Computational Modelling for Biomedical Imaging

Daniel Alexander
D.Alexander@cs.ucl.ac.uk

Course aims

- Practical introduction to the computational tools underlying modelling and indirect estimation.
- Framed within the application of biomedical imaging.
- Learn about some interesting imaging and image analysis techniques.

2

Related courses

- Inverse problems in imaging GV08
  - Simon Arridge
- Machine vision GI14
  - Gabriel Brostow
- Medical imaging MPHYGB10 and MPHYGB11
  - Various
- Information processing in medical imaging MPHYGB06
  - Sebastian Ourselin

3

Prerequisites

- Engineering mathematics
  - Probability
  - Statistics
  - Calculus
- Mathematical programming
  - Matlab

4

Course components

- Danny Alexander: fundamentals of parameter estimation.
- Gary Zhang: Parametric and non-parametric models.
- Ivana Drobnjak: practical modelling and estimation.

5

Week starting	Monday 16-18	Wednesday 11-12	Wednsday 12-14 (lab)	Milestones
9/1/12	Danny (Intro; parameter estimation	Ivana (intro)	Matt and Simon	
16/1/12	Danny (Bayesian methods)	Gemma/ Andrew/Jamie		
23/1/12	Danny and Laura (model selection)	Gemma/ Andrew/Jamie		
30/1/12	Danny (experiment design)	Gemma/ Andrew/Jamie		
6/2/12	Danny and Bernard (validation)	Ivana (Group assignment)		Project outlines finalized
13/2/12	Reading week: no lectures or lab classes			CW1 deadline: 15/2/12
20/2/12	Gary	Danny CW1 review		DW2 set
27/2/12	Gary	Ivana		
5/3/12	Gary	Ivana		
12/3/12	Gary	Ivana		

Assessment

- Three courseworks:
- CW1: Fundamentals of modelling and estimation
  - Danny 30%
- CW2: Multi-scale and non-parametric models
  - Gary 20%
- Group and individual projects
  - Ivana 50%
- Collaboration vs plagiarism.

7

Learning outcomes

- Practical modelling and estimation techniques:
  - Experience with fitting algorithms and common models
  - How to handle real-world data
- Common aims in imaging science
- Some specific imaging techniques
- Some matlab programming

8

Lecture support

- Lab classes
  - Tuesdays 14-16
  - Lab demonstrators: Simon Richardson, Matt Rowe.
  - CS Accounts?
- Office hour
  - To be decided
- Project leaders
  - Gemma Morgan, Andrew Melbourne, Jamie McClelland
- Website:
- Email list?
- Pen and paper!

9

Part 1

- Modelling, optimization and estimation
- Model selection
- Experiment design
- Validation

Recommended reading:
www.causascientia.org/**math**_stat/Tutorial.pdf

10

Lecture 1

- Basic ideas, concepts and definitions
- Optimization and parameter estimation
- Parametric mapping
- Example: diffusion imaging
- (Bayesian models)

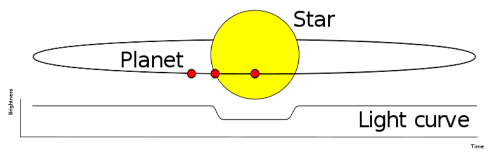
11

Indirect measurement

- What do we want to know?
- What can we measure?
- What relates what we can measure to what we want to measure?

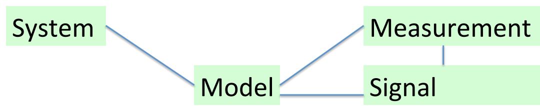
12

Example: exoplanets



What is a model?

- Set of equations (mathematical model) or algorithms (computational model) that relate properties of a system to an observable signal.



For a more general discussion of this question, see: http://www.emily-griffiths.postgrad.shef.ac.uk/models.pdf

What is a parameter?

- Parameters are, often unknown, properties of the system that become variables of the model.

What is data?

- Data are measurements from a device that have sensitivity to the parameters of the system.
- We aim to estimate parameters by fitting models to data

What is noise?

