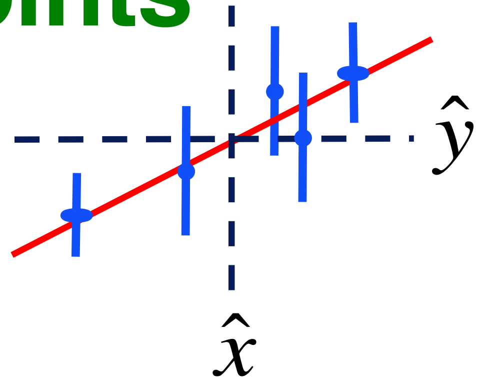
ADA09 - 10am Mon 03 Sep 2022

Iterated Optimal Scaling
Linear Regression
Hessian Matrix
(= inverse of Parameter Covariances)

Non-Linear Models:

1. Linearised Regression
2. Amoeba algorithm
3. MCMC algorithm

Review: Fit a line to *N* data points



Correlated parameters: ☹

$$y = a\,x + b$$

Orthogonal parameters: ☺

$$y = a\,(x - \hat{x}) + b$$

Pivot point:

$$\hat{x} \equiv \frac{\sum x_i / \sigma_i^2}{\sum 1/\sigma_i^2}$$

For intercept *b*, set *a=0* and find *b* by **optimal average**:

$$\hat{b} = \frac{\sum y_i / \sigma_i^2}{\sum 1/\sigma_i^2}, \qquad \mathrm{Var}\left[\hat{b}\right] = \frac{1}{\sum 1/\sigma_i^2}$$

For slope *a*, set *b=0* and find *a* by **optimal scaling**:

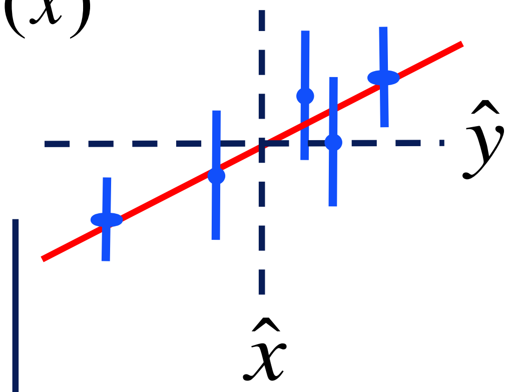
$$\hat{a} = \frac{\sum y_i (x_i - \hat{x}) / \sigma_i^2}{\sum (x_i - \hat{x})^2 / \sigma_i^2}, \quad \mathrm{Var}\left[\hat{a}\right] = \frac{1}{\sum (x_i - \hat{x})^2 / \sigma_i^2}$$

No need to iterate. (Why?)

Fit a line => fit 2 patterns => fit *M* patterns

Model: $y = b + a\left(x - \hat{x}\right) \quad = \quad \alpha_1\, P_1(x) + \alpha_2\, P_2(x)$

2 Patterns: $P_1(x) = 1 \qquad P_2(x) = \left(x - \hat{x}\right)$



Pivot point:

$$\hat{x} \equiv \frac{\sum x_i / \sigma_i^2}{\sum 1 / \sigma_i^2}$$

Iterated Optimal Scaling:

$$\hat{\alpha}_1 = \frac{\sum \left(y_i - \hat{\alpha}_2\, P_2(x_i)\right) P_1(x_i) / \sigma_i^2}{\sum P_1^2(x_i) / \sigma_i^2}, \quad \mathrm{Var}\left[\hat{\alpha}_1\right] \approx \frac{1}{\sum P_1^2(x_i) / \sigma_i^2}$$

$$\hat{\alpha}_2 = \frac{\sum \left(y_i - \hat{\alpha}_1\, P_1(x_i)\right) P_2(x_i) / \sigma_i^2}{\sum P_2^2(x_i) / \sigma_i^2}, \quad \mathrm{Var}\left[\hat{\alpha}_2\right] \approx \frac{1}{\sum P_2^2(x_i) / \sigma_i^2}$$

Iterate (if patterns not orthogonal).

