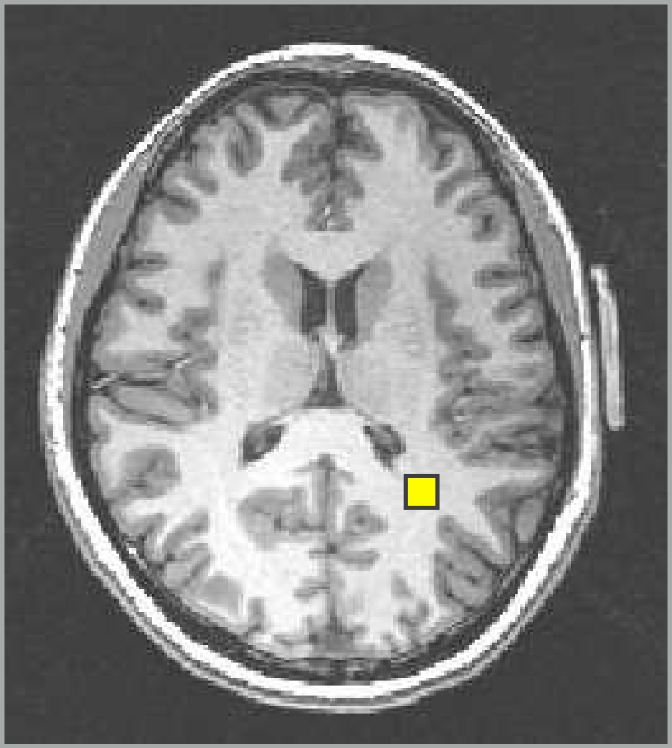
# Lecture 8
# Analyzing the diffusion weighted signal

Please install AFNI

http://afni.nimh.nih.gov/afni/

**Next lecture, DTI**

For this lecture, think in terms of a single voxel



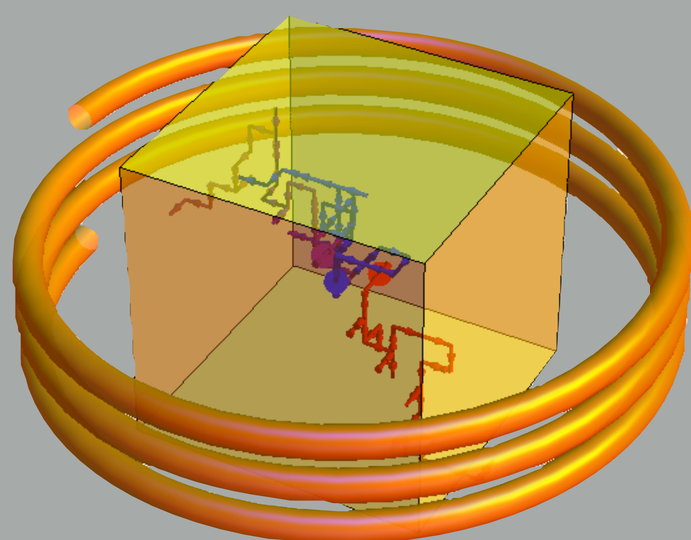
We're still looking only at a single voxel experiment

