

# Numerical Schemes for Linear and Non-linear Enhancement of DW-MRI

Eric J. Creusen<sup>1,2</sup>, Remco Duits<sup>1,2</sup>, and Tom C.J. Dela Haije<sup>2</sup>

<sup>1</sup> Eindhoven University of Technology, The Netherlands  
Department of Mathematics and Computer Science

<sup>2</sup> Department of Biomedical Engineering  
{E.J.Creusen,R.Duits}@tue.nl, T.C.J.Dela.Haije@student.tue.nl

**Abstract.** We consider left-invariant diffusion processes on DTI data by embedding the data into the space  $\mathbb{R}^3 \times S^2$  of 3D positions and orientations. We then define and solve the diffusion equation in a moving frame of reference defined using left-invariant derivatives. The diffusion process is made adaptive to the data in order to do Perona-Malik-like edge preserving smoothing, which is necessary to handle fiber structures near regions of large isotropic diffusion such as the ventricles of the brain. The corresponding partial differential systems are solved using finite difference stencils. We include experiments both on synthetic data and on DTI-images of the brain.

**Keywords:** DTI, DW-MRI, scale spaces, Lie groups, adaptive diffusion, Perona-Malik diffusion.

## 1 Introduction

Diffusion-Weighted Magnetic Resonance Imaging (DW-MRI) are MRI techniques for non-invasively measuring local water diffusion inside tissue. The water diffusion profiles of the imaged area allow inference of the underlying tissue structure. For instance in the brain, diffusion is less constrained parallel to nerve fibers than perpendicular to them, and so the water diffusion gives information about the fiber structures present. This allows for the extraction of clinical information concerning biological fiber structures from DW-MRI scans.

The diffusion of water molecules in tissue over some time interval  $t$  can be described by a diffusion propagator which is the probability density function  $\mathbf{y} \mapsto p_t(Y_t = \mathbf{y} | Y_0 = \mathbf{y}_0)$  of finding a water molecule at time  $t \geq 0$  and at position  $\mathbf{y}$  given that it started at  $\mathbf{y}_0$  on  $t = 0$ . Here the family of random variables  $(Y_t)_{t \geq 0}$  describes the distribution of water molecules over time. The function  $p_t$  can be directly related to MRI signal attenuation of diffusion weighted image sequences and so can be estimated given enough measurements. The exact methods to do this are described by e.g. Alexander [1].

Diffusion Tensor Imaging (DTI), introduced by Basser et al. [2] assumes that  $p_t$  can be described for each voxel by an anisotropic Gaussian function, i.e.

$$p_t(Y_t = \mathbf{y} | Y_0 = \mathbf{y}_0) = \frac{1}{\sqrt{(4\pi t)^3 \det(D(\mathbf{y}_0))}} \exp\left(\frac{-(\mathbf{y} - \mathbf{y}_0)^T D(\mathbf{y}_0)^{-1} (\mathbf{y} - \mathbf{y}_0)}{4t}\right),$$

where  $D$  is a tensor field of  $3 \times 3$  positive definite symmetric tensors that each describe the local Gaussian diffusion process. The tensors contain 6 parameters for each voxel, which means the tensor field requires at least 6 DW-MRI images.

The drawback of approximating  $p_t$  with an anisotropic Gaussian function is that it is only able to estimate one preferred direction per voxel. However, if more complex structures such as crossing, kissing or diverging fibers are present the Gaussian assumption fails, as was demonstrated by Alexander et al. [1]. In practice though, large areas of the brain can be approximated well with DTI tensors and in the regions where complex fiber structures are present the diffusion profile can be inferred by taking contextual information into consideration [12].

Since DTI tensors cannot contain information regarding crossings the DTI data needs to be represented in a form that does allow crossing fiber structures. A representation that suits these demands can be obtained by viewing a DTI image as a probability function on positions and orientation:  $U : \mathbb{R}^3 \times S^2 \rightarrow \mathbb{R}^+$ , where  $S^2 = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\| = 1\}$  denotes the 2-sphere and where for each position  $\mathbf{y}$  and orientation  $\mathbf{n}$ ,  $U(\mathbf{y}, \mathbf{n})$  gives the probability density that a water molecule starts at  $\mathbf{y}$  and travels in direction  $\mathbf{n}$ . Using the Gaussian assumption this distribution is given by

$$U(\mathbf{y}, \mathbf{n}) = \frac{3}{4\pi \int_{\Omega} \text{trace}\{D(\mathbf{y}')\} d\mathbf{y}'} \mathbf{n}^T D(\mathbf{y}) \mathbf{n}, \mathbf{y} \in \mathbb{R}^3, \mathbf{n} \in S^2.$$

Such functions on position and orientation are then visualized by the surfaces  $S_{\mu}(U)(\mathbf{x}) = \{\mathbf{x} + \mu U(\mathbf{x}, \mathbf{n}) \mathbf{n} \mid \mathbf{n} \in S^2\} \subset \mathbb{R}^3$ , which are called glyphs. A figure is generated by visualizing all these surfaces for varying  $\mathbf{x}$  and with a suitable value for  $\mu > 0$  that determines the size of the glyphs. Note that for DTI data a different visualization based on ellipsoids is commonly used, which isn't suitable to visualize crossing fibers.

To reduce noise and to infer information about fiber crossings contextual information can be used. This enhancement is useful both for visualization purposes and as a preprocessing step for other algorithms such as fiber tracking algorithms, which may have difficulty in noisy or incoherent regions. This enhancement, done through linear and nonlinear adaptive diffusion processes, is the main focus of this paper. Special attention is given to the implementation of these algorithms through the use of finite difference schemes.

