

Implementation Notes on Beltrami Framework for Free Water DTI

Marc Golub

July 30, 2018

1 Introduction

The purpose of this document is to briefly explain the logic behind the implementation of a free water elimination algorithm for single shell DTI, based on the work of (Pasternak, Sochen, Gur, Intrator, & Assaf, 2009). The mathematical concepts and the meaning of the metrics (from an image processing perspective) used in this algorithm are explained in more detail in (Gur, Pasternak, & Sochen, 2009).

2 Algorithm Overview

The objective of the algorithm is to fit a bi-tensor model for each voxel, which predicts that the signal attenuation of a dwMRI acquisition is given by:

$$[A_{bitensor}]_k = f[A_{tissue}]_k + (1 - f)A_{water} \quad (1)$$

Where $0 < f < 1$ is the tissue volume fraction and $(1 - f)$ is the free water fraction. This model assumes that each voxel is composed by two compartments, one which is responsible for anisotropic diffusion (tissue), and the other for isotropic diffusion (free moving water in the extracellular space):

$$[A_{tissue}]_k = e^{-bq_k^T D q_k} \quad (2)$$

$$A_{water} = e^{-bd} \quad (3)$$

Where b reflects the strength and time duration of the applied gradients, q_k is the normalized direction of the k th applied gradient (a minimum of 6 non-colinear

directions is needed), $d = 3 \times 10^{-3} \text{mm}^2 \text{s}^{-1}$ is the diffusion coefficient of free water at body temperature, and D is the diffusion tensor, which is symmetric positive definite (SPD).

$$D = \begin{bmatrix} D_1 & D_4 & D_5 \\ D_4 & D_2 & D_6 \\ D_5 & D_6 & D_3 \end{bmatrix} \quad (4)$$

There are 7 independent parameters to be estimated, in theory, one can set a fixed value for f and perform weighted linear least squares (similar to classic DTI) to fit the best D , this process can be repeated for every value of f between 0 and 1. Therefore, there is an infinite number of viable solutions for the bi-tensor model, choosing the best one is not trivial.

To stabilize the convergence, additional constraints are needed. The Beltrami framework makes use of local information, namely the spatial derivatives of each tensor component to construct a metric that measures distances between tensors, this metric is used to enforce piece-wise smoothness (from voxel to voxel) of the tensor components, which helps the convergence.

A good initialization of f is important, restricting f to a narrower interval also helps:

$$0 < f_{min} < f < f_{max} < 1$$

After initializing f and D , gradient descent is performed, the tensor components are parameterized into Iwasawa coordinates X and the fitting is performed on these coordinates. There are 7 evolution equations, in each iteration 3 incrementals are computed for each Iwasawa parameter: the **fidelity**, **laplace-beltrami** and **levi-civita** terms. A separate incremental is computed for f .

$$X^n = X^{n-1} + dt \times (\Delta F + \Delta L + \Delta C) \quad (5)$$

$$f^n = f^{n-1} + dt \times \Delta f \quad (6)$$

Where $X^n = [X_1, X_2, X_3, X_4, X_5, X_6]^n$ is the vector of 6 independent Iwasawa coordinates at the n th iteration, dt is the time step, F, L, C are the fidelity, laplace and civita terms. At each iteration, if f becomes larger than f_{max} or smaller than f_{min} , it is projected back. At the end of each iteration, X is converted back to tensor form D . The following diagram gives an overview of the Beltrami algorithm.

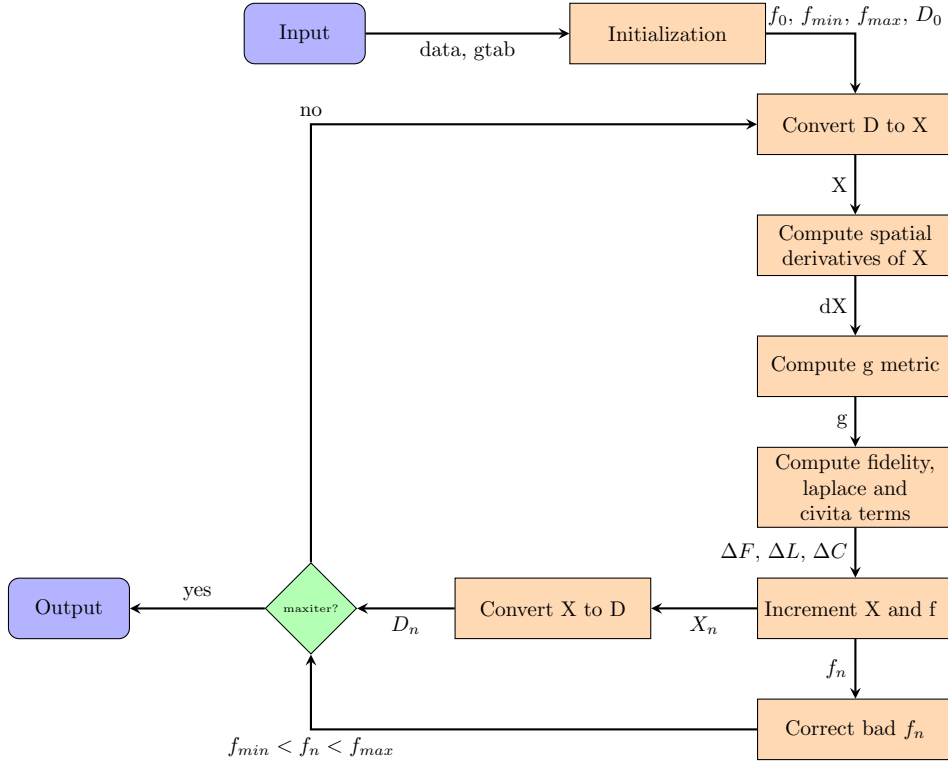


Figure 1: Overview of the Beltrami algorithm.