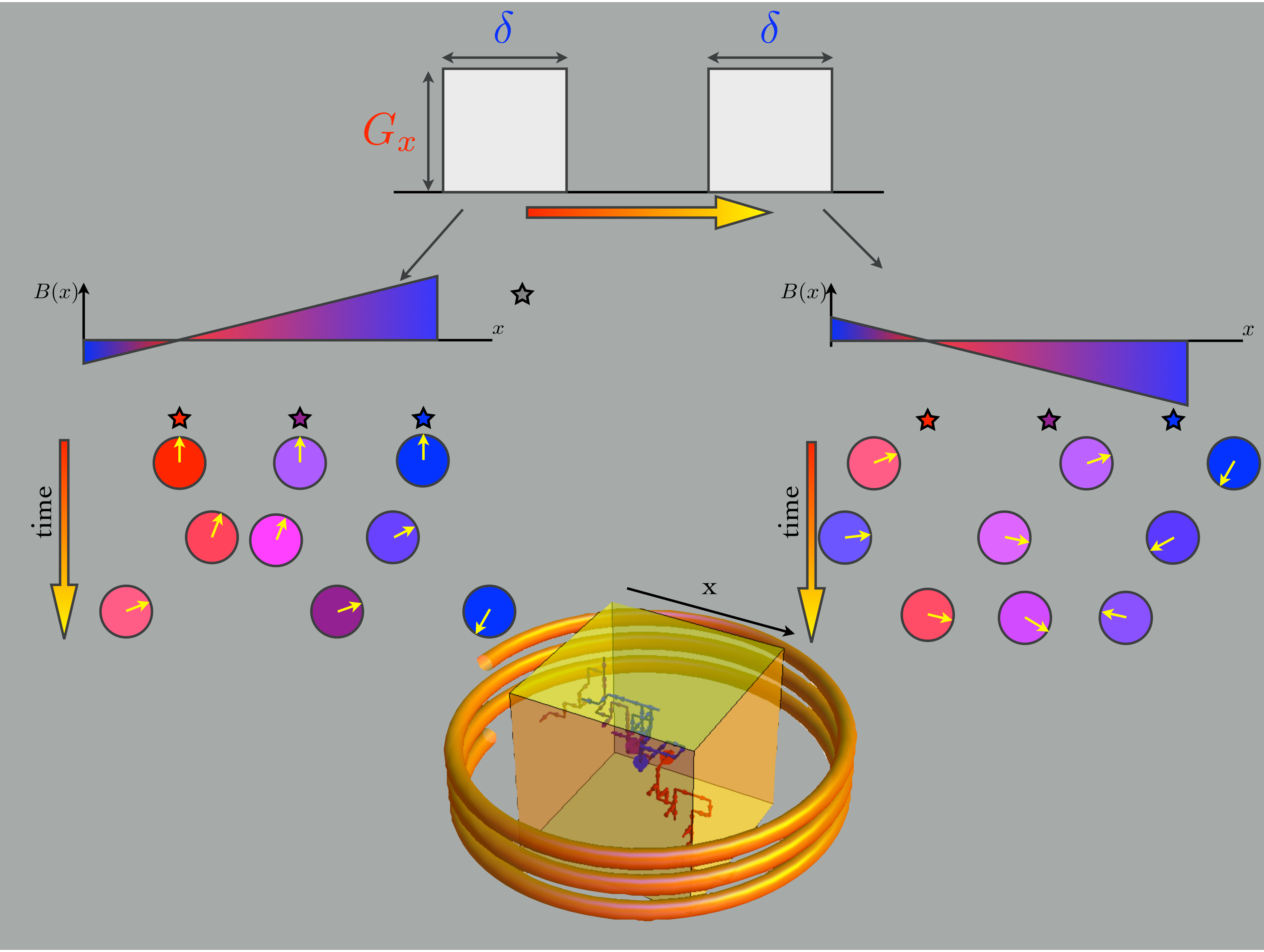
**Last Lecture:**

Single diffusion encoding direction
to estimate a diffusion coefficient D

**This Lecture:**

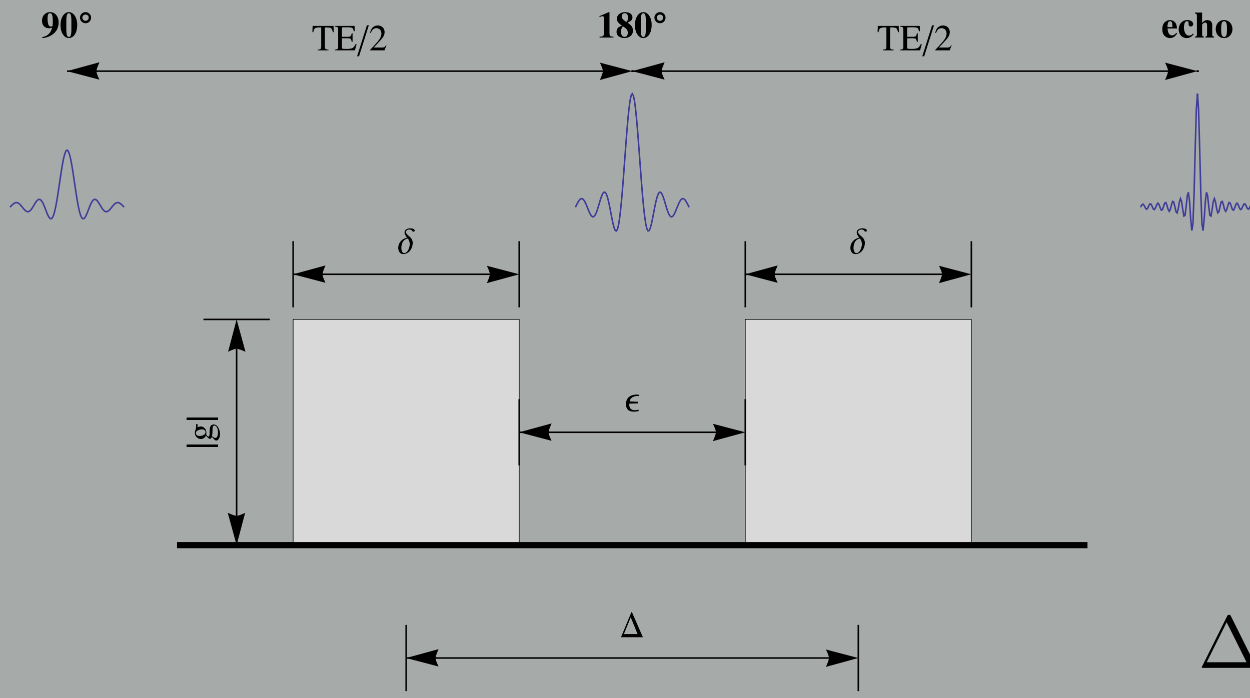
*M*ultiple diffusion encoding directions
to estimate a diffusion tensor **D**





δ
δ
$G_x$
$B(x)$
$x$
$B(x)$
$x$
time
time
x

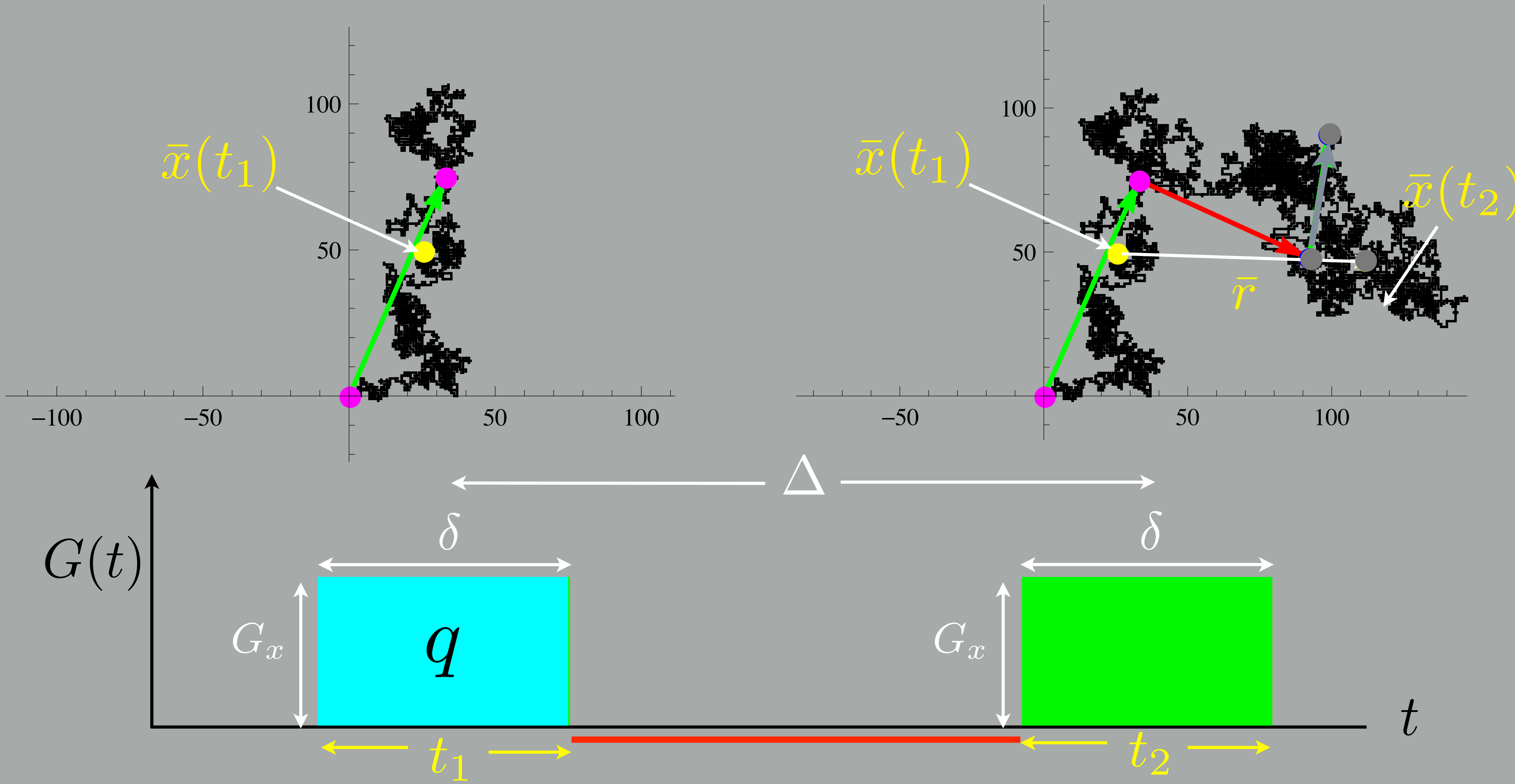
# Phases of diffusing spins




$$\Delta = \delta + \epsilon$$

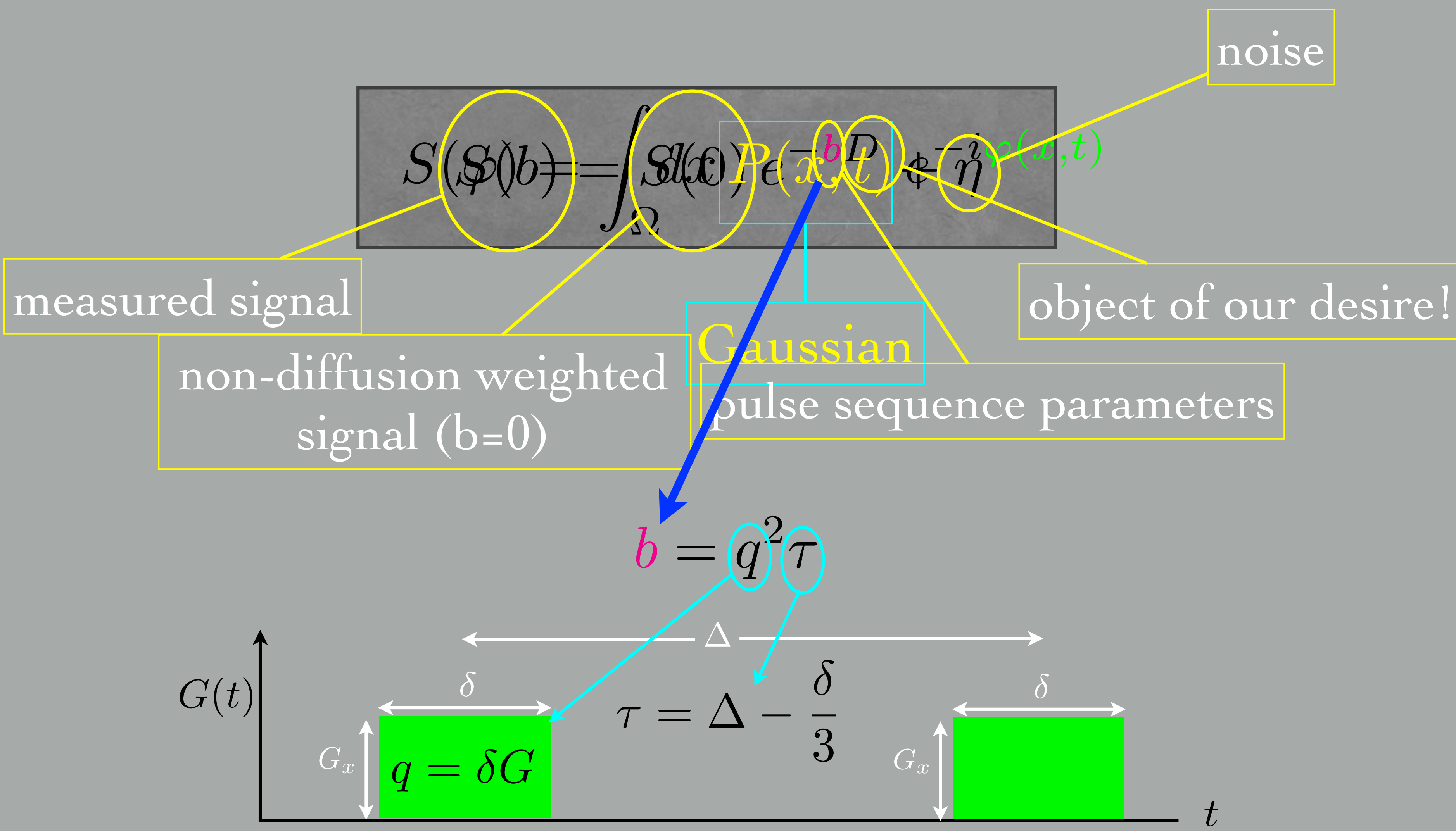
$$\varphi(\tau) = \int_0^\tau G(t)x(t)\,dt$$

