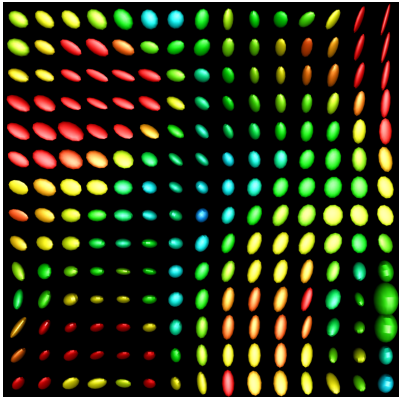
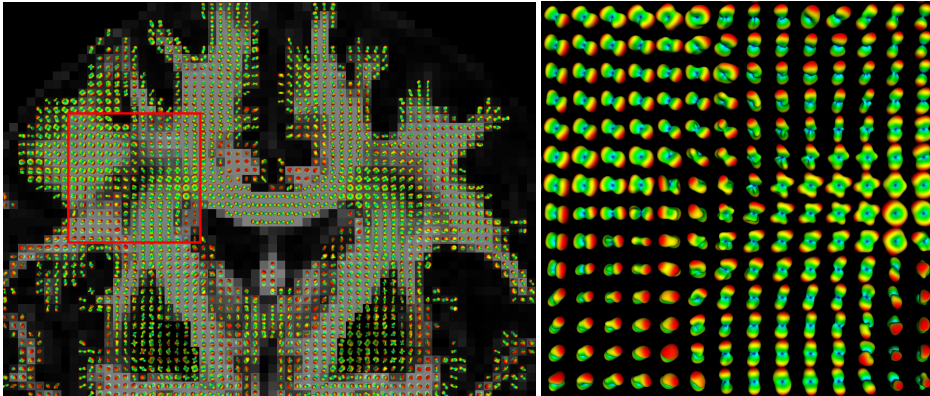


High Order Models in Diffusion MRI and Applications

DTI



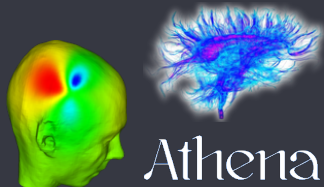
GDTI



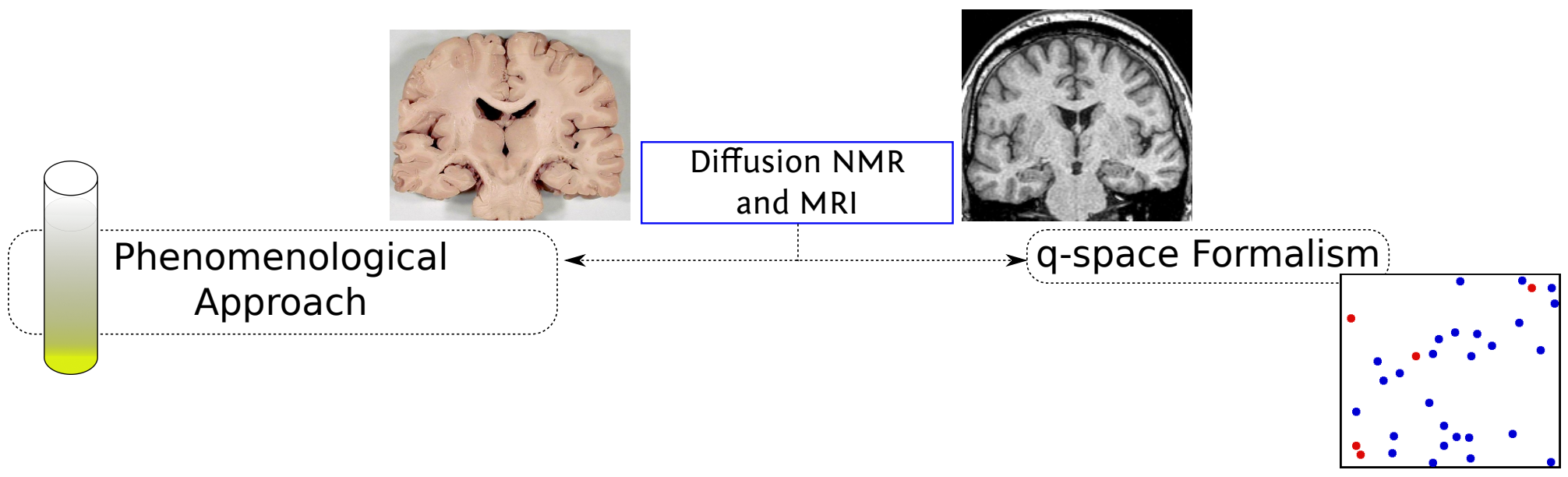
Aurobrata Ghosh

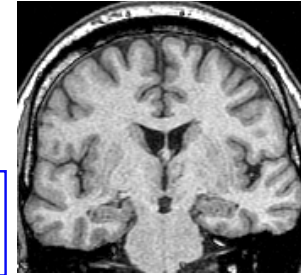
PhD thesis supervised by Dr. Rachid Deriche

Université de Nice
INRIA Sophia Antipolis - Méditerranée

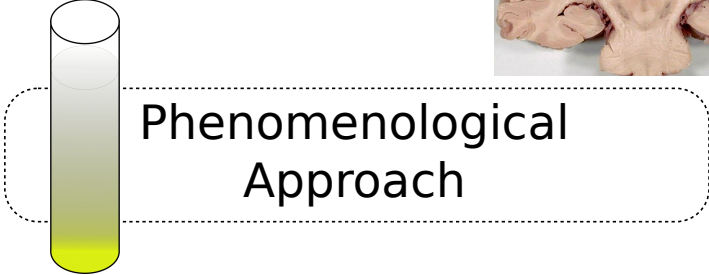


Reviewers Dr. Peter J Basser, STBB, NICHD, Bethesda, USA
Dr. Baba C Vemuri, University of Florida, Gainesville, USA
Examiners Dr. Habib Benali, INSERM / Pitié-Salpêtrière, Paris, France
Dr. Pierre Comon, I3S / CNRS / Université de Nice, France
Dr. Rachid Deriche, INRIA Sophia Antipolis - Méditerranée, France
Dr. Nikos Paragios, ECP / INRIA Saclay Ile-de-France, France



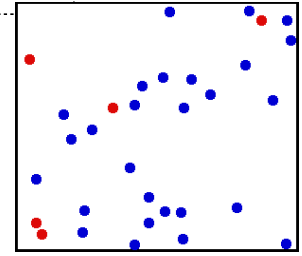


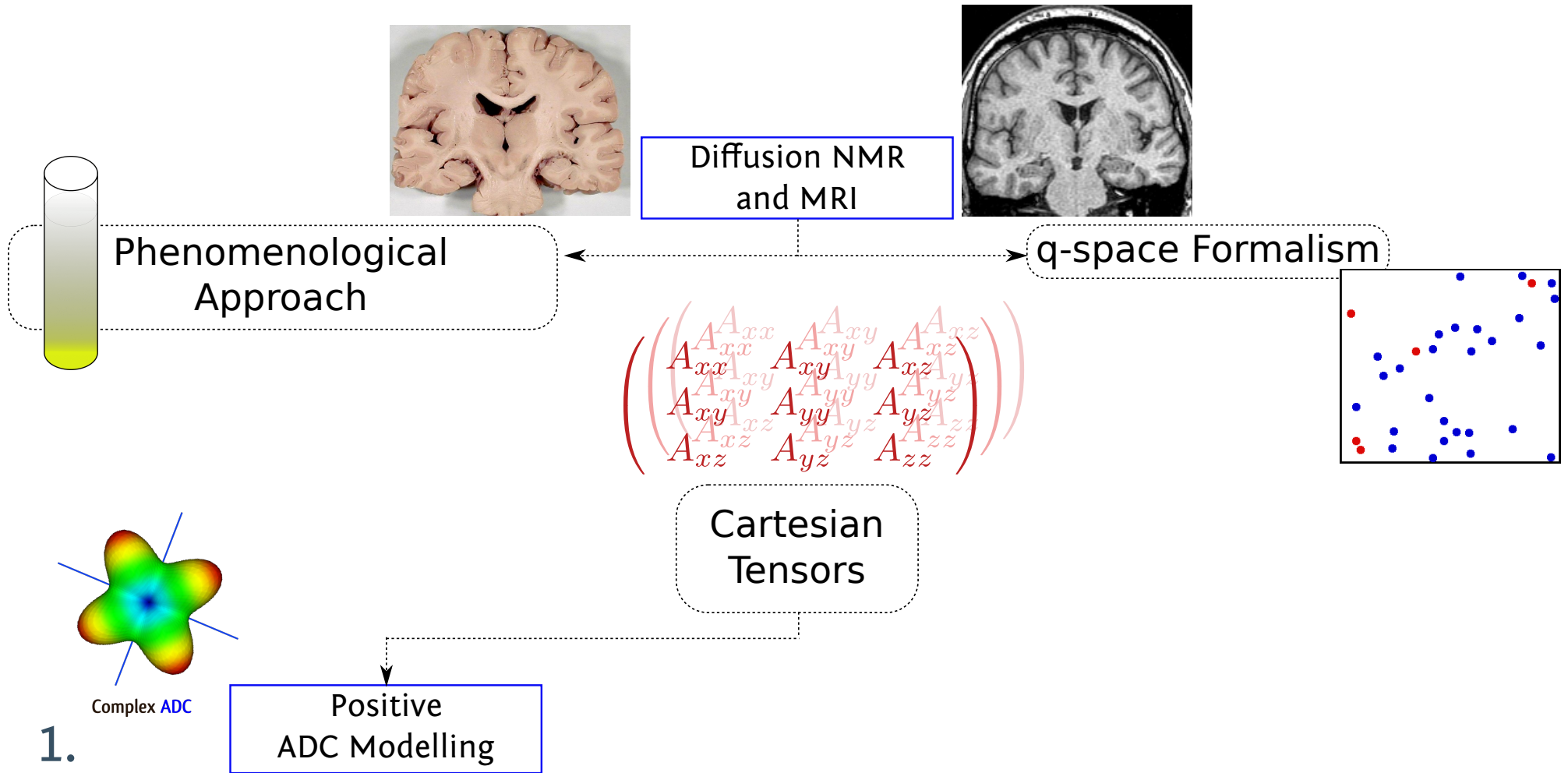
Diffusion NMR
and MRI

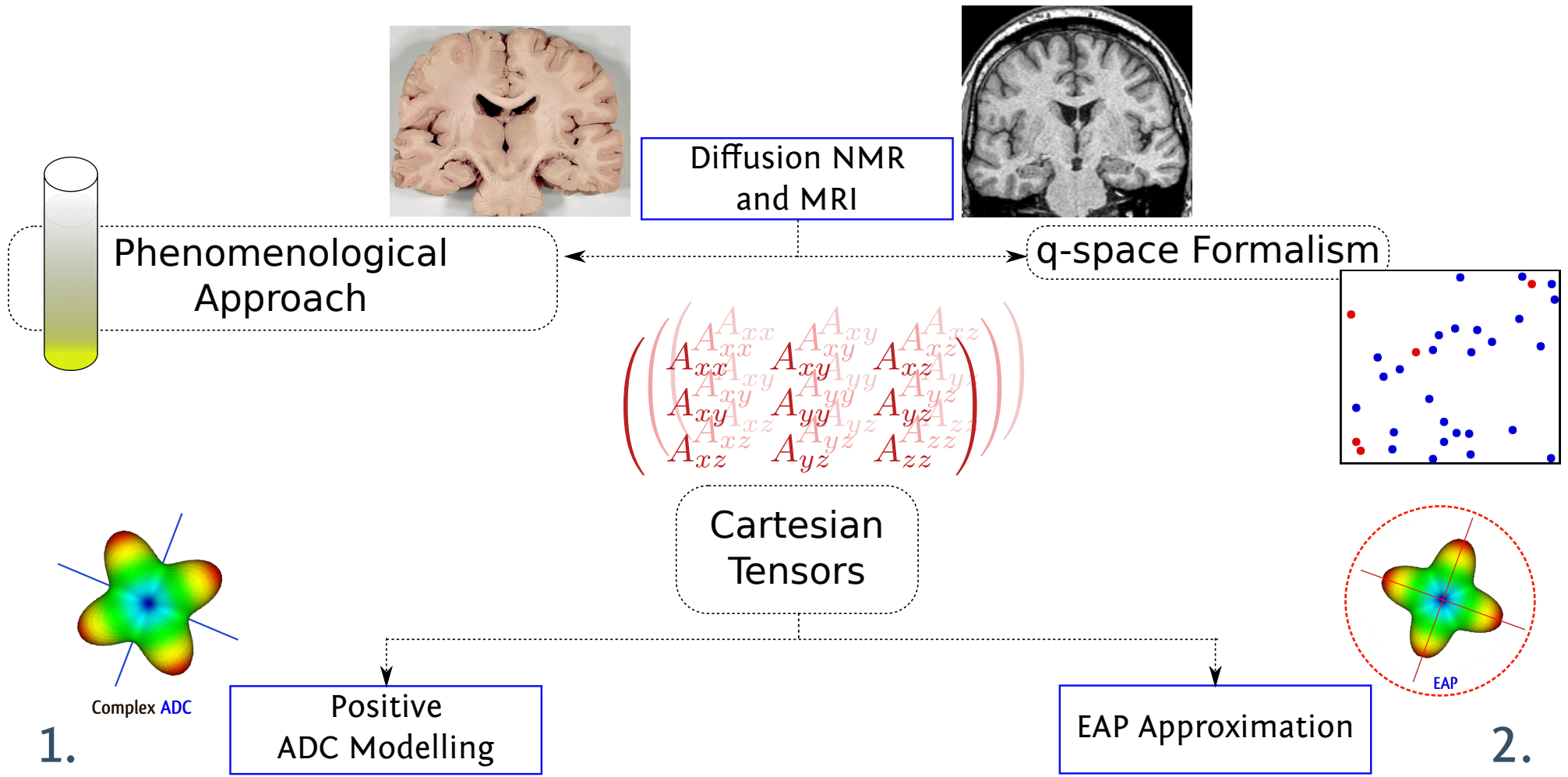


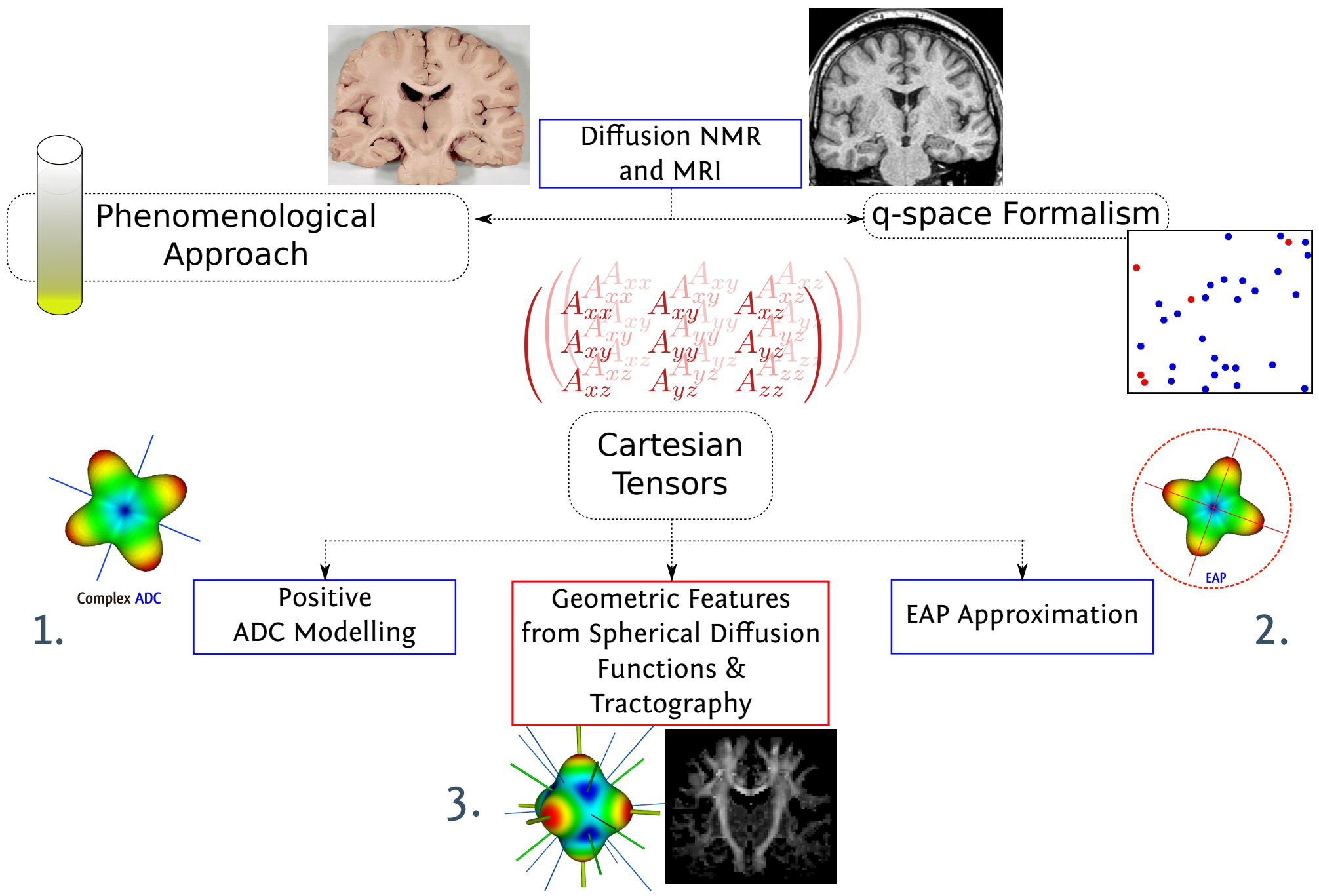
$$\begin{pmatrix} A_{xx} & A_{xy} & A_{xz} \\ A_{xy} & A_{yy} & A_{yz} \\ A_{xz} & A_{yz} & A_{zz} \end{pmatrix}$$

Cartesian
Tensors





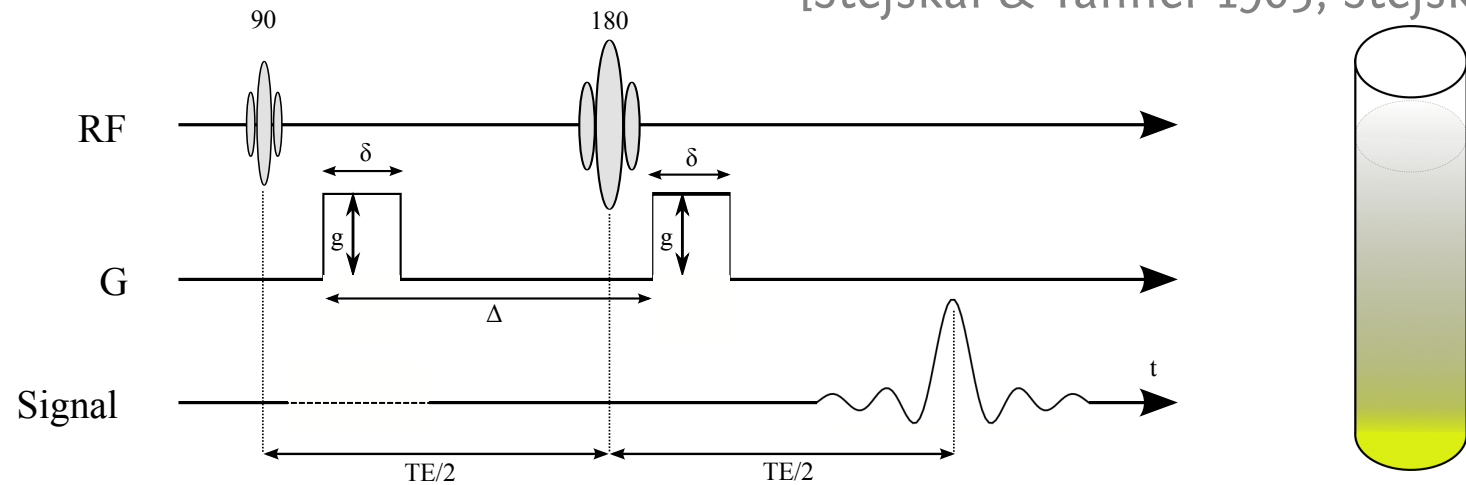




Introduction

1. Think Positive: 4th Order Symmetric Tensors
 - A Riemannian Approach
 - A Polynomial Approach (Ternary Quartic)
2. Fast and Analytical EAP Approximation from a 4th Order Tensor
3. Geometric Features from Spherical Diffusion Functions
 - Maxima Extraction
 - Peak Fractional Anisotropy

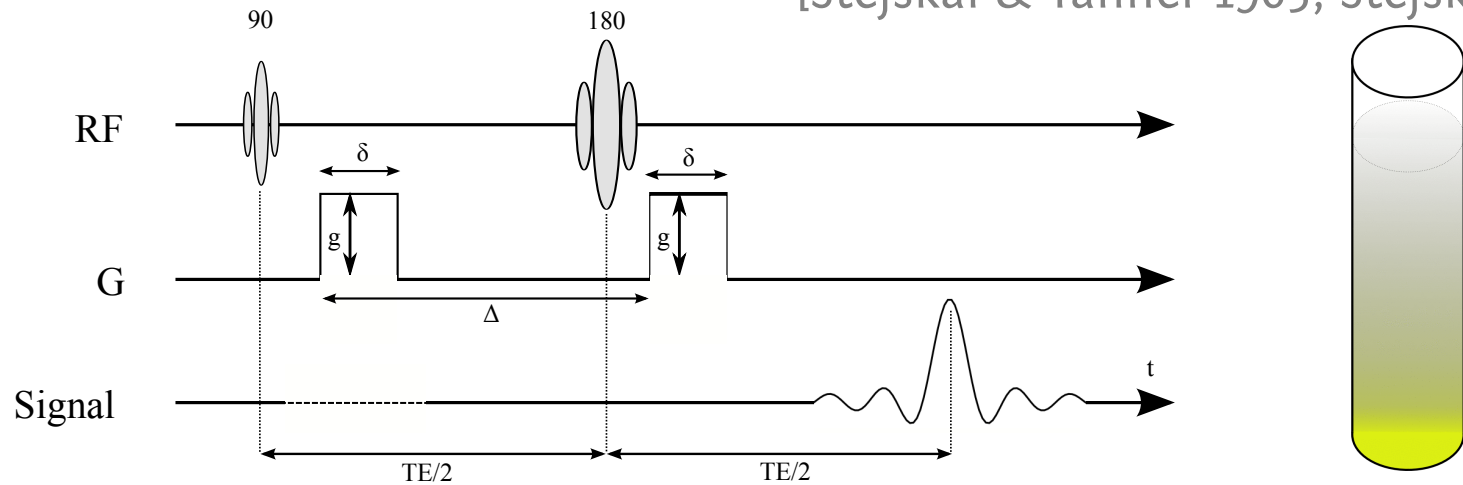
[Stejskal & Tanner 1965, Stejskal 1965]



★ Stejskal-Tanner signal: a phenomenological approach

⦿ Signal: $S = S_0 \exp(-bD)$ $S = S_0 \exp(-b \cdot \underbrace{g^T D g}_{ADC})$

[Stejskal & Tanner 1965, Stejskal 1965]



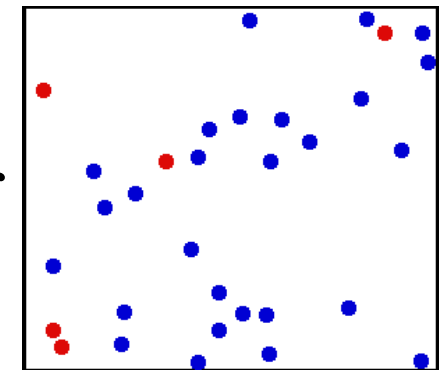
★ Stejskal-Tanner signal: a phenomenological approach

☉ Signal: $S = S_0 \exp(-bD)$ $S = S_0 \exp(-b \cdot \underbrace{\mathbf{g}^T \mathbf{D} \mathbf{g}}_{ADC})$

★ q-Space formalism: a random walk model

☉ Fourier Transform: $\frac{S(\mathbf{q}, \Delta)}{S_0} = \int P(\mathbf{r}, \Delta) e^{2\pi i \mathbf{q} \cdot \mathbf{r}} d\mathbf{r}$

$P(\mathbf{r}, \Delta)$: Ensemble Average Propagator (EAP)



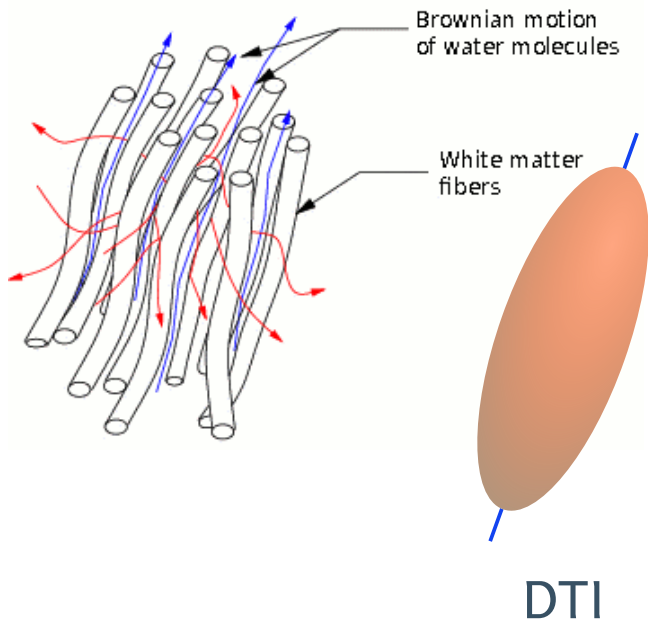
[Stejskal 1965, Callaghan 1993]

★ Stejskal-Tanner signal [Basser 1994]

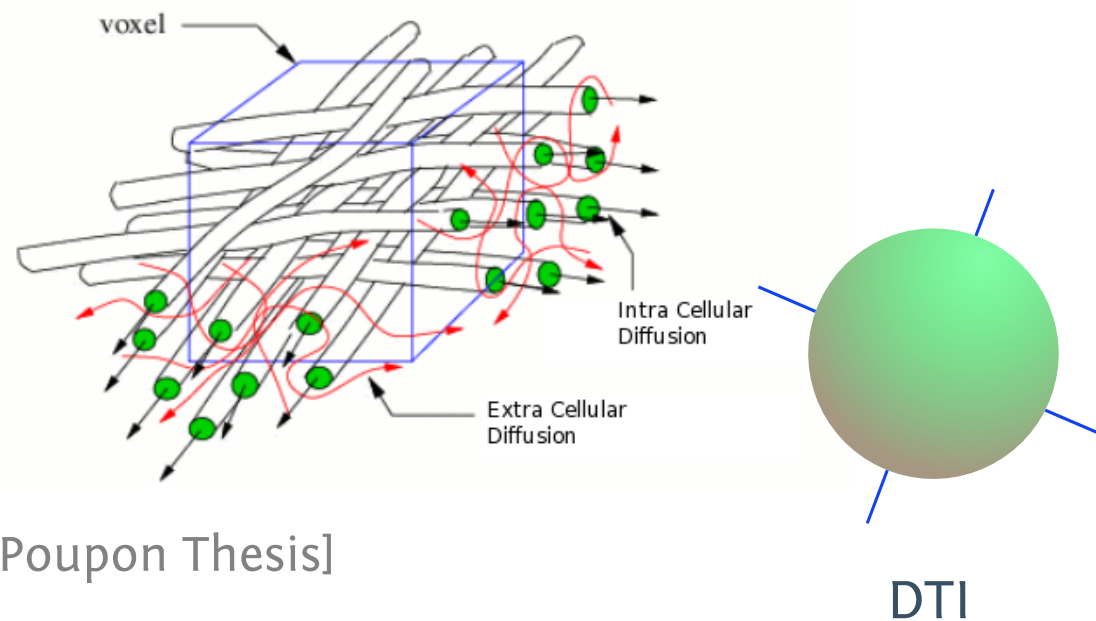
$$\frac{S_i}{S_0} = \exp(-b\mathbf{g}_i^T \mathbf{D} \mathbf{g}_i)$$

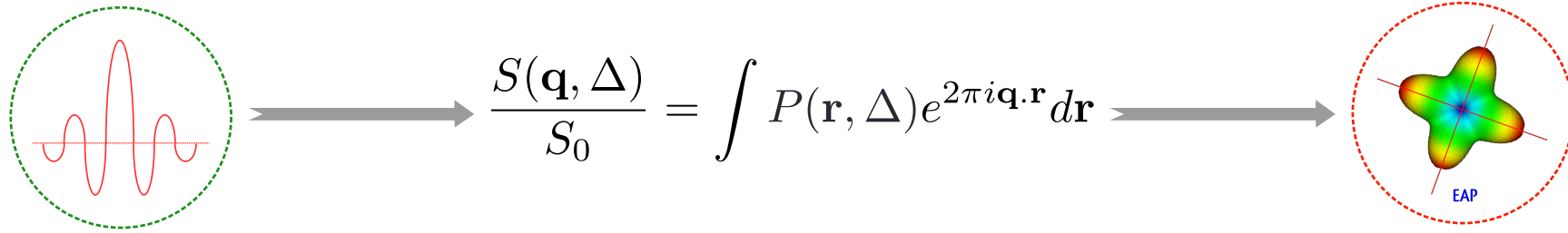
★ Free diffusion / Gaussian diffusion [Basser 1994]

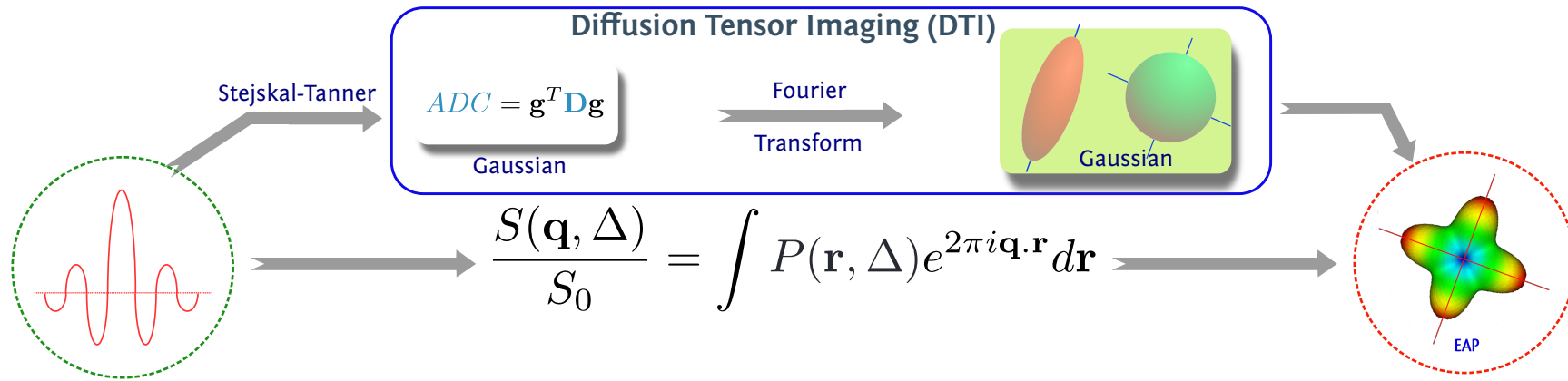
$$P(\mathbf{r}, t) = \frac{1}{\sqrt{|\mathbf{D}|}(4\pi t)^3} \exp\left(-\frac{\mathbf{r}^T \mathbf{D}^{-1} \mathbf{r}}{4t}\right)$$

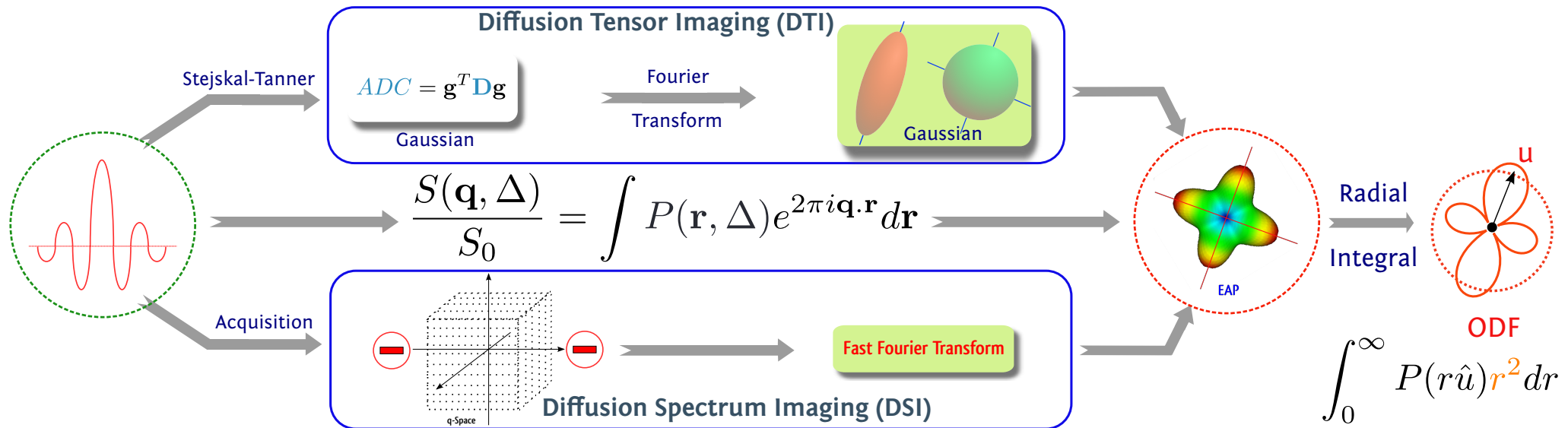


Fibers: [Poupon Thesis]