The steps and metrics used in this algorithm are explained with more detail in the following sections.

3 Initialization

The volume fraction f is initialized first, using only the $S0$ image, the initial f_0 is then used to correct the observed attenuations for the presence of free water. Regular DTI is applied to the corrected attenuations to obtain the initial guess for the diffusion components.

3.1 Volume fraction f_0 , f_{min} and f_{max}

By taking equation 1 and solving with respect to :

$$f = \frac{[\hat{A}]_k - A_{water}}{[A_{tissue}]_k - A_{water}}$$

Where $[A_{bitensor}]_k$ was replaced by $[\hat{A}]_k$, to minimize the expression above, one

must minimize the numerator and maximize the denominator.

$$f_{min} = \frac{\min([\hat{A}]_k) - A_{water}}{A_{tmax} - A_{water}}$$

$$f_{max} = \frac{\max([\hat{A}]_k) - A_{water}}{A_{tmin} - A_{water}}$$

Where $\min([\hat{A}]_k)$ and $\max([\hat{A}]_k)$ are the minimum and maximum observed attenuations, $A_{tmin} = e^{-b\lambda_{max}}$ and $A_{tmax} = e^{-b\lambda_{min}}$ are the minimum and maximum expected attenuations in tissue. The values f_{min} and f_{max} remain fixed for the entire algorithm. Note that the equations above are different from the ones provided in the article.

The f_0 image is initialized by:

$$f_0 = 1 - \frac{\log(S_0/S_t)}{\log(S_w/S_t)} \quad (7)$$

Where S_t is the typical intensity of a voxel containing only tissue and S_w for a voxel with free water, usually picked from the ventricles with CSF. The S_t and S_w values are picked from S_0 image and depend on the dataset. Voxels for which f_0 lies outside f_{min} and f_{max} , are replaced by $\frac{f_{min}+f_{max}}{2}$.

3.2 Initial D

Having f_0 , and solving equation 1 with respect to $[A_{tissue}]_k$:

$$[\hat{A}_t]_k = \frac{[\hat{A}]_k - (1 - f_0)A_{water}}{f_0} \quad (8)$$

The quantity above can be regarded as the observed tissue attenuation, corrected for free water. Regular DTI is applied to this quantity to obtain the initial diffusion tensor components.

4 Iwasawa decomposition

Let P_3 denote the space of all 3×3 SPD matrices, any $D \in P_3$ can be decomposed into a unique representation:

$$D = N^T AN = \begin{bmatrix} 1 & 0 & 0 \\ X_4 & 1 & 0 \\ X_5 & X_6 & 1 \end{bmatrix} \begin{bmatrix} X_1 & 0 & 0 \\ 0 & X_2 & 0 \\ 0 & 0 & X_3 \end{bmatrix} \begin{bmatrix} 1 & X_4 & X_5 \\ 0 & 1 & X_6 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$D = \begin{bmatrix} X_1 & X_1X_4 & X_1X_5 \\ X_1X_4 & X_2 + X_1X_4^2 & X_1X_4X_5 + X_2X_6 \\ X_1X_5 & X_1X_4X_5 + X_2X_6 & X_3 + X_1X_5^2 + X_2X_6^2 \end{bmatrix} \quad (10)$$

Where X_1, \dots, X_6 are the 6 independent Iwasawa parameters. According to the authors of the article, this parameterization is done to simplify the complicated expressions of the metrics described further ahead, thus simplifying the numerical implementation.

Taking equation 10 and solving with respect to X (with the help of a *Wolfram Mathematica* script):

$$\begin{aligned} X_1 &= D_1 & X_4 &= \frac{D_4}{D_1} \\ X_2 &= D_2 - \frac{D_4^2}{D_1} & X_5 &= \frac{D_5}{D_1} \\ X_3 &= \frac{D_3D_4^2 + D_5(D_2D_5 - 2D_4D_6) + D_1(D_6^2 - D_2D_3)}{D_4^2 - D_1D_2} & X_6 &= \frac{D_1D_6 - D_4D_5}{D_1D_2 - D_4^2} \end{aligned}$$

Where D_1, \dots, D_6 are the diffusion tensor components from equation 4. The implementation can be simplified if the conversion from D to X is done in a specific order:

$$\begin{aligned} X_1 &= D_1 \\ X_4 &= \frac{D_4}{D_1} \\ X_5 &= \frac{D_5}{D_1} \\ X_2 &= D_2 - X_1X_4^2 \\ X_6 &= \frac{D_6 - X_1X_4X_5}{X_2} \\ X_3 &= D_3 - X_1X_5^2 - X_2X_6^2 \end{aligned}$$

5 The induced metric γ

The Beltrami framework is heavily based on Riemannian geometry, which is a field of mathematics that studies manifolds equipped with a metric that measures distances between points on the manifold. A manifold is a topological space that is locally euclidean. In computer vision, an image can be locally regarded as the product of 2 spaces: $E = B \times F$, where B is the base manifold, F is the feature space and E is the total space. This concept is known as "fiber

bundle”, where the features (fibers) are attached to a base, thus the name. A specific image is a section of the fiber bundle, defined by the unique mapping $X : B \mapsto E$. For example, the space of grayscale images can be locally described by $\mathbb{R}^3 = \mathbb{R}^2 \times \mathbb{R}^+$, where the base manifold is the 2D image domain \mathbb{R}^2 and the feature space is the intensity \mathbb{R}^+ ($0 < I < 255$). In this case, a specific grayscale image is a section of the total space \mathbb{R}^3 (also called the spatial-feature manifold), given by the mapping $X : \mathbb{R}^2 \mapsto \mathbb{R}^3$, $(x, y) \mapsto (x, y, I(x, y))$, which assigns every pixel in the image domain to a point in the spatial-feature manifold. In other words, a grayscale image can be seen as a 2D surface embedded in 3D space.