# Diffusion phase in a Bipolar Pulse




$$\varphi(x,t) = \underbrace{G\delta}_{q} \underbrace{\left[\bar{x}(t_2) - \bar{x}(t_1)\right]}_{\bar{r}} = q\bar{r}$$

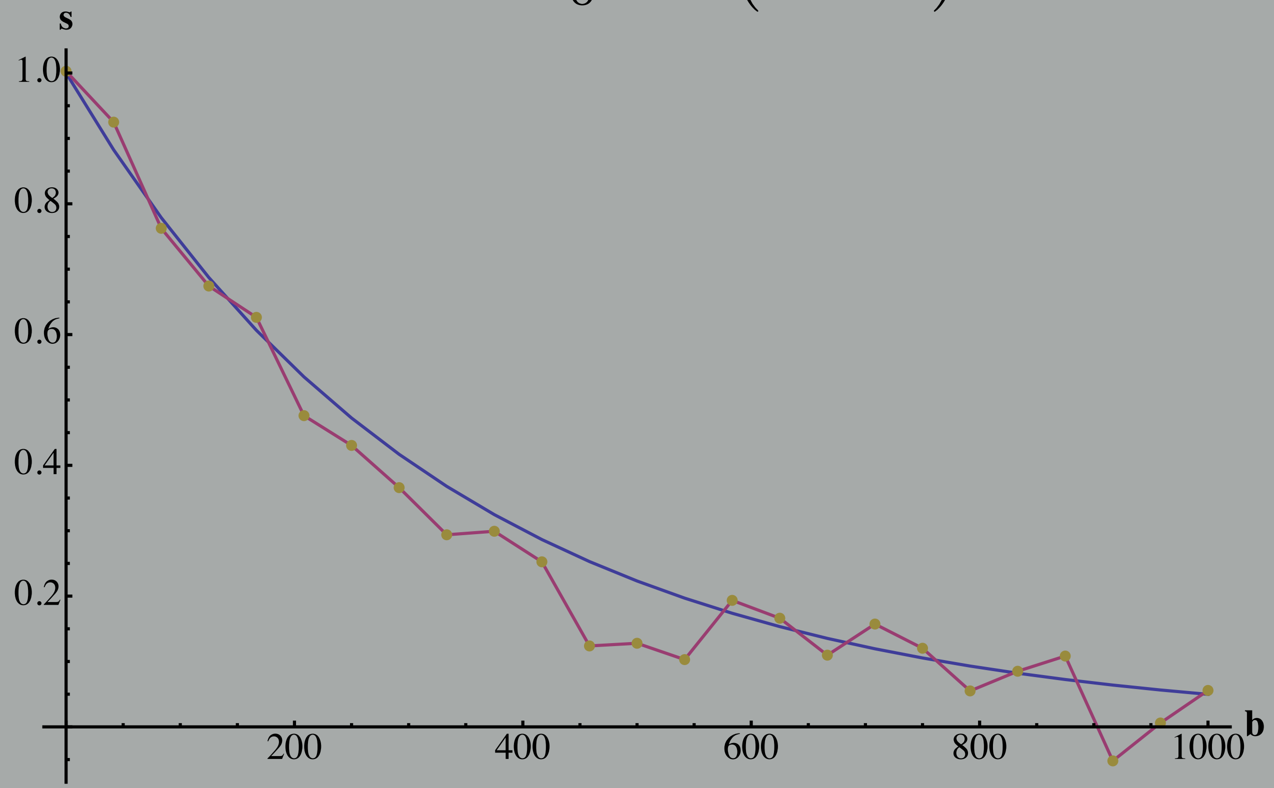
# The Estimation Problem for Gaussian Diffusion



# The signal from 1D Gaussian Diffusion

$$s(b_i) = s_0 e^{-b_i D} + \eta_i$$

$$\text{where} \quad s_0 \equiv s(b = 0)$$



**Consider only two measurements and write data in vector form**

$$\begin{pmatrix} s(b_1) \\ s(b_2) \end{pmatrix} = s_0 \begin{pmatrix} \exp(-b_1 D) \\ \exp(-b_2 D) \end{pmatrix} + \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}$$

$$= s_0 \exp\left[ - \begin{pmatrix} b_1 D \\ b_2 D \end{pmatrix} \right] + \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}$$

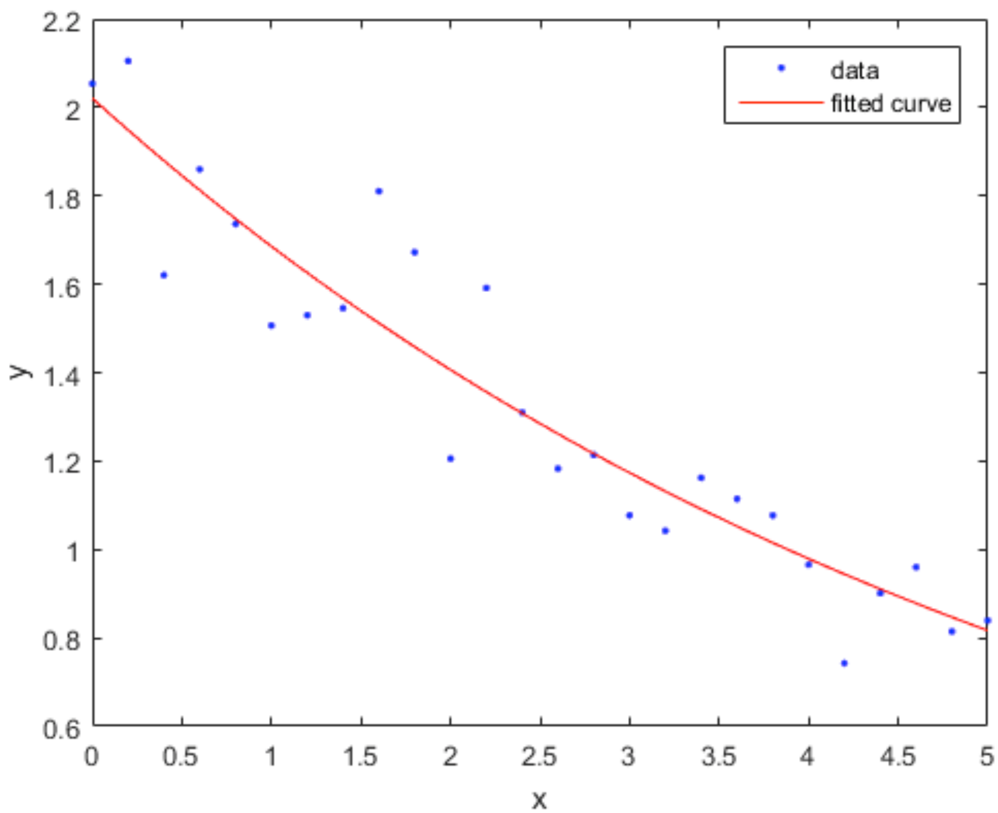
**This clearly generalizes to *n* measurements**

**Fit a Single-Term Exponential Model**

Generate data with an exponential trend and then fit the data using a single-term exponential. Plot the fit and data.