### 1.1 The Euclidean Motion Group $SE(3)$

A function on  $\mathbb{R}^3 \times S^2$  can also be seen as a function embedded in the *Euclidean motion group*  $SE(3) = \mathbb{R}^3 \rtimes SO(3)$ , where  $SO(3)$  represents the (noncommutative) group of 3D rotations defined as a matrix group by

$$SO(3) = \{\mathbf{R} \mid \mathbf{R} \in \mathbb{R}^{3 \times 3}, \mathbf{R}^T = \mathbf{R}^{-1}, \det(\mathbf{R}) = 1\}.$$

Expressed in Euler angles, this becomes

$$\mathbf{R}_{(\alpha, \beta, \gamma)} = \mathbf{R}_{\gamma}^{e_x} \mathbf{R}_{\beta}^{e_y} \mathbf{R}_{\alpha}^{e_z}, \quad (1)$$

where  $\mathbf{e}_1 = \mathbf{e}_x$ ,  $\mathbf{e}_2 = \mathbf{e}_y$  and  $\mathbf{e}_3 = \mathbf{e}_z$  are the unit vectors in the coordinate axes and  $\mathbf{R}_\alpha^{\mathbf{e}_i}$  denotes a counterclockwise rotation of  $\alpha$  around vector  $\mathbf{e}_i$ . Here, an Euler angle parametrization is used that has a discontinuity at  $\mathbf{n} = (\pm 1, 0, 0)$ , so that the tangent space of  $SE(3)$  is well defined at the unity element  $(\mathbf{0}, \mathbf{I})$ .

For  $g, g' \in SE(3)$  the group product and inverse are given by

$$\begin{aligned} gg' &= (\mathbf{x}, \mathbf{R})(\mathbf{x}', \mathbf{R}') = (\mathbf{x} + \mathbf{R} \cdot \mathbf{x}', \mathbf{R}\mathbf{R}') \\ g^{-1} &= (\mathbf{x}, \mathbf{R})^{-1} = (-\mathbf{R}^{-1}\mathbf{x}, \mathbf{R}^{-1}). \end{aligned}$$

To get correspondence between  $SO(3)$  and  $S^2$ , we introduce equivalence classes on  $SO(3)$ . Two group elements  $g, h \in SO(3)$  are equivalent if  $g^{-1}h = \mathbf{R}_\alpha^{\mathbf{e}_z}$  for some angle  $\alpha \in [0, 2\pi)$ . This equivalence relation induces sections of equivalent group members, called the *left cosets* of  $SO(3)$ . If we associate  $SO(2)$  with rotations around the z-axis, then formally we can use this equivalence to write  $S^2 \equiv SO(3)/SO(2)$  to denote these left cosets.

If we extend this equivalence relation to  $SE(3)$ , i.e.  $g, h \in SE(3)$ ,  $g$  is equivalent to  $h$  if  $g^{-1}h = (\mathbf{0}, \mathbf{R}_\alpha^{\mathbf{e}_z})$ , we obtain the left coset of  $SE(3)$  which equals the space of positions and orientations. To stress that this space has been embedded in  $SE(3)$  and to stress the induced (quotient)group structure we write the space of positions and orientations as  $\mathbb{R}^3 \rtimes S^2 := (\mathbb{R}^3 \rtimes SO(3))/(\{\mathbf{0}\} \times SO(2))$ .

Now, we can express any function on position and orientation  $U : \mathbb{R}^3 \rtimes S^2 \rightarrow \mathbb{R}$  with an equivalent function on  $SE(3)$   $\tilde{U} : \mathbb{R}^3 \rtimes SO(3) \rightarrow \mathbb{R}$  by solving for  $(\mathbf{x}, \mathbf{n}) = (\mathbf{x}, \mathbf{R}_n \mathbf{e}_z)$ , where  $\mathbf{R}_n$  is any rotation matrix that maps  $\mathbf{e}_z$  to  $\mathbf{n}$ .

Every group element from  $SE(3)$  can be associated with a representation, which is nothing else than an action that translates and rotates a function. The left- and right-regular representations on  $\mathbb{L}_2(SE(3))$  are given by

$$\begin{aligned} (\mathcal{L}_g \circ U)(h) &= U(g^{-1}h), \quad g, h \in SE(3), \quad U \in \mathbb{L}_2(SE(3)) \\ (\mathcal{R}_g \circ U)(h) &= U(hg). \end{aligned} \tag{2}$$

Duits and Franken [8,9] demonstrated that every reasonable linear operation on functions on  $SE(3)$  must be left-invariant by showing that the orientation marginal  $\int_{S^2} U(\mathbf{y}, \mathbf{n}) d\sigma(\mathbf{n})$  commutes with rotations and translations only under such operations, which explains our choice for left-invariant processes in this paper. Formally an operator  $\Phi : \mathbb{L}_2(SE(3)) \rightarrow \mathbb{L}_2(SE(3))$  is left invariant iff

$$\forall g \in SE(3) : (\mathcal{L}_g \circ \Phi \circ U) = (\Phi \circ \mathcal{L}_g \circ U).$$

The right regular representation is left-invariant and can be used to generate left-invariant derivatives, as is shown in the next section. It should be noted that because of the non commutative structure of  $SE(3)$  the left regular representation itself is *not* left-invariant.