Having defined the geometry of the spatial-feature manifold and the map X , one can choose a metric h to measure distances on the manifold:

$$ds^2 = h_{ij} dX^i dX^j \quad (11)$$

Under the assumption that infinitesimal distances on the spatial-feature space are equal to infinitesimal distances on the image space, the metric h induces a metric γ on the image domain, this is also called the pullback metric:

$$ds^2 = \gamma_{\mu\nu} dx^\mu dx^\nu \quad (12)$$

For the grayscale case, h and γ are 3×3 and 2×2 tensors, respectively, $dx^1 = dx$ and $dx^2 = dy$. By the chain rule:

$$dX^i = \partial_\mu X^i dx^\mu = \frac{\partial X^i}{\partial x} dx + \frac{\partial X^i}{\partial y} dy \quad (13)$$

Combining equations 11, 12 and 13, gives the expression for the pullback metric:

$$\gamma_{\mu\nu} = h_{ij} \partial_\mu X^i \partial_\nu X^j \quad (14)$$

Where $\partial_\mu X^i = \frac{\partial X^i}{\partial x^\mu}$ is the partial derivative of the i th component of the spatial-feature manifold with respect to x^μ .

Similarly to the grayscale case, a specific DTI volume can be regarded as a section of a fiber bundle, described by the embedding map $X : \mathbb{R}^3 \mapsto \mathbb{R}^3 \times P_3$, where $\mathbb{R}^3 \times P_3$ is the 9-dimensional spatial-feature space, and P_3 is the 6-dimensional feature space of all SPD matrices.

From now on, let Σ denote the 3D image manifold, M is the 9D spatial-feature manifold, γ and h the respective metrics, and $X : \Sigma \mapsto M$ the embedding map. Having the Iwasawa decomposition in mind, the embedding map is given by:

$$X : (x, y, z) \mapsto (x, y, z, X_1, X_2, X_3, X_4, X_5, X_6) \quad (15)$$

X^i will denote the i th component of this map.

With this parameterization, the spatial-feature metric h is defined as:

$$h = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta \frac{1}{X_1^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta \frac{1}{X_2^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta \frac{1}{X_3^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \beta \frac{2X_1(X_3+X_2X_6^2)}{X_2X_3} & -\beta \frac{2X_1X_6}{X_3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\beta \frac{2X_1X_6}{X_3} & \beta \frac{2X_1}{X_3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta \frac{2X_2}{X_3} \end{bmatrix}$$

The parameter β is introduced to control the ratio between the feature space and image space metrics. Expression 14 can be regarded as a multiplication of 3 matrices, let the quantity $\partial_\mu X^i \equiv dX$ be the matrix that holds all spatial derivatives of the Iwasawa parameters, with $\mu = 1 : 3$ rows and $i = 1 : 9$ columns:

$$\partial_\mu X^i \equiv dX = \begin{bmatrix} 1 & 0 & 0 & \frac{\partial X_1}{\partial x} & \frac{\partial X_2}{\partial x} & \frac{\partial X_3}{\partial x} & \frac{\partial X_4}{\partial x} & \frac{\partial X_5}{\partial x} & \frac{\partial X_6}{\partial x} \\ 0 & 1 & 0 & \frac{\partial X_1}{\partial y} & \frac{\partial X_2}{\partial y} & \frac{\partial X_3}{\partial y} & \frac{\partial X_4}{\partial y} & \frac{\partial X_5}{\partial y} & \frac{\partial X_6}{\partial y} \\ 0 & 0 & 1 & \frac{\partial X_1}{\partial z} & \frac{\partial X_2}{\partial z} & \frac{\partial X_3}{\partial z} & \frac{\partial X_4}{\partial z} & \frac{\partial X_5}{\partial z} & \frac{\partial X_6}{\partial z} \end{bmatrix} \quad (16)$$

Where the 1's and 0's appear because for $i = 1 : 3$, X^i are the spatial coordinates x, y, z . The pullback metric is given in matrix form by:

$$[\gamma] = [dX][h][dX]^T \quad (17)$$

Which results in a 3×3 symmetric tensor. The 6 independent components were computed with the help of a *Wolfram Mathematica* script:

$$\begin{aligned}
\gamma_{11} &= 1 + \beta \left(\sum_{i=4}^9 h_{ii} \left(\frac{\partial X^i}{\partial x} \right)^2 + 2h_{78} \frac{\partial X^7}{\partial x} \frac{\partial X^8}{\partial x} \right) \\
\gamma_{22} &= 1 + \beta \left(\sum_{i=4}^9 h_{ii} \left(\frac{\partial X^i}{\partial y} \right)^2 + 2h_{78} \frac{\partial X^7}{\partial y} \frac{\partial X^8}{\partial y} \right) \\
\gamma_{33} &= 1 + \beta \left(\sum_{i=4}^9 h_{ii} \left(\frac{\partial X^i}{\partial z} \right)^2 + 2h_{78} \frac{\partial X^7}{\partial z} \frac{\partial X^8}{\partial z} \right) \\
\gamma_{12} &= \beta \left(\sum_{i=4}^9 h_{ii} \frac{\partial X^i}{\partial x} \frac{\partial X^i}{\partial y} + h_{78} \left(\frac{\partial X^7}{\partial y} \frac{\partial X^8}{\partial x} + \frac{\partial X^7}{\partial x} \frac{\partial X^8}{\partial y} \right) \right) \\
\gamma_{13} &= \beta \left(\sum_{i=4}^9 h_{ii} \frac{\partial X^i}{\partial x} \frac{\partial X^i}{\partial z} + h_{78} \left(\frac{\partial X^7}{\partial z} \frac{\partial X^8}{\partial x} + \frac{\partial X^7}{\partial x} \frac{\partial X^8}{\partial z} \right) \right) \\
\gamma_{23} &= \beta \left(\sum_{i=4}^9 h_{ii} \frac{\partial X^i}{\partial y} \frac{\partial X^i}{\partial z} + h_{78} \left(\frac{\partial X^7}{\partial z} \frac{\partial X^8}{\partial y} + \frac{\partial X^7}{\partial y} \frac{\partial X^8}{\partial z} \right) \right)
\end{aligned}$$

To implement the equations above, the partial derivatives are discretized into finite differences. In the work of (Pasternak et al., 2009), the difference scheme was not specified, a forward difference scheme was used in this implementation.