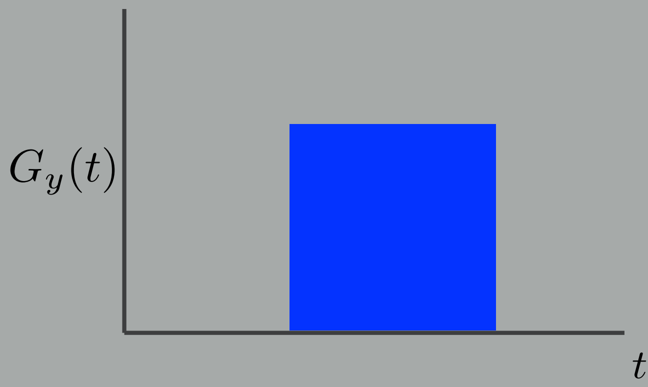
```
x = (0:0.2:5)';
y = 2*exp(-0.2*x) + 0.1*randn(size(x));
f = fit(x,y,'exp1')
plot(f,x,y)
```

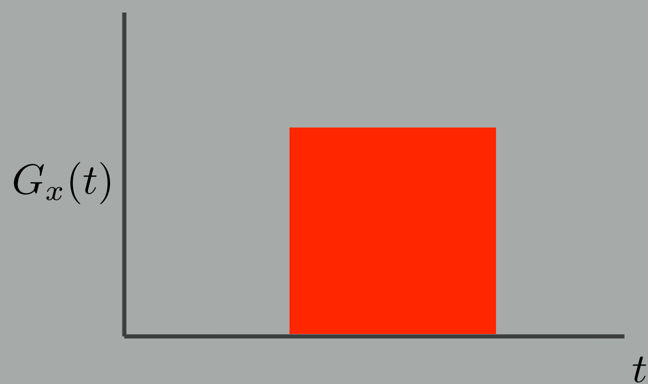
```
f =

     General model Exp1:
     f(x) = a*exp(b*x)
     Coefficients (with 95% confidence bounds):
       a =       2.021  (1.89, 2.151)
       b =     -0.1812  (-0.2104, -0.152)
```

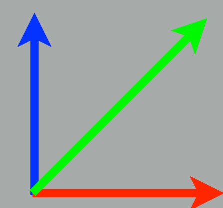


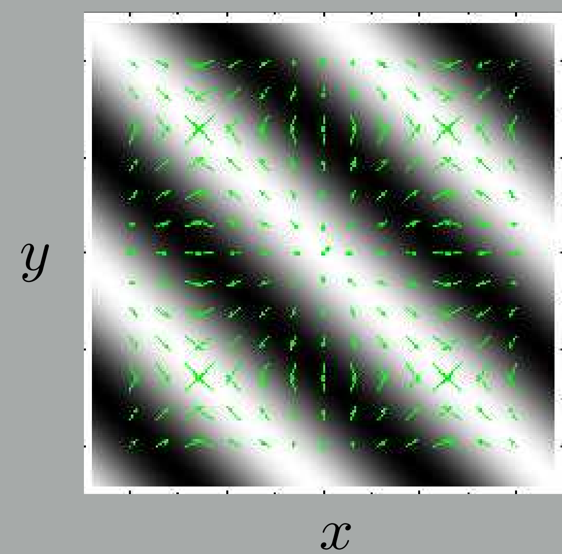
# Recall: gradients add like vectors

$$\boldsymbol{k} \cdot \boldsymbol{x} = k_x x + k_y y = \gamma G_x t x + \gamma G_y t y$$



