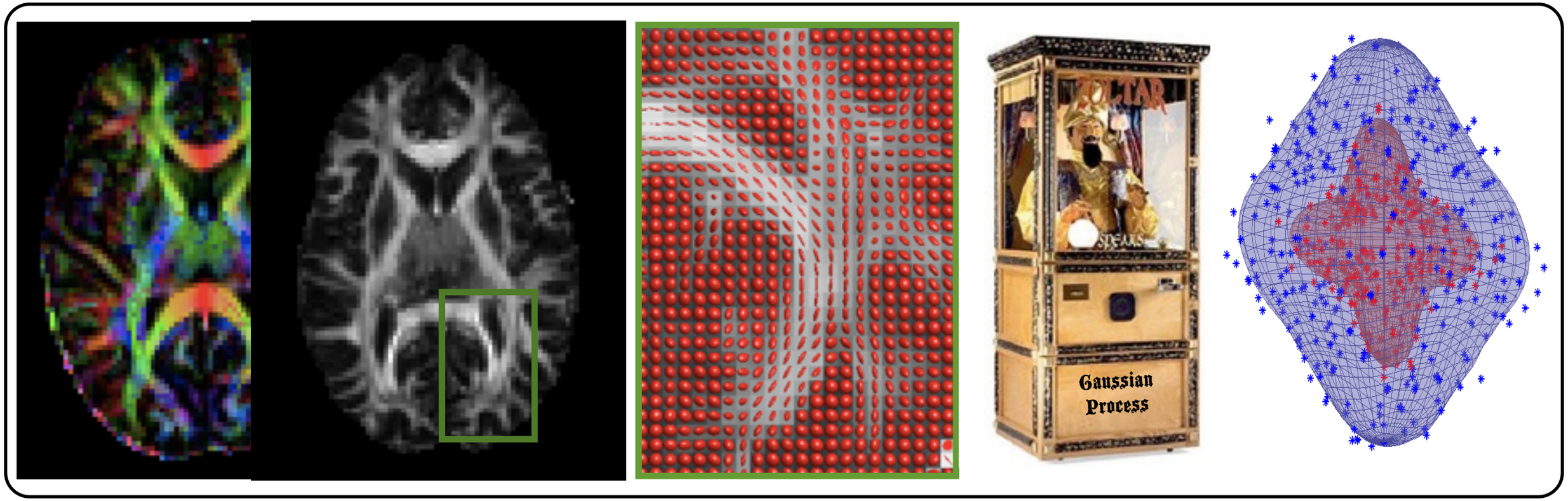
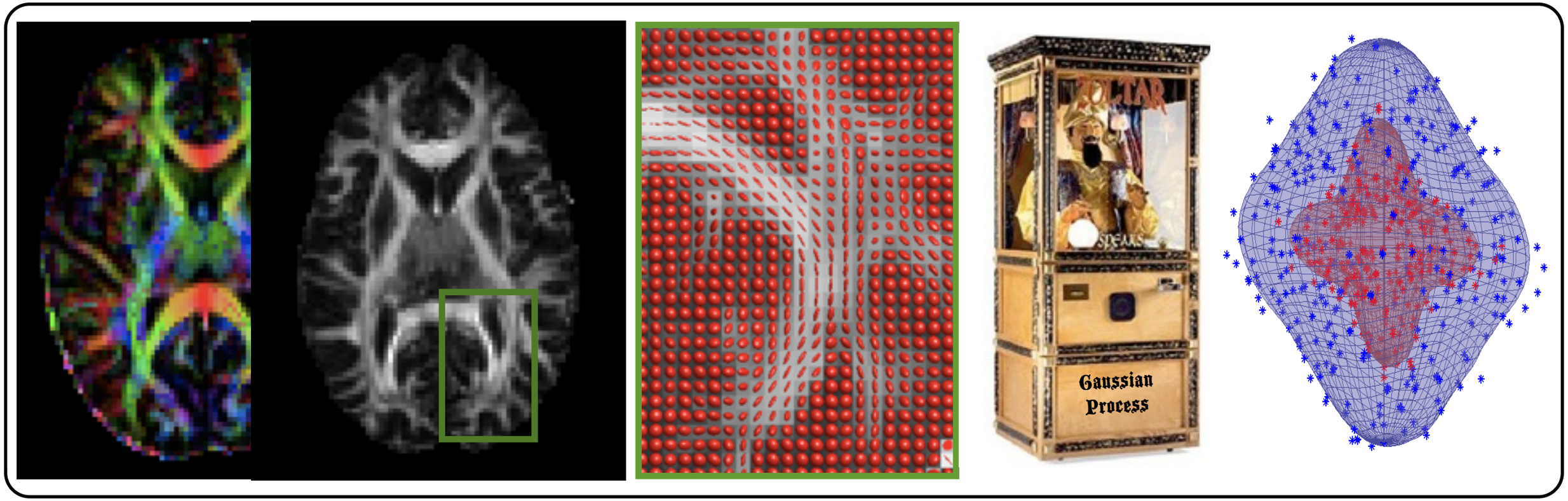
Diffusion MRI, Distortion Correction & DTI



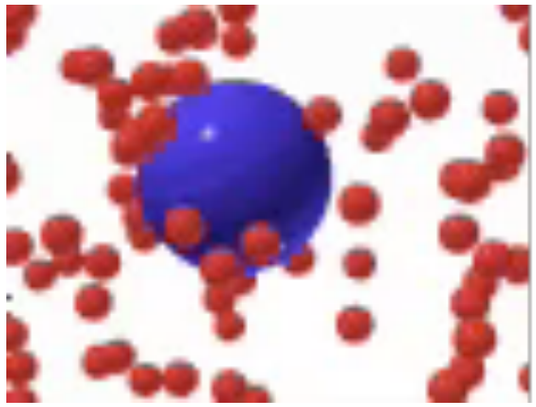
Overview

- What is Diffusion? Diffusion-weighted MRI
- Diffusion Tensor Model and DTI
- Pre-processing of Diffusion data



Diffusion - Brownian Motion

Robert Brown (1773-1858)



Molecules are in constant motion at non-zero absolute temperatures

Diffusion = thermally-driven random motion

Diffusion - Brownian Motion

Albert Einstein (1879-1955)

How can we describe this motion?
For an ensemble of molecules, in *n*-dimensional space:

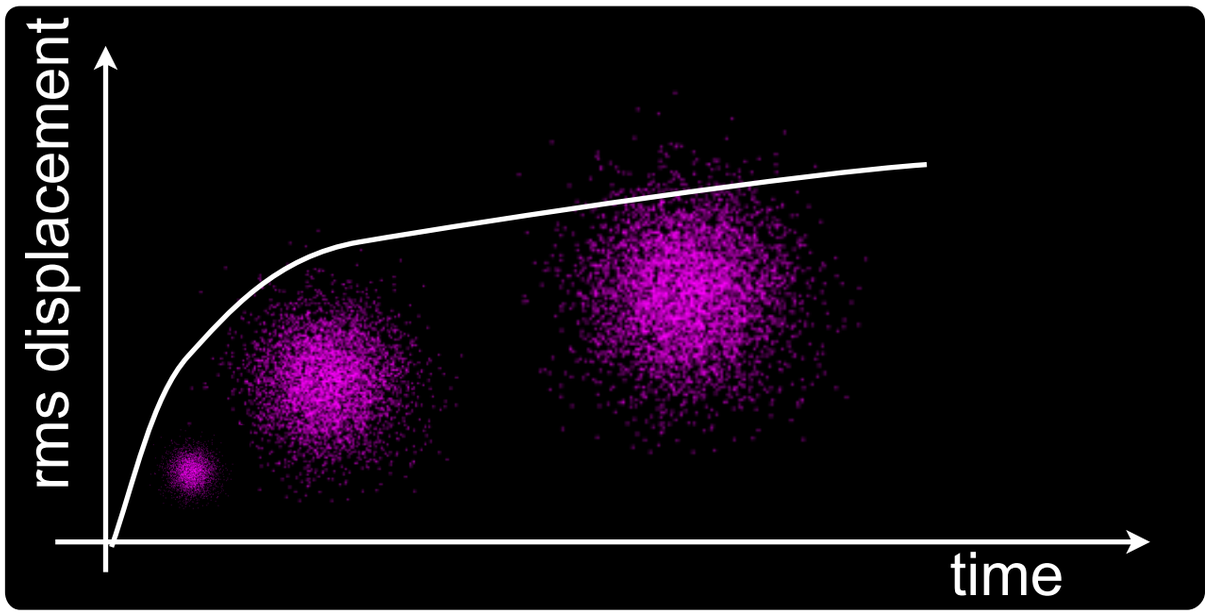
$$< x^2 >= 2nDt$$

mean squared displacement

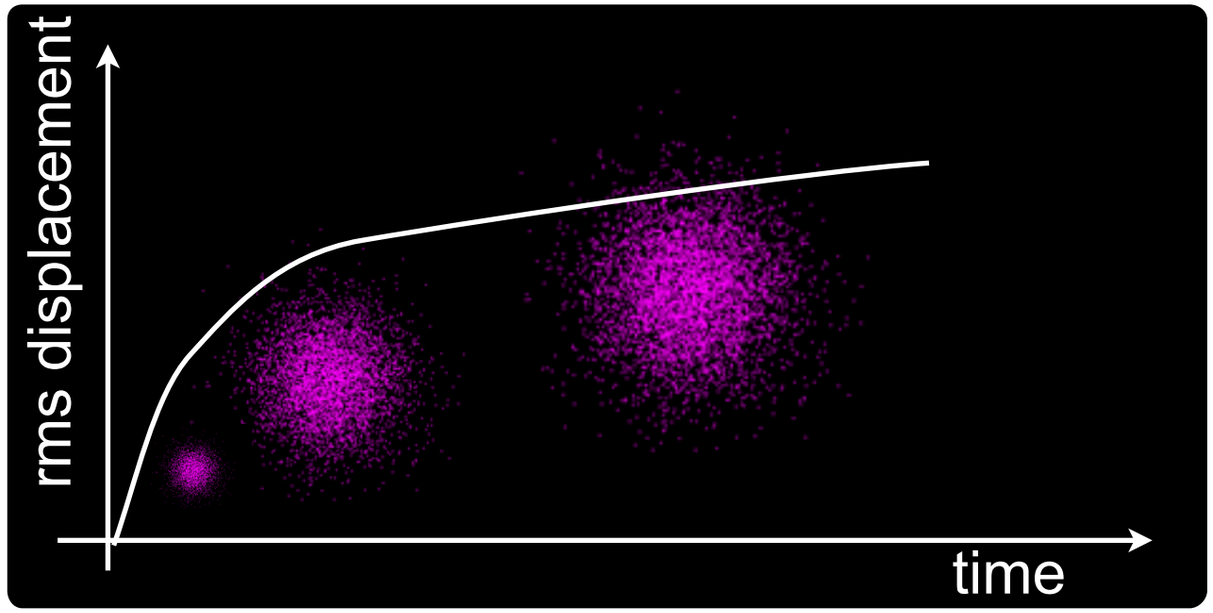
Diffusion coefficient

time

Valid for a homogeneous, barrier-free medium.



Diffusion - Brownian Motion



$$< x^2 >= 2nDt$$

D ~ 2.4 μm²/ms
t~50ms

➡ $x=\sqrt{6Dt}$ *~27μm*

Diffusion - Brownian Motion

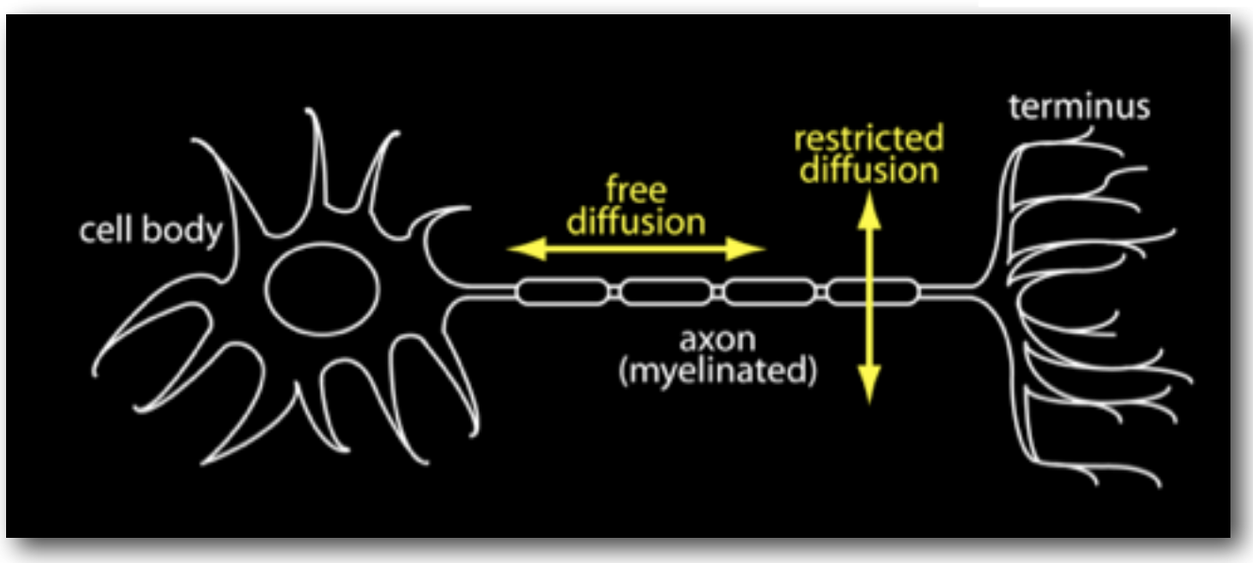
Another way to describe Einstein's equation:

For a barrier-free medium, **diffusion displacements of an ensemble follow a Normal distribution** with N(0, 2tD):

- Zero-mean displacement
(some will diffuse left, some will diffuse right, equal distances on average)

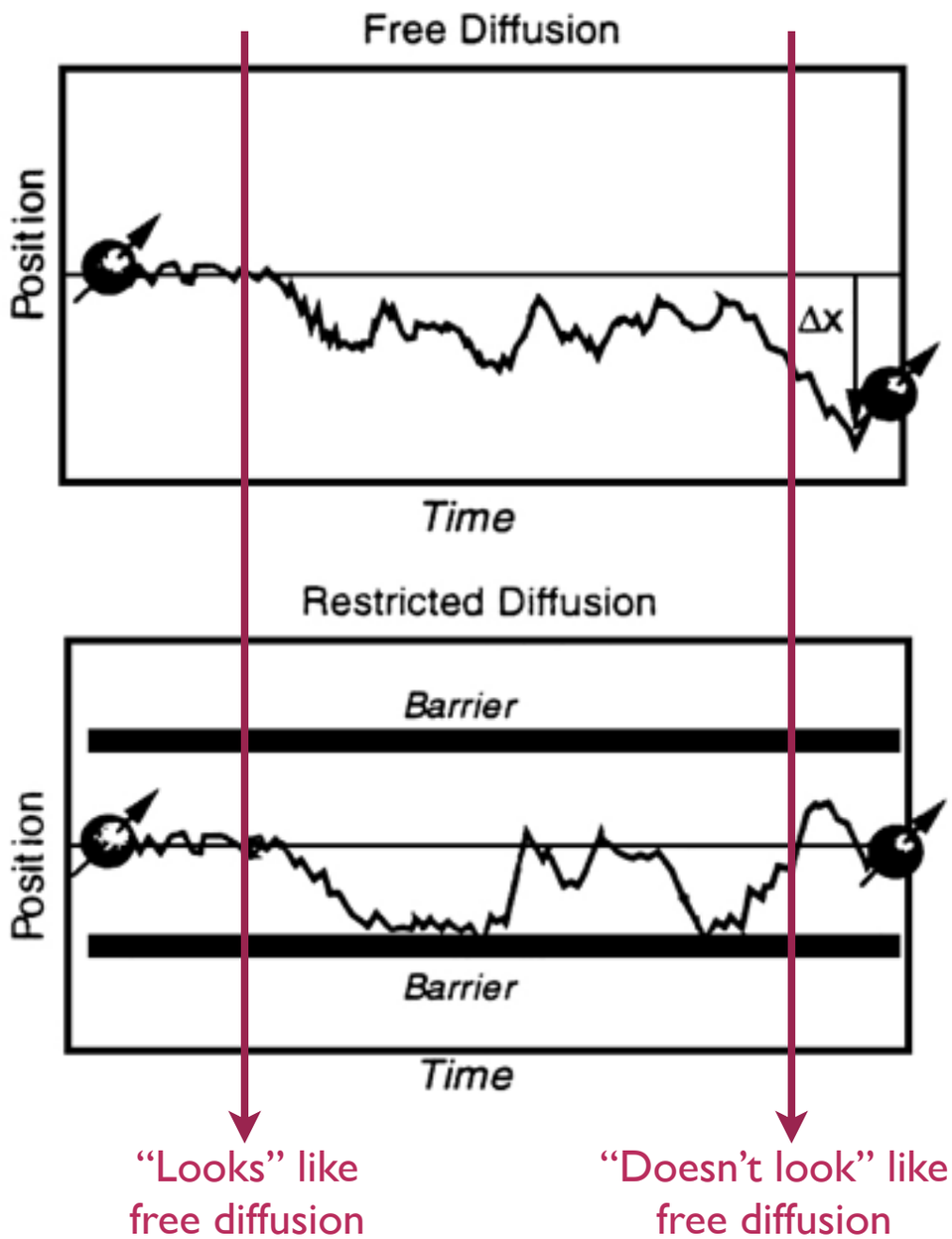
- Variance proportional to time and the diffusion coefficient

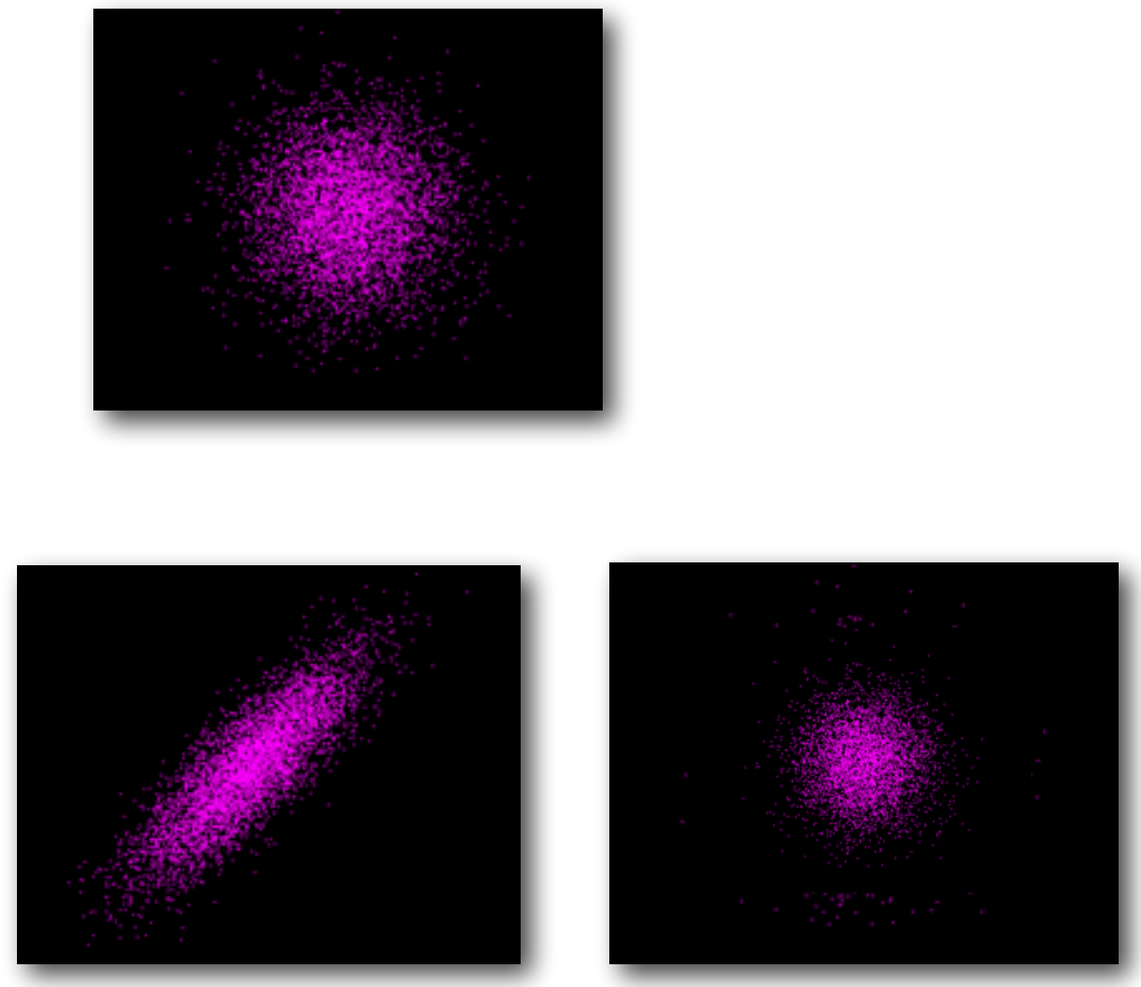
Water Diffusion in the Brain. Why is it Interesting?



Diffusion is restricted by tissue boundaries, membranes, etc.
Marker for tissue microstructure (healthy and pathology)
Diffusion is **anisotropic** in white matter [Beualieu, NMR Biomed, 2002]

Apparent Diffusion

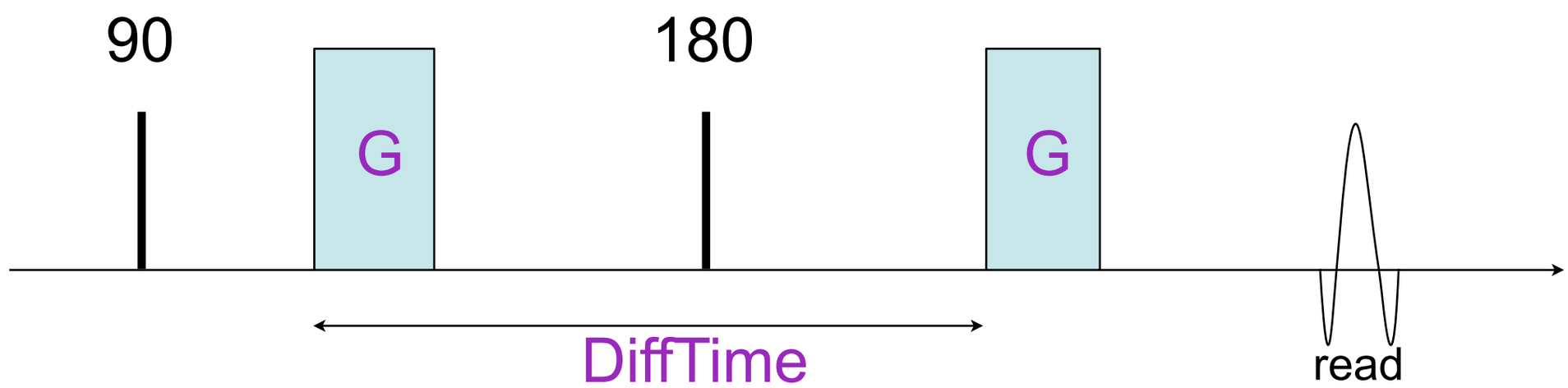




Observed diffusion in tissues depends on the experiment =
"Apparent diffusion" &
"Apparent diffusion coefficient" (ADC)

Measuring Diffusion with MRI: Diffusion-Weighted Imaging (DWI)

Pulsed-Gradient Spin-Echo Sequence:
To achieve diffusion-weighting along a direction **x**, apply strong magnetic field gradients along **x**.



If particles diffuse during the allowed time (DiffTime), a signal attenuation is observed, compared to the signal with G=0.

[Stejskal & Tanner, 1965]

Measuring Diffusion with MRI: Diffusion-Weighted Imaging (DWI)

Diffusion-weighting is modulated by:

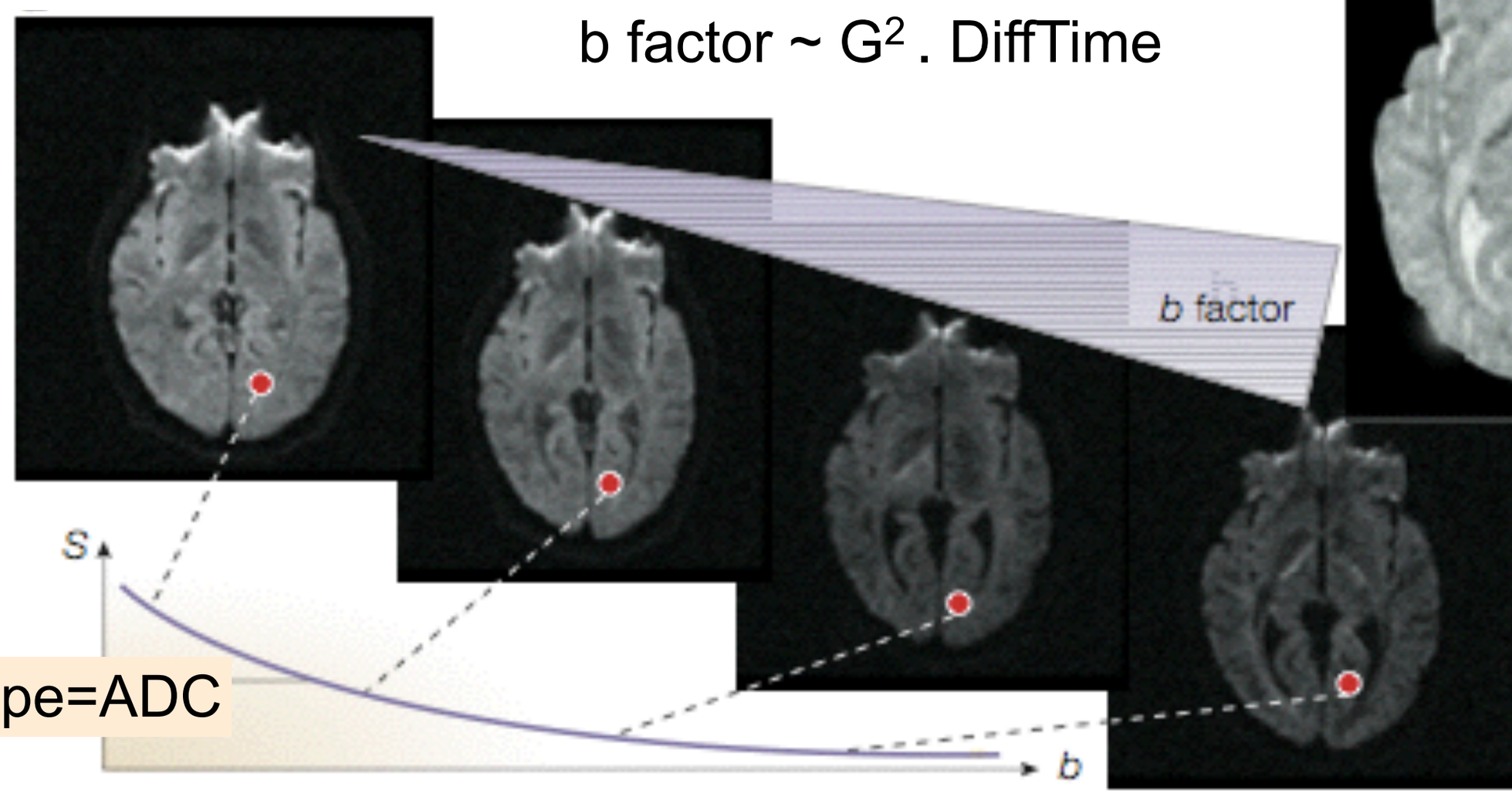
. Gradient **strength**

. Diffusion **time**

. Gradient **orientation**

b factor ~ G^2 . DiffTime

b=0

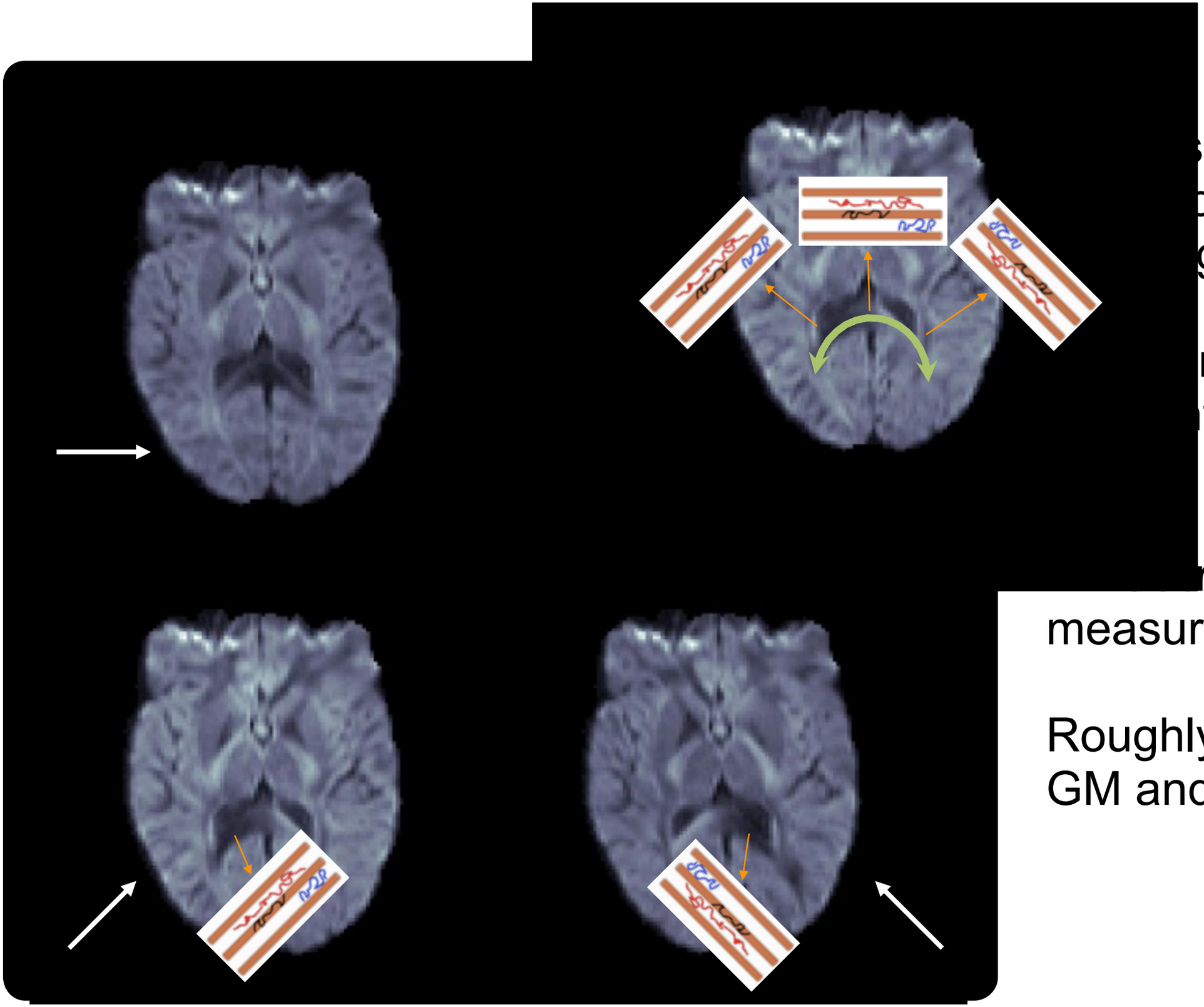


slope=ADC

$S=S_0\exp(-b*ADC)$

[LeBihan, 2003]

Orientation Contrast in DWI



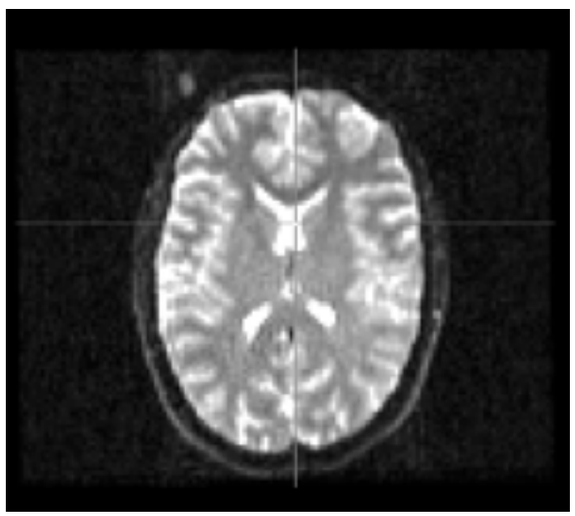
se diffusion is
opic in WM,
g a gradient G
ifferent
ns **x**, gives
t contrast in

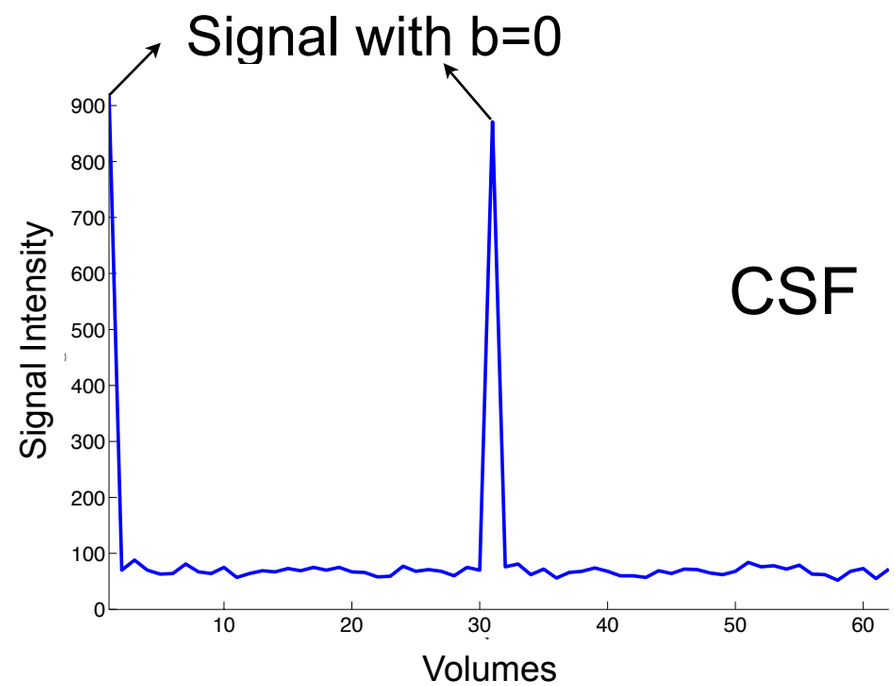
ropic
measurements in WM

Roughly **Isotropic** in GM and CSF.

A Typical DWI Protocol

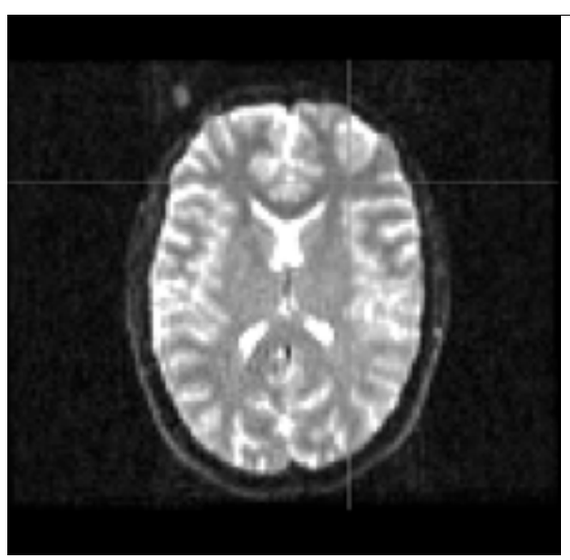
- Normally a few (at least one) b=0 volumes acquired, along with volumes at higher b (~1000 s/mm^2).
- Different gradient directions are applied for the high b volumes.

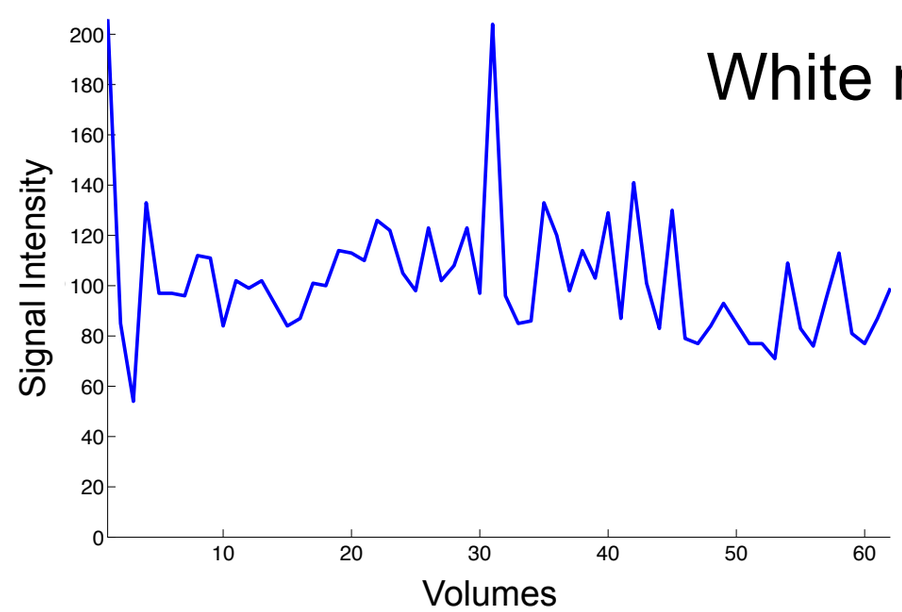




CSF

relates to ADC





White matter

relates to anisotropy

DWI Summary

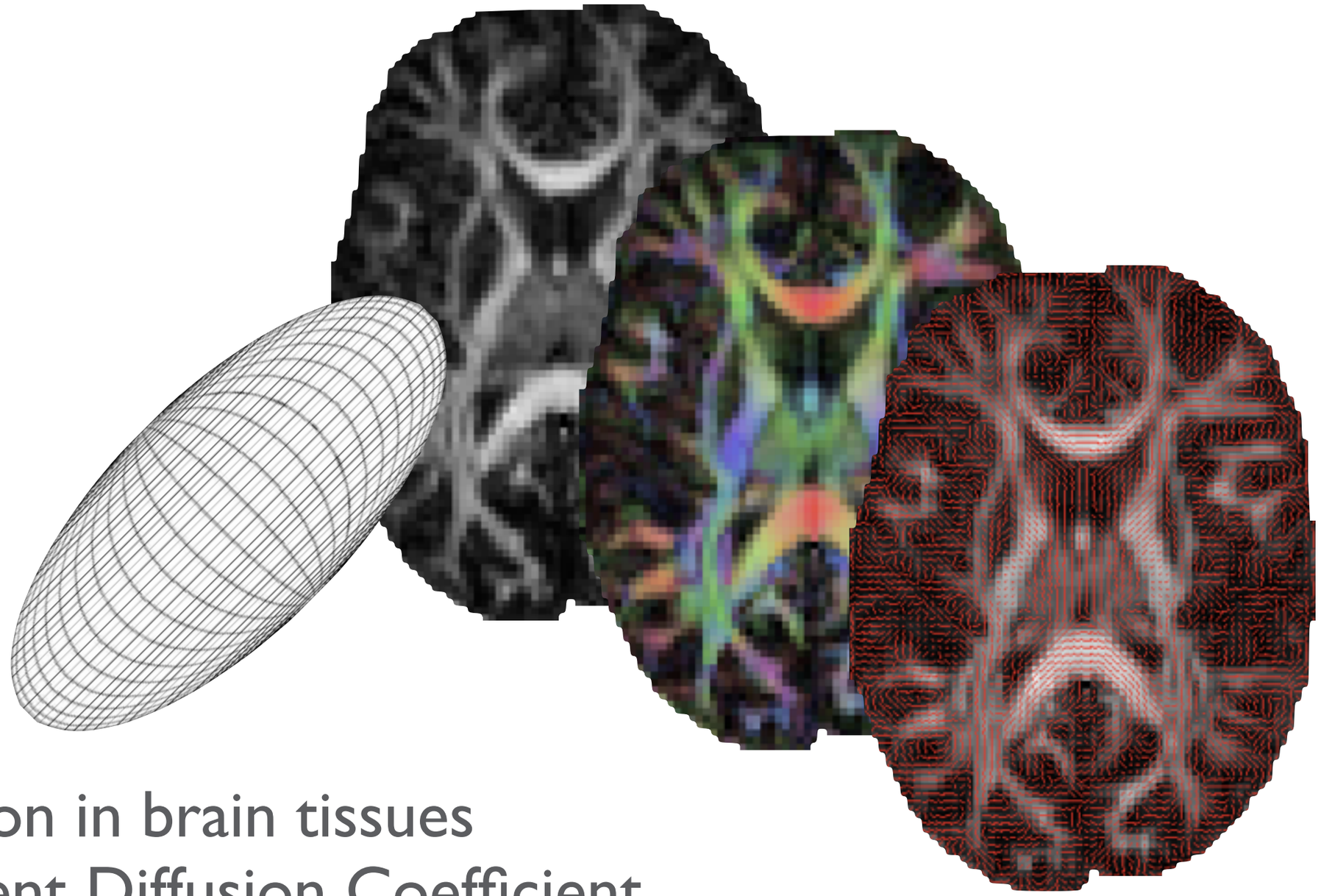
- Images acquired with a Gradient along **x**, have contrast that is sensitive to diffusion of water molecules along **x**.

- When diffusion occurs, signal is attenuated compared to the one with no diffusion-weighting.

- In WM, measurements are anisotropic.

- In GM and CSF, measurements are roughly isotropic.

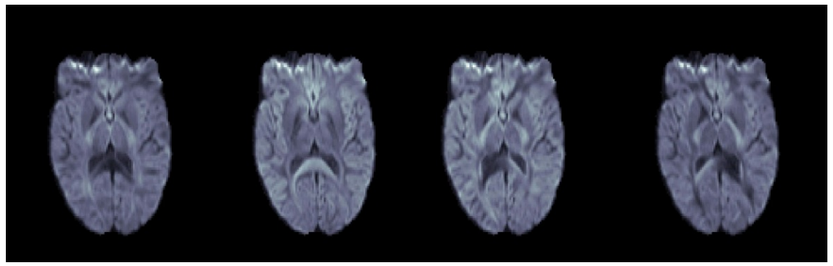
Diffusion Tensor Imaging - basic principles



- Diffusion in brain tissues
- Apparent Diffusion Coefficient
- Diffusion Tensor model
- Tensor-derived measures

Diffusion Tensor Imaging (DTI)

- Apply **the diffusion tensor model** to a set of DWI images.

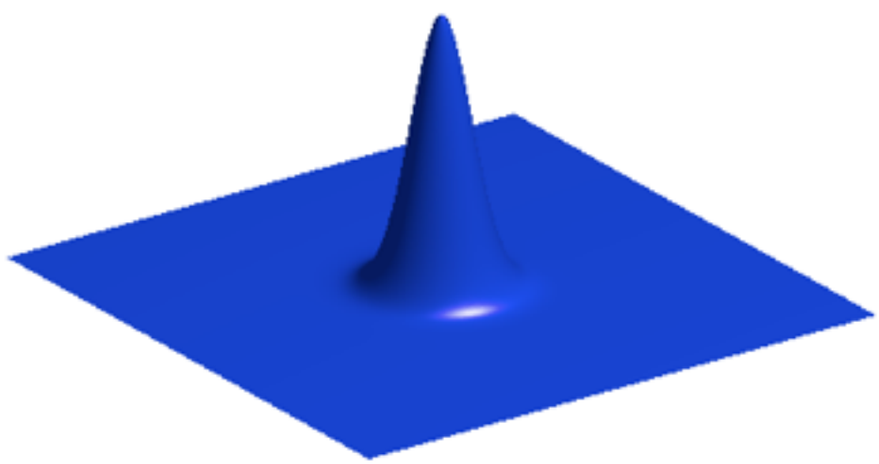


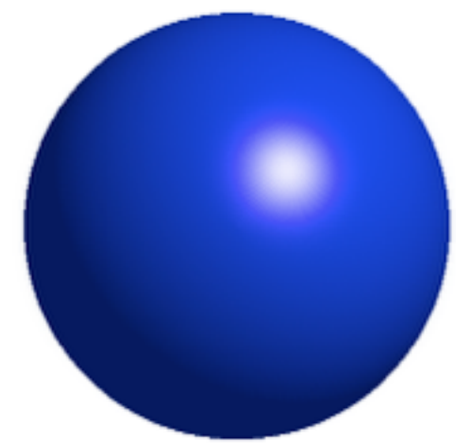
Diffusion Tensor Imaging (DTI)

Two dimensions

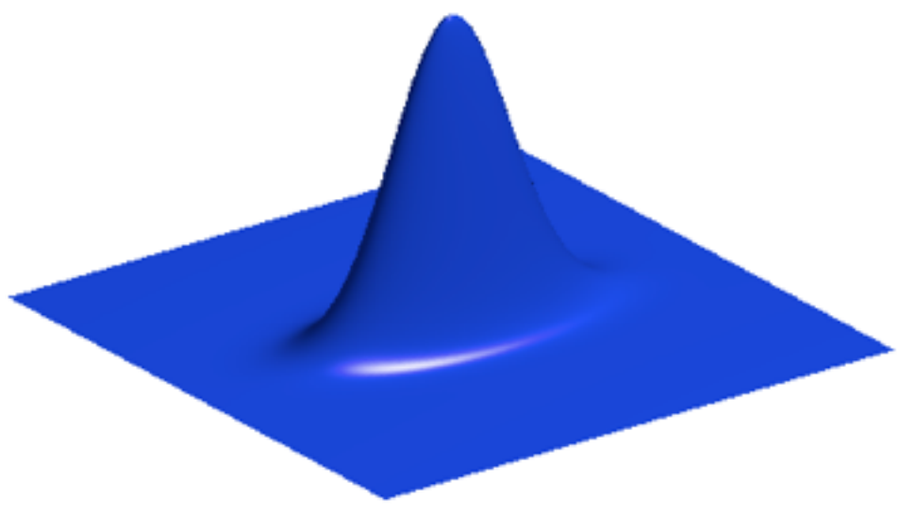
Three dimensions

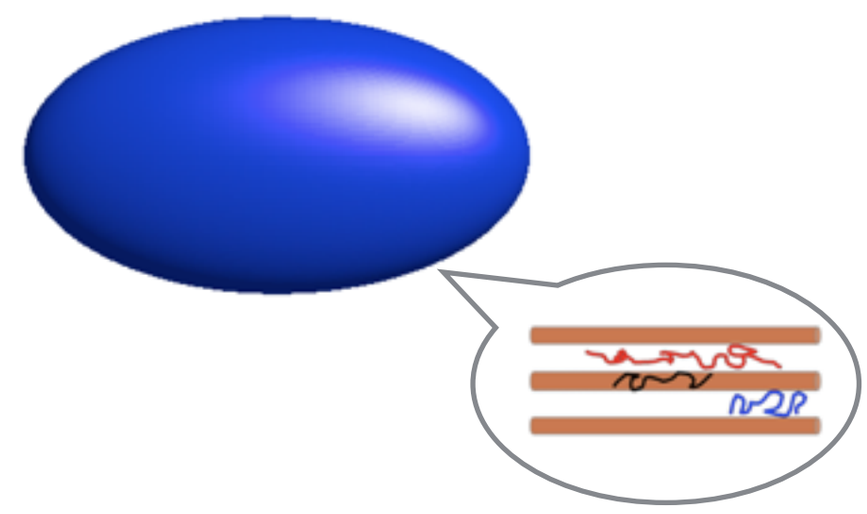
Scalar D (same D for all directions)





Tensor **D** - DTI (D can be different for different directions)





Diffusion Tensor Imaging (DTI)

Diffusion Tensor Model. In each voxel:

b-factor for gradient *j* (known)

Unit vector representing the direction of gradient *j* (known)

$$s_j = s_0 \exp(-b_j \, \boldsymbol{x}_j^T \, \boldsymbol{D} \, \boldsymbol{x}_j)$$

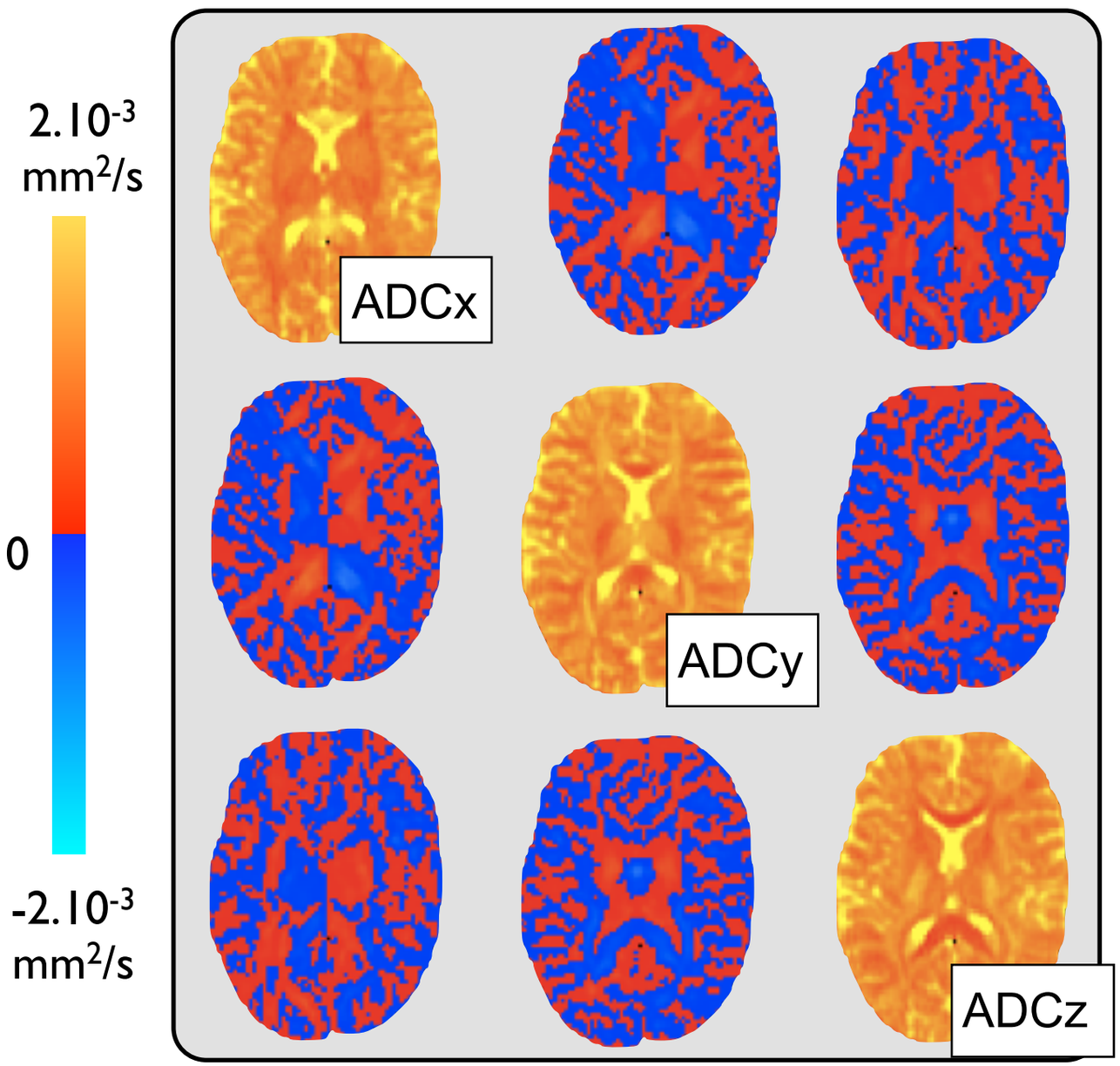
Signal measured after applying a Gradient *j* with direction $\boldsymbol{x}_j$ and b-factor b_j (measured)

3x3 Diffusion Tensor (unknown)

Signal measured with no diffusion gradient applied

[Basser, Biophys J,1994], [Basser et al , J Magn Res, 1994]

The Elements of the Diffusion Tensor



$$\mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}$$

- Tensor is **symmetric** (6 unknowns)

- **Diagonal Elements** are proportional to the diffusion displacement variances (**ADCs**) along the three directions of the experiment coordinate system

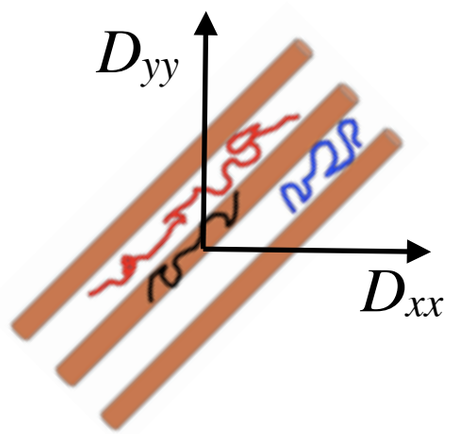
-**Off-diagonal Elements** are proportional to the **correlations** (covariances) of displacements along these directions

$N_3\ (0,\ 2t\mathbf{D})$

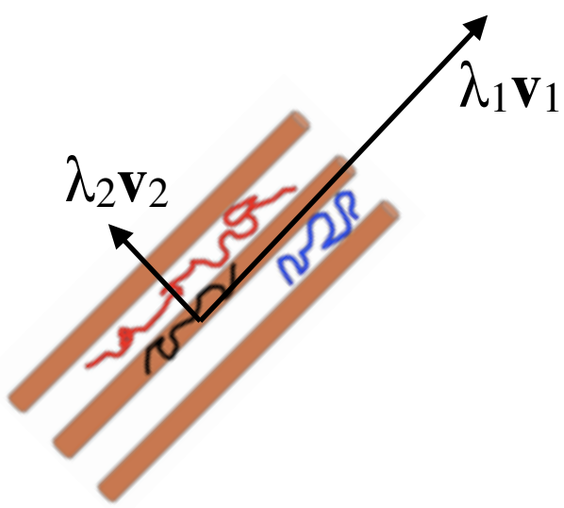
The Diffusion Tensor Eigenspectrum

$$\mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix}$$

Once D is estimated, we get ADCs along the scanner's coordinate system. But we want ADCs along a local coordinate system in each voxel, determined by the anatomy.