- Noise is any influence on the data that does not come from the system under investigation.
- Types of noise:
  - Measurement error
  - Quantization error
  - Catastrophic failure
  - Modelling error

Components of a model

- **Physical model** has several, at least some unknown, parameters
  $\mathbf{x} = (x_1, \ldots, x_N)^T$
- **Signal** or **device model** has several, usually known, settings
  $\mathbf{y} = (y_1, \ldots, y_M)^T$
- **Noise model** has, known or unknown, parameters
  $\mathbf{z} = (z_1, \ldots, z_L)^T$

Components of a model

- The full model predicts signal, deterministically via physical and signal models, and measurements stochastically via noise model
- For additive noise

$$A(\mathbf{y}) = S(\mathbf{x};\mathbf{y}) + \eta(\mathbf{z})$$

19

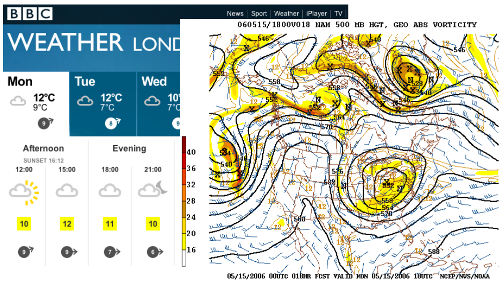
What can we do with a model?

- Learn about the world
- Estimate parameters
- Make predictions

20

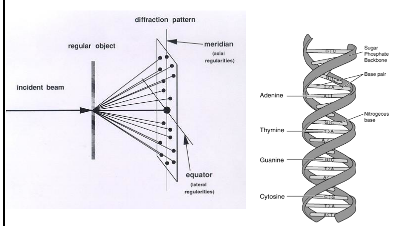
More Examples

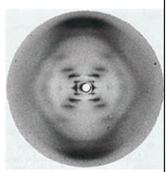
- Weather forecast



More examples

- X-ray diffraction crystalography

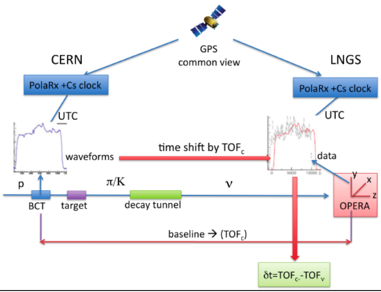




22

More examples

- Speed of neutrinos



23

Estimation

- Fit the model to the data.
- Find values for the parameters so that predicted measurements from the model match measured data best.
- Ie, minimize the difference between predicted and measured data with respect to the model parameters.

24

Optimization

- Numerical techniques for locating the minima (or maxima) of a function.
  $\tilde{\mathbf{x}} = \operatorname{argmax} f(\mathbf{x})$
- f is the objective function.
- $\mathbf{x}$ is the vector of parameters.

Least squares fit

- Minimizes sum of square differences (SSD):
  $f(\mathbf{x}) = \sum_{i=1}^{K} (A(\mathbf{y}_i) - S(\mathbf{x};\mathbf{y}_i))^2$
- K is the number of measurements and $A(y_i)$ is the i-th measurement, which corresponds to device settings y_i.

Linear least squares

- A linear model has form $S(\mathbf{x};\mathbf{y}) = \sum_{i=1}^{N} g_i(\mathbf{y}) x_i$
- To minimize SSD, we simply need to solve a matrix equation, $G\mathbf{x} = \mathbf{A}$ for $\mathbf{x}$.
- To obtain $\tilde{\mathbf{x}} = G^{+}\mathbf{A}$, where $G^{+} = (G^T G)^{-1} G^T$

Non-linear optimization

- Seeks a sequence $\mathbf{x}_0$, $\mathbf{x}_1$, … that converges to $\tilde{\mathbf{x}}$ where $\nabla f(\tilde{\mathbf{x}}) = 0$
- Gradient descent: $\mathbf{x}_{i+1} = \mathbf{x}_i + \gamma \nabla f(\mathbf{x}_i)$
- Newton's method: $\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma H^{-1}(\mathbf{x}_i) \nabla f(\mathbf{x}_i)$
  where H is the Hessian of f.

An Introduction to the Conjugate Gradient Method Without the Agonizing Pain
Jonathan Richard Shewchuk
http://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf

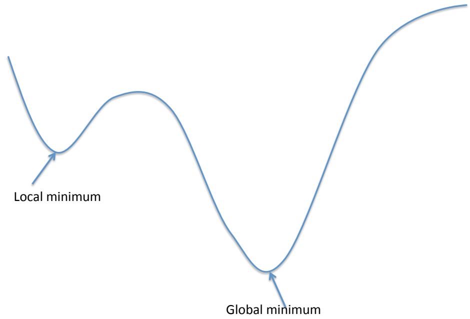
Non-linear optimization

- Quasi-Newton methods compute H numerically from successive iterations
- Gauss-Newton approximates H from first derivatives specifically for least squares.
- Levenberg-Marquardt interpolates between gradient descent and Gauss-Newton for reliable convergence.
- See `fminunc` in matlab.