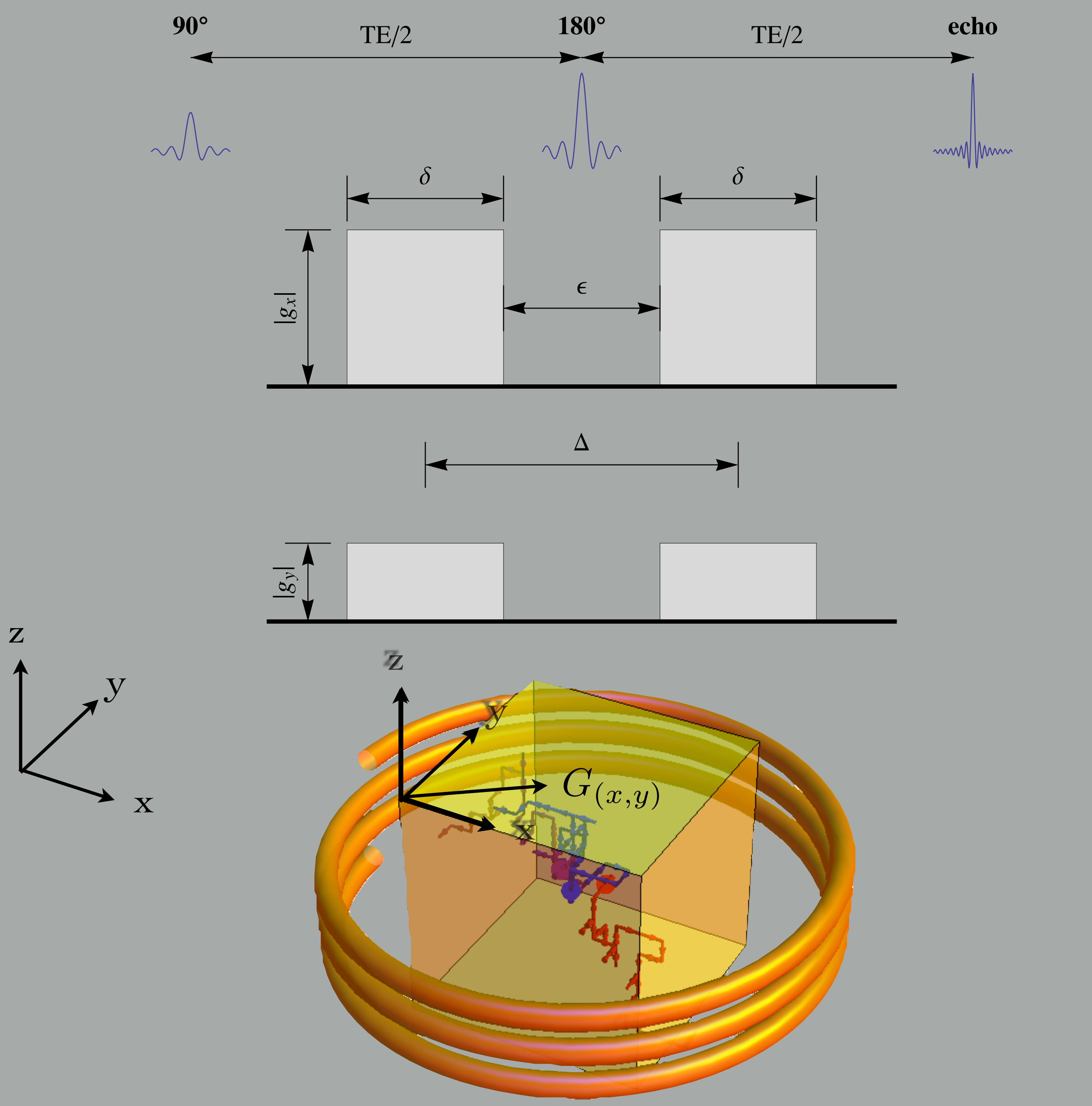



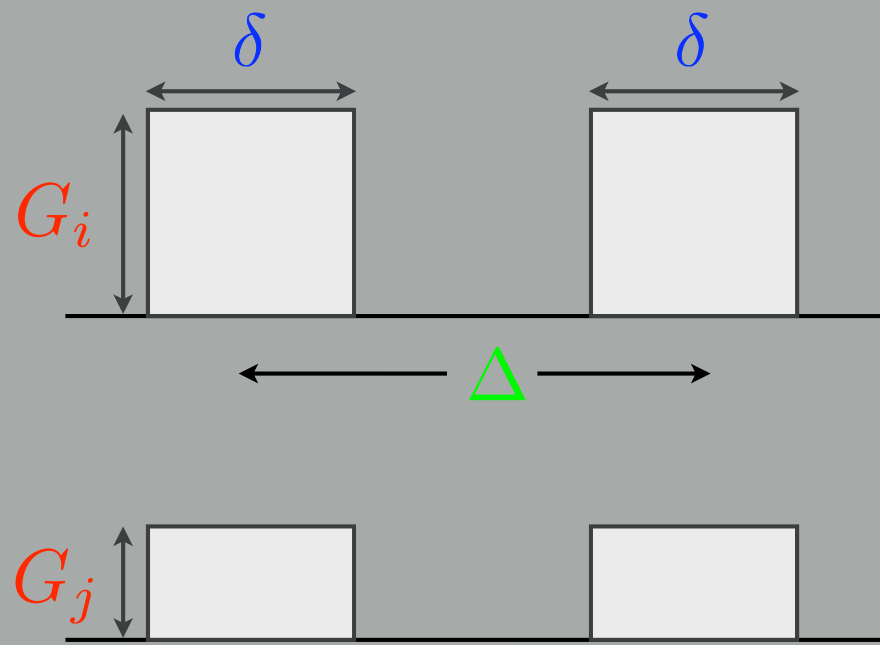



spatial modulation of the phase

# Directional Diffusion Encoding



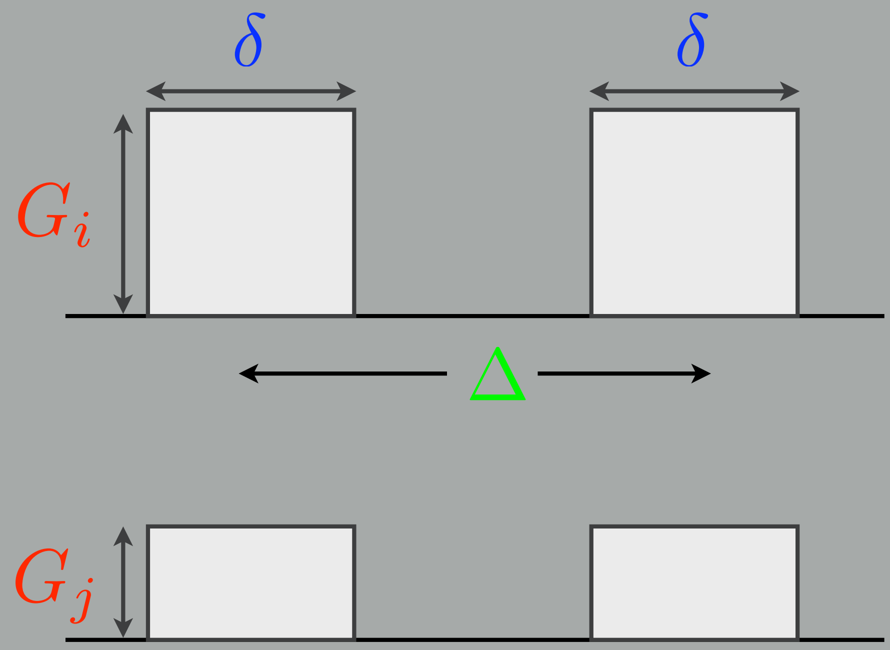
# Ideal b-matrix




$$b_{ij} = G_i G_j \delta^2 \left(\Delta - \delta/3\right)$$

$$b_{ij} = \underbrace{q_i q_j}_{q_i q_j} \underbrace{\tau}_{\tau} \quad \text{where} \quad \begin{cases} q_i = G_i \delta \\ \tau = \Delta - \delta/3 \end{cases}$$

# Ideal b-matrix




$$b_{ij} = q_i q_j \tau \quad \text{where} \begin{cases} q_i = G_i \delta \\ \tau = \Delta - \delta/3 \end{cases}$$

# The b-matrix

$$b_{ij}(\tau) = \int_0^{\tau} q_i(t) q_j(t) dt$$

*i=(x,y,z)*

where $$q = \int g(t)\, dt$$

<u>For constant diffusion gradients</u>

$$b_{ij}(\tau) = q_i q_j \tau$$

## The NMR signal for 1D Gaussian diffusion

$$s(q,\tau) = s(0) \int P(\bar{r},\tau) e^{-iq\cdot\bar{r}} d\bar{r}$$

$$P(\bar{r},\tau) = \frac{1}{\sqrt{4\pi D\tau}} e^{-\bar{r}^2/(4D\tau)}$$

$$\boxed{s(q,\tau) = s(0)e^{-bD}}$$

# The NMR signal for 3D Gaussian diffusion

$$s(q,\tau) = \int P(\bar{r},\tau) e^{-iq\cdot\bar{r}}\, d\bar{r}$$

$$P(\bar{r},\tau) = \frac{1}{\sqrt{(4\pi\tau)^3 |D|}} e^{-\bar{r}^t D^{-1} \bar{r}/4\tau}$$

$$\Downarrow$$

$$\boxed{s(q,\tau) = s(0) e^{-bD}}$$

# b and D

$$\boldsymbol{b} = \begin{pmatrix} q_x^2 & q_x q_y & q_x q_z \\ q_y q_x & q_y^2 & q_y q_z \\ q_z q_x & q_z q_y & q_z^2 \end{pmatrix} \tau$$