LINEAR REGRESSION:
Generalise model to *M* patterns:

$$y = \sum_{k=1}^{M} \alpha_k\, P_k(x)$$

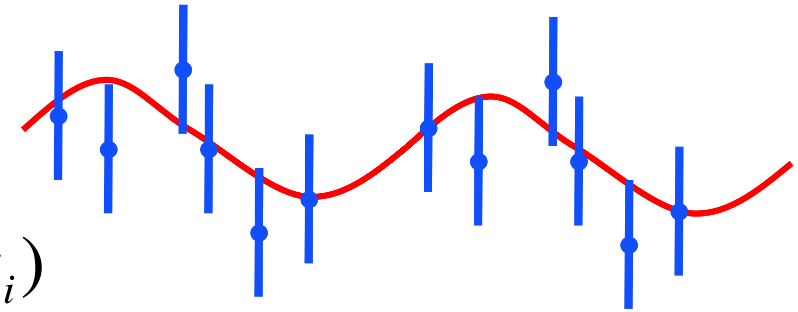
Iterated Optimal Scaling: simple algorithm, easy to code, often adequate.

Example: Sine Curve + Background

Data : $X_i \pm \sigma_i$ at $t = t_\mathrm{i}$

Model : $X(t) = A + S\sin(\omega t) + C\cos(\omega t)$

3 Patterns : 1, $s_i = \sin(\omega t_i)$, $c_i = \cos(\omega t_i)$



Iterated Optimal Scaling:

$$\hat{A} = \frac{\sum \left(X_i - \hat{S}\, s_i - \hat{C}\, c_i \right) / \sigma_i^2}{\sum 1/\sigma_i^2}, \quad \mathrm{Var}\left[\hat{A}\right] \approx \frac{1}{\sum 1/\sigma_i^2}$$

$$\hat{S} = \frac{\sum \left(X_i - \hat{A} - \hat{C}\, c_i \right) s_i / \sigma_i^2}{\sum s_i^2 / \sigma_i^2}, \quad \mathrm{Var}\left[\hat{S}\right] \approx \frac{1}{\sum s_i^2 / \sigma_i^2}$$

$$\hat{C} = \frac{\sum \left(X_i - \hat{A} - \hat{S}\, s_i \right) c_i / \sigma_i^2}{\sum c_i^2 / \sigma_i^2}, \quad \mathrm{Var}\left[\hat{C}\right] \approx \frac{1}{\sum c_i^2 / \sigma_i^2}$$

Variance formulas assume orthogonal parameters, otherwise give error bars too small.

Use inverse of Hessian matrix (see later).

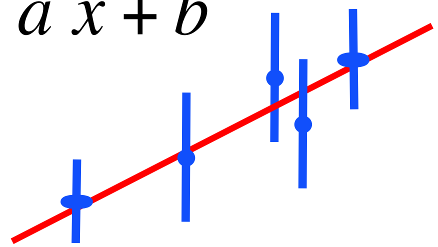
Iterate (if patterns not orthogonal).

χ^2 analysis of the straight line fit

$$\chi^2 \equiv \sum_{i=1}^{N} \left(\frac{y_i - (a\,x_i + b)}{\sigma_i} \right)^2$$

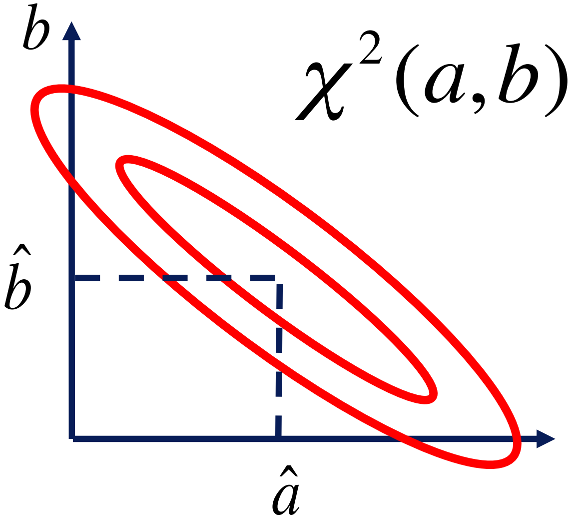
$$0 = \frac{\partial \chi^2}{\partial a} = -2 \sum x \left(y - a\,x - b \right) / \sigma^2$$

