77m -> (res / 2. - 1)* ( d7^3 /12 + rtr^3 /60 - ( d7 * rtr^2 ) /12 + (d7^2 * rtr)/4 )
                                               } /. Deltas /. time1 /. time2 /. time3

(* Gauss/mm crusher are 10 % of maximum , for an EPI pulse sequence *)
subamp ={ Gsl -> 0.281, Gsrf -> 0.248, Gpdp -> 0.123, Grdp -> 0.129, Gro -> 0.489,
     Gcr -> 0.201, Gcp -> 0.192, Gcs -> -0.191 , Gpe -> 0.0244 } :

(* 10^-18 needed to convert usec to sec *)

Brr = N[Expand[gamma^2 * 10^-18 ( Grdp^2 tau22 + 2 Gdr Grdp tau23 +
               2 Gcr Grdp tau24 + 2 Gro Grdp tau27 + Gdr^2 tau33 +
               2 Gcr Gdr tau34 + Gcr^2 tau44 + Gro^2 * (tau77 + tau77m) ) /. taus ]]

Bpp = N[Expand[gamma^2 * 10^-18 ( Gpdp^2 tau22 + 2 Gdp Gpdp tau23 +
               2 Gcp Gpdp tau24 + 2 Gpdp Gpe tau26 + Gdp^2 tau33 +
               2 Gcp tau34 + Gcp^2 tau44 + Gpe^2 tau66 ) /. taus ]]

Bss = N[Expand[gamma^2 * 10^-18 ( Gsl^2 (tau11 + tau15 + tau55 /4) +
               2 Gsl Gsrf (tau12 + tau25 /2) + 2 Gsl Gds (tau13 + tau35 /2) +
               2 Gsl Gcs (tau14 +tau45 /2) + Gsrf^2 tau22 + 2 Gds Gsrf tau23 +
               2 Gcs Gsrf tau24 + Gds^2 tau33 + 2 Gds Gcs tau34 + Gcs^2 tau44 ) /. taus ]]

Brp = N[Expand[gamma^2 * 10^-18 ( Gpdp Grdp tau22 + Gdr Gdp tau33 +
               Gcr Gcp tau44 + (Gpdp Gdr + Grdp Gdp) tau23 +
               (Gpdp Gcr + Grdp Gcp) tau24 + Grdp Gpe tau26 + Gpdp Gro tau27 +
               (Gcp Gdr + Gcr Gdp) tau34 + Gpe Gro tau67 ) /. taus]]

Bps = N[Expand[gamma^2 * 10^-18 ( Gsl Gpdp (tau12 + tau25 /2 ) +
               Gsl Gdp (tau13 + tau35 /2 ) + Gsl Gcp (tau14 + tau45 /2 ) +
               Gsl Gpe tau16 + Gpdp Gsrf tau22 + (Gpdp Gds + Gsrf Gdp) tau23 +
               (Gpdp Gcs + Gsrf Gcp) tau24 + Gsrf Gpe tau26 +
               Gdp Gds tau33 + (Gcp Gds + Gcs Gdp) tau34 + Gcp Gcs tau44 ) /. taus ]]

Brs = N[Expand[gamma^2 * 10^-18 (Gsl Grdp (tau12 + tau25 /2) +
               Gsl Gdr (tau13 + tau35 /2) + Gsl Gcr (tau14 + tau45 /2) +
               Gsl Gro (tau17 + tau57 /2) + Gsrf Grdp tau22 +
               (Gsrf Gdr + Grdp Gds) tau23 + (Gsrf Gcr + Grdp Gcs) tau24 +
               Gsrf Gro tau27 + Gdr Gds tau33 + (Gcs Gdr + Gcr Gds) tau34 +
                Gcr Gcs tau44 ) /. taus ]]

(* diffusion gradient Gauss/mm, Gd -> {0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0} or whatever values *)
(*calculate b matrix values *)

Brr /. subamp /. Gdr ->{ 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}
Bpp /. subamp /. Gdp -> { 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}
Bss /. subamp /. Gds -> { 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}
Brp /. subamp /. Gdr -> { 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}
       % /. Gdp -> { 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}
```

```
Bps /. subamp /. Gdp -> { 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}
       % /. Gds -> { 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}
Brs /. subamp /. Gdr -> { 0,.1,.2,.3,.4,.5,.6,.7,8,.9,1.0}
       % /. Gds -> { 0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1.0}

(* use these values to construct the b-matrix
  [ Brr , Brp , Brs ]
  [ Brp , Bpp , Bps ]
  [ Brs , Bps , Bss ] *)

End[]
```