known

$$\boldsymbol{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix}$$

desired

# The NMR signal

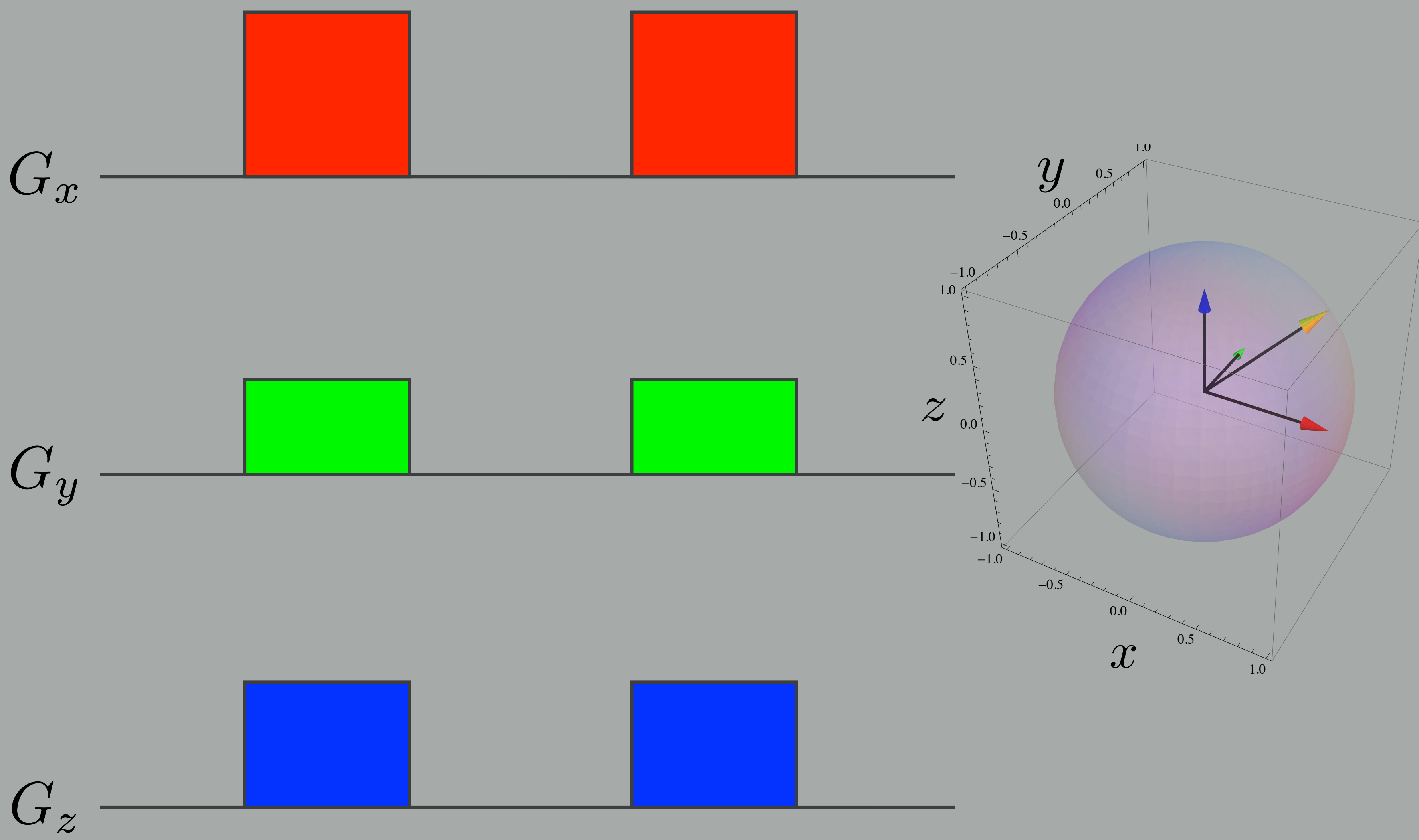
3D Gaussian diffusion

$$\boxed{s(\boldsymbol{q}, \tau) = s(0) e^{-\boldsymbol{bD}}}$$

$$s(\boldsymbol{b}) = s(0) \exp\left( - \sum_{i}^{3} \sum_{j}^{3} b_{ij} D_{ij} \right)$$

$$bD = \tau q^2 D \longrightarrow \boldsymbol{bD} = \tau \boldsymbol{q}^t \cdot \boldsymbol{D} \cdot \boldsymbol{q}$$

# A single diffusion-weighting direction



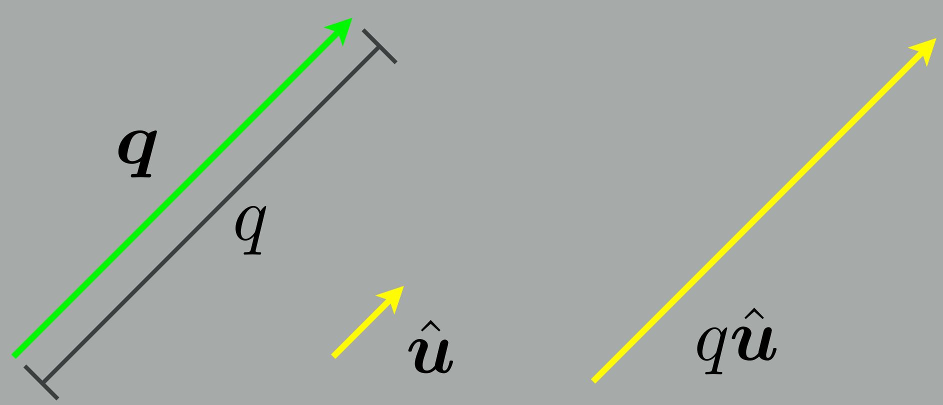
# bD

$$
\begin{aligned}
\frac{1}{\tau}\sum_i \sum_j b_{ij} D_{ij} \quad = \quad & q_x^2 D_{xx} + q_x q_y D_{xy} + q_x q_z D_{xz} + \\
& q_y q_x D_{yx} + q_y^2 D_{yy} + q_y q_z D_{yz} + \\
& q_z q_x D_{yx} + q_z q_y D_{zy} + q_z^2 D_{zz}
\end{aligned}
$$

# Rearranging the directions

$$\boldsymbol{bD} = \tau \boldsymbol{q}^t \cdot \boldsymbol{D} \cdot \boldsymbol{q}$$

$$\boldsymbol{q} = q\hat{\boldsymbol{u}} \qquad q \equiv |\boldsymbol{q}|$$




$$\boxed{\boldsymbol{bD} = q^2 \tau\, \boldsymbol{u}^t \cdot \boldsymbol{D} \cdot \boldsymbol{u}}$$

# The NMR signal

$$\boldsymbol{bD} = q^2\tau \underbrace{\boldsymbol{u}^t \cdot \boldsymbol{D} \cdot \boldsymbol{u}}_{\tilde{\boldsymbol{D}}}$$

$$\Downarrow$$

$$s(\boldsymbol{q}, \tau) = s(0)e^{-\boldsymbol{bD}} = e^{-q^2\tau\,\tilde{\boldsymbol{D}}}$$

$$\Downarrow$$

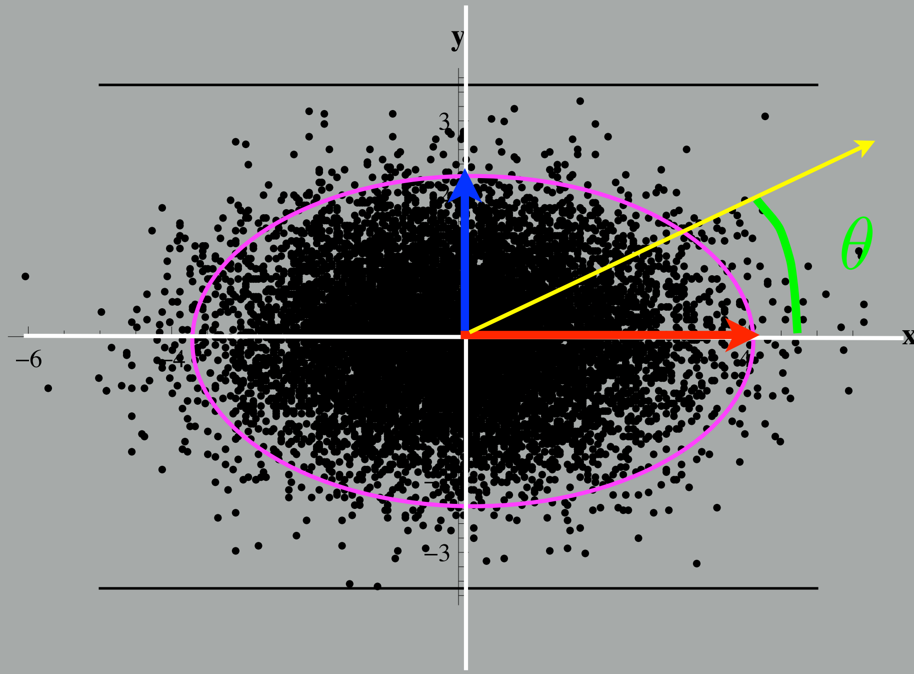
$$\boxed{s(\boldsymbol{q}, \tau) = e^{-b\tilde{\boldsymbol{D}}}}$$