## 1.2 Left-Invariant Derivatives

By viewing functions on  $\mathbb{R}^3 \rtimes S^2$  as probability density functions of oriented particles, it becomes a natural idea to describe these particles in a moving coordinate system. This is done by attaching a coordinate system to each point

$(\mathbf{x}, \mathbf{R}) \in SE(3)$  such that one of the spatial axes points in the direction of  $\mathbf{n} = \mathbf{R}_n \mathbf{e}_z$ . In this section we will introduce diffusion equations for these oriented particles, and for these processes this coordinate system is the natural choice to easily differentiate between motion forward, sideways and rotations. We can obtain such a coordinate system by starting at the identity element  $(\mathbf{0}, \mathbf{I})$  of  $SE(3)$ , which corresponds to  $(\mathbf{0}, \mathbf{e}_z)$  and attaching a suitable coordinate system using Euler angles. We express a basis of tangent vectors at the unity element by

$$A_1 = \partial_x, A_2 = \partial_y, A_3 = \partial_z, A_4 = \partial_\gamma, A_5 = \partial_\beta, A_6 = \partial_\alpha,$$

where we use the coordinate system in the parametrization of  $SE(3):(\mathbf{x}, R) = (x, y, z, R_{(\alpha, \beta, \gamma)})$  (see Eq. (1)). Here  $A_i$  can be viewed both as tangent vectors and as local differential operators.

We construct a moving frame of reference attached to fibers in the space  $\mathbb{R}^3 \times S^2$  by using the derivative of the right-regular representation  $\mathcal{R}$ :

$$\mathcal{A}_i|_g U = (d\mathcal{R}(A_i)U)(g) = \lim_{t \downarrow 0} \frac{U(g e^{tA_i}) - U(g)}{t}, \quad i = 1, 2, 3, 4, 5, 6, \quad (3)$$

where  $\mathcal{R}$  is defined by Eq. (2) and  $e^{tA_i}$  is the exponential map in  $SE(3)$  [8], which can be seen as the group element obtained by traveling distance  $t$  in the  $A_i$  direction from the identity element. We note that  $\mathcal{A}_6 U(\mathbf{y}, \mathbf{n}) = 0$ , because  $U(\mathbf{y}, \mathbf{n})$  is constant within equivalence classes and that  $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_4$  and  $\mathcal{A}_5$  are defined on  $SE(3)$  and not on  $\mathbb{R}^3 \times S^2$ . We therefor use combinations of these operators that *are* well-defined on  $\mathbb{R}^3 \times S^2$  in the diffusion generator (see section 1.3).

Analytical formulas for these left-invariant derivatives, expressed in charts of Euler angles can be found in [8] where they are used to analytically approximate Green's functions of convection diffusion processes. Here, we focus on the numerical aspects, and instead give only the finite difference schemes (section 2) since they do not suffer from the discontinuities of the Euler angle parametrization.

### 1.3 Convection-Diffusion Processes

The left-invariant derivatives given in the previous section can be used to write the equations for diffusion processes on  $SE(3)$  [8], which can remove noise while preserving complex structures such as crossings and junctions[12].

The general convection-diffusion equation with diffusion matrix  $\mathbf{D}$  and convection parameters  $\mathbf{a}$  is given by:

$$\begin{cases} \partial_t W(\mathbf{y}, \mathbf{n}, t) = Q^{\mathbf{D}, \mathbf{a}}(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_5)W(\mathbf{y}, \mathbf{n}, t) \\ W(\mathbf{y}, \mathbf{n}, 0) = U(\mathbf{y}, \mathbf{n}) \end{cases} \quad (4)$$

where the convection-diffusion generator  $Q^{\mathbf{D}, \mathbf{a}}$  is given by

$$Q^{\mathbf{D}, \mathbf{a}}(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_5) = \sum_{i=1}^5 \left( -a_i \mathcal{A}_i + \sum_{j=1}^5 \mathcal{A}_i D_{ij} \mathcal{A}_j \right) \quad (5)$$

and  $a_i$  are convection parameters and  $D_{ij}$  diffusion coefficients. In this paper,  $a_i = 0$  for all  $i = 1, \dots, 6$  because only pure diffusion processes are studied, but other processes, like contour completion [4], can be obtained by also including convection terms. In the linear case,  $a_i$  and  $D_{ij}$  are chosen constant and the solution to these evolution equations can be obtained by an  $SE(3)$ -convolution of the initial data with the process's Green's function [8,9,5] or by using finite difference methods.  $SE(3)$ -convolutions (see [8]) are generally computationally more expensive than finite difference stencils and they can not handle adaptive schemes, so finite difference schemes are used exclusively in this paper.

## 2 Finite Difference Schemes for $\mathbb{R}^3 \rtimes S^2$ Diffusion

To approximate the required left-invariant derivatives of the evolution equations of Eq. (4), we use finite difference approximations [8] of Eq. (3). These derivatives are approximated in the usual way, with the (conceptually) small difference that the steps are taken in the  $\mathcal{A}_i$  direction rather than the  $\mathbf{e}_i$  direction. The forward finite difference approximation of the left-invariant derivatives are given by

$$\begin{aligned} \mathcal{A}_1^f U(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y}+h R_{\mathbf{n}} \mathbf{e}_x, \mathbf{n}) - U(\mathbf{y}, \mathbf{n})}{h}, & \mathcal{A}_4^f U(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y}, R_{\mathbf{n}} R_{h_a}^{\mathbf{e}_x} \mathbf{e}_z) - U(\mathbf{y}, \mathbf{n})}{h_a}, \\ \mathcal{A}_2^f U(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y}+h R_{\mathbf{n}} \mathbf{e}_y, \mathbf{n}) - U(\mathbf{y}, \mathbf{n})}{h}, & \mathcal{A}_5^f U(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y}, R_{\mathbf{n}} R_{h_a}^{\mathbf{e}_y} \mathbf{e}_z) - U(\mathbf{y}, \mathbf{n})}{h_a}, \\ \mathcal{A}_3^f U(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y}+h R_{\mathbf{n}} \mathbf{e}_z, \mathbf{n}) - U(\mathbf{y}, \mathbf{n})}{h}, \end{aligned} \quad (6)$$

where  $h$  is the spatial stepsize and  $h_a$  the angular step size in radians.