Diagonalize the estimated tensor in each voxel



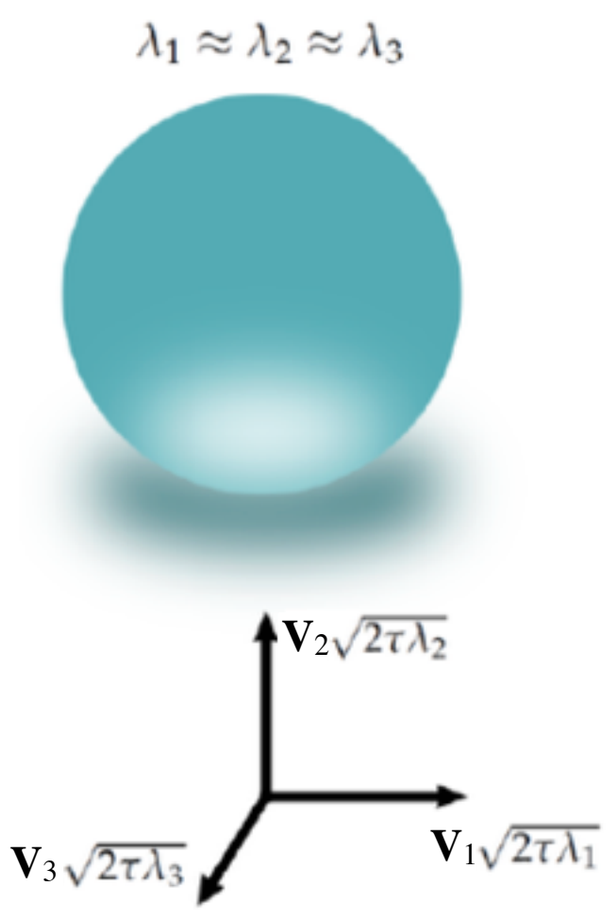
$$\mathbf{D} = [\mathbf{v_1}|\mathbf{v_2}|\mathbf{v_3}]^{\mathrm{T}} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} [\mathbf{v_1}|\mathbf{v_2}|\mathbf{v_3}]$$

eigenvectors - $\mathbf{v}_1$=direction of max diffusivity

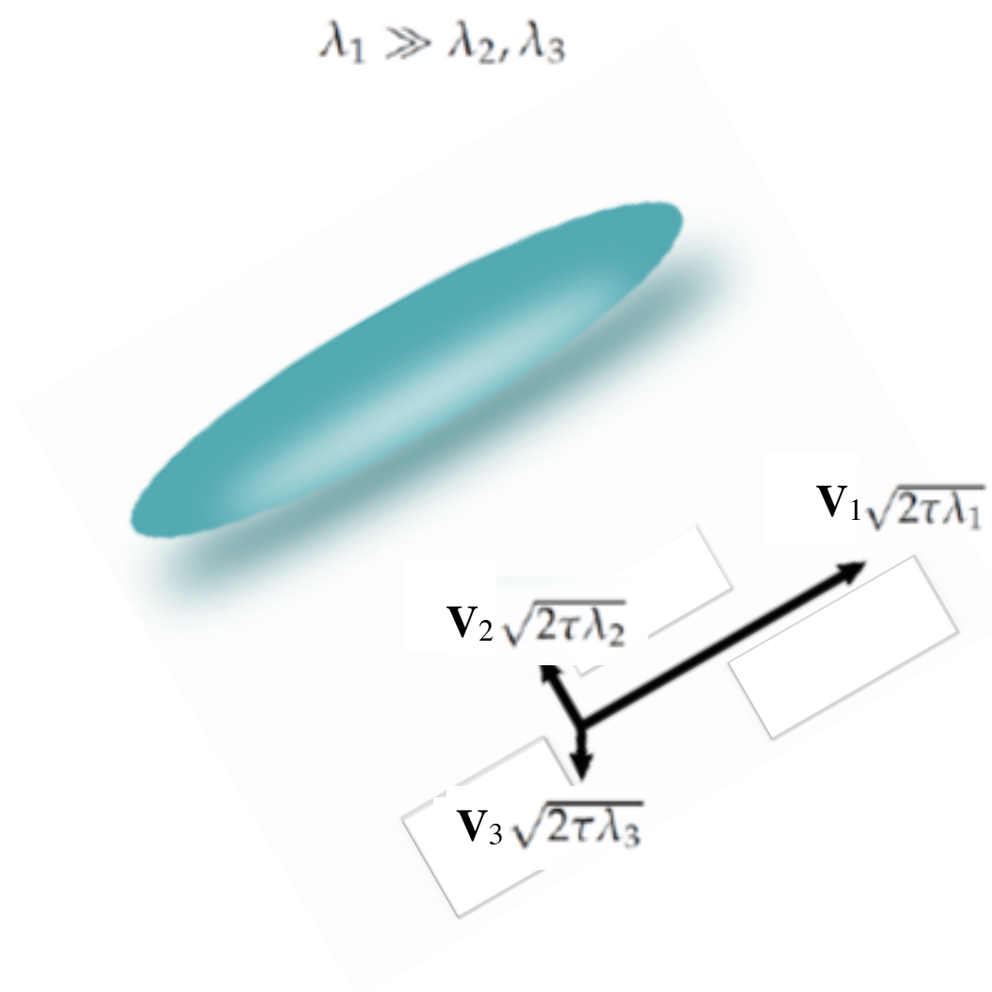
eigenvalues: ADCs along $\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3$

The Diffusion Tensor Ellipsoid

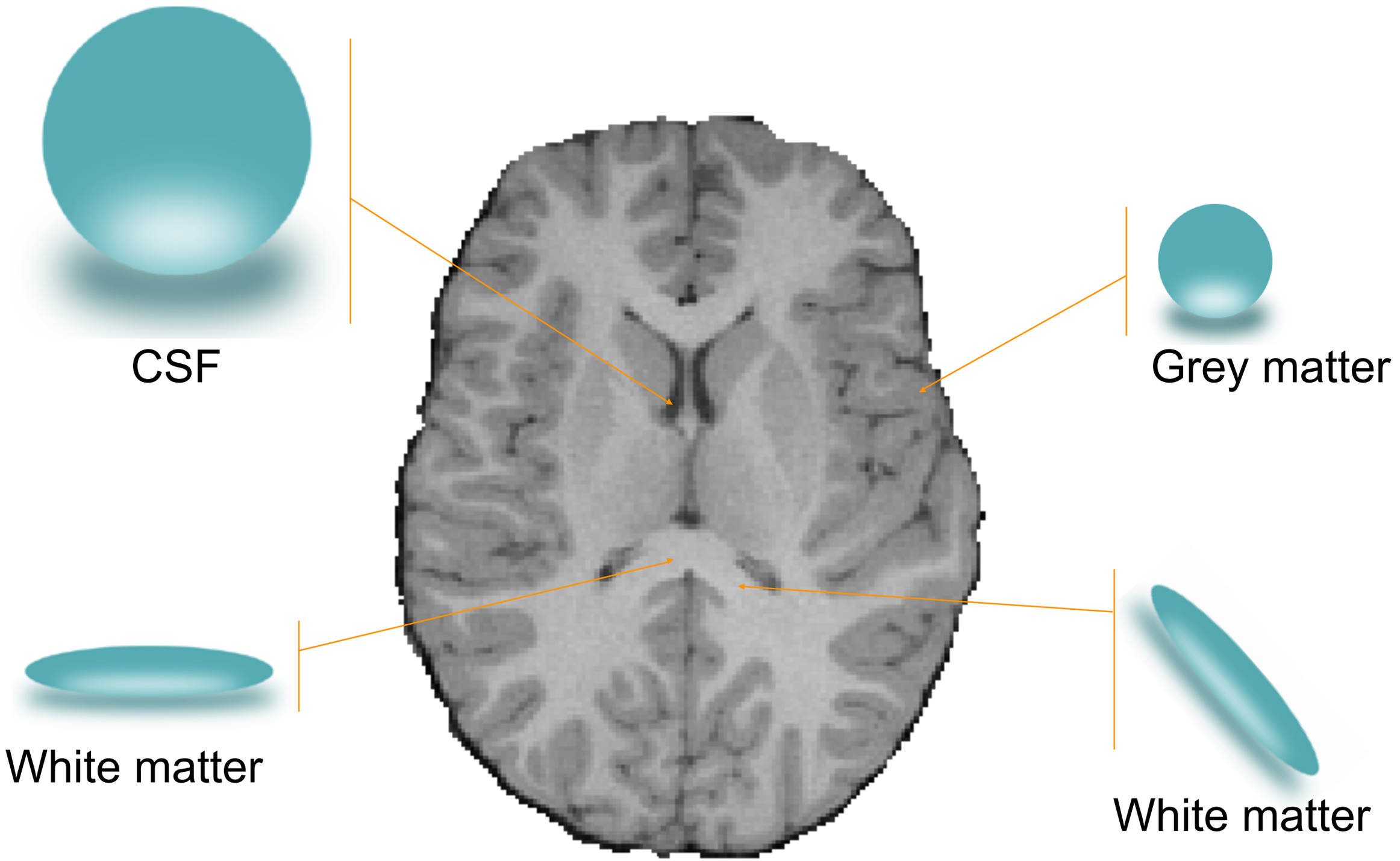
Isotropic voxel



Anisotropic voxel

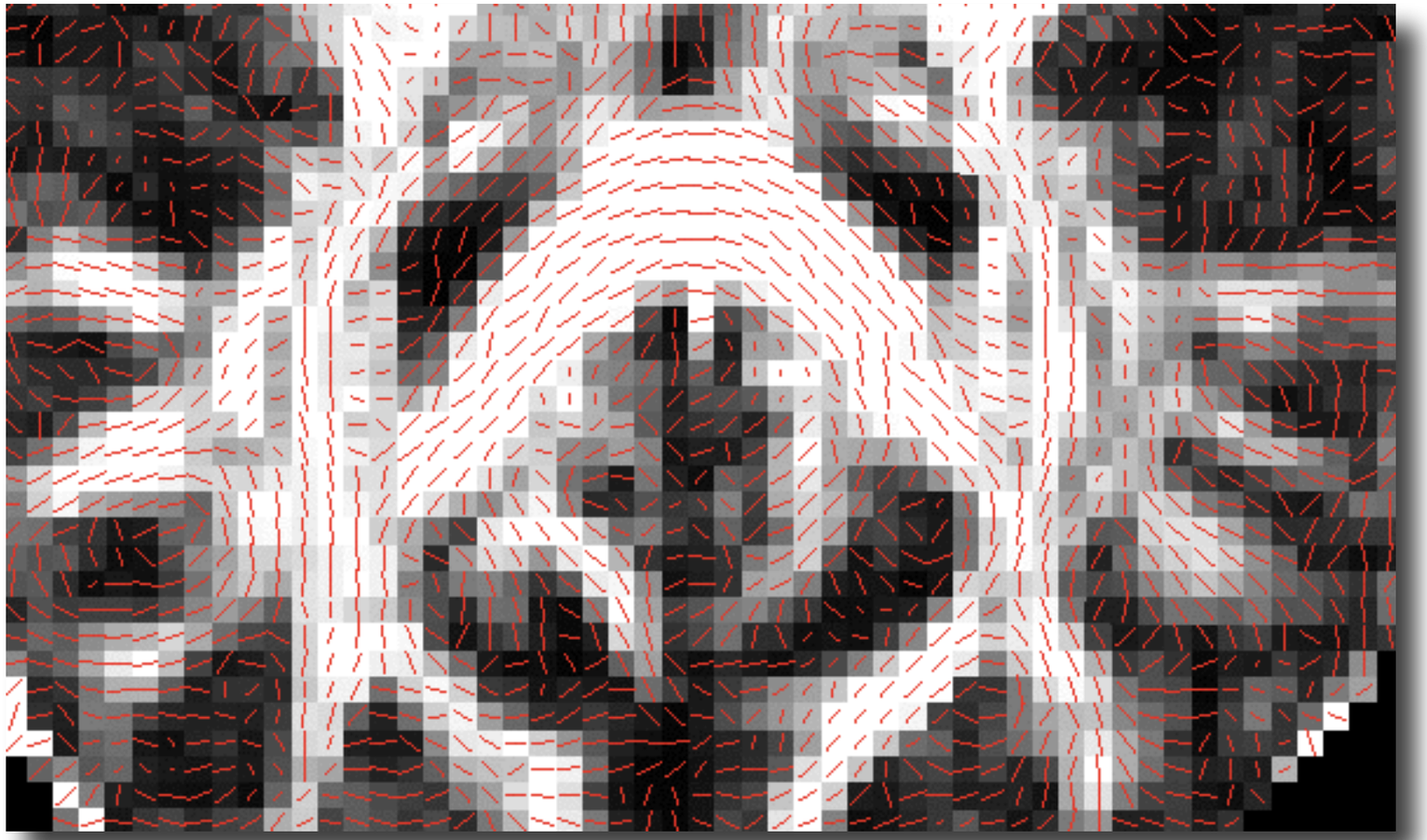


The Diffusion Tensor Ellipsoid

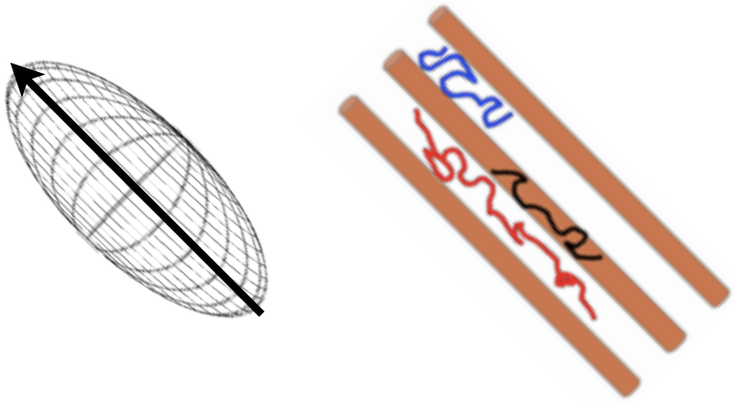


Estimates of Principal Fibre Orientation in WM

$\mathbf{v}_1$ map
Principal Diffusion Direction



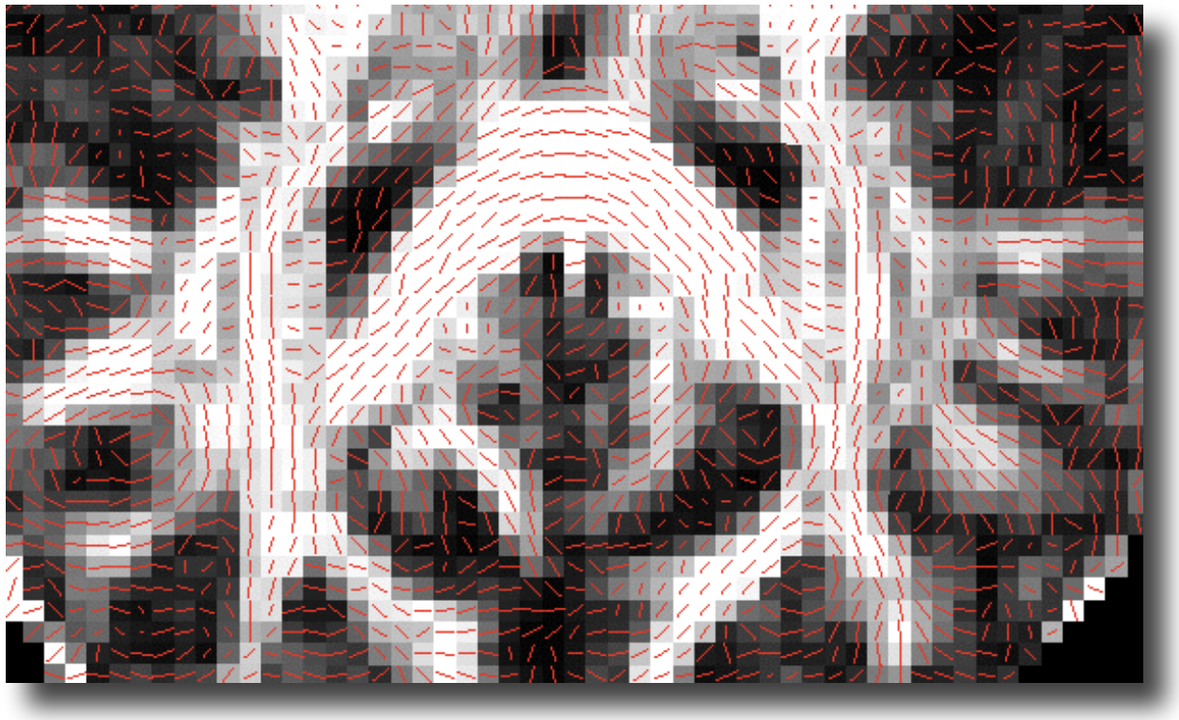
Principal Diffusion Direction

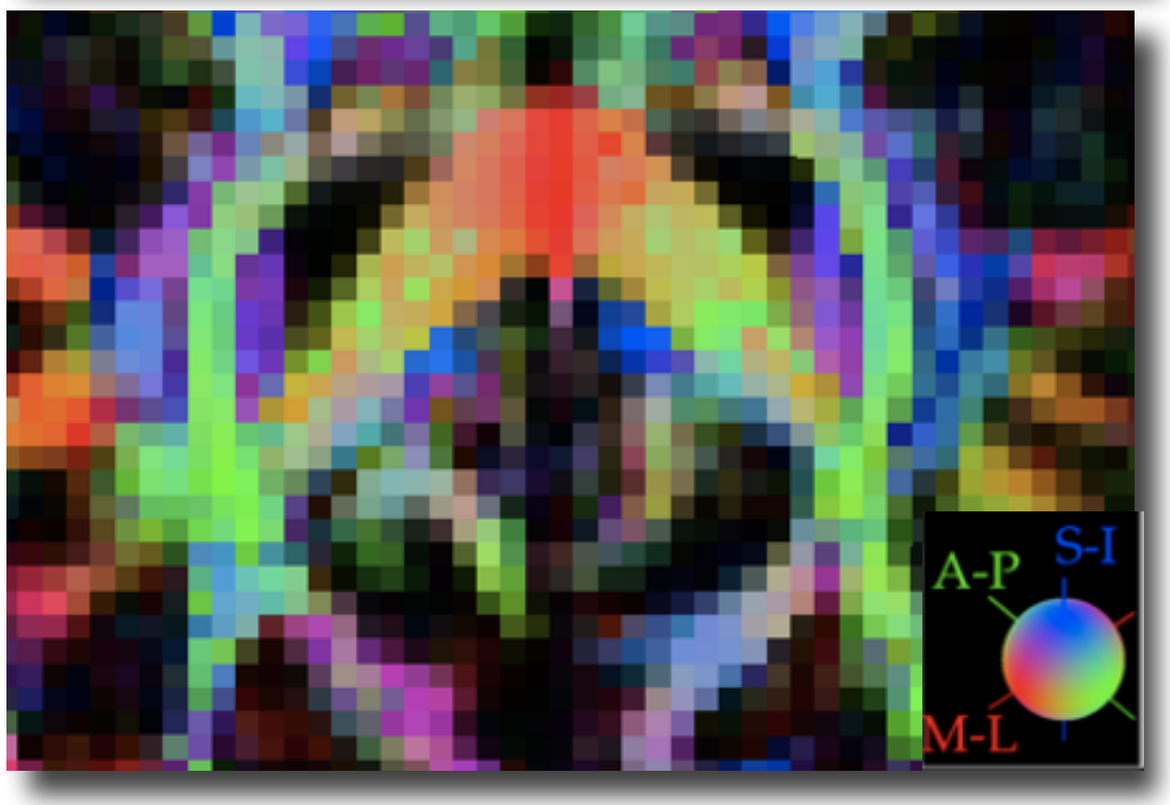


Assumption!!

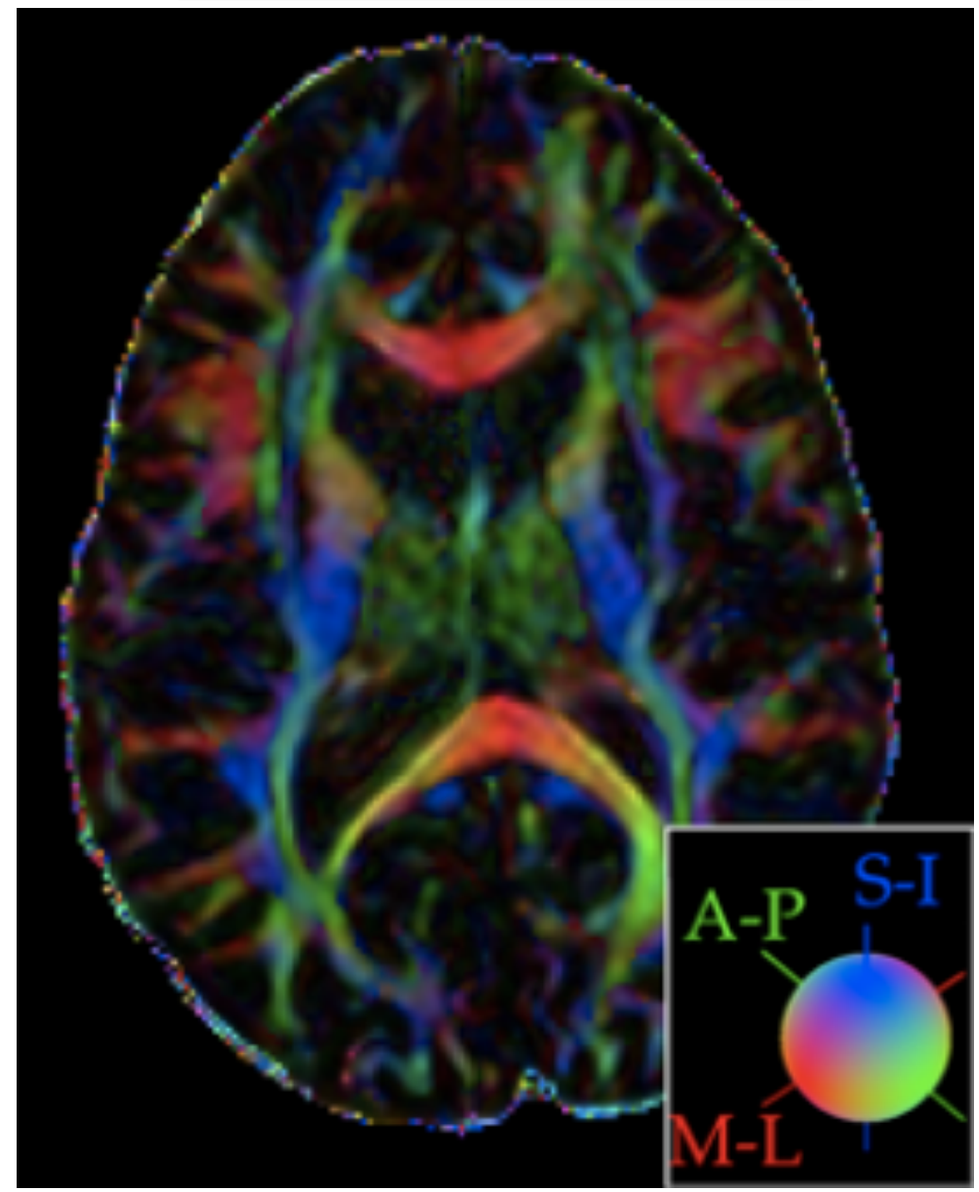
Direction of maximum diffusivity in voxels with anisotropic profile is an **estimate of the major fibre orientation**.

Estimates of Principal Fibre Orientation in WM

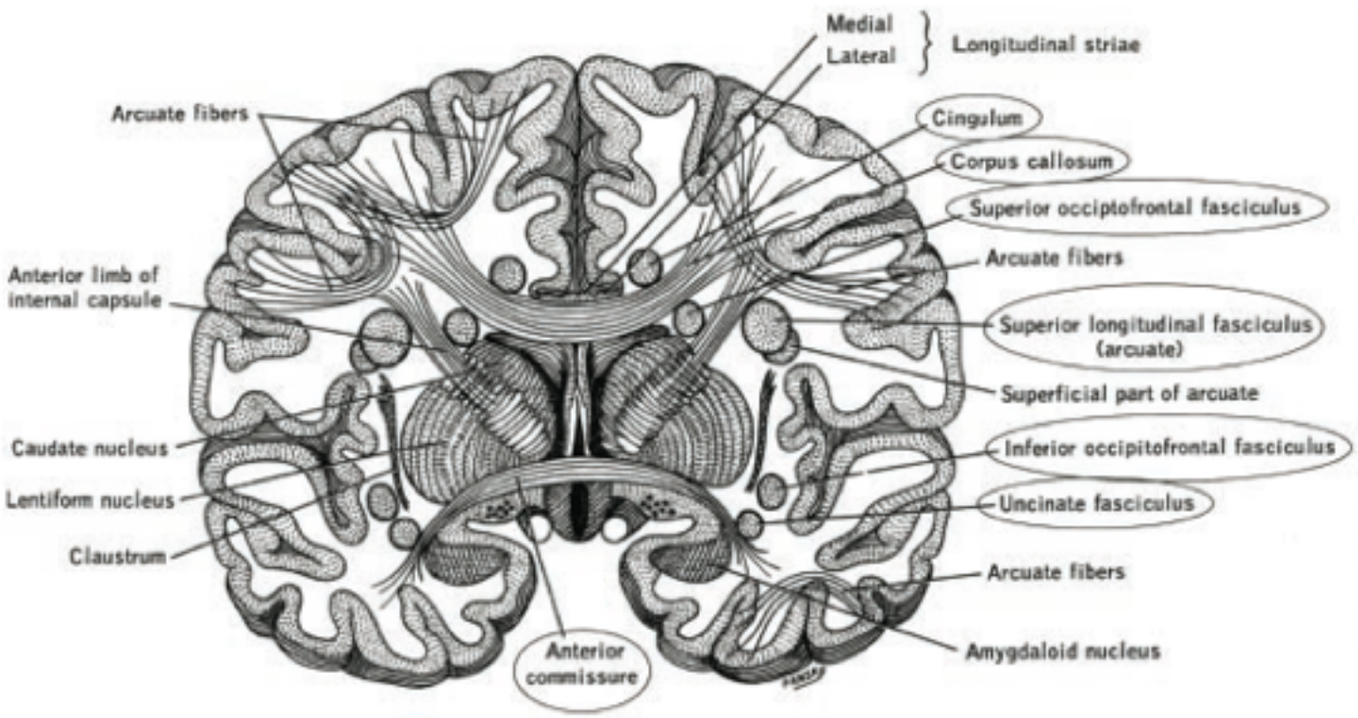


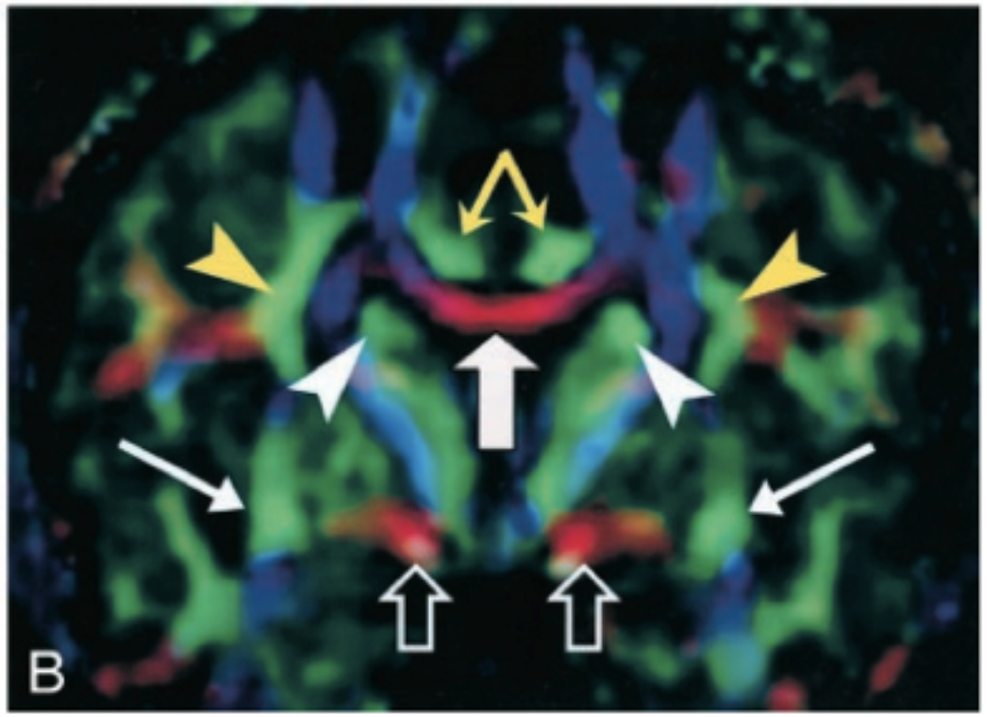


Colour-coded $\mathbf{v}_1$ map

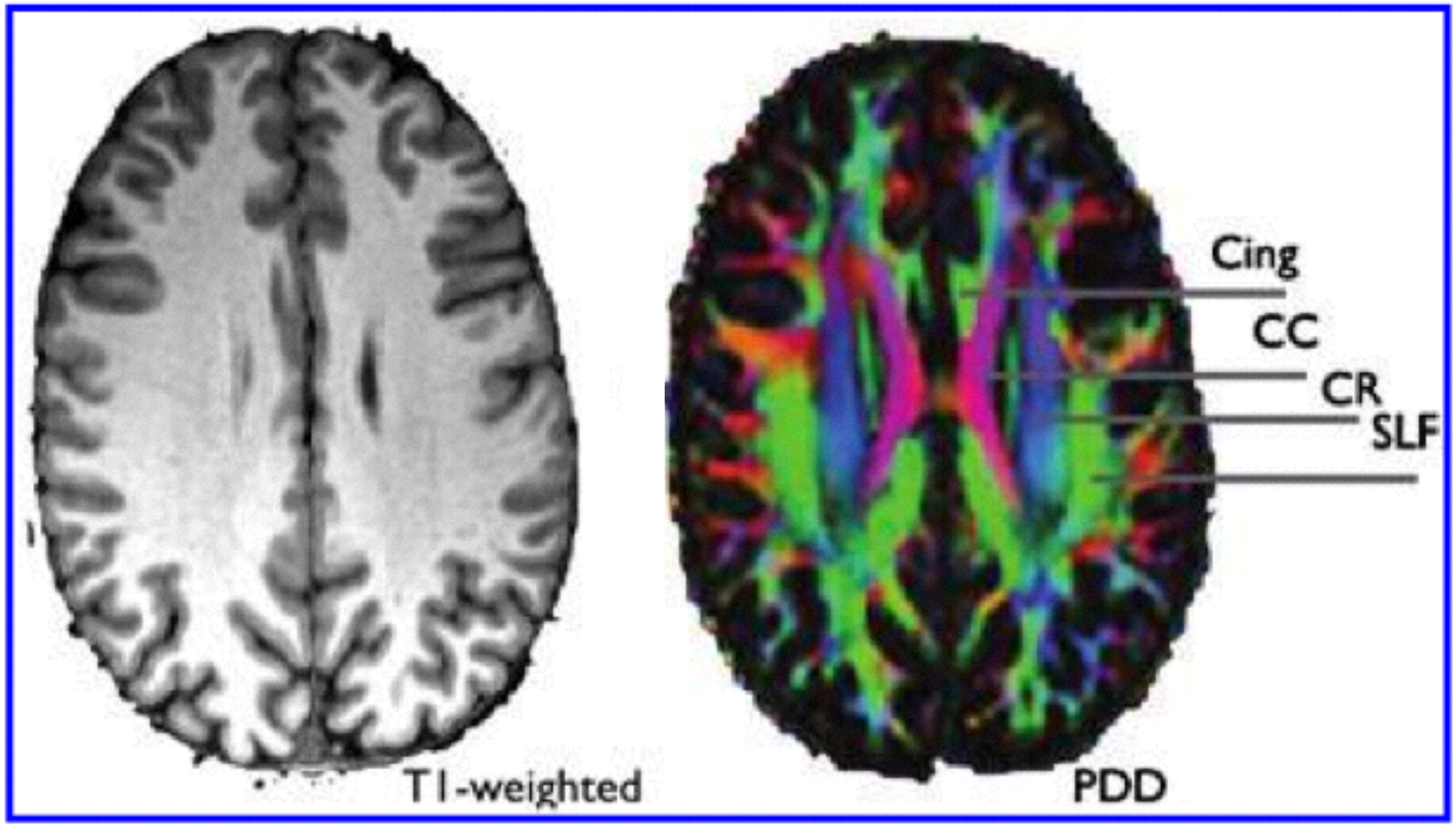


Estimates of Principle Fibre Orientation in WM





Directional contrast in DTI



Quantitative Diffusion Maps

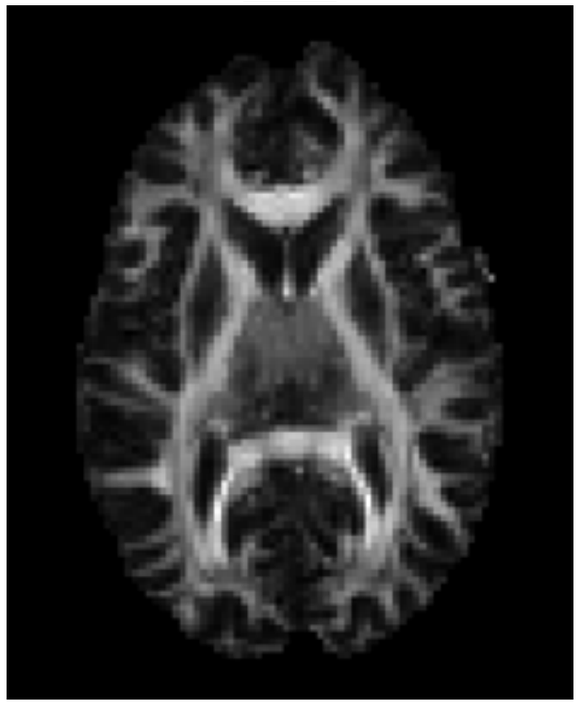
Fractional Anisotropy (FA) ~ Eigenvalues Variance (normalised)
Mean Diffusivity (MD) = Eigenvalues Mean

$$FA = \sqrt{\frac{3\sum_{i=1}^{3}(\lambda_i - \overline{\lambda})^2}{2\sum_{i=1}^{3}\lambda_i^2}}, \qquad FA \text{ in } [0,1]$$

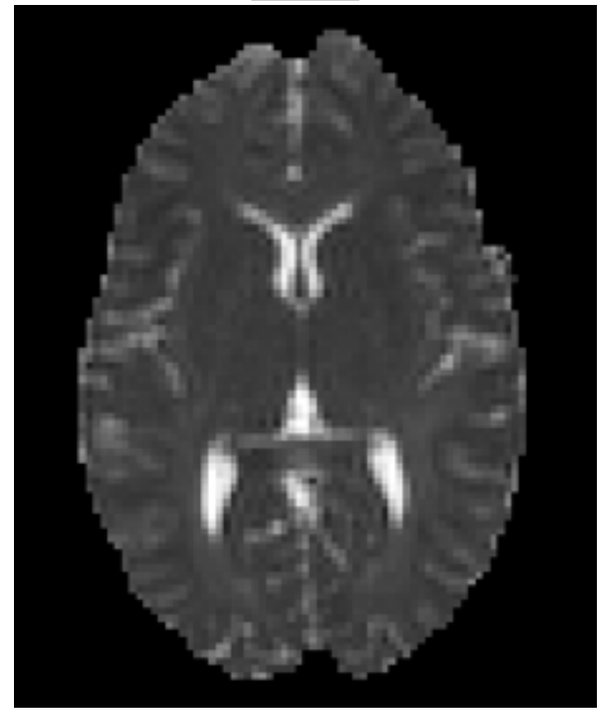
$$MD = \frac{D_{xx} + D_{yy} + D_{zz}}{3} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}$$

Quantitative Diffusion Maps

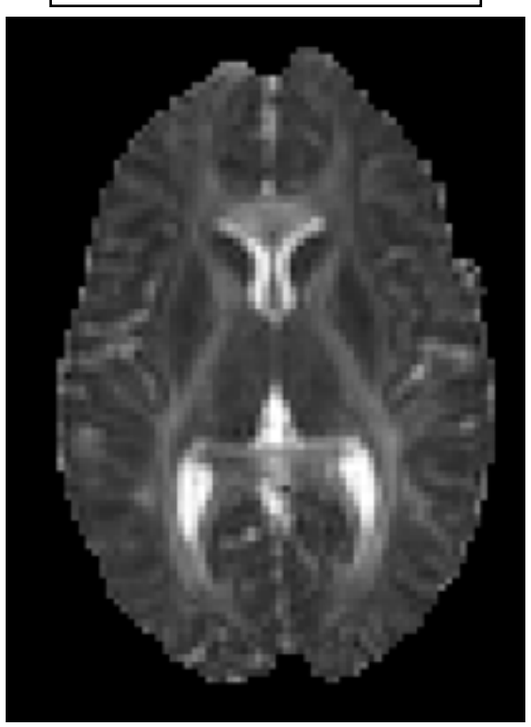
FA



MD



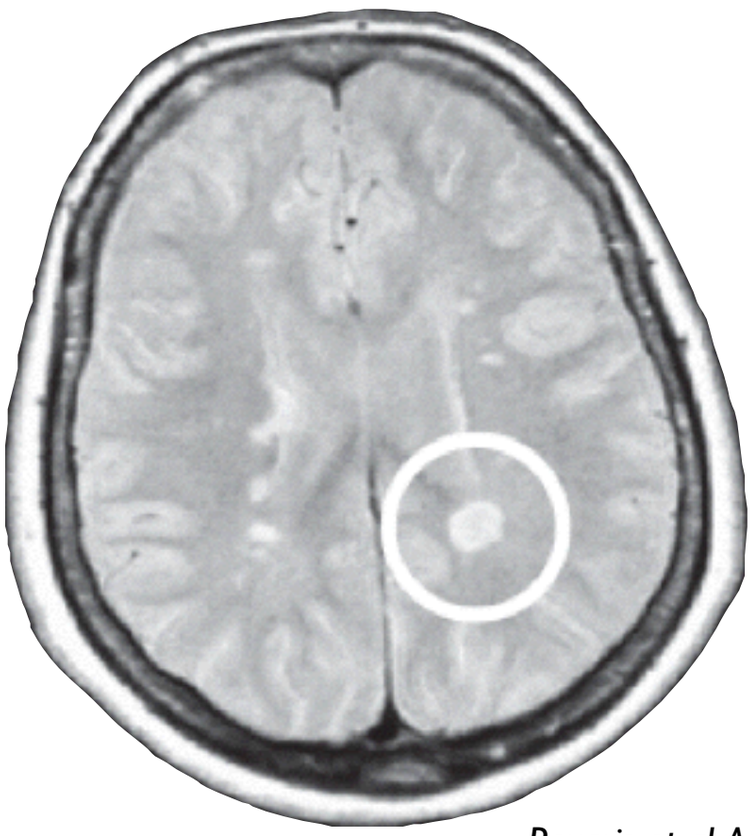
Longitudinal ADC
(λ_1)



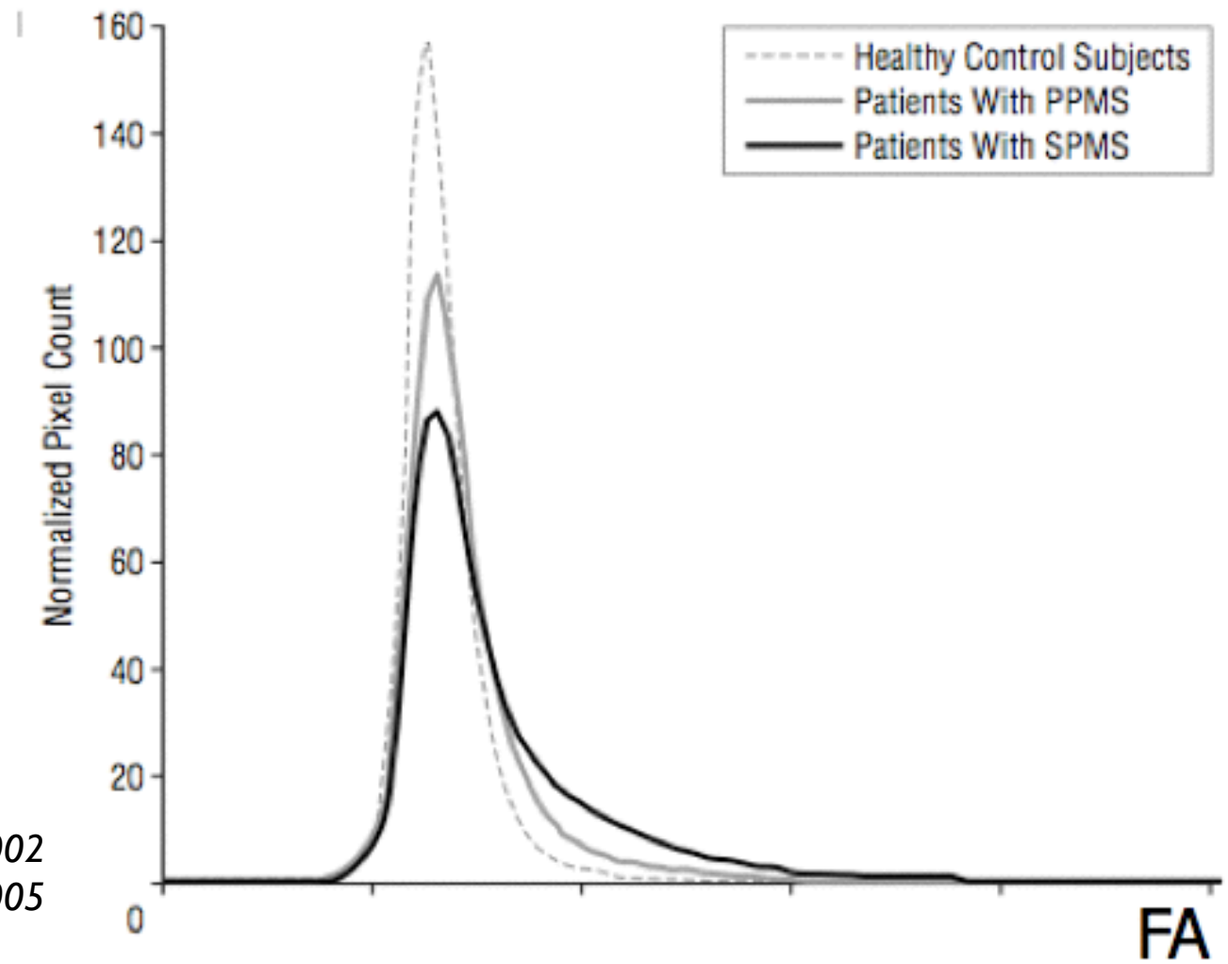
Transverse ADC
$(\lambda_2+\lambda_3)/2$

Quantitative Diffusion Maps

FA decrease/ MD increase has been associated in many studies with tissue breakdown (loss of structure).



Rovaris et al, Arch Neurol 2002
Gallo et al, Arch Neurol 2005



Fractional Anisotropy changes in MS normal appearing white matter

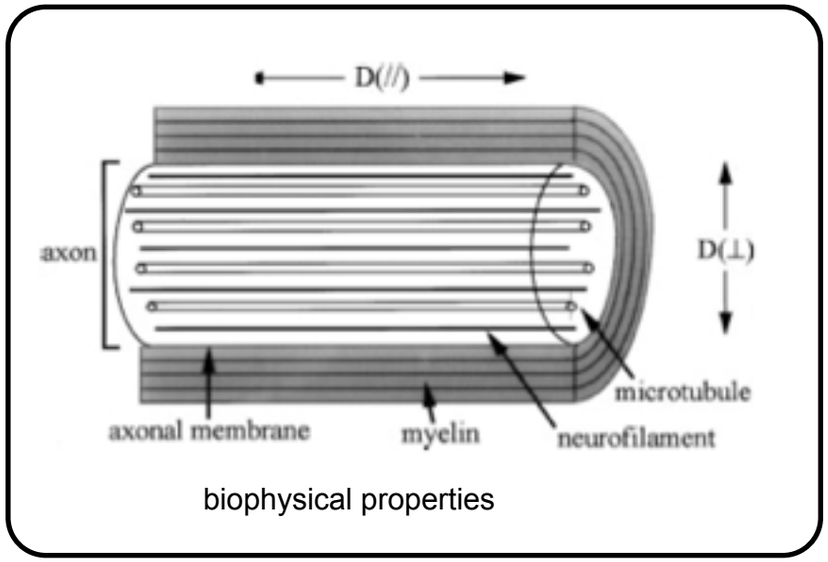
Quantitative Diffusion Maps

FA decrease in WM can be caused:

a) Decrease of longitudinal ADC.
Axonal breakdown?