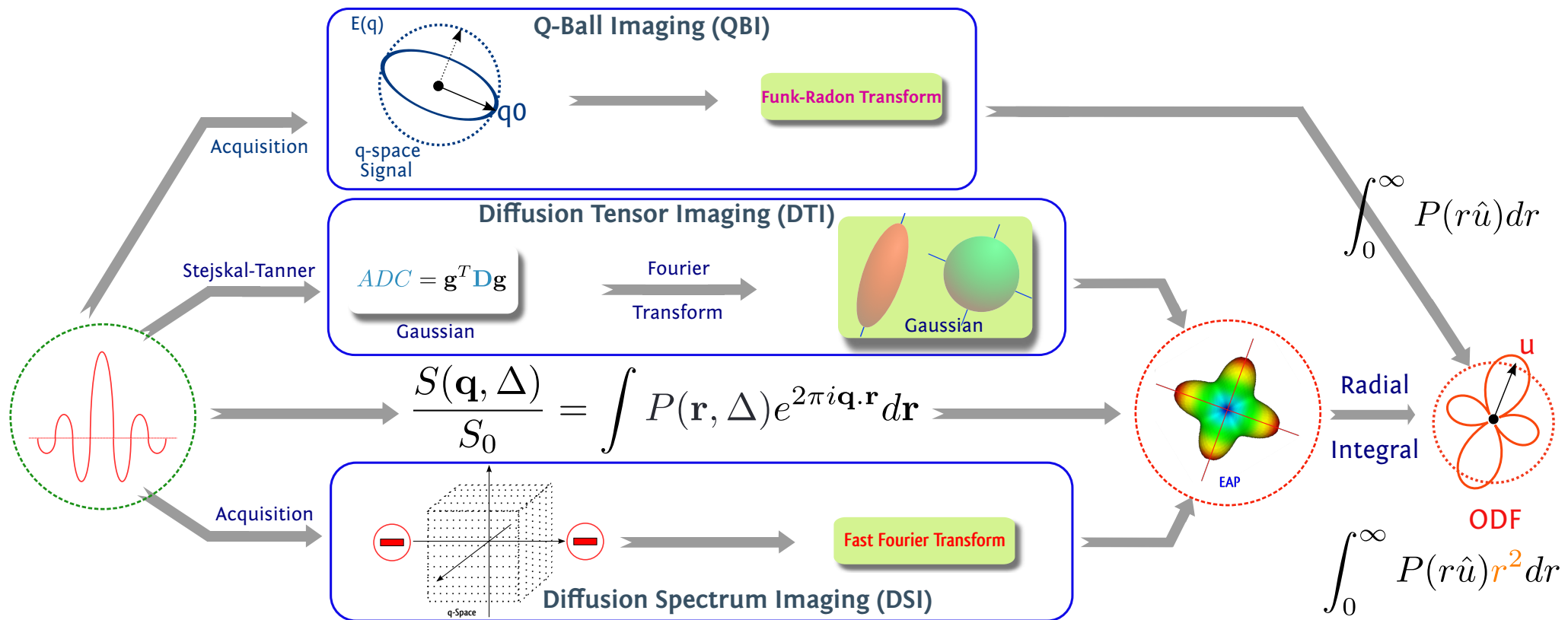






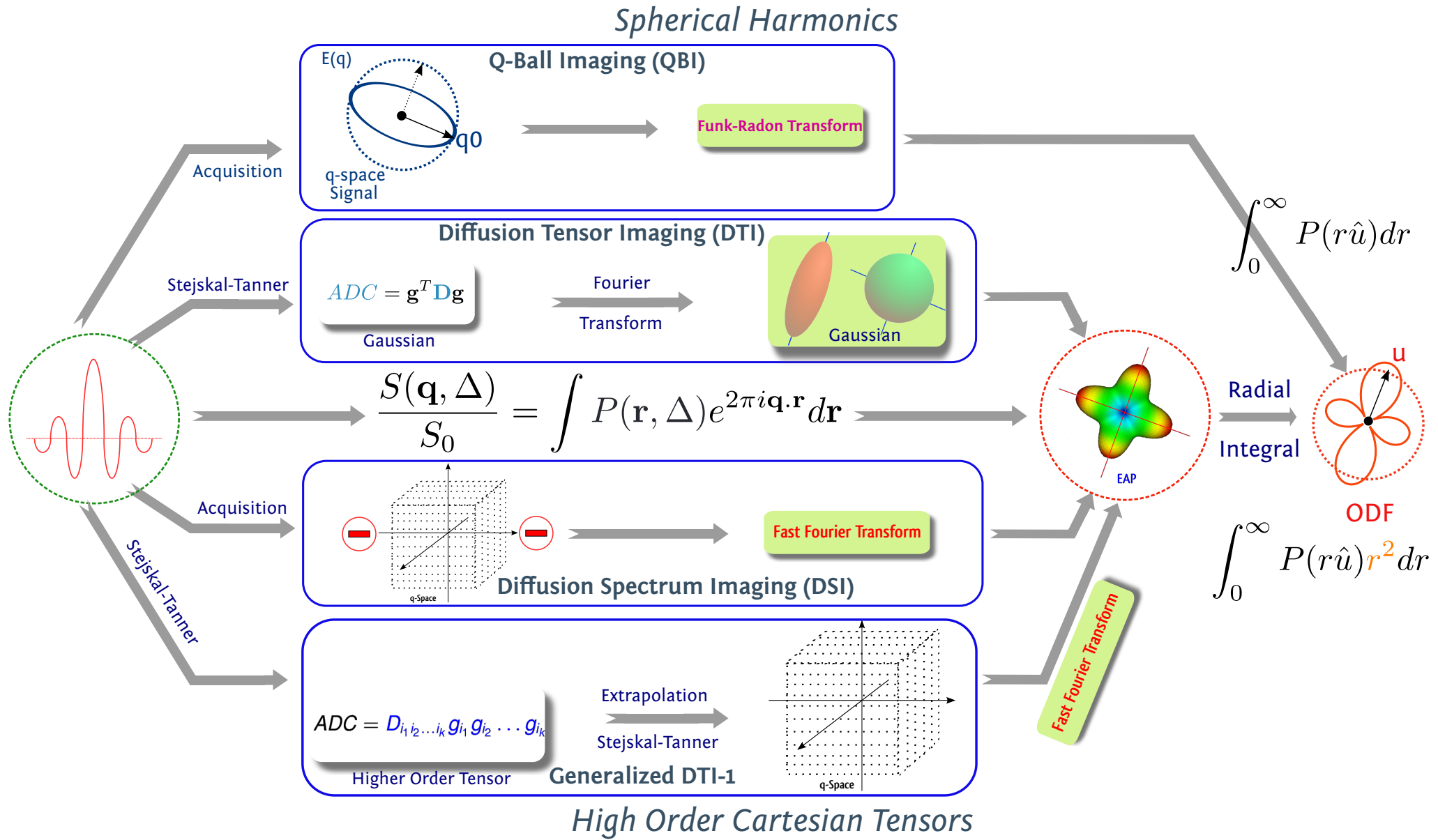


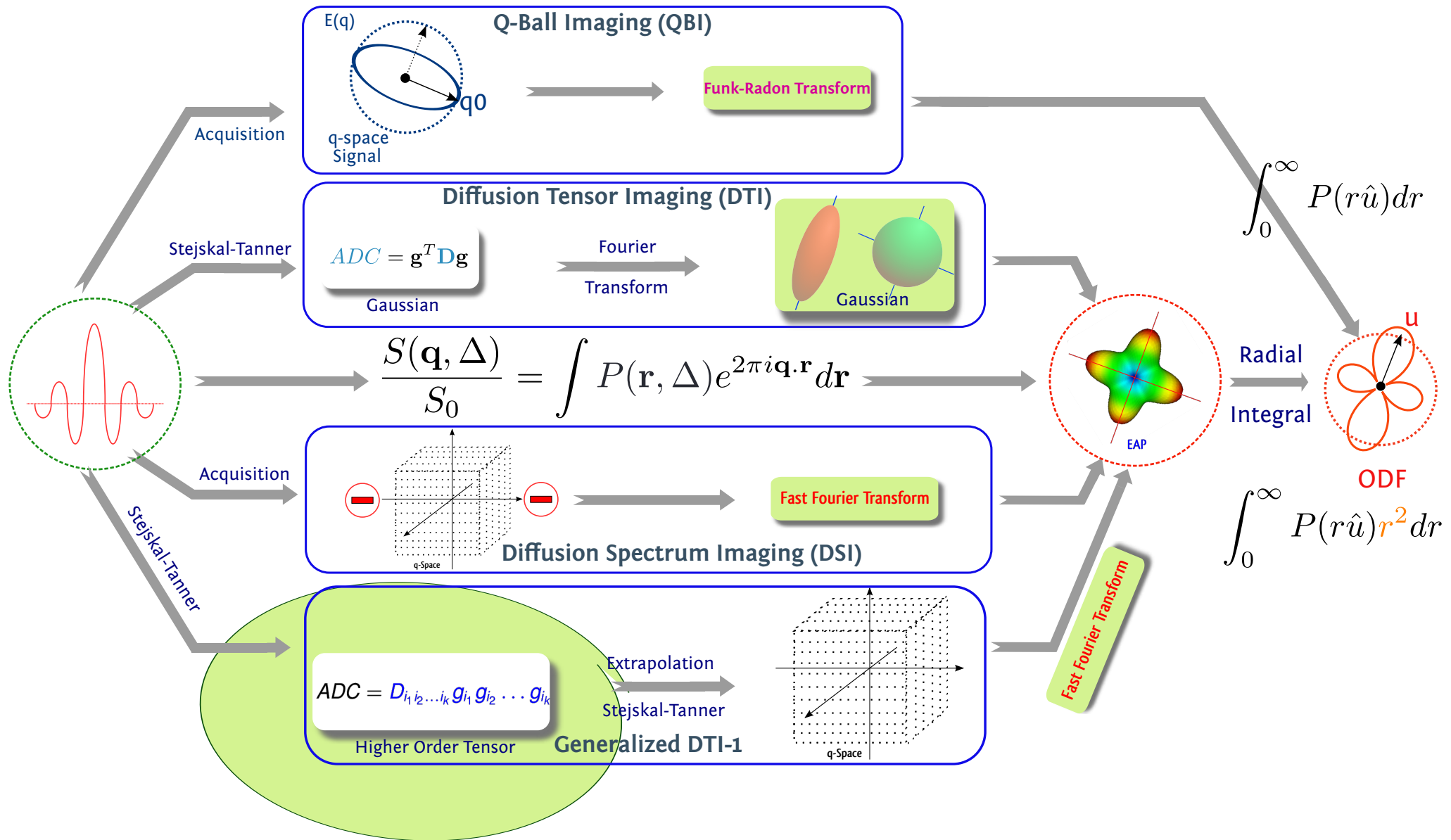
- ⊖ Lengthy acquisition
- ⊖ Strong b -values



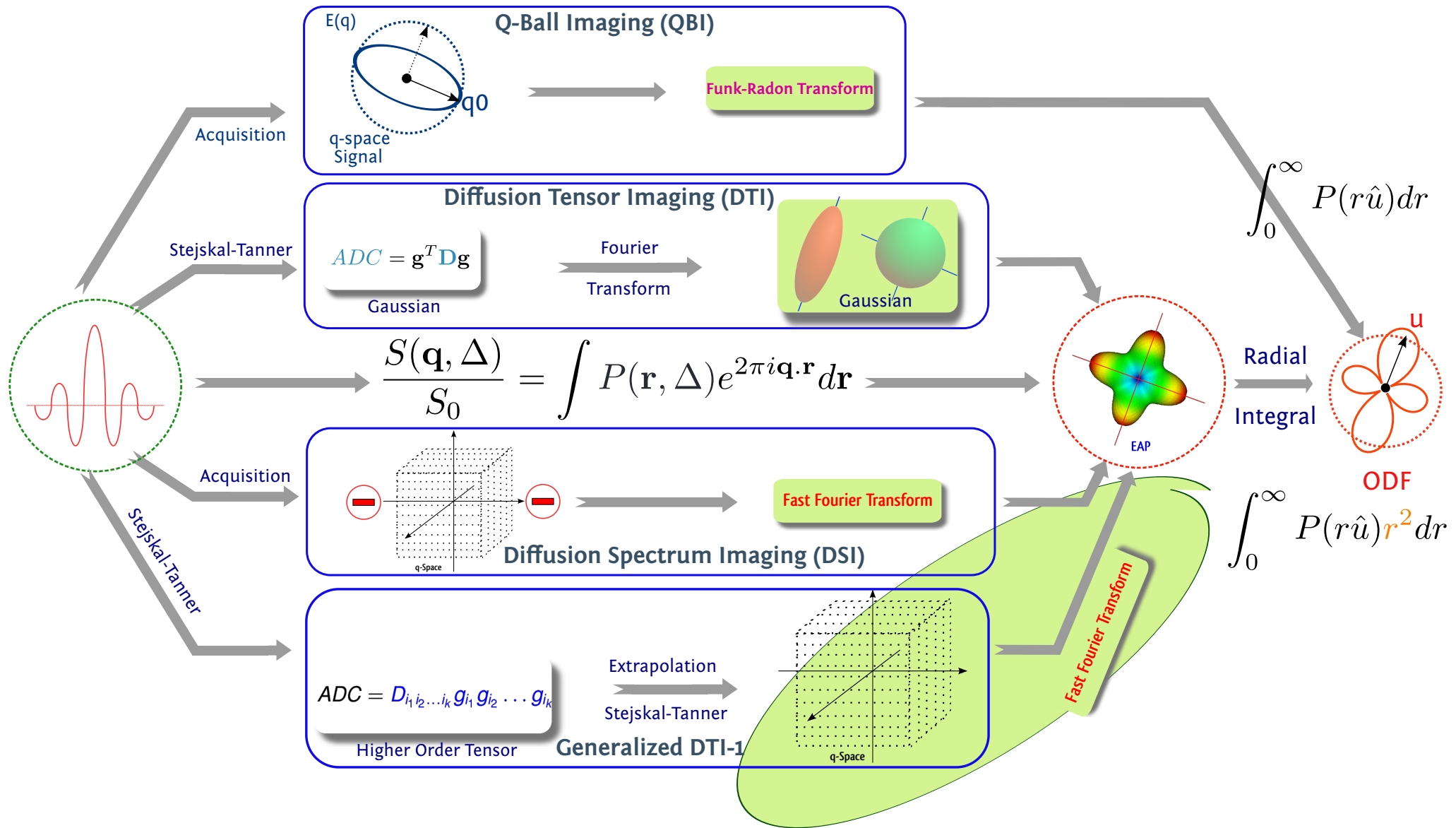
- ④ Function: Sphere → Sphere [Tuch 2004]
- ④ Angular information
- ④ Analytical Expression (Spherical Harmonics)

[Anderson 2005, Hess 2006, Descoteaux 2007]

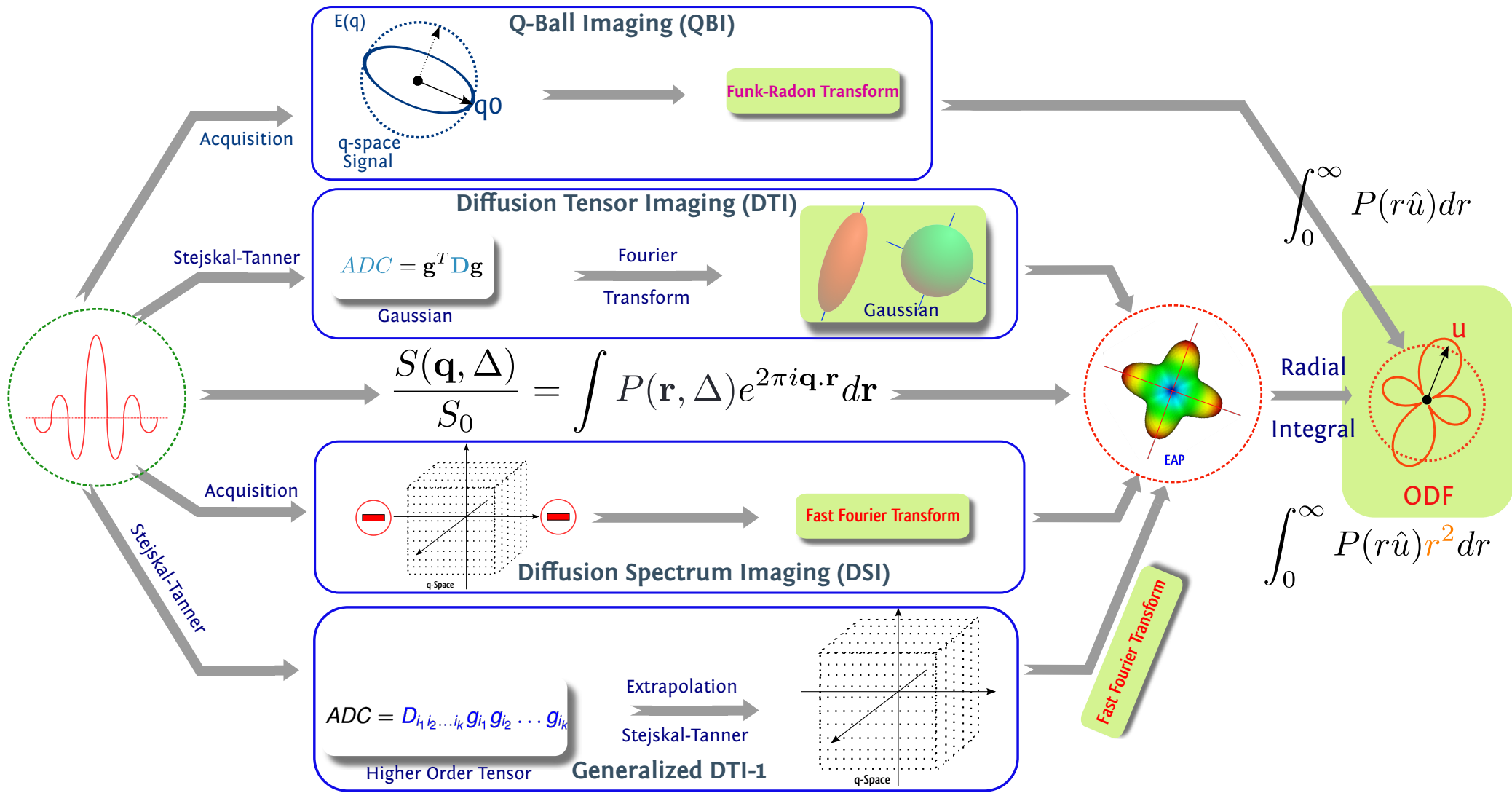




1. Think Positive: High Order Symmetric Tensors



2. Analytical EAP Approximation from a 4th Order Tensor



3. Geometric Features from Spherical Diffusion Functions

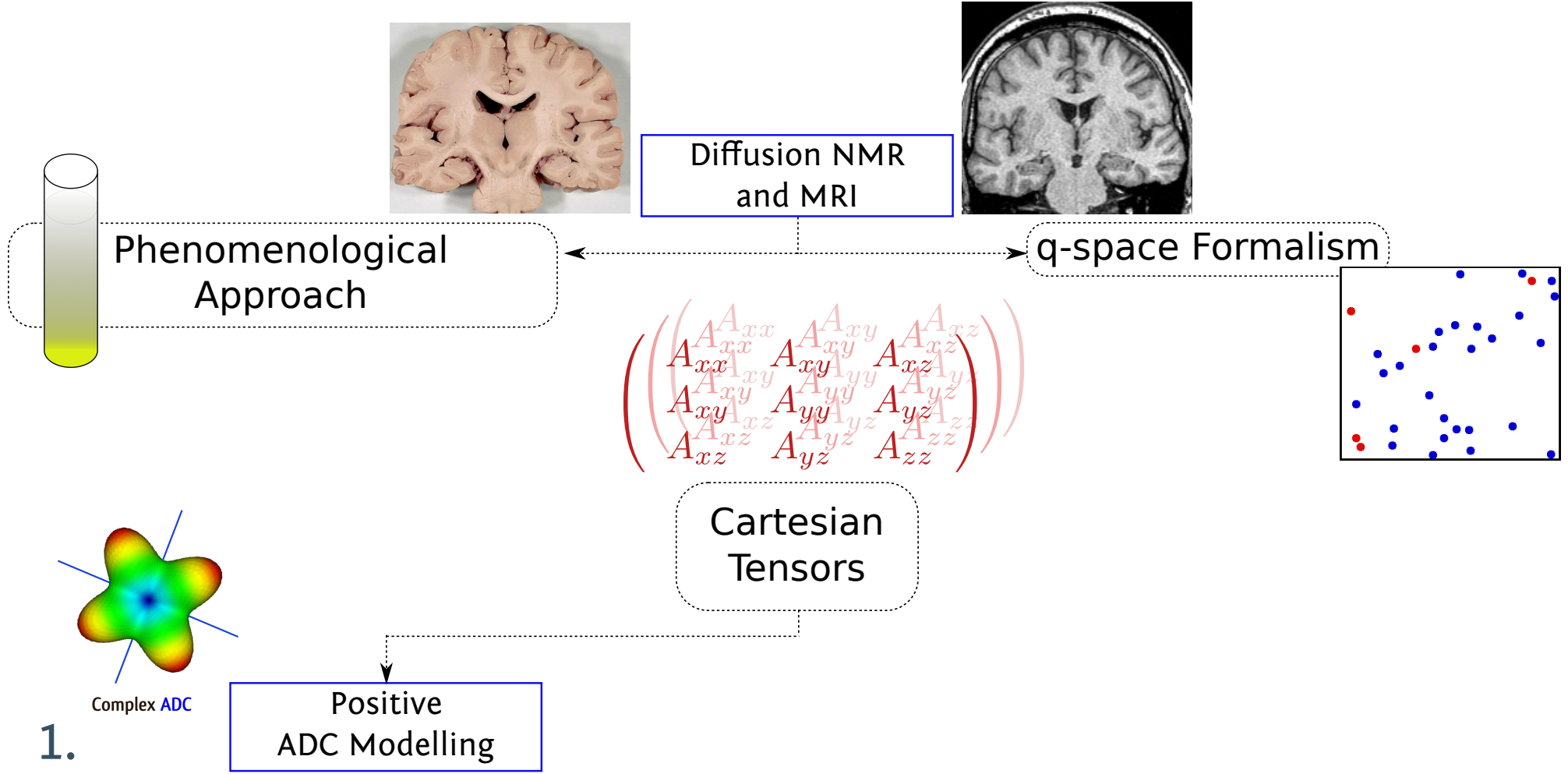
- ★ Research at INRIA Sophia Antipolis Méditerranée:
- ⦿ DTI with positivity constraint (*tensors*):
 - > Riemannian Metric: Odysée, Asclepios [Pennec 2006, Lenglet 2006]
 - > Log-Euclidean Metric: Asclepios [Arsigny 2006]
- ⦿ Analytical QBI (*high order models*): Odysée [Descoteaux 2008]



★ Explore *high order Cartesian tensor models*

Introduction

1. Think Positive: 4th Order Symmetric Tensors
 - A Riemannian Approach
 - A Polynomial Approach (Ternary Quartic)
2. Fast and Analytical EAP Approximation from a 4th Order Tensor
3. Geometric Features from Spherical Diffusion Functions
 - Maxima Extraction
 - Peak Fractional Anisotropy



★ Diffusion is a physical process: Positive DTI

- ⌚ Riemannian Metric: [Bachelor'05, Moakher'05, Pennec'06, Lenglet'06]

$$d(\mathbf{D}_1, \mathbf{D}_2)_{Rm} = \sqrt{\text{tr}(\ln^2(\mathbf{D}_1^{-1/2} \mathbf{D}_2 \mathbf{D}_1^{-1/2}))}$$

- ⌚ Log-Euclidean Metric: [Arsigny 2006]

$$d(\mathbf{D}_1, \mathbf{D}_2)_{LE} = \sqrt{\text{tr}(\{\ln(\mathbf{D}_1) - \ln(\mathbf{D}_2)\}^2)}$$

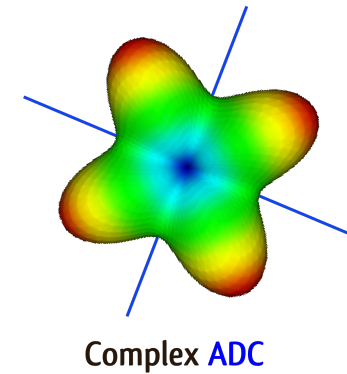
★ DTI limited to Gaussian diffusion

$$ADC = \mathbf{g}^T \mathbf{D} \mathbf{g} = g_1^2 d_{11} + g_2^2 d_{22} + g_3^2 d_{33} + \dots \text{2nd order polynomial}$$

[Ozarslan 2003]

$$ADC = D_{j_1 j_2 \dots j_k} g_{j_1} g_{j_2} \dots g_{j_k} = \sum_{j_1=1}^3 \sum_{j_2=1}^3 \dots \sum_{j_k=1}^3 D_{j_1 j_2 \dots j_k} g_{j_1} g_{j_2} \dots g_{j_k}$$

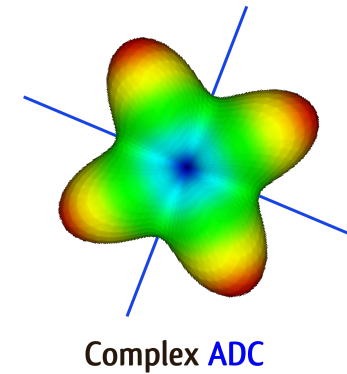
- ★ Higher Order symmetric Tensors
- ★ Complex ADC profile



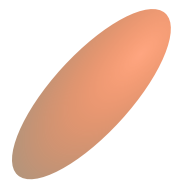
[Ozarslan 2003]

$$ADC = D_{j_1 j_2 \dots j_k} g_{j_1} g_{j_2} \dots g_{j_k} = \sum_{j_1=1}^3 \sum_{j_2=1}^3 \dots \sum_{j_k=1}^3 D_{j_1 j_2 \dots j_k} g_{j_1} g_{j_2} \dots g_{j_k}$$

- ★ Higher Order symmetric Tensors
- ★ Complex ADC profile



$k = 2$ (DTI)

$$\sum_{j_1=1}^3 \sum_{j_2=1}^3 D_{j_1 j_2} g_{j_1} g_{j_2} = \mathbf{g}^T \mathbf{D} \mathbf{g}$$


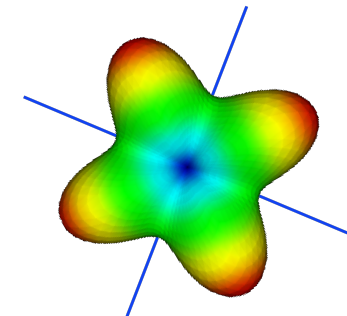
$k = 4$

$$\sum_{j_1=1}^3 \sum_{j_2=1}^3 \sum_{j_3=1}^3 \sum_{j_4=1}^3 D_{j_1 j_2 j_3 j_4} g_{j_1} g_{j_2} g_{j_3} g_{j_4}$$

$$ADC = D_{j_1 j_2 \dots j_k} g_{j_1} g_{j_2} \dots g_{j_k} = \sum_{j_1=1}^3 \sum_{j_2=1}^3 \dots \sum_{j_k=1}^3 D_{j_1 j_2 \dots j_k} g_{j_1} g_{j_2} \dots g_{j_k}$$

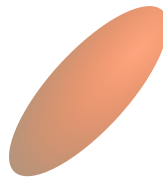
★ Higher Order symmetric Tensors

★ Complex ADC profile



Complex ADC

$k = 2$ (DTI)



$$\sum_{j_1=1}^3 \sum_{j_2=1}^3 D_{j_1 j_2} g_{j_1} g_{j_2} = \mathbf{g}^T \mathbf{D} \mathbf{g}$$

$k = 4$

$$\sum_{j_1=1}^3 \sum_{j_2=1}^3 \sum_{j_3=1}^3 \sum_{j_4=1}^3 D_{j_1 j_2 j_3 j_4} g_{j_1} g_{j_2} g_{j_3} g_{j_4}$$

☉ Positive ADC?

★ Isometric mapping: Matrix \rightarrow Vector

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix} \longrightarrow \mathbf{d} = \begin{bmatrix} D_{xx} \\ D_{yy} \\ D_{zz} \\ \sqrt{2}D_{xy} \\ \sqrt{2}D_{xz} \\ \sqrt{2}D_{yz} \end{bmatrix}$$

★ Major+Minor symmetry: 4th order 3D \rightarrow 6D Matrix:

[Moakher 2006, Basser 2007]

$$\mathbf{A} = \left(\left(\begin{pmatrix} A_{xxx} & A_{xyx} & A_{xxz} \\ A_{xyx} & A_{yyy} & A_{yyz} \\ A_{xxz} & A_{yyz} & A_{zzz} \end{pmatrix} \right) \right)$$

$$\mathbf{A} = \begin{pmatrix} A_{xxxx} & A_{xxyy} & A_{xxzz} & \sqrt{2}A_{xxxy} & \sqrt{2}A_{xxxz} & \sqrt{2}A_{xxyz} \\ A_{xxyy} & A_{yyyy} & A_{yyzz} & \sqrt{2}A_{yyxy} & \sqrt{2}A_{yyxz} & \sqrt{2}A_{yyyz} \\ A_{xxzz} & A_{yyzz} & A_{zzzz} & \sqrt{2}A_{zzxy} & \sqrt{2}A_{zzxz} & \sqrt{2}A_{zzyz} \\ \sqrt{2}A_{xxxy} & \sqrt{2}A_{yyxy} & \sqrt{2}A_{zzxy} & 2A_{xyxy} & 2A_{xyxz} & 2A_{xyyz} \\ \sqrt{2}A_{xxxz} & \sqrt{2}A_{yyxz} & \sqrt{2}A_{zzxz} & 2A_{xyxz} & 2A_{xxzz} & 2A_{xzyz} \\ \sqrt{2}A_{xxyz} & \sqrt{2}A_{yyyz} & \sqrt{2}A_{zzyz} & 2A_{xyyz} & 2A_{xzyz} & 2A_{yzyz} \end{pmatrix}$$

★ Isometric mapping: Matrix \rightarrow Vector

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix} \longrightarrow \mathbf{d} = \begin{bmatrix} D_{xx} \\ D_{yy} \\ D_{zz} \\ \sqrt{2}D_{xy} \\ \sqrt{2}D_{xz} \\ \sqrt{2}D_{yz} \end{bmatrix}$$

★ Major+Minor symmetry: 4th order 3D \rightarrow 6D Matrix:

[Moakher 2006, Basser 2007]

$$\mathbf{A} = \left(\left(\left(\begin{matrix} A_{xxx} & A_{xyx} & A_{xzx} \\ A_{xyx} & A_{yyy} & A_{yzy} \\ A_{xzx} & A_{yzy} & A_{zzz} \end{matrix} \right) \right) \right)$$

$$\mathbf{A} = \mathbf{A}^s + \mathbf{A}^a$$

$$\mathbf{A} = \begin{pmatrix} A_{xxxx} & A_{xxyy} & A_{xxzz} & \sqrt{2}A_{xxxy} & \sqrt{2}A_{xxxz} & \sqrt{2}A_{xxyz} \\ A_{xxyy} & A_{yyyy} & A_{yyzz} & \sqrt{2}A_{yyxy} & \sqrt{2}A_{yyxz} & \sqrt{2}A_{yyyz} \\ A_{xxzz} & A_{yyzz} & A_{zzzz} & \sqrt{2}A_{zzxy} & \sqrt{2}A_{zzxz} & \sqrt{2}A_{zzyz} \\ \sqrt{2}A_{xxxy} & \sqrt{2}A_{yyxy} & \sqrt{2}A_{zzxy} & 2A_{xxyy} & 2A_{xyxz} & 2A_{xyyz} \\ \sqrt{2}A_{xxxz} & \sqrt{2}A_{yyxz} & \sqrt{2}A_{zzxz} & 2A_{xyxz} & 2A_{xxzz} & 2A_{xzyz} \\ \sqrt{2}A_{xxyz} & \sqrt{2}A_{yyyz} & \sqrt{2}A_{zzyz} & 2A_{xyyz} & 2A_{xzyz} & 2A_{yzyz} \end{pmatrix}$$

★ Total symmetry:
GDTI [Moakher 2006]

★ GDTI 4th order Stejskal-Tanner equation: [Moakher 2006]

$$-\frac{1}{b} \ln \left(\frac{S_i}{S_0} \right) = \hat{A}_{j_1 j_2 j_3 j_4} g_{j_1} g_{j_2} g_{j_3} g_{j_4} = \mathbf{d}_i^T \hat{\mathbf{A}} \mathbf{d}_i \quad \mathbf{d}_i = \mathbf{g}_i \times \mathbf{g}_i$$

★ GDTI 4th order Stejskal-Tanner equation: [Moakher 2006]

$$-\frac{1}{b} \ln \left(\frac{S_i}{S_0} \right) = \hat{A}_{j_1 j_2 j_3 j_4} g_{j_1} g_{j_2} g_{j_3} g_{j_4} = \mathbf{d}_i^T \hat{\mathbf{A}} \mathbf{d}_i \quad \mathbf{d}_i = \mathbf{g}_i \times \mathbf{g}_i$$

★ Riemannian gradient descent on Sym_6^+

$$\hat{\mathbf{A}}_{t+1} = \hat{\mathbf{A}}_t^{\frac{1}{2}} \exp \left(-\epsilon \cdot \hat{\mathbf{A}}_t^{\frac{1}{2}} \left[\sum_{i=1}^N \left(\frac{1}{b} \ln \left(\frac{S_i}{S_0} \right) + \mathbf{d}_i^T \hat{\mathbf{A}} \mathbf{d}_i \right) \cdot (\mathbf{d}_i \mathbf{d}_i^T) \right] \hat{\mathbf{A}}_t^{\frac{1}{2}} \right) \hat{\mathbf{A}}_t^{\frac{1}{2}}$$

Guarantees *positive definiteness*: $ADC = \mathbf{d}_i^T \hat{\mathbf{A}} \mathbf{d}_i > 0$

⦿ Estimate totally symmetric: $\hat{\mathbf{A}} \longrightarrow \hat{\mathbf{A}}^s$

★ GDTI 4th order Stejskal-Tanner equation: [Moakher 2006]

$$-\frac{1}{b} \ln \left(\frac{S_i}{S_0} \right) = \hat{A}_{j_1 j_2 j_3 j_4} g_{j_1} g_{j_2} g_{j_3} g_{j_4} = \mathbf{d}_i^T \hat{\mathbf{A}} \mathbf{d}_i \quad \mathbf{d}_i = \mathbf{g}_i \times \mathbf{g}_i$$

★ Riemannian gradient descent on Sym_6^+

$$\hat{\mathbf{A}}_{t+1} = \hat{\mathbf{A}}_t^{\frac{1}{2}} \exp \left(-\epsilon \cdot \hat{\mathbf{A}}_t^{\frac{1}{2}} \left[\sum_{i=1}^N \left(\frac{1}{b} \ln \left(\frac{S_i}{S_0} \right) + \mathbf{d}_i^T \hat{\mathbf{A}} \mathbf{d}_i \right) \cdot (\mathbf{d}_i \mathbf{d}_i^T) \right] \hat{\mathbf{A}}_t^{\frac{1}{2}} \right) \hat{\mathbf{A}}_t^{\frac{1}{2}}$$

Guarantees *positive definiteness*: $ADC = \mathbf{d}_i^T \hat{\mathbf{A}} \mathbf{d}_i > 0$

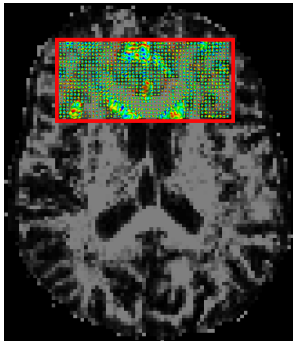
⊙ Estimate totally symmetric: $\hat{\mathbf{A}} \longrightarrow \hat{\mathbf{A}}^s$

⊖ Riemannian constraint is too strong

$$\mathbf{d}^T \hat{\mathbf{A}} \mathbf{d} > 0 \quad \forall \mathbf{d} \in \mathbf{R}^6$$