Non-linear optimization

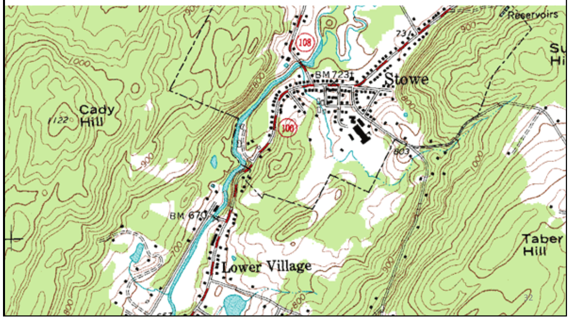
- Other deterministic techniques
  - Simplex method (`fminsearch` in matlab)
  - Powell's method
  - Do not require derivatives
- Constrained optimization
  - Transformation method
  - Soft/penalized constraints
  - Trust region/active set methods (`fmincon` in matlab)

Global and local minima



31

Global and local minima



32

Stochastic techniques

- Repeated gradient descent
- Simulated annealing
- Population methods:
  - Genetic search
  - SOMA (self-organizing migratory algorithm)
  - Differential evolution

33

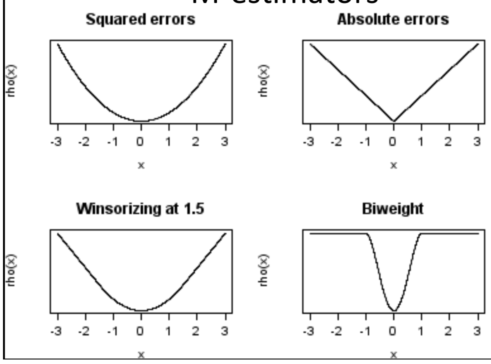
Objective functions

- Sum of square differences $\sum_{i=1}^{K}(A(\mathbf{y}_i)-S(\mathbf{x};\mathbf{y}_i))^2$
- L1 norm $\sum_{i=1}^{K}\left|A(\mathbf{y}_i)-S(\mathbf{x};\mathbf{y}_i)\right|$
- Robust statistics and M-estimators

$$\sum_{i=1}^{K}\rho(A(\mathbf{y}_i)-S(\mathbf{x};\mathbf{y}_i))$$

34

M-estimators



35

Maximum likelihood estimation

- Maximizes the likelihood of the data under the noise model

$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}}\ p(\mathbf{A} \mid \mathbf{x})$$

- For additive noise

$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \prod_{i=1}^{K} p(A(\mathbf{y}_i)-S(\mathbf{x};\mathbf{y}_i);\mathbf{z})$$

- For additive Gaussian noise

$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \sum_{i=1}^{K} \frac{(A(\mathbf{y}_i)-S(\mathbf{x};\mathbf{y}_i))^2}{\sigma_i^2}$$

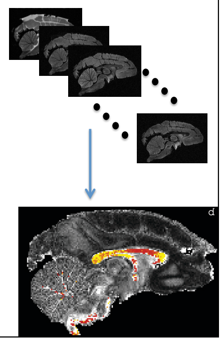
36

Noise model

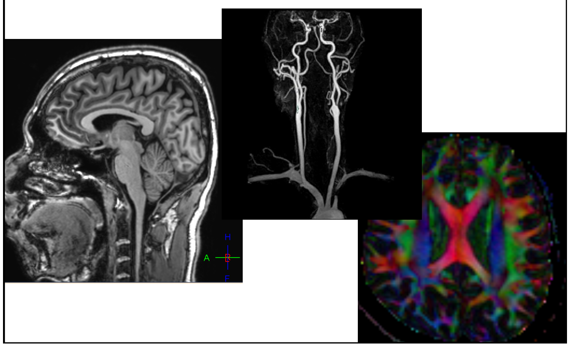
- Sum of squares implies identically distributed zero-mean Gaussian noise
- Weighted sum of squares also simple to solve.
- Other noise models produce other M-estimators:
  - Laplacian distributed noise for L1 norm
  - t-Distributions for robust estimators

Imaging application: parametric mapping

- Fits a model in every image pixel/voxel and produces an image of the model parameter.
- Input: several images acquired with different device settings.
- Output: set of fitted parameters in each pixel.