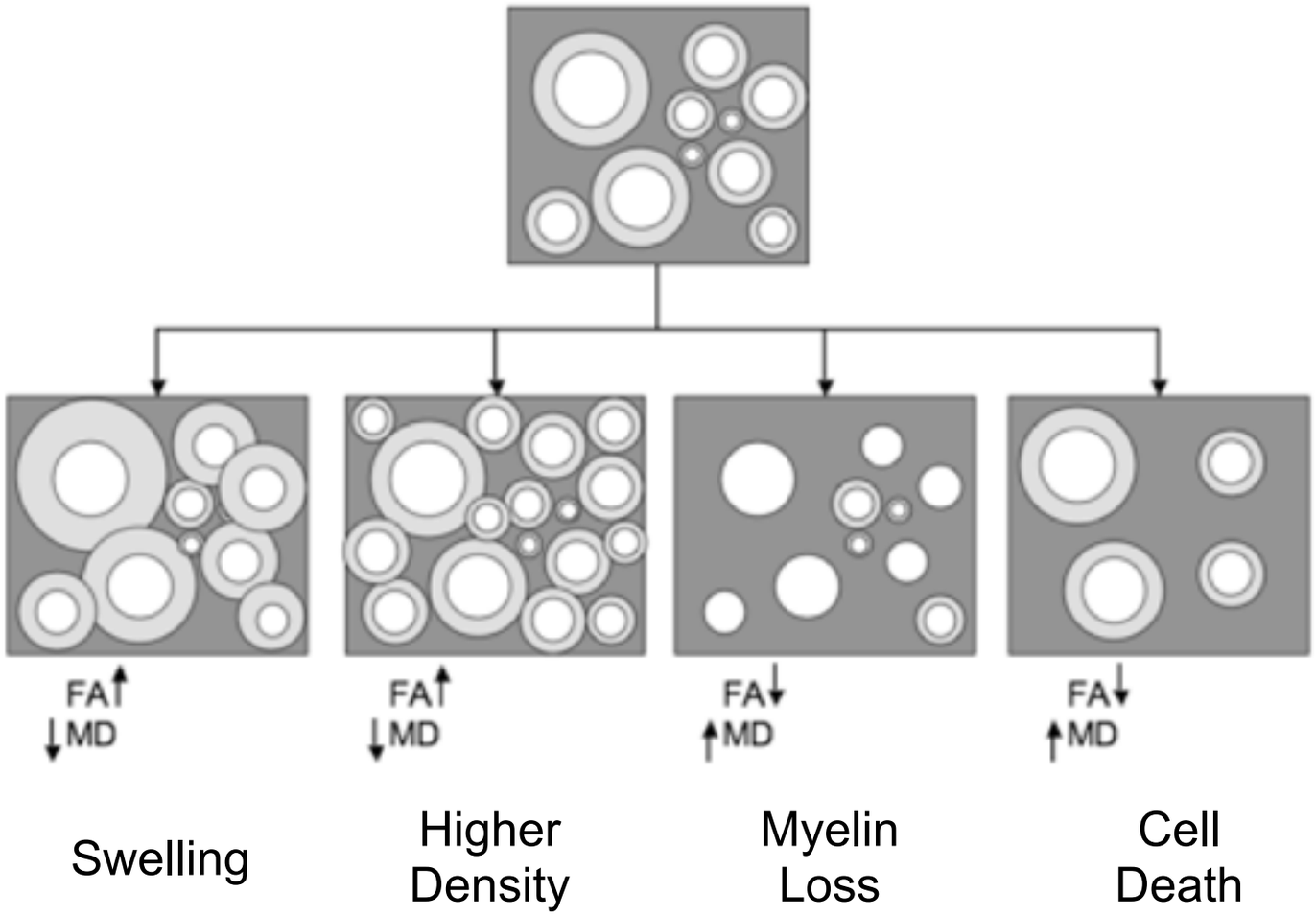
b) Increase of transverse ADC.
Myelin breakdown?

But do not overinterpret your results!
Always keep in mind that the DTI model is an oversimplification of reality!

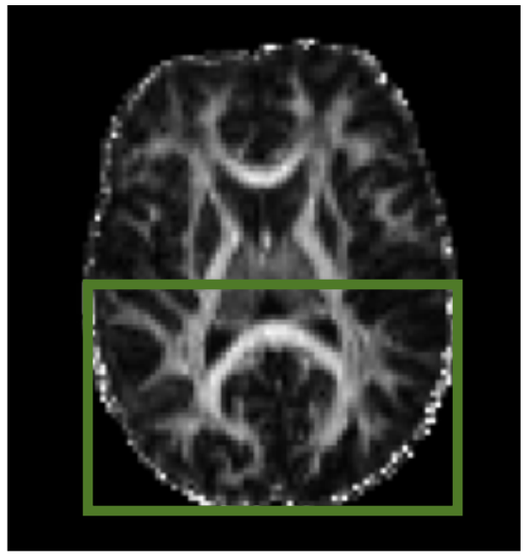


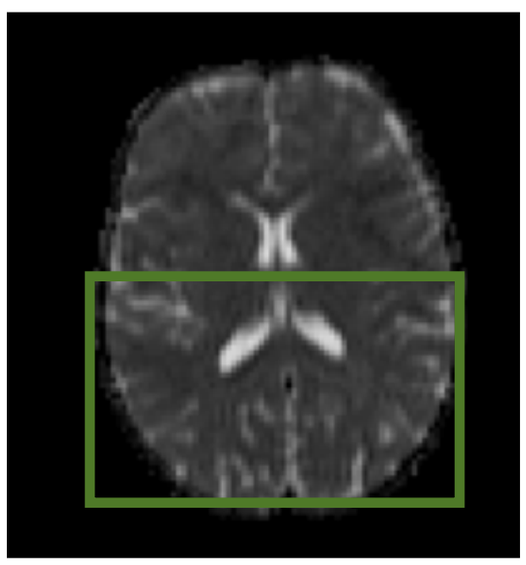
Quantitative Diffusion Maps

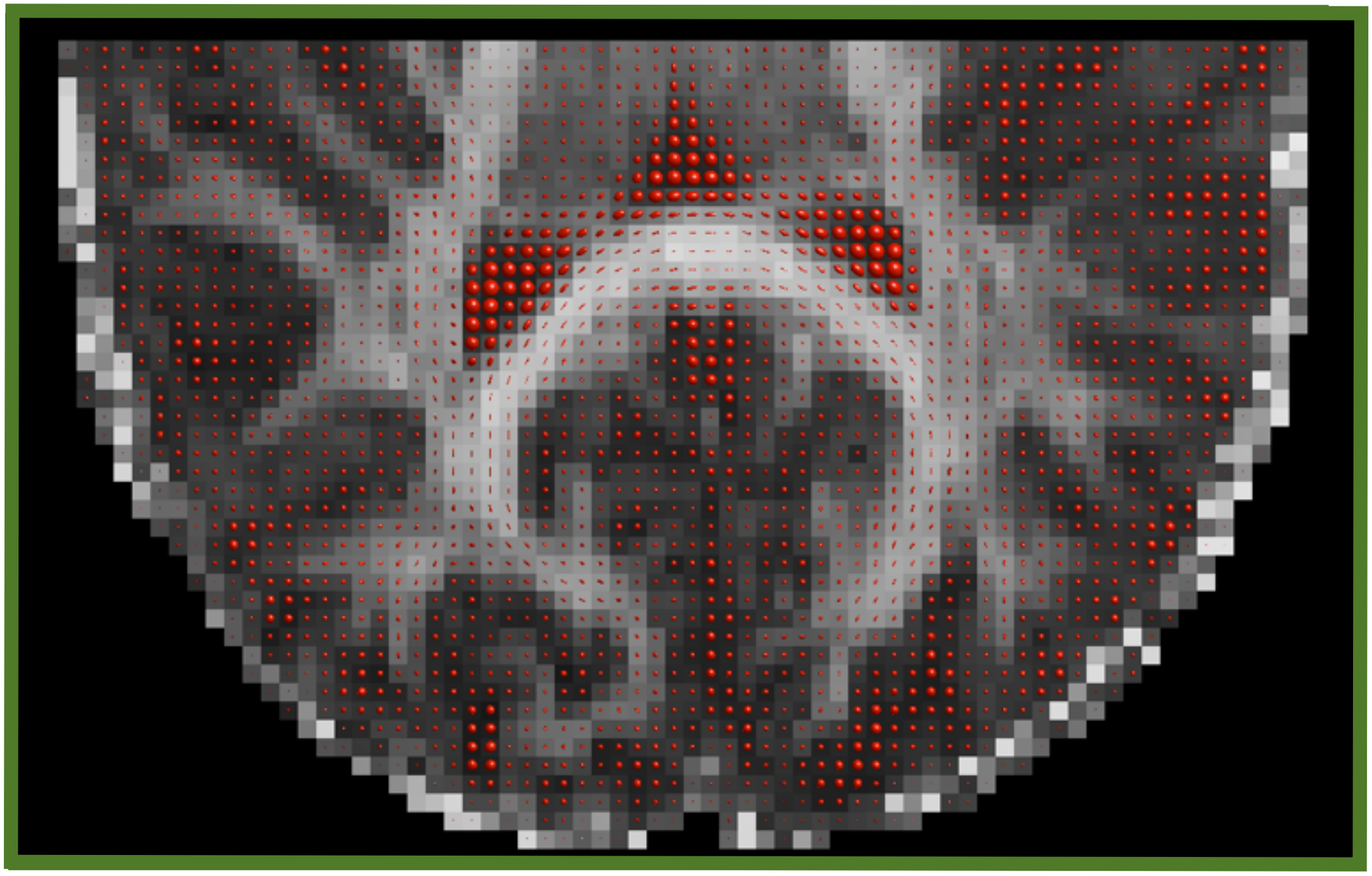
But do not overinterpret your results! Always keep in mind that the DTI model is an oversimplification of reality! Different configurations can have same effect on FA, MD!



Diffusion Tensor Ellipsoids

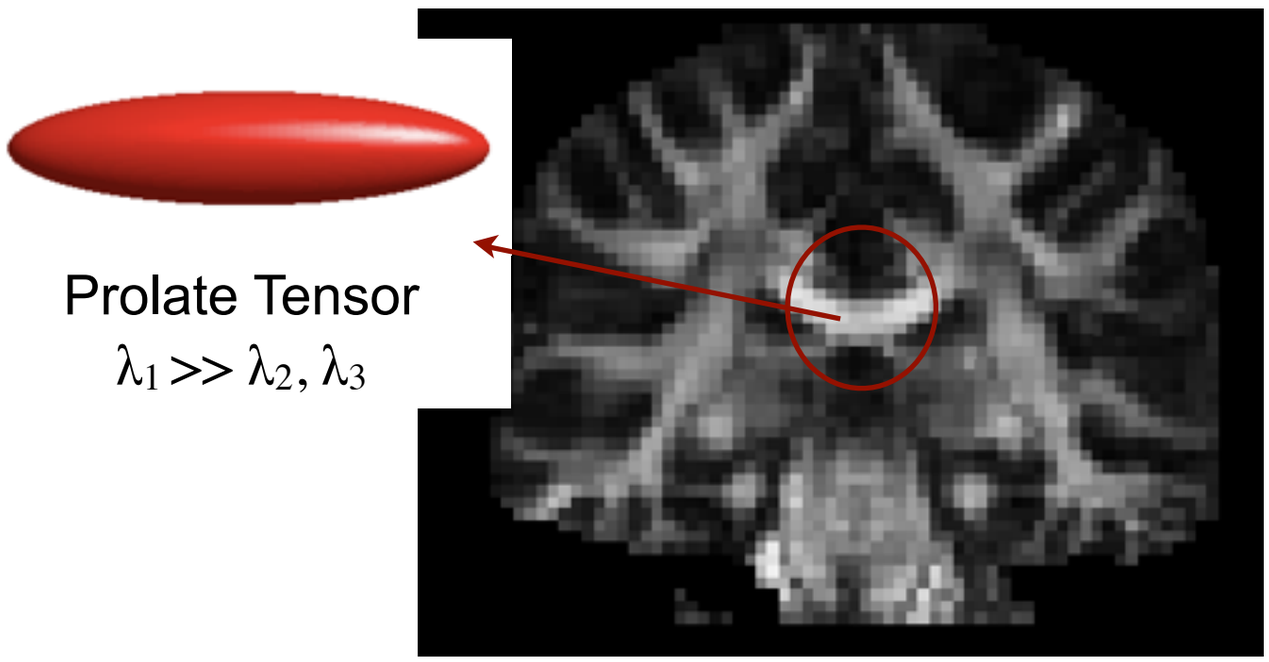






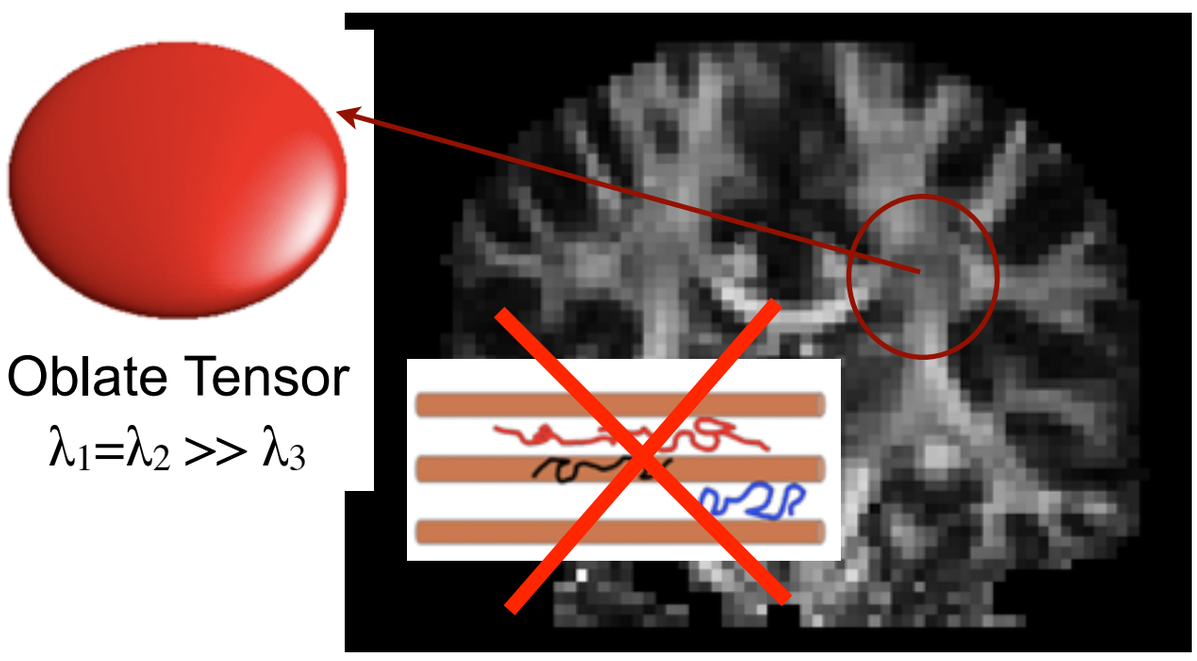
Tensor and FA in Crossing Regions

- In voxels containing two crossing bundles, FA is low and the tensor ellipsoid is pancake-shaped (oblate, planar tensor).



Prolate Tensor

$\lambda_1 >> \lambda_2, \lambda_3$



Oblate Tensor

$\lambda_1 = \lambda_2 >> \lambda_3$

Consequences:

- PDD not necessarily = direction of fibres
- FA changes difficult to interpret

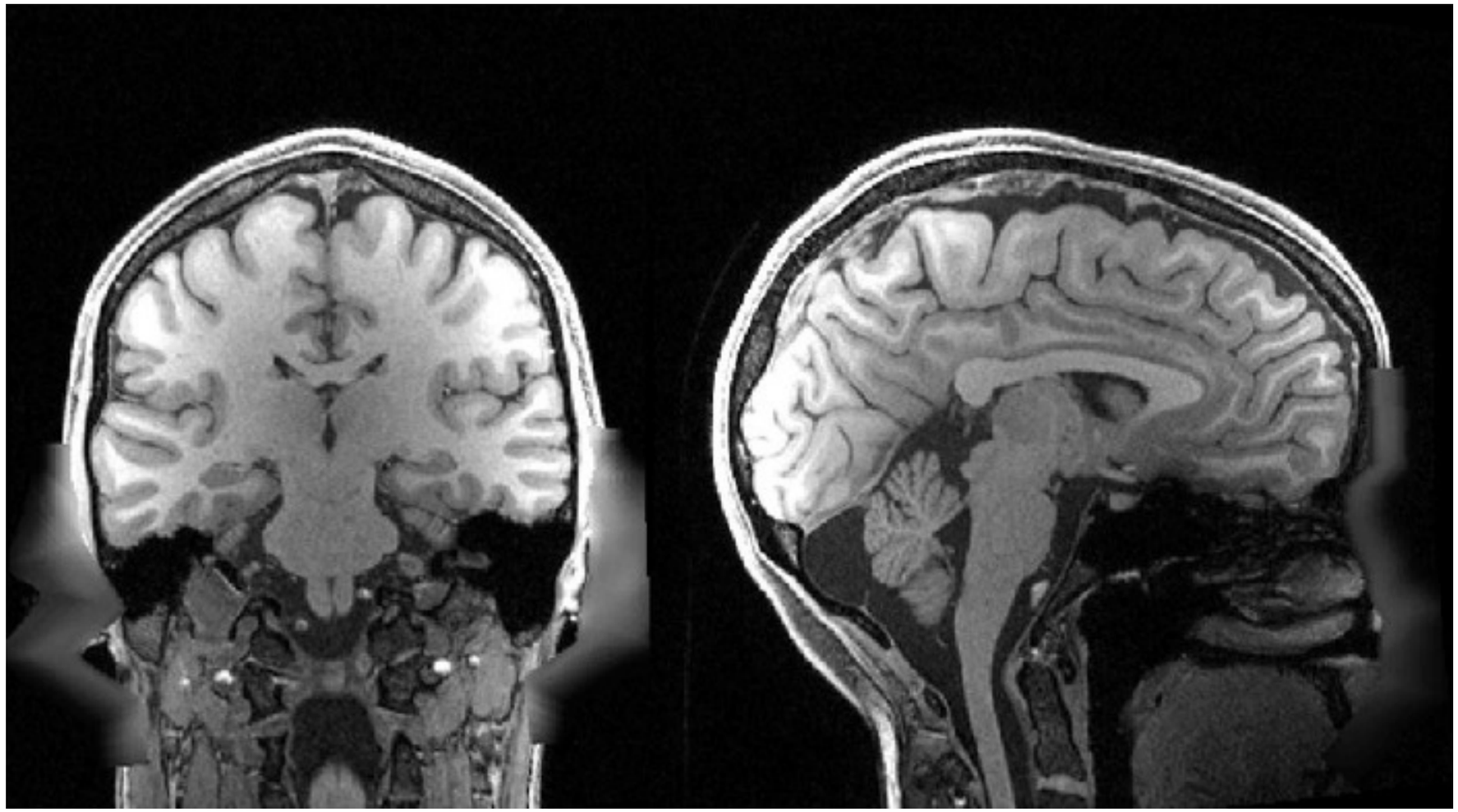
“Problems” with diffusion data

- Gradient non-linearities
- Distortions
  - Susceptibility-induced (topup)
  - Eddy current-induced
- Subject movement
  - Gross movement
  - Signal dropout

} (eddy)

Gradient non-linearities

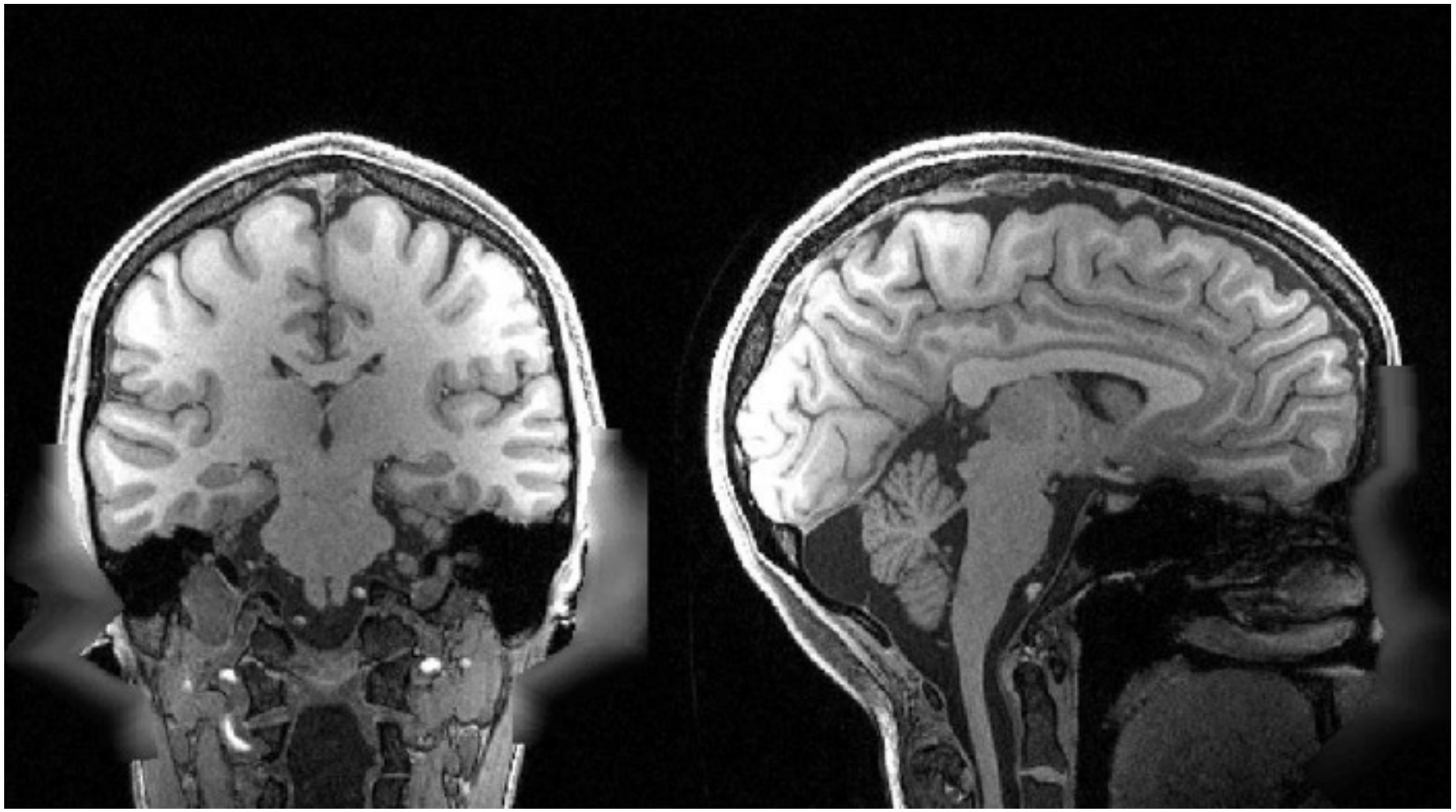
Gradients are not perfectly linear



If we take a subject like this and scan with “typical” gradients.

Gradient non-linearities

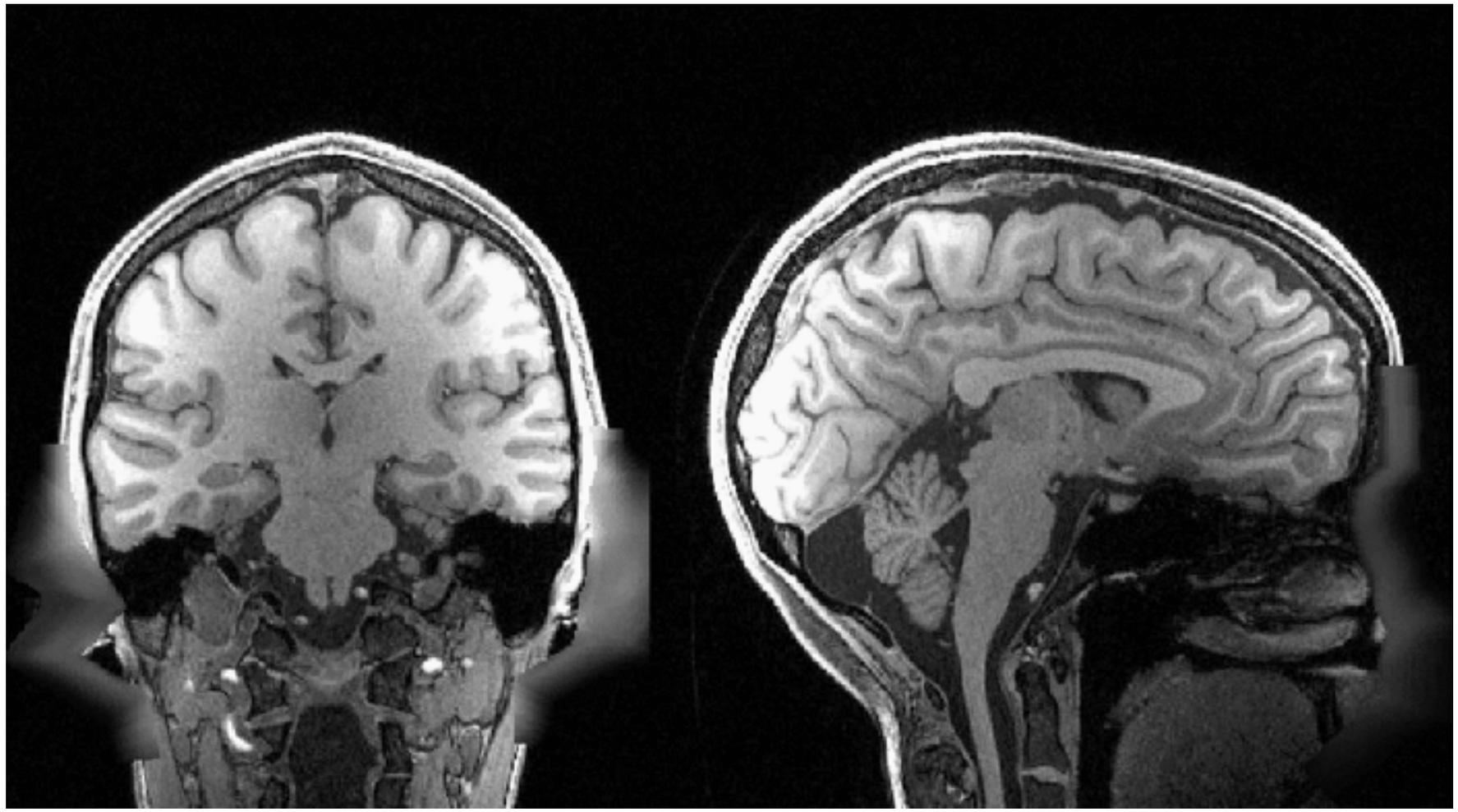
Gradients are not perfectly linear



He/she will look like this

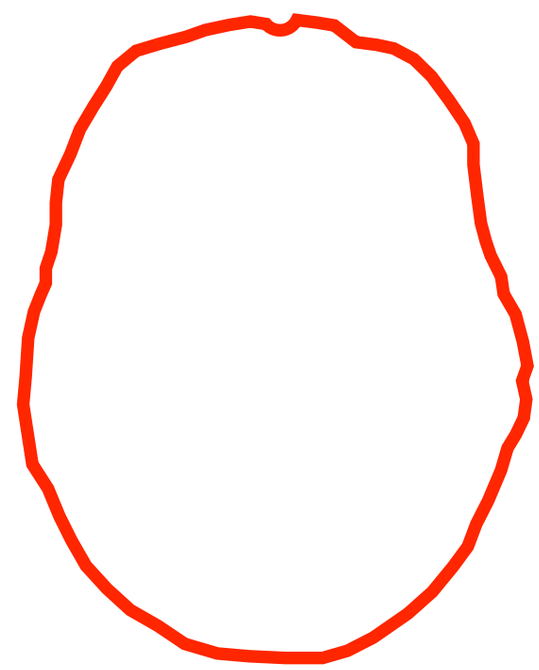
Gradient non-linearities

This is (mostly) fixed by your scanner

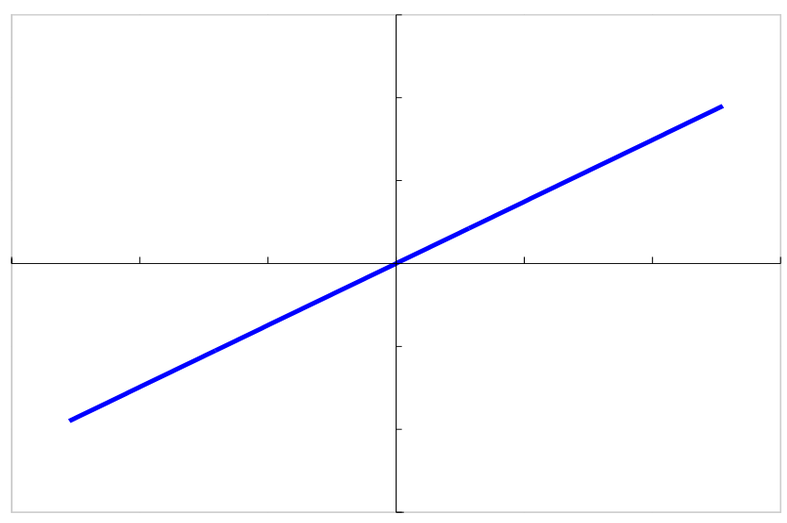


BUT, what about the diffusion encoding?

Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

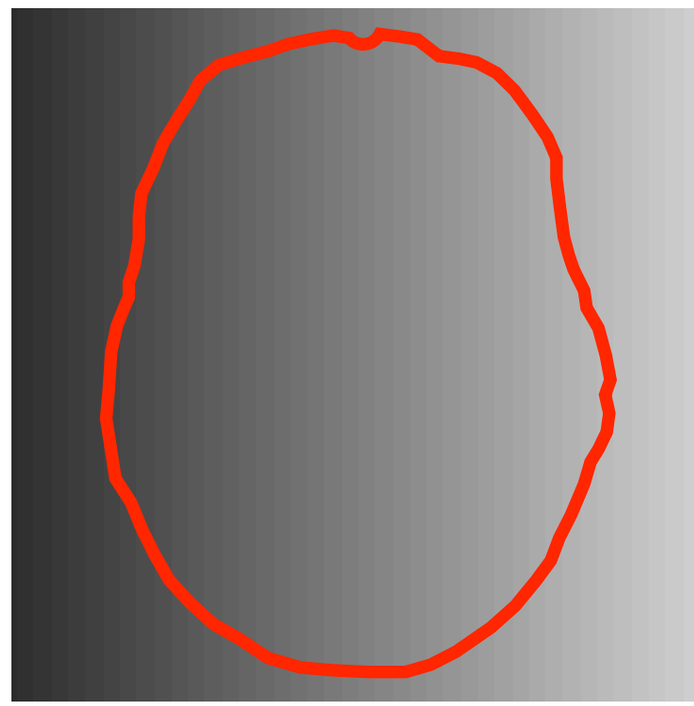


x-gradient

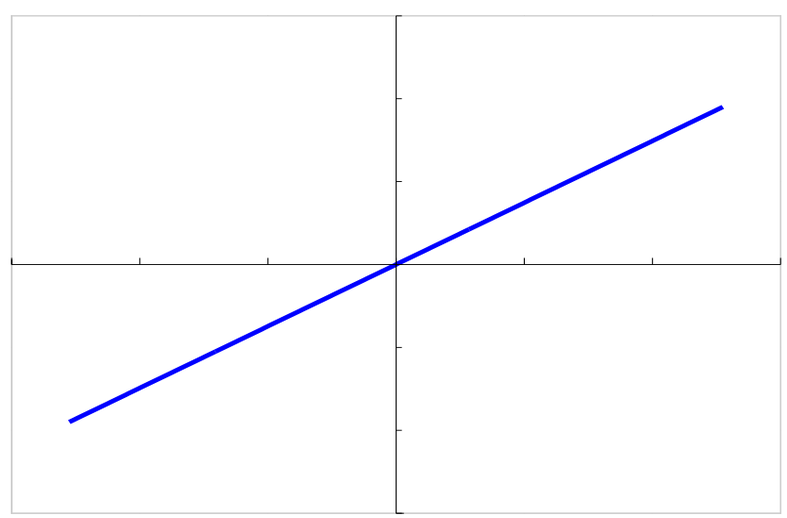


Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

field

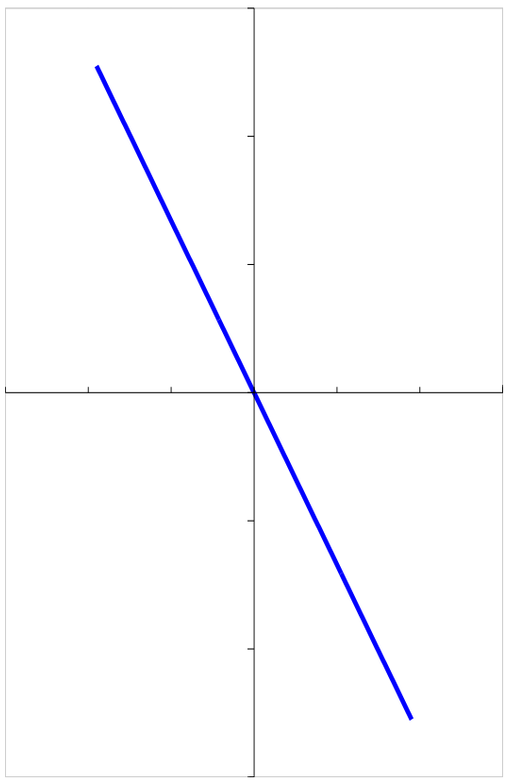


x-gradient

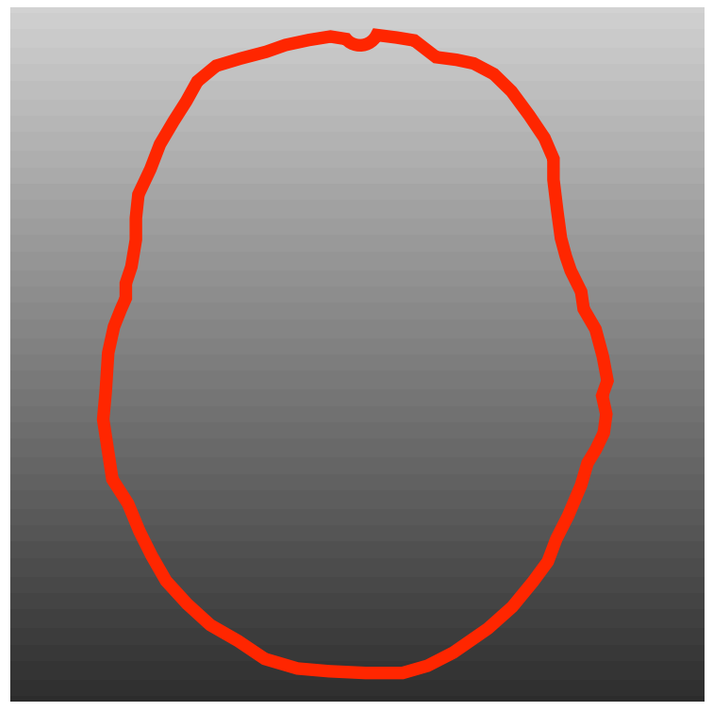


Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

y-gradient



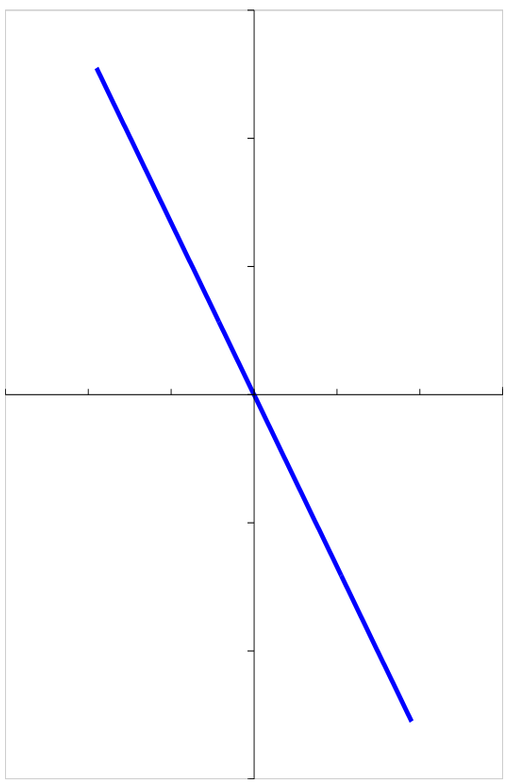
field

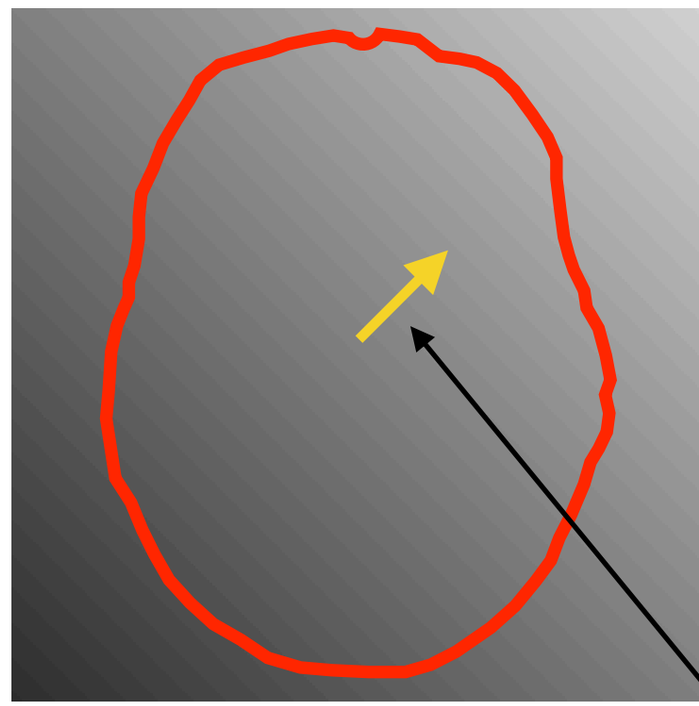


Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

field

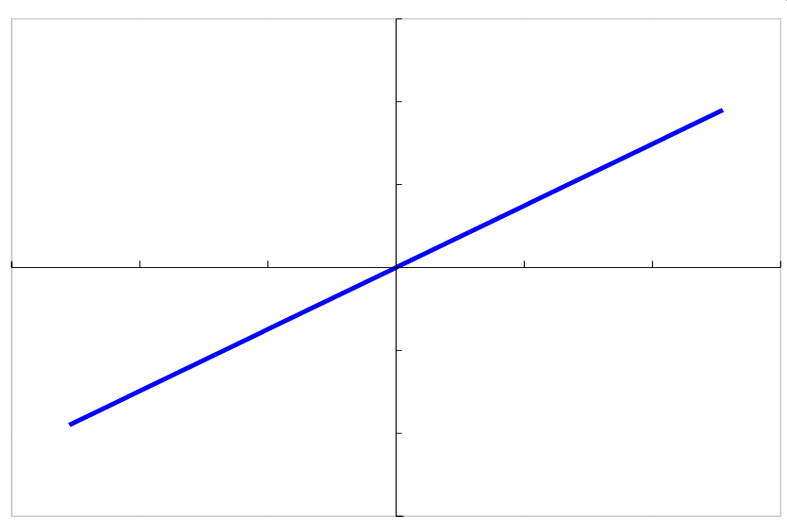
y-gradient





So when we play out a diffusion gradient [.7 .7] we get this

x-gradient

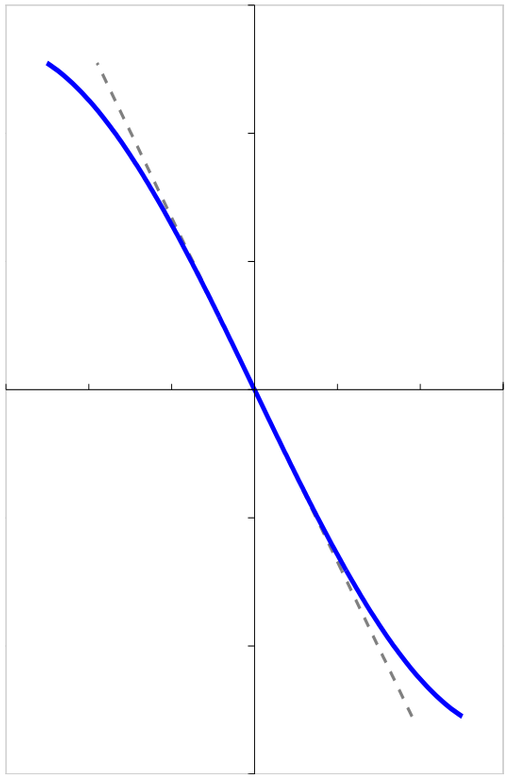


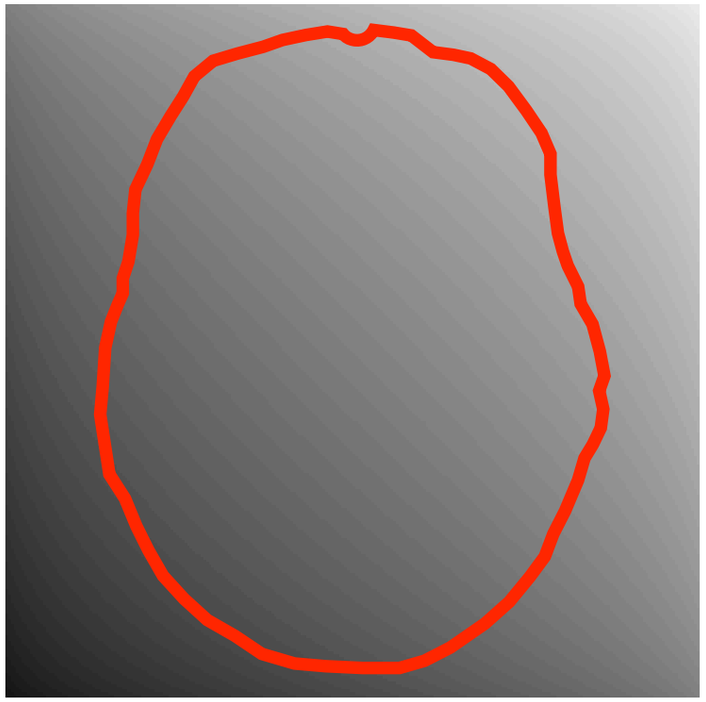
b-vector, same everywhere in the brain

Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

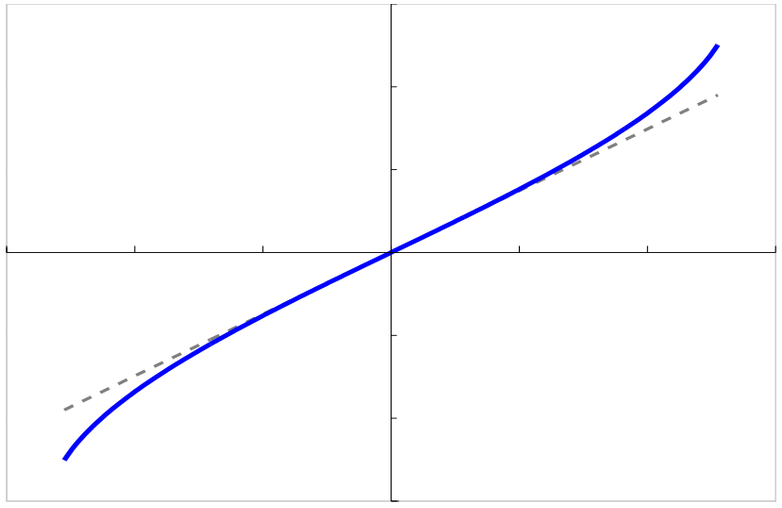
field

y-gradient





x-gradient

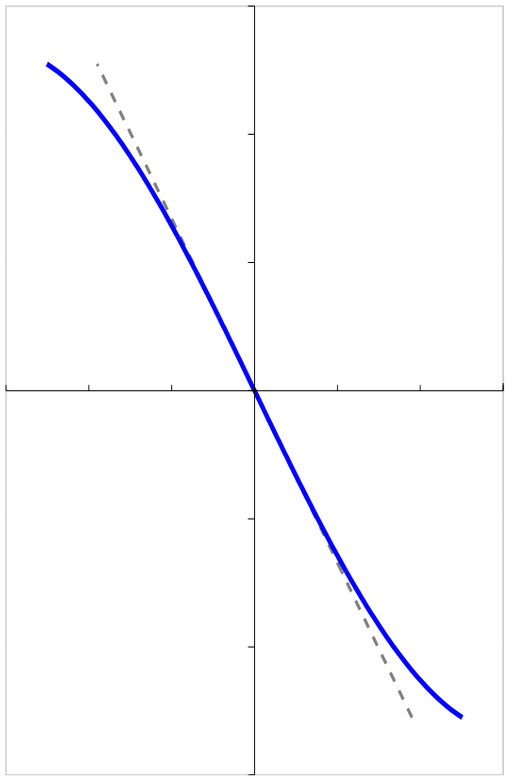


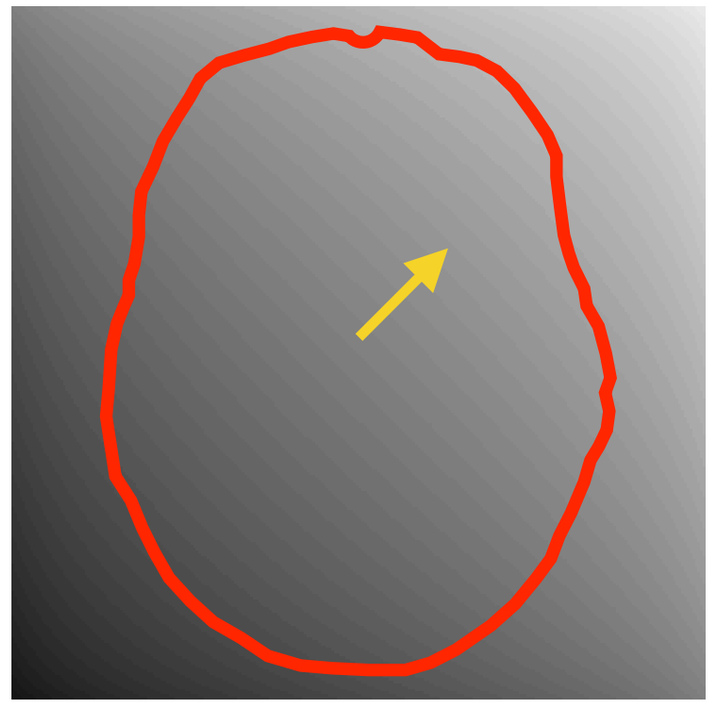
But what if the gradients really look like this?

Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

field

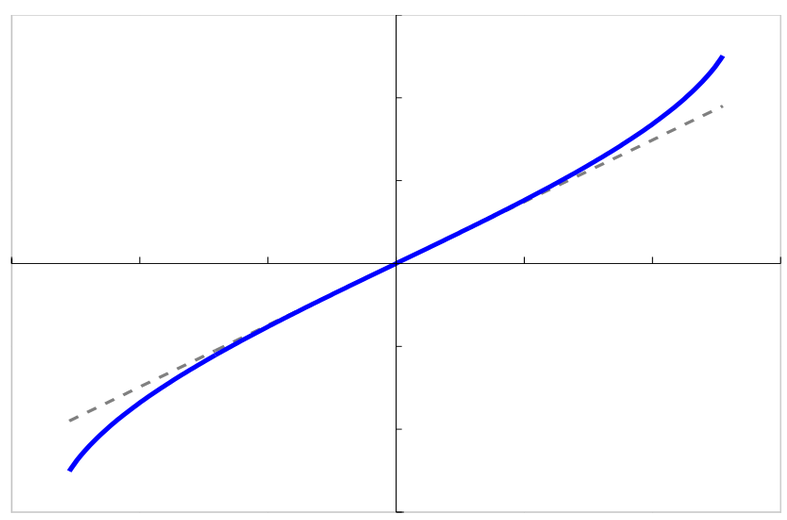
y-gradient





In the centre the b-vector is the same as before

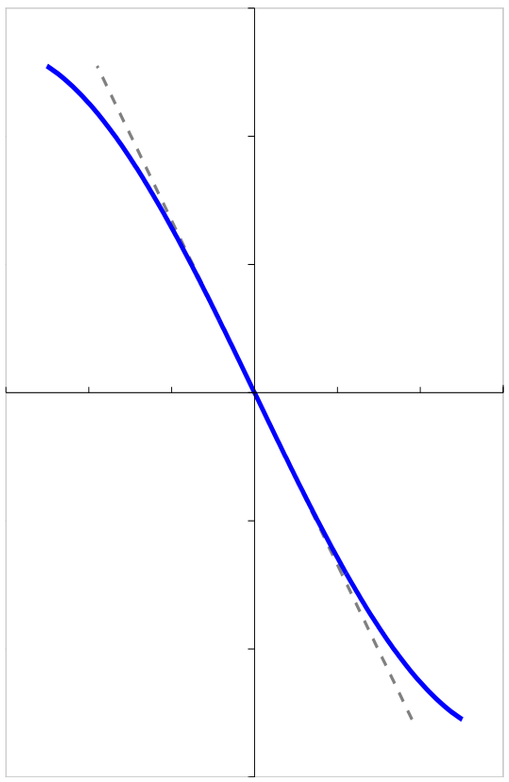
x-gradient



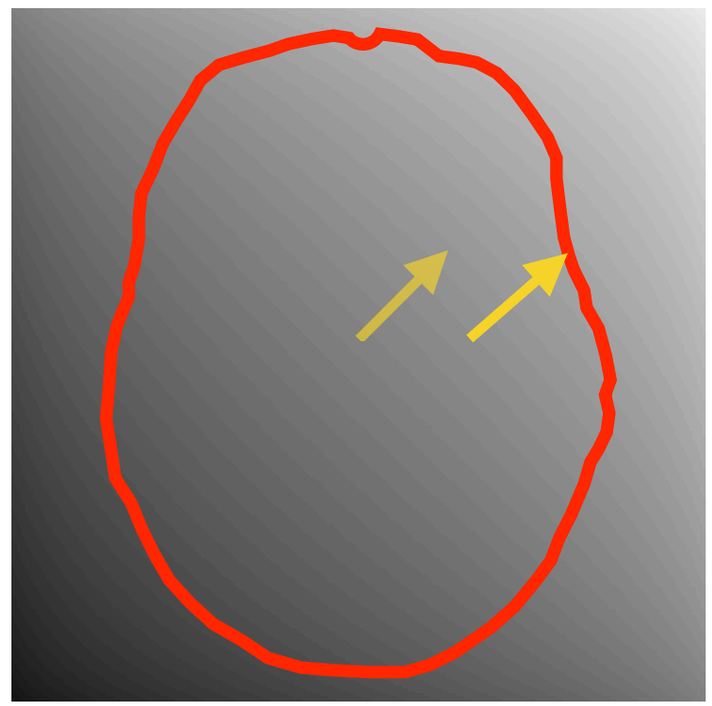
But what if the gradients really look like this?

Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

y-gradient

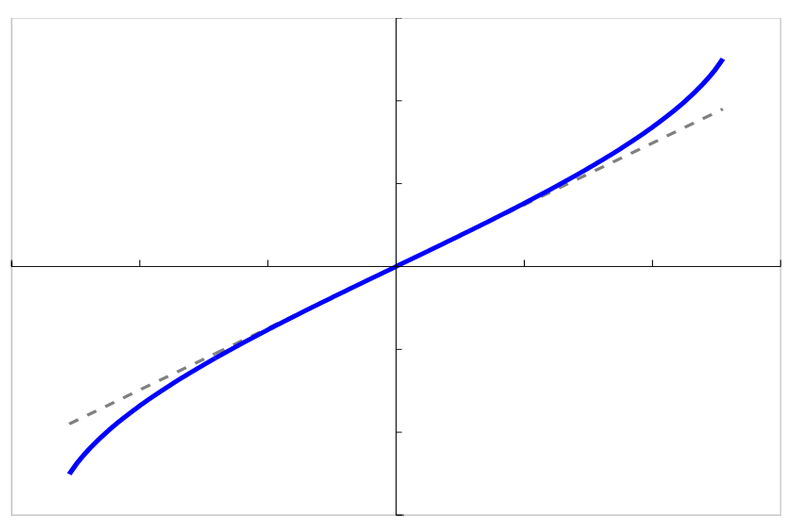


field



But as we move in the x-direction …

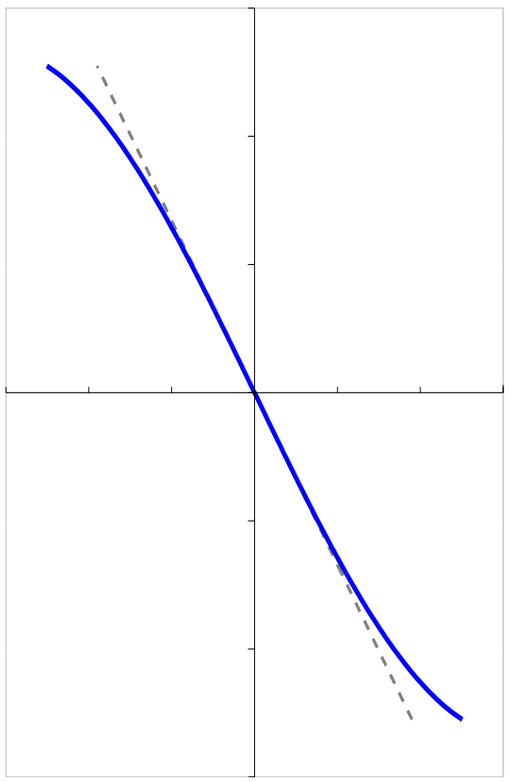
x-gradient



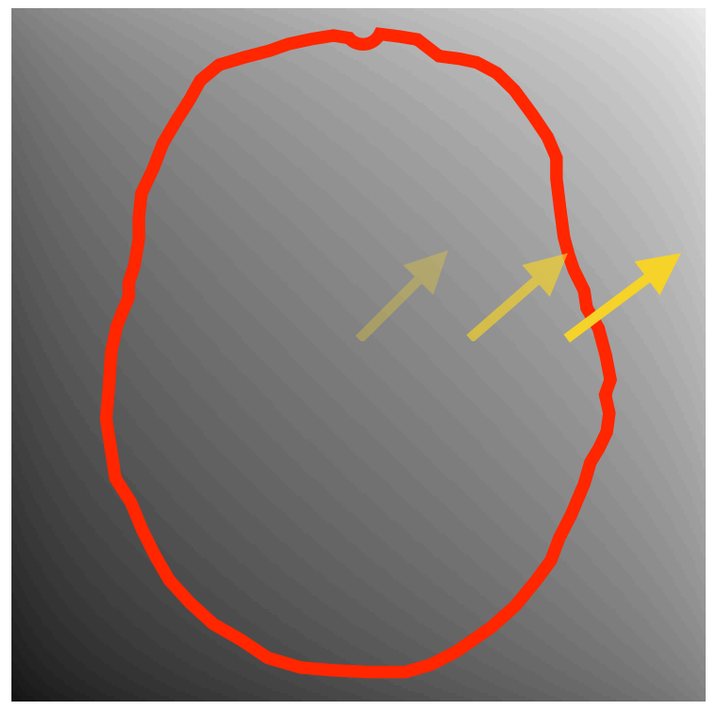
But what if the gradients really look like this?

Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

y-gradient

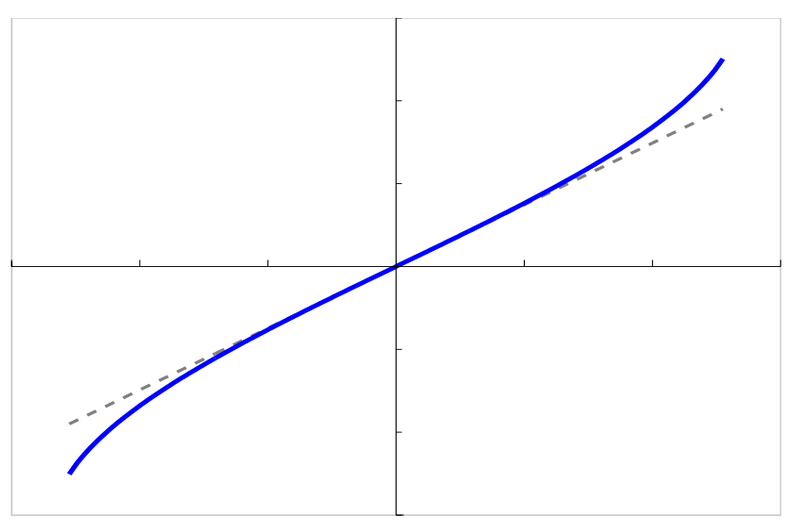


field



But as we move in the x-direction …

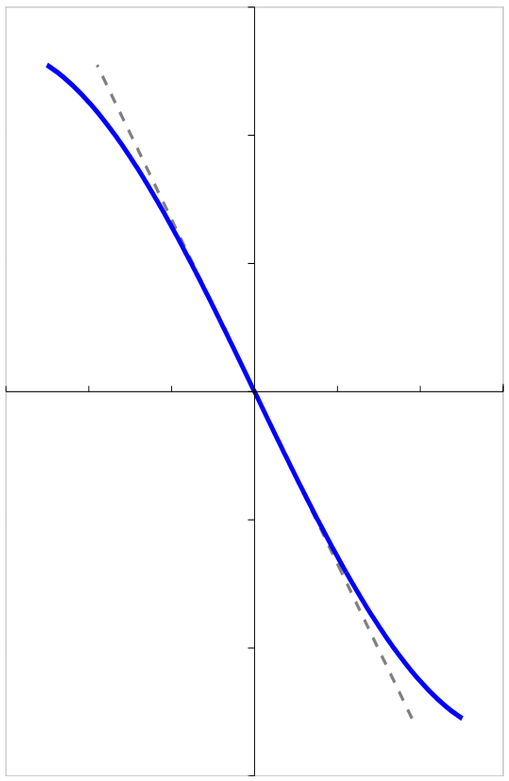
x-gradient



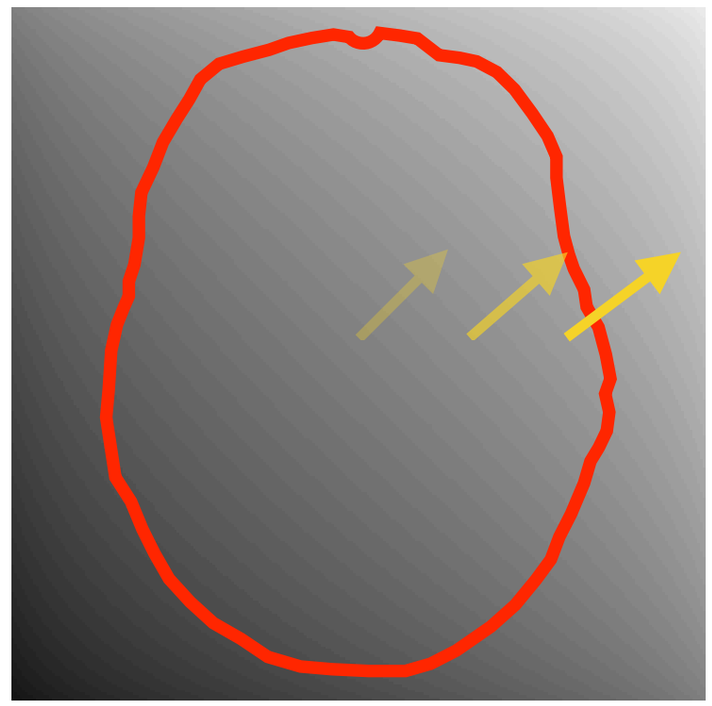
But what if the gradients really look like this?

Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

y-gradient

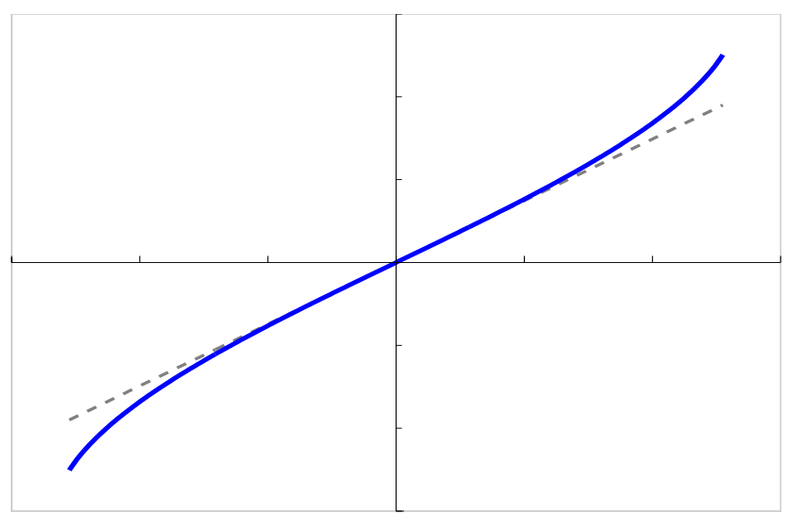


field



It grows (b-value increases) and rotates

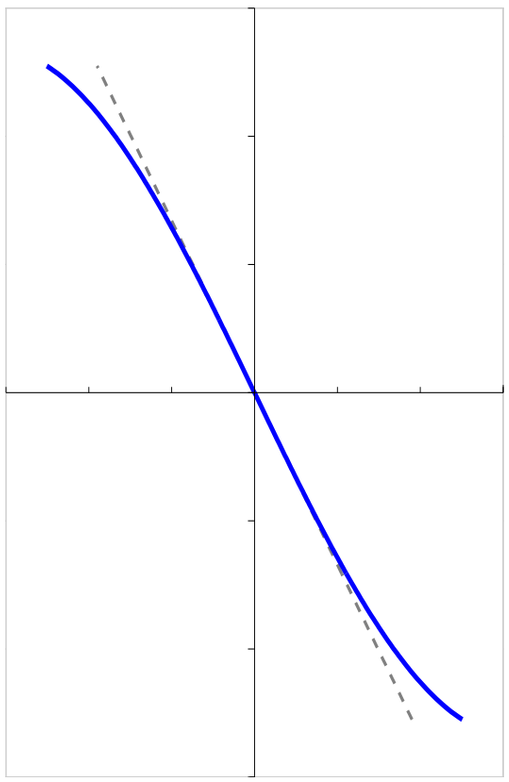
x-gradient



But what if the gradients really look like this?

Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

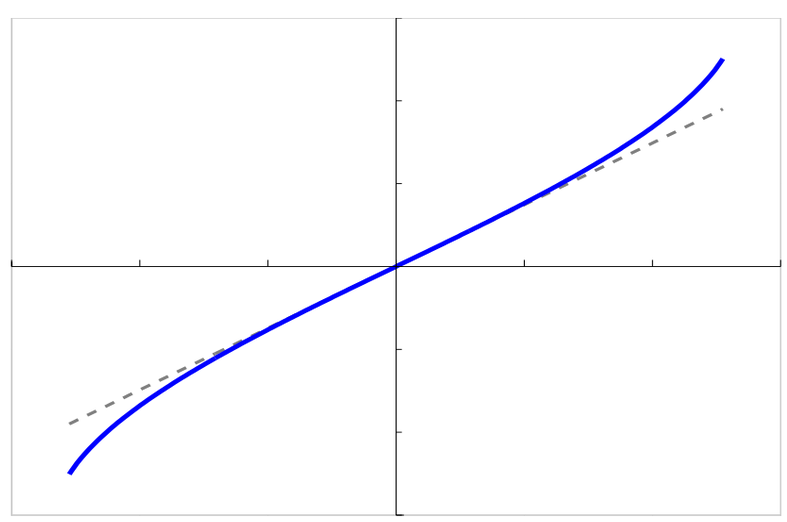
y-gradient



field

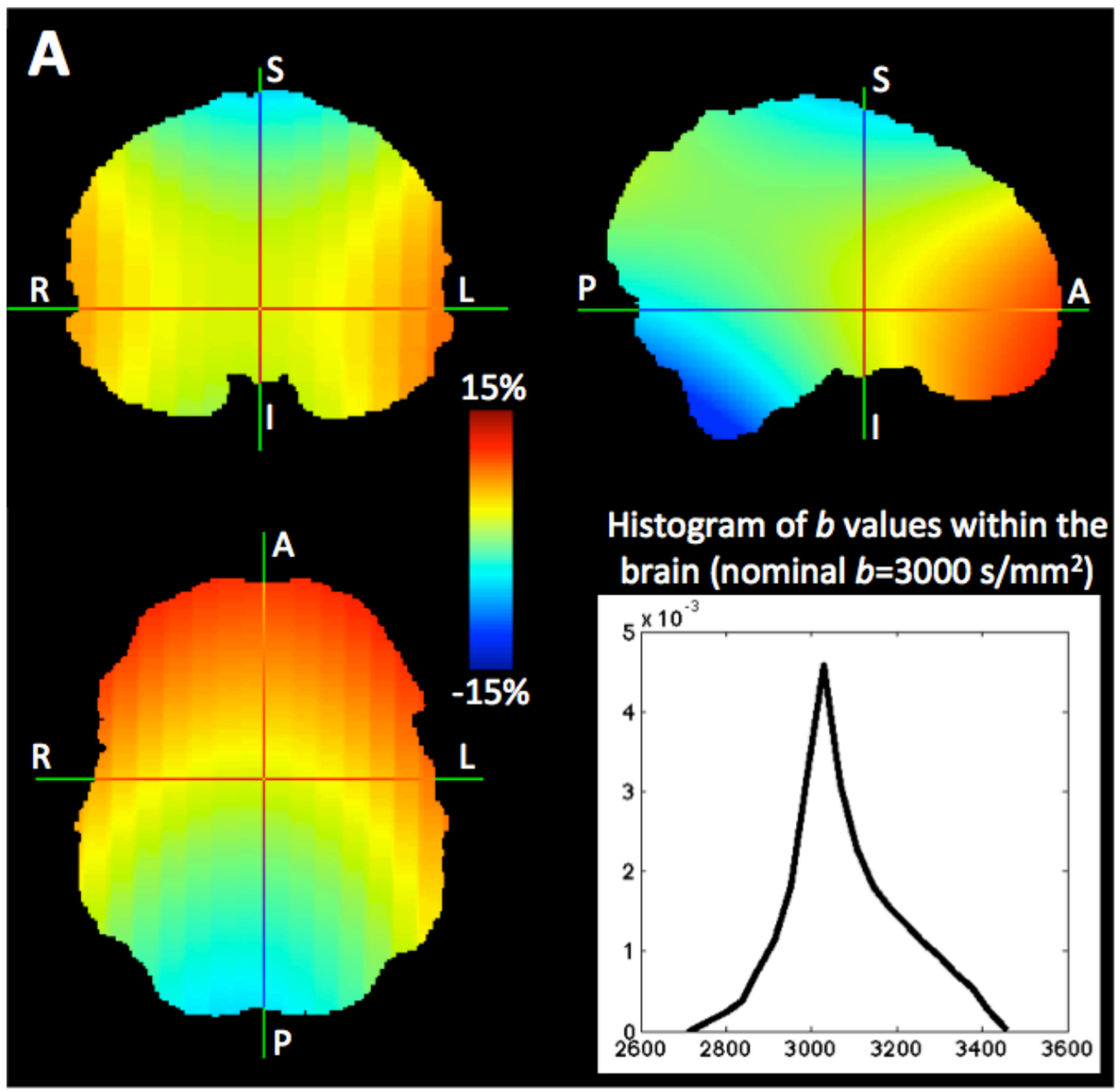
And if we move in the y-direction

x-gradient



But what if the gradients really look like this?

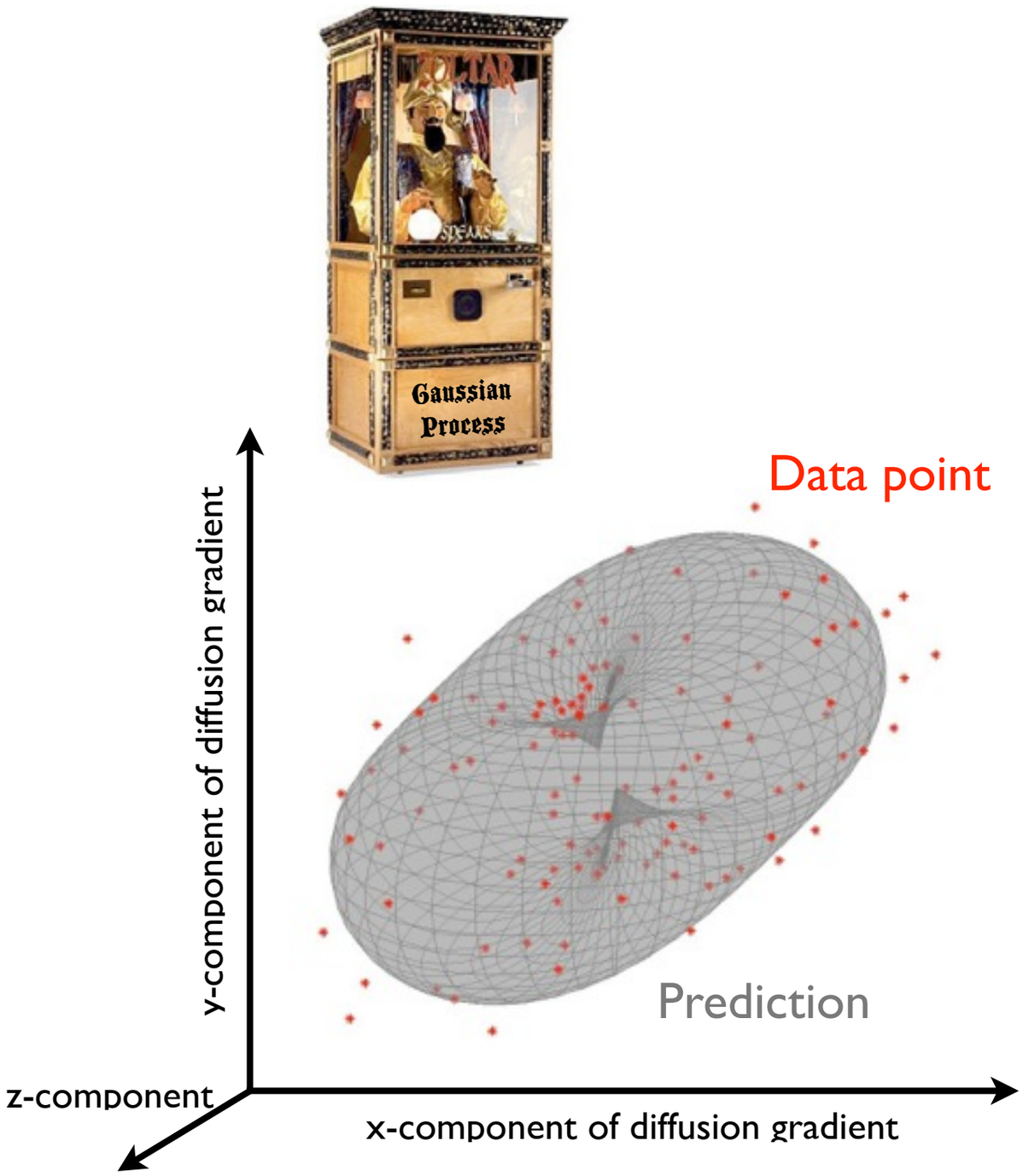
Gradient non-linearities and diffusion encoding (Bammer et al., 2003)

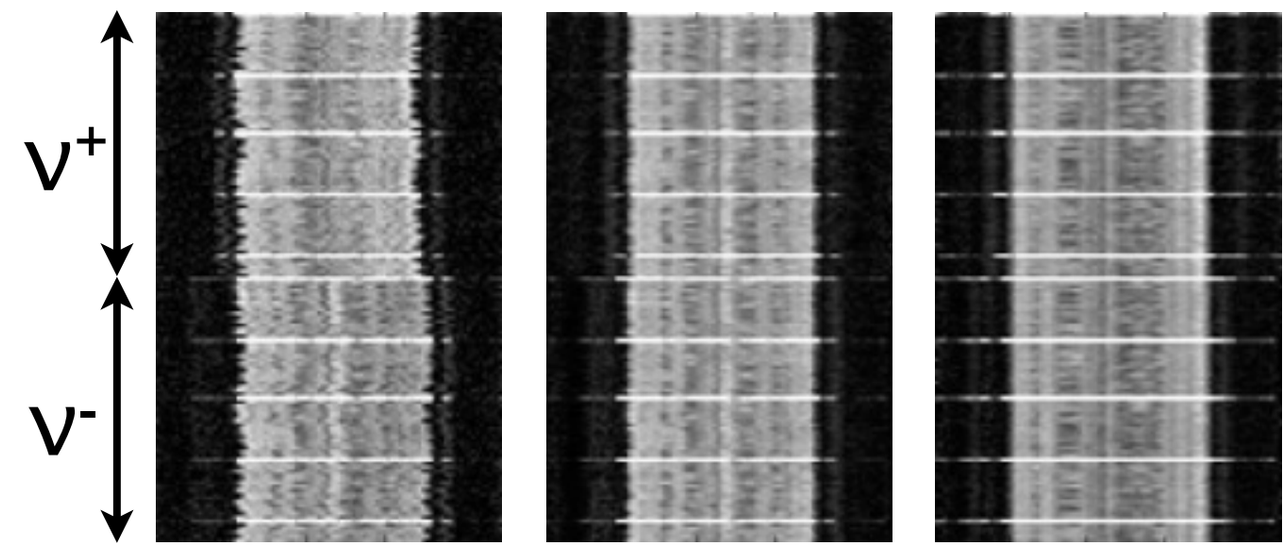


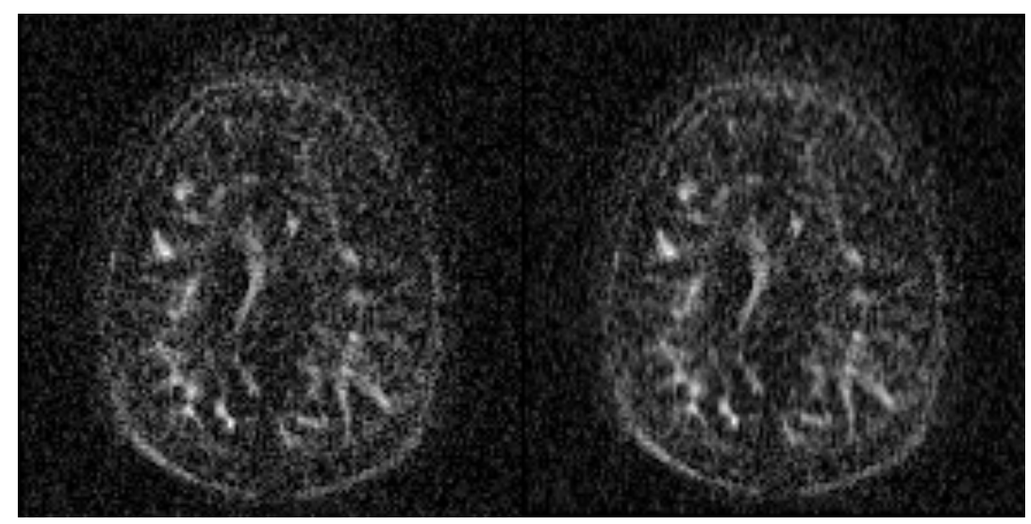
HCP data comes with gradient specifications and code to correct for this.

To do this at home you need gradient specifications from your vendor

eddy and topup - tools for processing of diffusion data



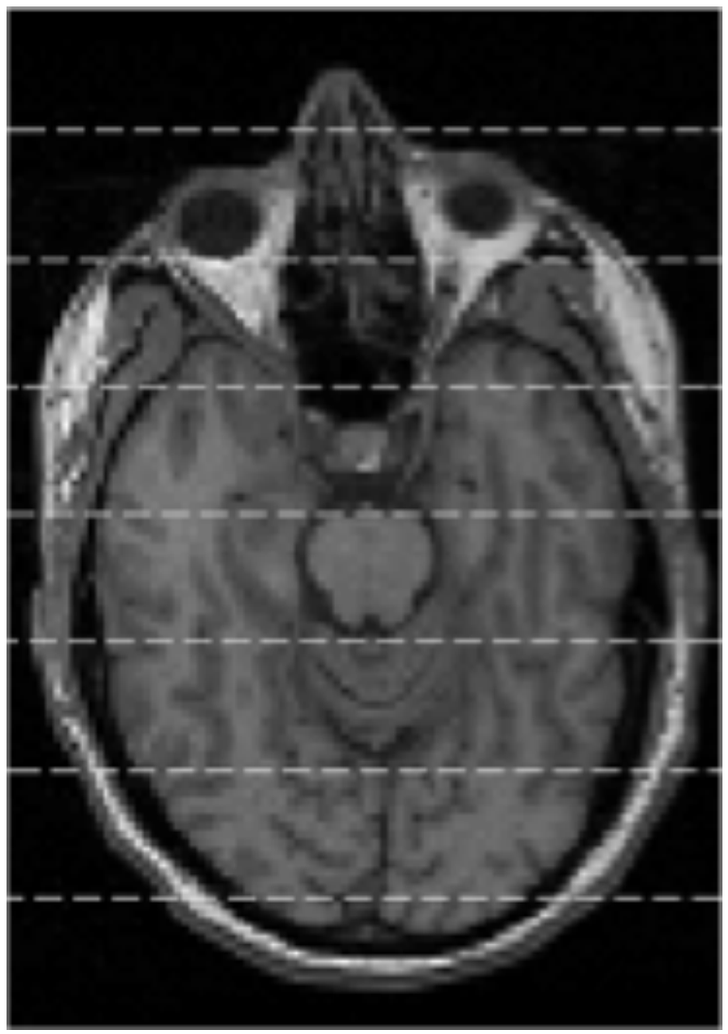


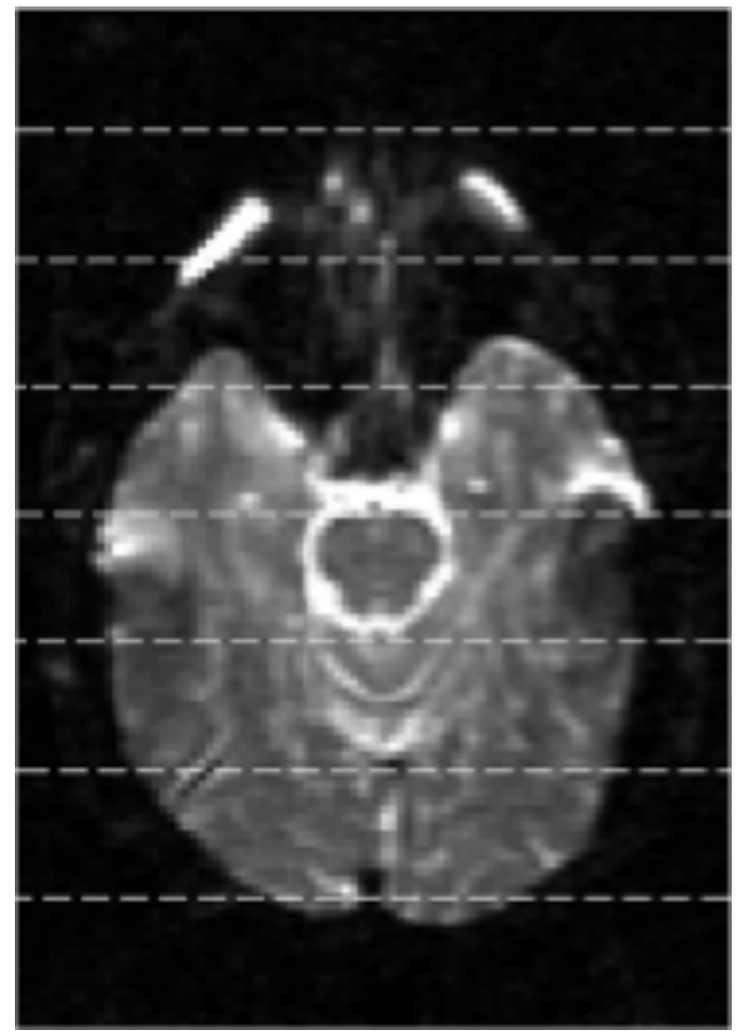


Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

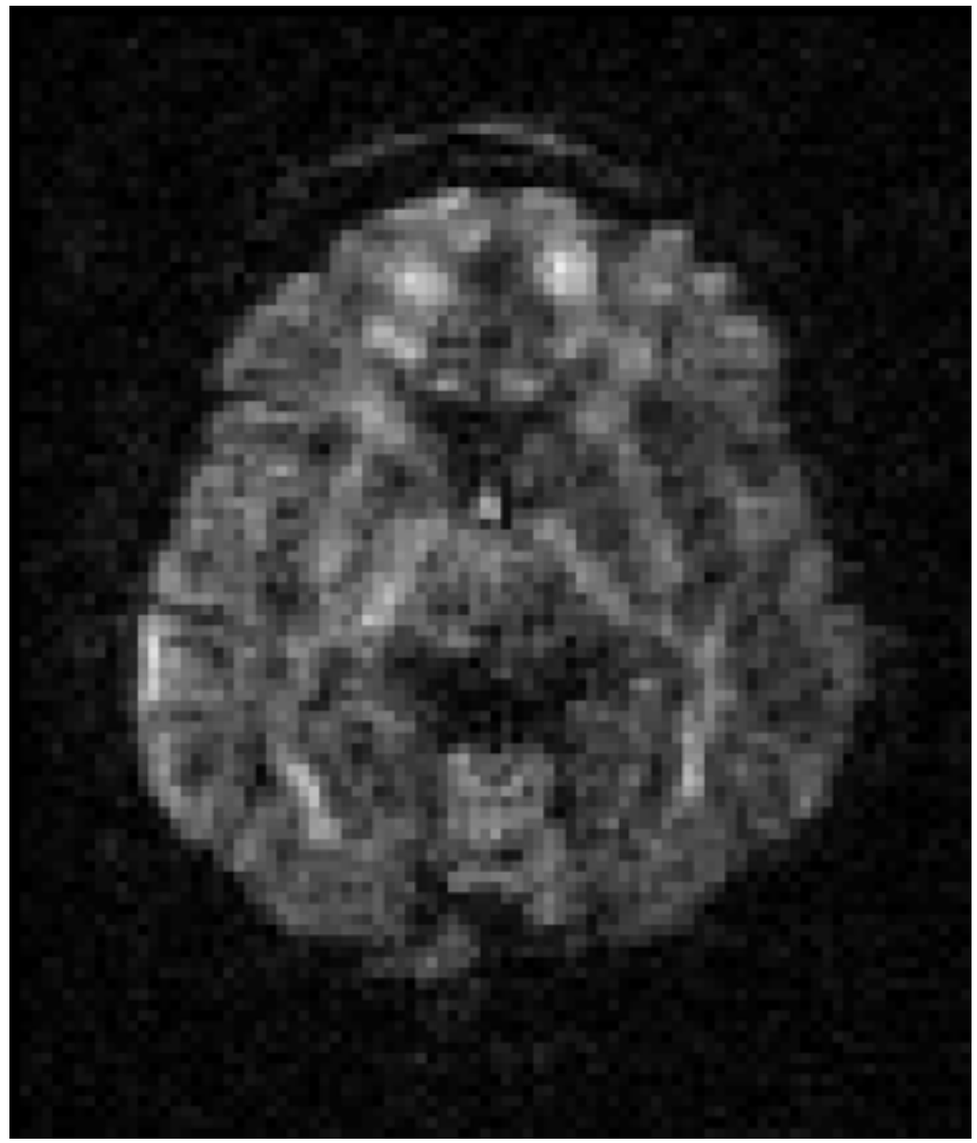
What is the problem with diffusion data?





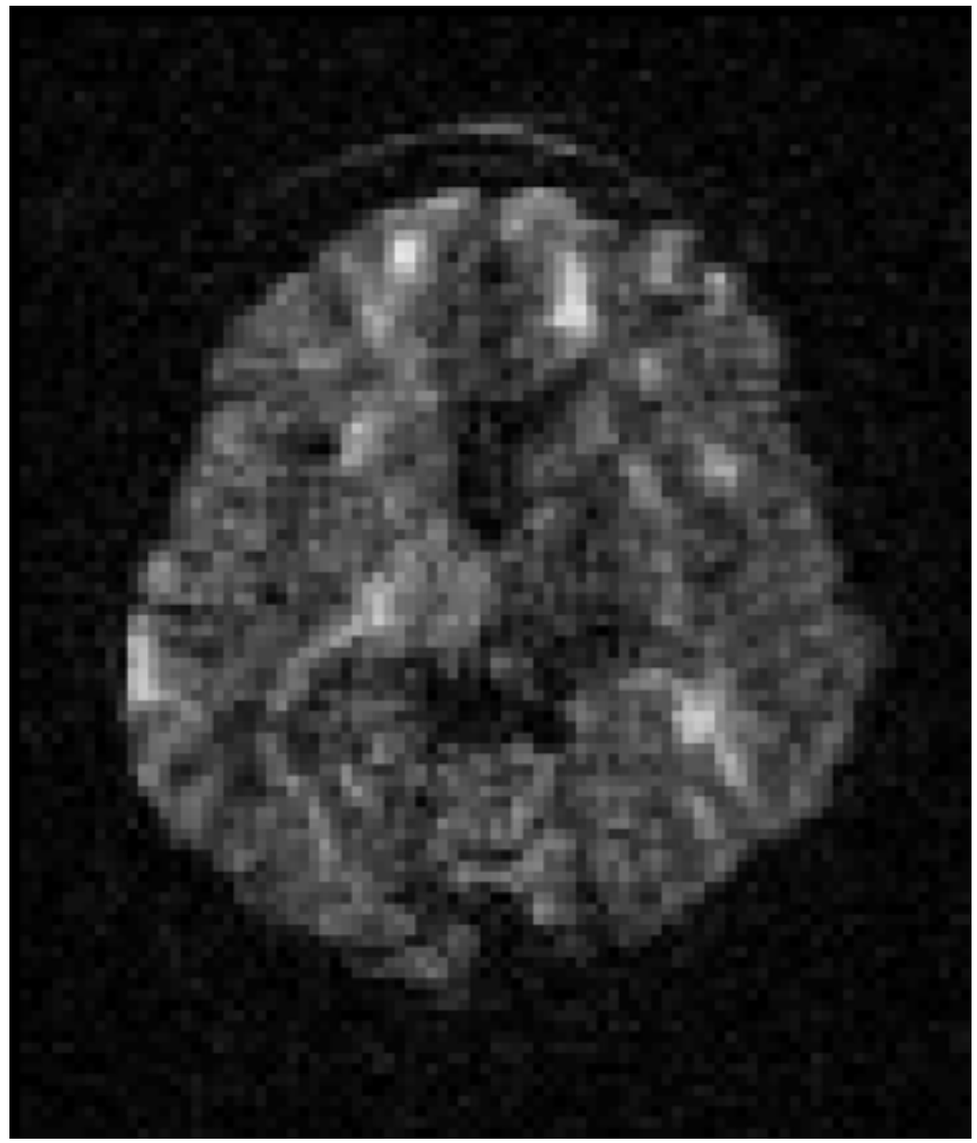
Well, it isn’t very anatomically faithful

What is the problem with diffusion data?



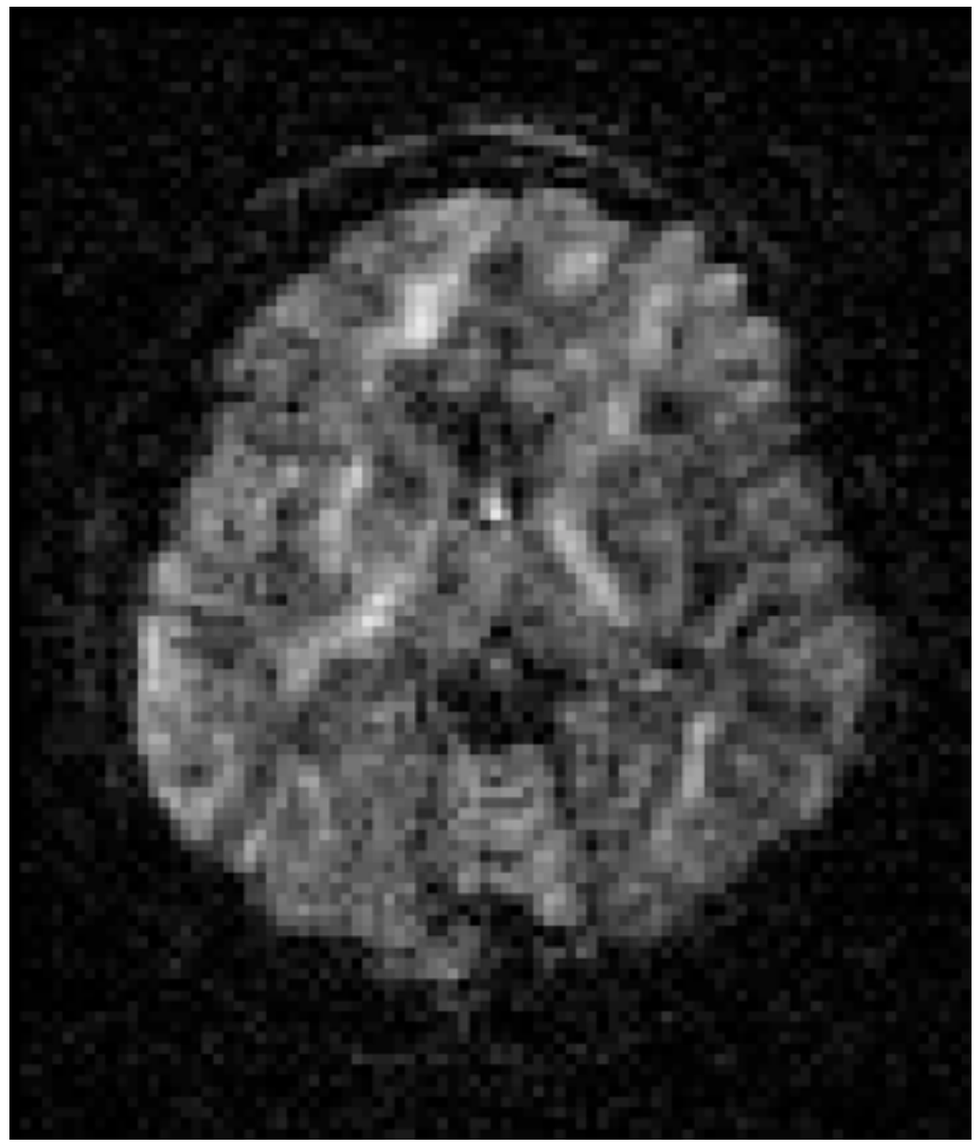
In fact, it isn’t even internally consistent

What is the problem with diffusion data?



In fact, it isn’t even internally consistent

What is the problem with diffusion data?



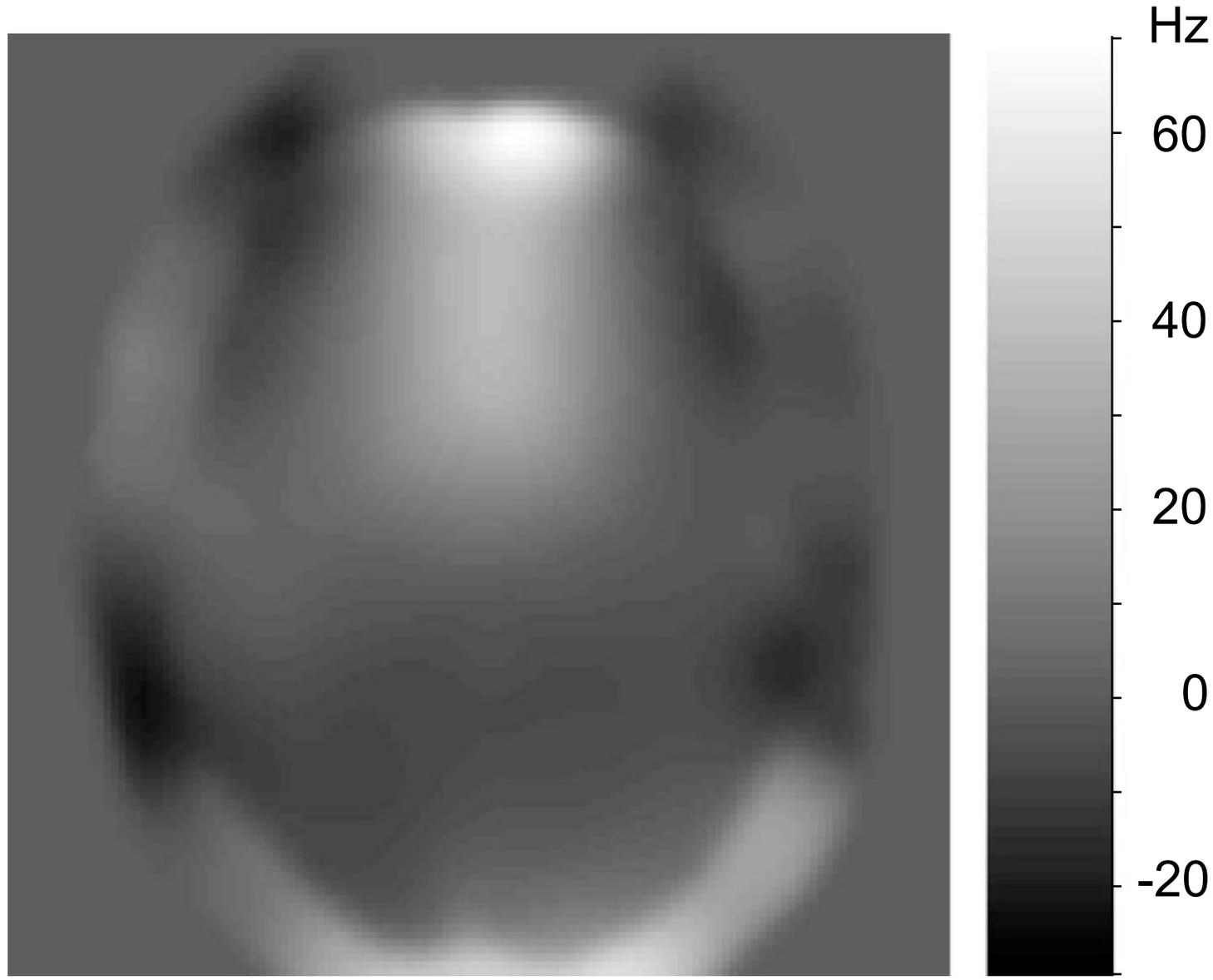
In fact, it isn’t even internally consistent

Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

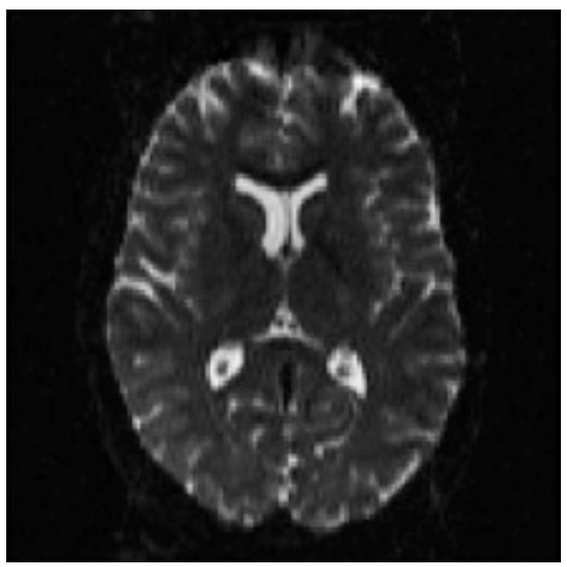
Off-resonance field ⇒ Distortions

An “off-resonance” field is a map of the difference between what we think the field is and what it really is.

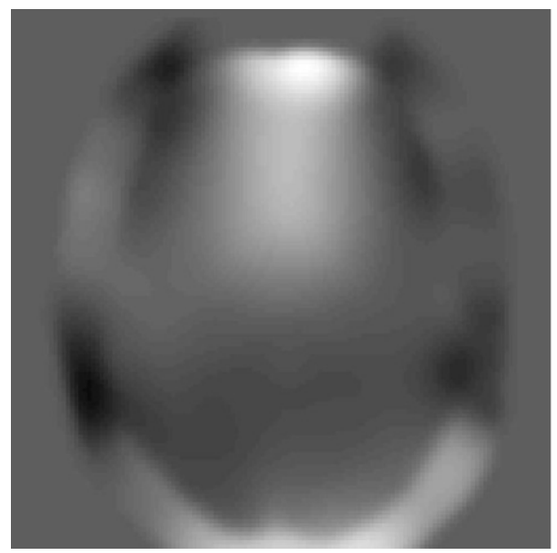


It is all caused by an “off-resonance” field

Off-resonance field ⇒ Distortions

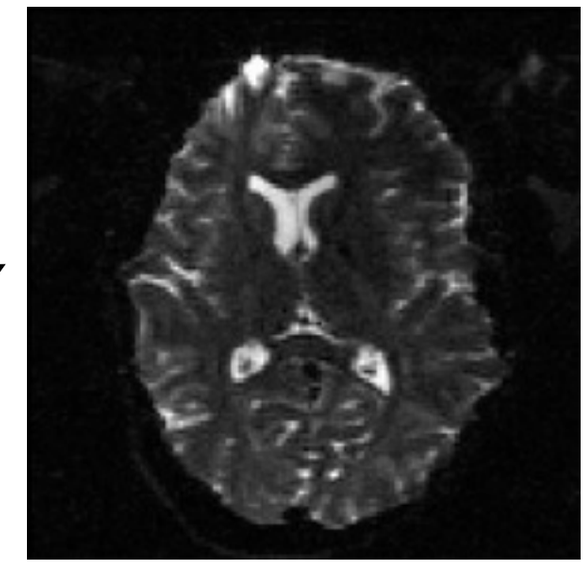


But this object

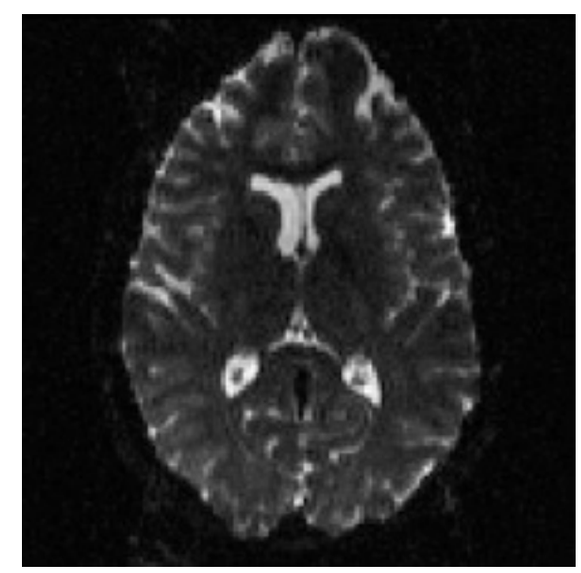


scanned in
this field

or this

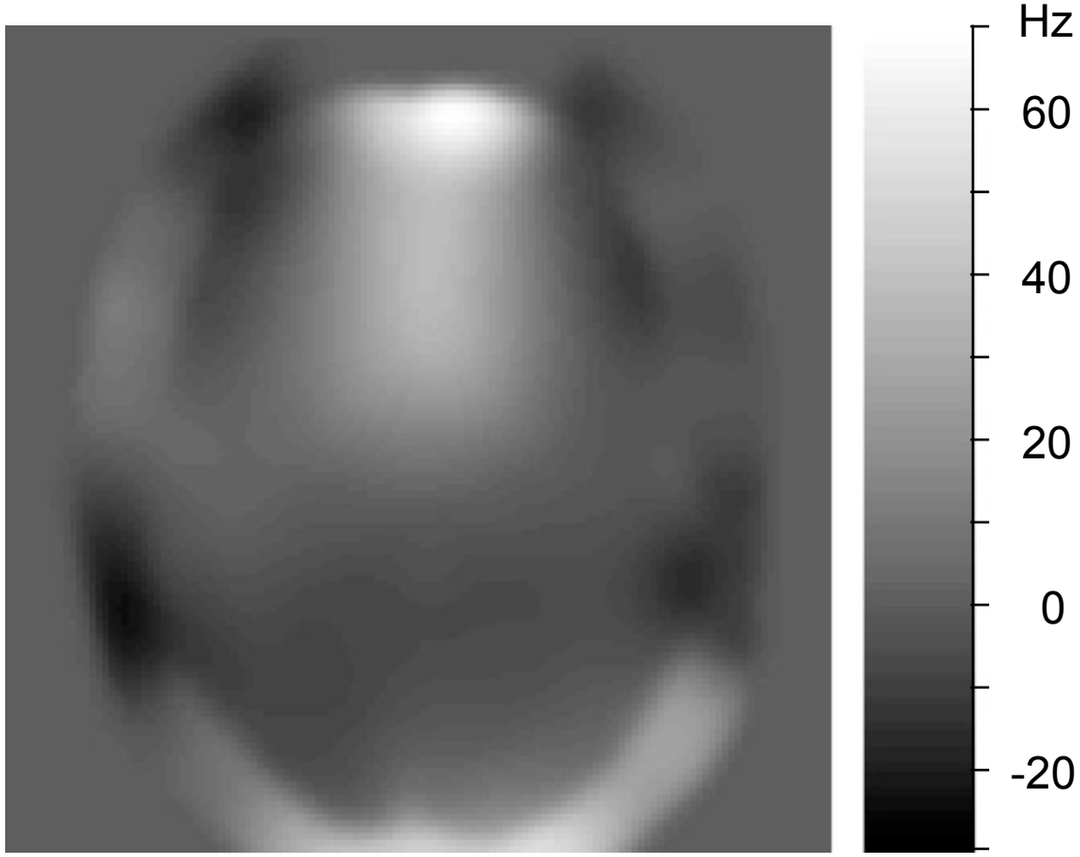


Can yield this



So there is clearly more to this story...