$$0 = \frac{\partial \chi^2}{\partial b} = -2 \sum \left(y - a\,x - b \right) / \sigma^2$$



The Normal Equations:

$$a \sum x^2 / \sigma^2 + b \sum x / \sigma^2 = \sum x\,y / \sigma^2$$
$$a \sum x / \sigma^2 + b \sum 1 / \sigma^2 = \sum y / \sigma^2$$



Matrix form:

$$\begin{pmatrix} \Sigma x^2 / \sigma^2 & \Sigma x / \sigma^2 \\ \Sigma x / \sigma^2 & \Sigma 1 / \sigma^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \Sigma x\,y / \sigma^2 \\ \Sigma y / \sigma^2 \end{pmatrix}$$

$$\underline{\underline{H}}\;\underline{\alpha} = \underline{c}(y)$$

(***H*** = Hessian matrix)

Solution: $\hat{\underline{\alpha}} = \underline{\underline{H}}^{-1}\, \underline{c}(y)$

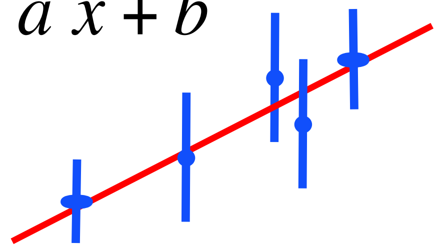
(***c*** = correlation vector)

χ^2 analysis of the straight line fit

Normal Equations: $\underline{\underline{H}}\,\underline{\alpha} = \underline{c}(y)$

$$\begin{pmatrix} \Sigma\, x^2/\sigma^2 & \Sigma\, x/\sigma^2 \\ \Sigma\, x/\sigma^2 & \Sigma\, 1/\sigma^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \Sigma\, x\, y/\sigma^2 \\ \Sigma\, y/\sigma^2 \end{pmatrix}$$

$$y = a\,x + b$$



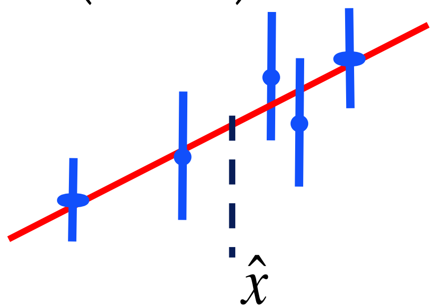
Solution: $\hat{\underline{\alpha}} = \underline{\underline{H}}^{-1}\,\underline{c}(y)$

$$\begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} \Sigma\, 1/\sigma^2 & -\Sigma\, x/\sigma^2 \\ -\Sigma\, x/\sigma^2 & \Sigma\, x^2/\sigma^2 \end{pmatrix} \begin{pmatrix} \Sigma\, x\, y/\sigma^2 \\ \Sigma\, y/\sigma^2 \end{pmatrix}$$

Hessian Determinant: $\Delta = \left(\Sigma\, 1/\sigma^2\right)\left(\Sigma\, x^2/\sigma^2\right) - \left(\Sigma\, x/\sigma^2\right)^2$