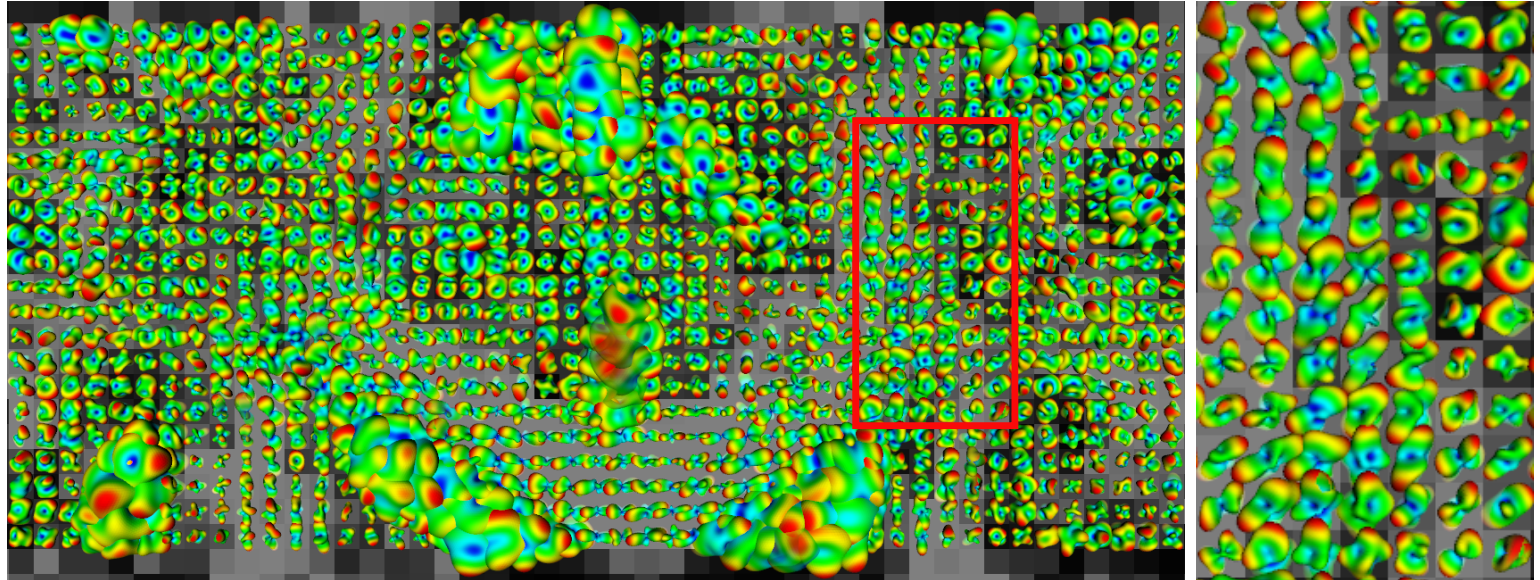
(81 dirs)	LS	SH	RM
Positive	181757	249263	249352
Negative	67595	89	0

249352 estimated tensors. ADC along 81 dirs

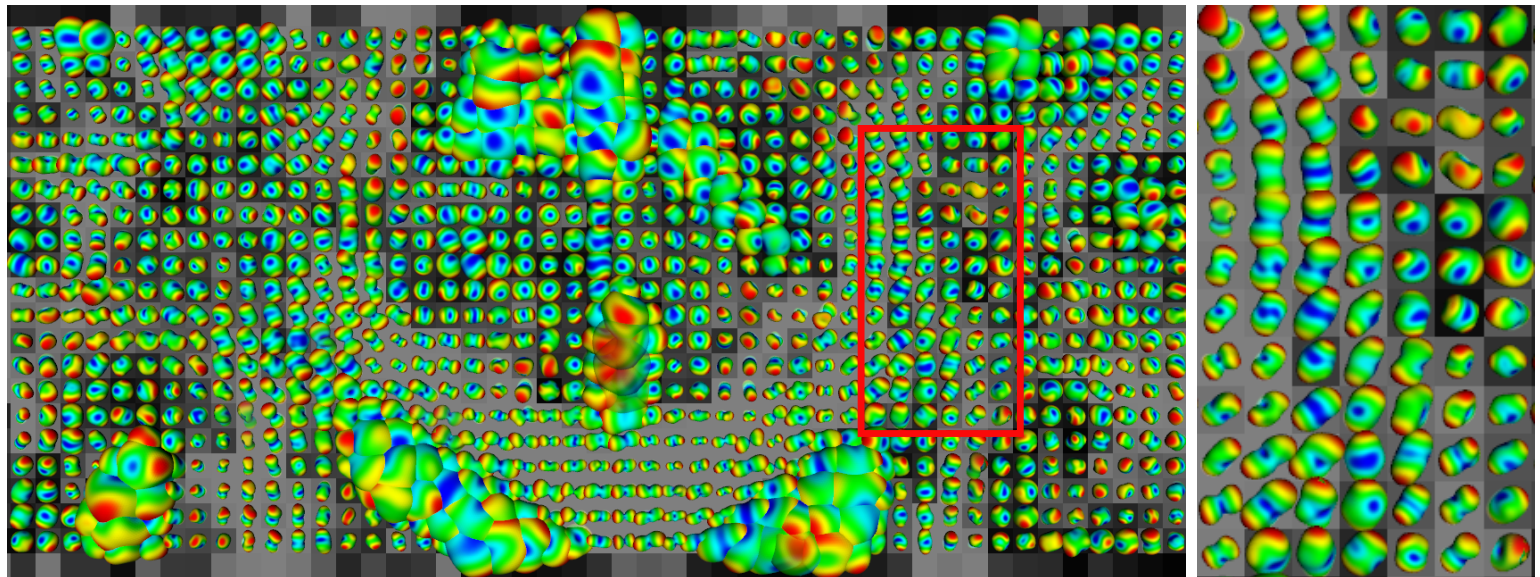


Real data:
[Poupon 2006]

Euclidean LS



Riemannian



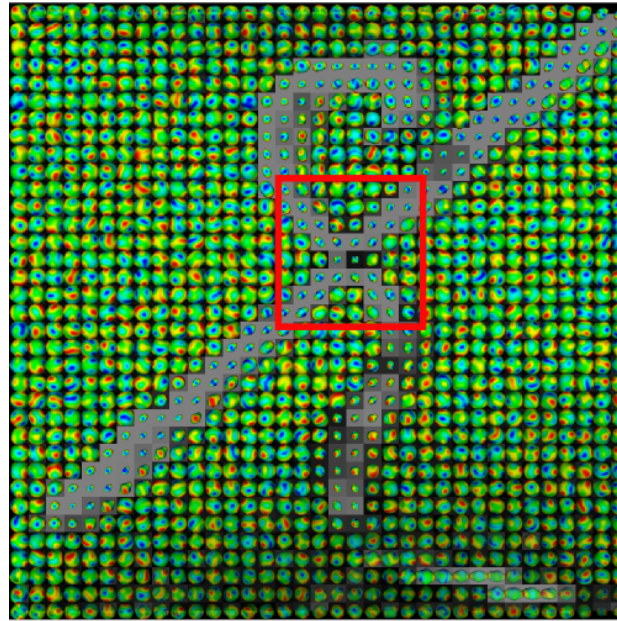
Rat phantom data:

[Campbell 2005]

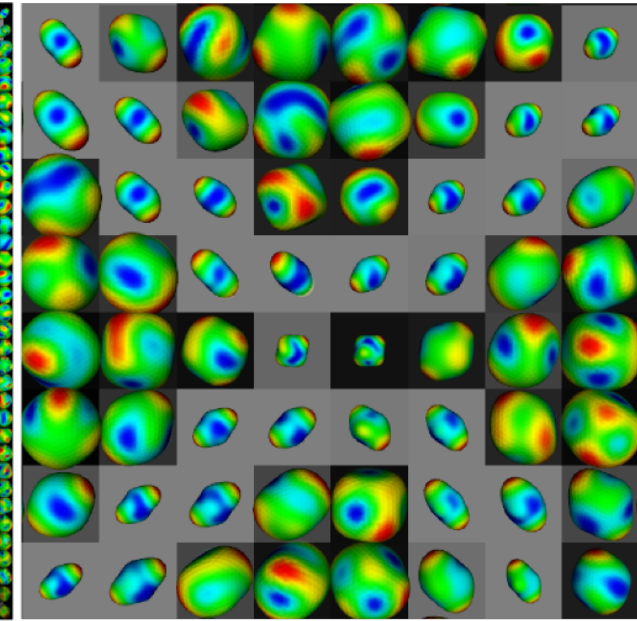
(a) Layout

(b),(c) GDTI \rightarrow EAP

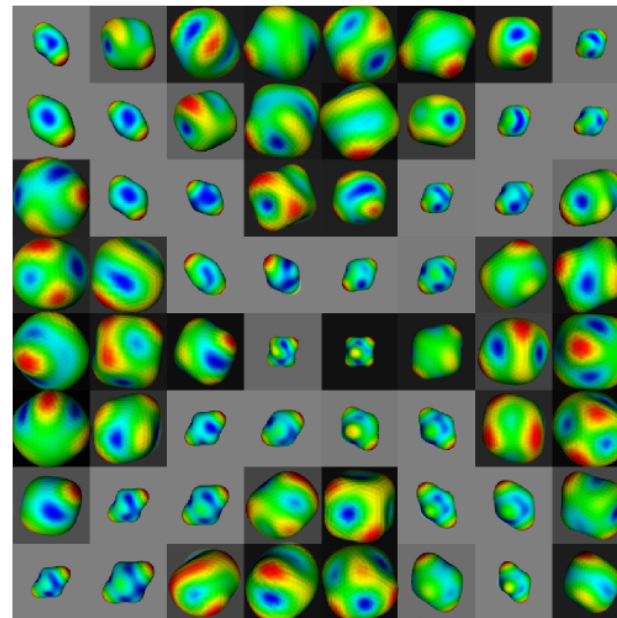
(d) ODF-SA



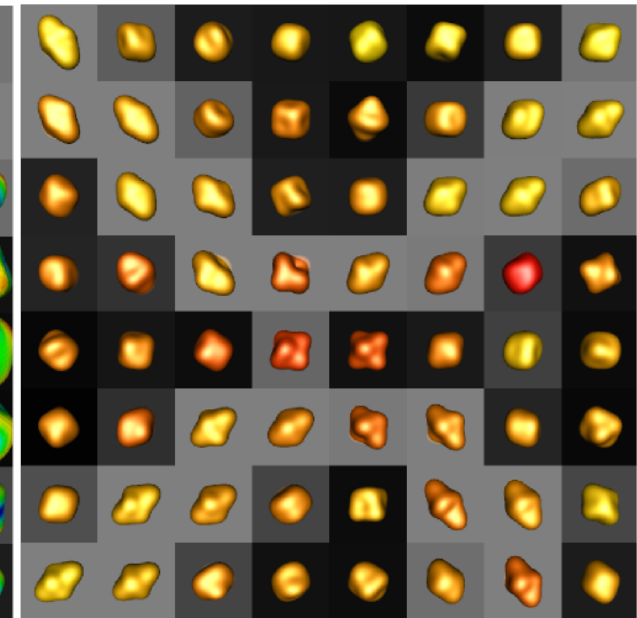
(a)



(b)



(c)



(d)

Introduction

1. **Think Positive: 4th Order Symmetric Tensors**
 - A Riemannian Approach
 - **A Polynomial Approach (Ternary Quartic)**
2. Fast and Analytical EAP Approximation from a 4th Order Tensor
3. Geometric Features from Spherical Diffusion Functions
 - Maxima Extraction
 - Peak Fractional Anisotropy

★ Riemannian approach on GDTI won't permit:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \implies ADC = ag_1^4 + bg_2^4 + cg_3^4$$

[Barmpoutis 2009]

★ Unreachable tensors (swelling effect?)

★ Matrix representation has to be *rank-deficient*

[Barmpoutis 2007,2009, Ghosh 2009]

★ A Homogeneous Polynomial

$$ADC = \sum_{j_1=1}^3 \sum_{j_2=1}^3 \sum_{j_3=1}^3 \sum_{j_4=1}^3 D_{j_1 j_2 j_3 j_4} g_{j_1} g_{j_2} g_{j_3} g_{j_4} = \sum_{m+n+p=4} D_{mnp} g_1^m g_2^n g_3^p$$

★ Ternary Quartics: think positive

Theorem (Hilbert 1888)

A *non-negative* real ternary quartic form $f(x, y, z)$ is the sum of three squares of quadratic forms.

[Barmpoutis 2007,2009, Ghosh 2009]

★ A Homogeneous Polynomial

$$ADC = \sum_{j_1=1}^3 \sum_{j_2=1}^3 \sum_{j_3=1}^3 \sum_{j_4=1}^3 D_{j_1 j_2 j_3 j_4} g_{j_1} g_{j_2} g_{j_3} g_{j_4} = \sum_{m+n+p=4} D_{mnp} g_1^m g_2^n g_3^p \geq 0$$

★ Ternary Quartics: think positive

Theorem (Hilbert 1888)

A *non-negative* real ternary quartic form $f(x, y, z)$ is the sum of three squares of quadratic forms.

★ Gram Matrix

$$ADC = (\mathbf{v}^T \mathbf{q}_1)^2 + (\mathbf{v}^T \mathbf{q}_2)^2 + (\mathbf{v}^T \mathbf{q}_3)^2 = \mathbf{v}^T \mathbf{Q} \mathbf{Q}^T \mathbf{v} = \mathbf{v}^T \mathbf{G} \mathbf{v}$$

$$\mathbf{v} = [g_1^2, g_2^2, g_3^2, g_1 g_2, g_1 g_3, g_2 g_3]^T, \quad \mathbf{Q} : 6 \times 3, \quad \mathbf{G} : \text{Rank} - 3$$

[Ghosh 2009]

★ Stejskal-Tanner equation:

$$\begin{aligned} -\frac{1}{b} \ln \left(\frac{S_i}{S_0} \right) &= (\mathbf{v}^T \mathbf{q}_1)^2 + (\mathbf{v}^T \mathbf{q}_2)^2 + (\mathbf{v}^T \mathbf{q}_3)^2 \\ &= [\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T] \begin{bmatrix} \mathbf{v}\mathbf{v}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}\mathbf{v}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{v}\mathbf{v}^T \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} = \mathbf{Q}^T \mathbf{V} \mathbf{Q} \end{aligned}$$

[Ghosh 2009]

★ Stejskal-Tanner equation:

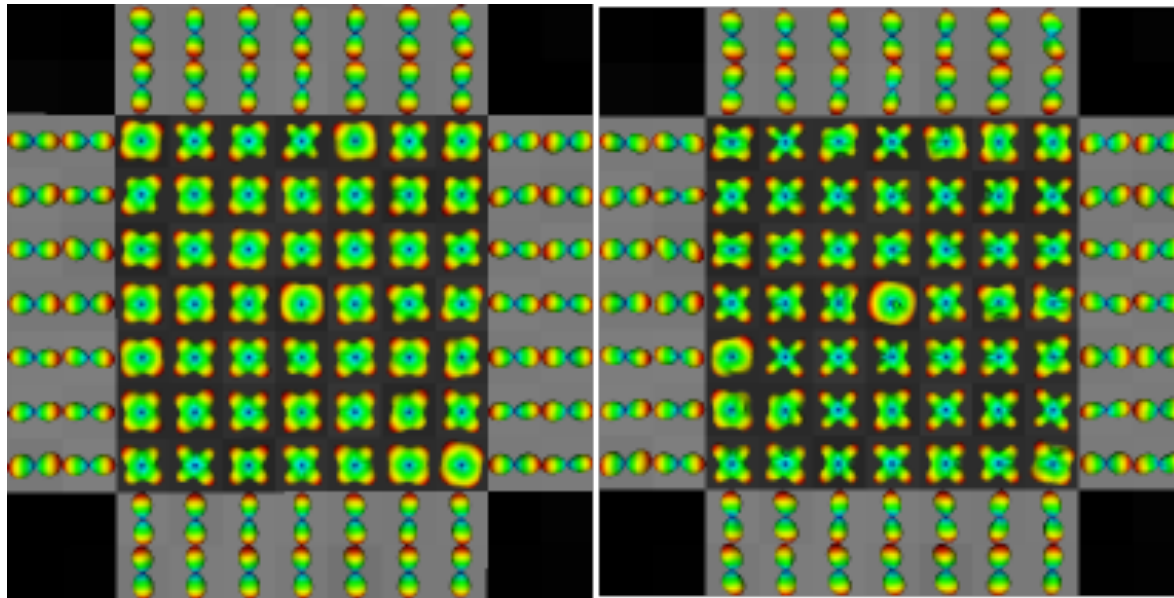
$$\begin{aligned}
 -\frac{1}{b} \ln \left(\frac{S_i}{S_0} \right) &= (\mathbf{v}^T \mathbf{q}_1)^2 + (\mathbf{v}^T \mathbf{q}_2)^2 + (\mathbf{v}^T \mathbf{q}_3)^2 \\
 &= [\mathbf{q}_1^T, \mathbf{q}_2^T, \mathbf{q}_3^T] \begin{bmatrix} \mathbf{v}\mathbf{v}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{v}\mathbf{v}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{v}\mathbf{v}^T \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} = \mathbf{Q}^T \mathbf{V} \mathbf{Q}
 \end{aligned}$$

★ Non-linear optimization:

$$\nabla E(\mathbf{Q}) = \sum_{i=1}^N \left(\frac{1}{b} \log \left(\frac{S_i}{S_0} \right) + \mathbf{Q}^T \mathbf{v}_i \mathbf{Q} \right) (\mathbf{v}_i + \mathbf{v}_i^T) \mathbf{Q}$$

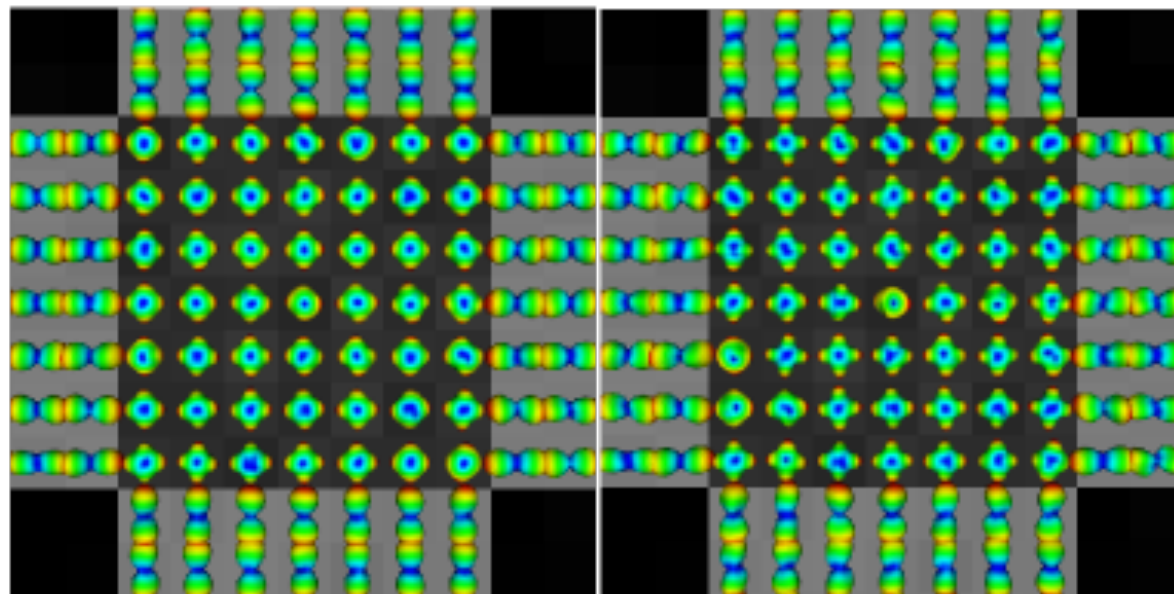
Guarantees *non-negativeness*: $ADC \geq 0$

★ Solves the correct problem



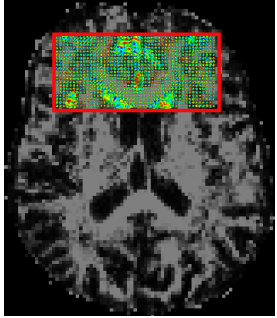
(a)

(b)



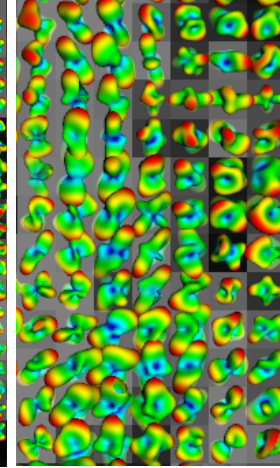
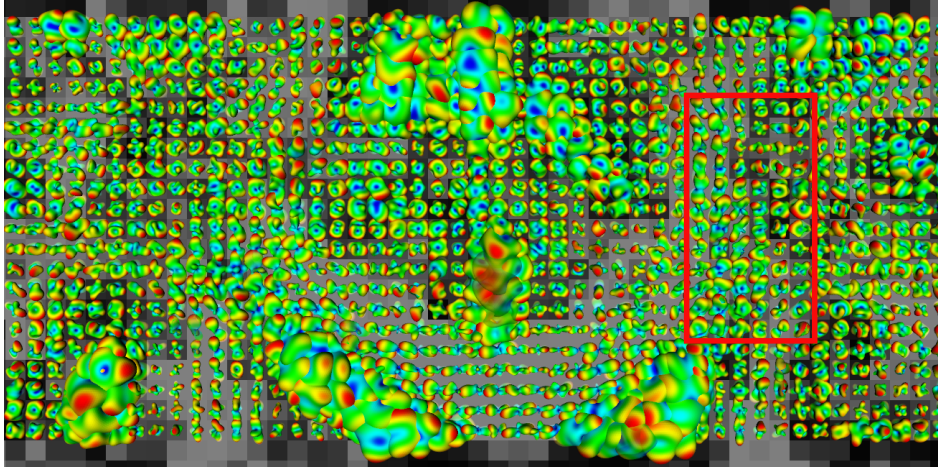
(c)

(d)

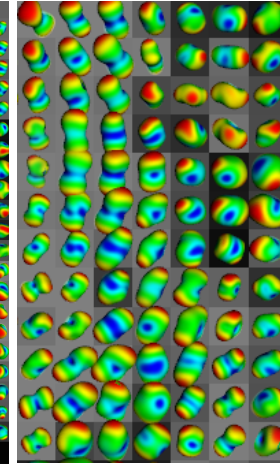
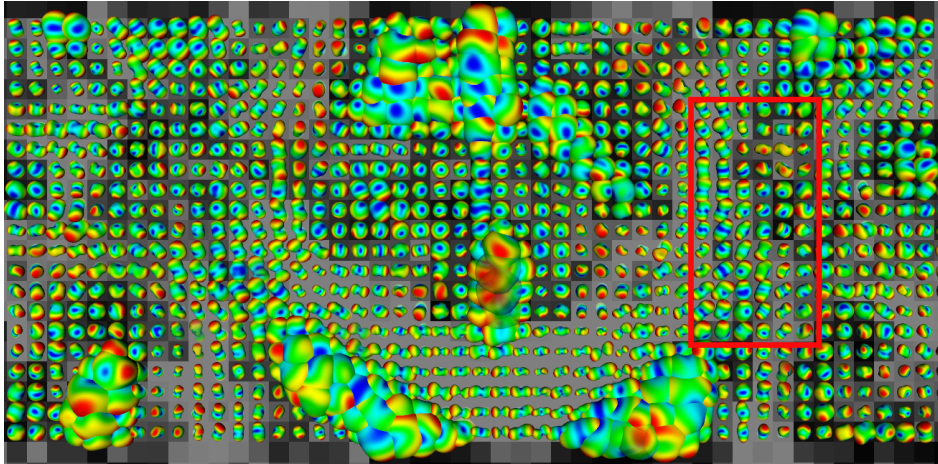


Real data:
[Poupon 2006]

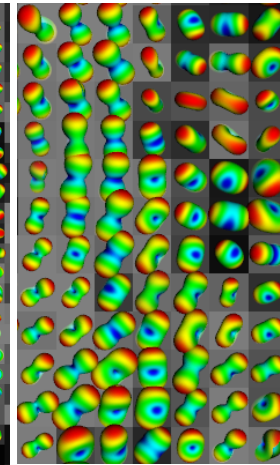
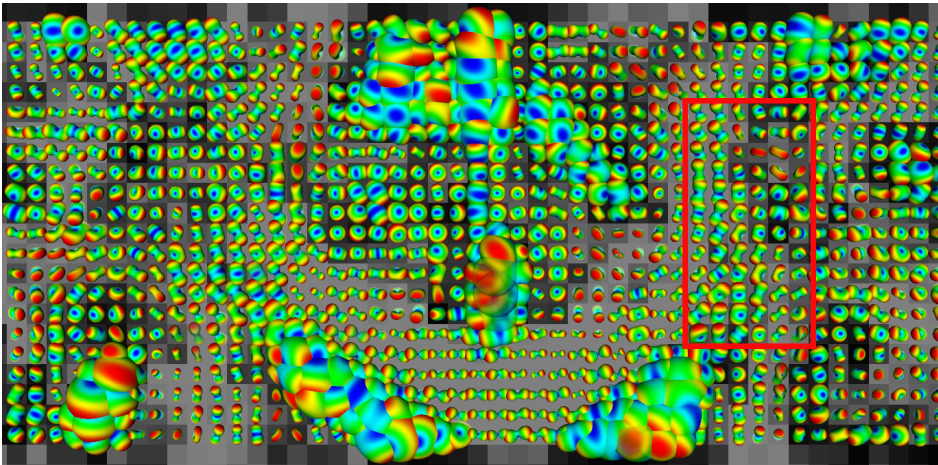
Euclidean LS



Riemannian



Ternary Quartic



(81 dirs)	LS	SH	RM	TQ
Positive	181757	249263	249352	249352
Negative	67595	89	0	0

249352 estimated tensors. ADC along 81 dirs

(987 tensors)	LS	Rm	TQ
Estimation	6s	35s	102s

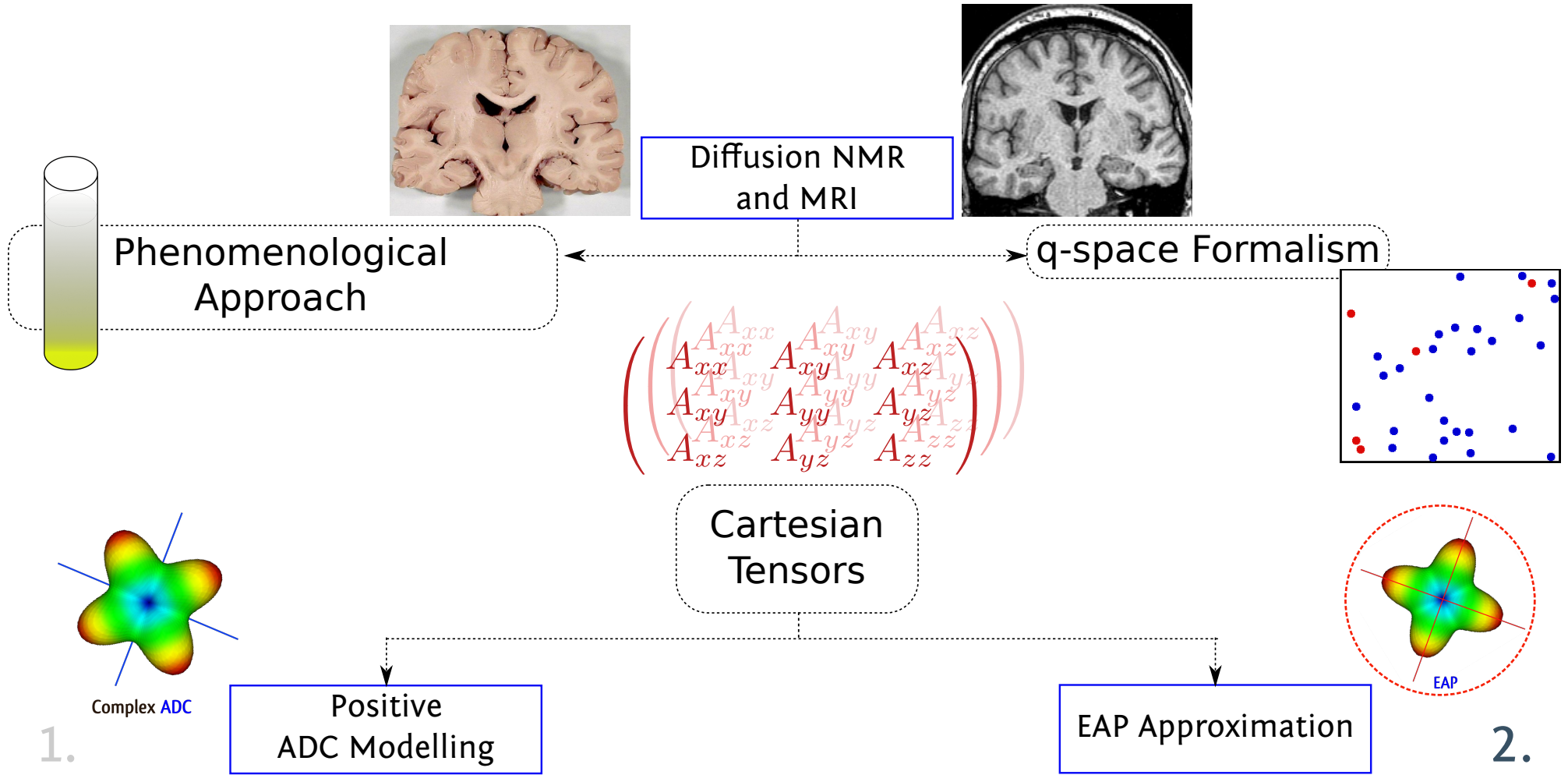
Time for estimating 987 tensors

- ★ Riemannian Approach : [Ghosh 2008]
 - ⦿ 4th order 3D tensor \rightarrow 2nd order 6D tensor
 - ⦿ Guarantees Positive Definiteness
 - ⊖ Solves over-constrained problem

- ★ A Polynomial Approach [Ghosh 2009]
 - ⦿ Hilbert's theorem on Ternary Quartics
 - ⦿ Solves the correct problem
 - ⊖ Guarantees non-negativeness

Introduction

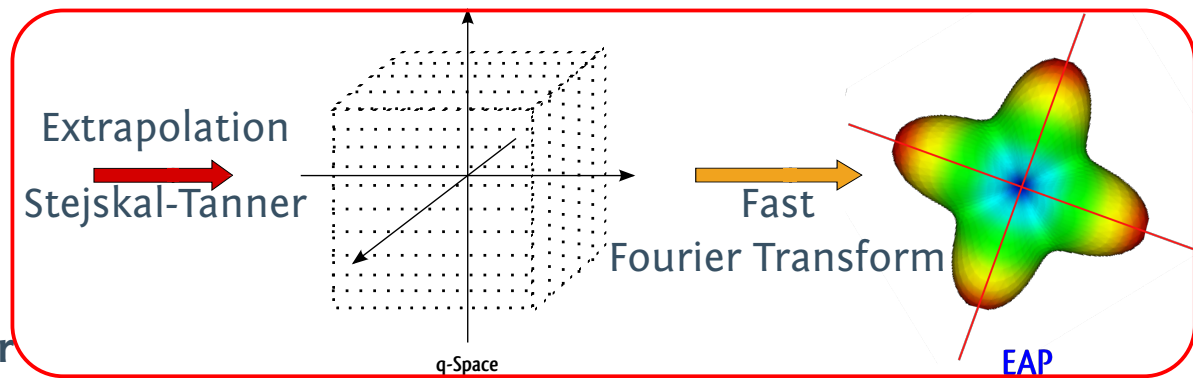
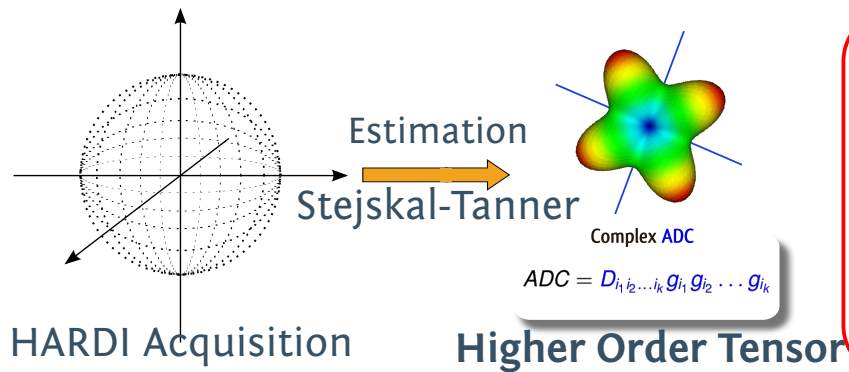
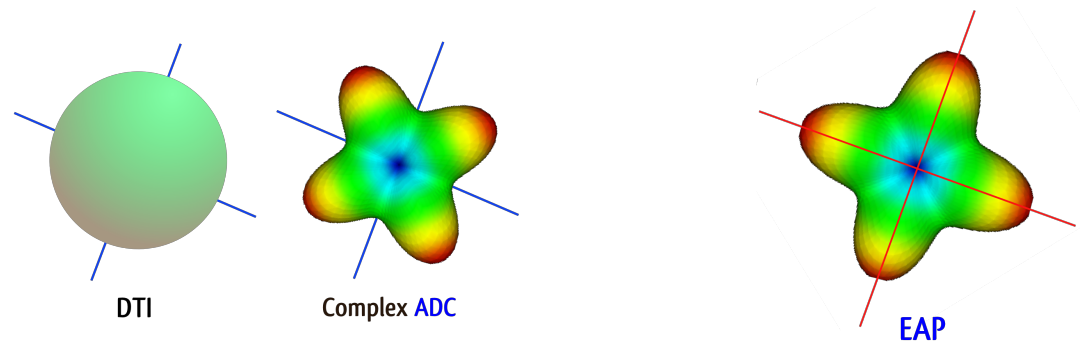
1. Think Positive: 4th Order Symmetric Tensors
 - A Riemannian Approach
 - A Polynomial Approach (Ternary Quartic)
2. Fast and Analytical EAP Approximation from a 4th Order Tensor
3. Geometric Features from Spherical Diffusion Functions
 - Maxima Extraction
 - Peak Fractional Anisotropy



[Ozarslan-2004]

★ Fiber directions?