MRI



Diffusion MRI

- Image intensity sensitive to water mobility
- Water mobility determined by local tissue architecture

Water diffusion in Brain Tissue

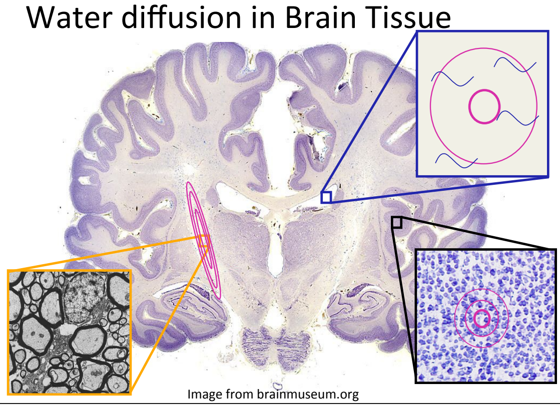
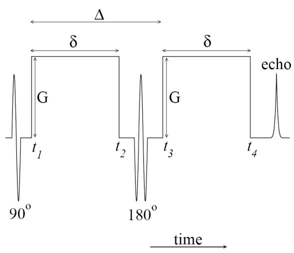


Image from brainmuseum.org

Estimate diffusivity

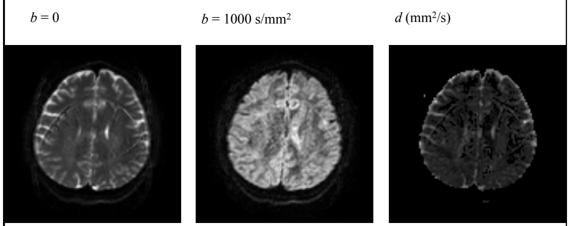


$$b = \left(\gamma\delta|\mathbf{G}|\right)^2(\Delta - \delta/3)$$

$$b = |\mathbf{q}|^2\, t$$

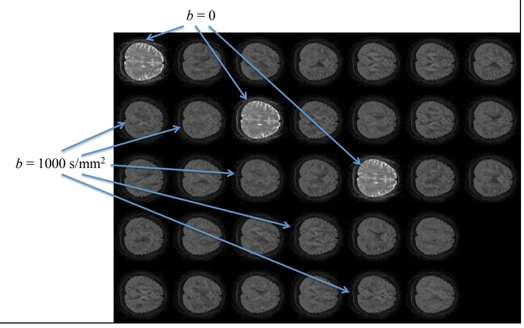
$$S(b) = S(0)\exp(-bd)$$

Estimate diffusivity



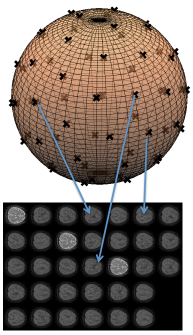
Always eyeball the data!

Data set



Gradient directions

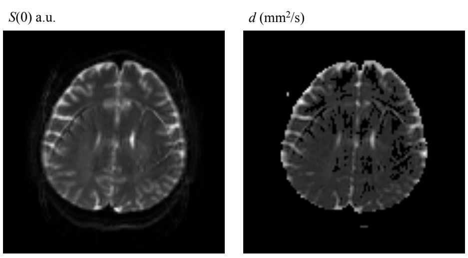
- For all the b=1000s/mm^2 images, $|\mathbf{q}|$ and t are the same…
- … but the orientation $\hat{\mathbf{q}}$ of $\mathbf{q}$ is unique.



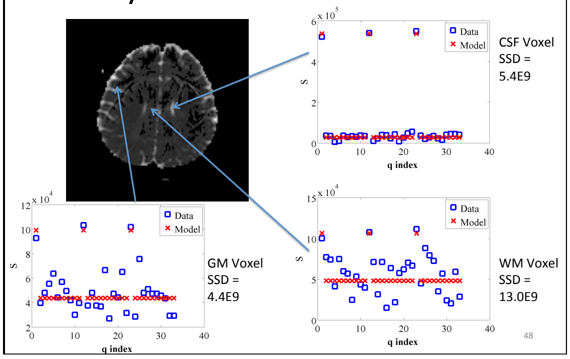
Estimation from full data set

$$\mathbf{A} = \begin{pmatrix} \log A(b_1) \\ \log A(b_2) \\ \vdots \\ \log A(b_K) \end{pmatrix} = \begin{pmatrix} 1 & b_1 \\ 1 & b_2 \\ \vdots & \vdots \\ 1 & b_K \end{pmatrix} \begin{pmatrix} \log S(0) \\ d \end{pmatrix} = Y\mathbf{x}$$