Orthogonal basis: $x \Rightarrow (x - \hat{x}) \qquad \hat{x} \equiv \left(\Sigma\, x/\sigma^2\right) \Big/ \left(\Sigma\, 1/\sigma^2\right)$

$$y = a\left(x - \hat{x}\right) + b$$



$$\Sigma\,(x - \hat{x})/\sigma^2 = 0, \qquad \Delta = \left(\Sigma\, 1/\sigma^2\right)\left(\Sigma\,(x - \hat{x})^2/\sigma^2\right)$$

$$\begin{pmatrix} \hat{a} \\ \hat{b} \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} \Sigma\, 1/\sigma^2 & 0 \\ 0 & \Sigma\,(x - \hat{x})^2/\sigma^2 \end{pmatrix} \begin{pmatrix} \Sigma\,(x - \hat{x})\, y/\sigma^2 \\ \Sigma\, y/\sigma \end{pmatrix}$$

$$\hat{a} = \frac{\Sigma\,(x - \hat{x})\, y/\sigma^2}{\Sigma\,(x - \hat{x})^2/\sigma^2} \qquad \hat{b} = \frac{\Sigma\, y/\sigma^2}{\Sigma\, 1/\sigma^2}$$

(Diagonal Hessian Matrix)

(same as Optimal Scaling)

The Hessian Matrix

$$H_{jk} \equiv \frac{1}{2}\frac{\partial^2 \chi^2}{\partial a_j \partial a_k},$$

= half the curvature of the χ^2 landscape

$$\chi^2 \equiv \sum_{i=1}^{N}\left(\frac{y_i - (a\,x_i + b)}{\sigma_i}\right)^2$$

$$\frac{\partial \chi^2}{\partial a} = -2\sum x\left(y - a\,x - b\right)/\sigma^2$$

$$\frac{\partial \chi^2}{\partial b} = -2\sum \left(y - a\,x - b\right)/\sigma^2$$

- Example: $y = a\,x + b$.

$$\frac{\partial^2 \chi^2}{\partial a^2} = 2\sum_i x_i^2/\sigma_i^2 \qquad \frac{\partial^2 \chi^2}{\partial a\,\partial b} = 2\sum_i x_i/\sigma_i^2$$

$$\frac{\partial^2 \chi^2}{\partial b^2} = 2\sum_i 1/\sigma_i^2, \quad \text{so} \;\; H = \begin{bmatrix} \sum_i x_i^2/\sigma_i^2 & \sum_i x_i/\sigma_i^2 \\ \sum_i x_i/\sigma_i^2 & \sum_i 1/\sigma_i^2 \end{bmatrix}$$

For linear models, Hessian matrix is independent of the parameters, and χ^2 surface is parabolic.

Parameter Uncertainties

Hessian matrix describes the **curvature** of the χ^2 surface :

$$\chi^2(\alpha) = \chi^2(\hat{\alpha}) + \sum_{j,k} \left(\alpha_j - \hat{\alpha}_j\right) H_{jk} \left(\alpha_k - \hat{\alpha}_k\right) + \ldots$$

$$H_{jk} \equiv \frac{1}{2} \frac{\partial^2 \chi^2}{\partial a_j \partial a_k},$$

For linear models, Hessian matrix is independent of the parameters, and χ^2 surface is parabolic.

For a one-parameter fit:

$$\text{if } \hat{\alpha} \text{ minimizes } \chi^2, \text{ then } \mathrm{Var}(\hat{\alpha}) = \frac{2}{\partial^2 \chi^2 / \partial \alpha^2}.$$

For a multi-parameter fit the covariance of any pair of parameters is an element of the **inverse-Hessian matrix**:

$$\mathrm{Cov}(a_j, a_k) = \left[H^{-1} \right]_{jk}$$

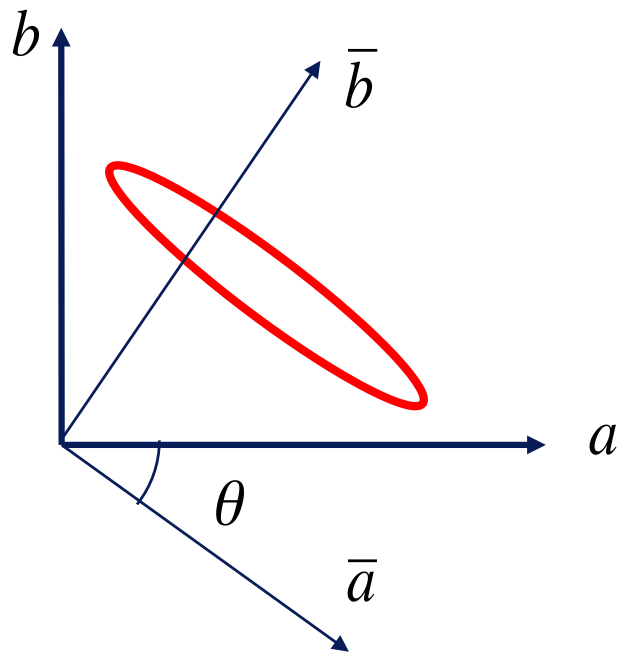
Principal Axes of the χ^2 Ellipsoid

Eigenvectors of *H* define the **principal axes** of the χ^2 ellipsoid.

Equivalent to **rotating** the coordinate system in parameter space.

$$y = a\,x + b$$

$$= \bar{a}\left(x\cos\theta - \sin\theta\right) + \bar{b}\left(x\sin\theta + \cos\theta\right)$$



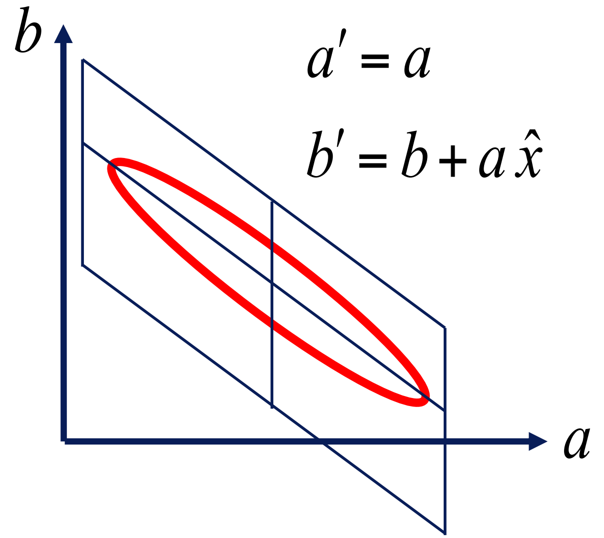
Note that **orthogonal patterns are not unique**.

Can also diagonalise *H* by :

$$a\,x + b \quad \rightarrow \quad a'(x - \hat{x}) + b'$$

This "**shears**" the parameter space, giving

$$H = \begin{bmatrix} \sum_i (x_i - \hat{x})^2 / \sigma_i^2 & 0 \\ 0 & \sum_i 1/\sigma_i^2 \end{bmatrix}$$



Diagonalising the Hessian matrix orthogonalises the parameters.

General Linear Regression
Scale *M* Patterns

Linear Model: $y(x) = a_1 P_1(x) + a_2 P_2(x) + \ldots = \sum_k^M a_k P_k(x)$

Example: Polynomial: $y(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{M-1} x^{M-1}$

$$\chi^2 \equiv \sum_{i=1}^{N} \left[\frac{y_i - y(x_i)}{\sigma_i} \right]^2 = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(y_i - \sum_j^M a_j P_j(x_i) \right)^2$$

Normal Equations:

$$0 = \frac{\partial \chi^2}{\partial a_k} = -2 \sum_i^N \left(y_i - \sum_j^M a_j P_j(x_i) \right) \frac{P_k(x_i)}{\sigma_i^2} \qquad k = 1 \ldots M$$

$$\sum_j^M \left(\sum_i^N \frac{P_{ji} P_{ki}}{\sigma_i^2} \right) (a_j) = \sum_i^N \frac{y_i P_{ki}}{\sigma_i^2} \qquad P_{ki} \equiv P_k(x_i)$$

$$\sum_j^M H_{jk} a_j = c_k(y) \qquad H_{jk} = \sum_i^N \frac{P_{ji} P_{ki}}{\sigma_i^2} \qquad c_k(y) = \sum_i^N \frac{y_i P_{ki}}{\sigma_i^2}$$