Analogously, the backward and central finite difference approximations can be obtained. For example:

$$\mathcal{A}_3^b U(\mathbf{y}, \mathbf{n}) = \frac{U(\mathbf{y}, \mathbf{n}) - U(\mathbf{y} - h R_{\mathbf{n}} \mathbf{e}_z, \mathbf{n})}{h}, \quad \mathcal{A}_4^b U(\mathbf{y}, \mathbf{n}) = \frac{U(\mathbf{y}, \mathbf{n}) - U(\mathbf{y}, R_{\mathbf{n}} R_{-h_a}^{\mathbf{e}_x} \mathbf{e}_z)}{h_a},$$

and

$$\begin{aligned} \mathcal{A}_3^c U(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y} + h R_{\mathbf{n}} \mathbf{e}_z, \mathbf{n}) - U(\mathbf{y} - h R_{\mathbf{n}} \mathbf{e}_z, \mathbf{n})}{2h}, \\ \mathcal{A}_4^c U(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y}, R_{\mathbf{n}} R_{h_a}^{\mathbf{e}_x} \mathbf{e}_z) - U(\mathbf{y}, R_{\mathbf{n}} R_{-h_a}^{\mathbf{e}_x} \mathbf{e}_z)}{2h_a}. \end{aligned} \quad (7)$$

We take second order centered finite differences by applying the discrete operators in the righthand side of Eq. (7) twice (where we replaced  $2h \mapsto h$ ), e.g. we have for  $p = 1, 2, 3$ :

$$\begin{aligned} ((\mathcal{A}_p^c)^2 U)(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y} + h R_{\mathbf{n}} \mathbf{e}_p, \mathbf{n}) - 2U(\mathbf{y}, \mathbf{n}) + U(\mathbf{y} - h R_{\mathbf{n}} \mathbf{e}_p, \mathbf{n})}{h^2}, \\ ((\mathcal{A}_{p+3}^c)^2 U)(\mathbf{y}, \mathbf{n}) &= \frac{U(\mathbf{y}, R_{\mathbf{n}} R_{\mathbf{e}_p, h_a} \mathbf{e}_z) - 2U(\mathbf{y}, \mathbf{n}) + U(\mathbf{y}, R_{\mathbf{n}} R_{\mathbf{e}_p, -h_a} \mathbf{e}_z)}{h_a^2}. \end{aligned}$$

### 2.1 Efficient Computation of Left-invariant Derivatives

So far, all approximations assume  $U : \mathbb{R}^3 \rtimes S^2 \rightarrow \mathbb{R}^+$  to be continuously differentiable. In practice we have discretized functions  $U[i, j, k, \mathbf{n}_l]$ , where  $i, j$  and

$k$  enumerate the discrete spatial grid and  $\mathbf{n}_l$  is an orientation from a tessellation of the sphere enumerated by  $l \leq N_o$ . The tessellation used in this paper is obtained by taking an icosahedron and regularly subdividing each face into 16 triangles before projecting the vertices back to the sphere. Every vertex of this shape becomes a sampling orientation and thus  $N_o = 162$ .

Because of this sampling, interpolation is necessary to approximate the (left-invariant) derivatives. Spatially, any regular 3D interpolation scheme such as linear interpolation or spline interpolation can be used. Since the approximations in Eq. (6) are only first order accurate, we use linear interpolation.

Since the three spatial derivatives only require neighboring samples with the same  $\mathbf{n}$ , they can be efficiently computed through a regular  $\mathbb{R}^3$  convolution or correlation for each orientation separately.

$$\mathcal{A}_p U[i, j, k, \mathbf{n}_l] \approx (\mathbf{M}_l^p \star U[\cdot, \cdot, \cdot, \mathbf{n}_l])[i, j, k] \quad p = 1, 2, 3,$$

where  $\star$  denotes the discrete spatial correlation and  $\mathbf{M}_l^p$  can be obtained by linear interpolation from the finite difference stencils Eq. (6). For example, in the case of forward finite difference stencils

$$\mathbf{M}_l^p[i, j, k] = \frac{1}{2h}(w_{l,p}^1[i, j, k] - w_{l,p}^2[i, j, k]),$$

$$\text{with } w_{l,p}^1[i, j, k] = N_{\mathbf{y}_l^p}[i, j, k] \text{ and } w_{l,p}^2[i, j, k] = \delta_{i0}\delta_{j0}\delta_{k0},$$

where  $\mathbf{y}_l^p = hR_{\mathbf{n}_l}e_p$  and  $N_{\mathbf{y}_l^p}[i, j, k]$  is the interpolation matrix required to interpolate point  $\mathbf{y}_l^p$ . Assuming cubic voxels of size 1,  $|\mathbf{y}_l^p| < 1$ , and using linear interpolation  $N_{\mathbf{y}_l^p}[i, j, k]$  is given by

$$N_{\mathbf{y}}[i, j, k] = \prod_{m=1}^3 v_{(\mathbf{y}_{p,l})^m}[\mathbf{z}^m], \quad \mathbf{z} = (i, j, k) \quad i, j, k \in \{-1, 0, 1\},$$

$$v_a[b] = \begin{cases} 1 - |a| & \text{if } b = 0 \\ H(ab)|a| & \text{if } b \in \{-1, 1\} \\ 0 & \text{otherwise.} \end{cases}$$

with heaviside function  $H(u)$  and where  $(\mathbf{y}_{p,l})^m$  and  $\mathbf{z}^m$  denote the  $m$ -th component of vectors  $\mathbf{y}_{p,l}$  and  $\mathbf{z}$ .