6 Evolution of X and f

Having the metric γ , a set of incrementals is computed to update the parameters. There are 6 evolution equations for X^i , $i = 4 : 9$, because the image domain coordinates ($i = 1 : 3$) are fixed. A separate incremental is computed for f .

6.1 The Fidelity term

The fidelity term maintains the fitted model close to the observed signal, and its expression is given by:

$$\Delta F^i = -\alpha b \frac{1}{\sqrt{|\gamma|}} \left(\sum_{k=1}^N \left([A_{bitensor}]_k - [\hat{A}]_k \right) [A_{tissue}]_k \left(q_k^T \frac{\partial D}{\partial X^i} q_k \right) \right) \quad (18)$$

Where α controls the weight of the fidelity term, $|\gamma|$ is the determinant of the metric, $[A_{bi-tensor}]_k$ and A_{tissue} are given by equations 1 and 2, $[\hat{A}]_k$ is the observed attenuation for the k th direction.

The $\frac{\partial D}{\partial X^i}$ matrix is obtained by differentiating D with respect to the i th entry of X elementwise:

$$\frac{\partial D}{\partial X^4} = \begin{bmatrix} 1 & X_4 & X_5 \\ X_4 & X_4^2 & X_4 X_5 \\ X_5 & X_4 X_5 & X_5^2 \end{bmatrix}$$

$$\frac{\partial D}{\partial X^5} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & X_6 \\ 0 & X_6 & X_6^2 \end{bmatrix}$$

$$\frac{\partial D}{\partial X^6} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial D}{\partial X^7} = \begin{bmatrix} 0 & X_1 & 0 \\ X_1 & 2X_1 X_4 & X_1 X_5 \\ 0 & X_1 X_5 & 0 \end{bmatrix}$$

$$\frac{\partial D}{\partial X^8} = \begin{bmatrix} 0 & 0 & X_1 \\ 0 & 0 & X_1 X_4 \\ X_1 & X_1 X_4 & 2X_1 X_5 \end{bmatrix}$$

$$\frac{\partial D}{\partial X^9} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & X_2 \\ 0 & X_2 & 2X_2 X_6 \end{bmatrix}$$

Note that all the matrices above are symmetric.

6.2 The Laplace-Beltrami operator

The Laplace-Beltrami term acts as a denoising operator, regularizing the image while maintaining important edges:

$$\Delta_\gamma X^i = \frac{1}{\sqrt{|\gamma|}} \partial_\mu (\sqrt{|\gamma|} \gamma^{\mu\nu} \partial_\nu X^i) \quad (19)$$

Where $\gamma^{\mu\nu}$ is the inverse of γ . If i is set to a fixed coordinate, then the quantity $\partial_\mu X^i$ is a column of the matrix dX from expression 16:

$$\partial_\mu X^i = \begin{bmatrix} \frac{\partial X^i}{\partial x} \\ \frac{\partial X^i}{\partial y} \\ \frac{\partial X^i}{\partial z} \end{bmatrix} \quad (20)$$

And the product $\gamma^{\mu\nu}\partial_\nu X^i$ becomes:

$$\begin{aligned}\gamma^{\mu\nu}\partial_\nu X^i &\equiv A^i = \begin{bmatrix} \gamma^{11} & \gamma^{12} & \gamma^{13} \\ \gamma^{12} & \gamma^{22} & \gamma^{23} \\ \gamma^{13} & \gamma^{23} & \gamma^{33} \end{bmatrix} \begin{bmatrix} \frac{\partial X^i}{\partial x} \\ \frac{\partial X^i}{\partial y} \\ \frac{\partial X^i}{\partial z} \end{bmatrix} \\ &= \begin{bmatrix} \gamma^{11}\frac{\partial X^i}{\partial x} + \gamma^{12}\frac{\partial X^i}{\partial y} + \gamma^{13}\frac{\partial X^i}{\partial z} \\ \gamma^{12}\frac{\partial X^i}{\partial x} + \gamma^{22}\frac{\partial X^i}{\partial y} + \gamma^{23}\frac{\partial X^i}{\partial z} \\ \gamma^{13}\frac{\partial X^i}{\partial x} + \gamma^{23}\frac{\partial X^i}{\partial y} + \gamma^{33}\frac{\partial X^i}{\partial z} \end{bmatrix} \end{aligned} \quad (21)$$

The Laplace-Beltrami operator can be seen as the divergence of the vector field A . Let A_n^i denote the n th entry of A^i , then:

$$\Delta_\gamma X^i = \frac{1}{\sqrt{|\gamma|}} \left(\frac{\partial}{\partial x}(\sqrt{|\gamma|}A_1^i) + \frac{\partial}{\partial y}(\sqrt{|\gamma|}A_2^i) + \frac{\partial}{\partial z}(\sqrt{|\gamma|}A_3^i) \right) \quad (22)$$

A backward difference scheme was used to compute the partial derivatives of A .