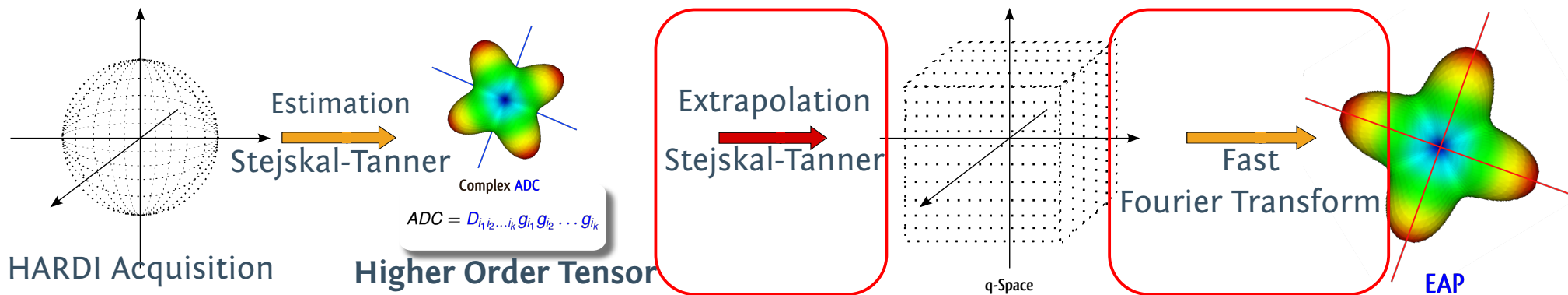
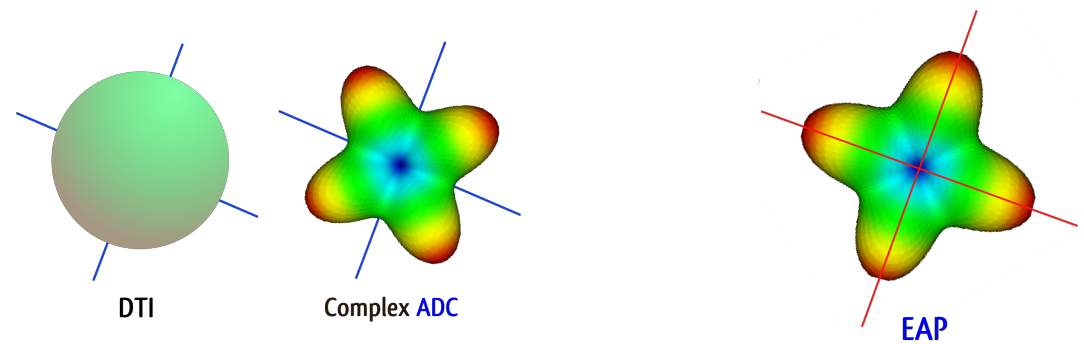
$$ADC = D_{i_1 i_2 \dots i_k} g_{i_1} g_{i_2} \dots g_{i_k}$$



[Ozarslan-2004]

★ Fiber directions?

$$ADC = D_{i_1 i_2 \dots i_k} g_{i_1} g_{i_2} \dots g_{i_k}$$



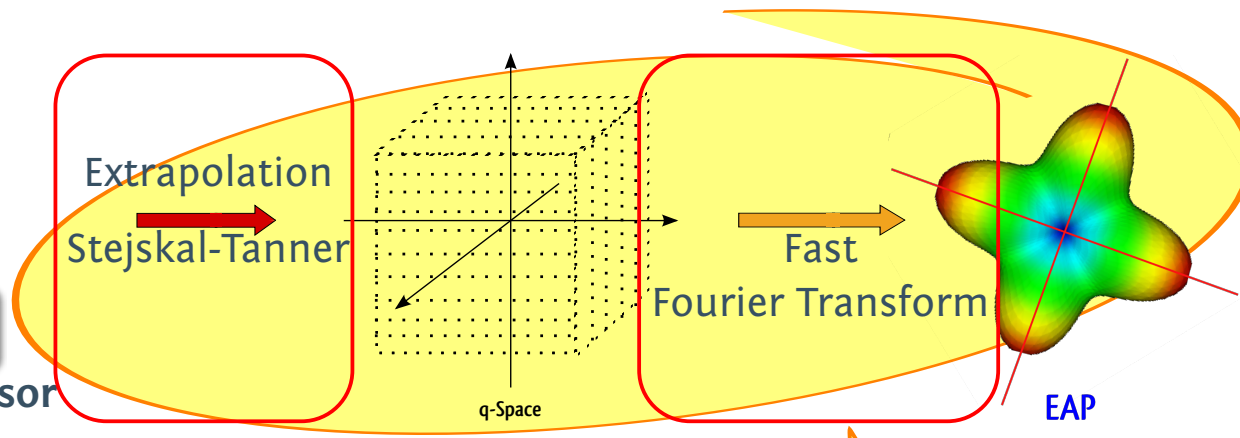
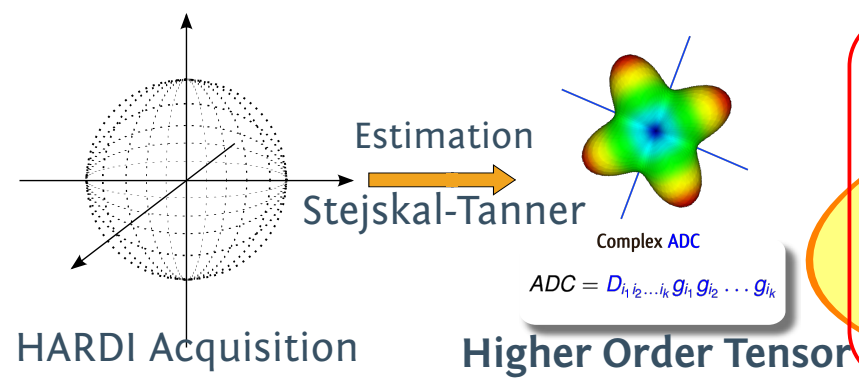
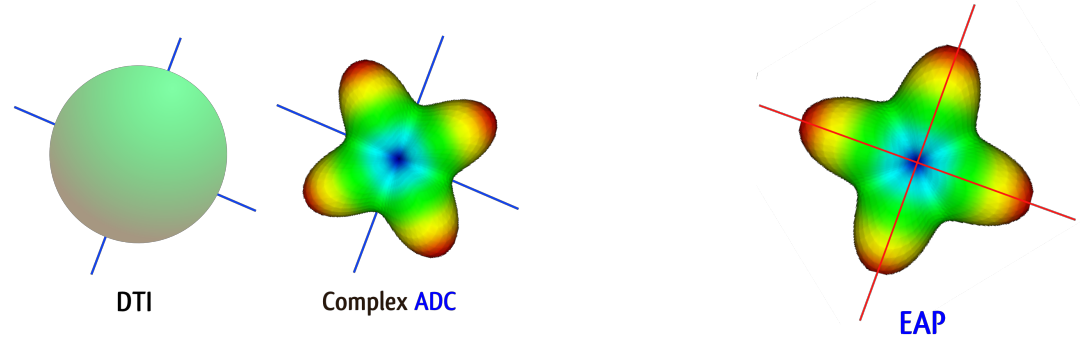
Extrapolation in q -space



Numerical computation

★ Fiber directions?

$$ADC = D_{i_1 i_2 \dots i_k} g_{i_1} g_{i_2} \dots g_{i_k}$$



Can the EAP reconstruction be improved?

$$E(\mathbf{q})_k = e^{-4\pi^2 |\mathbf{q}|^2 t} \sum_{m+n+p=k} D_{mnp} g_1^m g_2^n g_3^p$$

$t = (\Delta - \frac{\delta}{3}); \quad \mathbf{g} = \frac{\mathbf{q}}{|\mathbf{q}|}$

Radial: **order-2**

Angular: **order-k**

$$E(\mathbf{q})_k = e^{-4\pi^2 |\mathbf{q}|^2 t} \sum_{m+n+p=k} D_{mnp} g_1^m g_2^n g_3^p \quad t = (\Delta - \frac{\delta}{3}); \quad \mathbf{g} = \frac{\mathbf{q}}{|\mathbf{q}|}$$

Radial: order- k_1

Angular: order- k_2

★ Generalize separation of radial & angular

$$\begin{aligned} E(\mathbf{q})_{k_1, k_2} &= e^{-4\pi^2 \alpha |\mathbf{q}|^{k_1} t} \sum_{m+n+p=k_2} D_{mnp} g_1^m g_2^n g_3^p \\ &= e^{-4\pi^2 \alpha |\mathbf{q}|^{k_1 - k_2} t} \sum_{m+n+p=k_2} D_{mnp} q_1^m q_2^n q_3^p \end{aligned}$$

$$\alpha = 1 m^{2-k_1}$$

$$E(\mathbf{q})_k = e^{-4\pi^2 |\mathbf{q}|^2 t} \sum_{m+n+p=k} D_{mnp} g_1^m g_2^n g_3^p \quad t = (\Delta - \frac{\delta}{3}); \quad \mathbf{g} = \frac{\mathbf{q}}{|\mathbf{q}|}$$

Radial: order-**k1**

Angular: order-**k2**

★ Generalize separation of radial & angular

$$E(\mathbf{q})_{k,k} = e^{-4\pi^2 \alpha t} \sum_{m+n+p=k} D_{mnp} q_1^m q_2^n q_3^p$$

★ Homogeneous polynomial over \mathbb{R}^3 when **k1=k2=k**

$$E(\mathbf{q})_{k,k} = e\left(-4\pi^2\alpha t \sum D_{mnp} q_1^m q_2^n q_3^p\right)$$

$$E(\mathbf{q})_{k,k} = e^{\left(-4\pi^2\alpha t \sum D_{mnp} q_1^m q_2^n q_3^p\right) + 2\pi^2\beta(q_1^2 + q_2^2 + q_3^2)} \times e^{-2\pi^2\beta(q_1^2 + q_2^2 + q_3^2)}$$

$$\beta = 1m^2$$

$$\begin{aligned} E(\mathbf{q})_{k,k} &= e^{(-4\pi^2\alpha t \sum D_{mnp} q_1^m q_2^n q_3^p) + 2\pi^2\beta(q_1^2 + q_2^2 + q_3^2)} \times \\ & \quad e^{-2\pi^2\beta(q_1^2 + q_2^2 + q_3^2)} \\ &= h(\mathbf{q}) e^{-2\pi^2\beta(q_1^2 + q_2^2 + q_3^2)} \end{aligned}$$

$$\begin{aligned} E(\mathbf{q})_{k,k} &= e\left(-4\pi^2\alpha t \sum D_{mnp}q_1^m q_2^n q_3^p\right) \\ &= h(\mathbf{q})e^{-2\pi^2\beta(q_1^2+q_2^2+q_3^2)} \end{aligned}$$

★ Taylor expansion: $h_n(\mathbf{q}) = \sum_{l+s+u < n} C_{l,s,u} q_1^l q_2^s q_3^u$

$$E(\mathbf{q})_{k,k}^{(n)} = \underbrace{\left(\sum_{l+s+u < n} C_{l,s,u} q_1^l q_2^s q_3^u\right)}_{\text{Polynomial}} e^{-2\pi^2\beta(q_1^2+q_2^2+q_3^2)}$$

$$C_{l,s,u} q_1^l q_2^s q_3^u \times e^{-2\pi^2 \beta (q_1^2 + q_2^2 + q_3^2)}$$

★ Separable monomial

$$C_{l,s,u} q_1^l q_2^s q_3^u \times e^{-2\pi^2 \beta (q_1^2 + q_2^2 + q_3^2)}$$

★ Separable monomial

★ Fourier Transform (1D)

$$\mathbf{F}\{x^n e^{-2\pi^2 x^2}\}(t) = \left(\frac{i}{2\pi}\right)^n He_n(t) \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$$

★ Hermite Polynomial

$$P(\mathbf{r})_{k,k}^{(n)} = e^{\frac{-1}{2\beta}(r_1^2 + r_2^2 + r_3^2)} \times \left(\sum_{l+s+u < n} i^{l+s+u} C_{l,s,u} He_l(r_1) He_s(r_2) He_u(r_3) \right)$$

$$E(\mathbf{q})_{k,k} = e\left(-4\pi^2 \alpha t \sum D_{mnp} q_1^m q_2^n q_3^p\right)$$



$$E(\mathbf{q})_{k,k}^{(n)} = \left(\sum_{l+s+u < n} C_{l,s,u} q_1^l q_2^s q_3^u\right) e^{-2\pi^2 \beta (q_1^2 + q_2^2 + q_3^2)}$$



$$P(\mathbf{r})_{k,k}^{(n)} = e^{\frac{-1}{2\beta} (r_1^2 + r_2^2 + r_3^2)} \times \left(\sum_{l+s+u < n} i^{l+s+u} C_{l,s,u} He_l(r_1) He_s(r_2) He_u(r_3) \right)$$

★ $P(\mathbf{r})_{4,4}^{(5)}$

★ Symbolic computation + code generation (Maple)

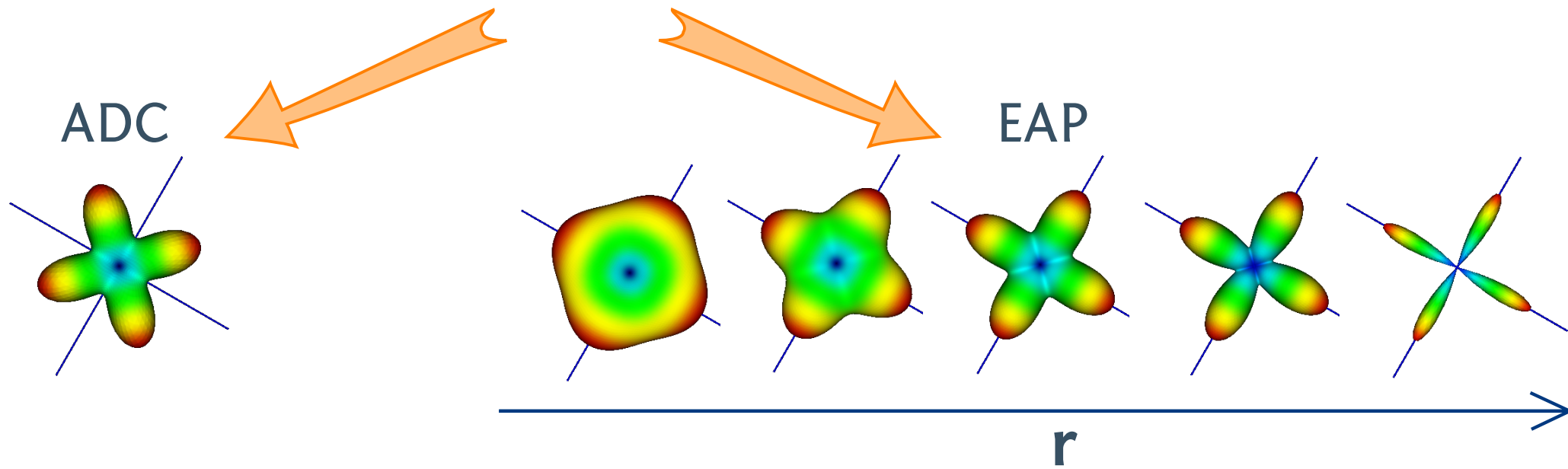
```
pdf = -std::pow(0.3141592654e1, 0.3e1 / 0.2e1) + (-0.35e2 + (double) (16 * std::abs(b) * M(3, 3)) +
(double) (8 * std::abs(b) * M(2, 0)) + (double) (16 * std::abs(b) * M(4, 4)) + (double) (8 *
std::abs(b) * M(0, 2)) - std::pow(vX[0], 0.4e1) - 0.2e1 * std::pow(vX[0], 0.2e1) * std::pow(vX[2],
0.2e1) + 0.8e1 * (double) std::abs(b) * (double) M(1, 0) * std::pow(vX[0], 0.2e1) * std::pow(vX[1],
0.2e1) + 0.8e1 * (double) std::abs(b) * (double) M(0, 1) * std::pow(vX[0], 0.2e1) * std::pow(vX[1],
0.2e1) + 0.16e2 * (double) std::abs(b) * (double) M(3, 3) * std::pow(vX[0], 0.2e1) * std::pow(vX[1],
0.2e1) - 0.48e2 * (double) std::abs(b) * (double) M(0, 0) * std::pow(vX[0], 0.2e1) - 0.8e1 *
(double) std::abs(b) * (double) M(1, 0) * std::pow(vX[0], 0.2e1) - 0.8e1 * (double) std::abs(b) *
(double) M(0, 1) * std::pow(vX[0], 0.2e1) - 0.2e1 * std::pow(vX[0], 0.2e1) * std::pow(vX[1], 0.2e1)
+ 0.8e1 * sqrt(0.2e1) * vX[2] * vX[0] + (double) std::abs(b) * (double) M(1, 4) * std::pow(vX[1],
0.2e1) + 0.8e1 * sqrt(0.2e1) * vX[2] * (double) std::abs(b) * (double) M(1, 5) * std::pow(vX[1],
0.3e1) + 0.16e2 * vX[2] * vX[0] * (double) std::abs(b) * (double) M(5, 3) * std::pow(vX[1], 0.2e1) -
0.16e2 * (double) std::abs(b) * (double) M(3, 3) * std::pow(vX[0], 0.2e1) + 0.14e2 * std::pow(vX[1],
0.2e1) - std::pow(vX[1], 0.4e1) - 0.24e2 * sqrt(0.2e1) * vX[0] * (double) std::abs(b) * (double)
M(1, 3) * vX[1] + 0.8e1 * sqrt(0.2e1) * vX[0] * (double) std::abs(b) * (double) M(1, 3) *
std::pow(vX[1], 0.3e1) - 0.24e2 * sqrt(0.2e1) * vX[0] * (double) std::abs(b) * (double) M(3, 1) *
vX[1] + 0.8e1 * sqrt(0.2e1) * vX[0] * (double) std::abs(b) * (double) M(3, 1) * std::pow(vX[1],
0.3e1) + 0.8e1 * sqrt(0.2e1) * vX[2] * vX[1] * (double) std::abs(b) * (double) M(0, 5) *
std::pow(vX[0], 0.2e1) - 0.24e2 * sqrt(0.2e1) * vX[2] * (double) std::abs(b) * (double) M(4, 0) *
vX[0] + 0.16e2 * vX[2] * vX[1] * (double) std::abs(b) * (double) M(3, 4) * std::pow(vX[0], 0.2e1) +
0.8e1 * sqrt(0.2e1) * vX[1] * (double) std::abs(b) * (double) M(0, 3) * std::pow(vX[0], 0.3e1) -
0.24e2 * sqrt(0.2e1) * vX[1] * (double) std::abs(b) * (double) M(0, 3) * vX[0] + 0.8e1 * sqrt(0.2e1)
* vX[1] * (double) std::abs(b) * (double) M(3, 0) * std::pow(vX[0], 0.3e1) - 0.24e2 * sqrt(0.2e1) *
vX[1] * (double) std::abs(b) * (double) M(3, 0) * vX[0] + (double) (8 * std::abs(b) * M(1, 2)) +
(double) (8 * std::abs(b) * M(2, 1)) + (double) (16 * std::abs(b) * M(5, 5)) + (double) (24 *
std::abs(b) * M(0, 0)) + (double) (24 * std::abs(b) * M(1, 1)) + (double) (24 * std::abs(b) * M(2,
2)) + (double) (8 * std::abs(b) * M(1, 0)) + (double) (8 * std::abs(b) * M(0, 1)) - 0.8e1 * (double)
std::abs(b) * (double) M(0, 1) * std::pow(vX[1], 0.2e1) - 0.16e2 * (double) std::abs(b) * (double)
```

★ Compiled C-code

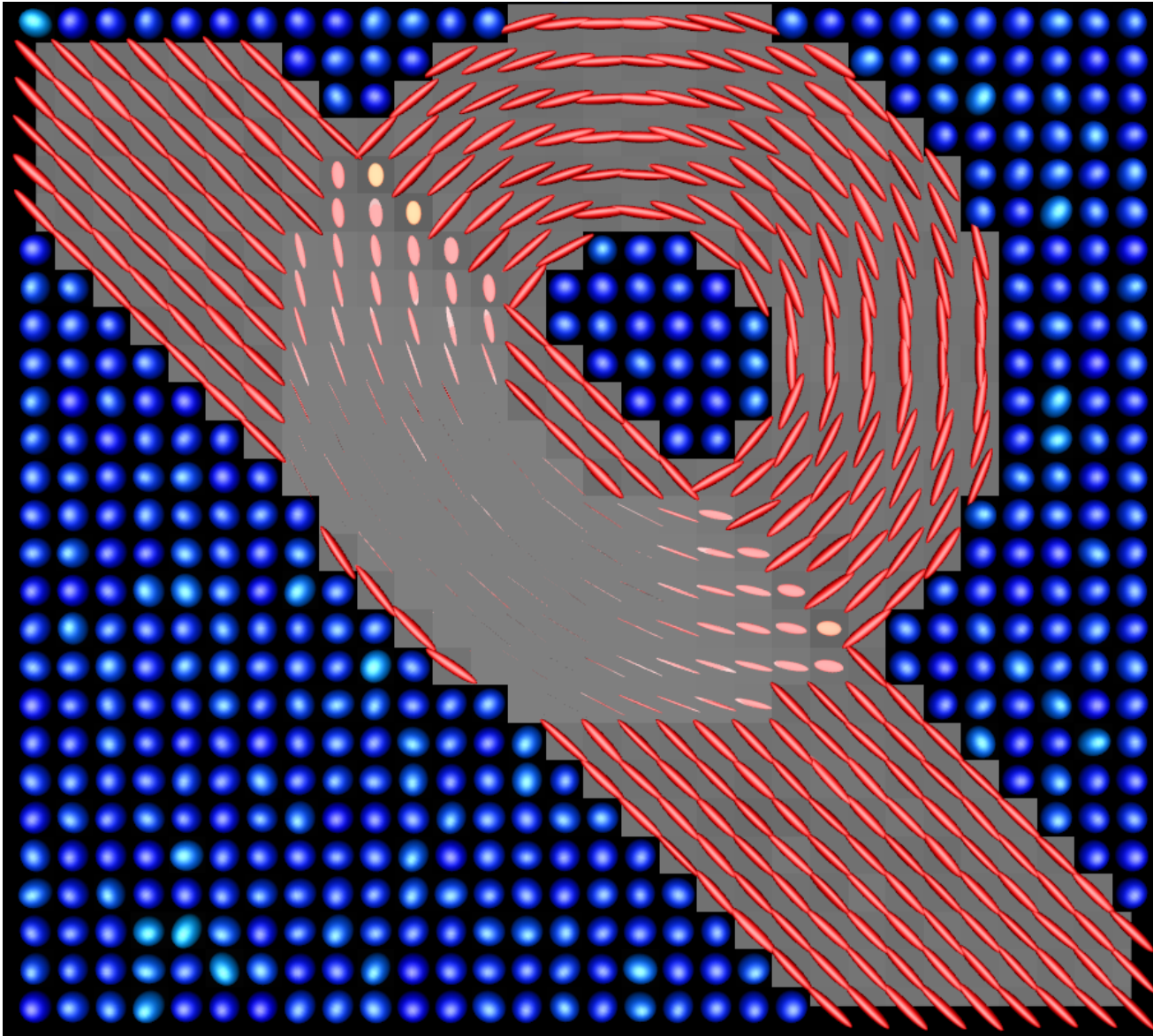
- ★ Multi-tensor model with diagonal:

$$[1390, 355, 355] \times 10^{-6} \text{ mm}^2/s$$

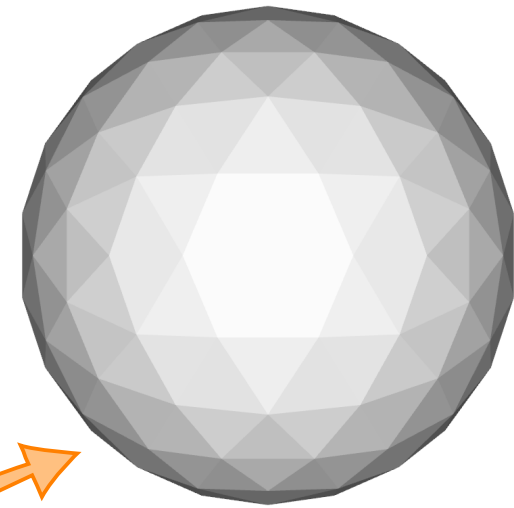
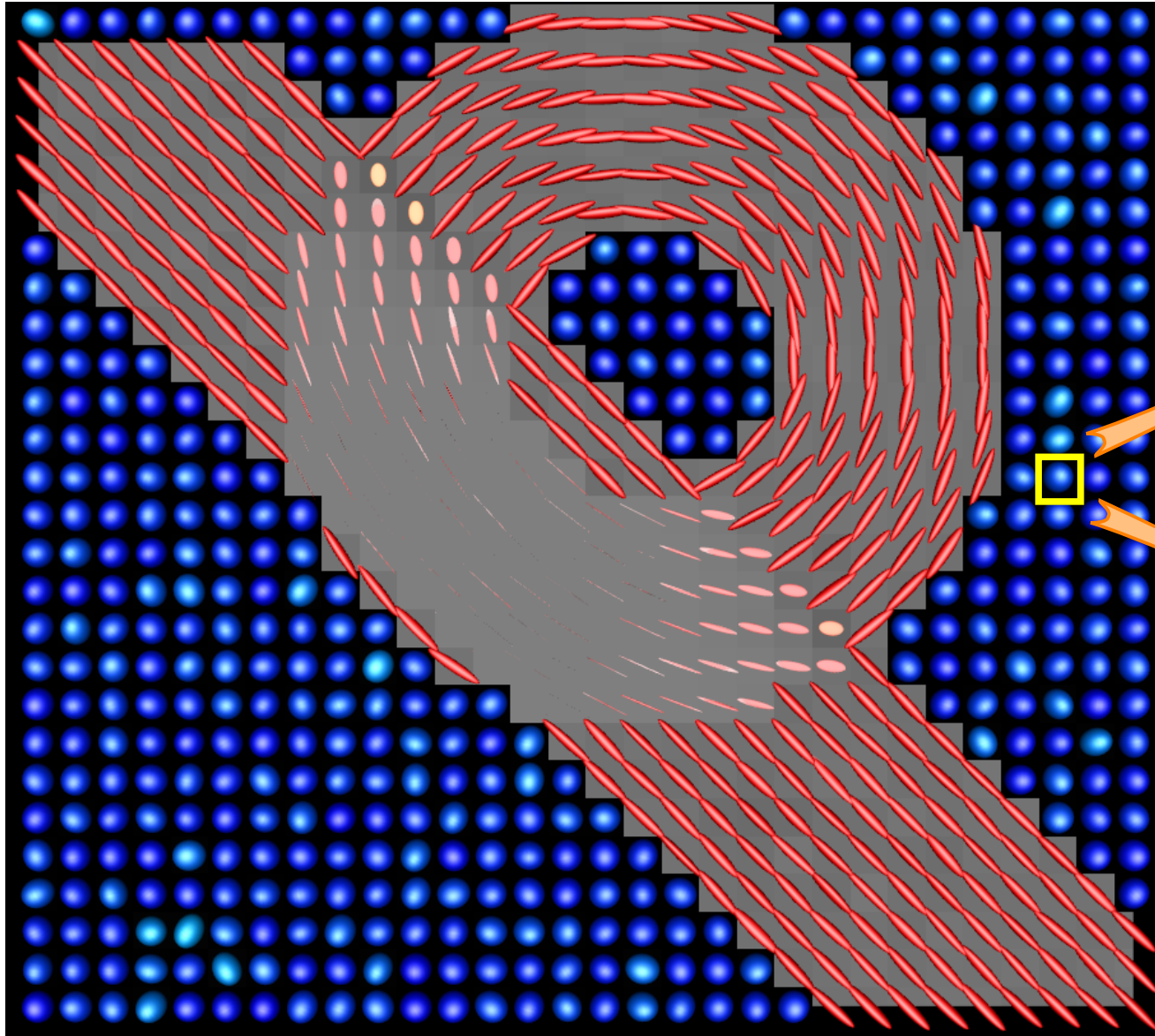
- ★ Complementary information



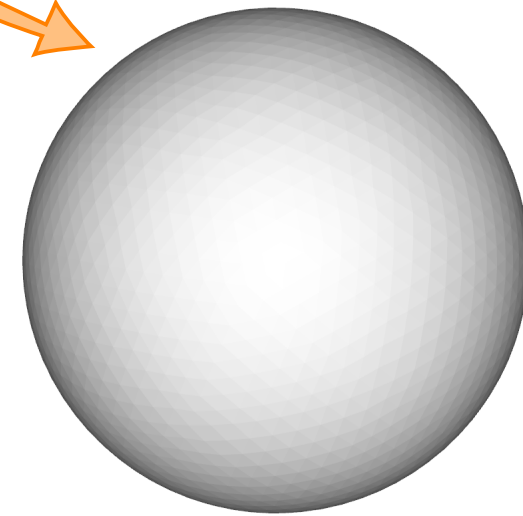
★ DTI



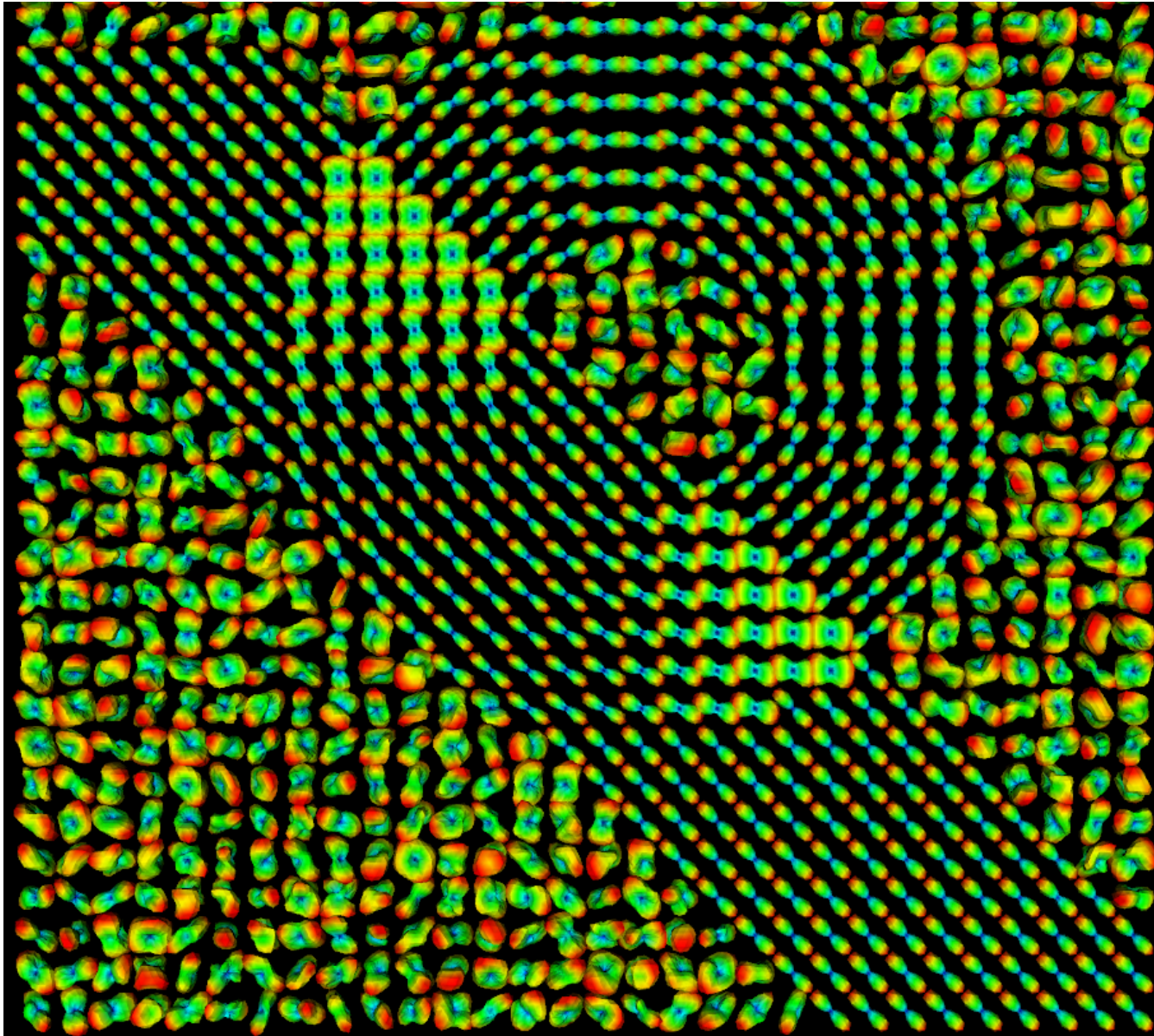
★ DTI



162 vertices



2562 vertices



G-DTI with Numerical Fourier Transform

- ★ 162 vertices
- ★ 30 x 30 voxels
- ⊖ Time: ~ 10 mins



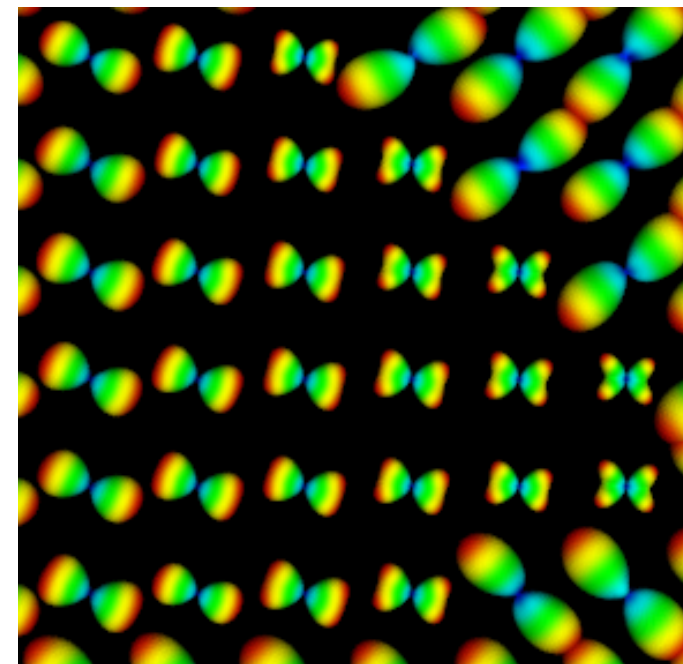
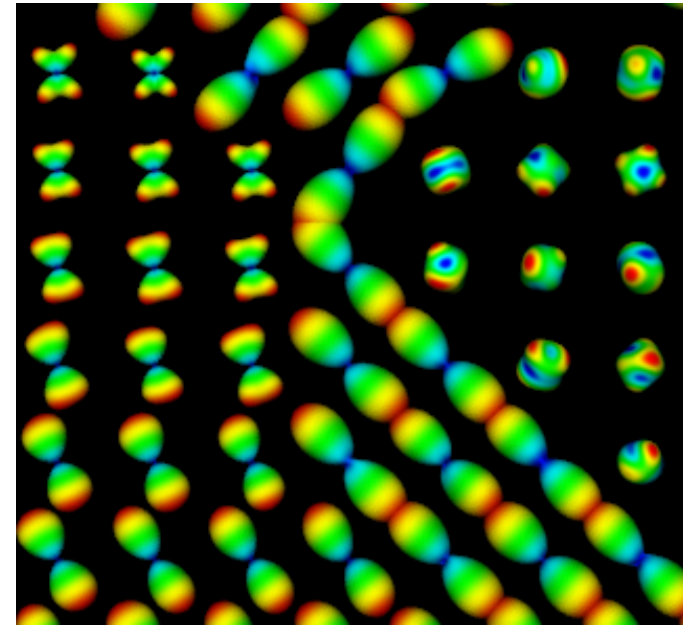
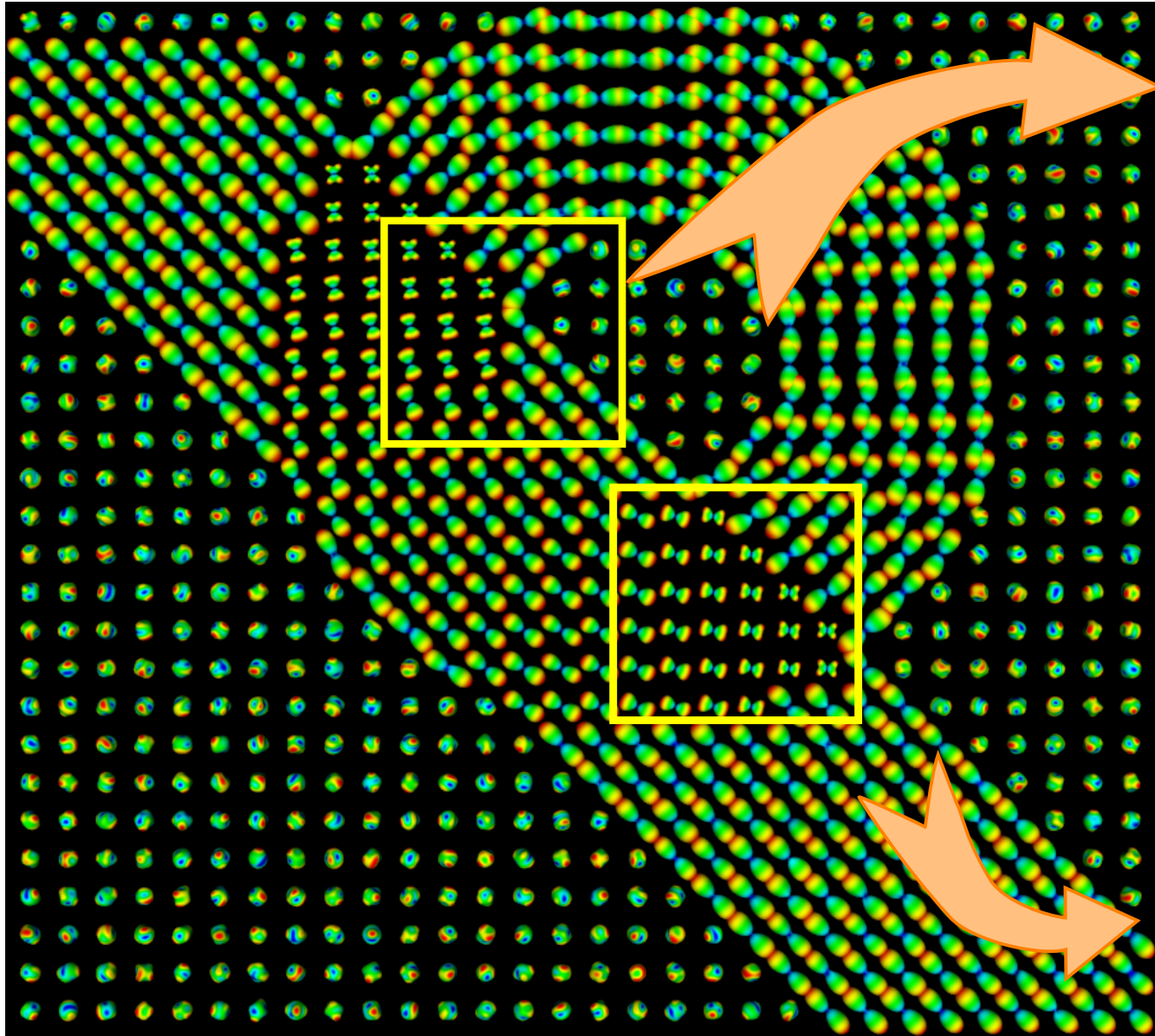
G-DTI with
Polynomial EAP
Approximation