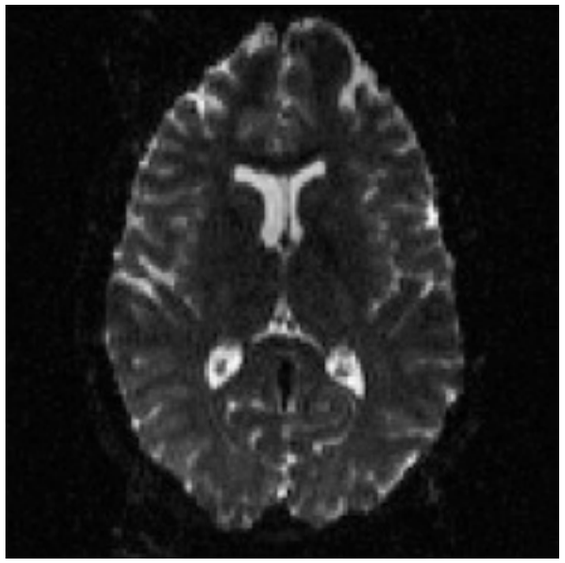
Off-resonance field ⇒ Distortions



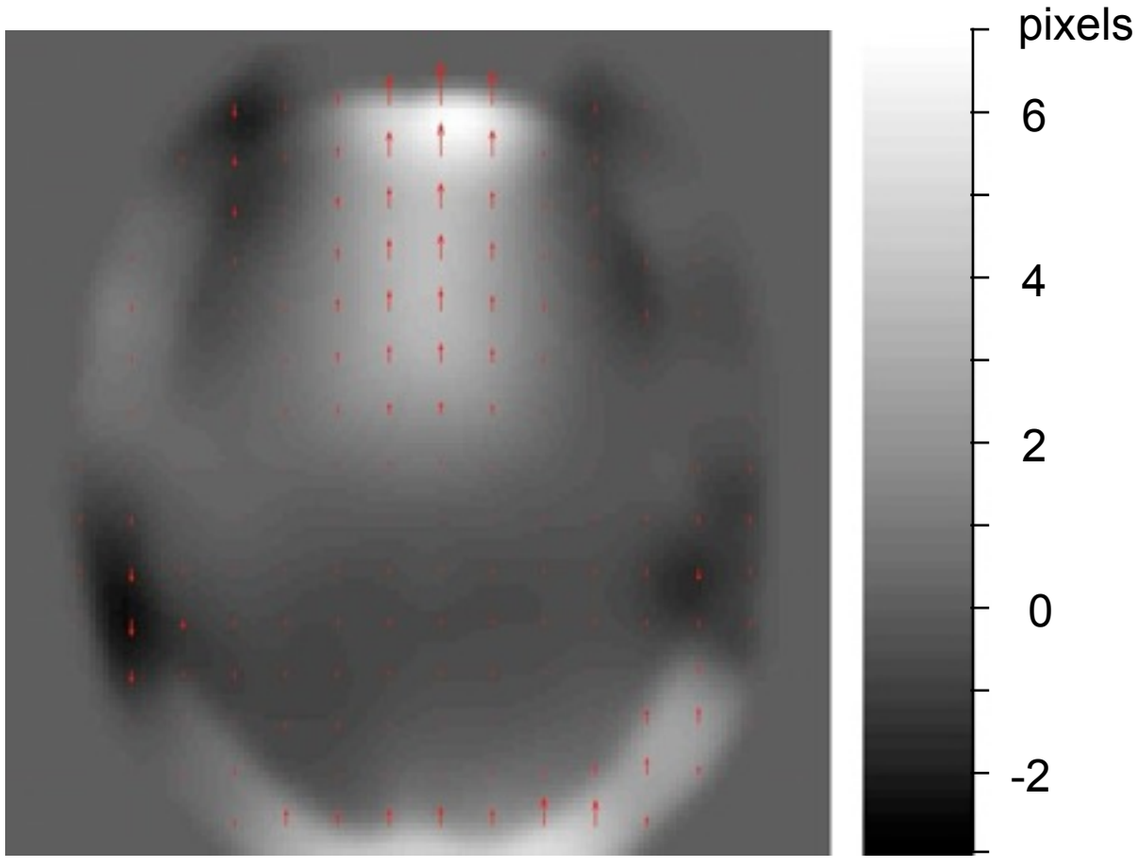
An off-resonance field is effectively a scaled voxel-displacement map.

If we know the imaging parameters we can do the translation.

Off-resonance field ⇒ Distortions



And know what to expect

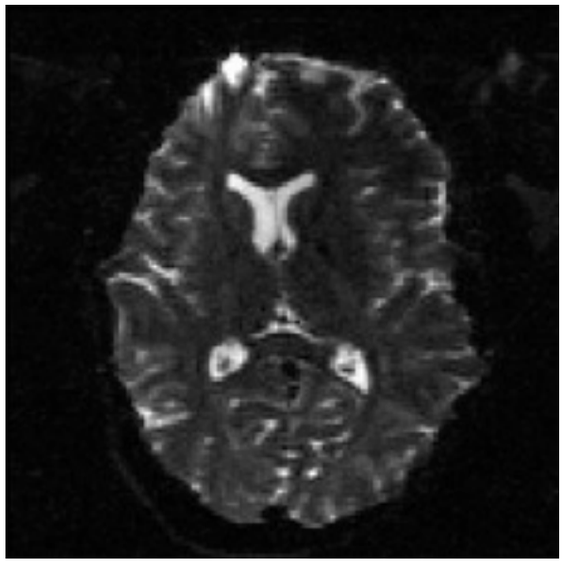


An off-resonance field is effectively a scaled voxel-displacement map.

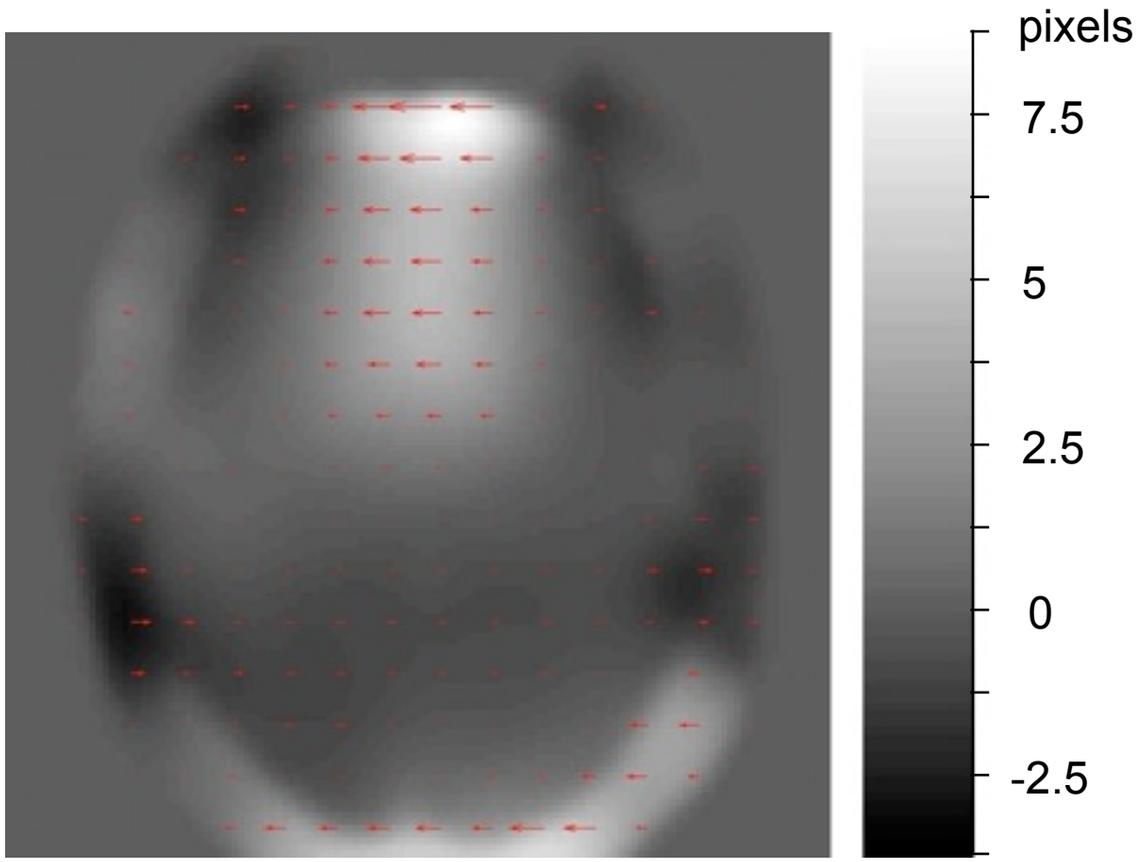
If we know the imaging parameters we can do the translation.

BW/pixel = 10Hz, **p** = [0 1 0]

Off-resonance field ⇒ Distortions



And know what to expect



So, an off-resonance field is effectively a scaled voxel-displacement map.

And if we know the imaging parameters we can do the translation.

BW/pixel = 8Hz, **p** = [-1 0 0]

Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

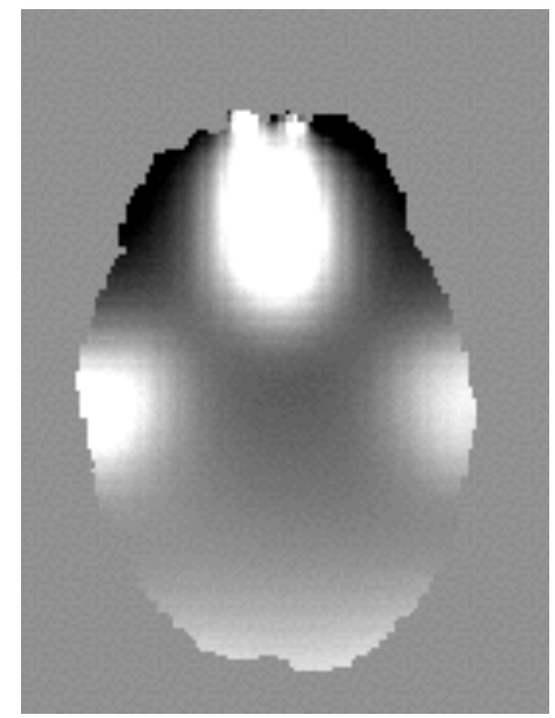
Where does the off-resonance field come from?

- There are two sources
- The first is the object (head) itself.

(CT of) Human head

Resulting field

$B_0 \odot$

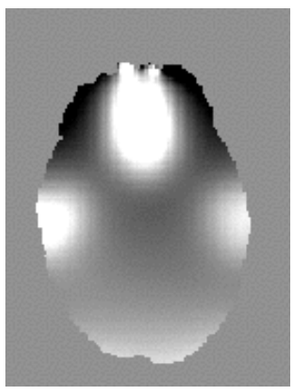


PPMs

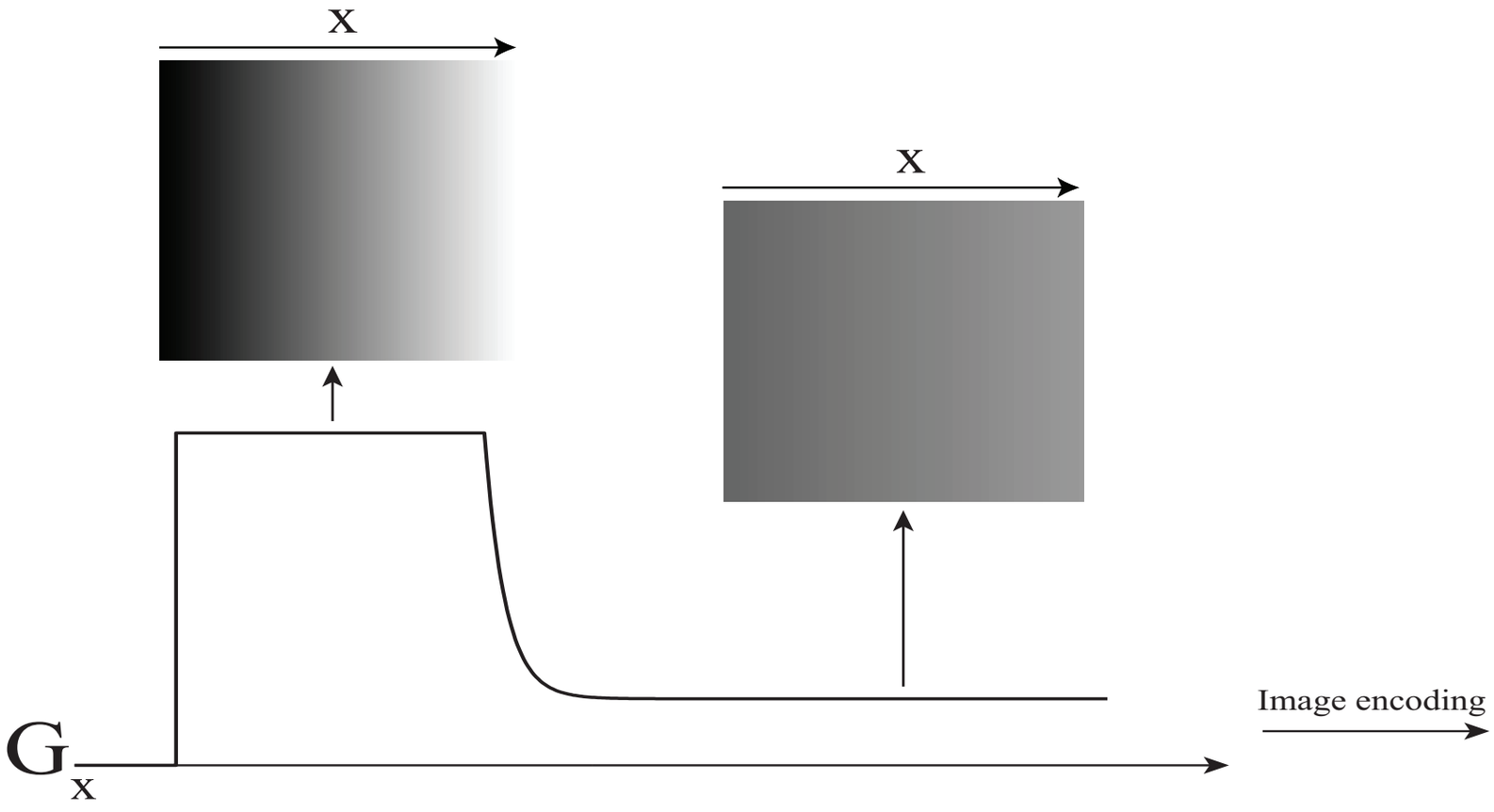
Must fulfil $\begin{cases} \nabla \times \mathbf{H} = 0 \\ \nabla \bullet \mathbf{B} = 0 \end{cases}$ (still)

Where does the off-resonance field come from?

- There are two sources
- The first is the object (head) itself.

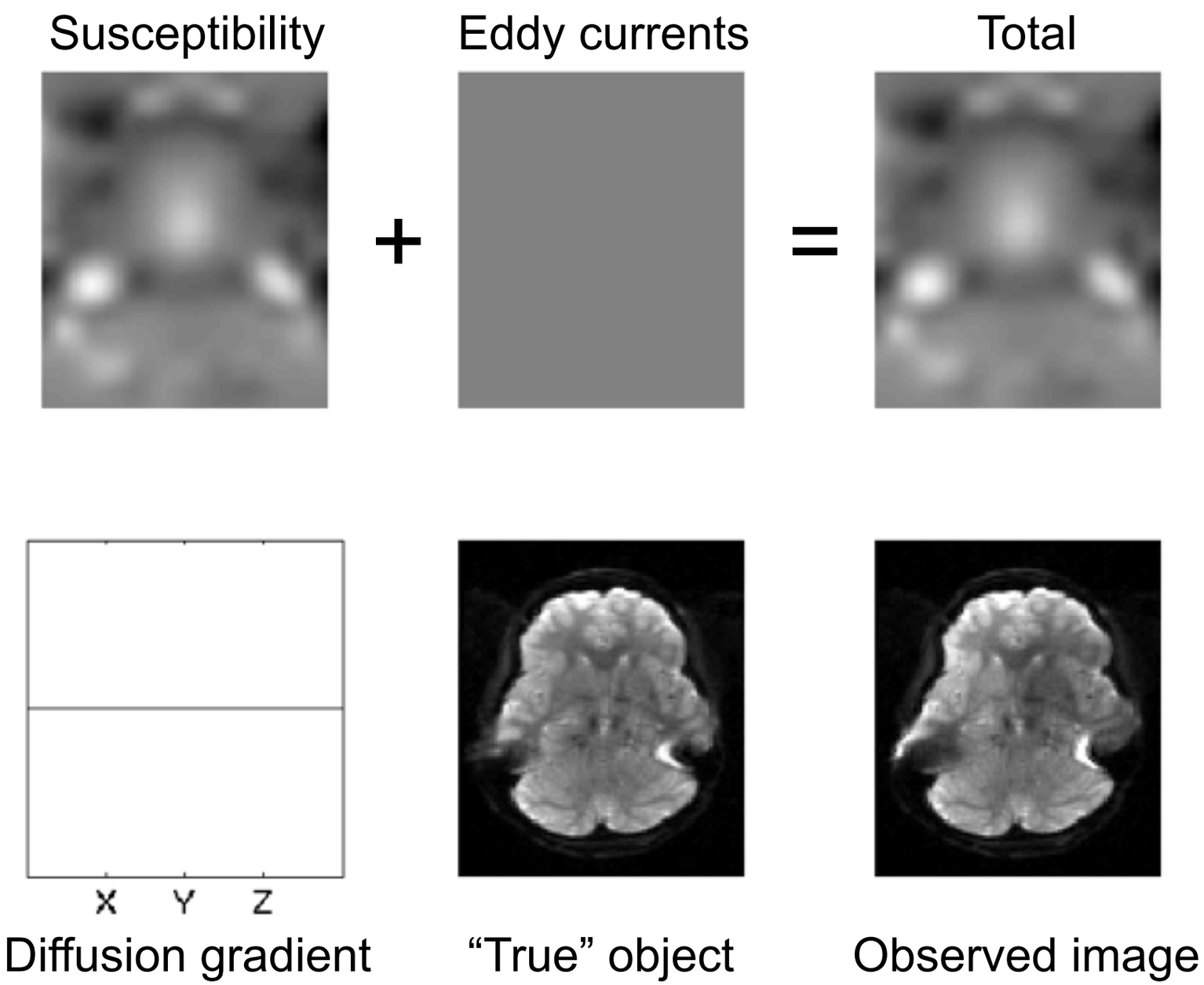


- The second is caused by the diffusion gradient



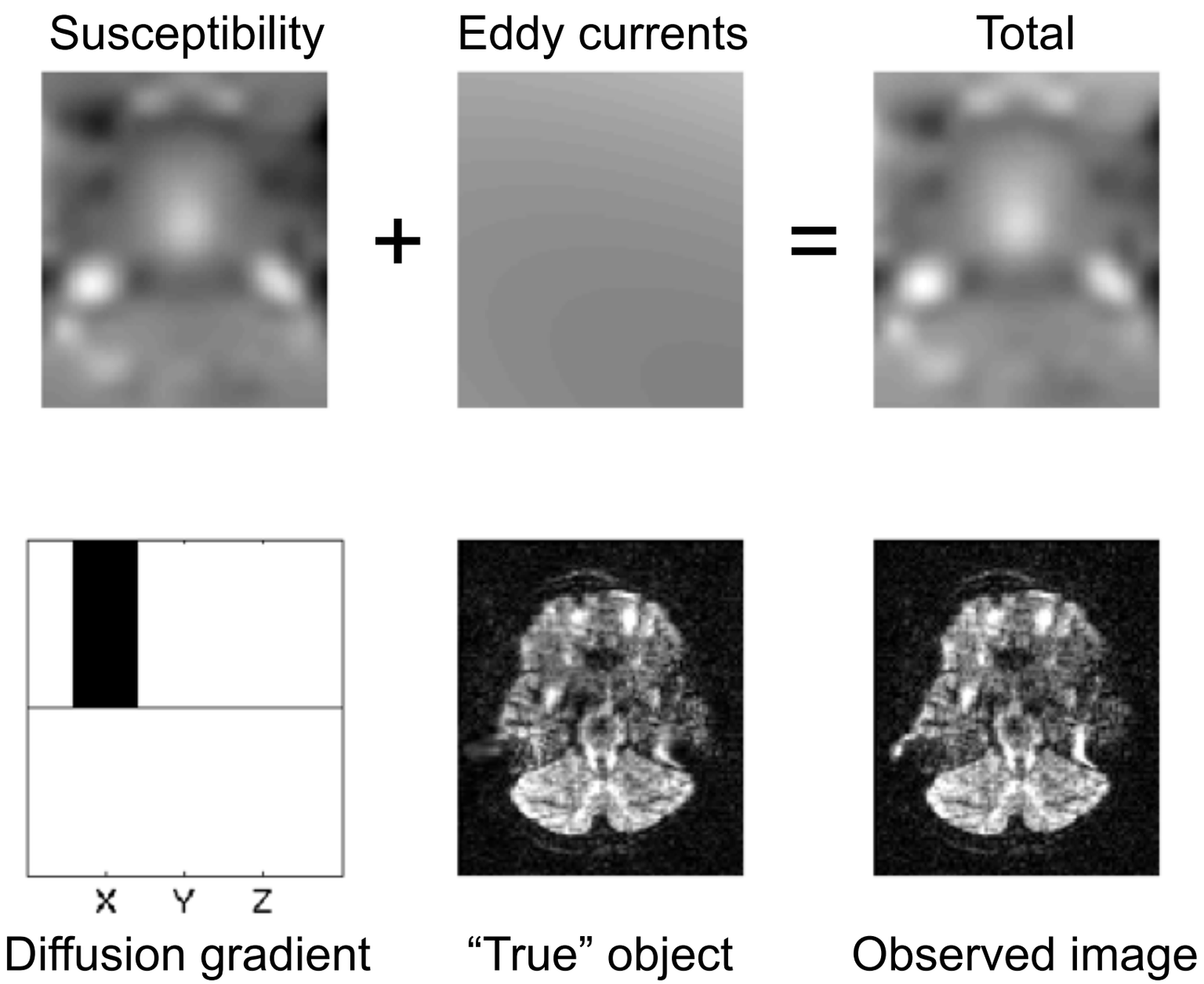
Where does the off-resonance field come from?

So for any diffusion weighted volume the off-resonance field is the sum of these two contributions



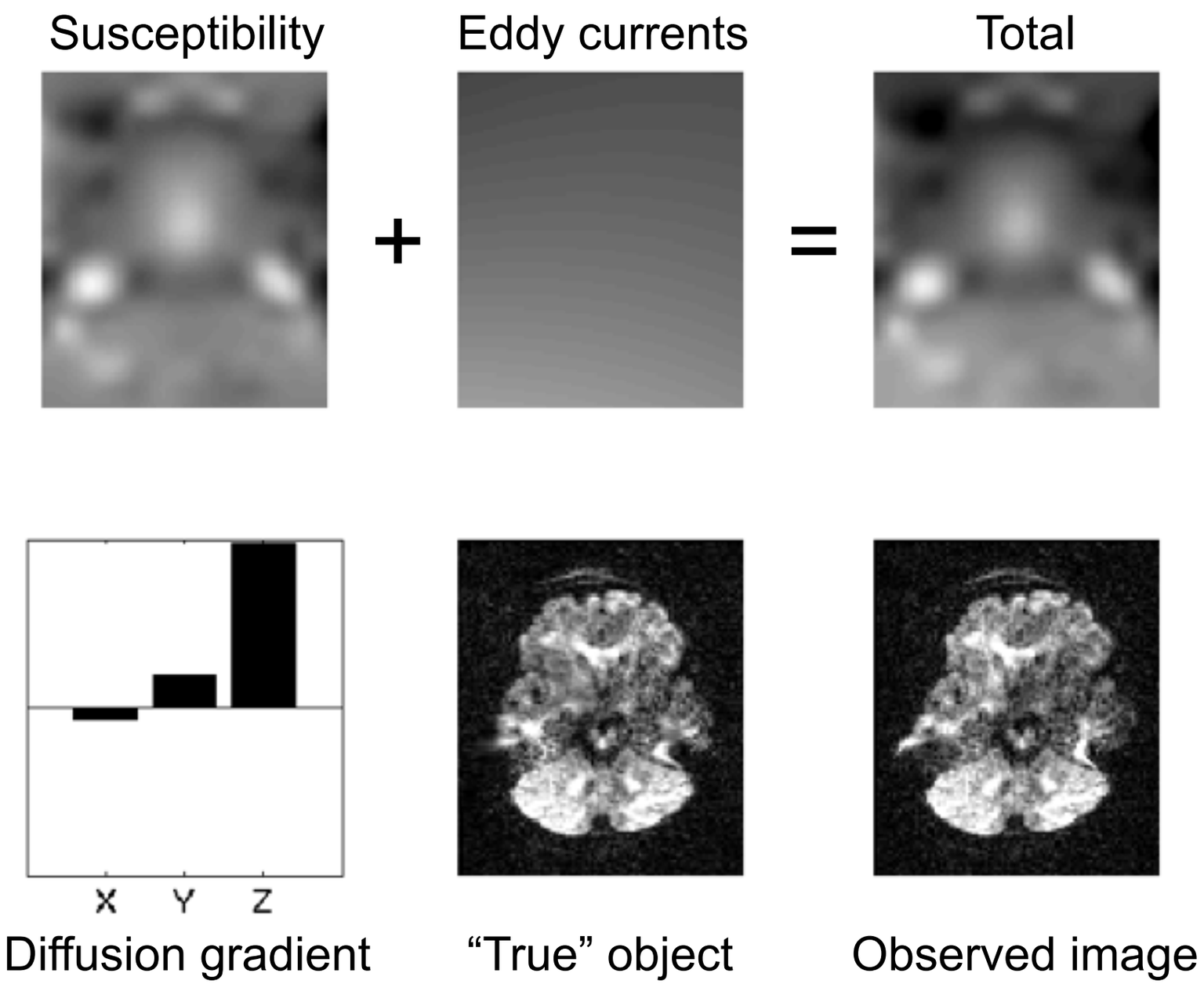
Where does the off-resonance field come from?

So for any diffusion weighted volume the off-resonance field is the sum of these two contributions



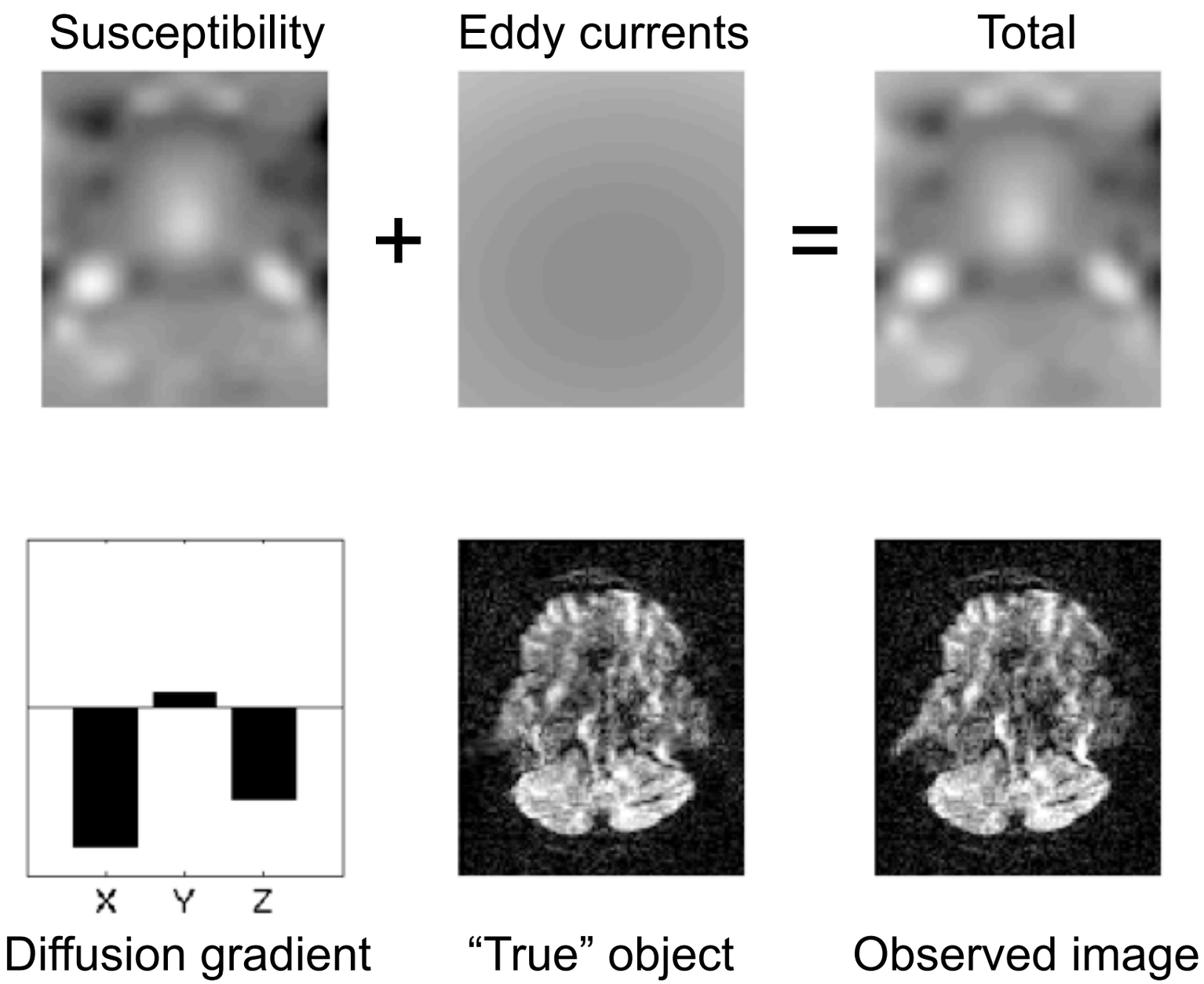
Where does the off-resonance field come from?

So for any diffusion weighted volume the off-resonance field is the sum of these two contributions



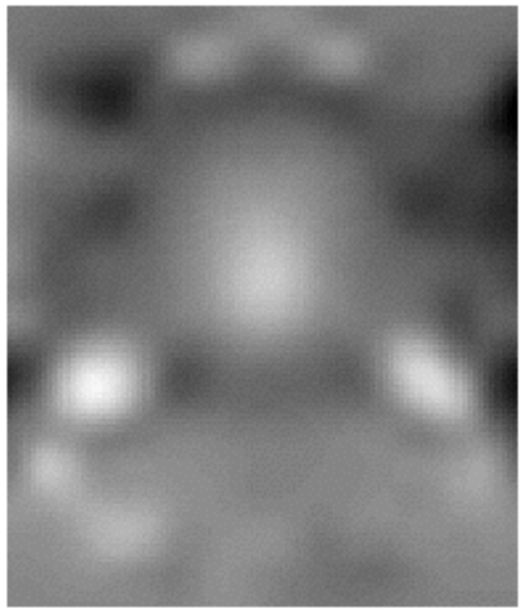
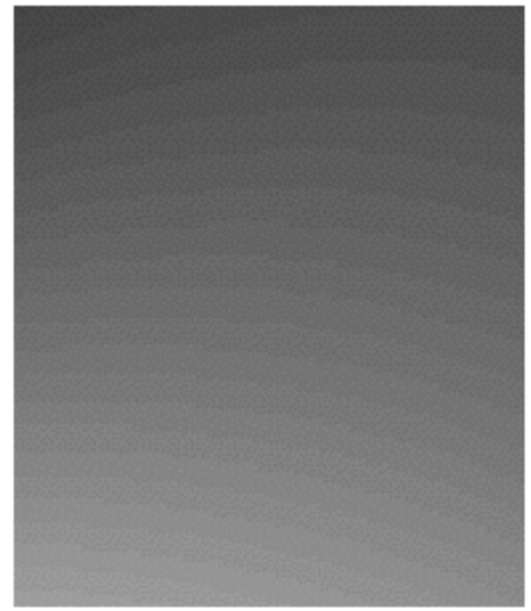
Where does the off-resonance field come from?

So for any diffusion weighted volume the off-resonance field is the sum of these two contributions



Separate estimation of susceptibility- and eddy current-fields

So, what we need to estimate is

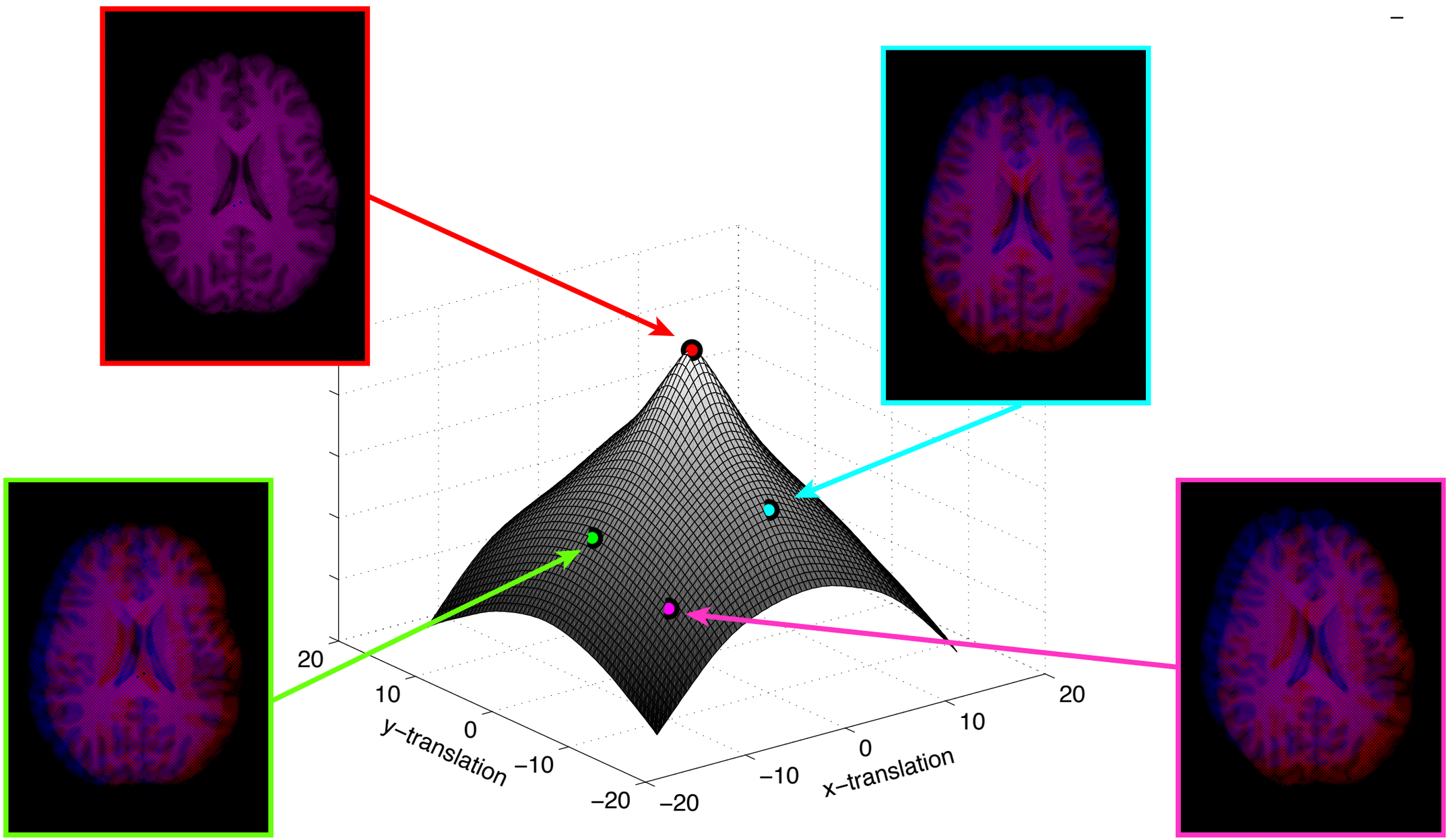
One of these per subject	One of these per volume
	

FSL-tools: topup eddy

Outline of the talk

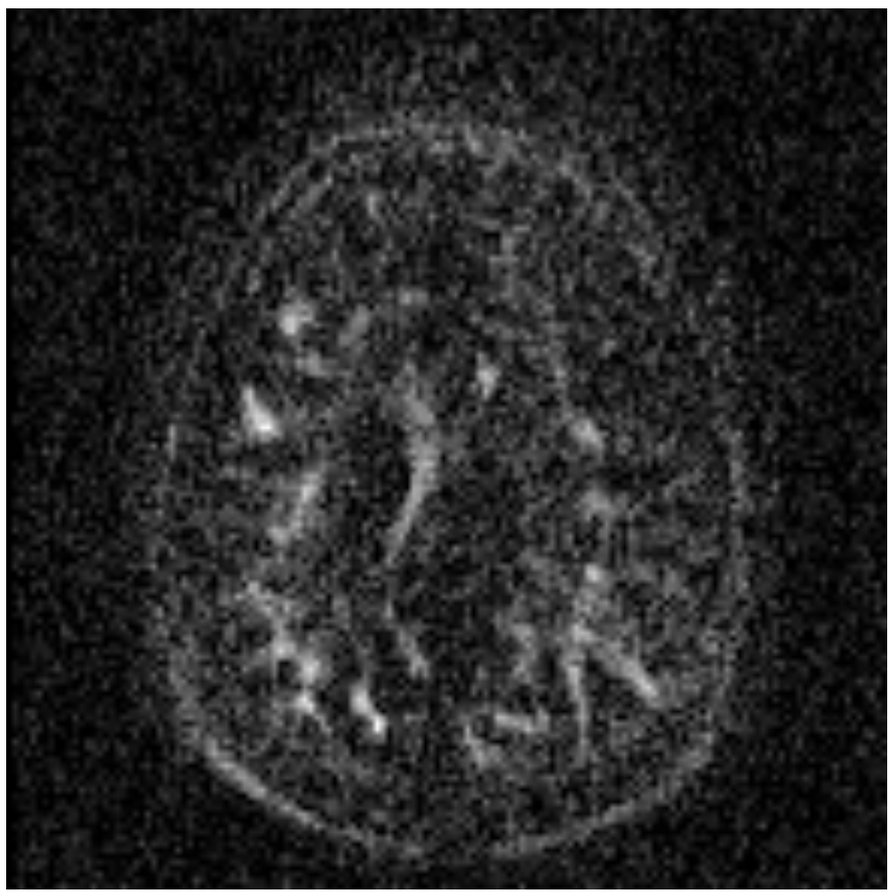
- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output
- Some results

Worlds shortest course on image registration

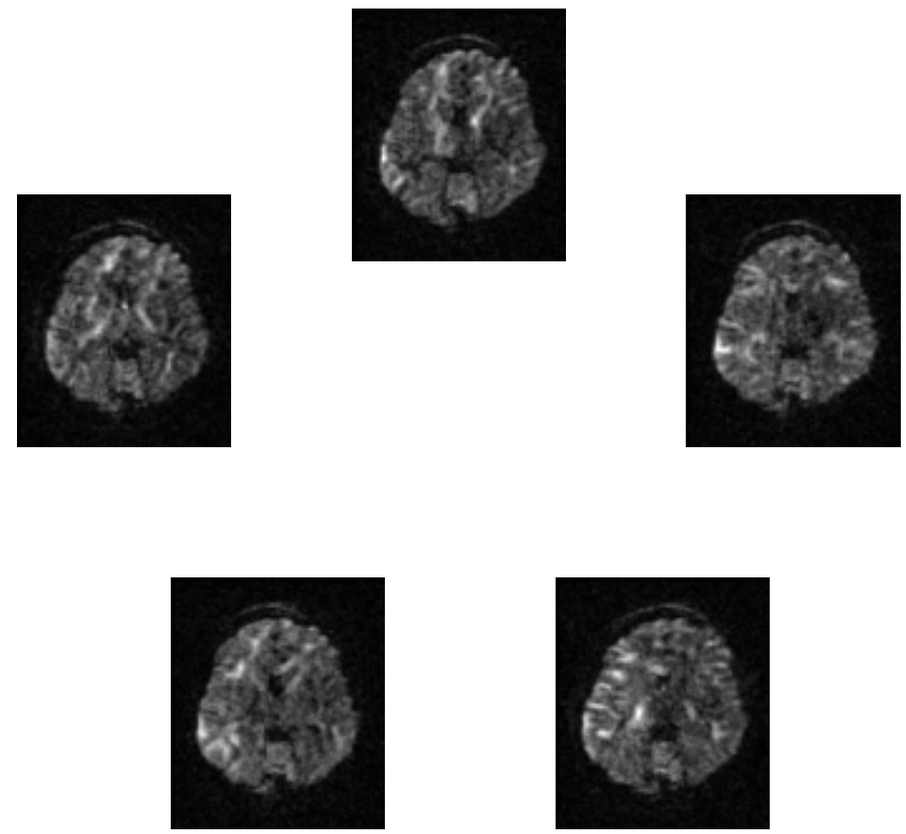


Maximising/minimising an objective/cost-function

But it is not easy to register diffusion weighted images



The different diffusion weighted images have different contrast.

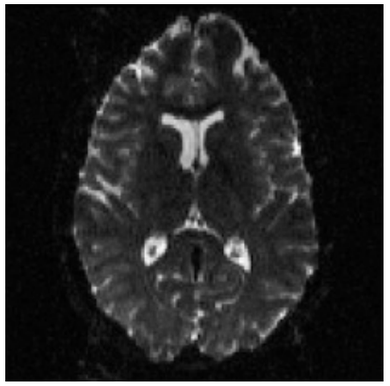


All the images are distorted, only differently. How do we know the truth?

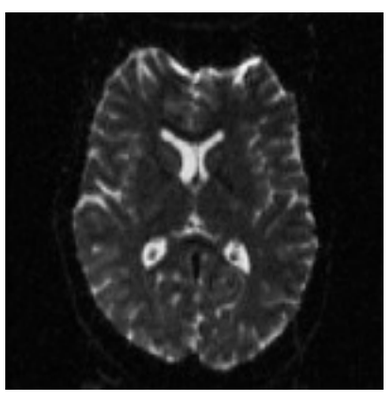
Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

How topup works (very briefly)



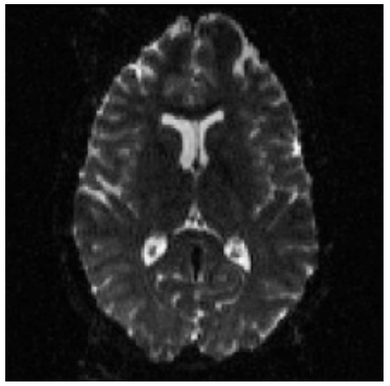
p=[0 1 0]



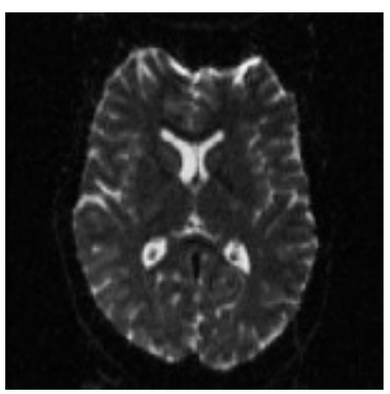
p=[0 -1 0]

Given two images acquired with different phase-encoding

How topup works (very briefly)



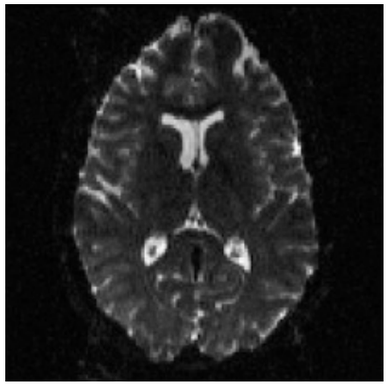
p=[0 1 0]



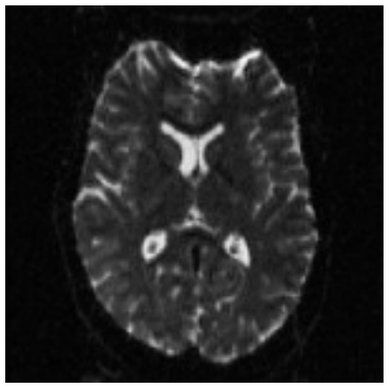
p=[0 -1 0]

topup “guesses” a field...

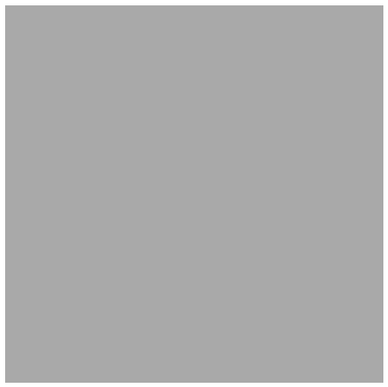
How topup works (very briefly)



p=[0 1 0]

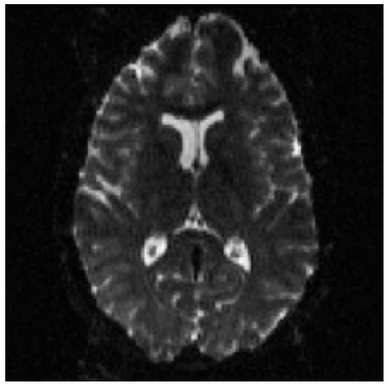


p=[0 -1 0]

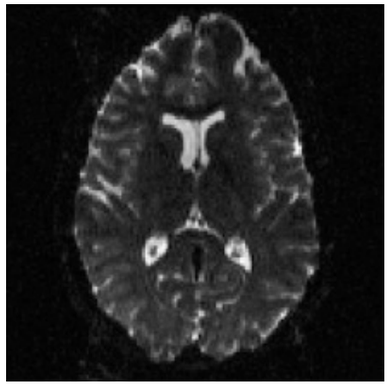


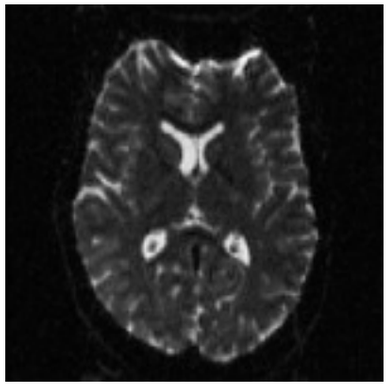
...calculates the displacement maps...

How topup works (very briefly)

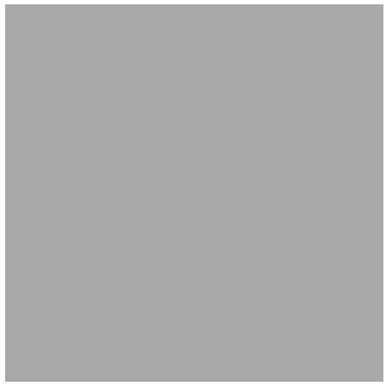


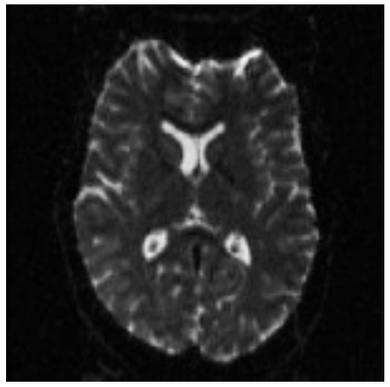
p=[0 1 0]





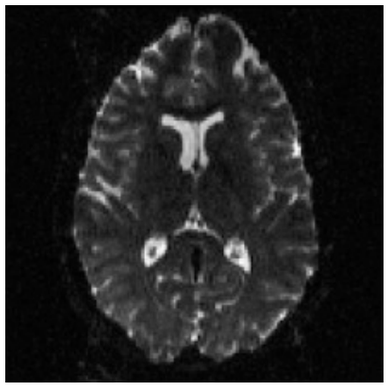
p=[0 -1 0]



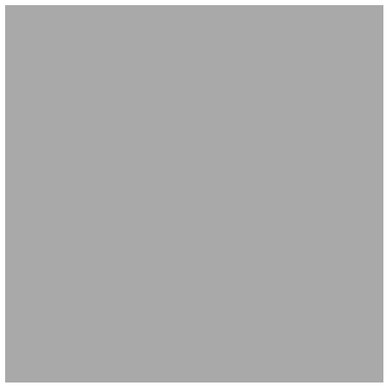


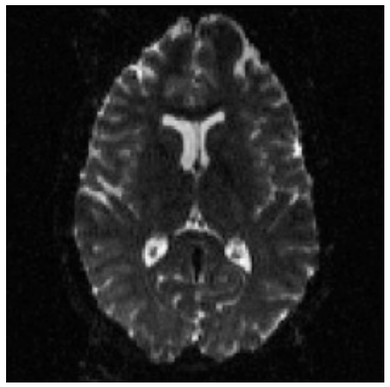
...”corrects” the images...