where $b = q^2\tau$

# Measuring the Diffusion Tensor

$$S(b, \hat{r}) = S(0)e^{-b\tilde{D}} + \eta$$

$$\boldsymbol{D} = \begin{pmatrix} D_x & 0 \\ 0 & D_y \end{pmatrix}$$

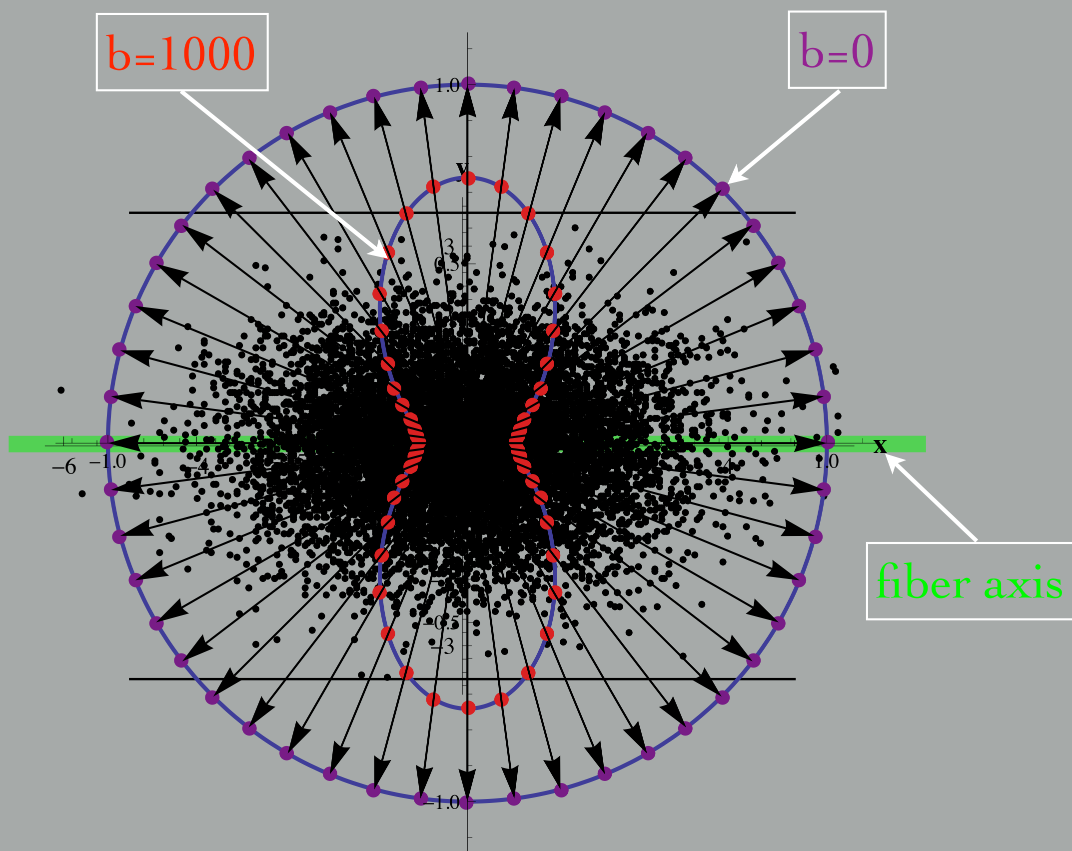



$$\hat{r} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$$

$$\tilde{D} = \hat{r}^t D \hat{r} = D_x \cos^2\theta + D_y \sin^2\theta$$

projection of an ellipsoid!
not like projection of a vector

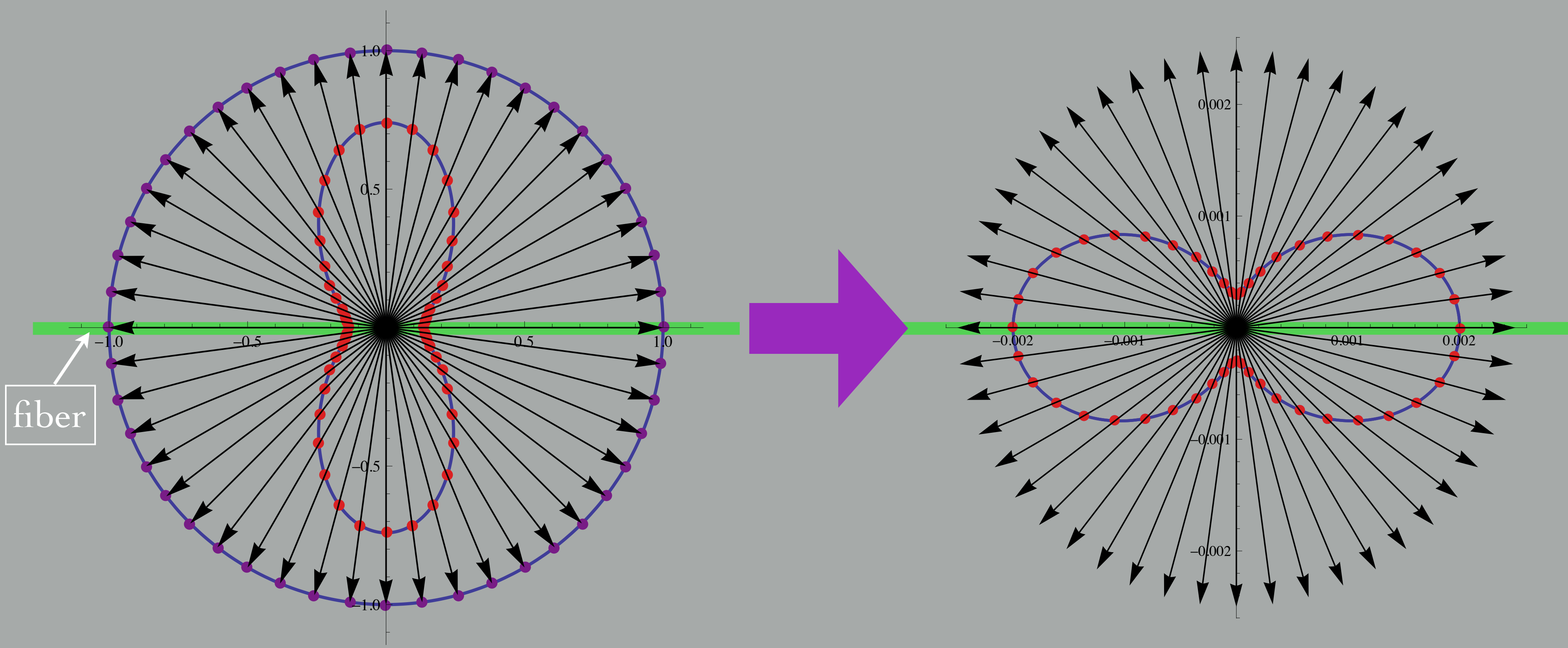
# Measuring the Diffusion Tensor




$$S(b,\theta) = S(0)e^{-bD(\theta)} + \eta$$

$$D(\theta) = D_x \cos^2\theta + D_y \sin^2\theta$$

# The Shape of Diffusion



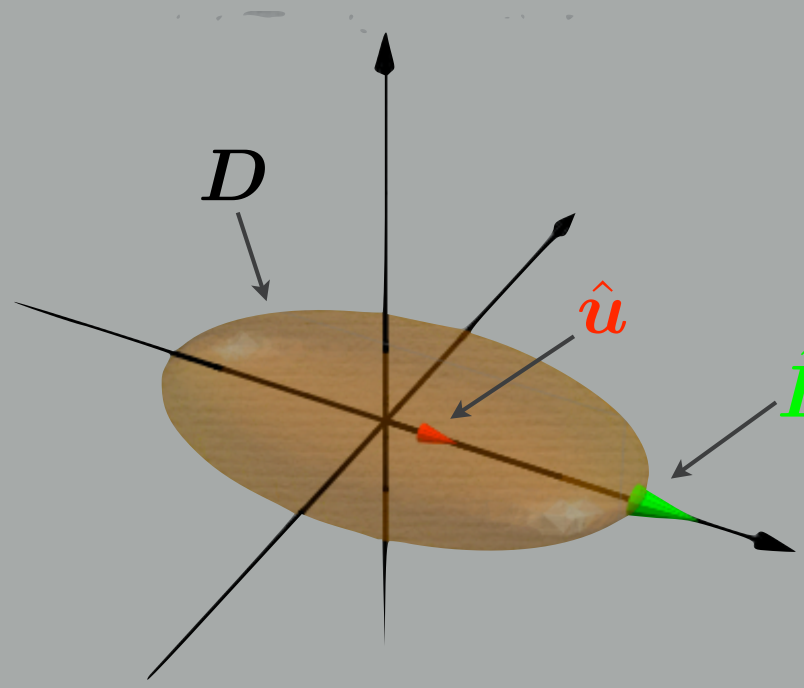

signal $S_b(\theta)$

$$D_{app}(\theta) = -\frac{1}{b}\log\left(\frac{S_b}{S_0}\right)$$

What is the meaning of $\tilde{\boldsymbol{D}}$?

$$\tilde{\boldsymbol{D}} \equiv \boldsymbol{u}^t \cdot \boldsymbol{D} \cdot \boldsymbol{u}$$

It is the projection of $\boldsymbol{D}$ along $\hat{\boldsymbol{u}}$



# Diffusion Tensor is Symmetric

$$\begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}$$

$\boldsymbol{D} = \boldsymbol{D}^t$ matrix form

$D_{ij} = D_{ji}$ component form

# bD

$$\frac{1}{\tau}\sum_i\sum_j b_{ij}D_{ij} \quad = \quad q_x^2 D_{xx} + q_x q_y D_{xy} + q_x q_z D_{xz} +$$

$$q_y q_x D_{yx} + q_y^2 D_{yy} + q_y q_z D_{yz} +$$

$$q_z q_x D_{yx} + q_z q_y D_{zy} + q_z^2 D_{zz}$$

$$\frac{1}{\tau}\sum_i\sum_j b_{ij}D_{ij} \quad = \quad q_x^2 D_{xx} + 2q_x q_y D_{xy} + 2q_x q_z D_{xz} +$$

$$q_y^2 D_{yy} + 2q_y q_z D_{yz} +$$

$$q_z^2 D_{zz}$$

# A computational simplification

$$s(b) = s(0)e^{-bD}$$

a trick: write $s(0) = e^{\log s(0)}$

$$s(b) = s(0)e^{-bd}$$

# Estimating the Diffusion Tensor

$$\boldsymbol{d} = \begin{pmatrix} D_{xx} \\ D_{yy} \\ D_{zz} \\ D_{xy} \\ D_{xz} \\ D_{yz} \\ -\log s(0) \end{pmatrix}$$