6.3 The Levi-Civita connection

Due to the Iwasawa parameterization and the choice of a non-euclidean metric h over the spatial-feature manifold, a third term appears in the evolution equations, the Levi-Civita term:

$$\Delta C^i = \Gamma_{jk}^i \gamma^{\mu\nu} \partial_\mu X^j \partial_\nu X^k \quad (23)$$

Where Γ_{jk}^i are the Christoffel symbols, computed with respect to the h metric:

$$\Gamma_{jk}^i = \frac{1}{2} h^{il} (\partial_j h_{lk} + \partial_k h_{jl} - \partial_l h_{jk}) \quad (24)$$

Here, ∂_j denotes the symbolic partial derivatives of h with respect to the Iwasawa parameters, h^{il} is the inverse of h and the indices i, j, k, l range from 4 to 9. Let \hat{h} be a sliced version of h , containing only the important indices, the inverse is then:

$$\hat{h}^{il} = \begin{bmatrix} X_1^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & X_2^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & X_3^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{X_2}{2X_1} & \frac{X_2 X_6}{2X_1} & 0 \\ 0 & 0 & 0 & \frac{X_2 X_6}{2X_1} & \frac{X_3 + X_2 X_6^2}{2X_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{X_3}{2X_2} \end{bmatrix}$$

The parameter β is now omitted because it cancels out in the Christoffel computation. Setting $i = 2$ for example, the only non-zero element of the inverse is for $l = 2$, and equation 24 becomes:

$$\hat{\Gamma}_{jk}^2 = \frac{1}{2} \hat{h}^{22} (\partial_j \hat{h}_{2k} + \partial_k \hat{h}_{j2} - \partial_2 \hat{h}_{jk})$$

By inspection, the only non-zero terms are those for which \hat{h} has a derivative with respect to X_2 , $(j, k) \in \{(2, 2), (4, 4), (6, 6)\}$:

$$\begin{aligned} \partial_2 \hat{h}_{22} &= -\frac{2}{X_2^3} \\ \partial_2 \hat{h}_{44} &= -\frac{2X_1}{X_2^2} \\ \partial_2 \hat{h}_{66} &= -\frac{2}{X_3} \end{aligned}$$

The set of Christoffel symbols used to update X_2 is then:

$$\hat{\Gamma}_{jk}^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{X_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{X_2^2}{X_3} \end{bmatrix}$$

The computation of Christoffel symbols by hand is cumbersome. Due to its sparsity and symmetry ($\Gamma_{jk}^i = \Gamma_{kj}^i$), there are only 26 unique Christoffel symbols, which were computed with a *Mathematica* script:

$$\begin{aligned}
\hat{\Gamma}_{11}^1 &= -\frac{1}{X_1} & \hat{\Gamma}_{44}^1 &= -\frac{X_1^2(X_3 + X_2X_6^2)}{X_2X_3} \\
\hat{\Gamma}_{45}^1 &= -\frac{1}{X_1} & \hat{\Gamma}_{55}^1 &= -\frac{X_1^2}{X_3} \\
\hat{\Gamma}_{22}^2 &= -\frac{1}{X_2} & \hat{\Gamma}_{44}^2 &= X_1 \\
\hat{\Gamma}_{66}^2 &= -\frac{X_2^2}{X_3} & \hat{\Gamma}_{33}^3 &= -\frac{1}{X_3} \\
\hat{\Gamma}_{44}^3 &= X_1X_6^2 & \hat{\Gamma}_{45}^3 &= -X_1X_6 \\
\hat{\Gamma}_{55}^3 &= X_1 & \hat{\Gamma}_{66}^3 &= X_2 \\
\hat{\Gamma}_{14}^4 &= \frac{1}{2X_1} & \hat{\Gamma}_{24}^4 &= -\frac{1}{2X_2} \\
\hat{\Gamma}_{46}^4 &= \frac{X_2X_6}{2X_3} & \hat{\Gamma}_{56}^4 &= -\frac{X_2}{2X_3} \\
\hat{\Gamma}_{24}^5 &= -\frac{X_6}{2X_2} & \hat{\Gamma}_{34}^5 &= \frac{X_6}{2X_3} \\
\hat{\Gamma}_{15}^5 &= \frac{1}{2X_1} & \hat{\Gamma}_{35}^5 &= -\frac{1}{2X_3} \\
\hat{\Gamma}_{46}^5 &= \frac{1}{2}\left(\frac{X_2X_6^2}{X_3} - 1\right) & \hat{\Gamma}_{56}^5 &= -\frac{X_2X_6}{2X_3} \\
\hat{\Gamma}_{44}^6 &= -\frac{X_1X_6}{X_2} & \hat{\Gamma}_{45}^6 &= \frac{X_1}{2X_2} \\
\hat{\Gamma}_{26}^6 &= \frac{1}{2X_2} & \hat{\Gamma}_{36}^6 &= -\frac{1}{2X_3}
\end{aligned}$$

The Christoffel symbols introduce a coupling between the Iwasawa parameters and maintain the diffusion tensor in SPD space during the evolution. Therefore, if the initialization of D is in P_3 , then the flow stays in P_3 . If the chosen metric for h were euclidean, a separate mechanism would be needed to maintain the positive definiteness of D .

Going back to equation 23, let $A_{\mu k} \equiv \gamma^{\mu\nu} \partial_\nu X^k$ and $B_{\mu k}^i \equiv \Gamma_{jk}^i \partial_\mu X^j$.

$$\Delta C^i = \underbrace{\Gamma_{jk}^i}_{B_{\mu k}^i} \partial_\mu X^j \overbrace{\gamma^{\mu\nu} \partial_\nu X^k}^{A_{\mu k}}$$

In matrix form:

$$\begin{aligned} [A] &= [\gamma]^{-1} [d\hat{X}] \\ [B]^i &= [d\hat{X}][\Gamma]^i \end{aligned}$$

Where $d\hat{X}$ is a sliced version of dX from expression 16 (columns 4 : 9). The i th Civita term is obtained by multiplying A and B^i elementwise and summing over all the elements:

$$\Delta C^i = \sum_{\mu=1}^3 \sum_{k=1}^6 A_{\mu k} B_{\mu k}^i$$

Note that the columns of $A_{\mu k}$ are also used to compute the Laplace-Beltrami terms in expression 22, this can be used to avoid repeating computations at the expense of memory by storing matrix A at each iteration.