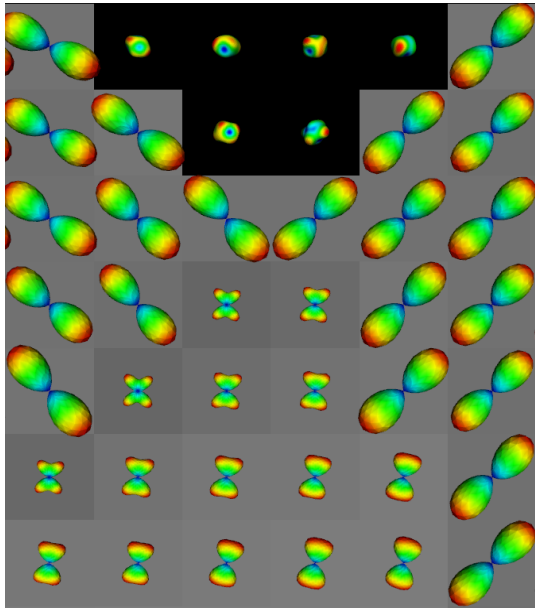
★ 2562 vertices

★ 30 x 30 voxels

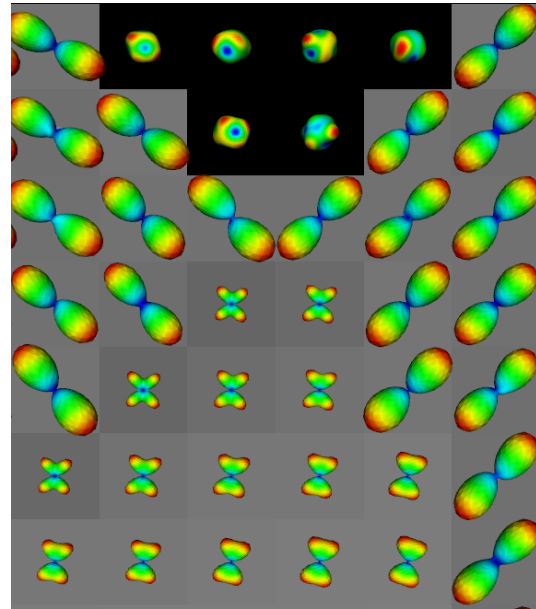
★ Time: ~ 1 min

★ G-DTI with EAP approximation

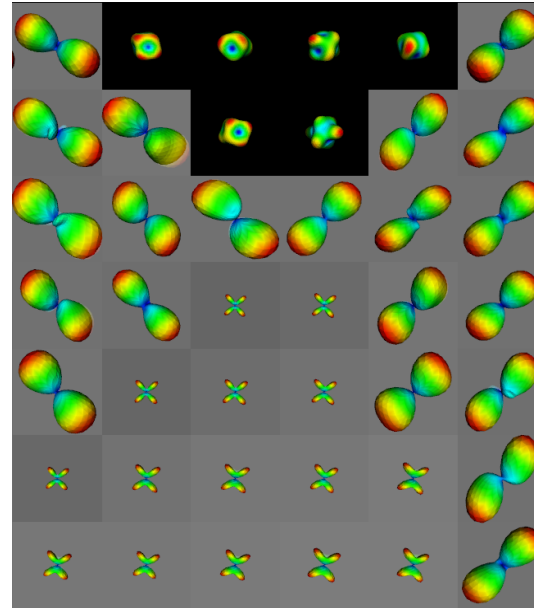




(a)



(b)=(d)



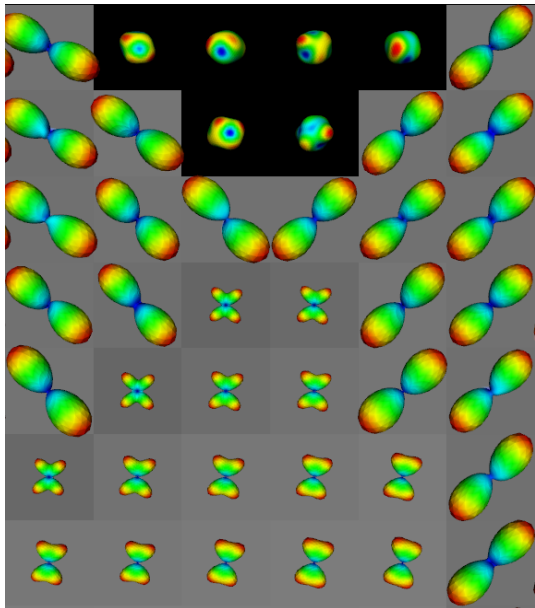
(c)

$$|\mathbf{r}| = 16\mu m$$

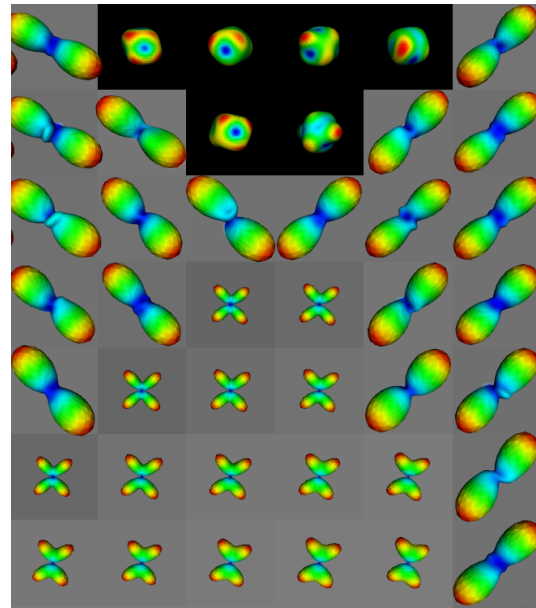
(a) $P(\mathbf{r})_{4,4}^{(5)}$

(b) $P(\mathbf{r})_{4,4}^{(7)}$

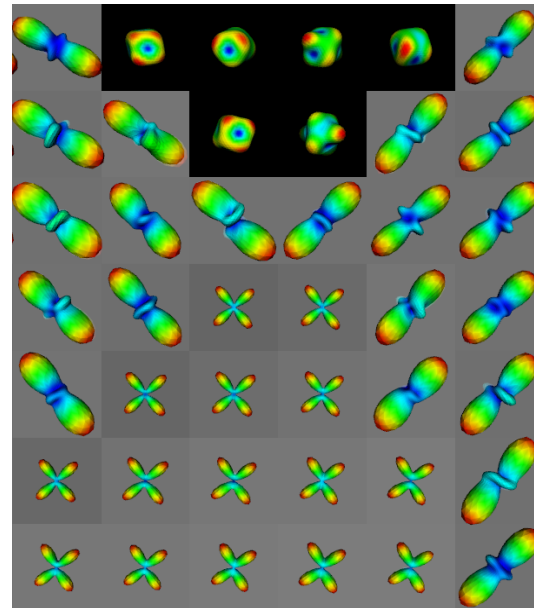
(c) $P(\mathbf{r})_{4,4}^{(9)}$



(d)=(b)



(e)



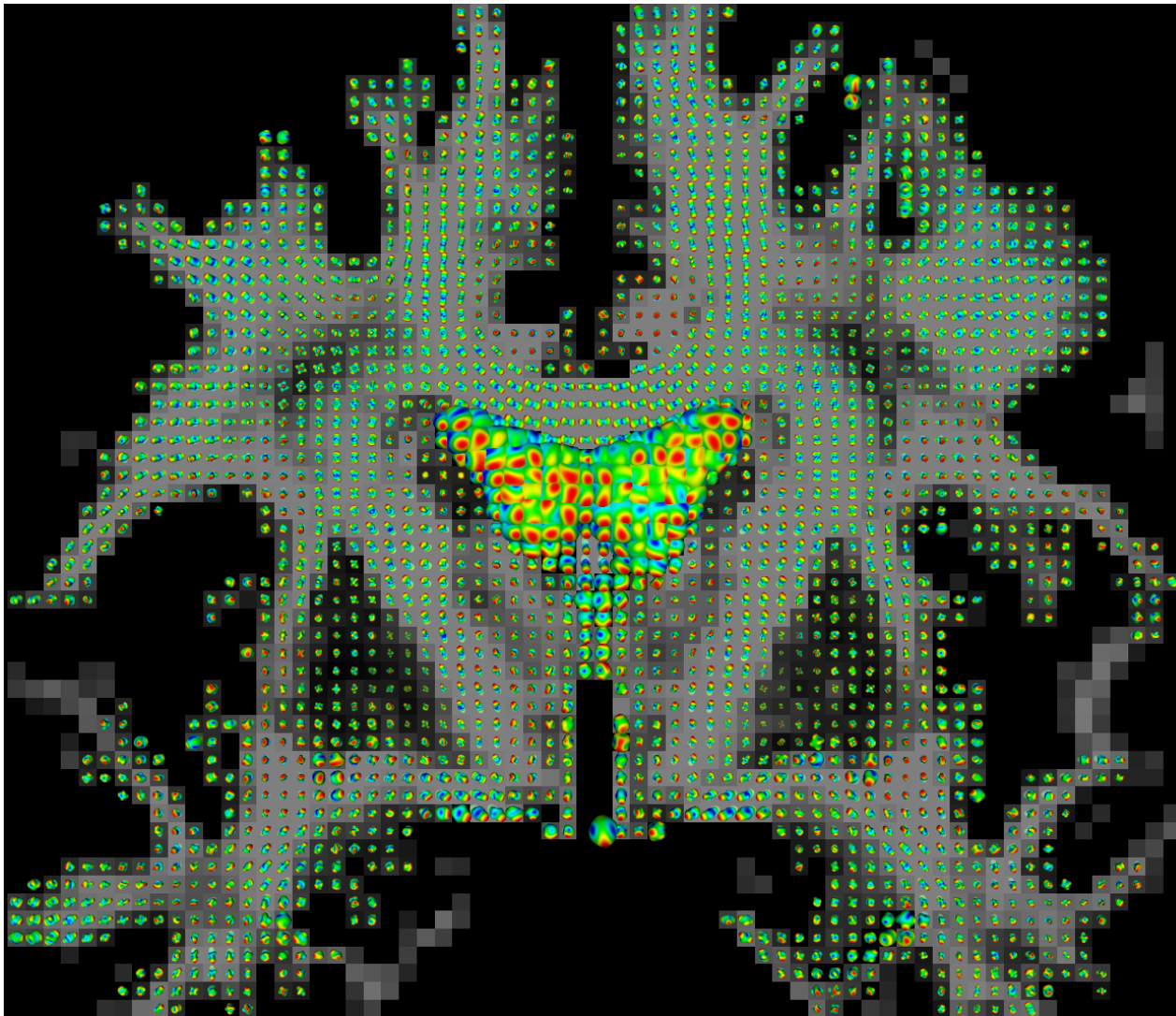
(f)

$$n = 7 \quad P(\mathbf{r})_{4,4}^{(7)}$$

(d) $|\mathbf{r}| = 16\mu m$

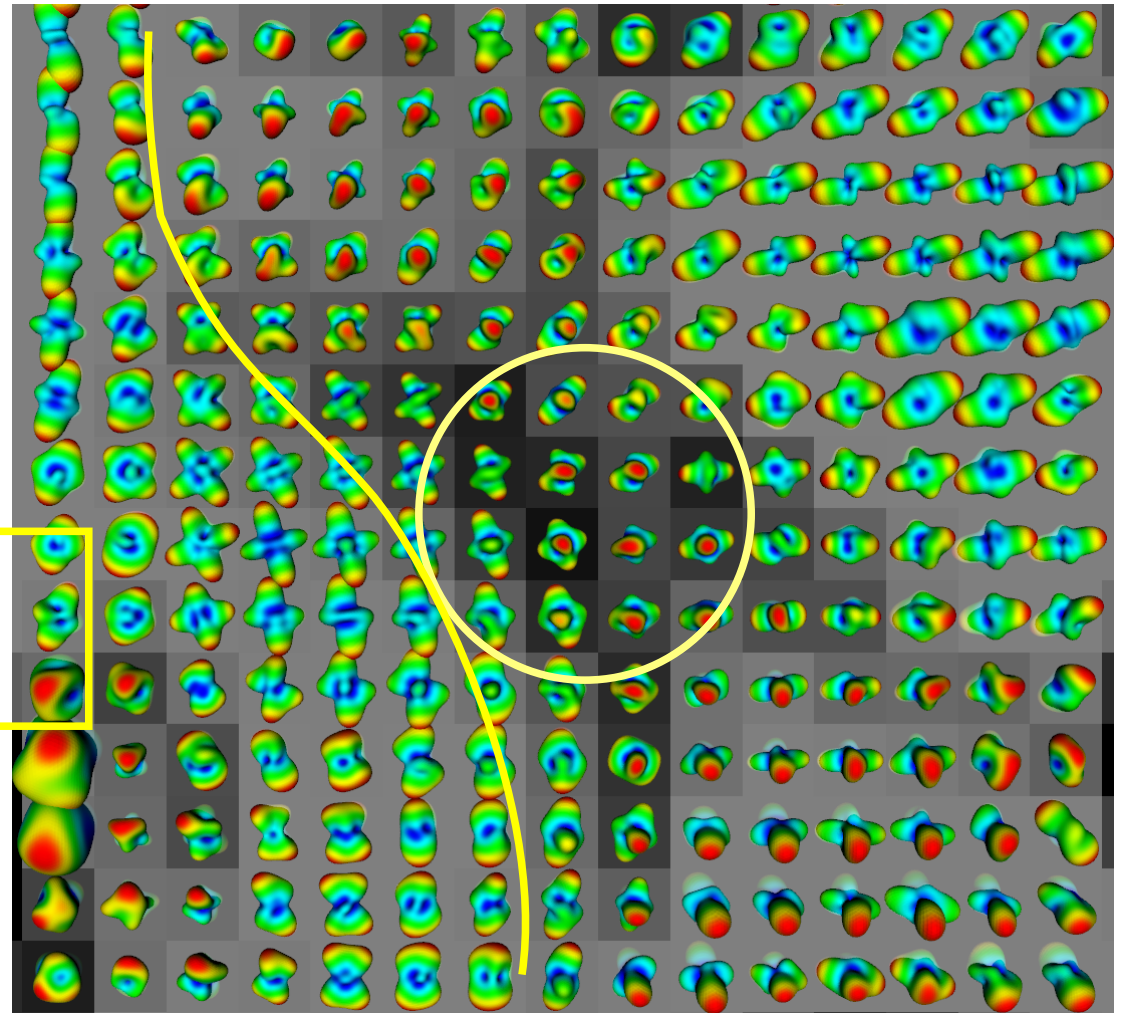
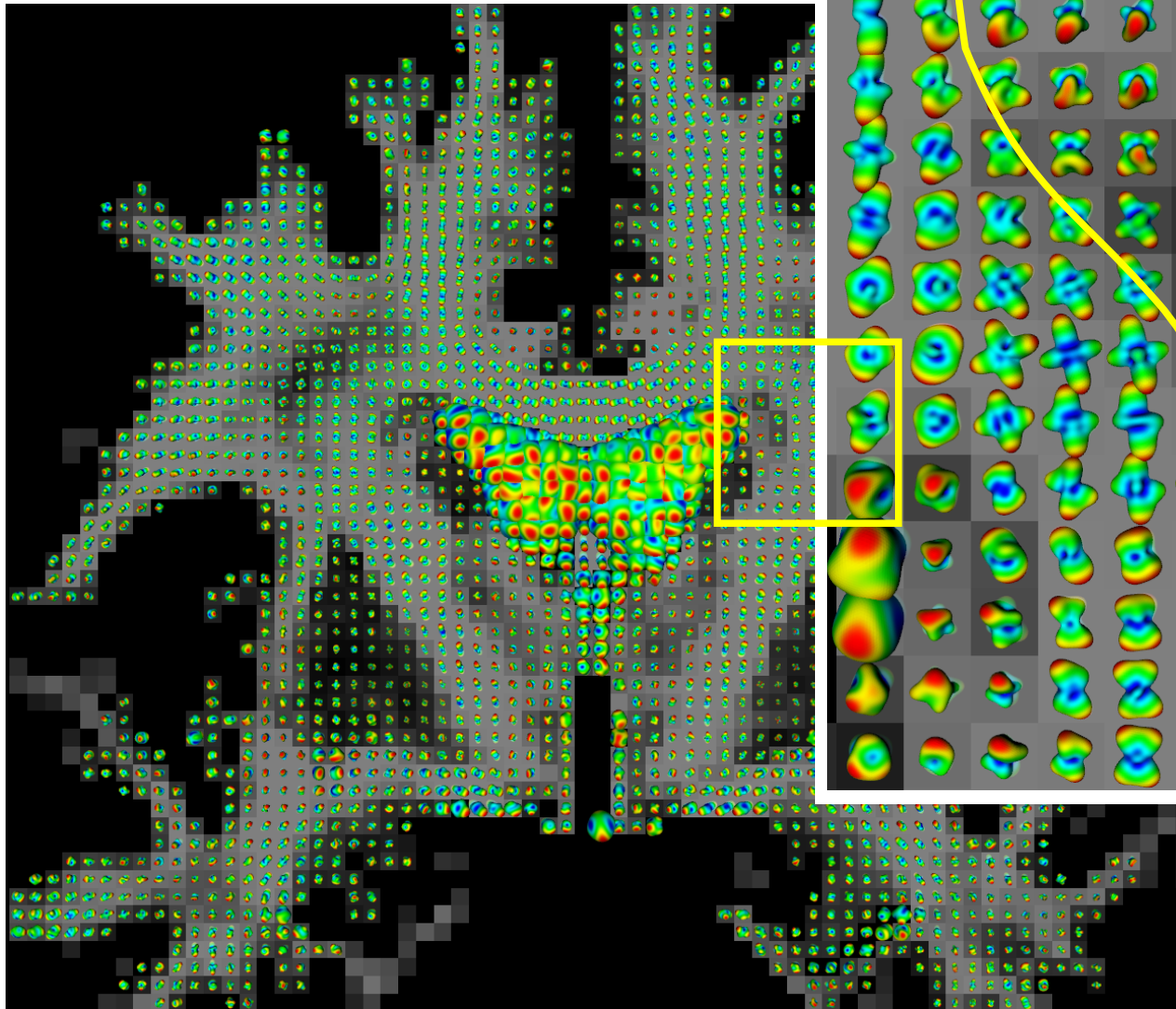
(e) $|\mathbf{r}| = 18\mu m$

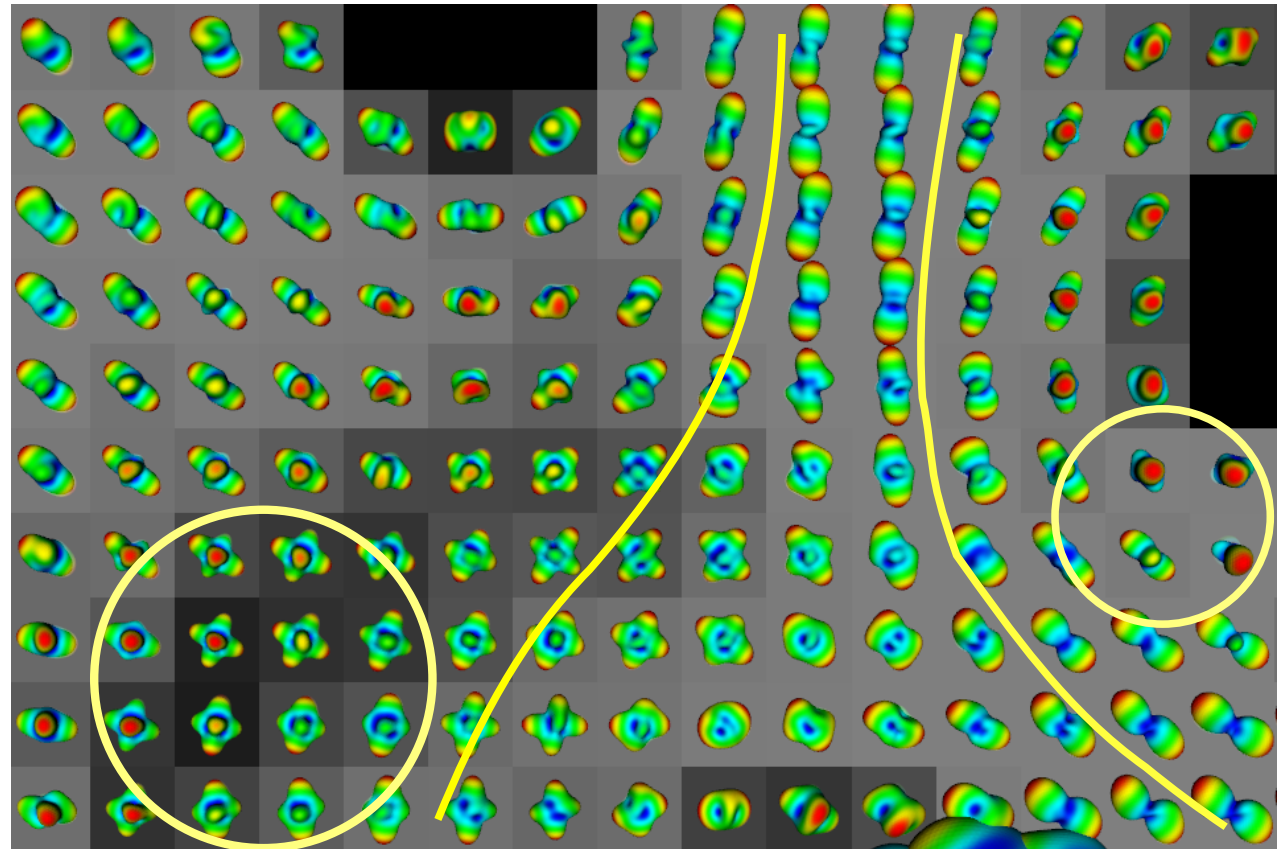
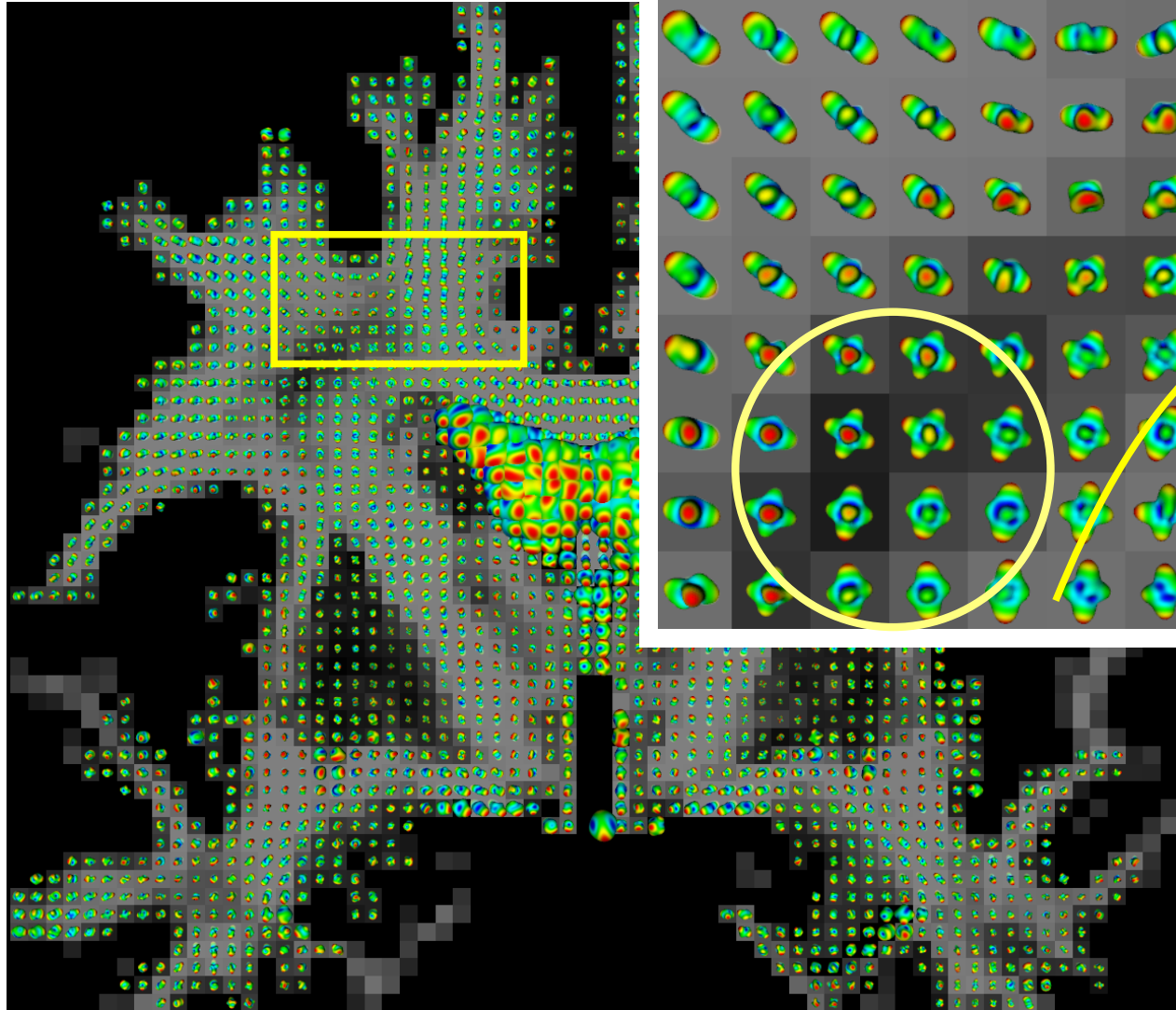
(f) $|\mathbf{r}| = 20\mu m$



- ★ 3T Siemens
- ★ 60 gradient directions
- ★ $b = 1000\text{s}/\text{mm}^2$

[Anwander-2008]



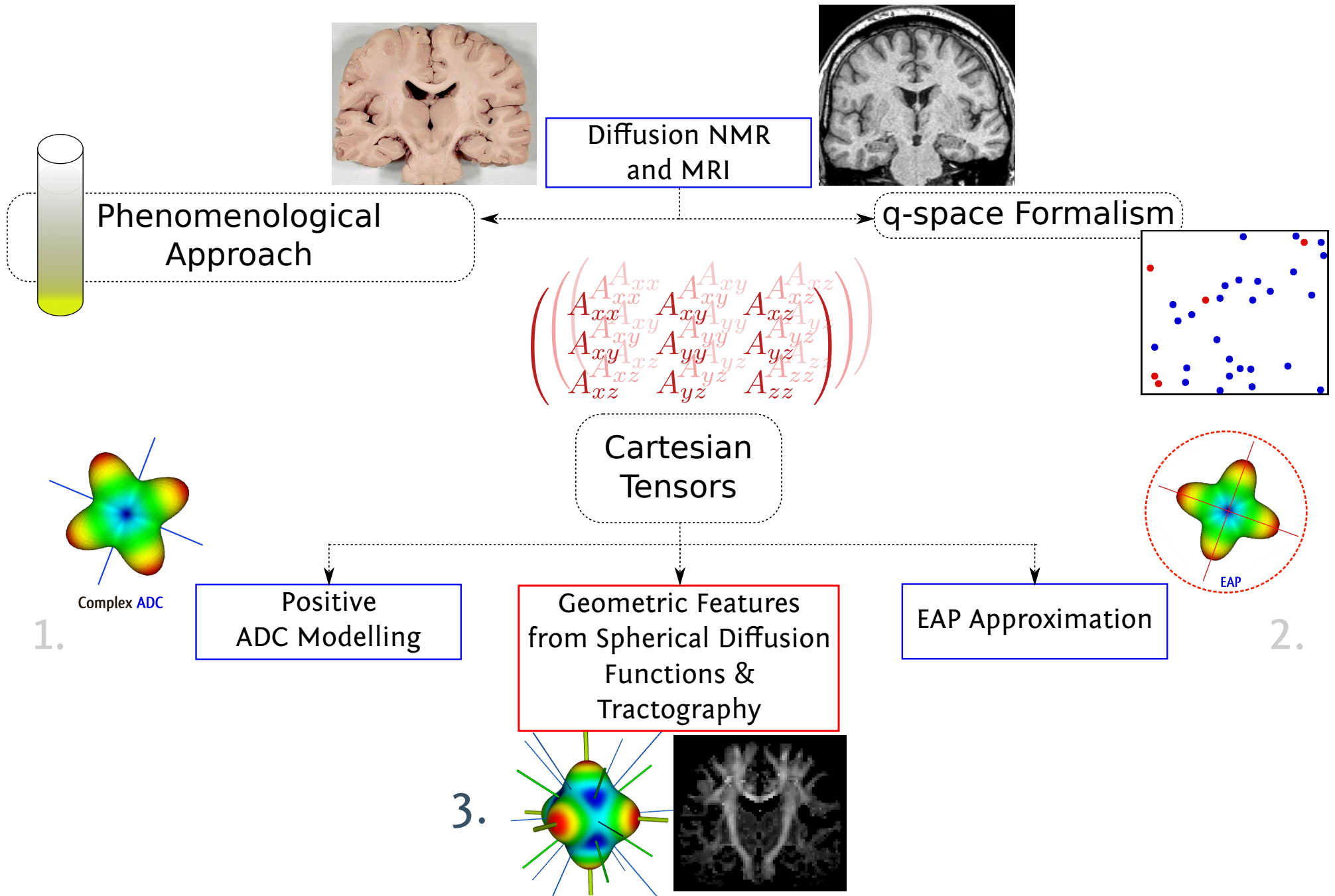


[Ghosh 2010]

- ★ Full EAP recovered
- ★ Closed-Form approximation
- ★ Efficient implementation
- ⊖ Polynomial approximation can cause negative values

Introduction

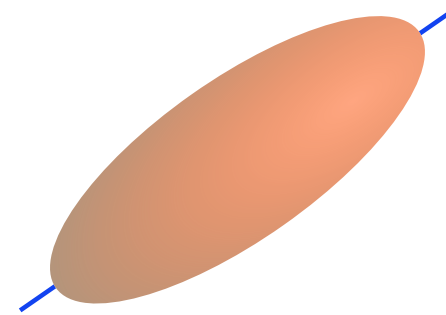
1. Think Positive: 4th Order Symmetric Tensors
 - A Riemannian Approach
 - A Polynomial Approach (Ternary Quartic)
2. Fast and Analytical EAP Approximation from a 4th Order Tensor
3. Geometric Features from Spherical Diffusion Functions
 - Maxima Extraction
 - Peak Fractional Anisotropy



★ Spherical Functions play a central role in Diffusion MRI

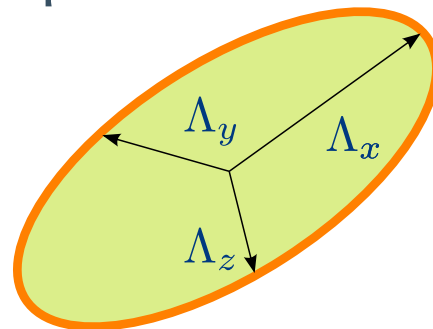
★ DTI Spherical Function: Diffusion Ellipsoid

$$\frac{\mathbf{r}^T \mathbf{D}^{-1} \mathbf{r}}{2t} = 1.$$



★ Geometric Characteristics:

⊙ Spectral Decomposition
(Fast & Easy)

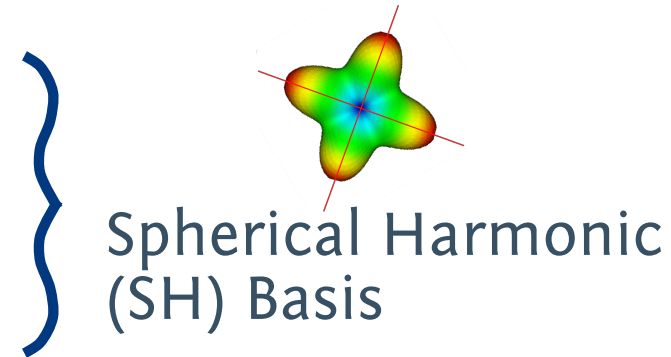


{ Tractography
Fractional Anisotropy
Other scalars (biomarkers)

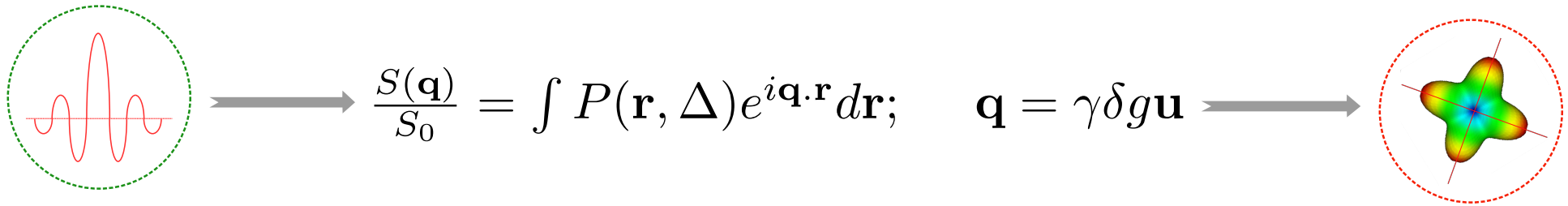
⊖ DTI has limitations in regions with fiber-crossings

★ Other Spherical Functions (SF) in dMRI:

- ⊙ Orientation Distribution Function (ODF)
- ⊙ Fiber Orientation Density (FOD)
- ⊙ PASMRI/MESD
- ⊙ DOT, ADC ... etc...