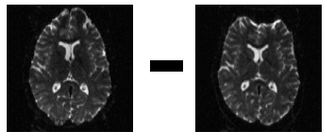
How topup works (very briefly)

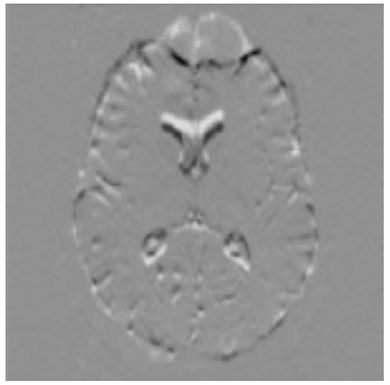


p=[0 1 0]

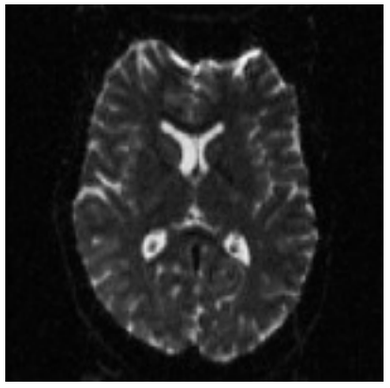




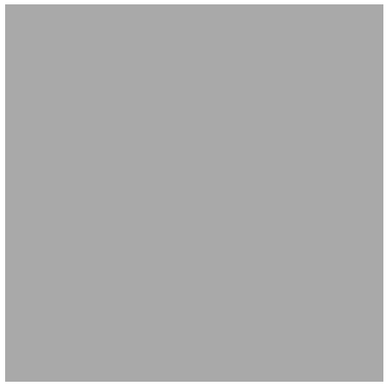


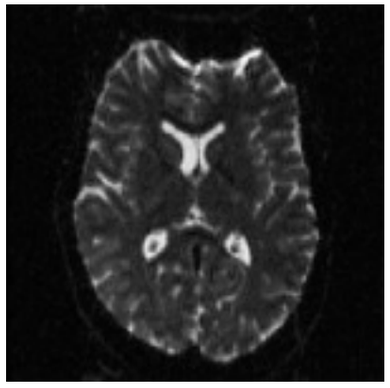


BAD!



p=[0 -1 0]

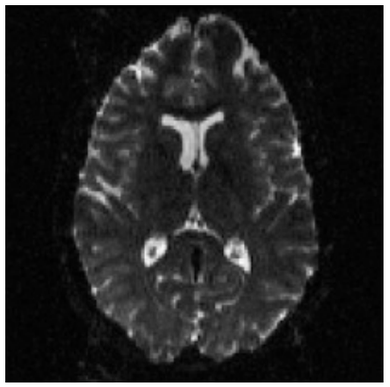




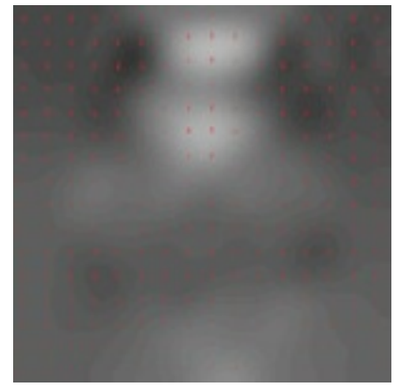
...and evaluates the results...

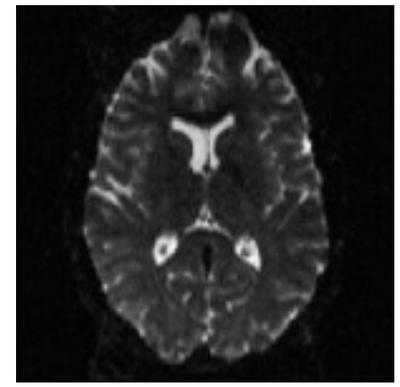
And **this** is the crucial bit.

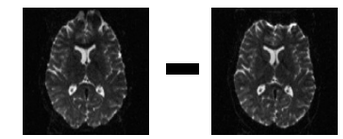
How topup works (very briefly)

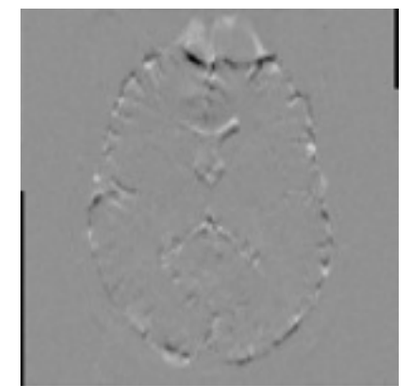


p=[0 1 0]

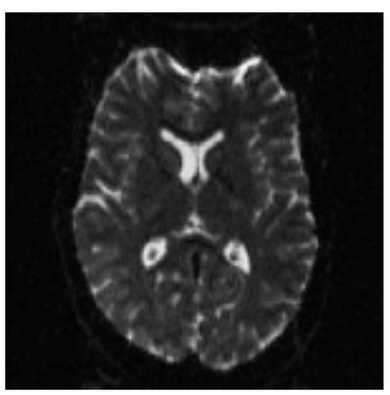




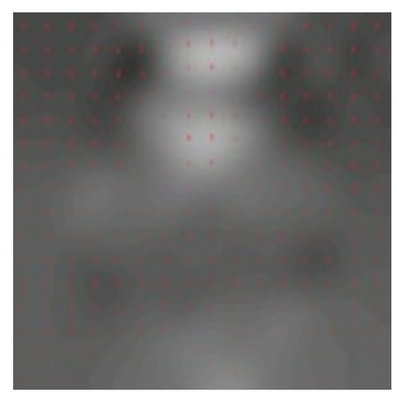


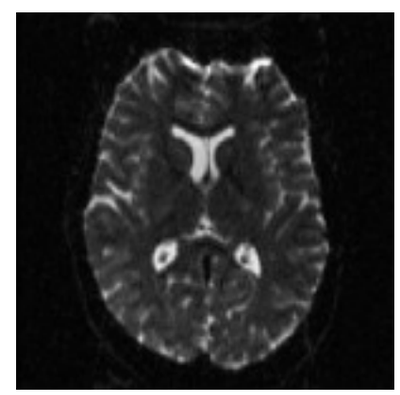


better



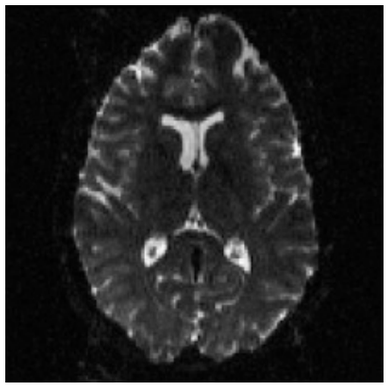
p=[0 -1 0]



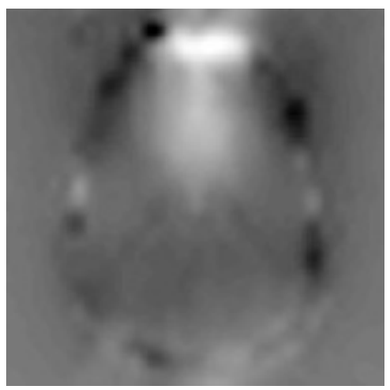


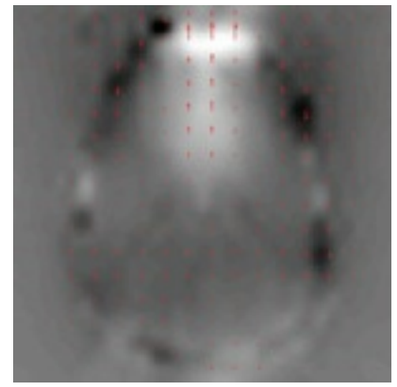
Because topup can then “guess” another field

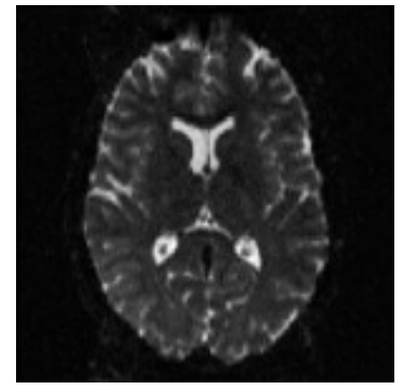
How topup works (very briefly)

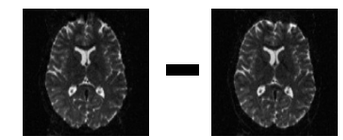


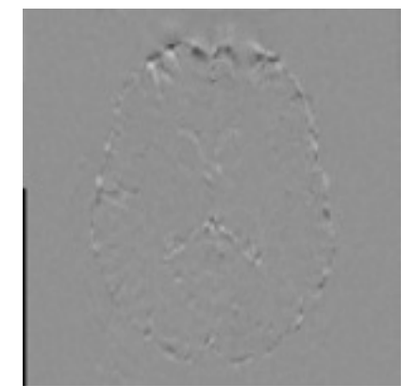
p=[0 1 0]



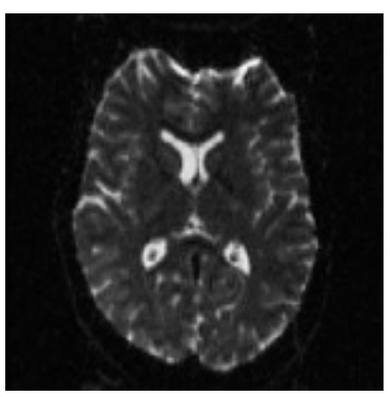




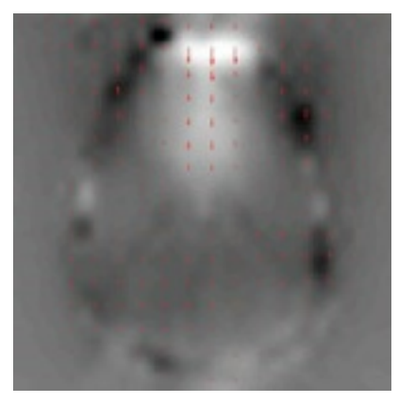


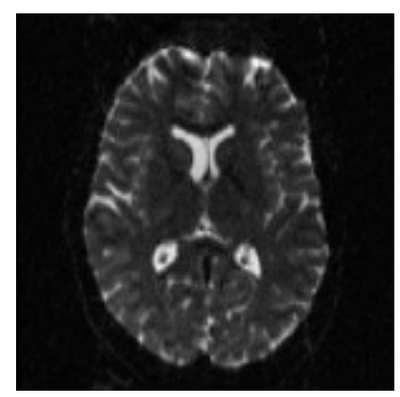


even better



p=[0 -1 0]



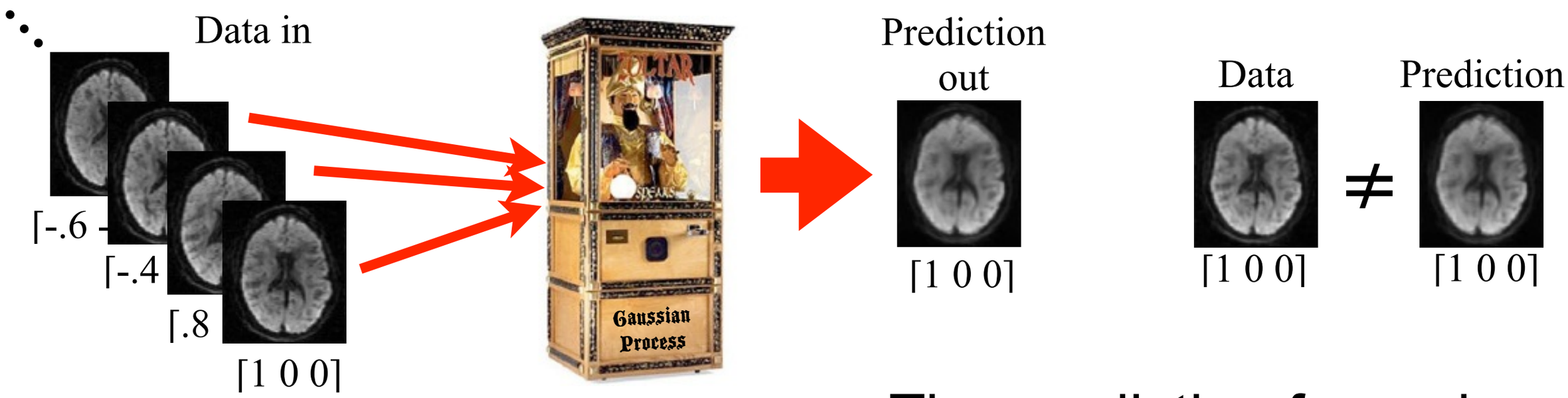


...and another...until it is happy,
and then it “knows” the field

Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

Zoltar -- The prediction maker



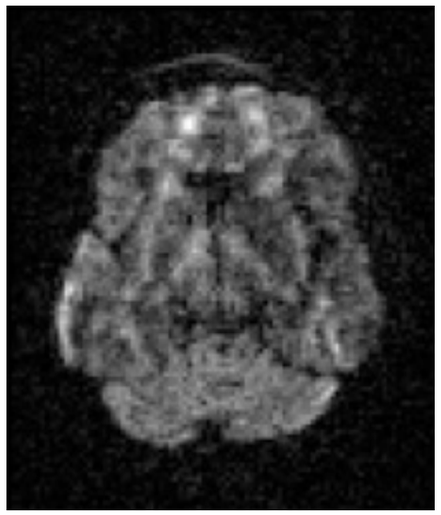
Given some data in, Zoltar will make a prediction what the data “should” be.

The prediction for a given dwi will not be identical to the “input” for that dwi

I know this sounds crazy, but please trust me on this.
(Zoltar is actually a Gaussian Process)

How eddy works: Loading step

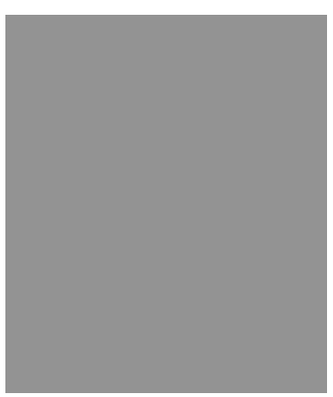
Pick the first dwi



Use current estimates of

Susc

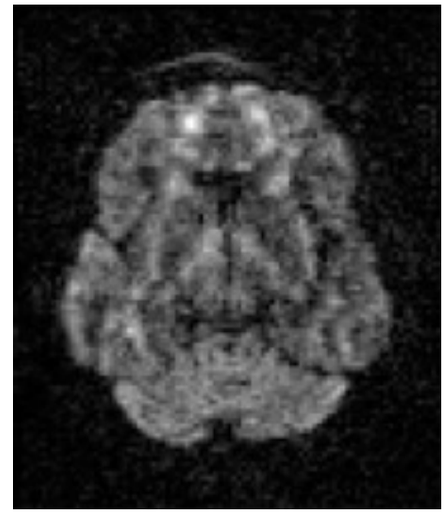
EC



MP

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

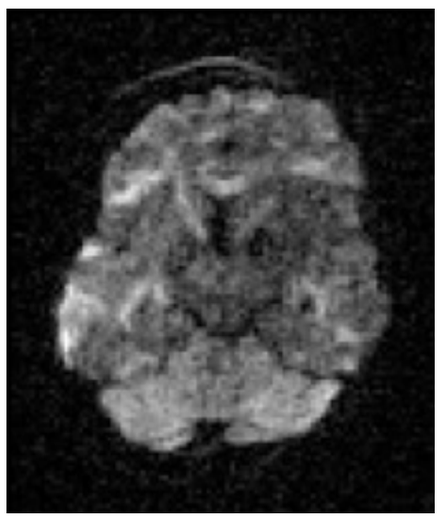
To correct image



And load into prediction maker

How eddy works: Loading step

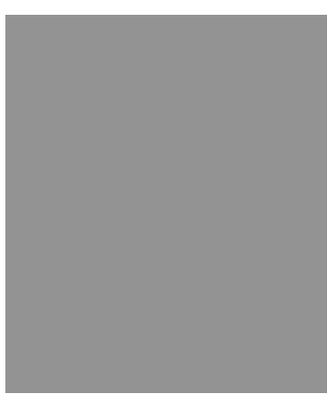
then the 2nd dwi



Use current estimates of

Susc

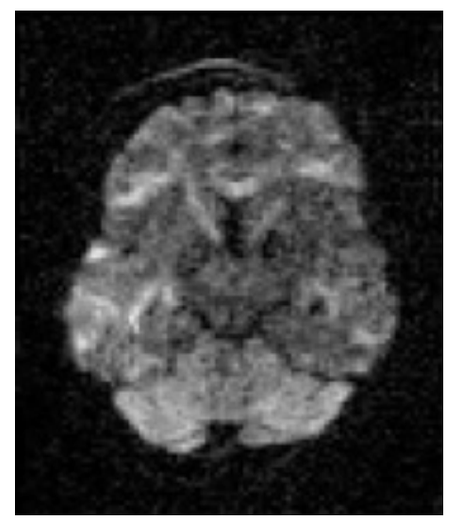
EC



MP

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

To correct 2nd image

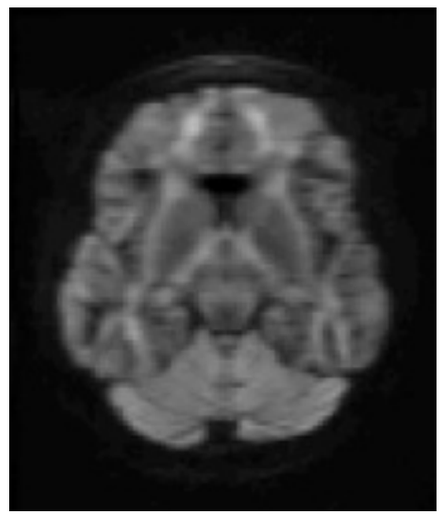


And load into prediction maker

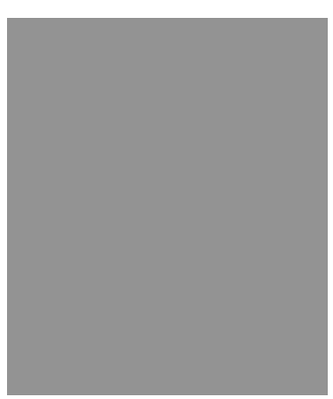
Until we have loaded all dwis

How eddy works: Estimation step

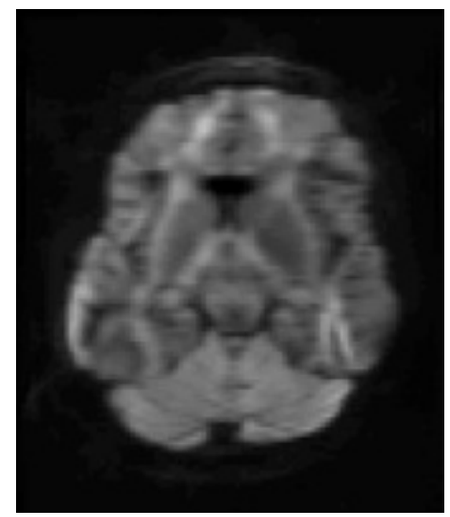
Draw a prediction
for first dwi



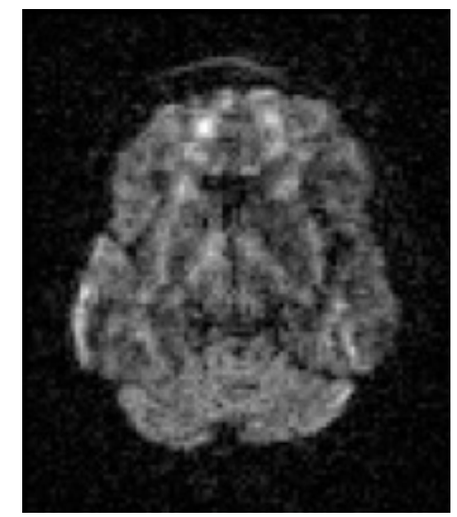
Use current estimates of

Susc	EC	MP
		$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Invert



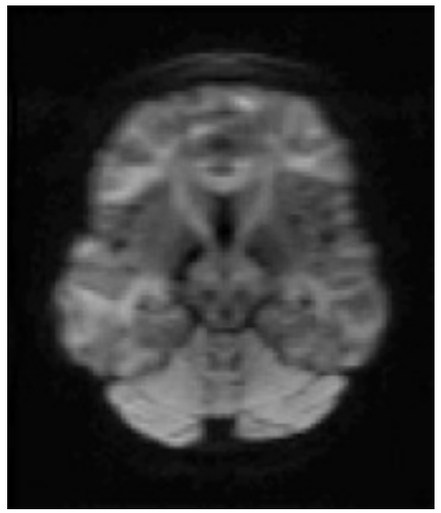
To get prediction in "observation space"



And compare to actual observation

How eddy works: Estimation step

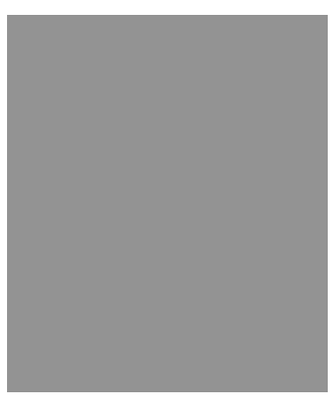
Draw a prediction
for 2nd dwi



Use current estimates of

Susc

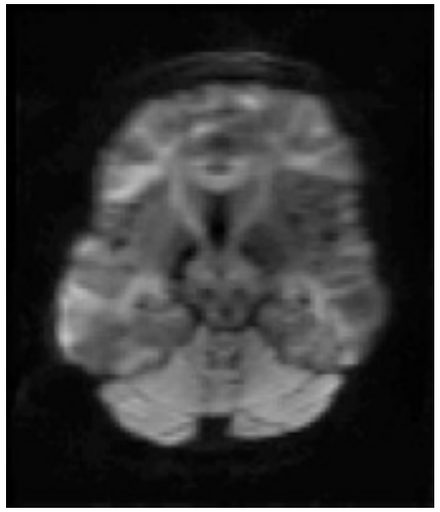
EC

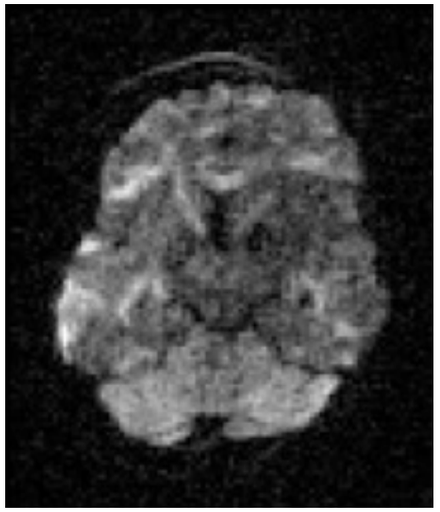


MP

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Invert



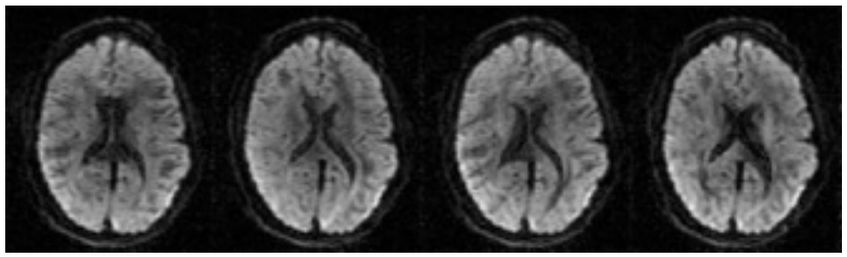


And then we repeat
the procedure for the
next dwi ...

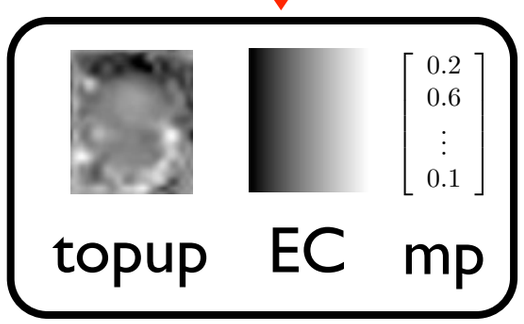
How eddy works

1. **For all scans**

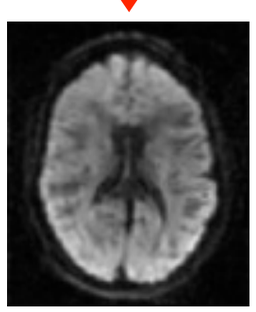
[1 0 0] [.6 -.4 -.7] [.8 .6 0] [-.4 .9 0]



…



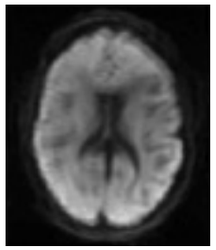
Use susceptibility field and current estimate of EC and movement to "unwarp" scan



Load into prediction maker

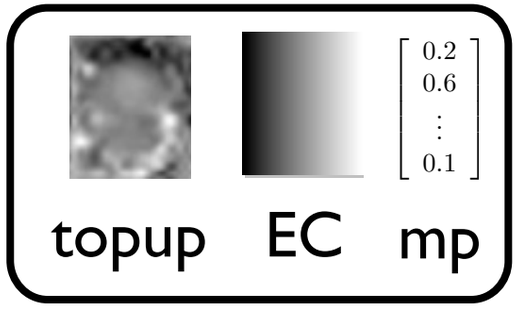
2. **For all scans**

[1 0 0]



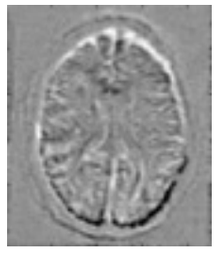
Get prediction

Invert current transform

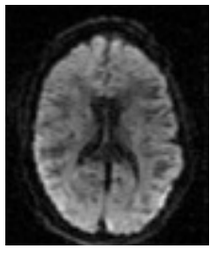


Get prediction in scan space

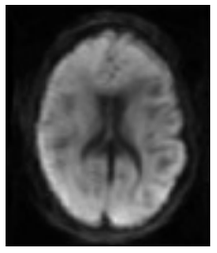
Use difference to update EC and mp



[1 0 0]

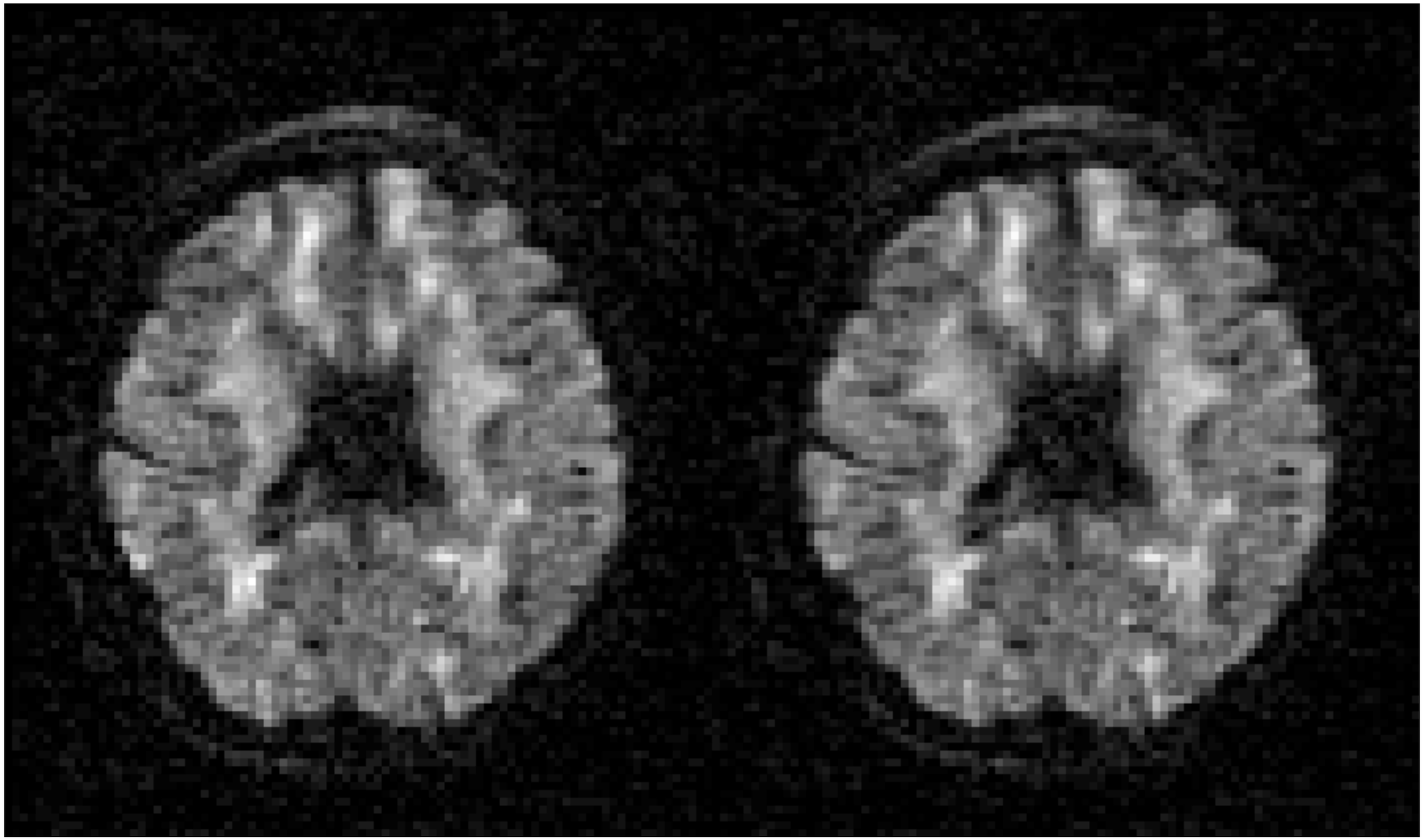


[1 0 0]

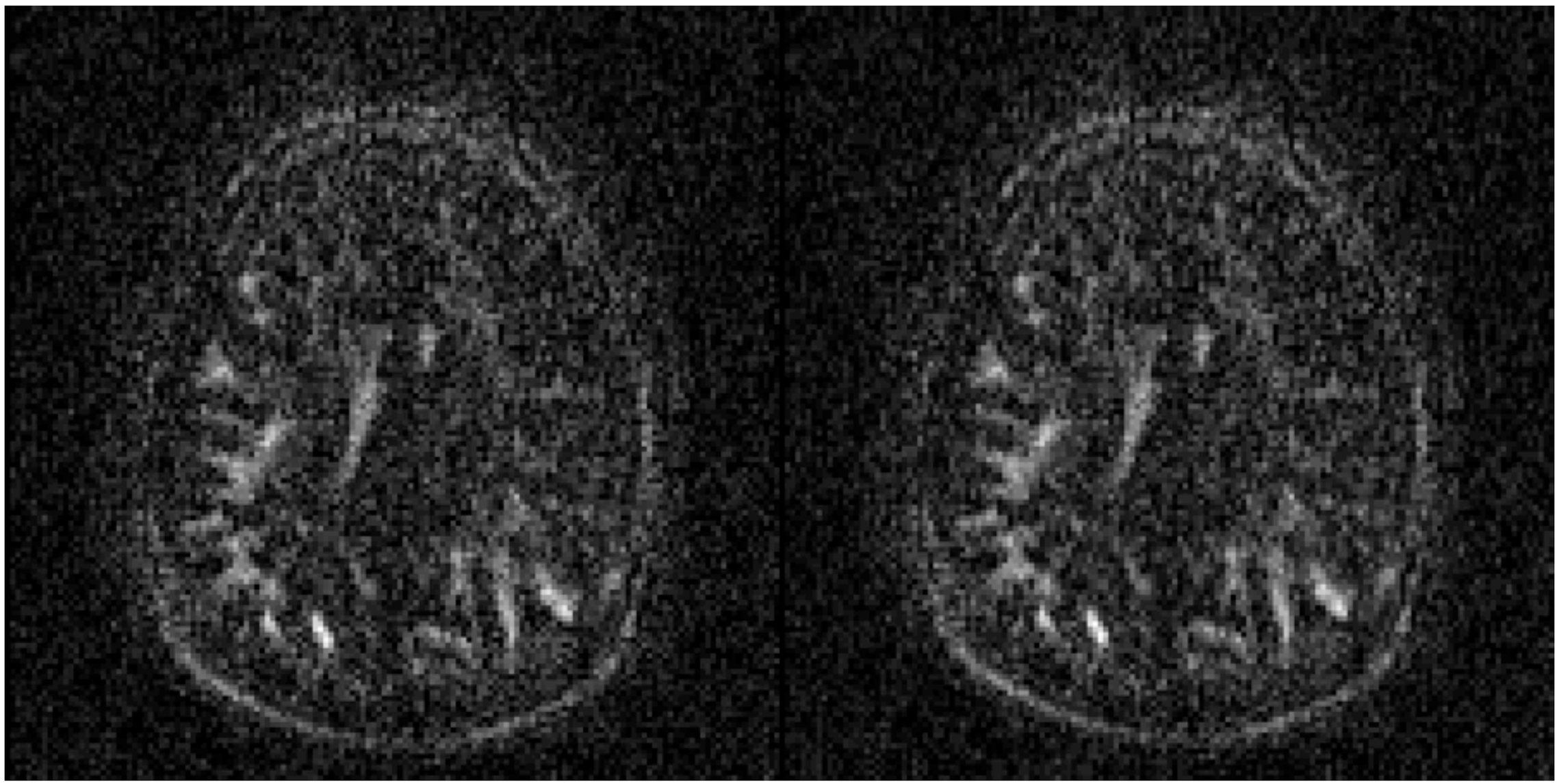


Compare to scan

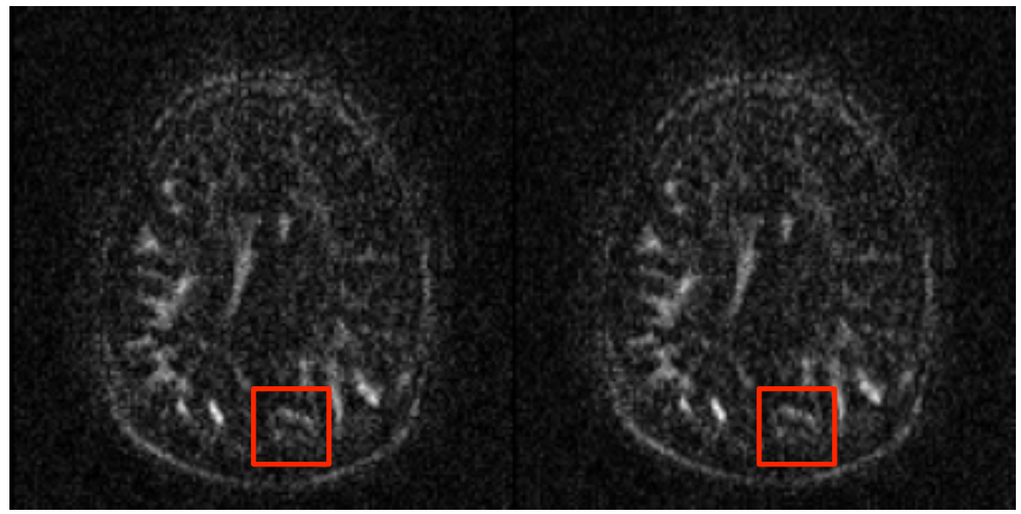
HCP-data, 150 directions, b=3000, blip-left-blip-right

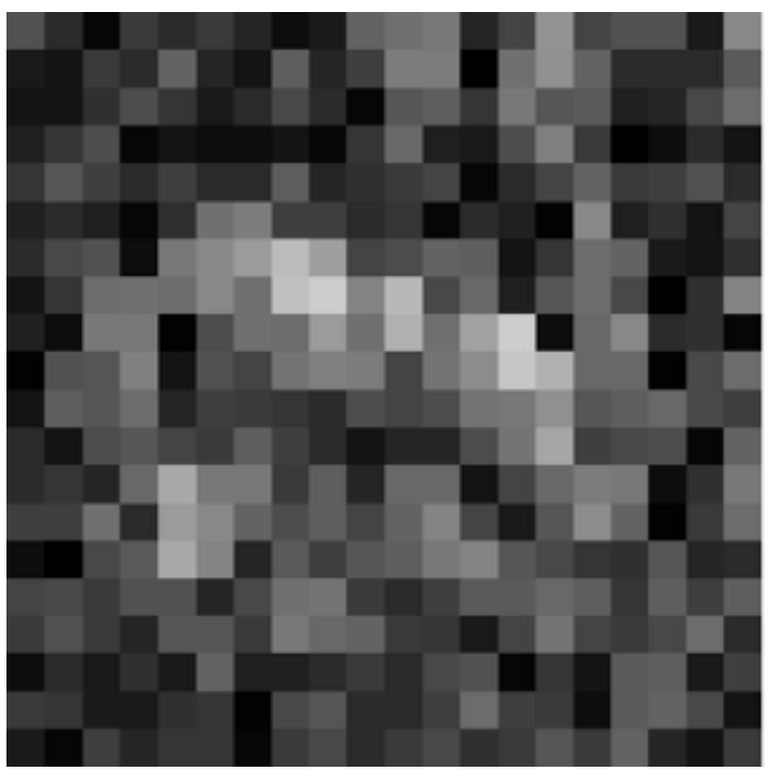


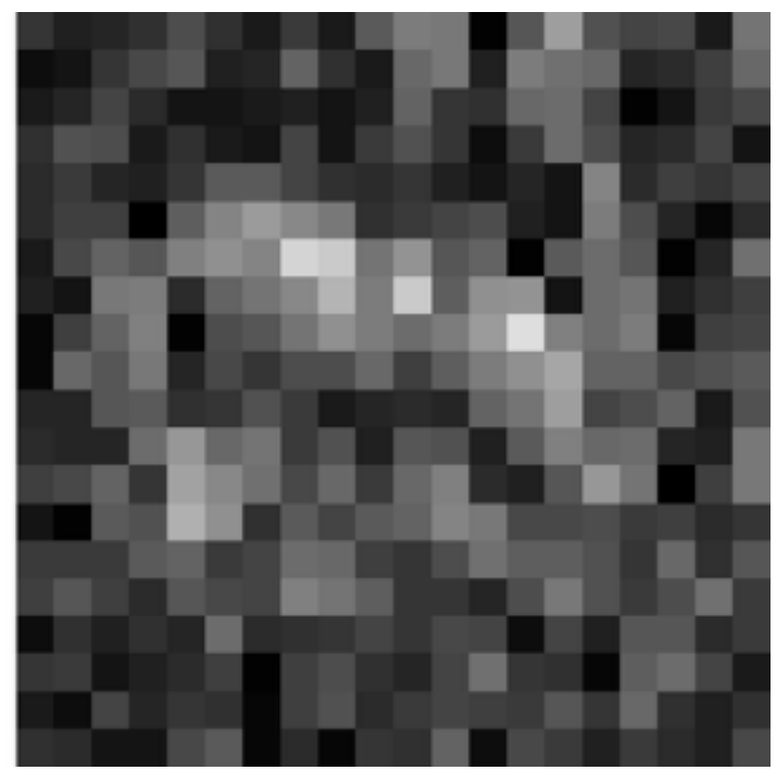
MGH-data, 198 directions, b=10000!



MGH-data, 198 directions, b=10000!







Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

Movement induced dropout

Diffusion encoding

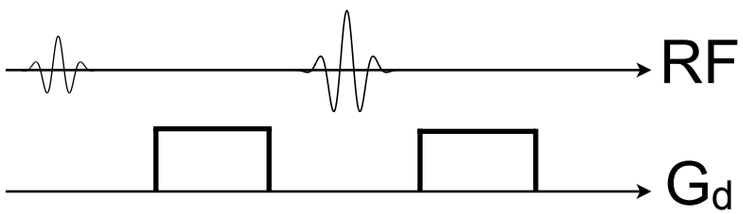
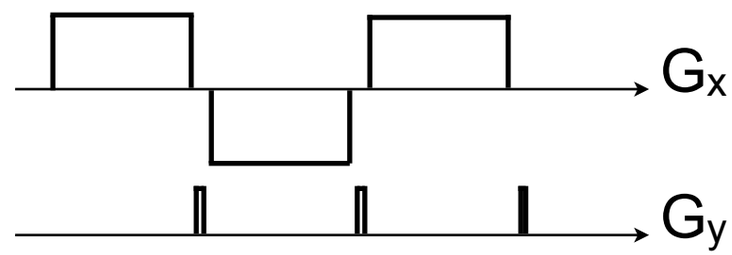
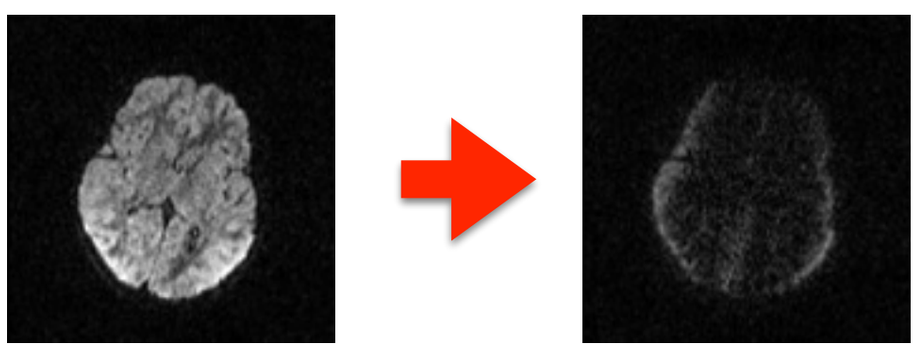


Image encoding

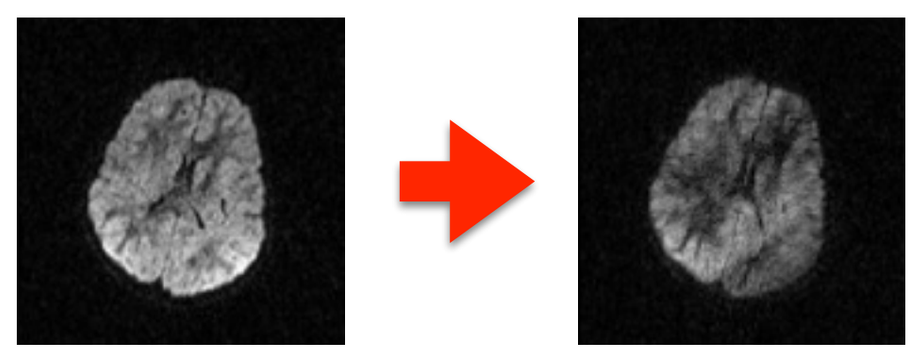


…

If there is movement during this part…



this can turn to this

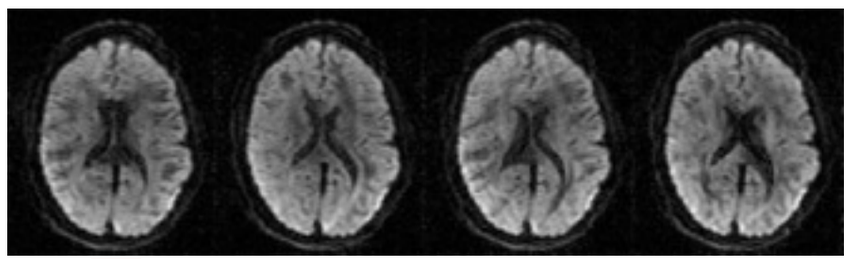


or this to this

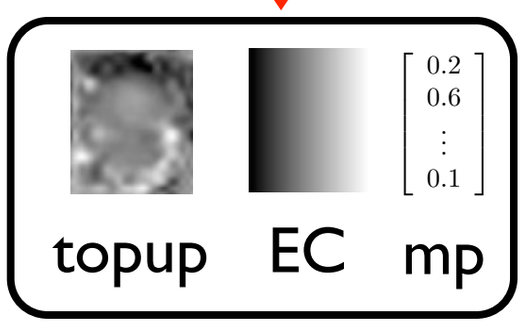
What can eddy do about it?
But first a little recap of eddy

1. **For all scans**

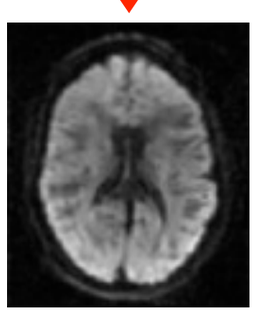
[1 0 0] [.6 -.4 -.7] [.8 .6 0] [-.4 .9 0]



...



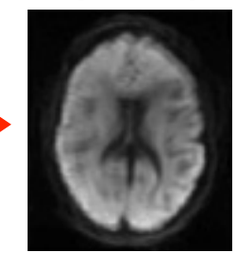
Use susceptibility field and current estimate of EC and movement to "unwarp" scan



Load into prediction maker

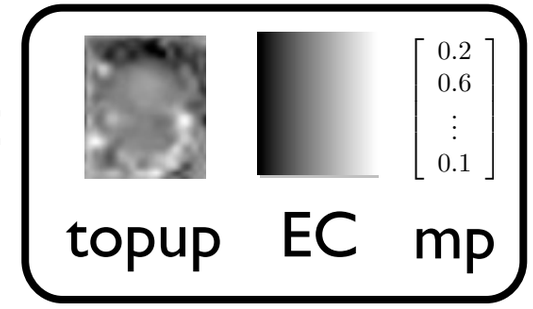
2. **For all scans**

[1 0 0]



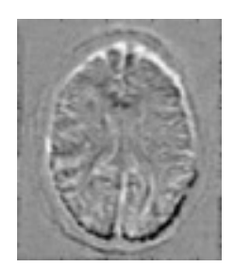
Get prediction

Invert current transform

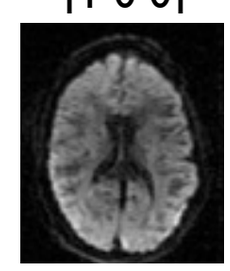


Get prediction in scan space

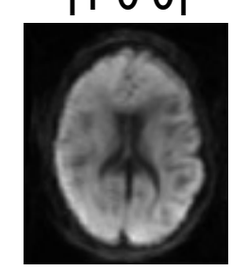
Use difference to update EC and mp



[1 0 0]

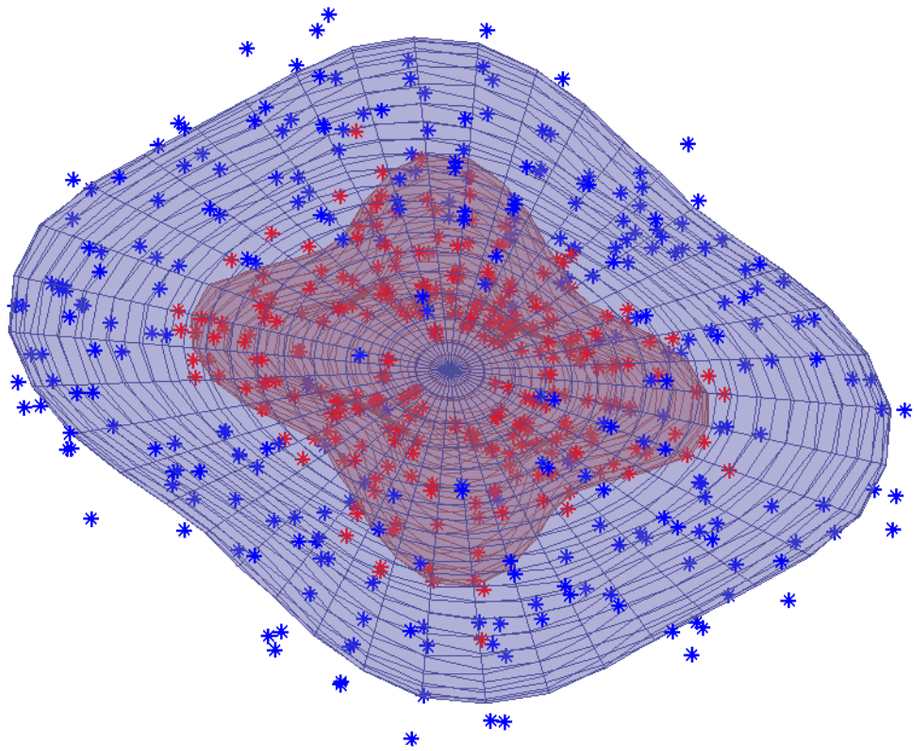


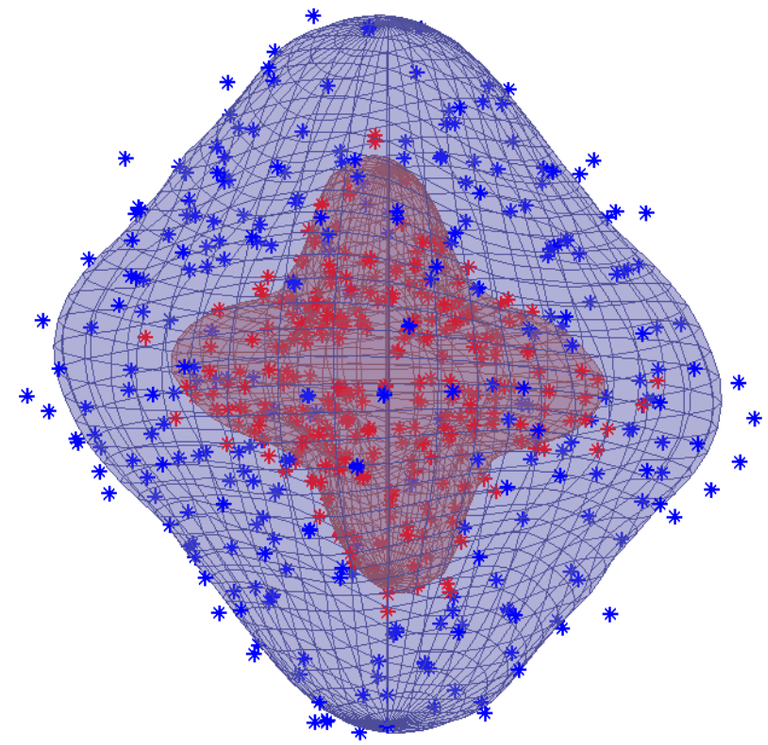
[1 0 0]



Compare to scan

How are the predictions made?