7 The free water incremental

The free water incremental does not depend on the metric γ and its expression is similar to the fidelity term:

$$\Delta f = -b \sum_{k=1}^N \left([A_{bitensor}]_k - [\hat{A}]_k \right) \left([A_{tissue}]_k - A_{water} \right) \quad (25)$$

Having defined all the incremental terms, the evolution equations are given by 5 and 6.

8 Tests and Results

8.1 Synthetic dataset

To test the algorithm, a synthetic dataset was used. To create the synthetic dataset, multi-shell fwDTI was performed on the *cenir_multib* dataset provided by dipy. The free water map and diffusion measures were extracted and used in the bi-tensor model in equation 1 to simulate a single-shell dataset with $b = 1000$ along 32 non colinear directions, the mean of the original $S0$ images was added as the new b_0 volume. Rician noise was also added. A slice of the synthetic dataset is shown below.

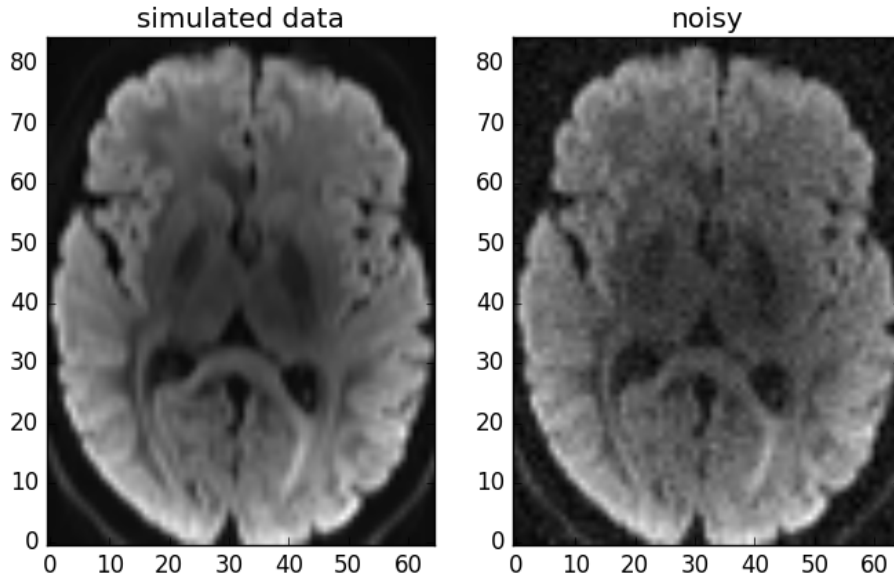


Figure 2: Synthetic dataset, $b = 1000$.

To initialize the tissue volume fraction, the $S_t = 2000$ and $S_w = 9000$ were handpicked from the S_0 image (maybe in the future, an automatic way to set these values may be implemented). To set f_{min} and f_{max} , $\lambda_{min} = 0.1 \times 10^{-3}$ and $\lambda_{max} = 1.1 \times 10^{-3} \text{mm s}^{-2}$. The following initialization was obtained for the free water (note that $fw = 1 - f$, f being from equation 7):

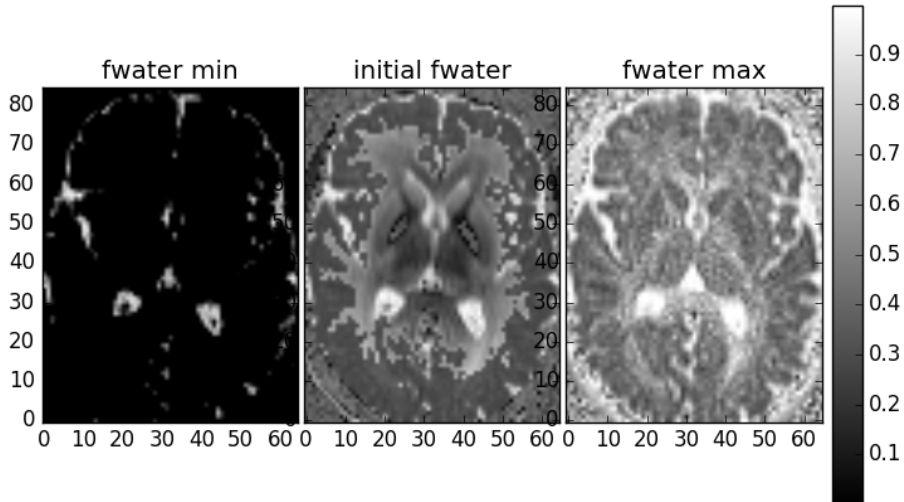


Figure 3: Initialization of fw_{min} , fw_0 and fw_{max}

Note that $fw_{min} = 0$ for most voxels. Also, many voxels in the fw_0 image

were corrected because the initial guess (given by 7) missed the $[fw_{min}, fw_{max}]$ interval, hence the pixelated effect.

For some reason, using equation 8 did not provide a good initialization for the diffusion components. Instead, the following quantity was used to apply regular DTI:

$$S_t = S_k - S_0(1 - f_0)A_{water}$$

This provided a "smoother" initial guess for the diffusion tensor.

The main difficulty to get this algorithm to work is to keep the Iwasawa coordinates $X_1...X_6$ in "well behaved" values. Since this algorithm deals with data in the order of 10^{-3} and the finite differences generate even smaller values, it is hard to control the γ metric, which tends to explode in some voxels. The choice of time step dt is also crucial, in the work of (Pasternak et al., 2009), the time step used is not specified.