⊖ Geometric Characteristics? Maxima, minima ...?



★ **ODF-Tuch:** [Tuch Thesis]

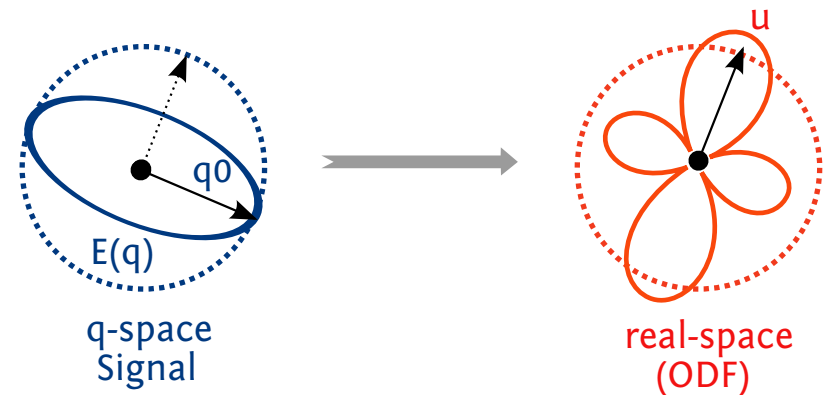
$$\Psi_T(\hat{u}) = \int_0^\infty P(r\hat{u}) dr$$

★ **ODF-Solid Angle:**

$$\Psi_{SA}(\hat{u}) = \int_0^\infty P(r\hat{u}) r^2 dr$$

(True marginal distribution)

[Wedeen 2000, Aganj 2010, Vega 2010]



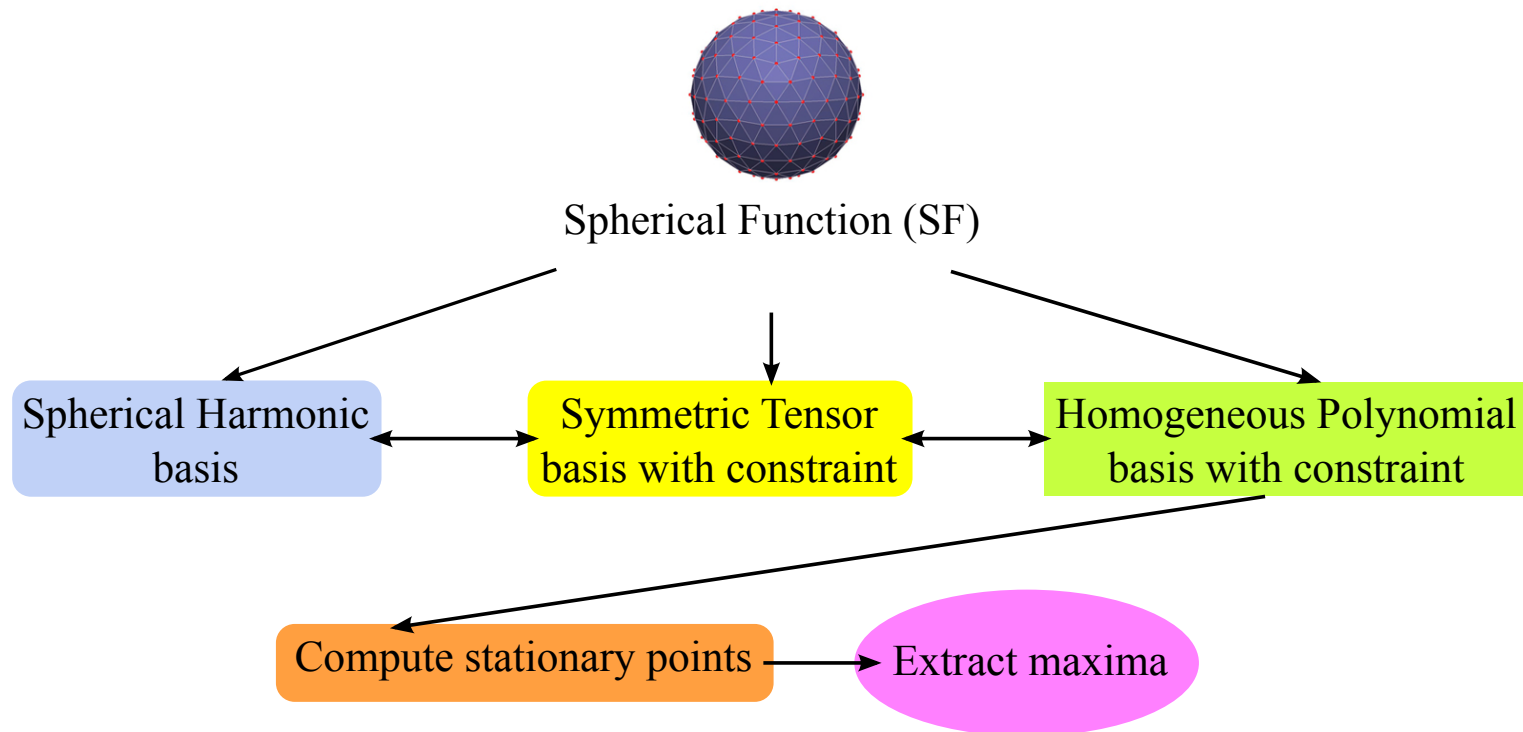
★ **Spherical Harmonics:**

[Descoteaux 2007]

$$\Psi(\theta, \phi) = \sum_{j=1}^N \underbrace{2\pi P_{d(j)}(0) c_j}_{f_j} Y_j(\theta, \phi)$$

- ★ Stationary Points (*extrema*) of a Spherical Function (SF)
- ⦿ Homogeneous Polynomial (HP) representation
- ⦿ Constrained Optimization formulation
 - ★ System of HPs
- ⦿ Categorize extrema: maxima, minima, saddle-points
- ⦿ Principal curvatures at extrema: κ_1 κ_2

- ★ 3 equivalent bases to represent a SF:



★ SH basis to Tensor basis: $c_i = \int_{\Omega} \left(\underbrace{A_{j_1 j_2 \dots j_d} g_{j_1} g_{j_2} \dots g_{j_d}}_{\text{Spherical Function}} \right) Y_i(\theta, \phi) d\Omega$

$\implies \mathbf{C} = \mathbf{M}\mathbf{A}$

(SH Transform)

★ Tensor basis to HP basis:

$$A_{j_1 j_2 \dots j_d} g_{j_1} g_{j_2} \dots g_{j_d} = \sum_{i_1 + i_2 + i_3 = d} A_{i_1 i_2 i_3} g_1^{i_1} g_2^{i_2} g_3^{i_3}$$

(Re-ordering of terms)

★ ODF as a HP: $P_{\Psi}(\mathbf{x} = [x_1, x_2, x_3]^T) = \sum_{i_1 + i_2 + i_3 = d} A_{i_1 i_2 i_3} x_1^{i_1} x_2^{i_2} x_3^{i_3}$
 ($\|\mathbf{x}\|_2 = 1$)

★ Non-linear Polynomial optimization:

$$F(\mathbf{x}, \lambda) = P(\mathbf{x}) - \lambda (\|\mathbf{x}\|_2^2 - 1).$$

(Lagrange Multipliers)

★ System of HPs: $\nabla F : \frac{\partial F}{\partial x_1} = \frac{\partial F}{\partial x_2} = \frac{\partial F}{\partial x_3} = \|\mathbf{x}\|_2^2 - 1 = 0$

★ Polynomial system solver: *Finds all extrema*

(Find all real roots)

[Mourrain 2009]

[Mourrain 2009]

- ★ Convert HP to **Bernstein Basis**: $p(x) = \sum_{i=0}^d c_i B_i^d(x; a, b)$
(exact arithmetic) (in the interval $[a, b]$)

Theorem (Descartes Rule 1D)

The number of real roots of $p(x)$ in $]a, b[$ is bounded by the number $V(\mathbf{c})$ of sign changes of $\mathbf{c} = (c_i)_{i=0..d}$, and is equal modulo 2. (*exclusion test* $V(\mathbf{c})=0$ no roots)

- ★ Analytically bracket / Numerically refine roots
- ★ ALL real roots (extrema) found by exclusion

$$\overline{H}_{(m+n) \times (m+n)}[F(\mathbf{x}, \lambda)] = \begin{pmatrix} \frac{\partial^2 F}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n} & \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 F}{\partial x_n \partial x_n} & \frac{\partial g_1}{\partial x_n} & \cdots & \frac{\partial g_m}{\partial x_n} \\ \\ \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_n} & 0 & \cdots & 0 \end{pmatrix}$$

$$\overline{H}_{(m+n) \times (m+n)} [F(\mathbf{x}, \lambda)] = \begin{pmatrix}
 \boxed{\begin{matrix} \frac{\partial^2 F}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 F}{\partial x_n \partial x_n} \end{matrix}} & \boxed{\begin{matrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_1} \\ \vdots & \vdots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \cdots & \frac{\partial g_m}{\partial x_n} \end{matrix}} \\
 \boxed{\begin{matrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_n} \end{matrix}} & \boxed{\begin{matrix} 0 & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & 0 \end{matrix}}
 \end{pmatrix}$$

$H[F(\mathbf{x}, \lambda)]_{n \times n}$
 $\frac{\partial}{\partial \mathbf{x}} \text{constraint}_{m \times n}$

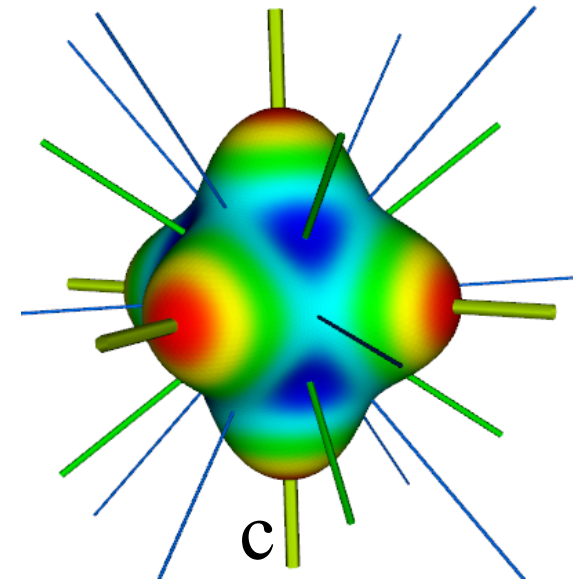
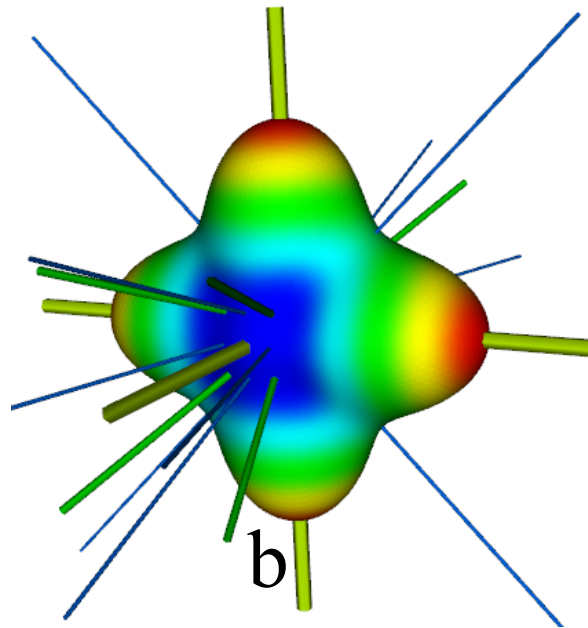
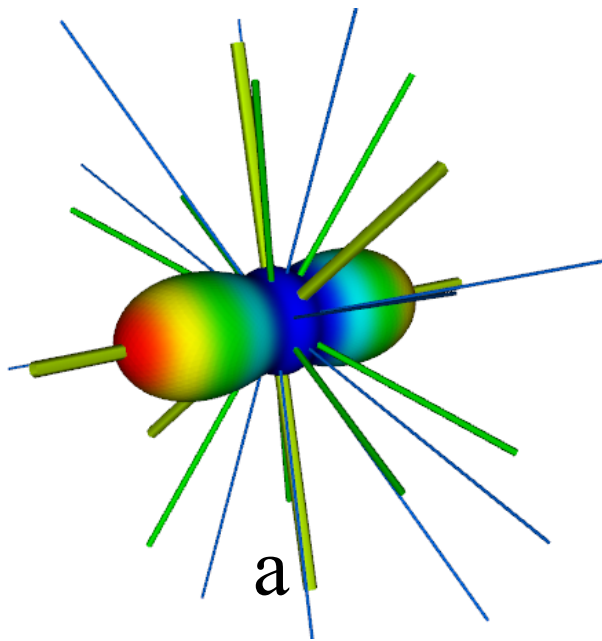
$\frac{\partial}{\partial \mathbf{x}} \text{constraint}_{m \times n}$
0-padding $_{m \times m}$

$$\bar{H}_{(m+n) \times (m+n)} [F(\mathbf{x}, \lambda)] = \begin{pmatrix}
 \begin{matrix} \color{orange}{r \times r} \\ \boxed{\begin{matrix} \frac{\partial^2 F}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 F}{\partial x_n \partial x_n} \end{matrix}} & \begin{matrix} \color{orange}{r \times m} \\ \boxed{\begin{matrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_1}{\partial x_n} & \cdots & \frac{\partial g_m}{\partial x_n} \end{matrix}} \end{matrix} \\
 \color{orange}{m \times r} \boxed{\begin{matrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_n} \end{matrix}} & \begin{matrix} \color{orange}{m \times m} \\ \boxed{\begin{matrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{matrix}} \end{matrix}
 \end{matrix}
 \end{pmatrix}$$

$H[F(\mathbf{x}, \lambda)]_{n \times n}$ $\frac{\partial}{\partial \mathbf{x}} \text{constraint}_{m \times n}$ $\frac{\partial}{\partial \mathbf{x}} \text{constraint}_{m \times n}$ $0\text{-padding}_{m \times m}$

★ **Maximum:** $(-1)^m |\bar{H}_r [F(\mathbf{x}, \lambda)]| > 0; \quad r = m + 1, \dots, n,$

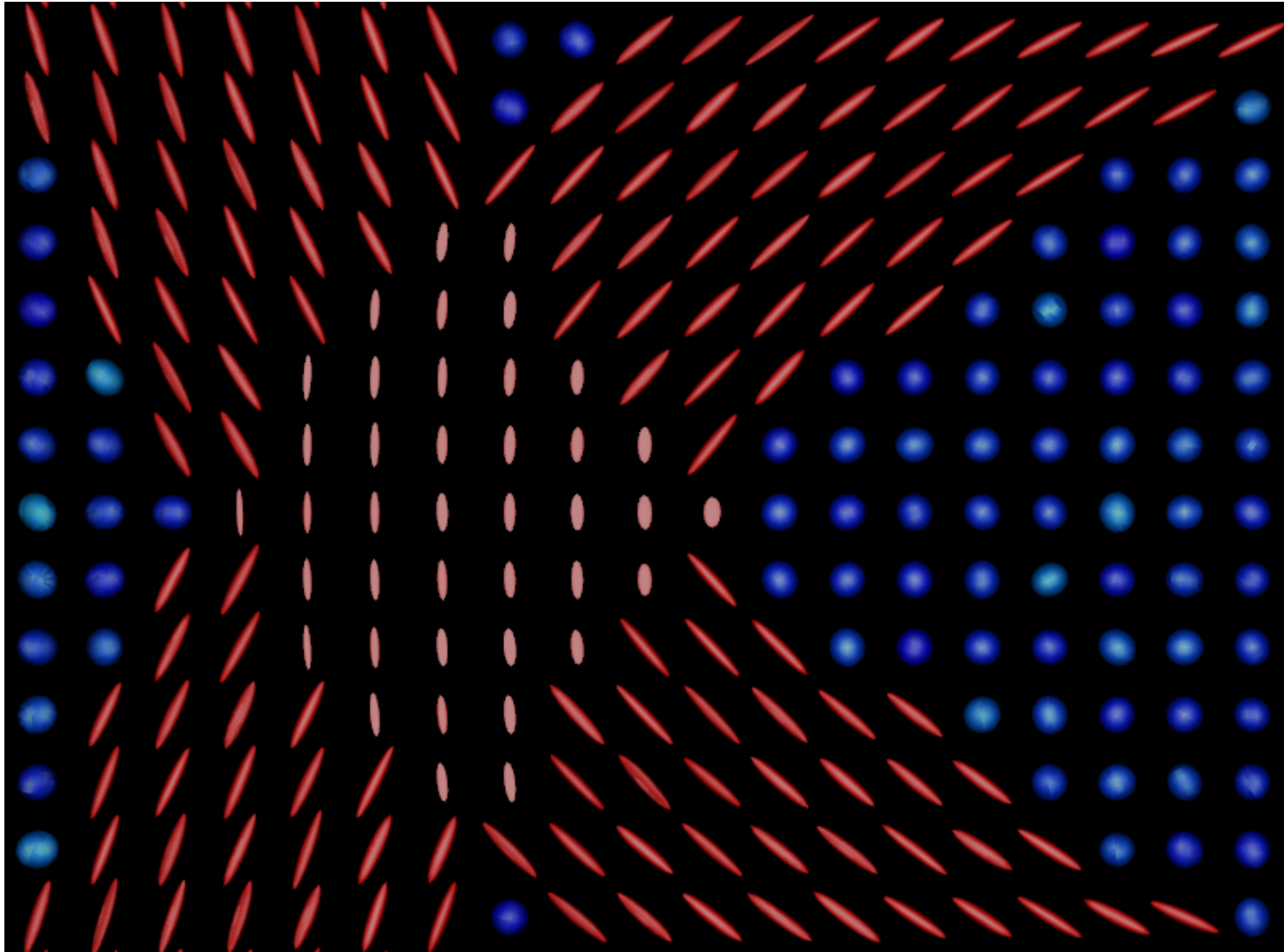
Minimum: $(-1)^r |\bar{H}_r [F(\mathbf{x}, \lambda)]| > 0 \quad (m = 1 \quad n = 3)$

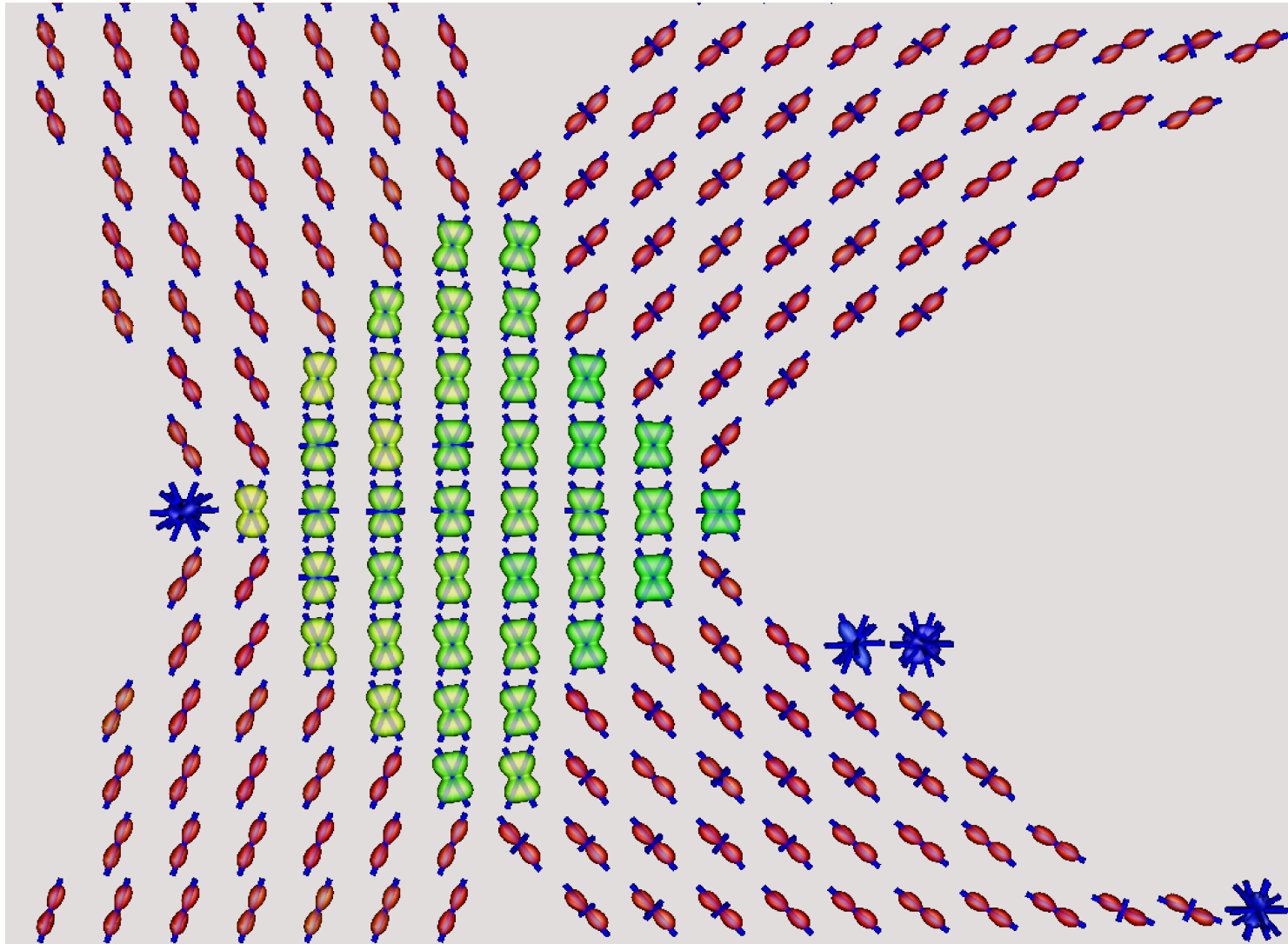


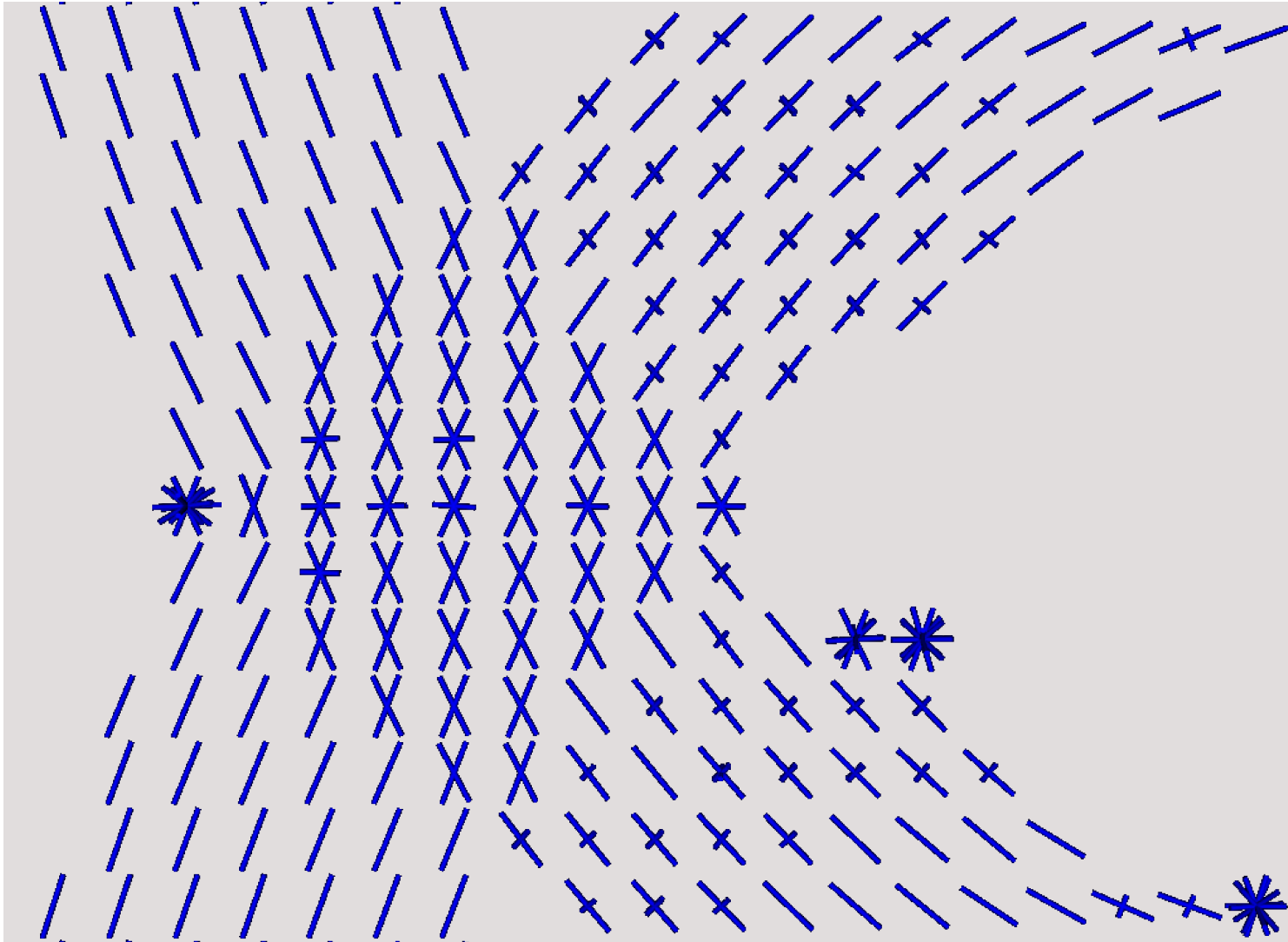
Glyph-colour: **red: high anisotropy**, blue: low anisotropy

Lines: **thick-green: Maxima**, thin-blue: Saddle points, **green: Minima**.

a) 1-fiber b) 2-fibers c) 3-fibers







(Synthetic data)

★ How good was the estimation?

- ① Take ground truth directions $\{d_i\}$ and generate *signal*
- ② Estimate ODF-Ts from signal
- ③ Compute maxima of ODF-Ts: $\{d'_i\}$
- ④ Compute (F= ODF as HP): $\|\nabla F(d)\|$ & $\|\nabla F(d')\|$

$\|\nabla F(d)\|$: estimation error

$\|\nabla F(d')\|$: maxima extraction error

	$ang(d, d')^\circ$		$\ \nabla F(d')\ $		$\ \nabla F(d)\ $	
	m	v	m	v	m	v
1-Fib	0.01	4.22e-05	2.07e-11	1.5e-21	4.9e-04	9.17e-08
2-Fib _{90°}	0.025	3.85e-05	5.58e-11	1.76e-21	3.6e-03	2.48e-06

m = mean, v = variance

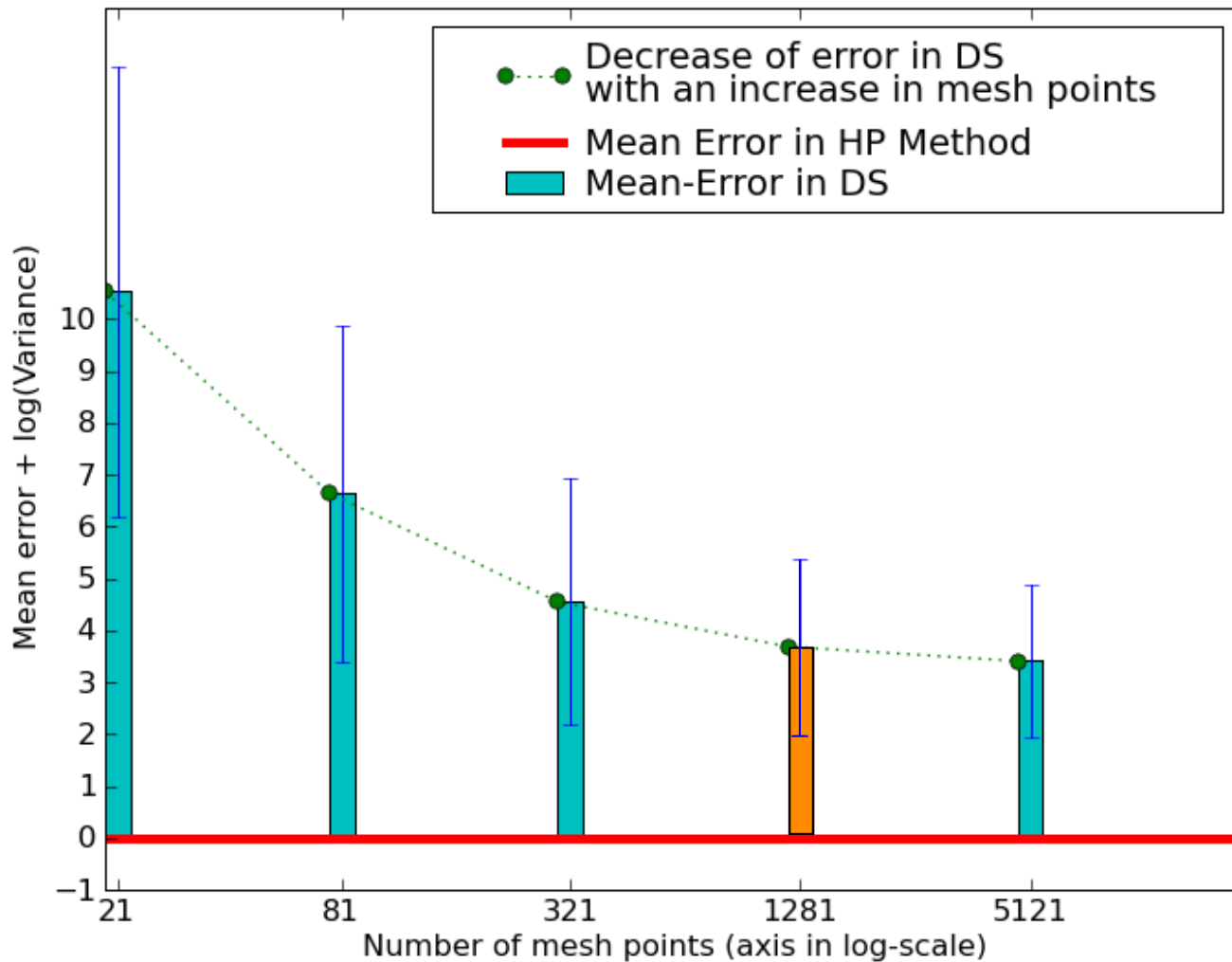
★ Noise-free 100 simulations (each)

(Human brain data)

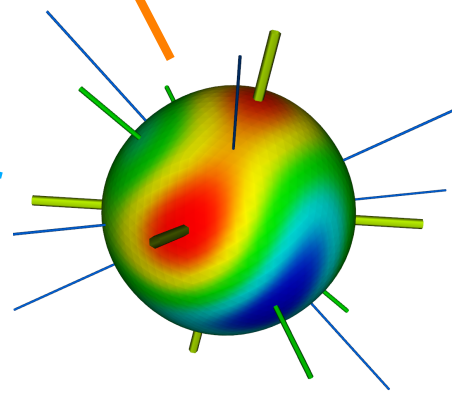
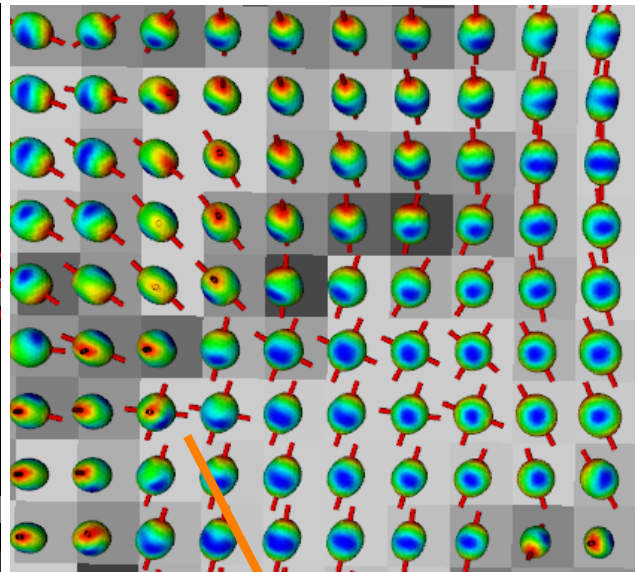
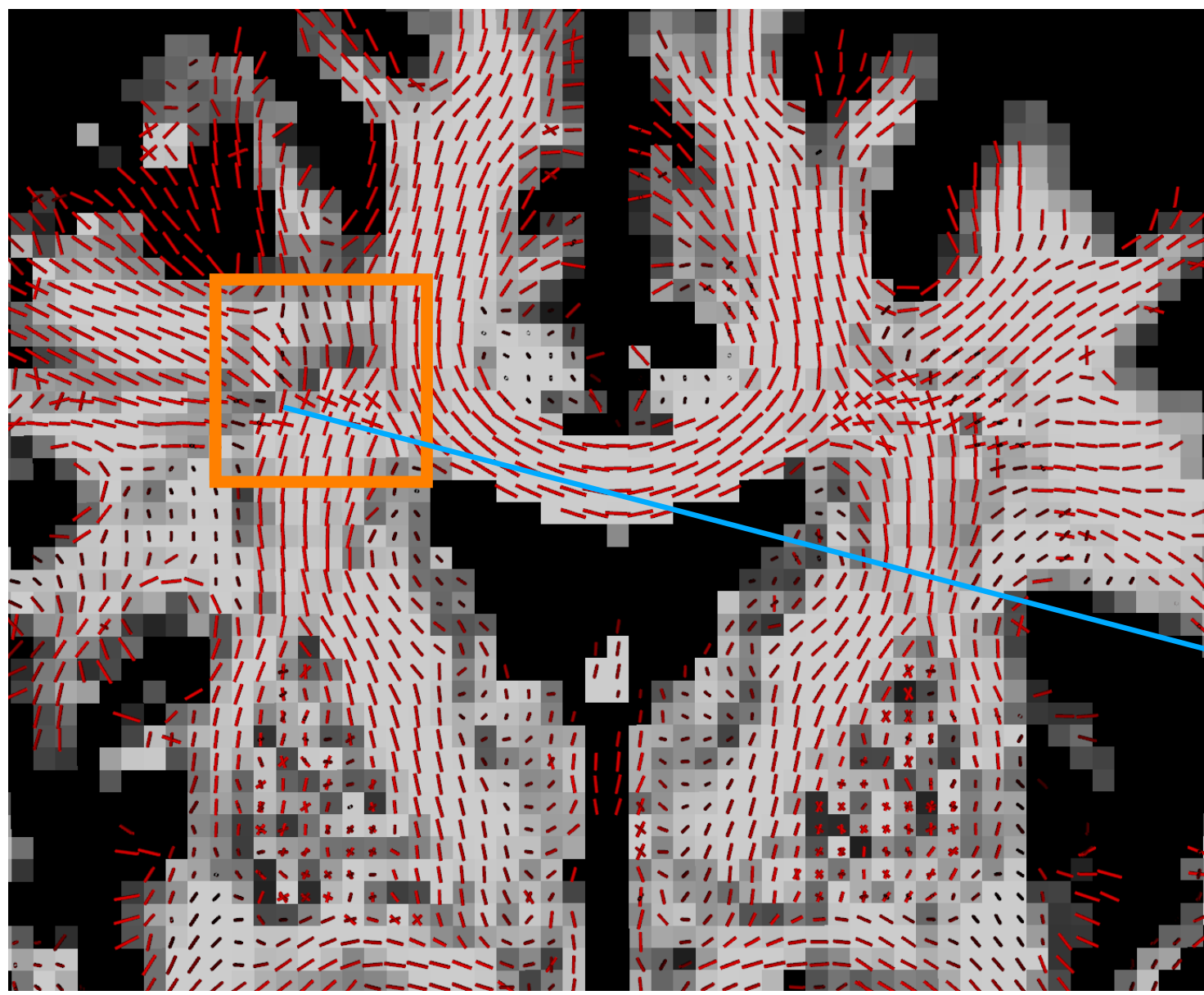
$\|\nabla F(d)\|$: HP maxima error