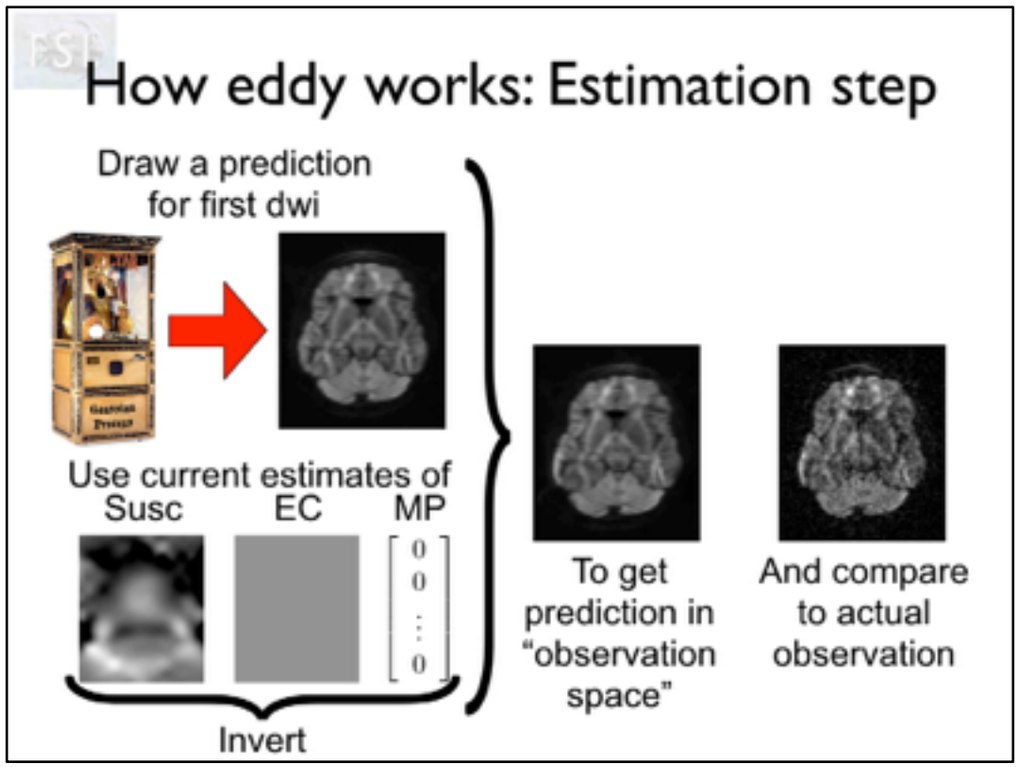




A Gaussian process that simply assumes that the signal varies smoothly as we move in Q-space
Very few assumptions. Hyperparameters calculated by leave-one-out.

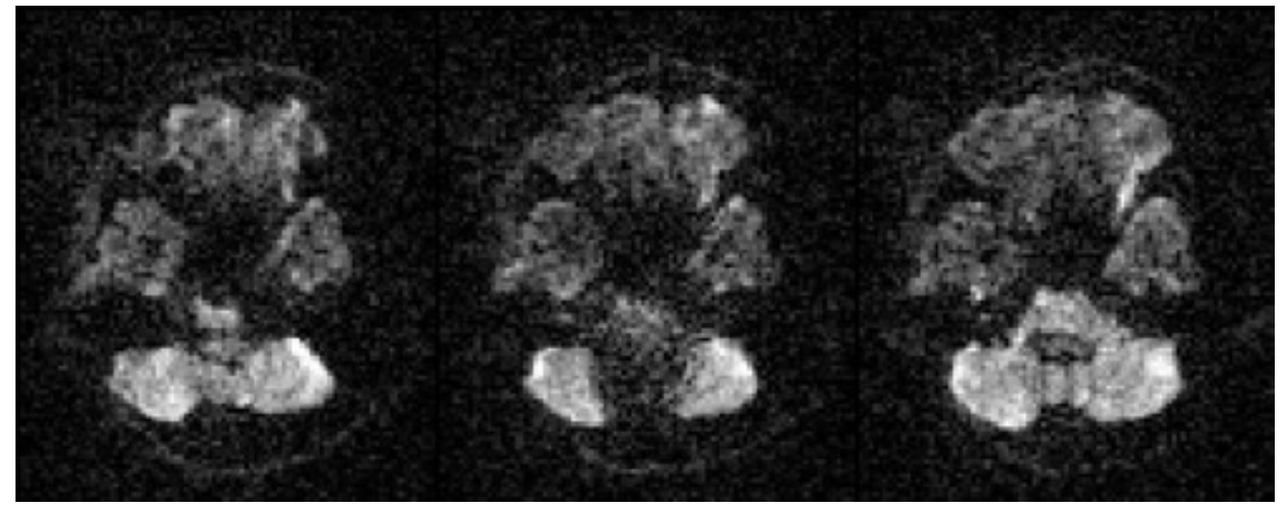
$$\hat{y}_{\mathbf{g}} = K(\mathbf{g}, \mathbf{G}) \left[K(\mathbf{G}, \mathbf{G}) + \sigma^2 \mathbf{I} \right]^{-1} \mathbf{y}$$

Outlier detection

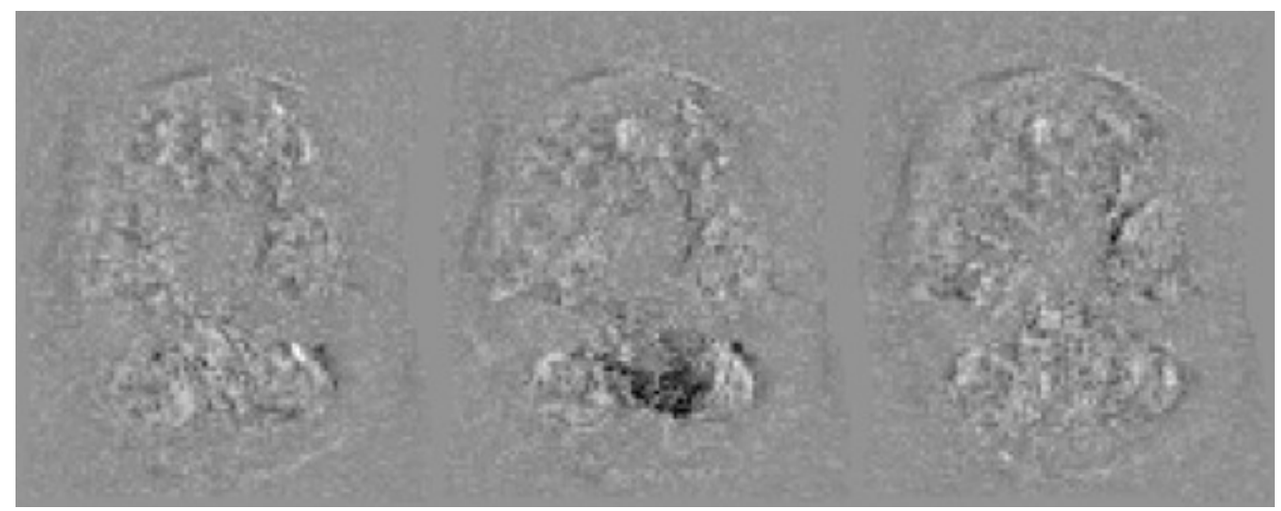


Remember that we do all comparisons in observation space.

Observed data



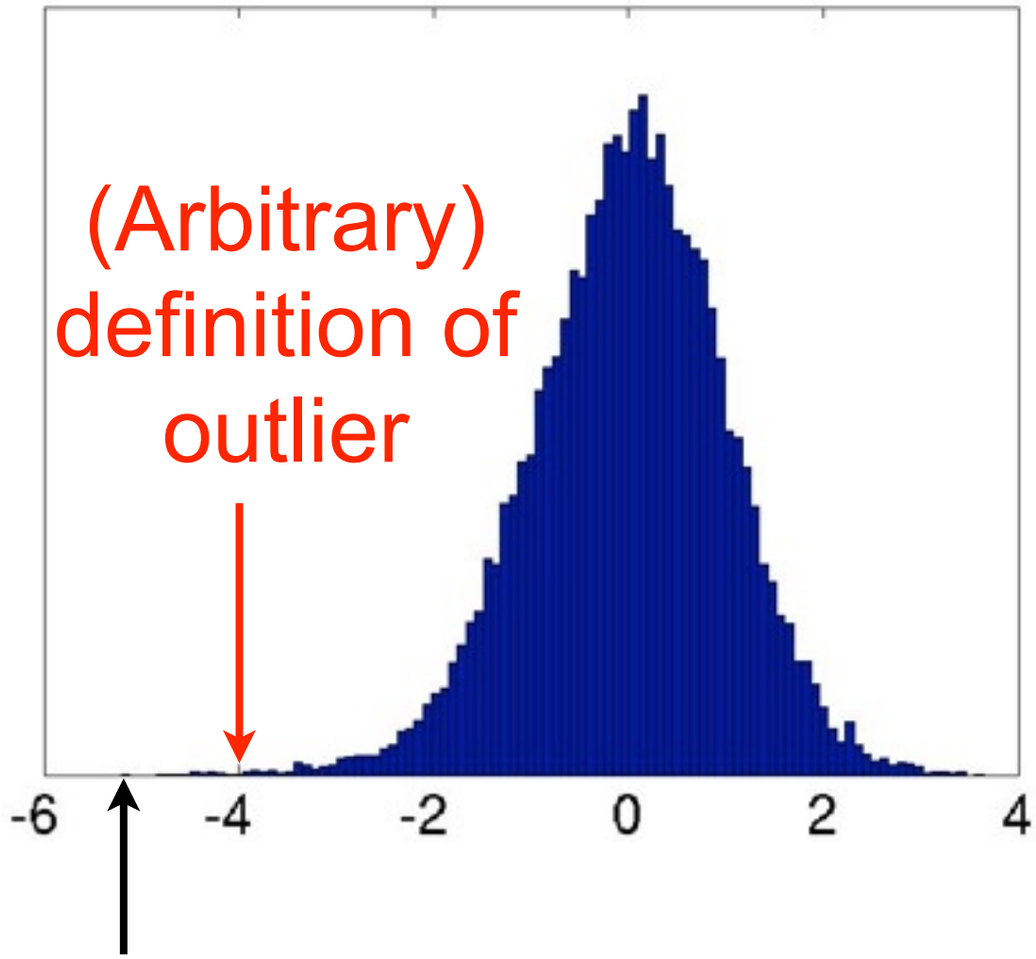
Observed - predicted



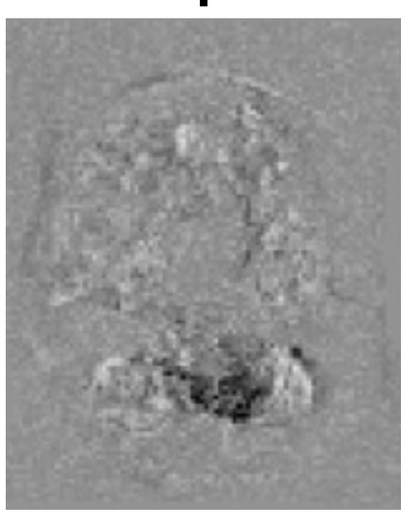
$\overline{x} = 0.084$ $\overline{x} = -0.791$ $\overline{x} = -0.125$

This allows us to calculate the per-slice mean difference between observation and prediction

Outlier detection



We can calculate the mean difference for every slice in every volume and get an empirical distribution that we can convert to z-scores

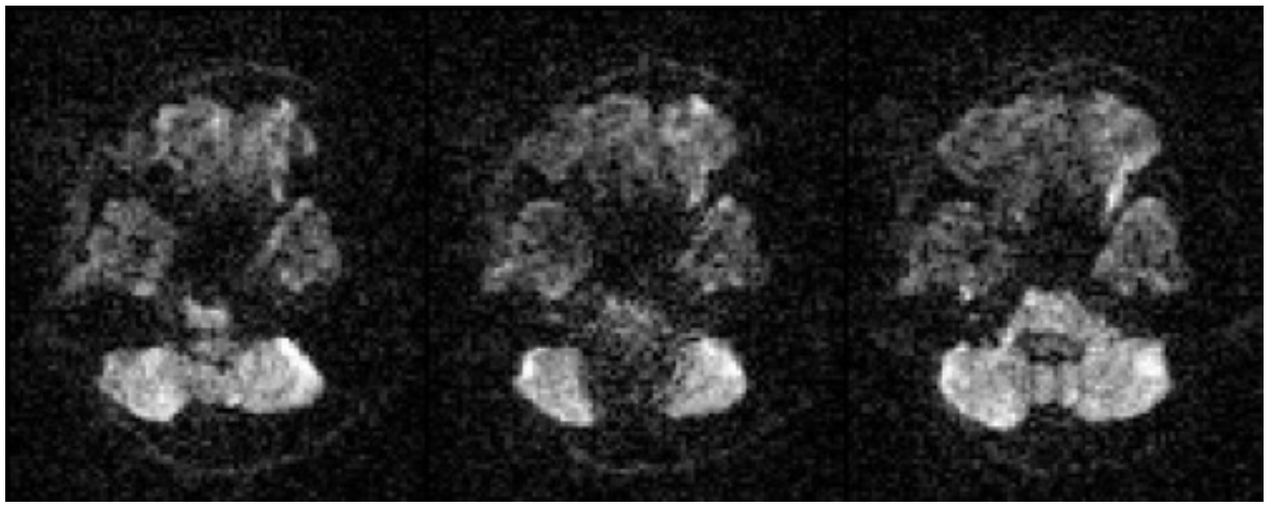


Worst slice

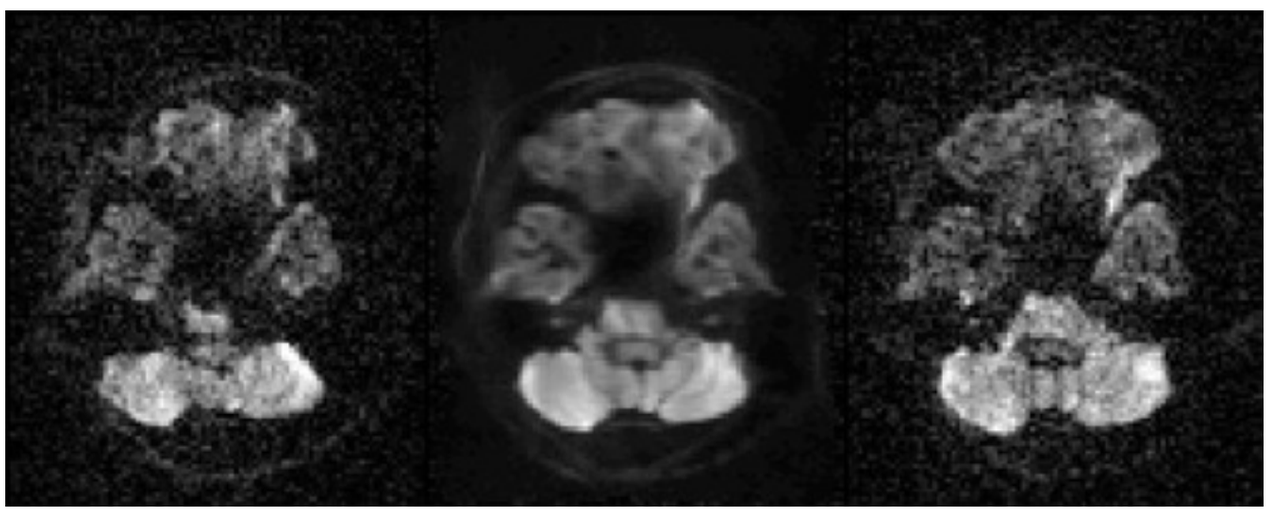
We can define an outlier slice as one with a z-score above an (arbitrary) threshold. We then have a choice of reporting outliers and/or replacing them with their predictions.

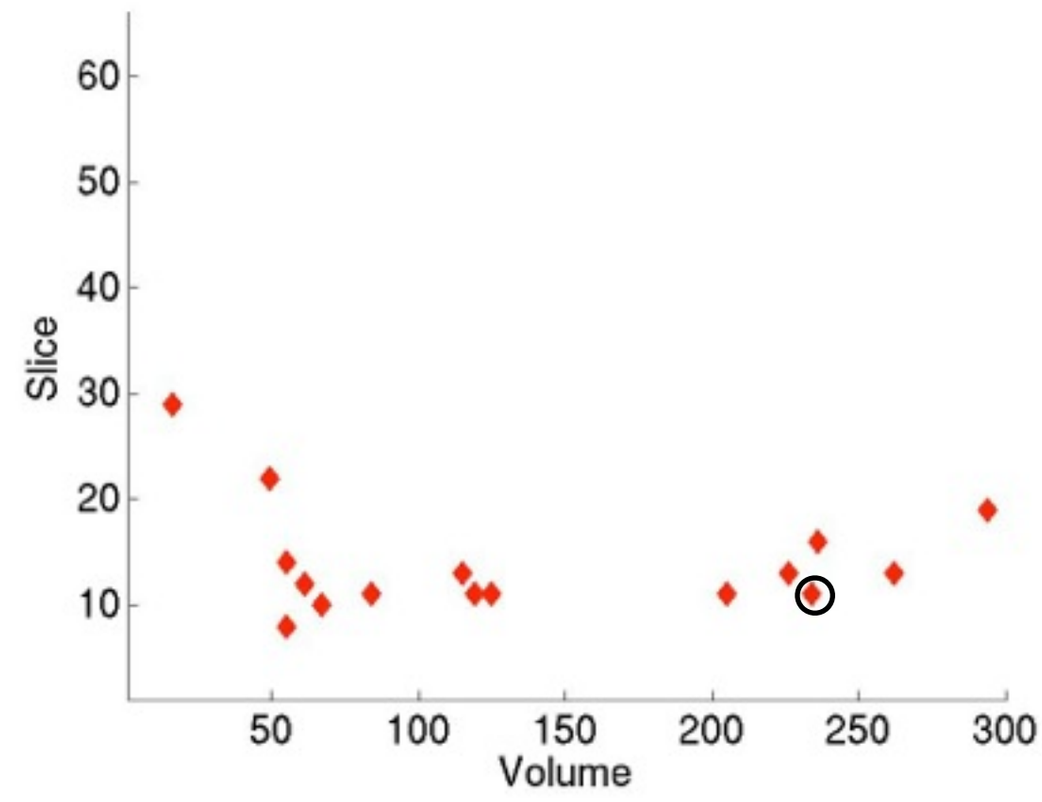
Outlier detection

Original data



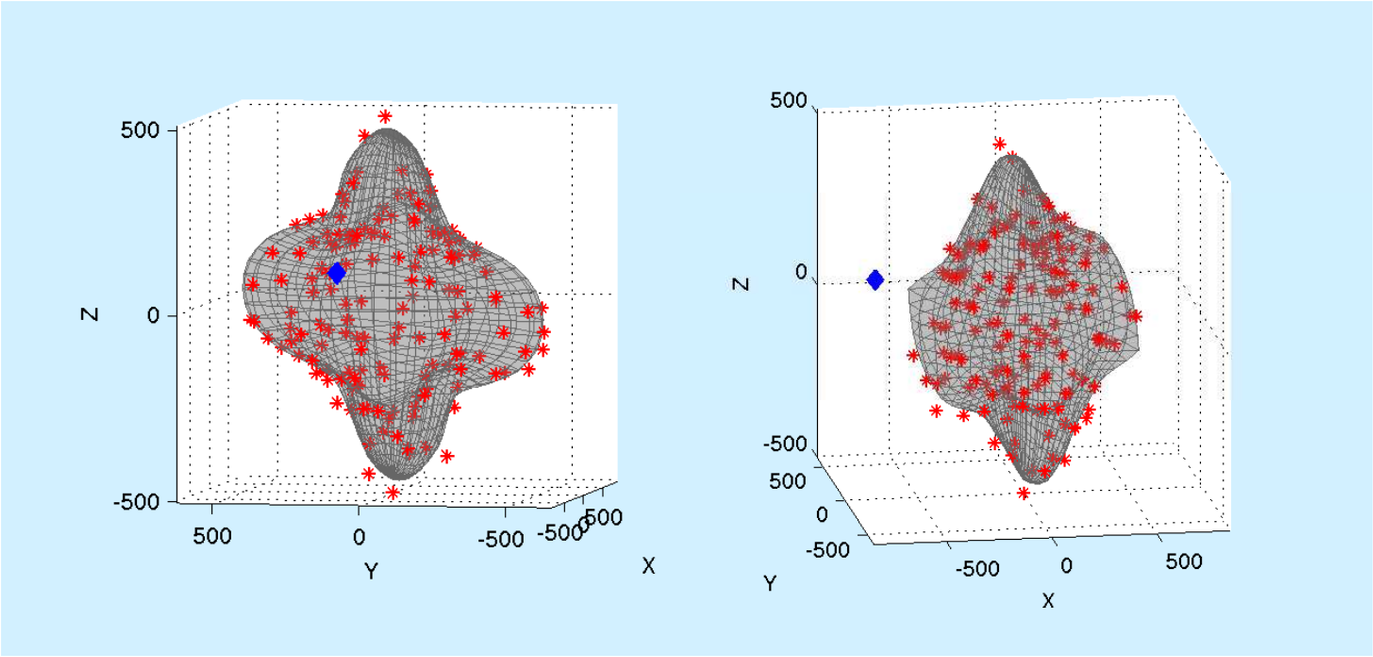
Data after replacement



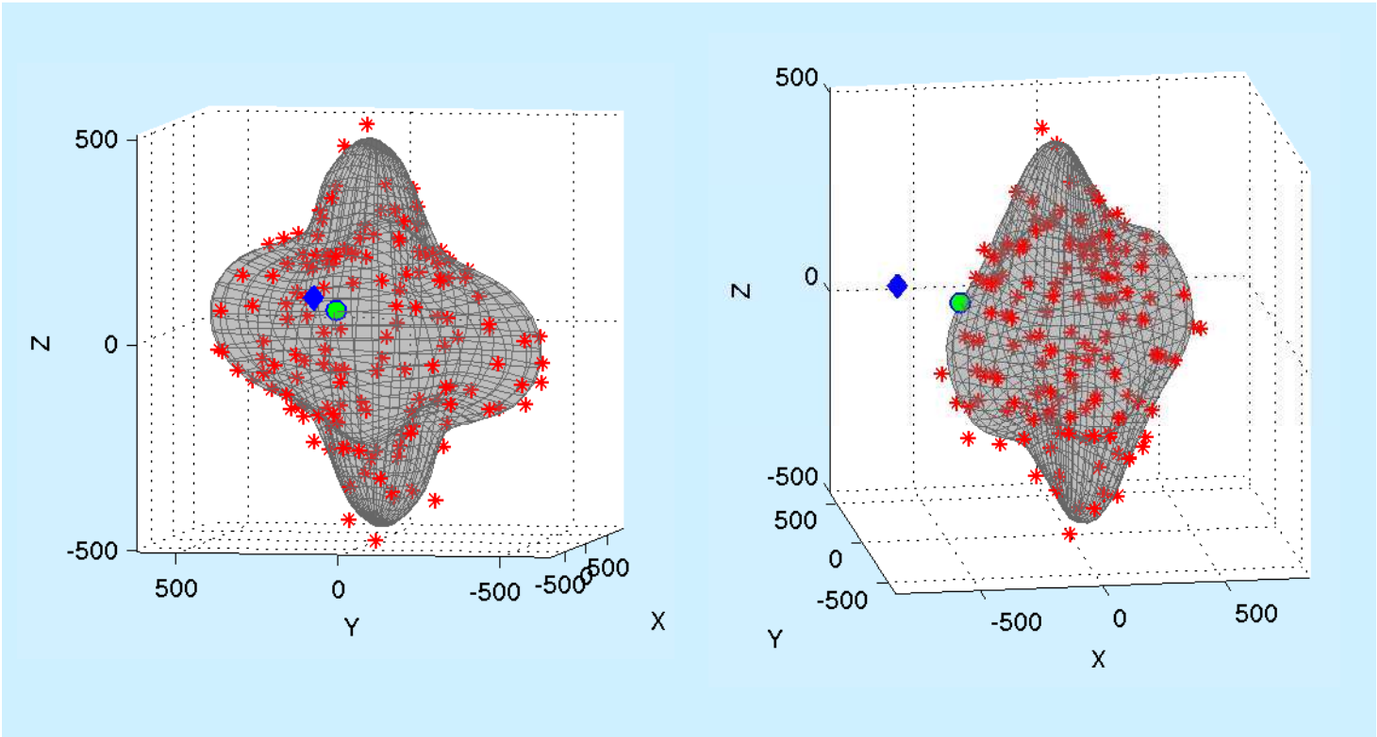


Outliers for a very still volunteer. Outliers mainly in basal slices.

How to make the “right” prediction



The outlier skews the predictions, but is still recognisable as an outlier

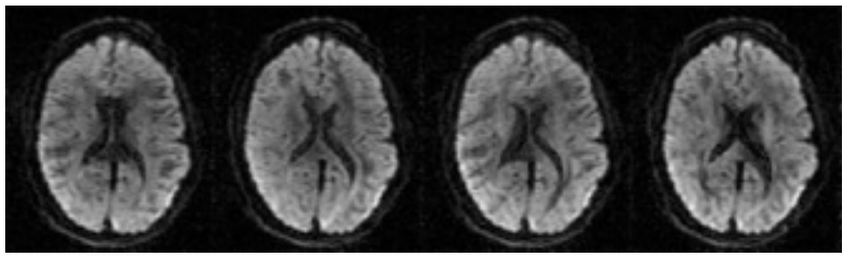


Remove the outlier and recalculate the “model”. The prediction is taken from this new “model”.

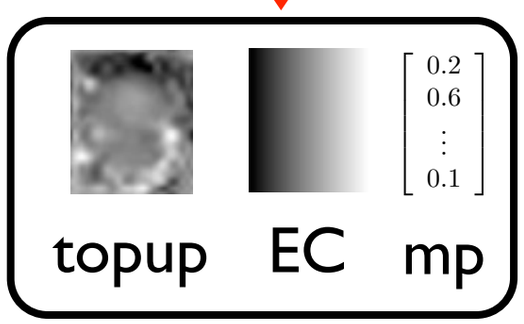
eddy revisited

1. **For all scans**

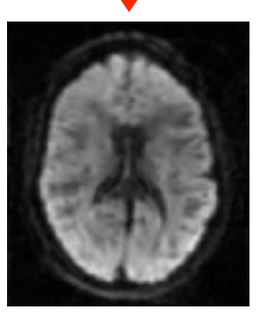
[1 0 0] [.6 -.4 -.7] [.8 .6 0] [-.4 .9 0]



…



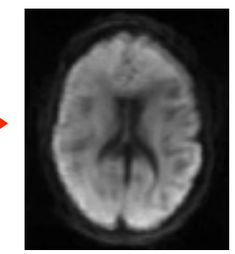
Use susceptibility field and current estimate of EC and movement to "unwarp" scan



Load into prediction maker

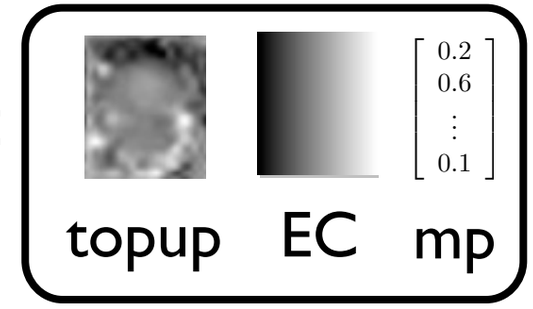
2. **For all scans**

[1 0 0]



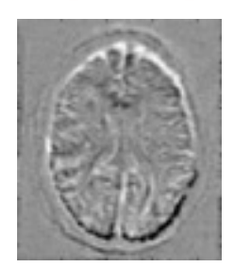
Get prediction

Invert current transform

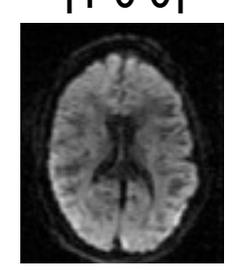


Use difference to update EC, mp and outlier list

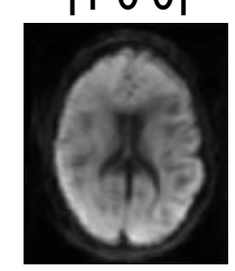
Get prediction in scan space



[1 0 0]

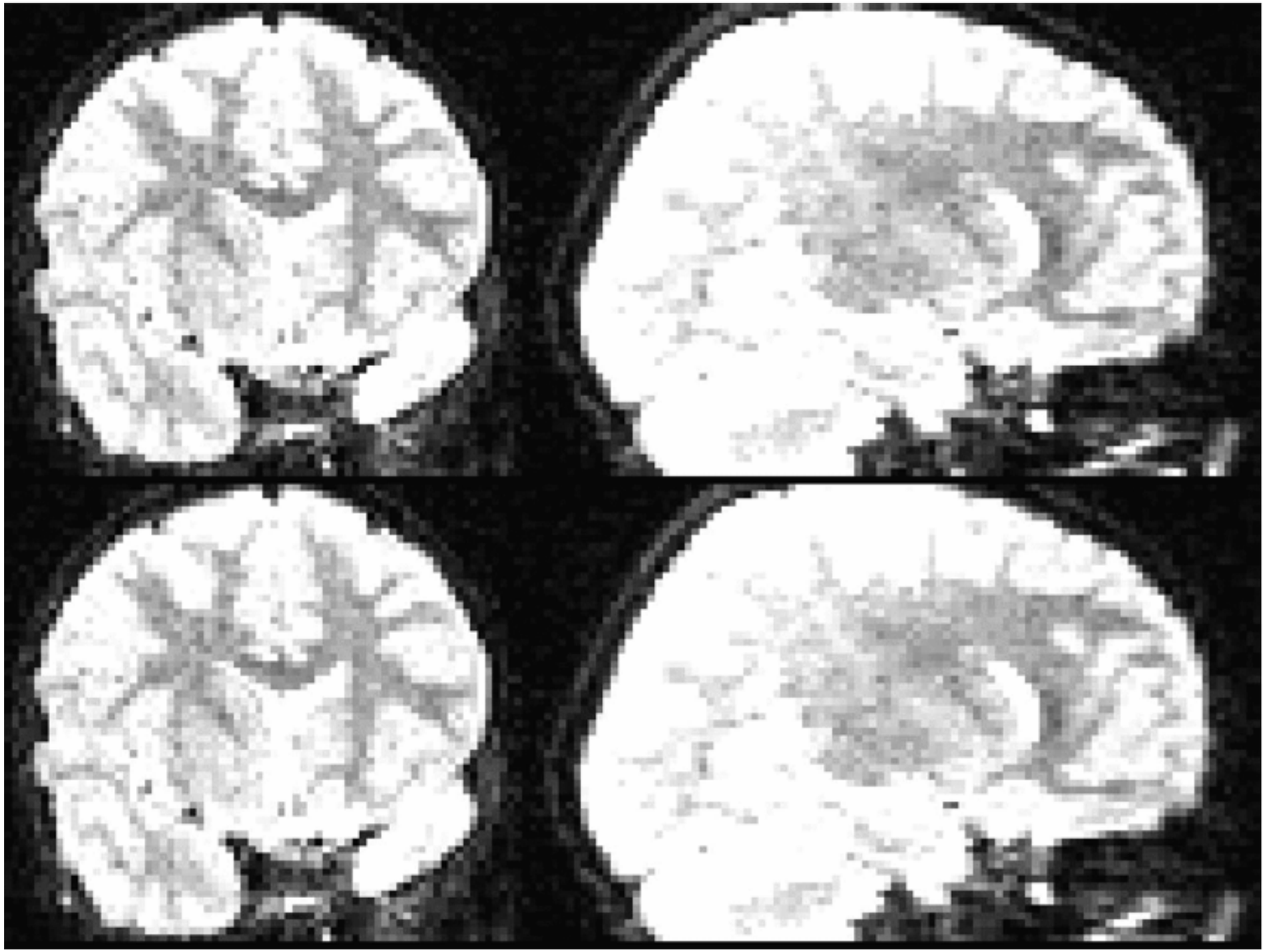


[1 0 0]



Compare to scan

Norwegian data. 32 directions. Hundreds of children.



Eight year old who gets tired towards the end of scanning

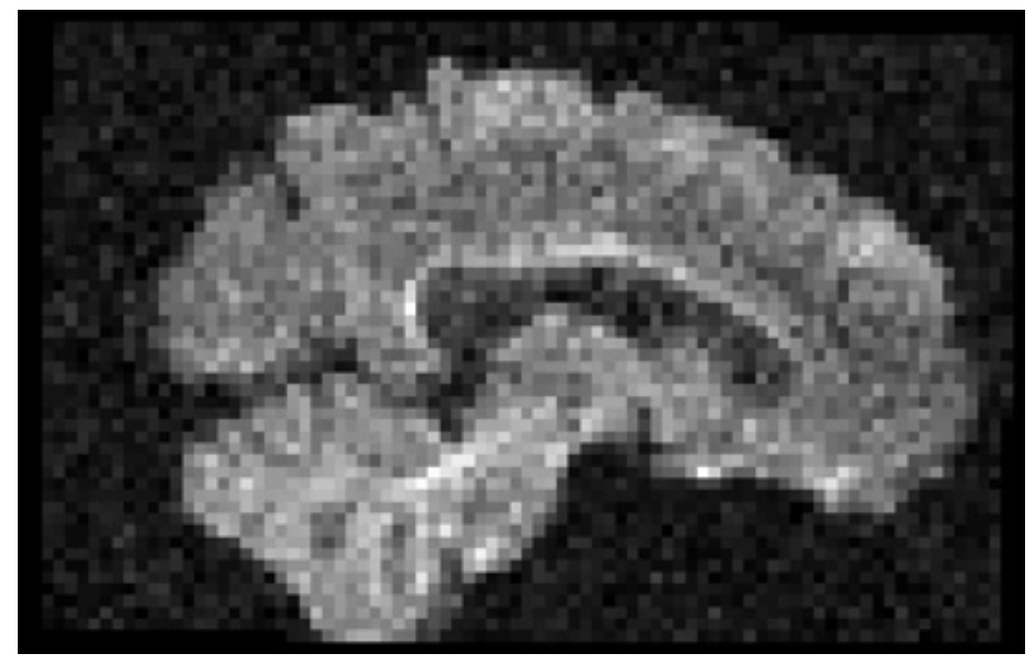
After outlier detection and replacement by eddy

How accurate are the predictions?

Simulated data

Simulations courtesy of Mark Graham, UCL.

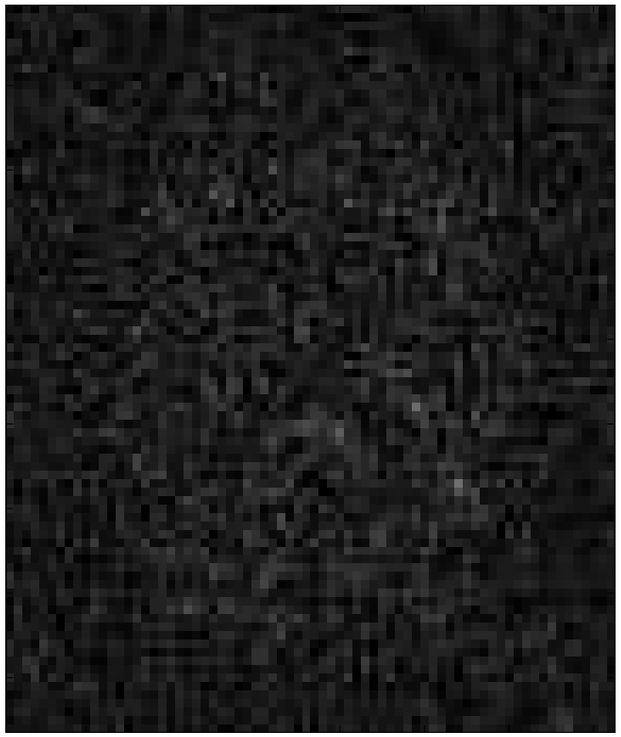
Before



After

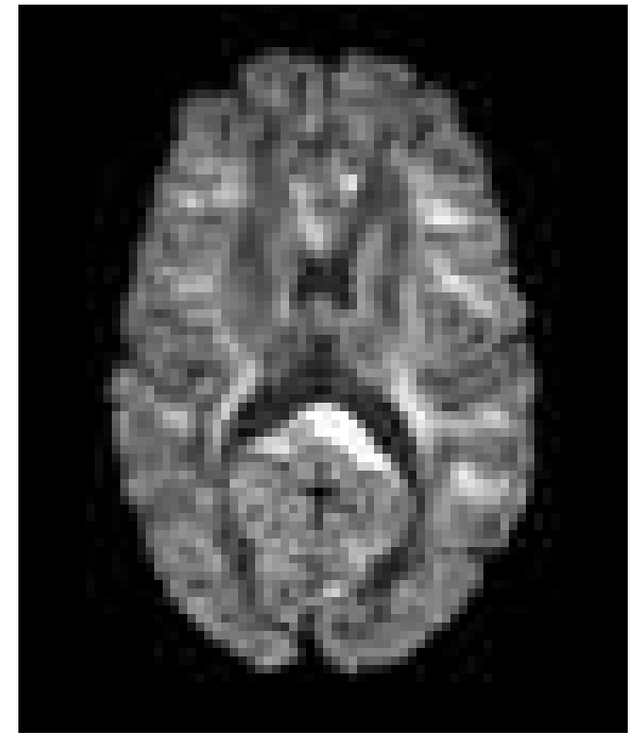
Looks good. But is it the “truth”?

How accurate are the predictions?



Outlier

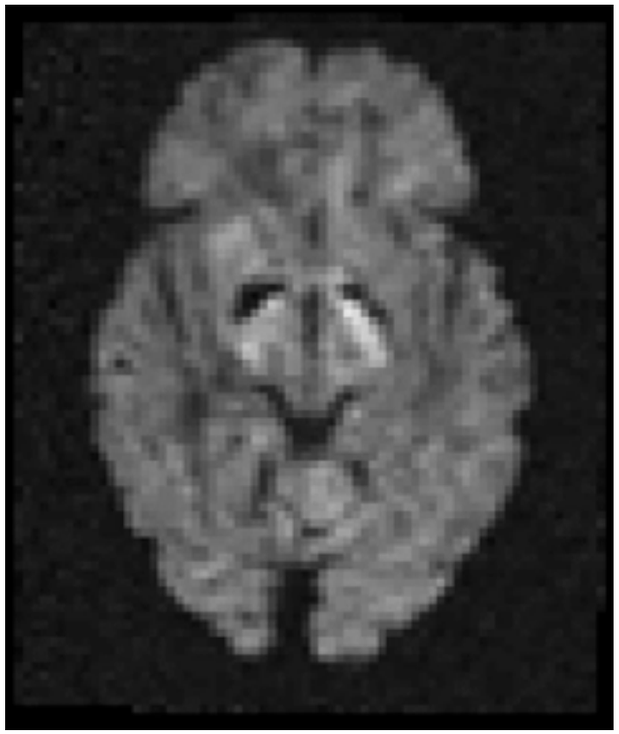
eddy's guesstimate



Ground truth

With the simulations we know the "ground truth"

How accurate are the predictions?



Slice 21

g=[.5 -.8 .4]

Slice 28

g=[-.3 -.7 -.7]

Slice 42

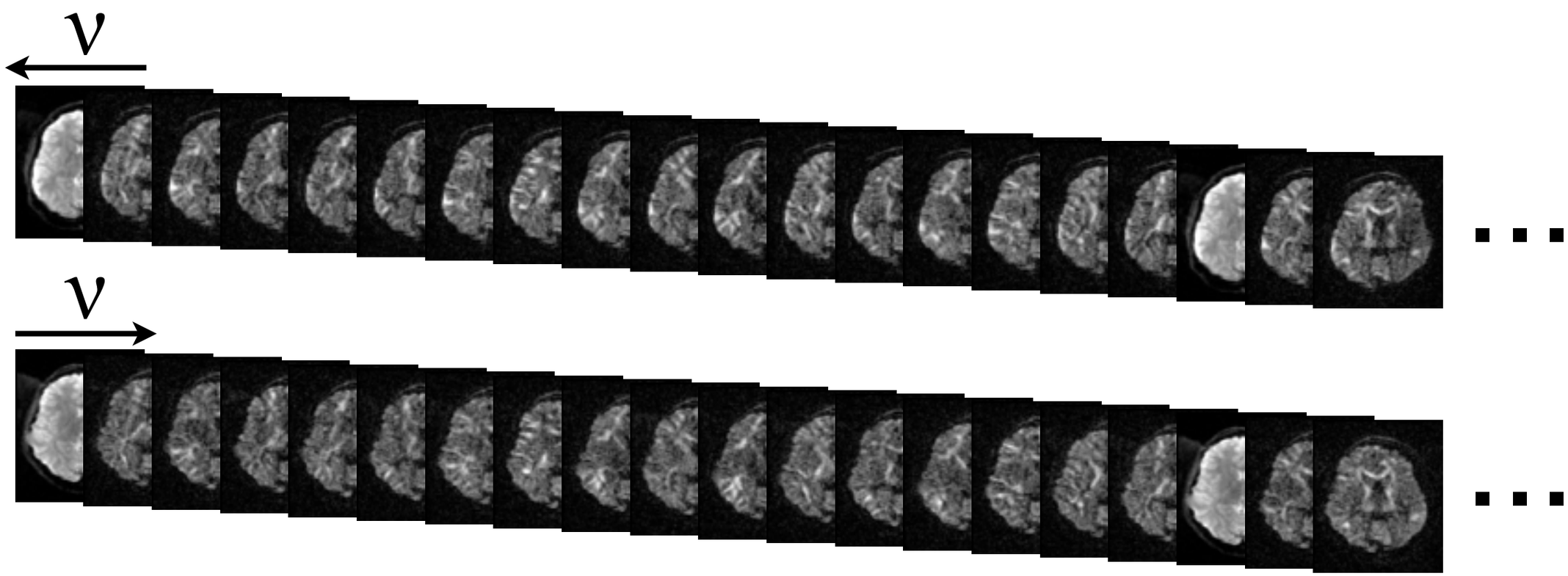
g=[.3 -.7 .6]

And some other slices/directions

Outline of the talk

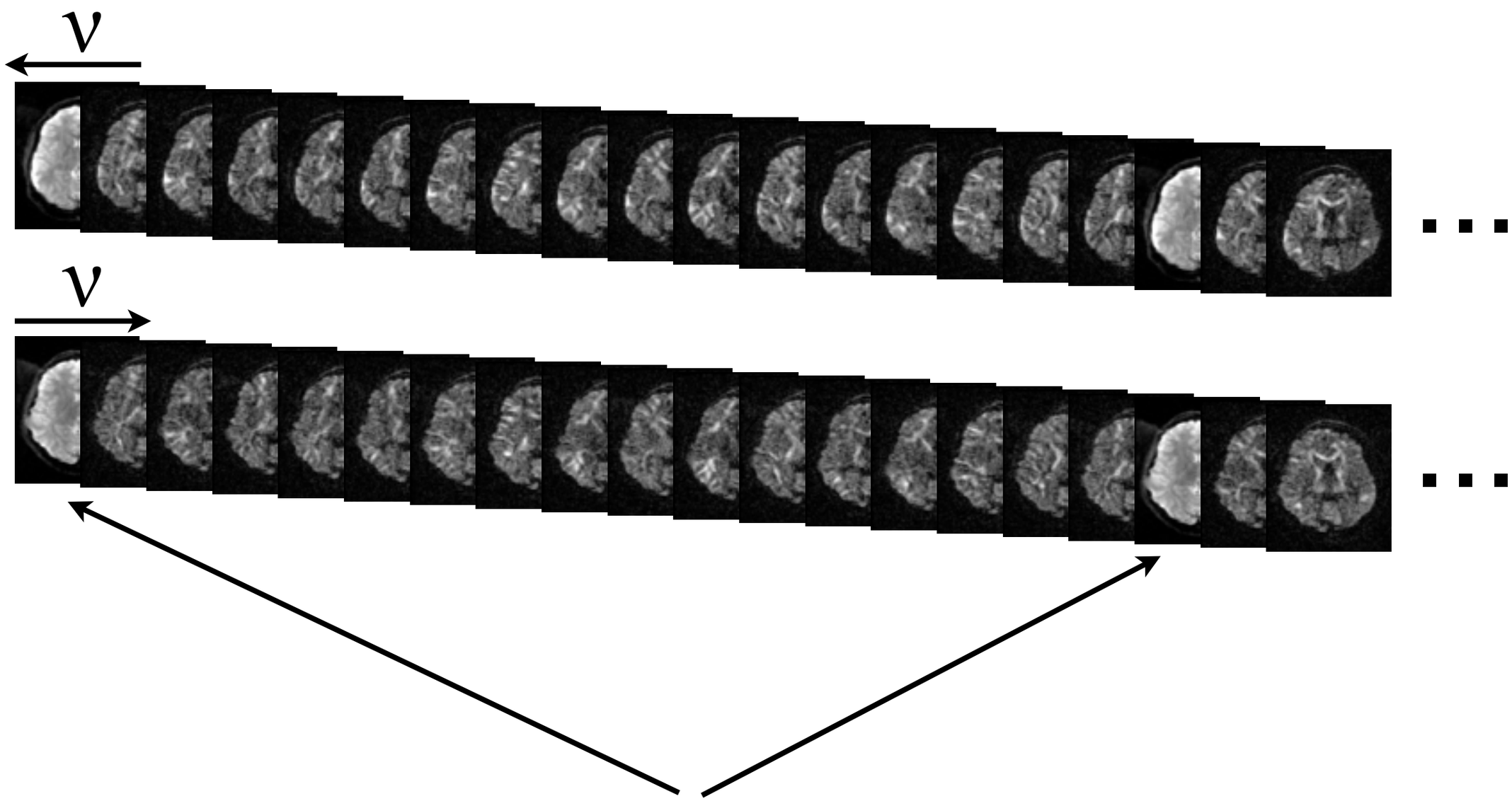
- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

Practicalities



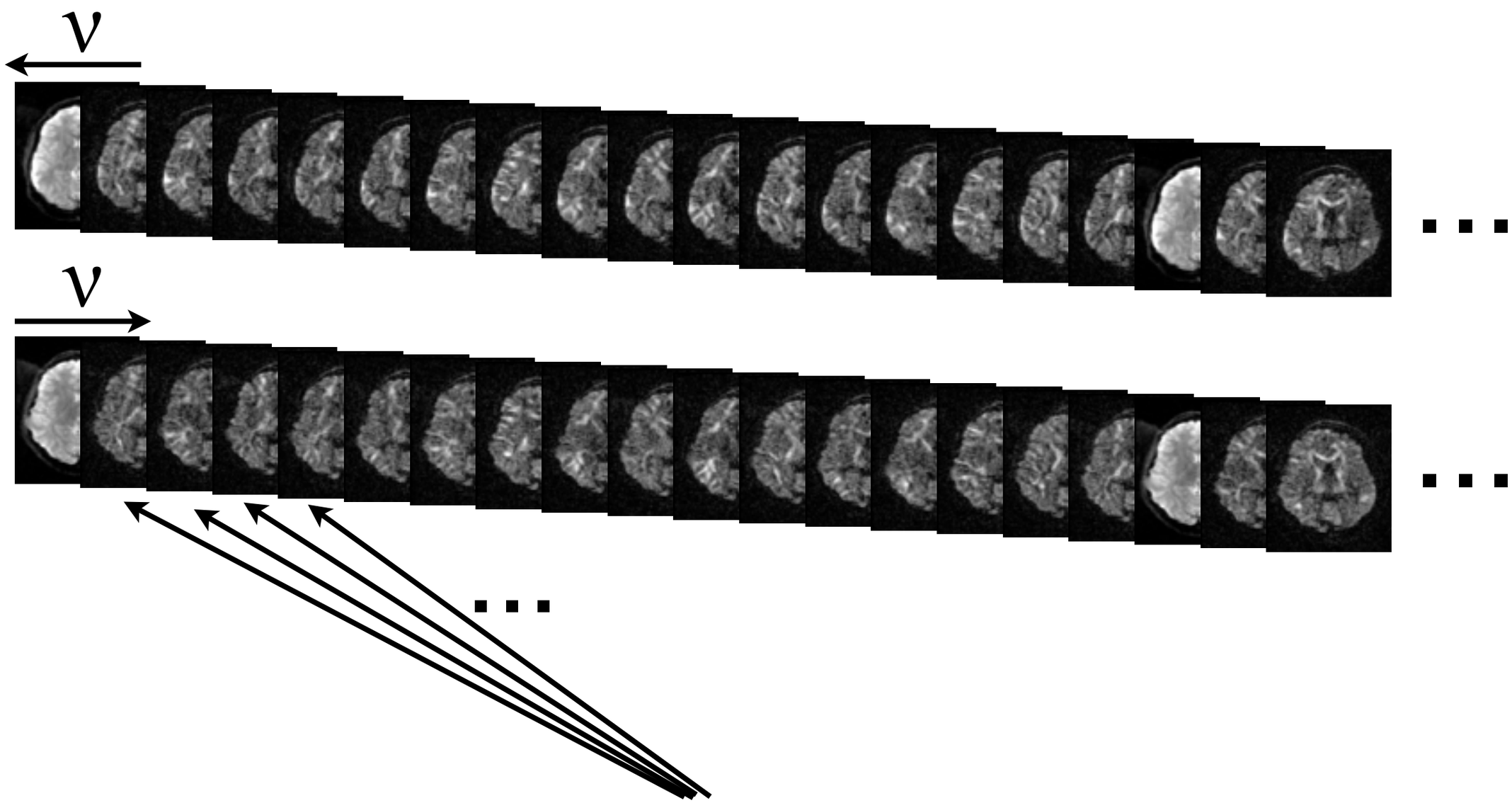
- Our example data consists of:
  - *N* diffusion weighted volumes and *n* *b*=0 volumes
  - *b*=0 volumes interspersed
  - Two repetitions, phase-encode L→R and R→L
  - Same diffusion table for both repetitions