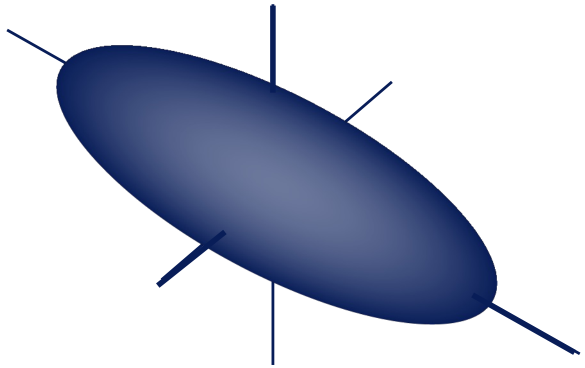
Principal Axes for general Linear Models

- In the general linear case we fit M functions $P_k(x)$ with scale factors a_k:

$$y(x) = \sum_{k=1}^{M} \alpha_k P_k(x)$$

- The (M x M) Hessian matrix has elements:

$$H_{jk} \equiv \frac{1}{2} \frac{\partial^2 \chi^2}{\partial \alpha_j \partial \alpha_k} = \sum_{i=1}^{N} \frac{P_j(x_i) P_k(x_i)}{\sigma_i^2}$$



- Normal equations (M equations for M unknowns):

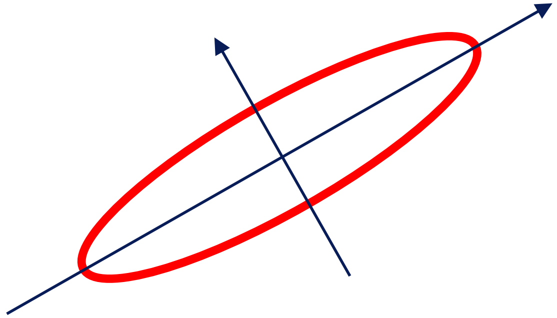
$$\sum_{k=1}^{M} H_{jk} \alpha_k = c_j \quad \text{where} \quad c_j = \sum_{i=1}^{N} \frac{y_i P_j(x_i)}{\sigma_i^2}$$

- This gives M-dimensional ellipsoidal surfaces of constant χ^2 whose principal axes are the M eigenvectors of the Hessian matrix H.
- Use standard matrix methods to find linear combinations of P_i that diagonalise H. (More details later…)

Linear vs Non-Linear Models

Linear Model : $y(x) = \sum_{k}^{M} \alpha_k \, P_k(x)$

M scale parameters α_k



$$H_{jk} \equiv \frac{1}{2} \frac{\partial^2 \chi^2}{\partial \alpha_j \, \partial \alpha_k} = \sum_{i=1}^{N} \frac{P_j(x_i)\, P_k(x_i)}{\sigma_i^2}$$

Elliptical χ^2 contours, unique solution by linear regression (matrix inversion).

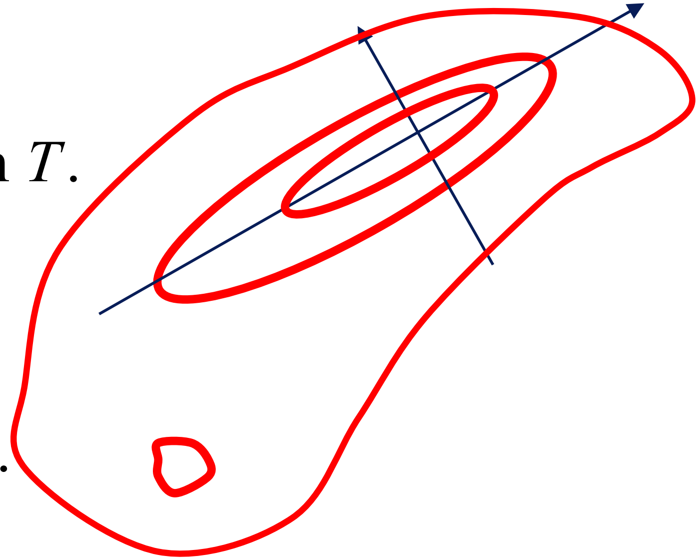
Non - Linear Models :

power-law: $y = A\, x^B$. Linear in A, non-linear in B.