$\|\nabla F(d')\|$: Discrete mesh search (DS) maxima error

Comparison of error in Discrete Search method (DS) vs the HP Method

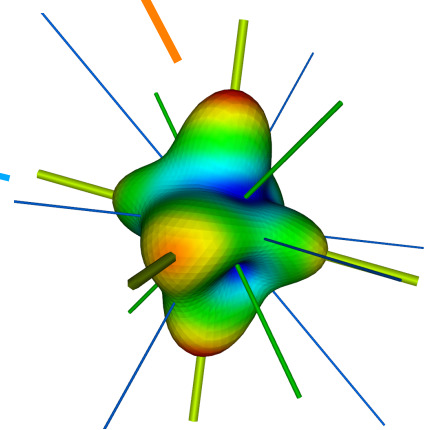
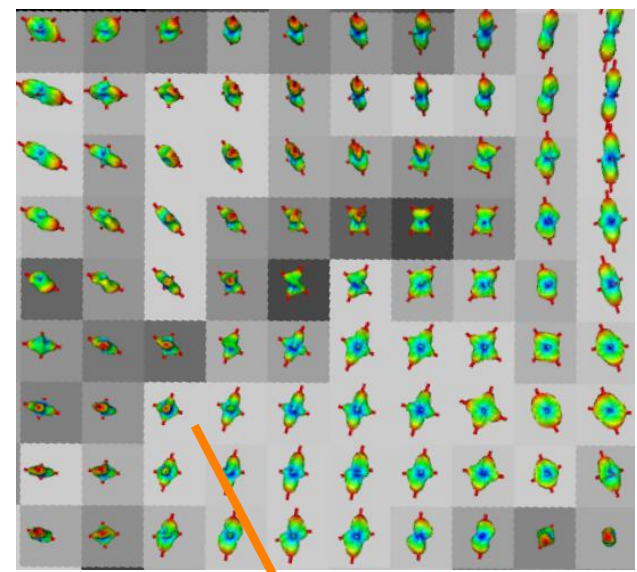
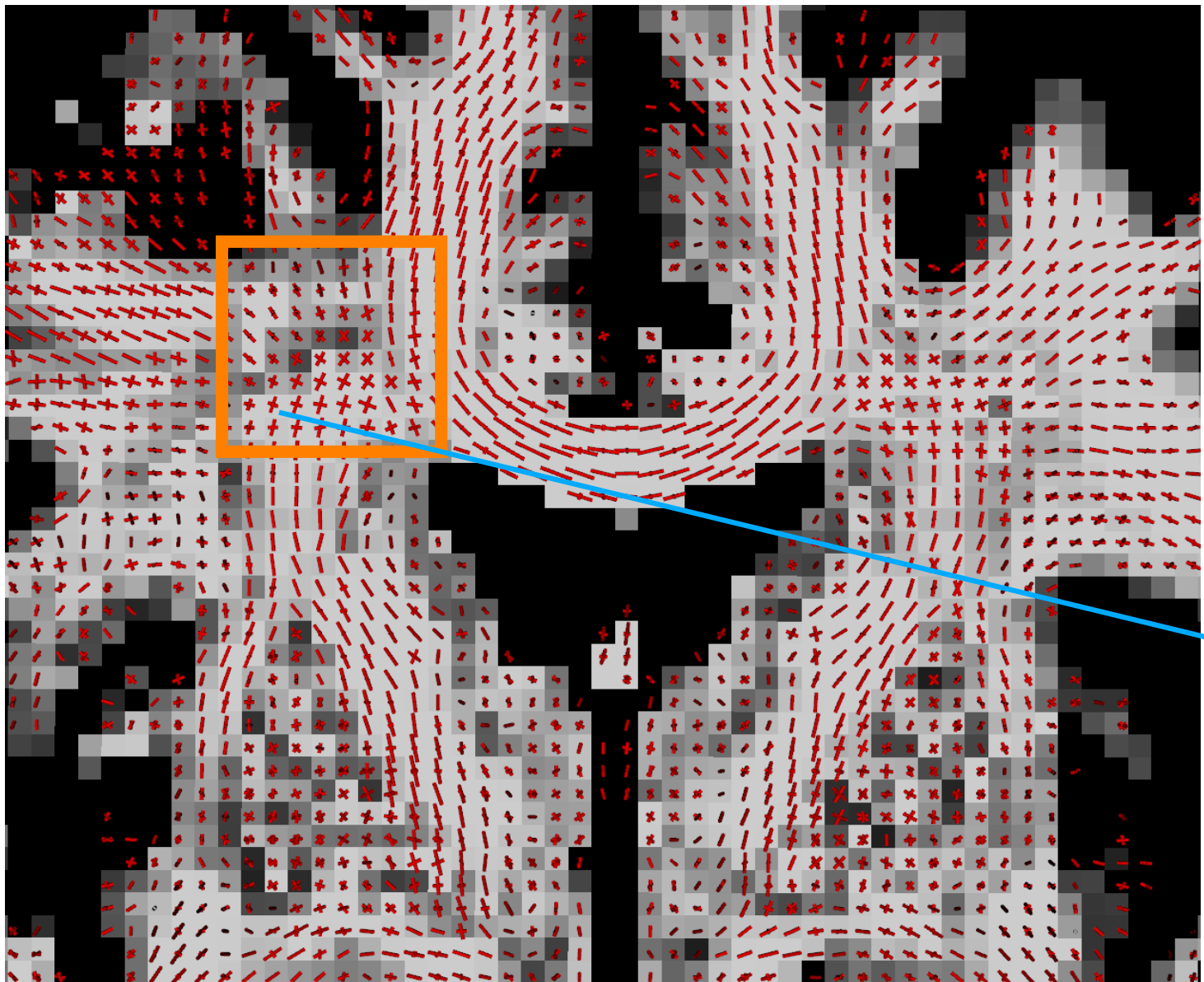


[Anwander 2007]



ODF-Tuch

[Anwander 2007]



ODF-SA

Introduction

1. Think Positive: 4th Order Symmetric Tensors
 - A Riemannian Approach
 - A Polynomial Approach (Ternary Quartic)
2. Fast and Analytical EAP Approximation from a 4th Order Tensor
3. Geometric Features from Spherical Diffusion Functions
 - Maxima Extraction
 - Peak Fractional Anisotropy

[Bloy 2008]

★ ODF in Cartesian coordinates:

$$\Psi : [\Psi(\theta, \phi)\sin(\theta)\cos(\phi), \Psi(\theta, \phi)\sin(\theta)\sin(\phi), \Psi(\theta, \phi)\cos(\theta)]^T$$

★ Form-I : e, f, g

Form-II: E, F, G

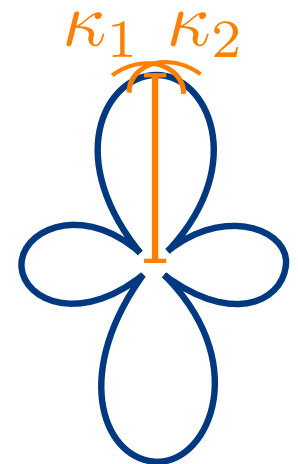
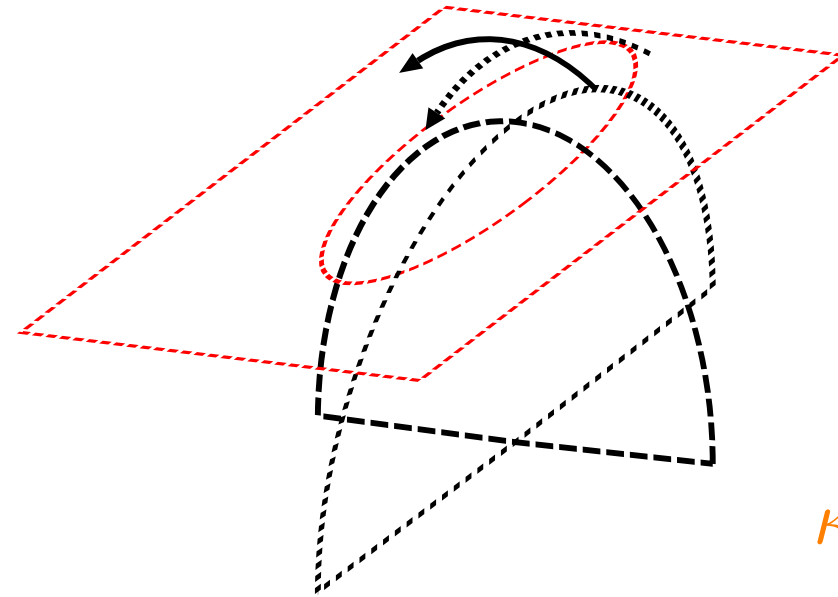
★ Principal Curvatures:

$$K = \frac{eg - f^2}{EG - F^2}$$

$$H = \frac{eG - 2fF + gE}{2(EG - F^2)}$$

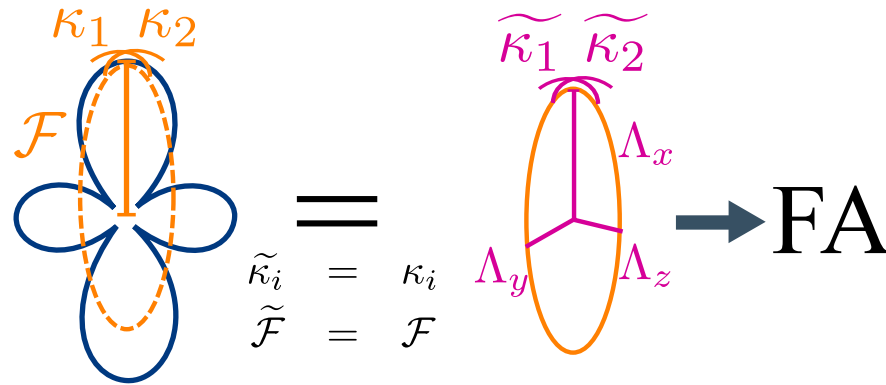
$$\kappa_1 = H + \sqrt{H^2 - K}$$

$$\kappa_2 = H - \sqrt{H^2 - K}$$

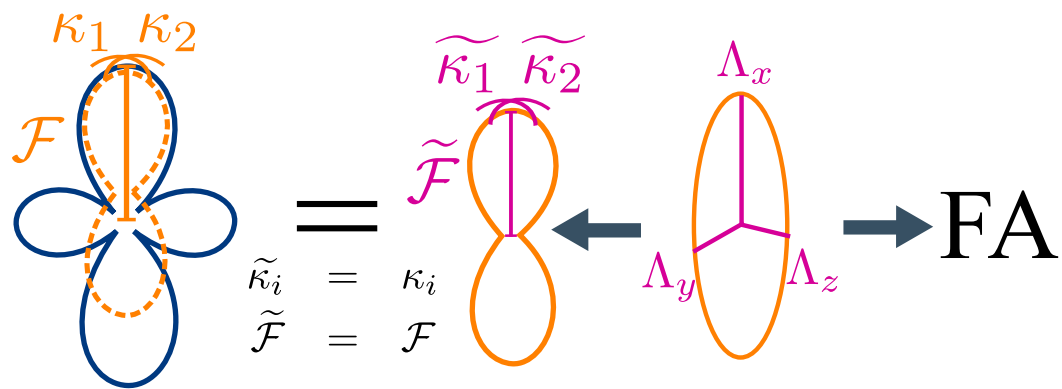


- ★ Compute bio-marker from SFs (ODF) using geometric characteristics

- ★ Ellipsoidal: Peak FA



- ★ ODF Peak FA: (Tuch & SA)



(Analytical ODF model)

$$FA(\Lambda_x(\kappa_1, \kappa_2, \mathcal{F}), \Lambda_y(\kappa_1, \kappa_2, \mathcal{F}), \Lambda_z(\kappa_1, \kappa_2, \mathcal{F}))$$

★ ODF-Tuch:
PFA-T

$$\Psi_T = \frac{1}{Z} \sqrt{\frac{1}{\frac{x^2}{\Lambda_x} + \frac{y^2}{\Lambda_y} + \frac{z^2}{\Lambda_z}}}$$

$$\widetilde{\kappa}_1 = \frac{Z\sqrt{\Lambda_x}}{\Lambda_y}; \quad \widetilde{\kappa}_2 = \frac{Z\sqrt{\Lambda_x}}{\Lambda_z}; \quad \widetilde{\mathcal{F}} = \frac{\sqrt{\Lambda_x}}{Z}$$

★ ODF-SA:
PFA-SA

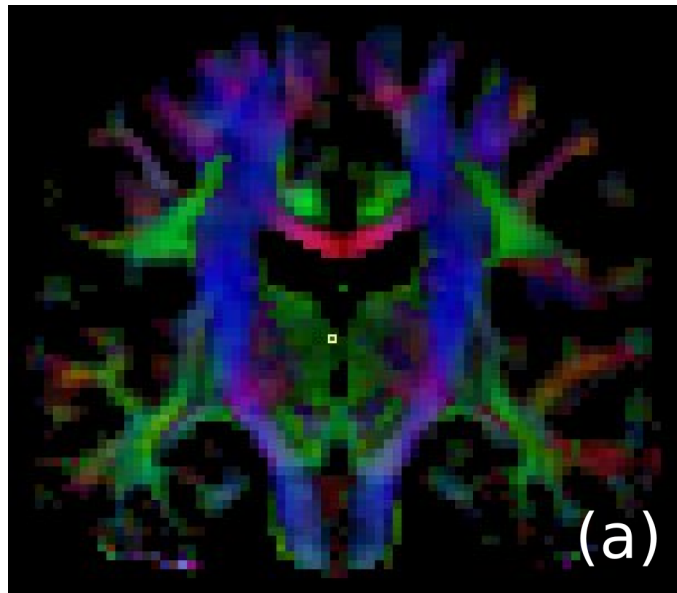
$$\Psi_{SA} = \frac{1}{4\pi\sqrt{\Lambda_x\Lambda_y\Lambda_z}} \left(\frac{1}{\frac{x^2}{\Lambda_x} + \frac{y^2}{\Lambda_y} + \frac{z^2}{\Lambda_z}} \right)^{\frac{3}{2}}$$

$$\widetilde{\kappa}_1 = \frac{4\pi\Lambda_y(3\Lambda_x - 2\Lambda_z)}{\Lambda_x\sqrt{\Lambda_y\Lambda_z}}; \quad \widetilde{\kappa}_2 = \frac{4\pi\Lambda_z(3\Lambda_x - 2\Lambda_y)}{\Lambda_x\sqrt{\Lambda_y\Lambda_z}}$$

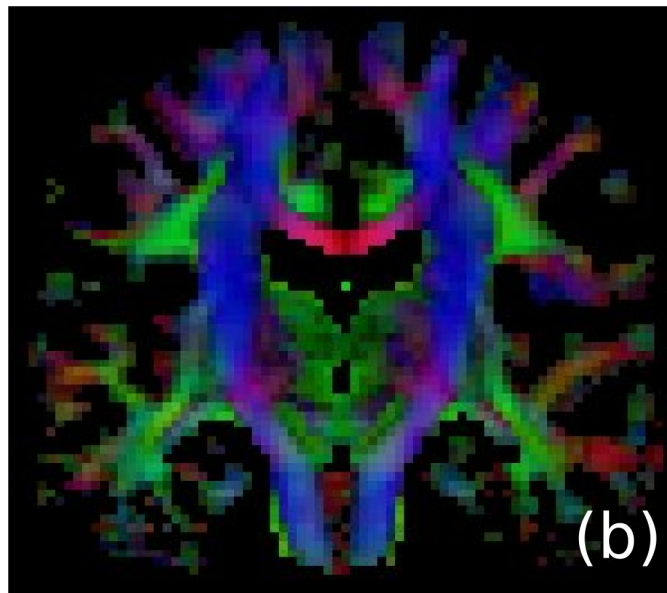
$$\widetilde{\mathcal{F}} = \frac{\Lambda_x^{3/2}}{4\pi\sqrt{\Lambda_x\Lambda_y\Lambda_z}}$$

★ Total PFA:

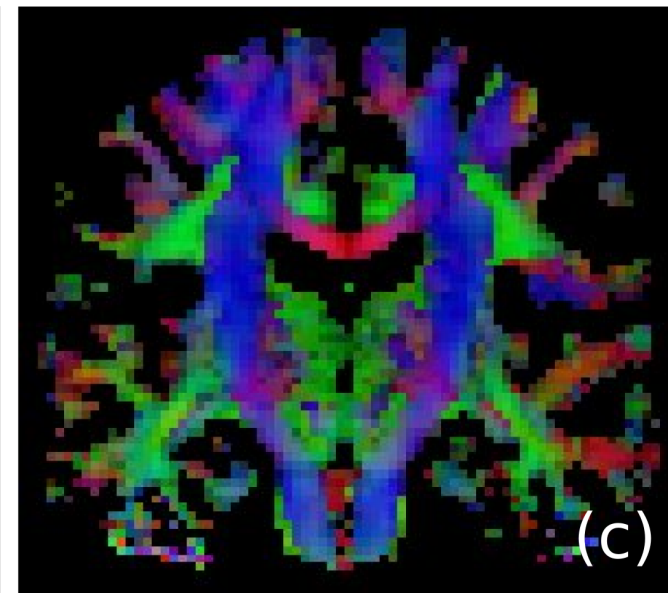
$$\text{TotalPFA} = \sum_{i=1}^N \mathcal{F}_i \cdot \text{PFA}_i.$$



Colour coded DTI
eigenvector
weighted by FA



Colour coded ODF-T
main-maximum
weighted by PFA-T



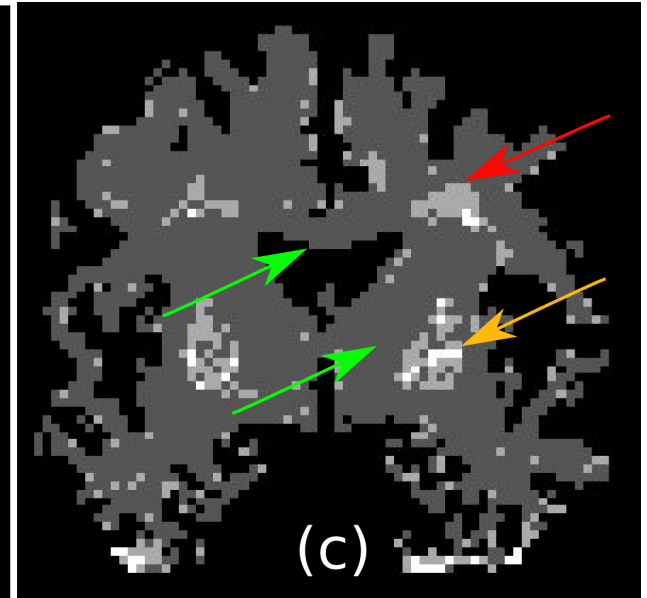
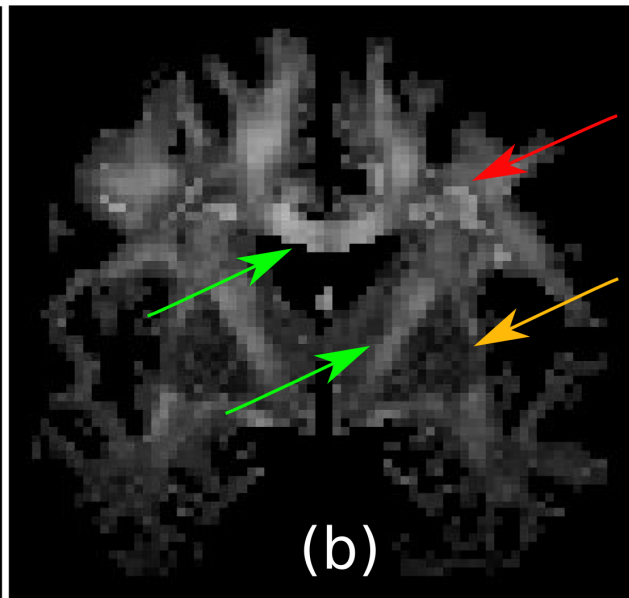
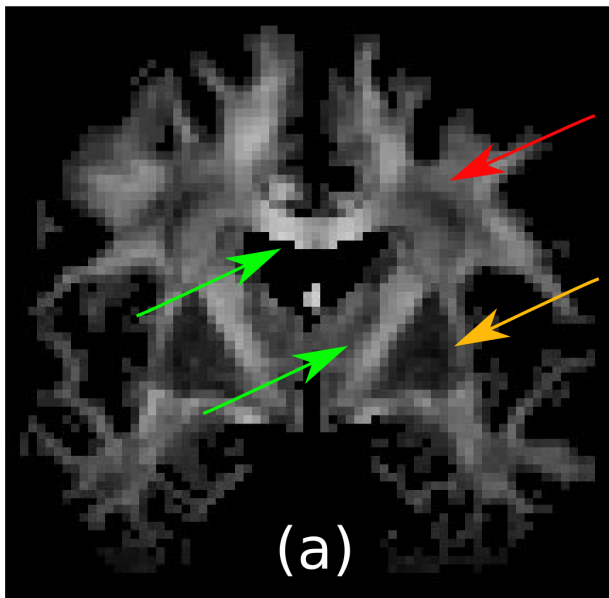
Colour coded ODF-SA
main-maximum
weighted by PFA-SA

GFA

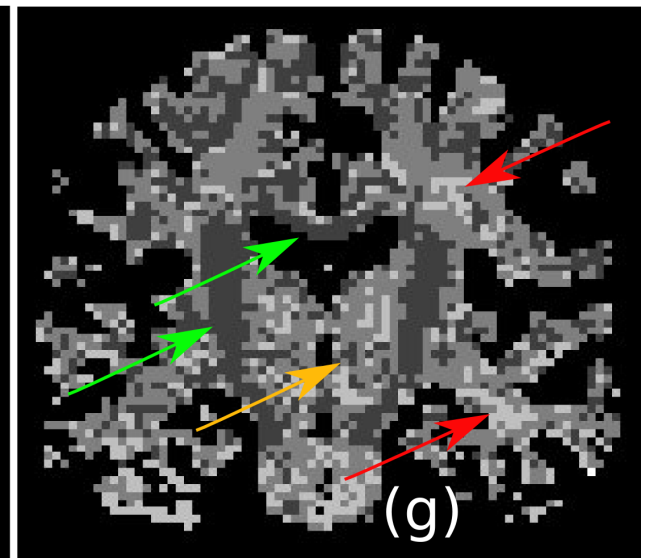
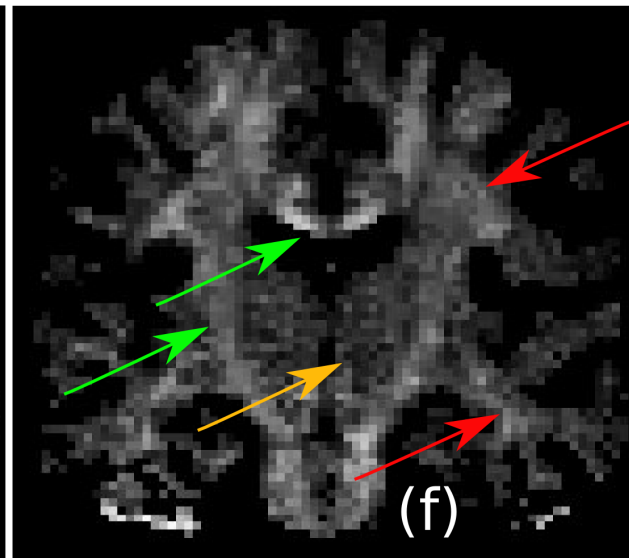
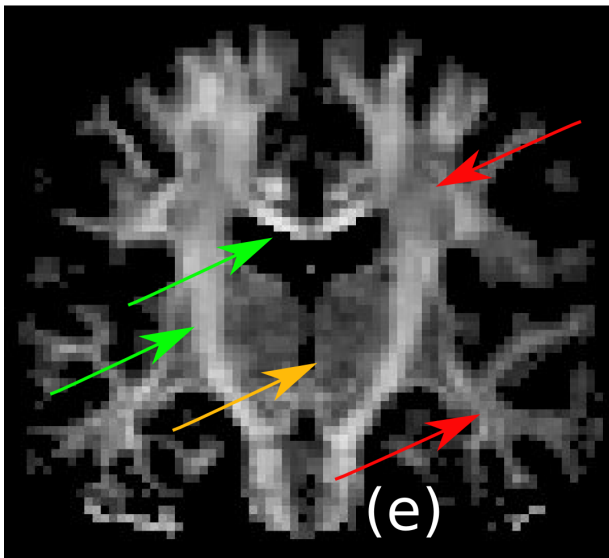
Total-PFA

No. Maxima

ODF-T



ODF-SA

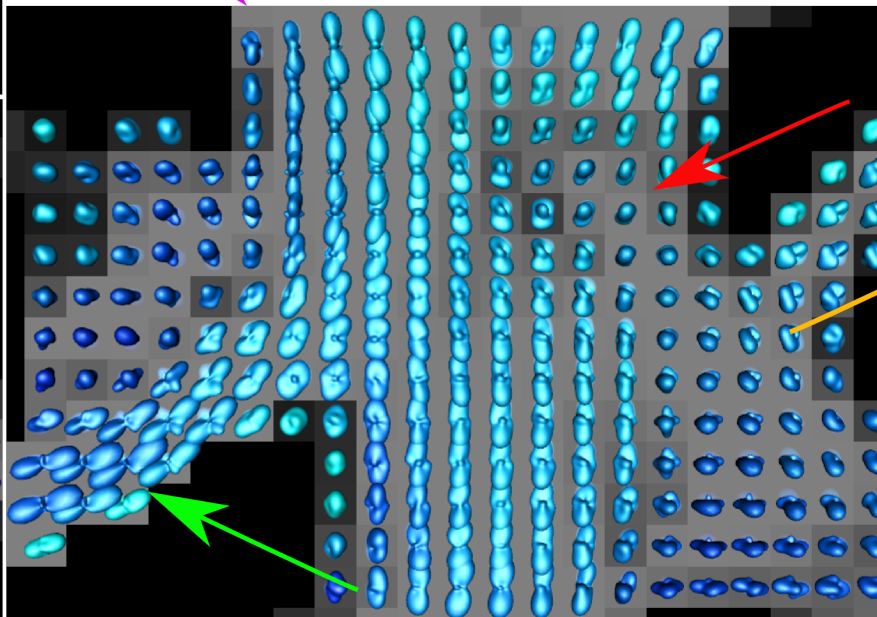
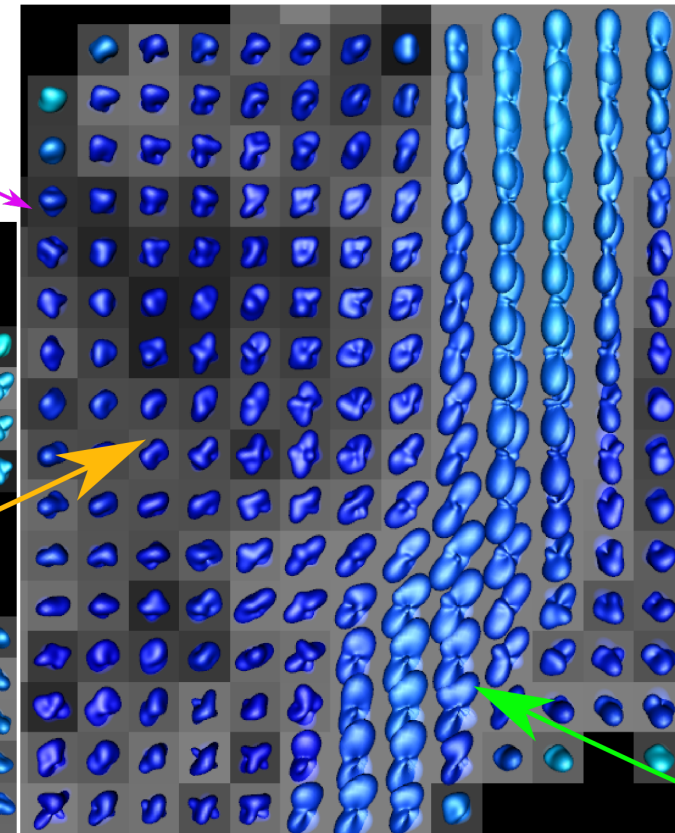
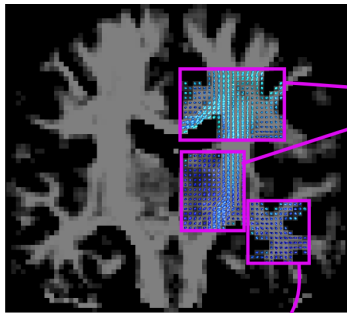
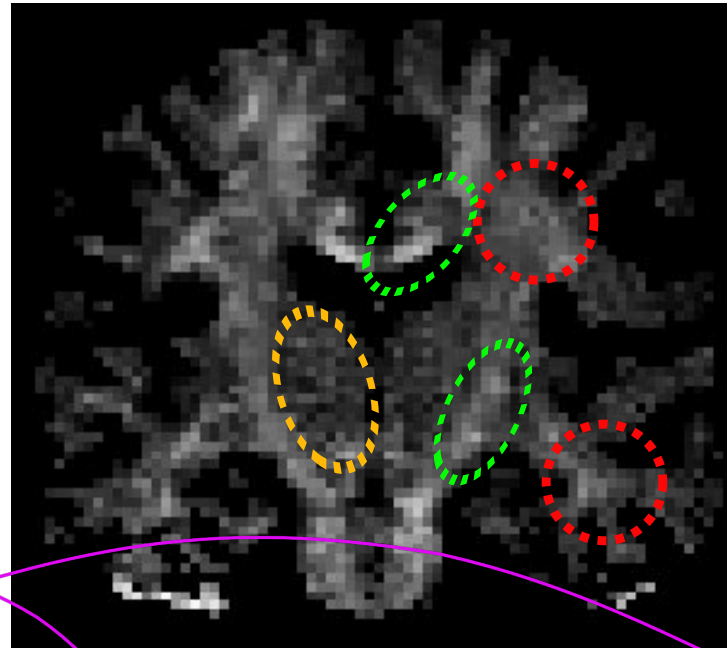


Green: single fiber,
high anisotropy

Red: multiple fibers,
high anisotropy

Orange: multiple "fibers",
low anisotropy

ODF-SA

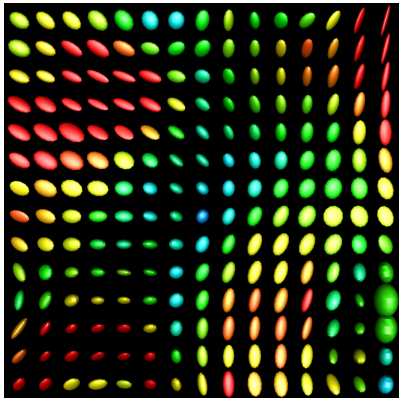


[Ghosh 2008b, Ghosh2011]

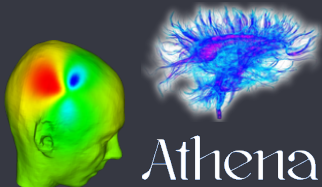
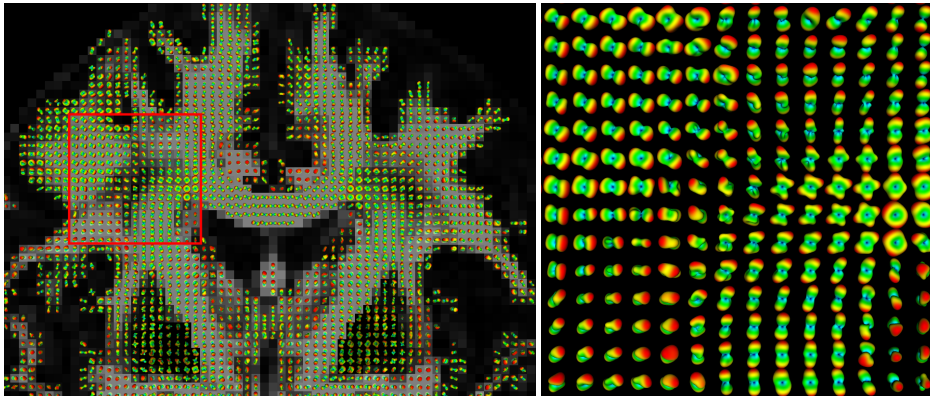
- ★ Maxima extraction from **ANY** kind of spherical function
- ★ **ALL** maxima are found
- ★ Peak Fractional Anisotropy (PFA): a new bio-marker
- ★ Extend Streamline & Tensorline tractography to any spherical function with maxima-extraction
- ◎ PFA as a stopping criterion in tractography?