Practicalities



Affected by susceptibility distortions

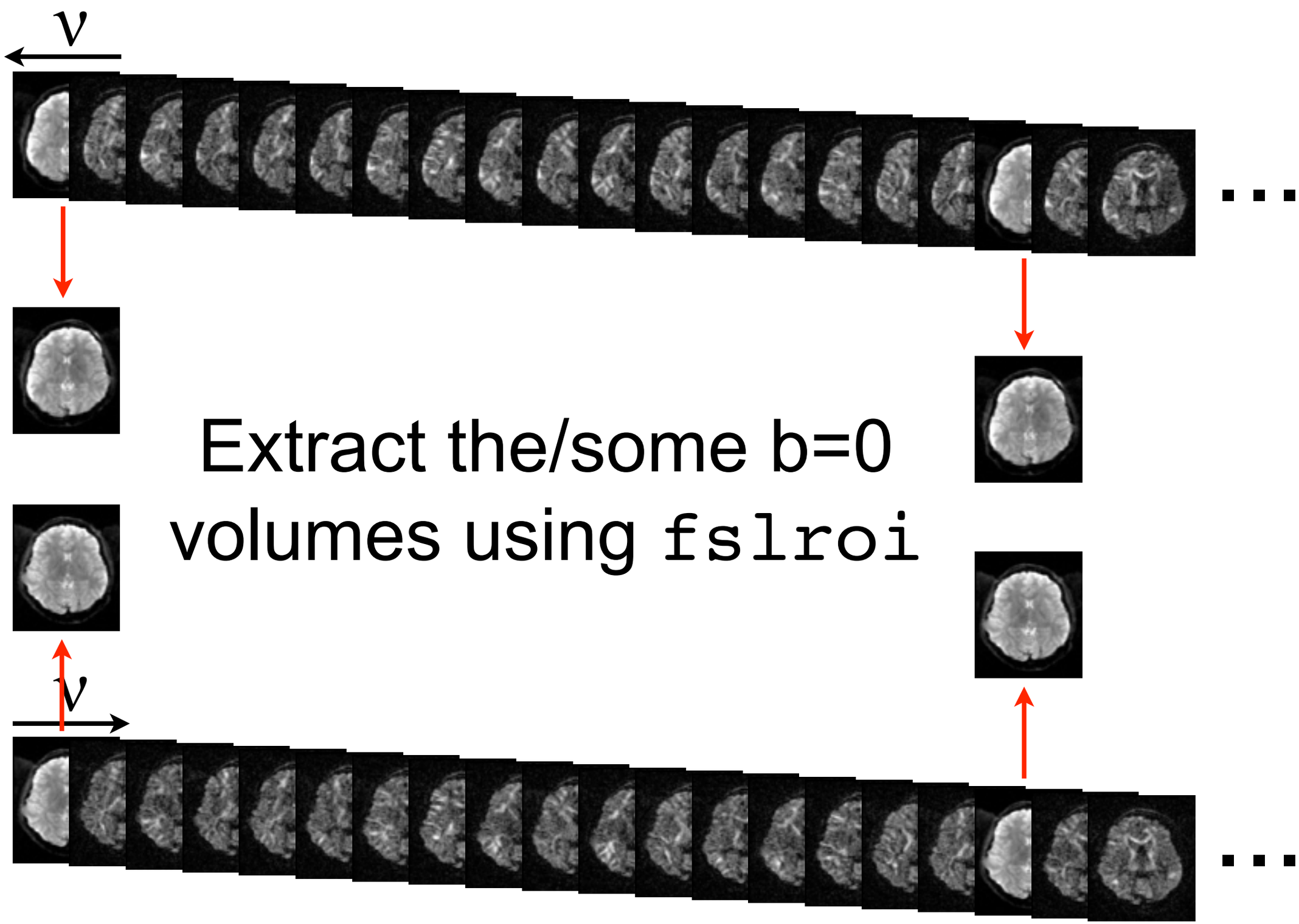
Practicalities



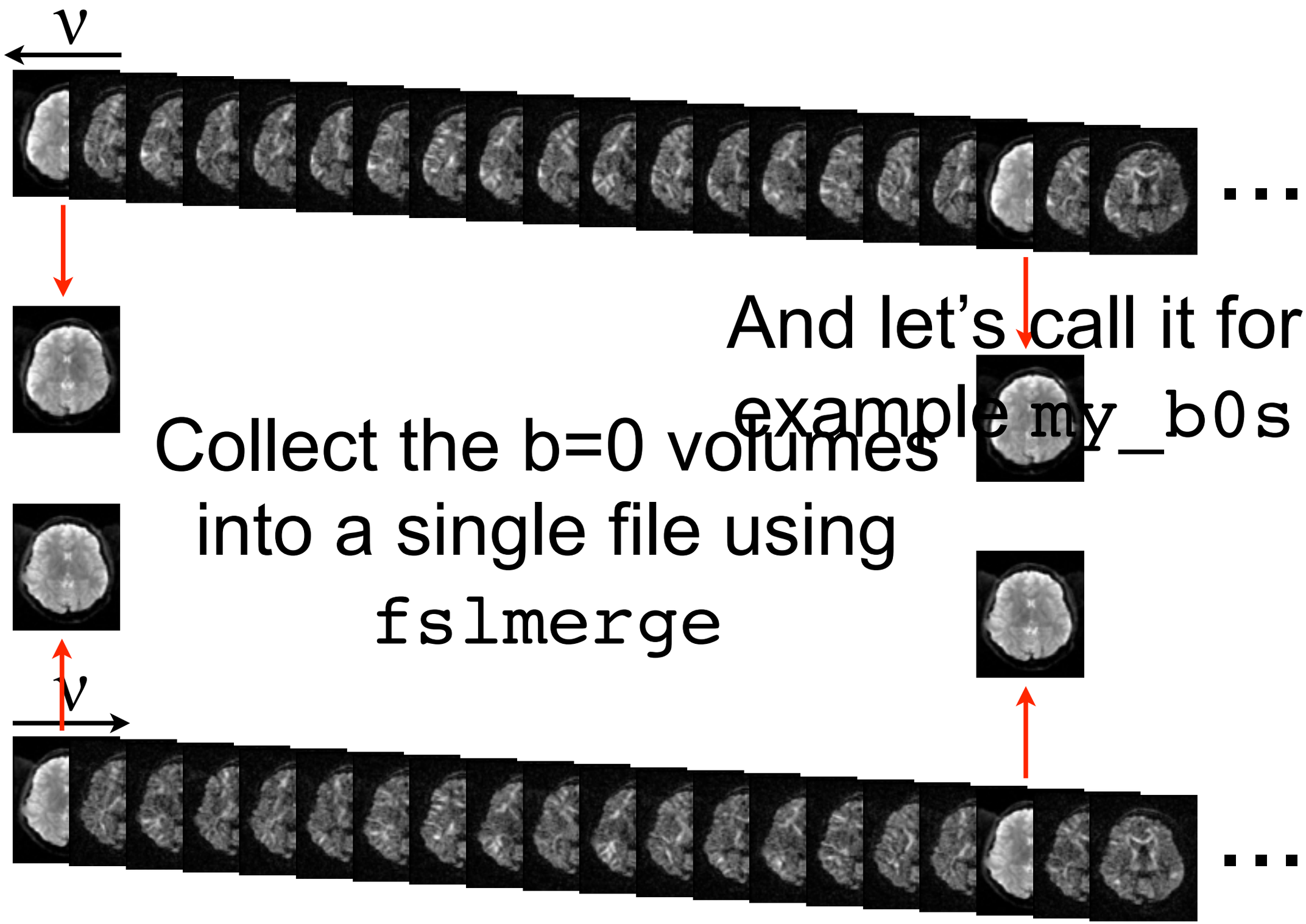
Affected by susceptibility distortions
AND eddy current distortions

And everything is of course affected by subject movement.

So, let's start with susceptibility

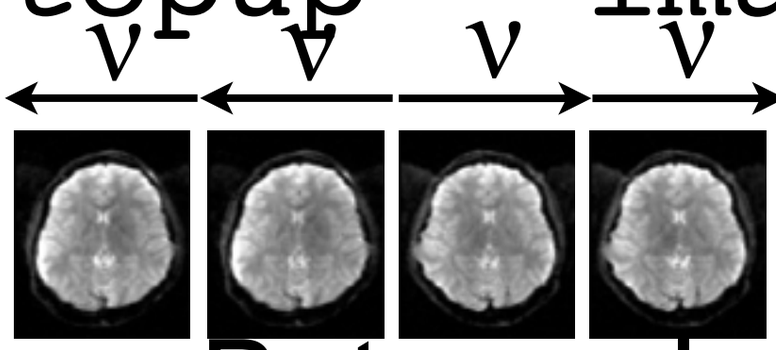


So, let's start with susceptibility



And the tool for that is `topup`

```
topup --imain=my_b0s
```



`my_b0s`

But we also need to inform topup about the acquisition parameters

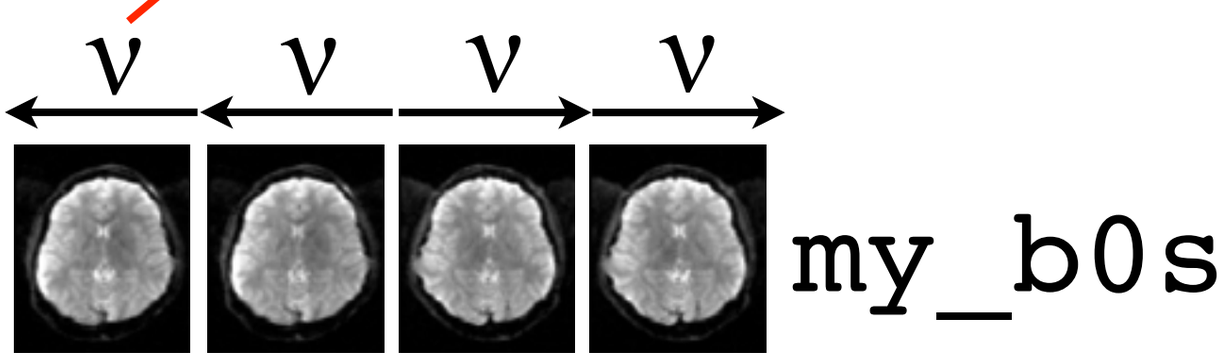
And the tool for that is `topup`

```
topup --imain=my_b0s
```

Means PE in x-direction, L→R

```
-1 0 0 0.051
```

Total readout time (in seconds)

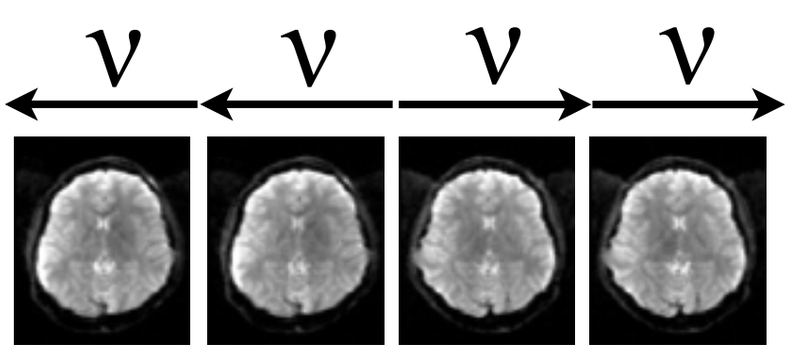


And the tool for that is `topup`

```
topup --imain=my_b0s --datain=acqparams.txt
```

```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

Text file that we can call for example `acqparams.txt`

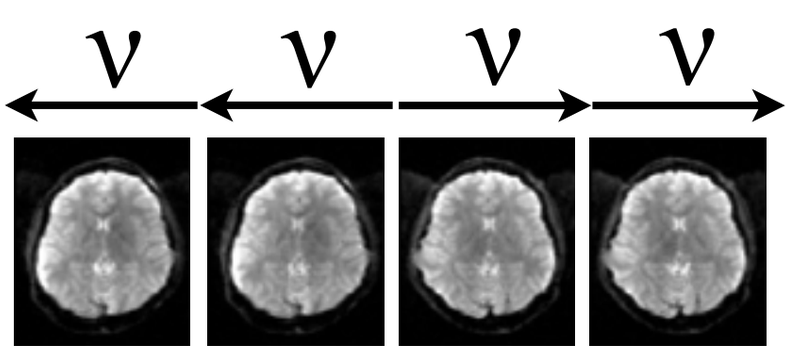


`my_b0s`

And the tool for that is `topup`

```
topup --imain=my_b0s --datain=acqparams.txt --config=b02b0.cnf
```

And then some technical details



`my_b0s`

```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

`acqparams.txt`

And the tool for that is `topup`

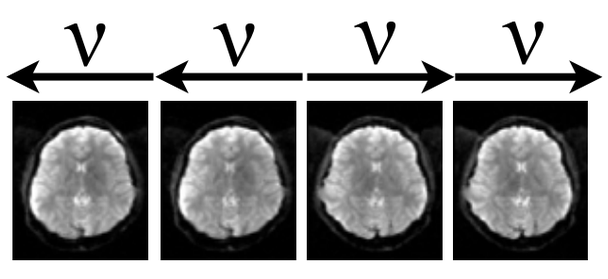
And finally we need to tell it where to put the results

```
topup --imain=my_b0s --datain=acqparams.txt --config=b02b0.cnf  --out=my_topup
```

my_topup_movpar.txt

```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```

Tells position of 2nd b=0 scan relative the first



my_b0s

```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

acqparams.txt

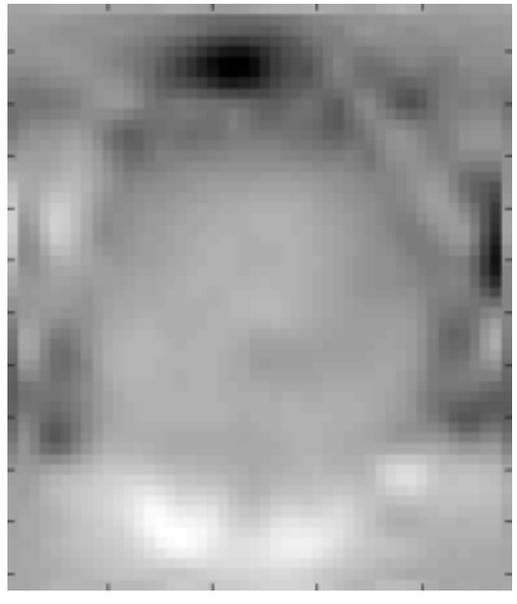
b02b0.cnf

And the tool for that is `topup`

And finally we need to tell it where to put the results

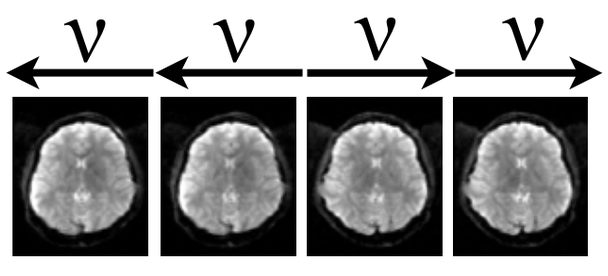
```
topup --imain=my_b0s --datain=acqparams.txt --config=b02b0.cnf --out=my_topup
```

my_topup_fieldcoef.nii



my_topup_movpar.txt

```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```



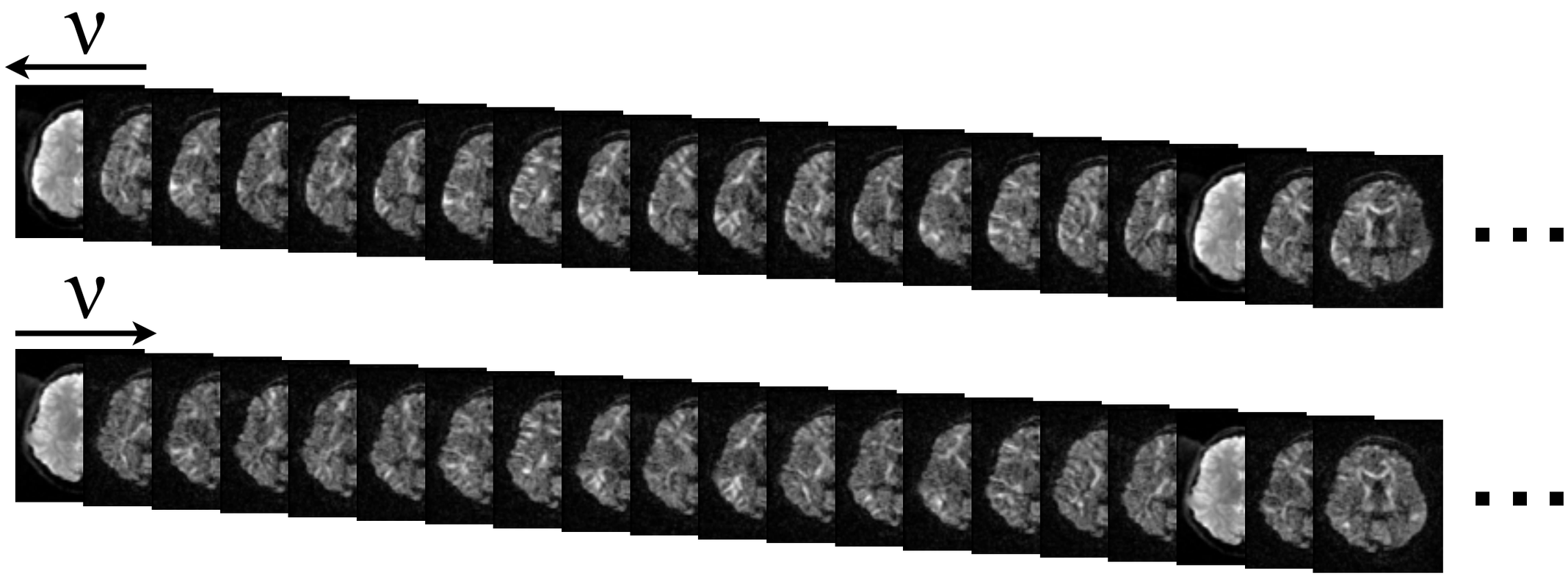
my_b0s

```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

acqparams.txt

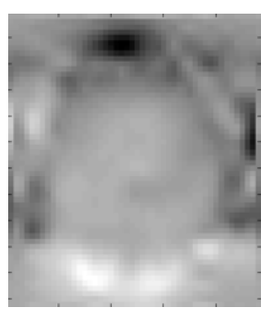
b02b0.cnf

Back to the full data-set



Now we want to correct the eddy current-distortions and subject movement in the whole data set.

my_topup_fieldcoef.nii



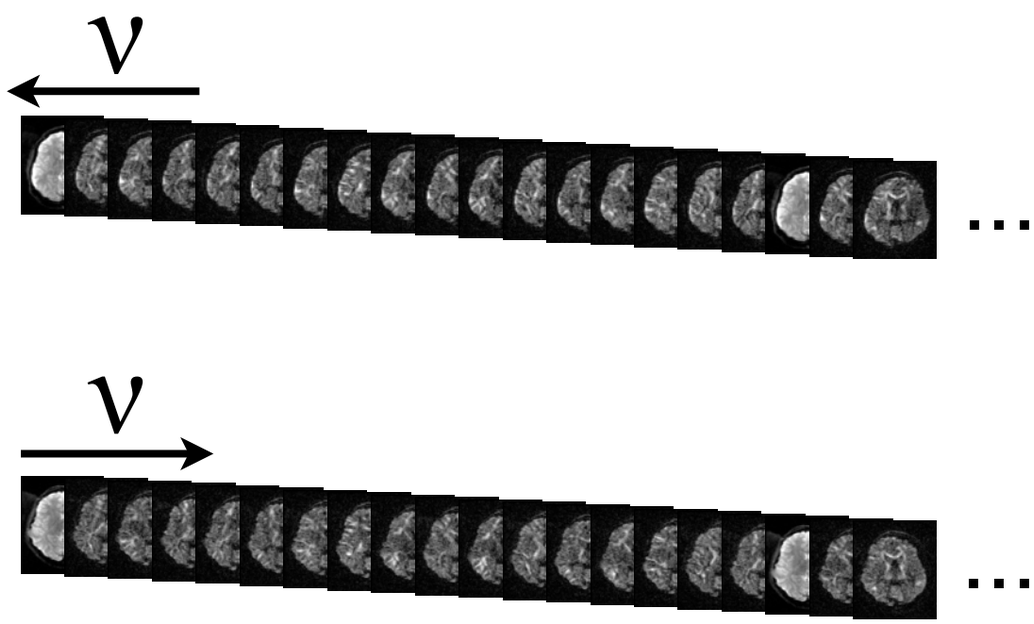
```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

acqparams.txt

```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```

my_topup_movpar.txt

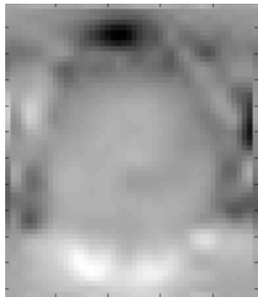
Collect all data in one file



LR_RL

The first thing we do is to collect all data in a single file using `fslmerge` and call it for example `LR_RL`

my_topup_fieldcoef.nii



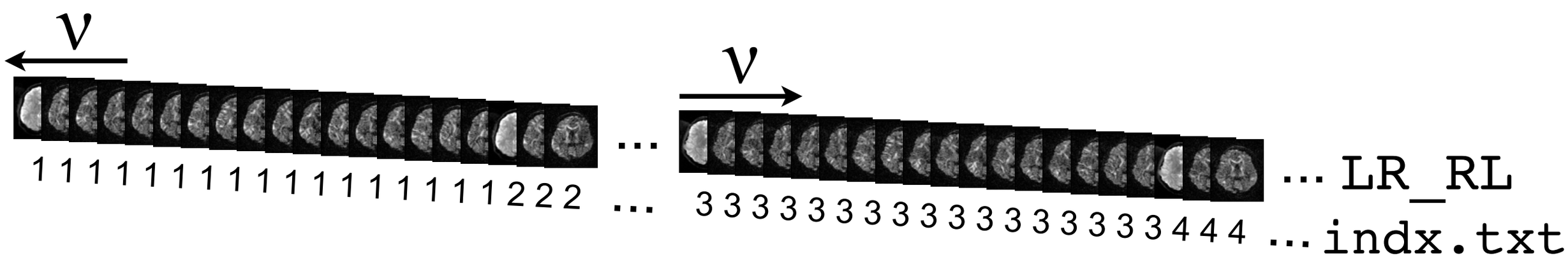
```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

acqparams.txt

```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```

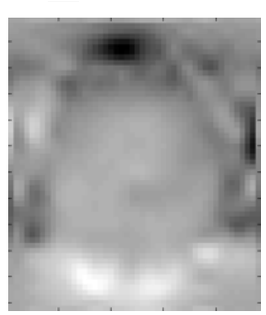
my_topup_movpar.txt

Inform eddy of acquisition parameters



Then we make a text file with one index for each volume, and call it for example `indx.txt`

my_topup_fieldcoef.nii



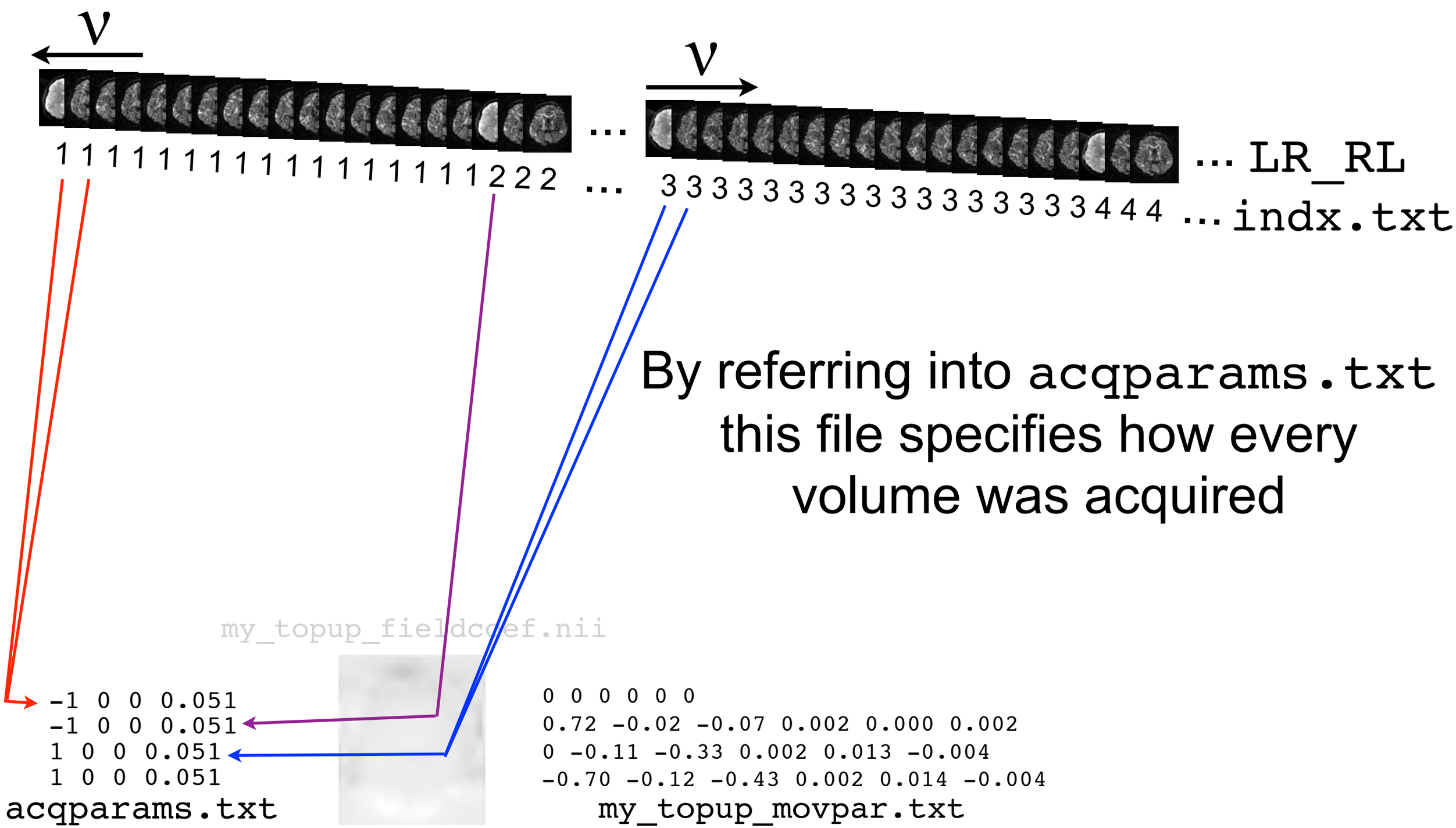
```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

acqparams.txt

```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```

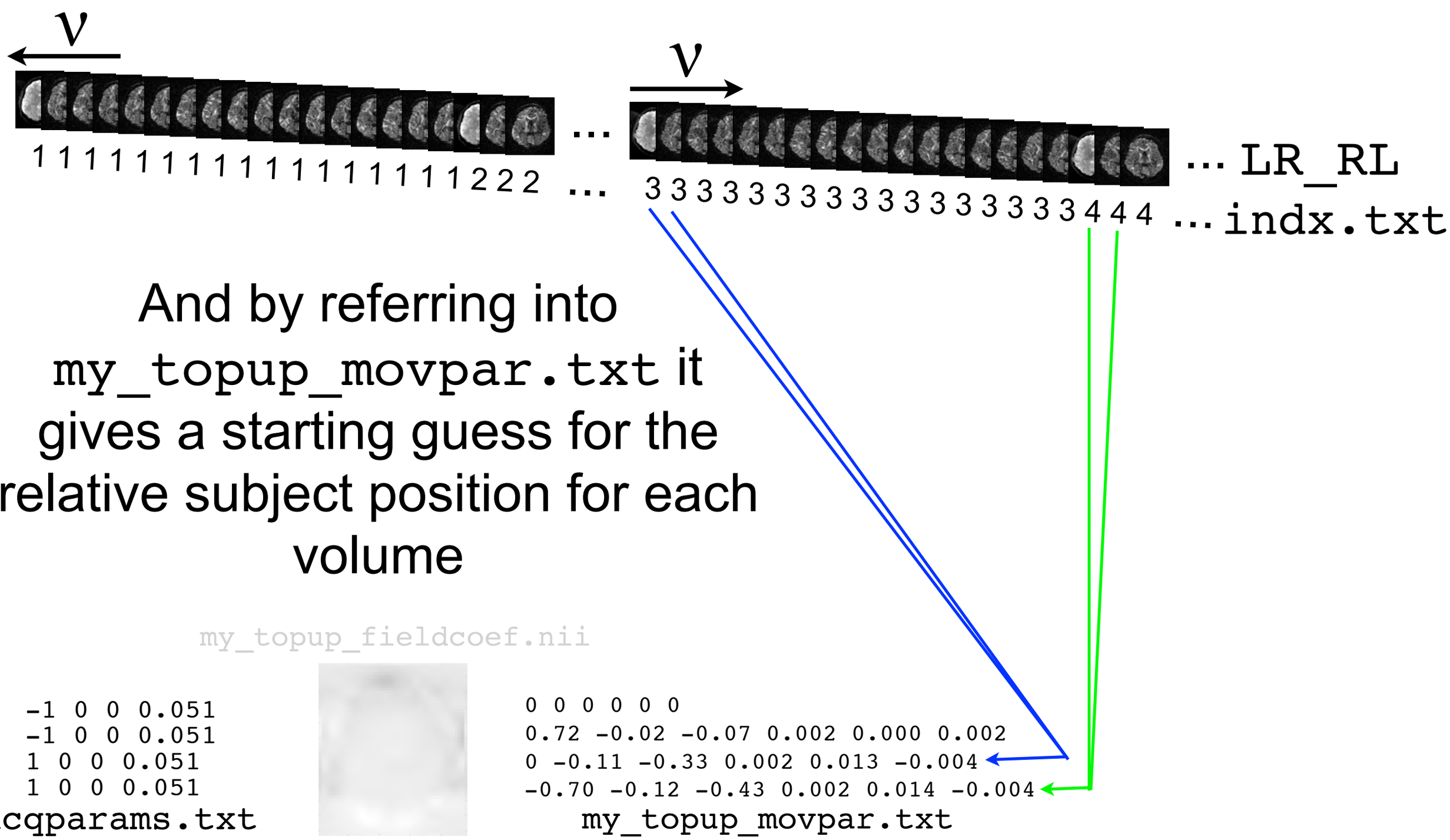
my_topup_movpar.txt

Inform eddy of acquisition parameters



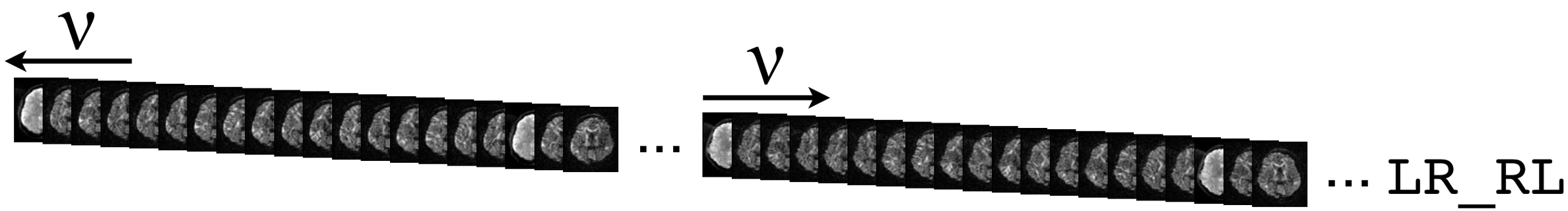
By referring into `acqparams.txt` this file specifies how every volume was acquired

Inform eddy of acquisition parameters



And by referring into `my_topup_movpar.txt` it gives a starting guess for the relative subject position for each volume

And of diffusion parameters



0 1000 1000 1000 ...

↑ ↗ ↘ ...

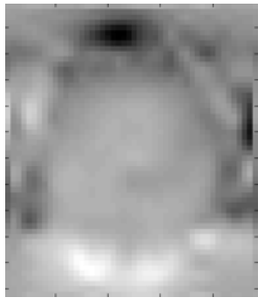
`bvals`

`bvecs`

text-files

And we also need to know the b-value and b-vector for each volume (same as for `dtifit` or `bedpost`).

`my_topup_fieldcoef.nii`



```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

`acqparams.txt`

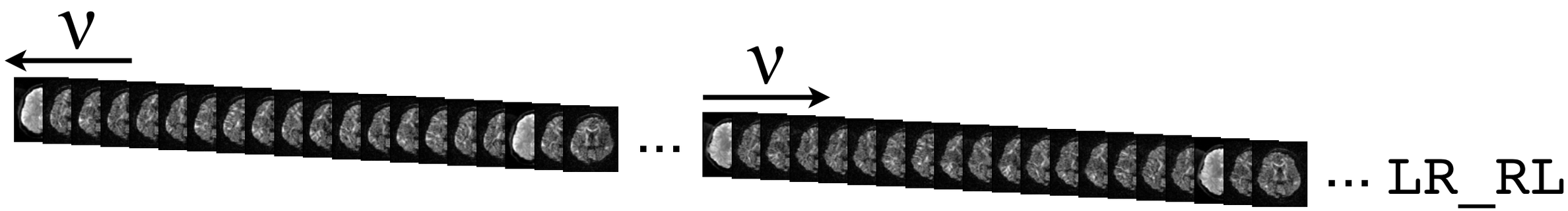
```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```

`my_topup_movpar.txt`

```
1111...
```

`indx.txt`

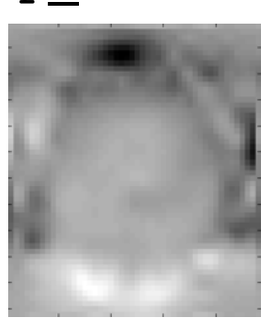
And where the brain is



brain_mask.nii

And finally a binary mask that tells eddy which voxels are brain. Also the same that is used for `dtifit/bedpost`.

my_topup_fieldcoef.nii



```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

acqparams.txt

```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```

my_topup_movpar.txt

```
1111...
```

indx.txt

```
0 1000 1000 1000 ...
```

bvals

↑ ↗ ↘ ...

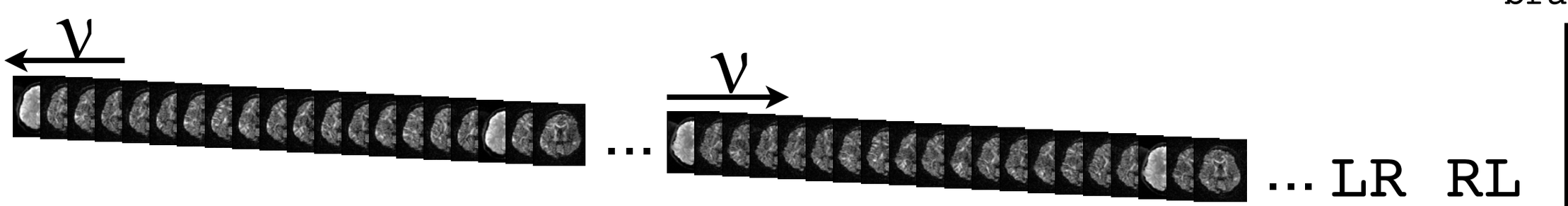
bvecs

And now we can run eddy

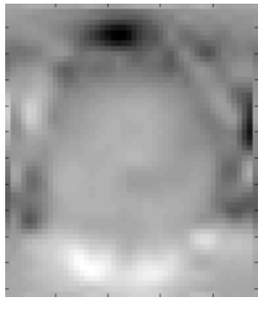
```
eddy --imain=LR_RL --acqp=acqparams.txt
--index=indx.txt --bvecs=bvecs
--bvals=bvals --mask=brain_mask
--topup=my_topup --out=my_eddy
```

And now we are ready for the most horrible command line in all of `fsl`

brain_mask.nii



my_topup_fieldcoef.nii



```
-1 0 0 0.051
-1 0 0 0.051
1 0 0 0.051
1 0 0 0.051
```

acqparams.txt

```
0 0 0 0 0 0
0.72 -0.02 -0.07 0.002 0.000 0.002
0 -0.11 -0.33 0.002 0.013 -0.004
-0.70 -0.12 -0.43 0.002 0.014 -0.004
```

my_topup_movpar.txt

```
1111...
```

indx.txt

```
0 1000 1000 1000 ...
```

bvals

↑ ↗ ↘ ...

bvecs

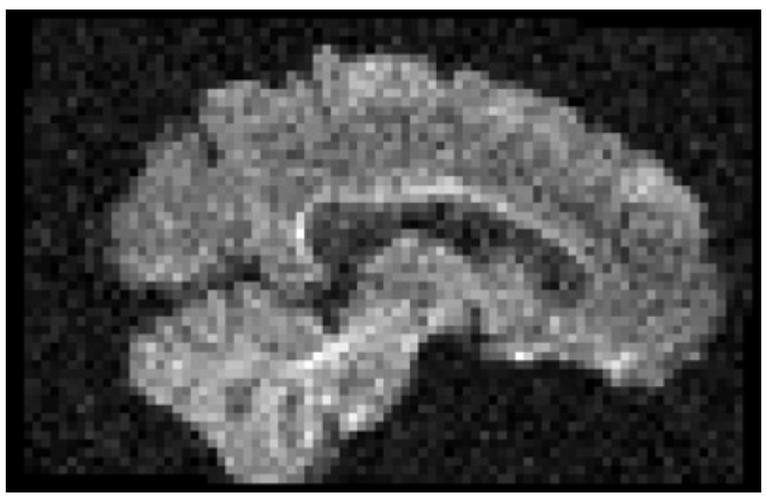
Outline of the talk

- What is the problem with diffusion data?
- Off-resonance field <-> Distortions
- Where does the off-resonance field come from?
- Worlds shortest course on image registration
- How topup works
- How eddy works
- Outliers
- Practicalities
- Output

Output

```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```

LR_RL.nii.gz



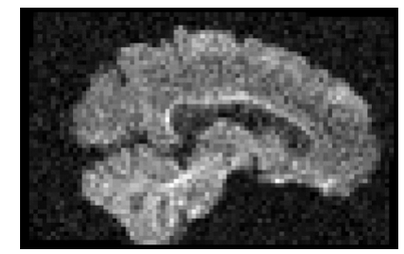
my_eddy.nii.gz

This is the “main” output, and chances are you will never need anything else.

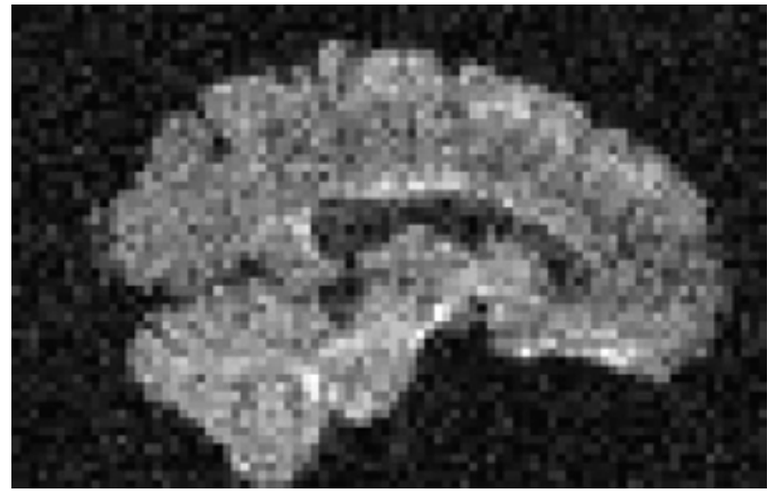
Output

```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```

LR_RL.nii.gz



my_eddy.nii.gz



my_eddy.eddy_outlier_free_data.nii.gz

This has been corrected only for outliers.
Distortions and movements have not been corrected.
Unlikely to be of interest.

Output

```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```

LR_RL.nii.gz

my_eddy.nii.gz

```
0          0          0          0          0          ...
0.114     -0.014     -0.126     -0.003      0.004      ...
0.190      0.050     -0.131     -0.003      0.006      ...
0.177      0.112     -0.184     -0.002      0.009      ...
.
.
.
```

my_eddy.eddy_parameters

This is a text-file with the estimated movement and EC parameters. Unlikely to be of interest.

Output

```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```

LR_RL.nii.gz

my_eddy.nii.gz

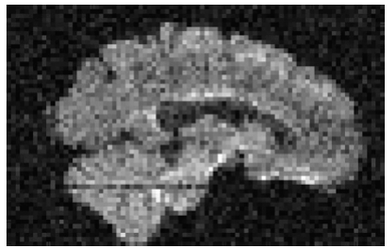
Cumulative	Relative preceding volume
0	0
0.299	0.299
0.401	0.148
0.491	0.136
.	.
.	.
.	.

my_eddy.eddy_movement_rms

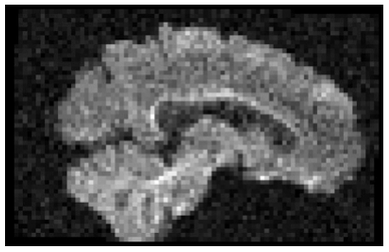
This is a text-file with the cumulative and “delta” movement RMS for each volume.
Useful for QC and for excluding volumes/subjects.

Output

```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```



LR_RL.nii.gz



my_eddy.nii.gz

Slice…

Volume…

```
0        0        -0.089     1.080    -0.002     ...
0        0         0.812     1.563     1.090     ...
0        0         0.107     0.606     0.207     ...
0        0        -0.975    -0.620    -0.467     ...
.
.
.
```

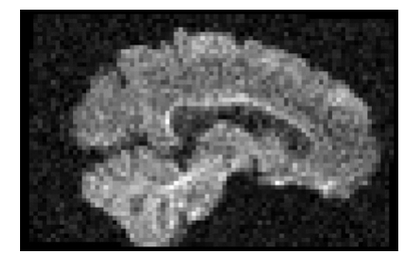
my_eddy.eddy_outlier_n_stdev_map

This is a text-file with one value per slice and volume. It specifies how many standard deviations away from the prediction it is. Useful for QC and for excluding volumes/subjects.

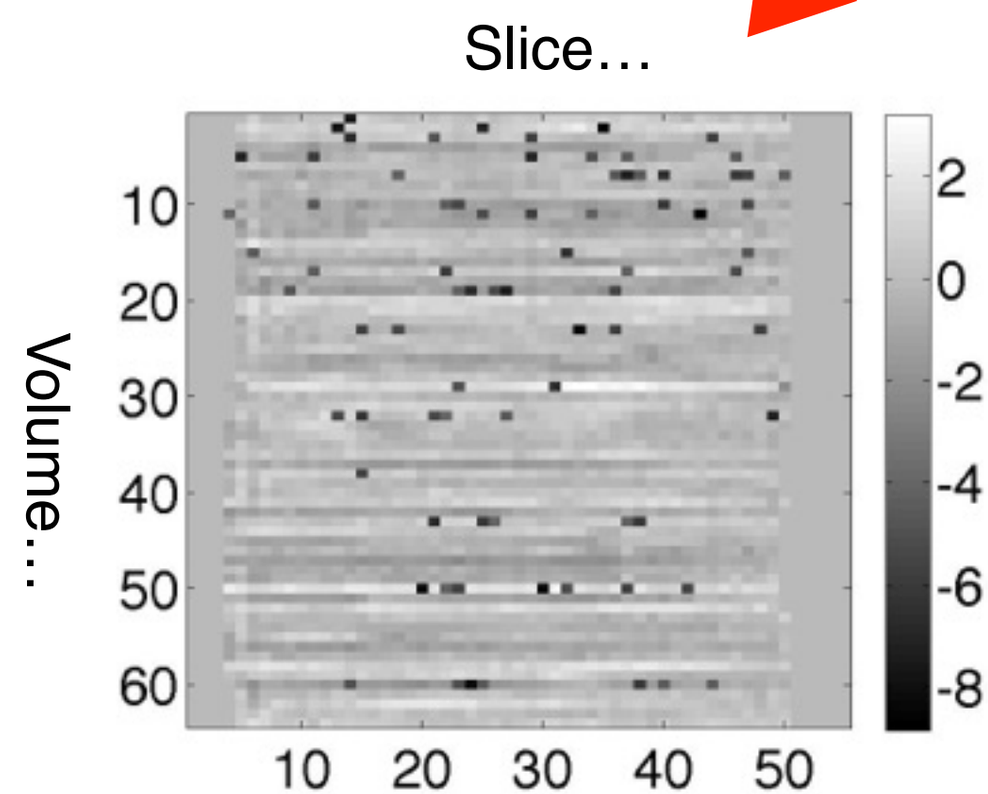
Output

```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```

LR_RL.nii.gz



my_eddy.nii.gz

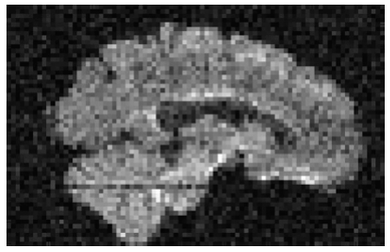


my_eddy.eddy_outlier_n_stdev_map

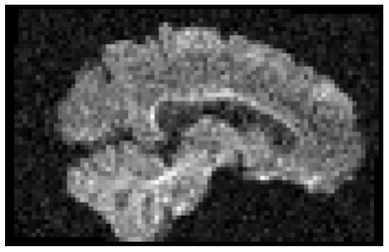
This is a text-file with one value per slice and volume. It specifies how many standard deviations away from the prediction it is. Useful for QC and for excluding volumes/subjects.

Output

```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```



LR_RL.nii.gz



my_eddy.nii.gz

Slice…

Volume…

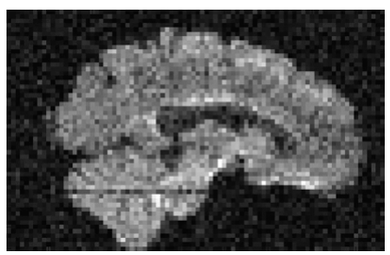
0	0	0	0	0	…
0	0	0	0	0	…
0	0	0	0	0	…
0	0	0	0	0	…
.					
.					
.					

my_eddy.eddy_outlier_map

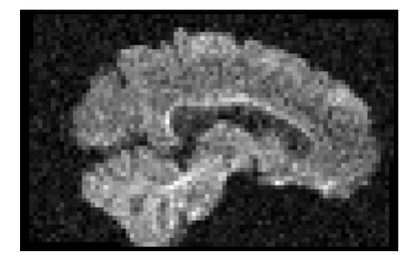
This is a text-file with one value per slice and volume. A “one” signifies an outlier, a “zero” signifies an OK slice. Useful for QC and for excluding volumes/subjects.

Output

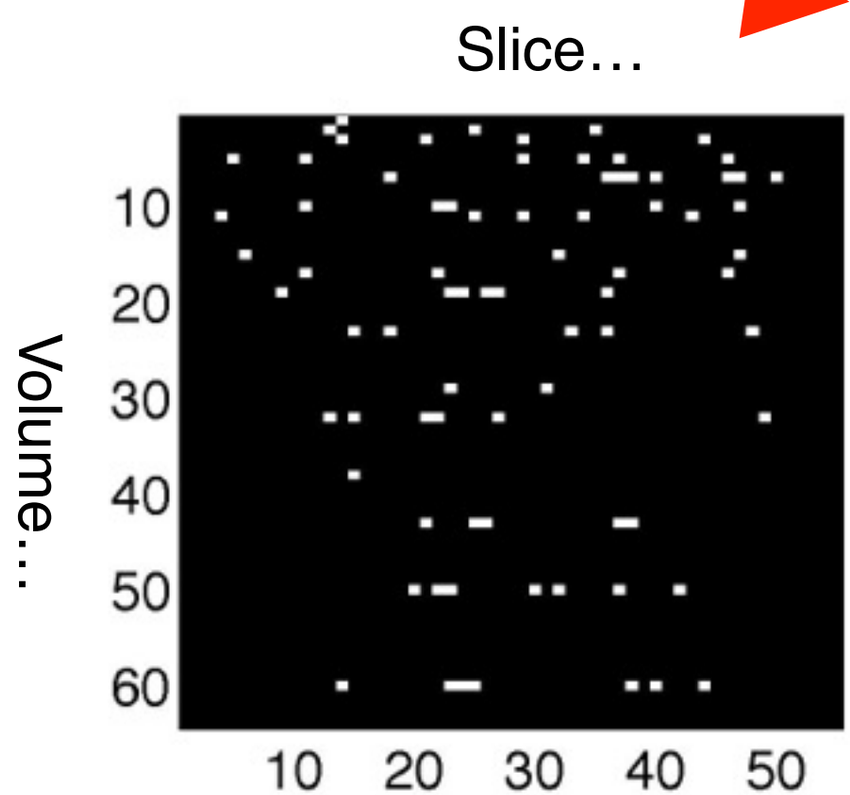
```
eddy --imain=LR_RL …bla bla… --out=my_eddy
```



LR_RL.nii.gz



my_eddy.nii.gz



my_eddy.eddy_outlier_map

This is a text-file with one value per slice and volume. A “one” signifies an outlier, a “zero” signifies an OK slice. Useful for QC and for excluding volumes/subjects.