There are *7* unknowns

# Estimating the Diffusion Tensor

$$\boldsymbol{B} = (q_x^2 \ , \ q_y^2 \ , \ q_z^2 \ , \ 2q_xq_y \ , \ 2q_xq_z \ , \ 2q_yq_z \ , \ 1) \ \tau$$

# Estimating the Diffusion Tensor

$$-\log s(b) = \tau(q_x^2\ ,\ q_y^2\ ,\ q_z^2\ ,\ 2q_xq_y\ ,\ 2q_xq_z\ ,\ 2q_yq_z\ ,\ 1)\begin{pmatrix} D_{xx} \\ D_{yy} \\ D_{zz} \\ D_{xy} \\ D_{xz} \\ D_{yz} \\ -\log s(0) \end{pmatrix}$$

$$\boldsymbol{y} \qquad\qquad \boldsymbol{B} \qquad\qquad \boldsymbol{D}$$

But there are *7* unknowns,
so we need *7* equations to solve for them

# Estimating the Diffusion Tensor

$$\boldsymbol{y} = -\begin{pmatrix} \log s(b_1) \\ \log s(b_2) \\ \vdots \\ \log s(b_n) \end{pmatrix}$$

We make 7 measurements,
each with a different direction

# The B-matrix

tensor dimensions →

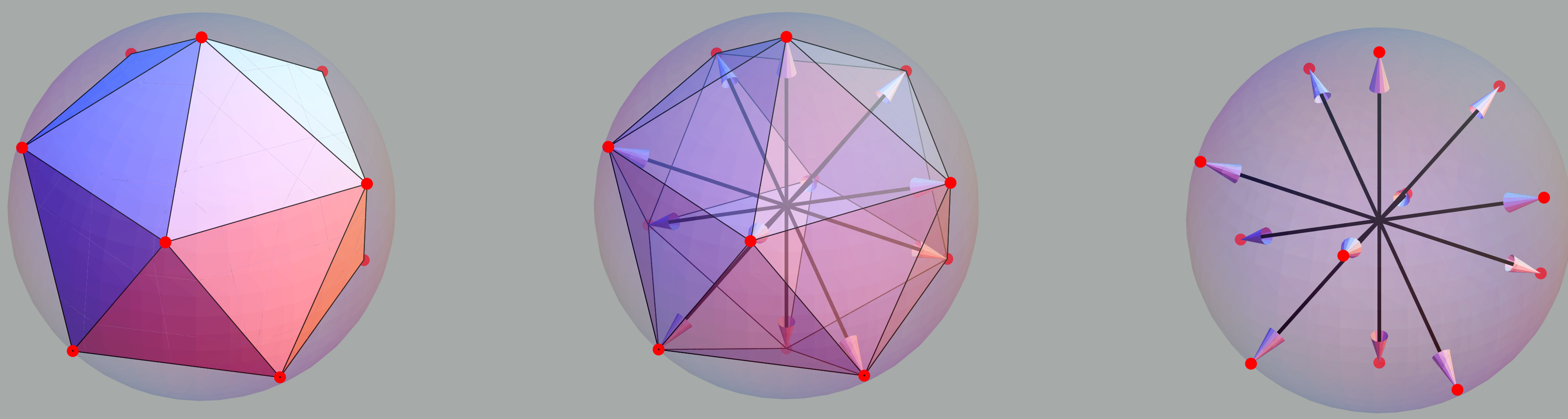
$$\boldsymbol{B} = \begin{pmatrix} \hat{q}_{1,x}^2 & \hat{q}_{1,y}^2 & \hat{q}_{1,z}^2 & \hat{q}_{1,x}\hat{q}_{1,y} & \hat{q}_{1,x}\hat{q}_{1,z} & \hat{q}_{1,y}\hat{q}_{1,z} & 1 \\ \hat{q}_{2,x}^2 & \hat{q}_{2,y}^2 & \hat{q}_{2,z}^2 & \hat{q}_{2,x}\hat{q}_{1,y} & \hat{q}_{2,x}\hat{q}_{2,z} & \hat{q}_{2,y}\hat{q}_{2,z} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{q}_{n,x}^2 & \hat{q}_{n,y}^2 & \hat{q}_{n,z}^2 & \hat{q}_{n,x}\hat{q}_{1,y} & \hat{q}_{n,x}\hat{q}_{n,z} & \hat{q}_{n,y}\hat{q}_{n,z} & 1 \end{pmatrix} \tau$$

↑ gradient directions

$$q_{j,k} = g_k \delta \quad j\text{'th direction}$$

$$\tau = \Delta - \delta/3$$

# Angular measurements



# Estimating the Diffusion Tensor

$$
-\begin{pmatrix} \log s(b_1) \\ \log s(b_2) \\ \vdots \\ \log s(b_n) \end{pmatrix} = \begin{pmatrix} \hat{q}_{1,x}^2 & \hat{q}_{1,y}^2 & \hat{q}_{1,z}^2 & \hat{q}_{1,x}\hat{q}_{1,y} & \hat{q}_{1,x}\hat{q}_{1,z} & \hat{q}_{1,y}\hat{q}_{1,z} & 1 \\ \hat{q}_{2,x}^2 & \hat{q}_{2,y}^2 & \hat{q}_{2,z}^2 & \hat{q}_{2,x}\hat{q}_{1,y} & \hat{q}_{2,x}\hat{q}_{2,z} & \hat{q}_{2,y}\hat{q}_{2,z} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \hat{q}_{n,x}^2 & \hat{q}_{n,y}^2 & \hat{q}_{n,z}^2 & \hat{q}_{n,x}\hat{q}_{1,y} & \hat{q}_{n,x}\hat{q}_{n,z} & \hat{q}_{n,y}\hat{q}_{n,z} & 1 \end{pmatrix} \begin{pmatrix} D_{xx} \\ D_{yy} \\ D_{zz} \\ D_{xy} \\ D_{xz} \\ D_{yz} \\ -\log s(0) \end{pmatrix}
$$

$\boldsymbol{y}$ $\boldsymbol{B}$ $\mathbf{d}$

## Least Squares

The matrix equation

$$\mathbf{y} = \mathbf{B}\mathbf{d}$$

has dimensions

$$[n \times 1] = [n \times m][m \times 1]$$

## Estimating the Diffusion Tensor

Solving for the diffusion tensor
is reduced to finding the solution
to the matrix equation

diffusion tensor elements

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{d}$$

data

b-matrix

# Estimating the Diffusion Tensor

Matrix equation

$$\boldsymbol{y} = \boldsymbol{B}^t \boldsymbol{d}$$

diffusion tensor elements

data

b-matrix

Matrix solution

$$\boldsymbol{d} = \boldsymbol{B}^+ \boldsymbol{y}$$

pseudo-inverse

# Least Squares

The least-squares solution to the matrix equation

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

is

$$?\ \hat{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{y}\ ?$$

NO!

## Least Squares

The least-squares solution to the matrix equation

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$$

is

$$\hat{\boldsymbol{x}} = \boldsymbol{A}^{+}\boldsymbol{y} \quad (\text{note that } \hat{\boldsymbol{x}} \neq \boldsymbol{A}^{-1}\boldsymbol{y})$$

where

$$\boldsymbol{A}^{+} \equiv (\boldsymbol{A}^{t}\boldsymbol{A})^{-1}\boldsymbol{A}^{t}$$

This is called the *pseudo-inverse* of $\boldsymbol{A}$

# Estimating the Diffusion Tensor

In practice

D calculated with 3dDWItoDT (AFNI)

eigensystem calculated by:

[evals,evecs] = eig(D)