For angular interpolation, either linear interpolation or spherical harmonics can be used. Spherical harmonics were used in previous work by Franken et al. [10]. In terms of stability in the diffusion process, both perform equally well (see section 2.2, Theorem 1), but Franken had to add an angular diffusion term  $t_{reg}$  which in practice was very sensitive: when set too large the data becomes too isotropic (destroying fiber structures) and when set too small the algorithm becomes unstable. As we will show next, linear interpolation is also computationally cheaper.

The angular derivatives only require samples of neighbors with the same  $\mathbf{y}$  and can therefore be computed by a matrix multiplication for each point  $\mathbf{y}$ :

$$\mathcal{A}_{p+3}^f U[i, j, k, \mathbf{n}_l] \approx \frac{1}{h} \sum_{l'=1}^{N_o} \mathbf{M}_{l'}^p U[i, j, k, \mathbf{n}_{l'}] - \frac{1}{h} U[i, j, k, \mathbf{n}_l], \quad (8)$$

where  $\mathbf{M}_{l'l'}$  is the interpolation matrix to interpolate  $\mathbf{n}_{p,l} = R_{\mathbf{n}_l} R_{\mathbf{e}_p, h_a} \mathbf{e}_3$  and is given by

$$M_{l'l'} = \begin{cases} 1 - \sum_{\mathbf{n}_j \in A_{p,l}} (\mathbf{n}_{p,l} - \mathbf{n}_{l'}) \cdot (\mathbf{n}_j - \mathbf{n}_{l'}) & \text{if } \mathbf{n}_{l'} \in A_{p,l} \\ 0 & \text{otherwise,} \end{cases}$$

where  $A_{p,l}$  is the triangle that contains point  $\mathbf{n}_{p,l}$ .  $A_{p,l}$  is sparse due to the linear interpolation which enables Eq. (8) to be computed cheaply. If  $M_{l'l'}$  is created using spherical harmonics then it becomes a full matrix and thus Eq. (8) is much more expensive to calculate.

## 2.2 Numerical Contour Enhancement

The *contour enhancement* process on  $\mathbb{R}^3 \times S^2$  can be obtained from Eq. (4) by setting  $a_i = 0$  (no convection),  $D_{33} \geq 0$ ,  $D_{44} = D_{55} \geq 0$  and other diffusion coefficients  $D_{ij}$  are set to zero. These settings yield the following evolution equation

$$\begin{cases} \partial_t W(\mathbf{y}, \mathbf{n}, t) = (D_{33}(\mathcal{A}_3)^2 + D_{44}((\mathcal{A}_4)^2 + (\mathcal{A}_5)^2)) W(\mathbf{y}, \mathbf{n}, t) \\ W(\mathbf{y}, \mathbf{n}, 0) = U(\mathbf{y}, \mathbf{n}) \end{cases} \quad (9)$$

This process can intuitively be understood as a description of the Brownian motion of oriented particles both in space (diffusion in direction  $\mathbf{n}$ ) and angular (changing direction) [8]. The simulation of this PDE is done by taking standard centered second order finite differences according to Eq. (2), and using a forward Euler scheme for the time discretization:

$$\begin{cases} W(\mathbf{y}, \mathbf{n}, t + \Delta t) = W(\mathbf{y}, \mathbf{n}, t) + \Delta t (D_{33}(\mathcal{A}_3^c)^2 + D_{44}((\mathcal{A}_4^c)^2 + (\mathcal{A}_5^c)^2)) W(\mathbf{y}, \mathbf{n}, t) \\ W(\mathbf{y}, \mathbf{n}, 0) = U(\mathbf{y}, \mathbf{n}) \end{cases}$$

Of these parameters,  $D_{44}$  and simulation time  $t$  are most important.  $D_{33}$  may be set to 1, as changing  $D_{33}$  is equivalent to scaling  $D_{44}$  and  $t$ , while  $\Delta t$  needs only be sufficiently small for the algorithm to remain stable and accurate.

**Theorem 1.** *The stability bound for the Euler forward finite difference scheme of the evolution described by Eq. (9) using the interpolation described in section 2.1 is given by*

$$\Delta t \leq \frac{1}{\frac{(4D_{11} + 2D_{33})}{h^2} + \frac{4D_{44}}{h_a^2}} \quad (10)$$

when using linear interpolation for the angular derivatives and by

$$\begin{aligned} \Delta t &\leq \frac{1}{\frac{4D_{11} + 2D_{33}}{h^2} + D_{44} \frac{L(L+1)}{2e^{t_{reg} L(L+1)}}} && \text{if } t_{reg} \cdot L(L+1) \leq 1 \\ \Delta t &\leq \frac{1}{\frac{4D_{11} + 2D_{33}}{h^2} + D_{44} \frac{1}{2e^{t_{reg}}}} && \text{if } t_{reg} \cdot L(L+1) > 1 \end{aligned} \quad (11)$$

when using spherical harmonics.

for proof see [4, Section 2.2.1] and [7, Appendix B].

In terms of stability, both algorithms can be made equally stable because both have regularizing parameters ( $h_a$  for linear interpolation and  $t_{reg}$  for spherical harmonics). There are, however other reasons to prefer linear interpolation over spherical harmonics (see section 2.1).