To get this algorithm to work, a rescaling was done to the b values, e.g., $b = 1000$ becomes $b = 1$, and the rescaled g_{tab} is provided as input to the algorithm, consequentially, the constants D_{iso} , λ_{min} , λ_{max} also need to be rescaled to the order of units. By doing this, it was much easier to discover the ideal dt and the limits to clip the Iwasawa parameters along the iterations. With $dt = 0.0006$ and 300 iterations, the following results were obtained:

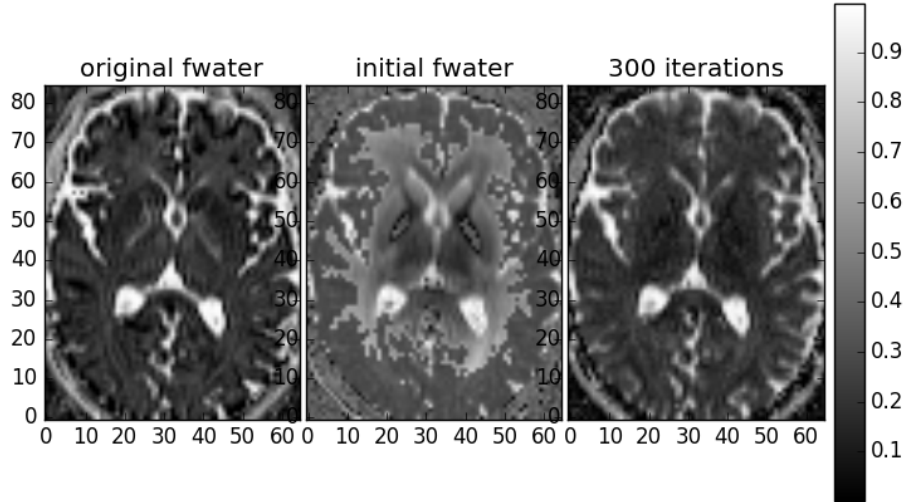


Figure 4: Free water map obtained with 300 iterations, $dt = 0.0006$.

The free water map obtained (right) appears to be close to the original map (left) estimated with the multi-shell fwDTI.

To compare the diffusion components, the FA maps were extracted:

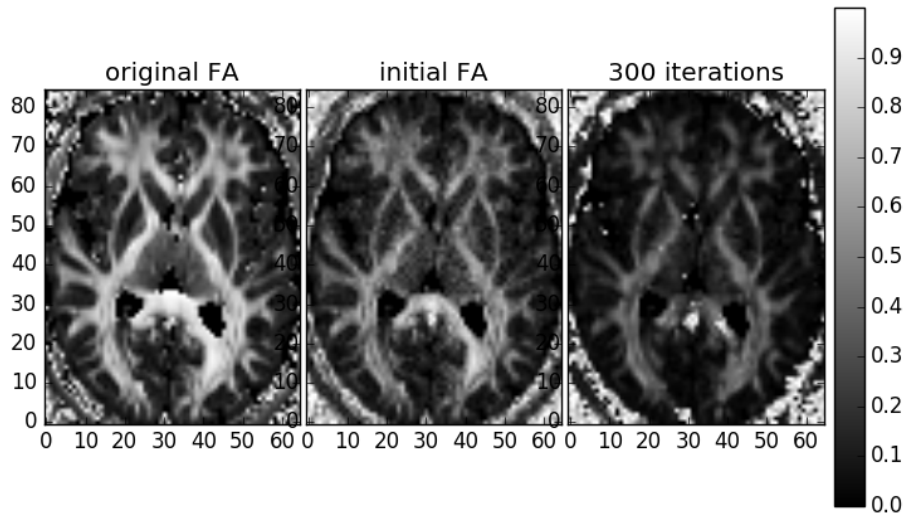


Figure 5: FA map obtained with 300 iterations, $dt = 0.0006$.

The final FA map (right) seems smoother than the initial guess (center), although the values look underestimated in comparison to the original (left), which is brighter (is this a bug?). Also, some problematic bright voxels appear in the final map, which is due to the exploding γ metric.

8.2 Real dataset

A small test was also performed on the *stanford_hardi* dataset, also provided by dipy, $S_t = 300$ and $S_w = 3000$ were used, with $dt = 0.0006$ and 100 iterations.