Parameter maps



Eyeball the fit to the data



Diffusion Tensor MRI

- Direct extension to 3D.
- Models d as a quadratic function of direction:
  $d(\hat{\mathbf{q}}) = \hat{\mathbf{q}}^T D \hat{\mathbf{q}}$

$$D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix} \qquad S(b,\hat{\mathbf{q}}) = S(0,0)\exp(-b\hat{\mathbf{q}}^T D \hat{\mathbf{q}})$$

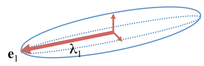
Basser et al, Biophysical Journal 1994

The Diffusion Tensor

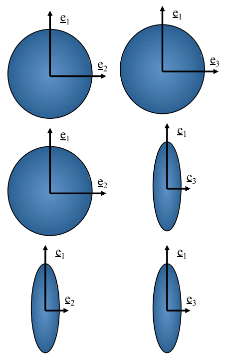
- Proportional to the covariance of a trivariate Gaussian.
- Positive definite, symmetric 3x3 matrix:

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix}$$

- Determines the elliptical shape and orientation of the contours



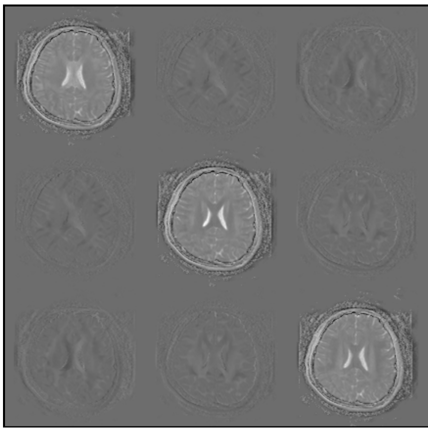
Types of DT



$\lambda_1 \approx \lambda_2 \approx \lambda_3$ – Isotropic
Grey matter, CSF

$\lambda_1 \approx \lambda_2 >> \lambda_3$ – Oblate
White matter

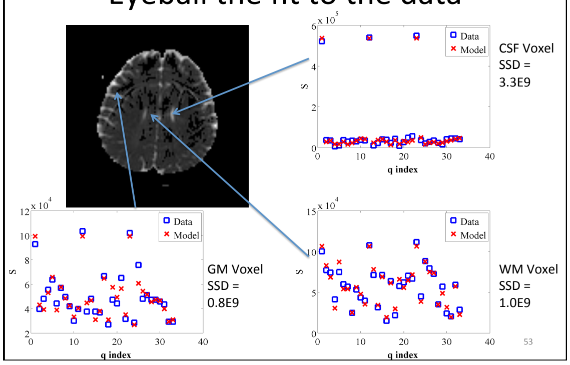
$\lambda_1 >> \lambda_2 \approx \lambda_3$ – Prolate
White matter



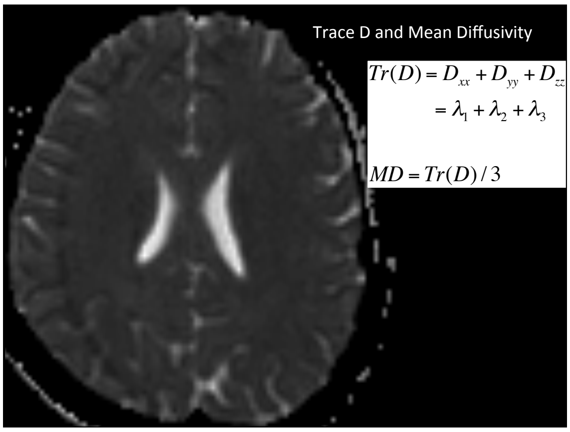
Elements of the diffusion tensor:

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

Eyeball the fit to the data



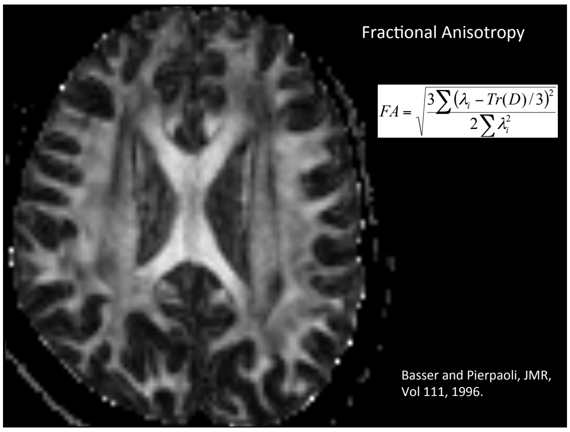
53

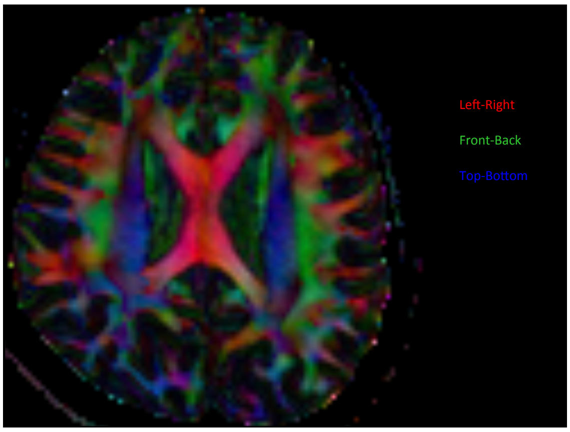


Trace D and Mean Diffusivity

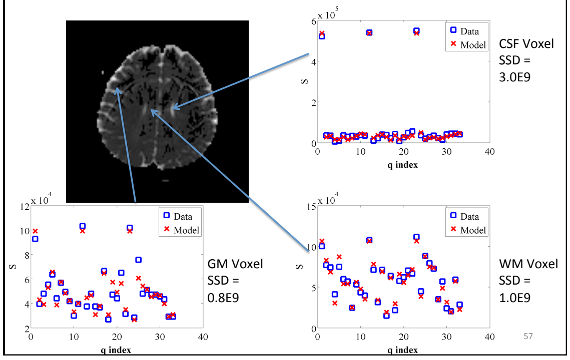
$$Tr(D) = D_{xx} + D_{yy} + D_{zz} = \lambda_1 + \lambda_2 + \lambda_3$$

$$MD = Tr(D)/3$$





Non-linear DT fit



Non-linear DT fit

```
voxdw = dwis(:,66,58,25);
ps = pinv(X)*log(voxdw);
startx = ps;
startx(1) = exp(startx(1));
h = optimset();
[par,RES,EXIT,OUT]=fminunc('DT_SSD',startx,h,voxdw,bvals,qhat);


function [sumRes, resJ] = DT_SSD(x, meas, bvals, qhat)

S0 = x(1);
Dxx=x(2);Dxy=x(3);Dxz=x(4);Dyy=x(5);Dyz=x(6);Dzz=x(7);
D = [[Dxx Dxy Dxz];[Dxy Dyy Dyz];[Dxz Dyz Dzz]]

% The abs is a cheat to add numerical stability.
S = S0*exp(-abs(sum(qhat.*(D*qhat)).*bvals))

sumRes = sum((meas - S').^2);
```


Rician noise

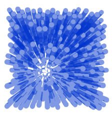
- Noise on MRI intensity is only approximately Gaussian distributed.
- The Rician noise model is more accurate, particularly when signal is low.
- Modulus of complex numbers with Gaussian distributed real and imaginary parts

$$p(M \mid S,\sigma) = \frac{M}{\sigma^2}\exp\left(\frac{-(M^2+S^2)}{2\sigma^2}\right) I_0\left(\frac{MS}{\sigma^2}\right)$$

- I_0 is the Bessel function of the first kind order 0.
- M is the measurement (data); S is the signal (model)
- σ is the standard deviation of the Gaussian distributions.

See **Alexander**, D. C. (2009). **Modelling**, **Fitting** and **Sampling** in Diffusion MRI.

Alternative models



- Compartment models

$$S(b,\hat{\mathbf{q}}) = S(0,0)(fS_I(b,\hat{\mathbf{q}}) + (1-f)S_E(b,\hat{\mathbf{q}}))$$

- Eg, ball and stick model

$$S_I(b,\hat{\mathbf{q}}) = \exp(-bd(\hat{\mathbf{q}}.\mathbf{n})^2)$$
$$S_E(b,\hat{\mathbf{q}}) = \exp(-bd)$$

Panagiotaki et al NeuroImage 2012