### 3 Perona-Malik Diffusion on $\mathbb{R}^3 \times S^2$

Linear contour enhancement has the disadvantage that it performs diffusion across areas where the gradient is very large. In particular, the neural tracts of the brain are sometimes located near the ventricles of the brain. These ventricles are structures that contain cerebrospinal fluid which shows up in DTI as unrestricted, isotropic glyphs much larger in magnitude than the restricted, anisotropic glyphs of the neural tracts. It is undesirable that these large isotropic diffusion profiles start to interfere with the oriented structures of the neural tracts when we apply a diffusion scheme, because they are likely to destroy fiber structures. A Perona-Malik [11] type scheme for diffusion can separate these two regions, and apply the diffusion within the neural tracts and within the ventricles, but prevents transport from one to the other.

Our approach is similar to recent work by Burgeth et al. [3] who used adaptive, edge preserving diffusion on the DTI tensor components separately. The difference is that here the diffusion considers both positions and orientations in the domain and therefor separates two crossing fibers in the domain so that it is better equipped to handle crossing structures.

We test the algorithm on a synthetic test image consisting of two crossing fibers consisting of oriented glyphs surrounded by isotropic spheres, (see Fig. 1) in which linear diffusion destroys the fiber structure, whereas nonlinear adaptive diffusion both preserves the fiber structures and denoises the entire dataset.

Mutual influence of the anisotropic regions (fibers) and isotropic regions (ventricles) is avoided by replacing the constant diffusivity  $D_{33}$  in 5 by

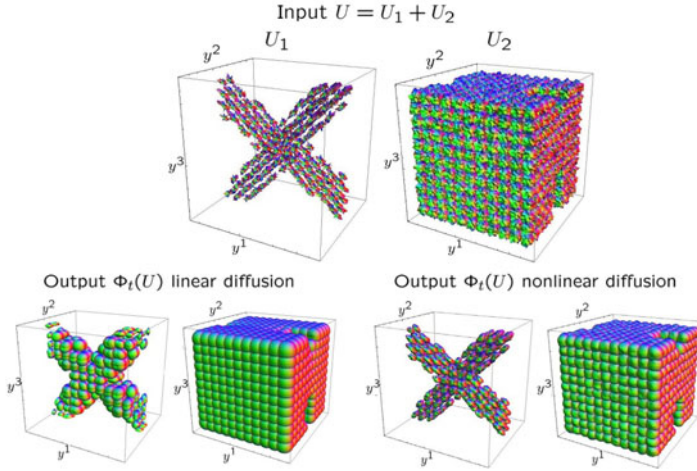
$$\mathcal{A}_3 D_{33} \mathcal{A}_3 \mapsto \mathcal{A}_3 \circ D_{33} e^{-\frac{(\mathcal{A}_3 W(\cdot, t))^2}{K^2}} \circ \mathcal{A}_3. \quad (12)$$

where for  $K \rightarrow \infty$ , linear contour enhancement is obtained. The idea is to set a soft threshold (determined by  $K$ ) on the amount of diffusion in  $\mathcal{A}_3$  direction. Within homogeneous regions one expects  $|\mathcal{A}_3 W(\mathbf{y}, \mathbf{n}, t)|$  to be small, whereas in the transition areas between ventricles and white matter where one needs to block the diffusion process, one expects a large  $|\mathcal{A}_3 W(\mathbf{y}, \mathbf{n}, t)|$ .

To implement this, we propose the following discretization scheme

$$\begin{aligned} \mathcal{A}_3(\tilde{D}_{33}\mathcal{A}_3)W(\mathbf{y}, \mathbf{n}, t) &\approx \frac{\tilde{D}_{33}(\mathbf{y} + \frac{1}{2}\mathbf{h}, \mathbf{n})\mathcal{A}_3W(\mathbf{y} + \frac{1}{2}\mathbf{h}, \mathbf{n}, t)}{h} - \frac{\tilde{D}_{33}(\mathbf{y} - \frac{1}{2}\mathbf{h}, \mathbf{n})\mathcal{A}_3W(\mathbf{y} - \frac{1}{2}\mathbf{h}, \mathbf{n}, t)}{h} \\ \mathcal{A}_3W(\mathbf{y} + \frac{1}{2}\mathbf{h}, \mathbf{n}, t) &\approx \frac{W(\mathbf{y} + \mathbf{h}, \mathbf{n}, t) - W(\mathbf{y}, \mathbf{n}, t)}{h} = \mathcal{A}_3^f W(\mathbf{y}, \mathbf{n}, t) \\ \mathcal{A}_3W(\mathbf{y} - \frac{1}{2}\mathbf{h}, \mathbf{n}, t) &\approx \frac{W(\mathbf{y}, \mathbf{n}, t) - W(\mathbf{y} - \mathbf{h}, \mathbf{n}, t)}{h} = \mathcal{A}_3^b W(\mathbf{y}, \mathbf{n}, t) \end{aligned}$$