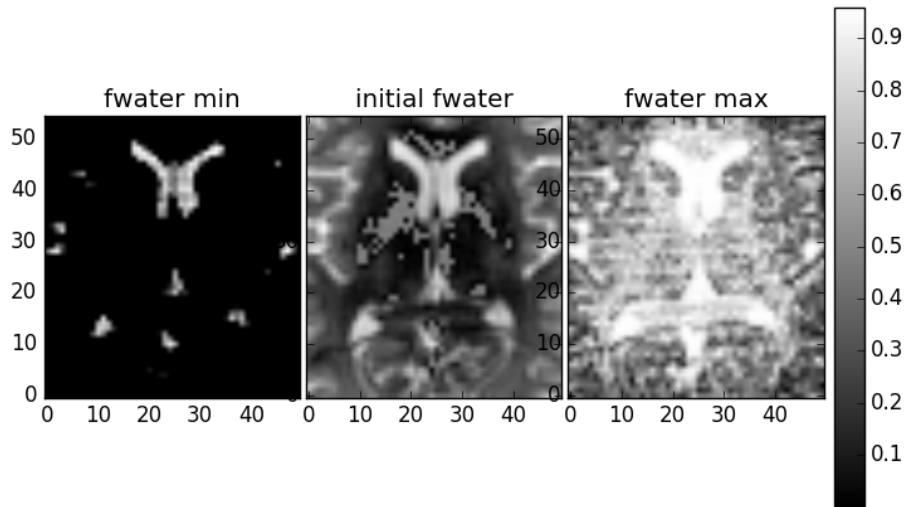


Figure 6: Initialization of $f_{w_{min}}$, f_{w_0} and $f_{w_{max}}$

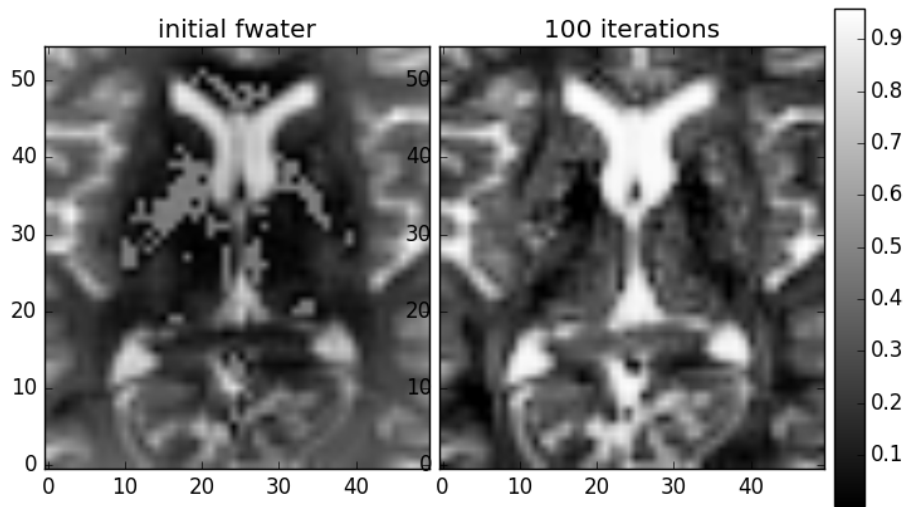


Figure 7: Free water map obtained for stanford dataset, with 100 iterations, $dt = 0.0006$

For this dataset, it is not possible to compare with an original free water map, since the fwDTI of dipy does not work with single-shell data.

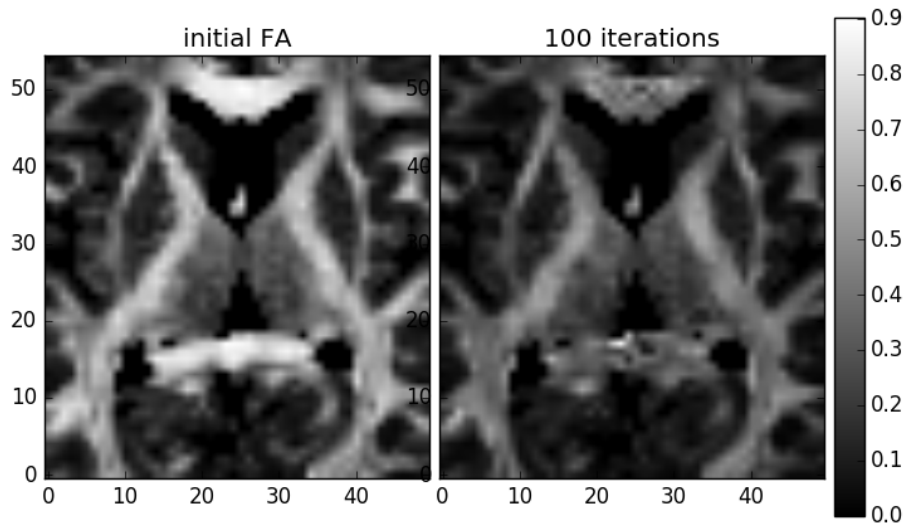


Figure 8: FA map obtained for stanford dataset, with 100 iterations, $dt = 0.0006$

Again, the final FA image is darker than the original.

Finally, the $\sqrt{|\gamma|}$ maps are presented:

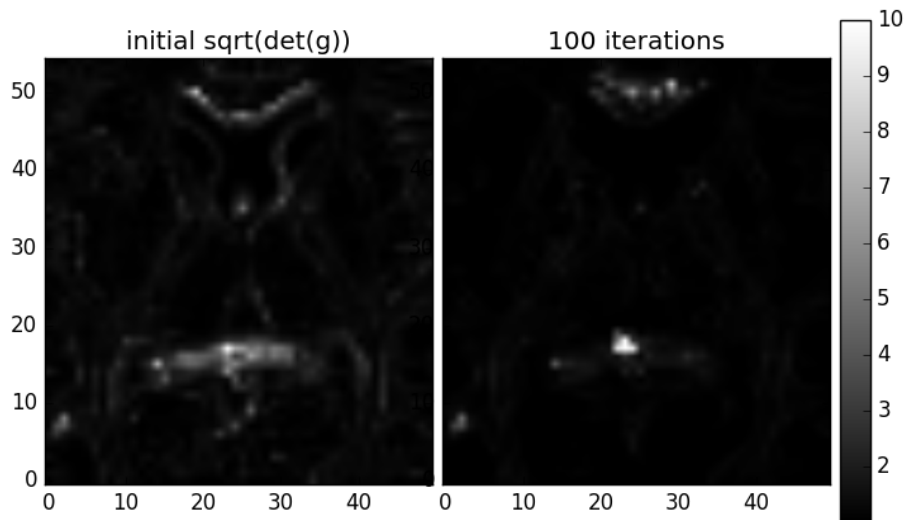


Figure 9: $\sqrt{|\gamma|}$ map obtained for stanford dataset, with 100 iterations, $dt = 0.0006$

As seen in the figure above, the metric seems to be more intense on the contours of the image, which supports the idea that γ is supposed to regularize while maintaining important edges. As mentioned before, as the number of iterations increases, the metric tends to explode (bright spot on the right image).

9 Conclusions

Further testing with real datasets needs to be done to assess the algorithm. Some modifications need to be done to increase efficiency. It would also be good to come up with some strategies to avoid overflows and the exploding γ metric.

References

- Gur, Y., Pasternak, O., & Sochen, N. (2009). Fast gl (n)-invariant framework for tensors regularization. *International journal of computer vision*, 85(3), 211.
- Pasternak, O., Sochen, N., Gur, Y., Intrator, N., & Assaf, Y. (2009). Free water elimination and mapping from diffusion mri. *Magnetic resonance in medicine*, 62(3), 717–730.