Conclusion and Perspectives

DTI



GDTI



High Order Models in
Diffusion MRI
and Applications

- ★ Generalized DTI: Higher Order Cartesian Tensors (HOTs)
Positive Diffusion Constraint
 - ◎ 3D 4th Order tensors → 6D 2nd Order tensors
Riemannian metric for symmetric positive definite matrices
 - ★ Positive-definiteness; solves *constrained* problem
 - ◎ Polynomial Interpretation of HOTs
Hilbert's theorem on Ternary Quartics
 - ★ Positive-semi-definiteness; solves *correct* problem
- ★ Analytical approximation of EAP from HOTs
 - ◎ Closed-form formulation
 - ◎ Efficient implementation

- ★ Geometric Features from Spherical Diffusion functions
 - ⦿ All maxima found from any kind of spherical function
 - ⦿ Peak Fractional Anisotropy (PFA): a new bio-marker
 - ⦿ Streamline & Tensorline tractography for any kind of spherical function
-

★ Perspectives:

- ⦿ Alternate Cartesian tensor (& coordinates) approach
- ⦿ Cartesian Fourier Transform
- ⦿ Cartesian descriptors: Kurtosis, Cumulants...

Members of the Jury:

Dr. Peter Basser

Dr. Baba Vemuri

Dr. Habib Benali

Dr. Pierre Comon

Dr. Rachid Deriche

Dr. Nikos Paragios

Thank you

ANR project NucleiPark
France-Parkinson Association

Acknowledgements:



Dr. Rachid Deriche

Emmanuel Caruyer

Jian Cheng

Sylvain Merlet

Anne-Charlotte Philippe

Odyssée

Athena / NeuroMathComp

Dr. Alfred Anwander

Dr. Maxime Descoteaux

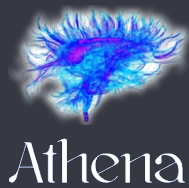
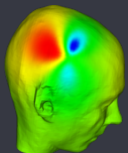
Dr. Maher Moakher

Dr. Bernard Mourrain

Dr. Evren Ozarslan

Dr. Elias Tsigaridas

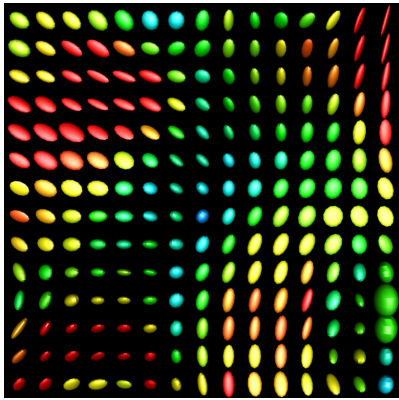
Dr. Demian Wassermann



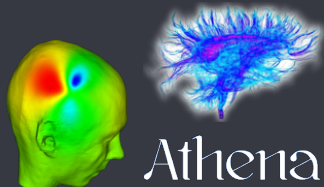
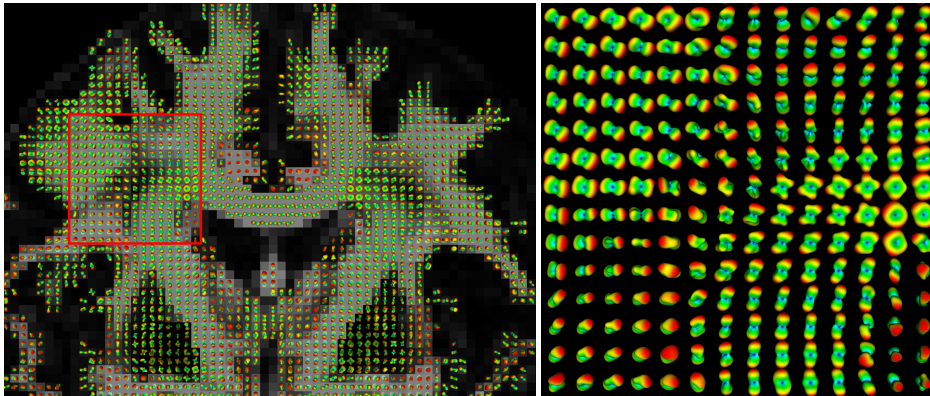
Athena

Extras

DTI



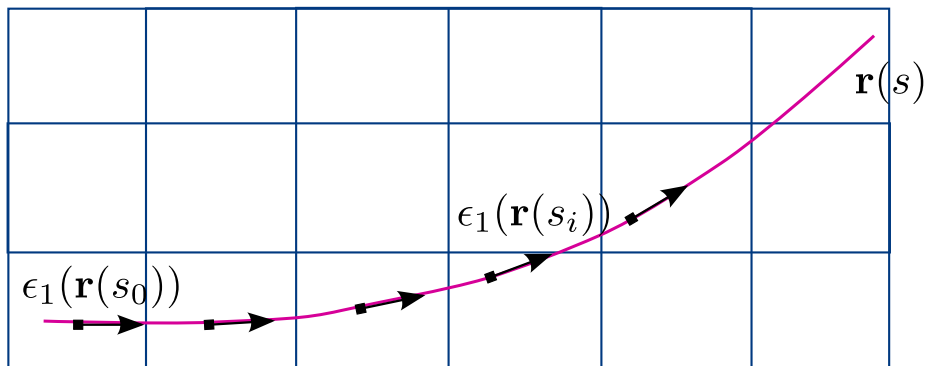
GDTI



High Order Models in
Diffusion MRI
and Applications

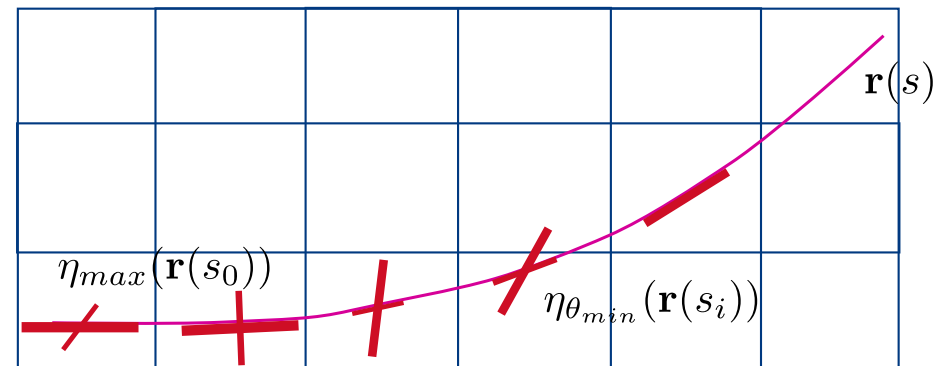
★ DTI Streamline: tractography [Basser 2000]

$$\frac{d\mathbf{r}(s)}{ds} = \mathbf{t}(s) = \epsilon_1(\mathbf{r}(s)),$$



⊙ ODF Streamline: tractography

$$\frac{d\mathbf{r}(s)}{ds} = \eta_{\theta_{min}}(\mathbf{r}(s)), \quad \mathbf{r}(0) = \eta_{max}(0)$$



★ DTI Tensorline: [Weinstein 1999, Lazar 2003]

$$\mathbf{v}_{\text{out}} = f\epsilon_1 + (1 - f)((1 - g)\mathbf{v}_{\text{in}} + g\mathbf{D} \cdot \mathbf{v}_{\text{in}})$$

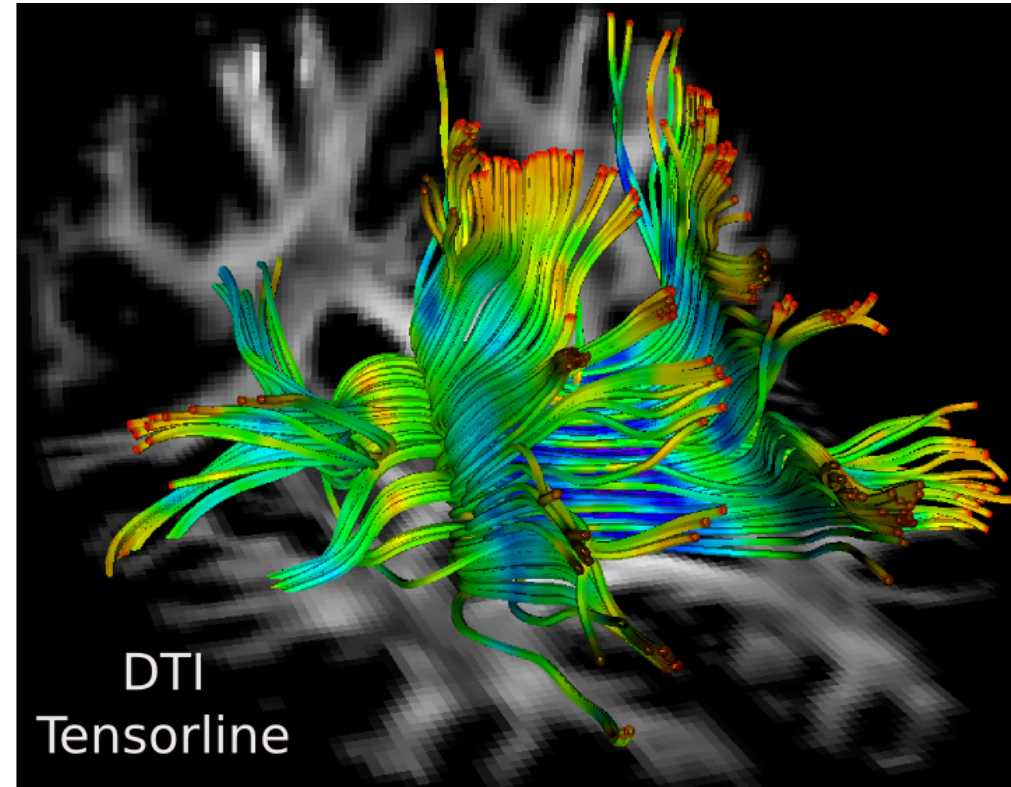
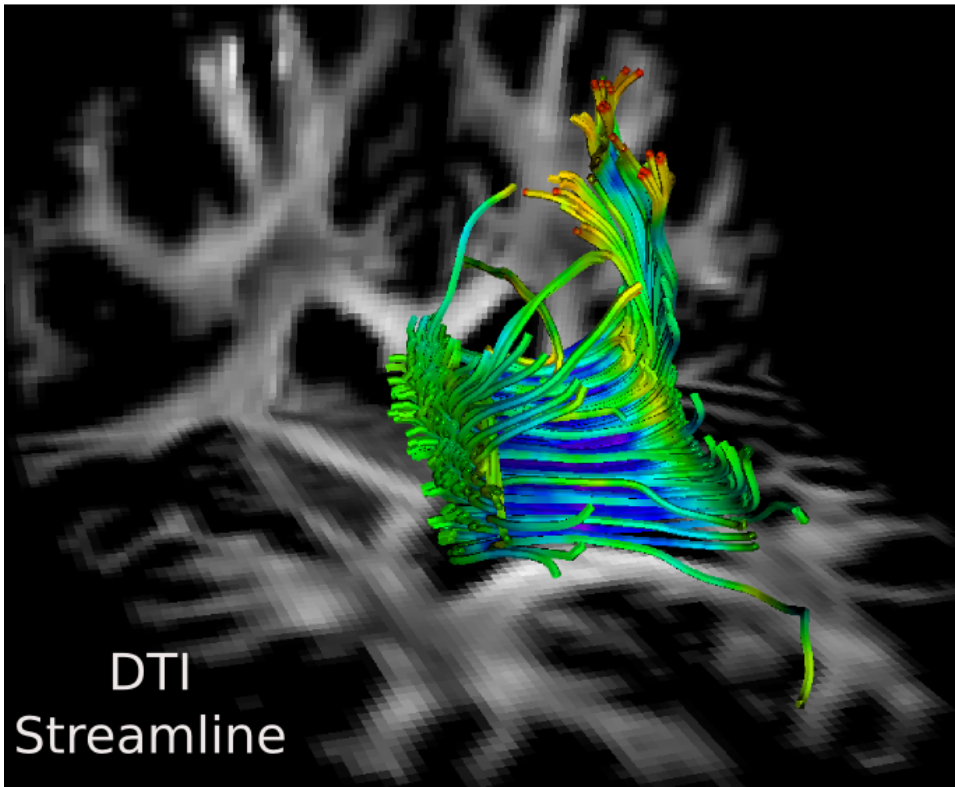
- ★ DTI Tensorline: [Weinstein 1999, Lazar 2003]

$$\mathbf{v}_{\text{out}} = f\epsilon_1 + (1 - f)((1 - g)\mathbf{v}_{\text{in}} + g\mathbf{D} \cdot \mathbf{v}_{\text{in}})$$

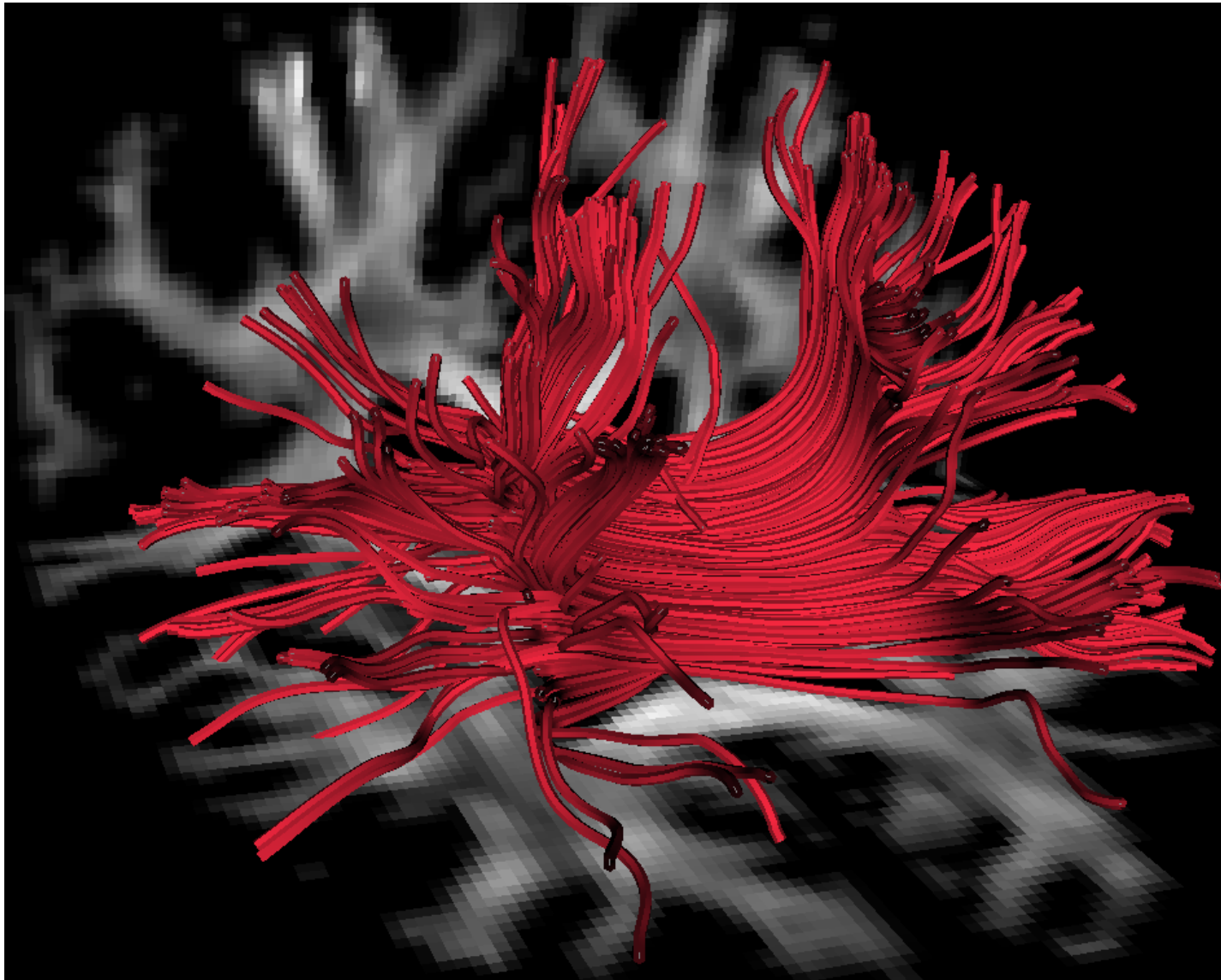
- ⊙ ODF Tensorline:

$$\mathbf{v}_{\text{out}} = f\eta_{\theta_{\text{min}}} + (1 - f)\mathbf{v}_{\text{in}}$$

- ★ Euclidean interpolation
- ★ 0.5mm step-length
- ★ Stopping Criterion: 0.87mm Radius of Curvature

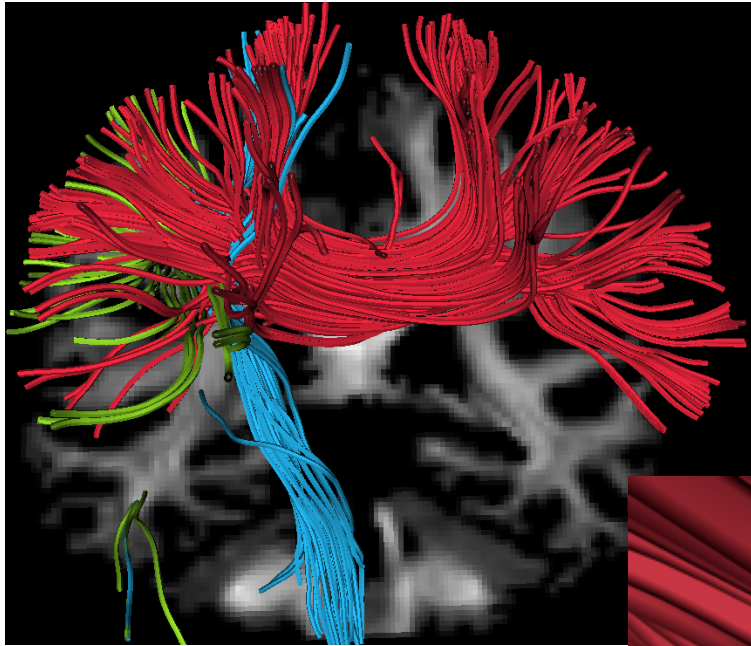


seeds: Corpus Callosum

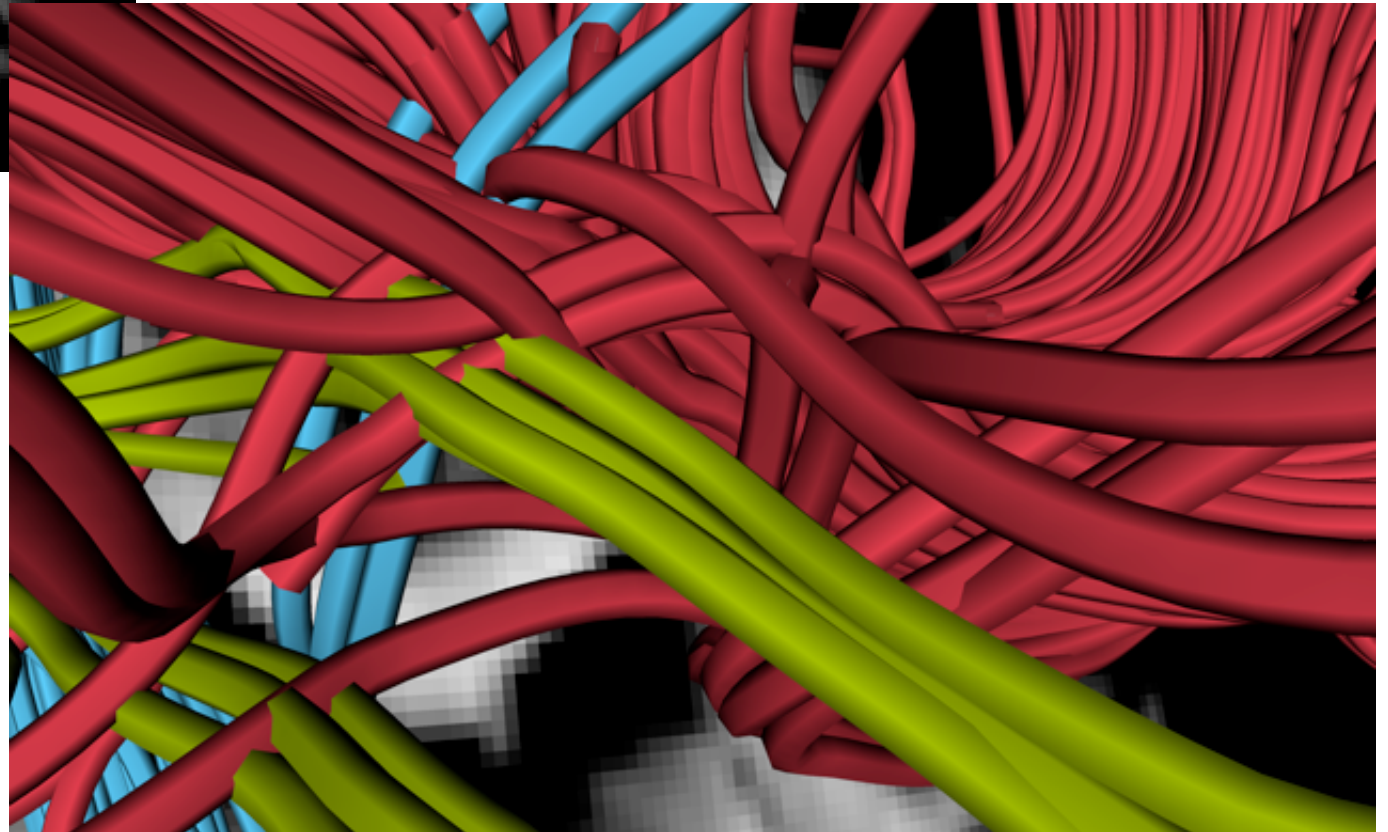


seeds: Corpus Callosum

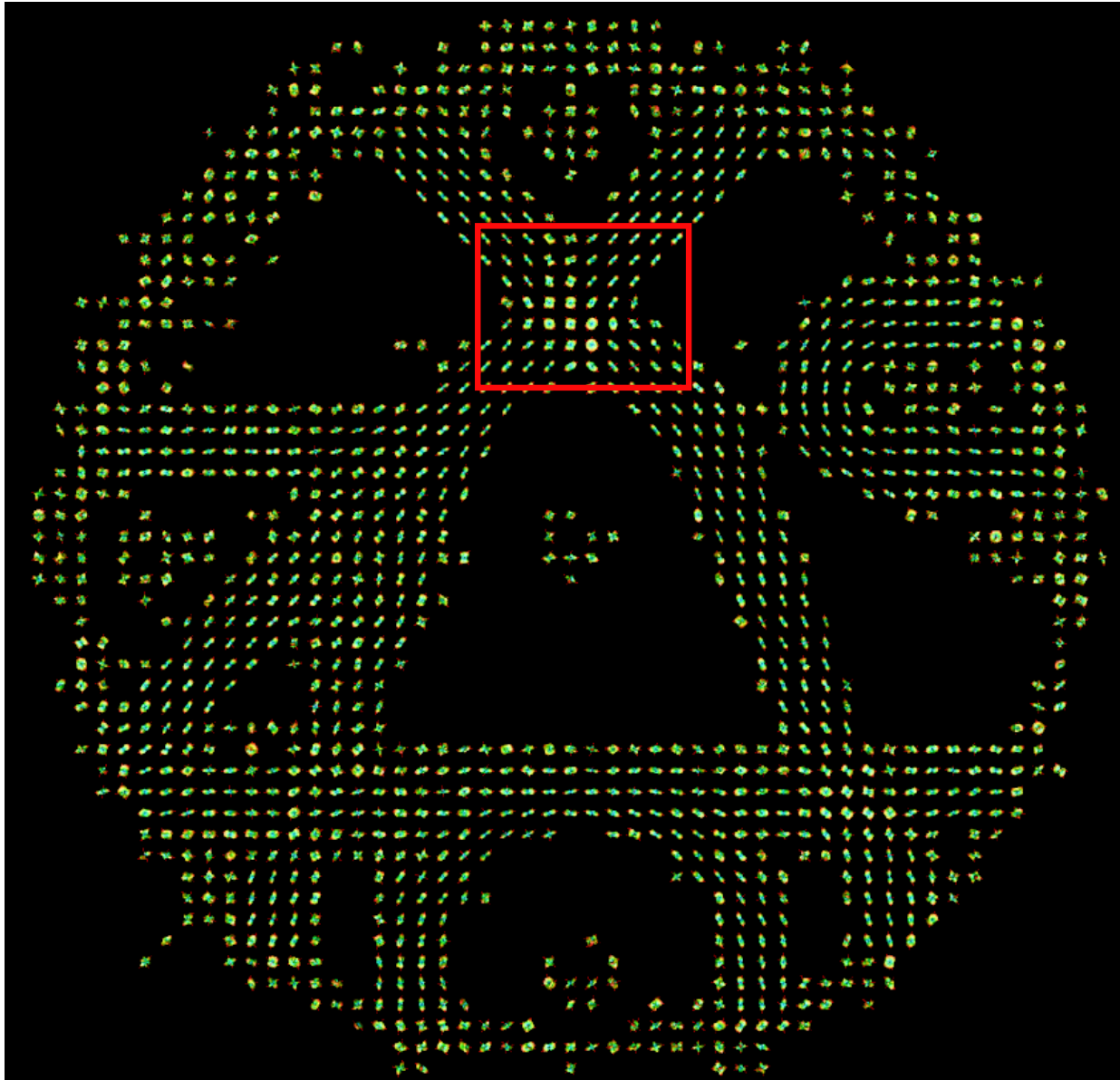
Results: ODF-SA Tensorline



blue: Cortico-Spinal Tract
green: Superior Longitudinal Fasciculus
red: Corpus Callosum



[C. Poupon et al. 2008]



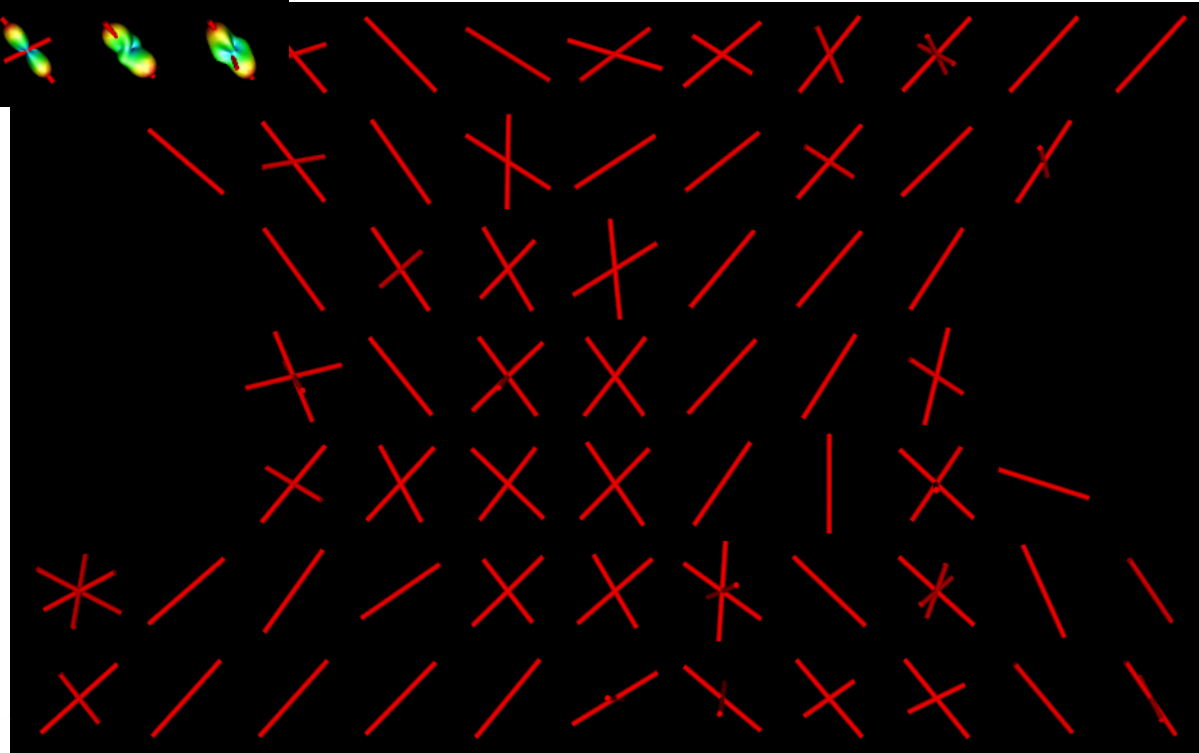
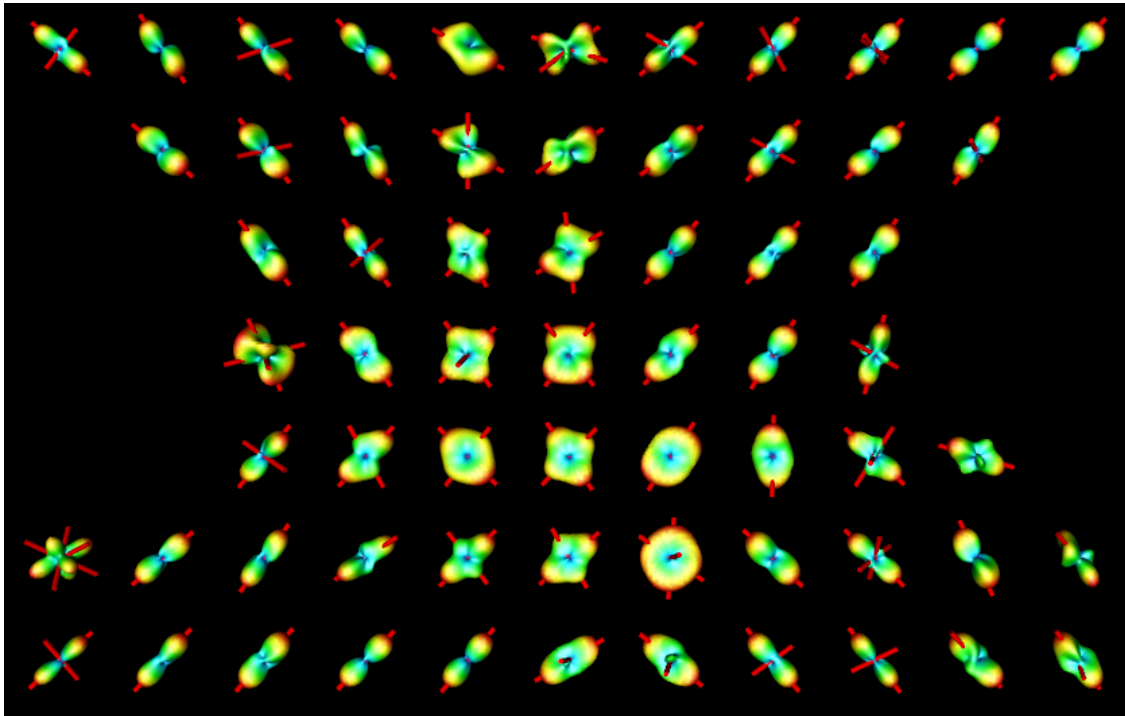
Spacing: 3x3x3

$b = 1500\text{s}/\text{mm}^2$

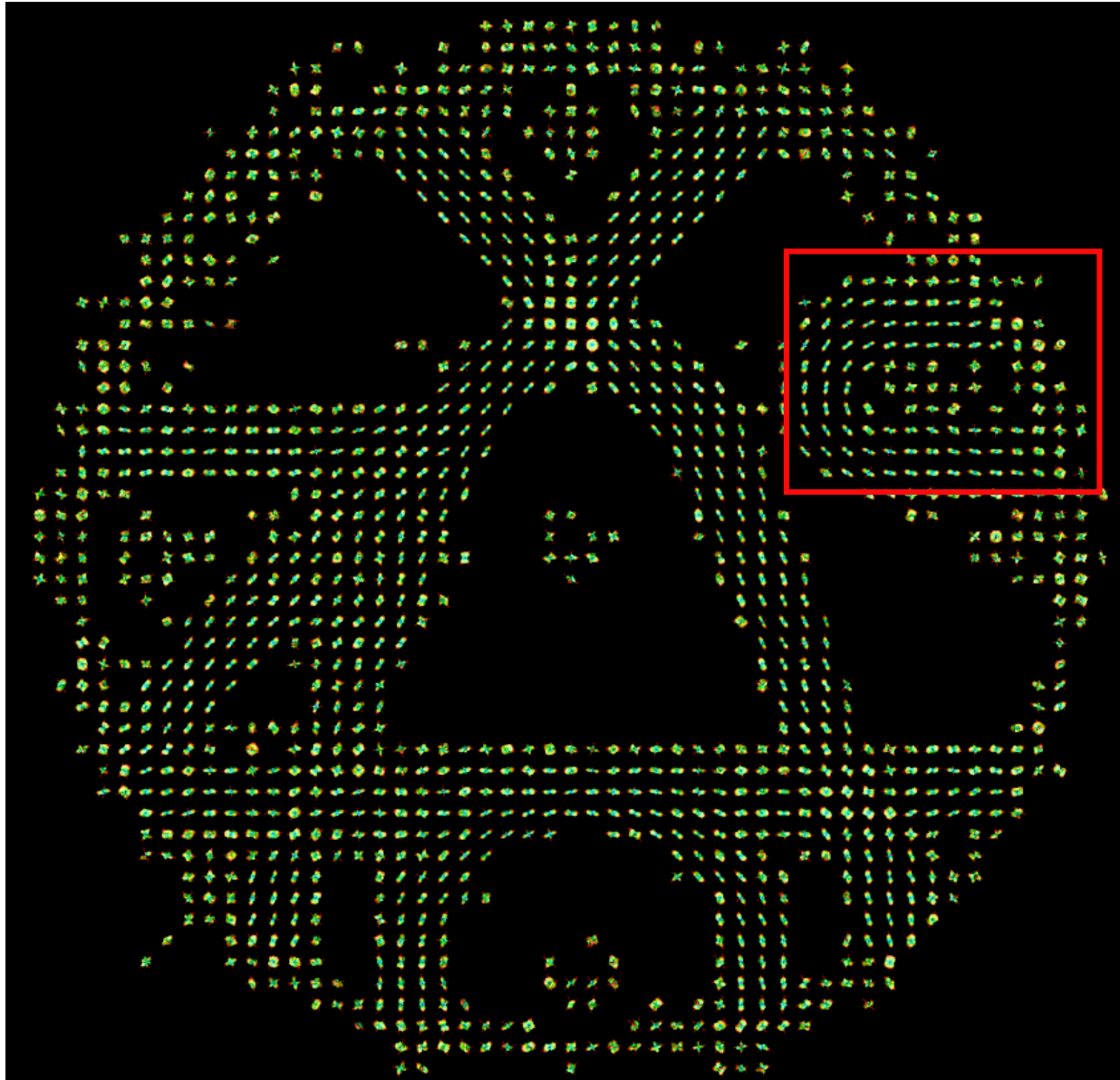
64 DWIs + T0

ODF-Tuch

[C. Poupon et al. 2008]



[C. Poupon et al. 2008]



Spacing: 3x3x3

$b = 1500\text{s}/\text{mm}^2$

64 DWIs + T0

ODF-Tuch

[C. Poupon et al. 2008]