**Fig. 1.** Adaptive Perona-Malik diffusion based on the data. Top row: Artificial  $15 \times 15 \times 15 \times 162$  input data that is the sum of a noisy fiber part and a noisy isotropic part. For the sake of visualization, we depict these parts separately. Bottom row left: Output of linear diffusion with  $t = 1$ ,  $D_{33} = 1, D_{44} = 0.04$  and  $\Delta t = 0.01$ . Bottom row right: Output of Perona-Malik adaptive diffusion with  $D_{33} = 1$ ,  $D_{44} = 0.015$ ,  $K = 0.05$ ,  $\Delta t = 0.01$ ,  $t = 1$

combining these three equations leads to

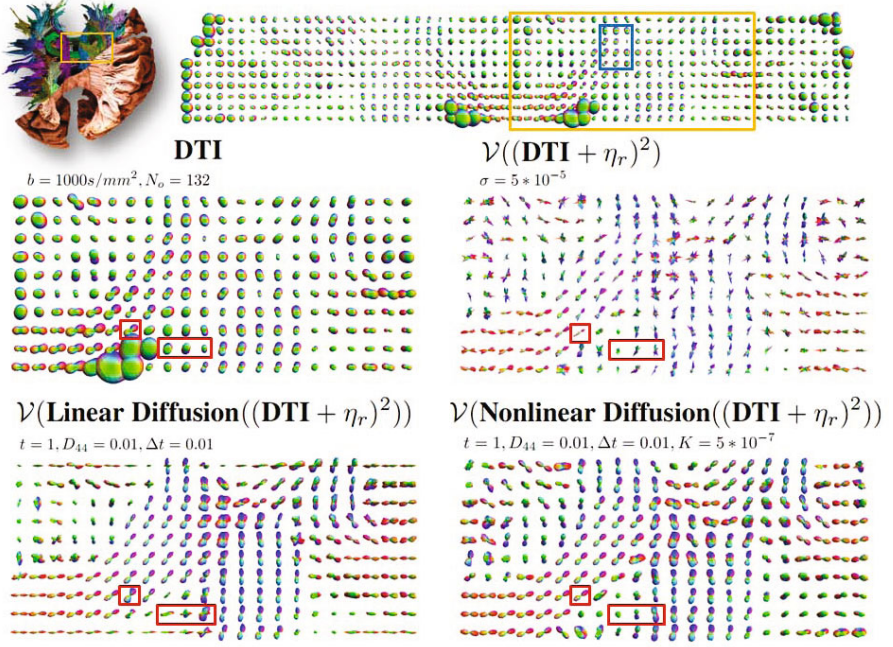
$$\mathcal{A}_3(\tilde{D}_{33}\mathcal{A}_3)W(\mathbf{y}, \mathbf{n}, t) \approx \frac{\tilde{D}_{33}(\mathbf{y} + \frac{1}{2}\mathbf{h}, \mathbf{n})\mathcal{A}_3^f W(\mathbf{y}, \mathbf{n}, t) - \tilde{D}_{33}(\mathbf{y} - \frac{1}{2}\mathbf{h}, \mathbf{n})\mathcal{A}_3^b W(\mathbf{y}, \mathbf{n}, t)}{h}$$

where for notational convenience  $\mathbf{h} = h \mathbf{R}_n \mathbf{e}_z$  and  $\tilde{D}_{33} = D_{33} e^{-\frac{(\max(|\mathcal{A}_3^f W|, |\mathcal{A}_3^b W|))^2}{\kappa^2}}$ . The  $\tilde{D}_{33}$  terms can easily be calculated with linear interpolation. Combined with the finite difference operators of section 2, this give the full discretization scheme.

The discretization scheme for  $\tilde{D}_{33}$  uses  $\max(|\mathcal{A}_3^f W|, |\mathcal{A}_3^b W|)$  because forward and backwards finite difference schemes individually induce shifts near discontinuities while central finite difference schemes sometimes allow diffusion across region boundaries. This happens when fiber voxels have more than one isotropic neighbor, then  $\mathcal{A}_3^c W$  may be close to zero because the stencil does not depend on the center point. Because of the spatial discretization and because every direction  $\mathbf{n}$  is considered, this is very likely to occur in almost all geometries.

## 4 Enhancement of DTI of the Human Brain

To test the algorithm on real data, a DTI brain scan was acquired from a healthy volunteer with 132 gradient directions and a b-value of  $1000 \text{ s/mm}^2$ . Linear contour enhancement (Eq. (9)) as well as Perona-Malik adaptive diffusion (Eq. (12)) was performed on it, as can be seen in Fig. 2.



**Fig. 2.** DTI data of the corpus callosum and corona radiata fibers in a human brain with  $b$ -value  $1000s/mm^2$  and 132 gradient directions on voxels of  $(2mm)^3$ . Top row: A coronal slice of the original data with a region of interest in the yellow square. The region in the blue square is shown for multiple values of  $K$  in Fig. 3. Middle row: The unprocessed region of interest (left) and with added Rician noise ( $\sigma = 5 \cdot 10^{-5}$ , see [4] for definition) and sharpening according to Eq.(13) (right). Bottom row: The result of linear contour enhancement (left) and Perona-Malik diffusion (right). Marked in red are areas in which the ventricles have induced crossing structures in the linear diffusion process.

Prčkovska et al. [12] showed that DTI combined with enhancement techniques can extrapolate crossing information from contextual information. It is interesting to see if such a method can be improved with a Perona-Malik type scheme, especially since the ventricles may make such methods unreliable in those areas.

Since visualization of larger datasets is difficult, only coronal slices through the center of the brain are depicted, where the ventricles are visible as large, isotropic spheres. Because of the relative isotropy of real data, sharpening techniques have to be employed. Squaring the input data is the simplest way to do this (and is used here), but other techniques such as  $\mathbb{R}^3 \times S^2$ -erosions are also an option [6]. For visualization, a min-max-normalization and another sharpening step are used, given by operator

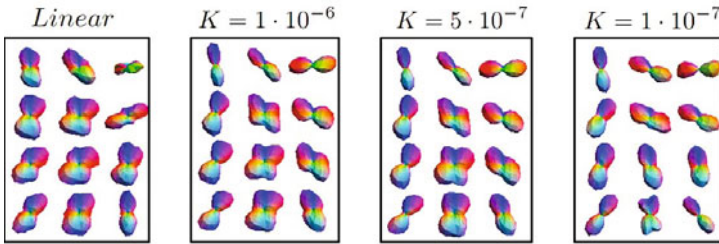
$$\mathcal{V}(U)(\mathbf{y}, \mathbf{n}) = \left( \frac{U(\mathbf{y}, \mathbf{n}) - U_{\min}(\mathbf{y})}{U_{\max}(\mathbf{y}) - U_{\min}(\mathbf{y})} \right)^2, \text{ with } U_{\min}(\mathbf{y}) = \min_{\max} \{U(\mathbf{y}, \mathbf{n}) \mid \mathbf{n} \in S^2\}. \quad (13)$$

From Fig. 2 it can be seen that the Perona-Malik method performs better than linear contour enhancement. The first effect is visible on the boundary of the

data. Linear contour enhancement diffuses signal outside of the boundaries of the image (because of zero padding boundary conditions for the calculation of derivatives), which causes artifacts visible as horizontal structures near the top and bottom edges. The same zero padding ensures a large derivative at these places so Perona-Malik does not suffer from this problem.

The second effect is visible around the ventricles (marked by red in Fig. 2). Linear diffusion shows some crossing structures directly to the right of the ventricles, while the surrounding glyphs do not suggest there should be any crossings there. It also affects the fibers of the corpus callosum to the top left of the ventricles by bending them a bit upwards and away from the ventricles.

Figure 3 shows the effect of parameter  $K$  on the diffusion profile of the area of fiber crossings where it can be seen that setting  $K$  too small leads to a shortcoming of the algorithm to correctly infer crossing information while setting it too large leads to the same result as linear diffusion.



**Fig. 3.** Area with fiber crossings of the corpus callosum and corona radiata for different values of  $K$ . All images created with  $D_{33} = 1$ ,  $D_{44} = 0.01$ ,  $t = 1$  and  $\Delta t = 0.01$

## 5 Conclusion

We have developed an edge-preserving, adaptive Perona-Malik smoothing process using finite difference schemes that can be used to remove high frequency noise and extrapolate fiber crossing information from DTI data by embedding the DTI data into a function on the space of positions and orientations  $\mathbb{R}^3 \times S^2$ . Our experiments have shown that the adaptive diffusion process performs better than linear processes in areas with large isotropic diffusion (such as the ventricles of the brain) since at these areas adaptive diffusivity strongly reduces the interference between isotropic glyphs in the ventricles and anisotropic glyphs of the fibers.

## References

1. Alexander, D.C., Barker, G.J., Arridge, S.R.: Detection and modeling of non gaussian apparent diffusion coefficient profiles in human brain data. *Magnetic Resonance in Medicine* 48, 331–340 (2002)
2. Basser, P.J., Mattiello, J., Lebihan, D.: MR diffusion tensor spectroscopy and imaging. *Biophysical Journal* 66, 259–267 (1994)

3. Burgeth, B., Pizarro, L., Didas, S., Weickert, J.: Coherence-Enhancing Diffusion for Matrix Fields. In: Proc. of Locally Adaptive Filtering in Signal and Image (to appear)
4. Creusen, E.J.: Numerical schemes for linear and non-linear enhancement of HARDI data. Master's thesis, Technische Universiteit Eindhoven, the Netherlands (2010)
5. Duits, R., Creusen, E., Ghosh, A., Dela Haije, T.: Diffusion, convection and erosion on  $\mathbb{R}^3 \times S^2$  and their application to the enhancement of crossing fibers. Technical report, TU/e, CASA, The Netherlands, Eindhoven, CASA-report nr.6 (January 2011), <http://www.win.tue.nl/casa/research/casareports/2011.html>
6. Duits, R., Dela Haije, T., Ghosh, A., Creusen, E.J., Vilanova, A., ter Haar Romeny, B.: Fiber enhancement in DW-MRI. In: Bruckstein, A.M., ter Haar Romeny, B.M., Bronstein, A.M., Bronstein, M.M. (eds.) SSVM 2011. LNCS, vol. 6667, pp. 1–13. Springer, Heidelberg (2011)
7. Duits, R., Franken, E.: Left-invariant diffusions on the space of positions and orientations and their application to crossing-preserving smoothing of HARDI images. CASA-report 18, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven (May 2009), <http://www.win.tue.nl/casa/research/casareports/2009.html>
8. Duits, R., Franken, E.: Left-invariant diffusions on the space of positions and orientations and their application to crossing-preserving smoothing of HARDI images. International Journal of Computer Vision 40 (2010)
9. Franken, E.: Enhancement of crossing elongated structures in images. PhD thesis, Eindhoven University of Technology (2008)
10. Franken, E.M., Duits, R., ter Haar Romeny, B.M.: Diffusion on the 3D Euclidean motion group for enhancement of HARDI data. In: Tai, X.-C., Mørken, K., Lysaker, M., Lie, K.-A. (eds.) SSVM 2009. LNCS, vol. 5567, pp. 820–831. Springer, Heidelberg (2009)
11. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. 12(7) (1990)
12. Prčkovska, V., Rodrigues, P., Duits, R., Vilanova, A., ter Haar Romeny, B.M.: Extrapolating fiber crossings from DTI data. Can we infer similar fiber crossings as in HARDI? In: CDMRI 2010 Proc. MICCAI Workshop Computational Diffusion MRI, China (2010)