blackbody: $f_\nu = \Omega\, B_\nu(\lambda, T)$. Linear in Ω, non-linear in T.

$$\chi^2(\alpha) = \chi^2(\hat{\alpha}) + \sum_{j,k} \left(\alpha_j - \hat{\alpha}_j\right) H_{jk} \left(\alpha_k - \hat{\alpha}_k\right) + \ldots$$

$H_{jk} \equiv \frac{1}{2} \frac{\partial^2 \chi^2}{\partial \alpha_j \, \partial \alpha_k}$ depends on the non-linear parameters.



Skewed or banana-shaped contours, multiple local minima, require **iterative methods**.

Method 1: Linearise the Non-Linear Model

Linearisation: use local linear approximation to the model, giving a quadratic approximation to χ^2 surface. Solve by linear regression, then iterate.

$$\text{Example: gaussian peak + background:}$$

$$\mu = A\,g + B \qquad g \equiv e^{-\eta^2/2} \qquad \eta \equiv \frac{x - x_0}{\sigma}$$

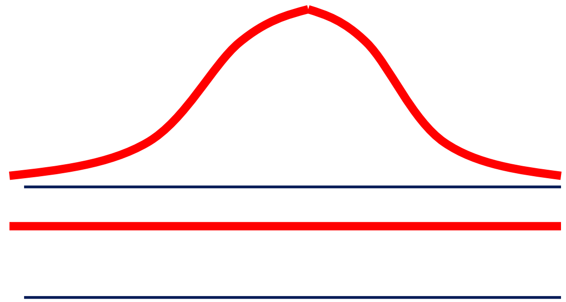
$$\Delta\mu \approx \Delta A \frac{\partial \mu}{\partial A} + \Delta B \frac{\partial \mu}{\partial B} + \Delta x_0 \frac{\partial \mu}{\partial x_0} + \Delta \sigma \frac{\partial \mu}{\partial \sigma}$$

$$\frac{\partial \mu}{\partial A} = g \qquad \frac{\partial \mu}{\partial x_0} = A\,g\,\eta/\sigma$$

$$\frac{\partial \mu}{\partial B} = 1 \qquad \frac{\partial \mu}{\partial \sigma} = A\,g\,\eta^2/\sigma$$

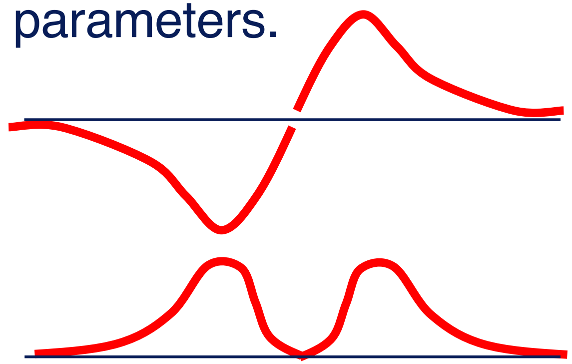
A and *B* are scale parameters.

$$\frac{\partial \mu}{\partial A} = g \qquad \frac{\partial \mu}{\partial B} = 1$$



x_0 and σ are non-linear parameters.

$$\frac{\partial \mu}{\partial x_0} \qquad \frac{\partial \mu}{\partial \sigma}$$



Guess x_0 and σ, fit linear parameters *A* and *B*, evaluate derivatives, adjust x_0 and σ using linear approximation, iterate.

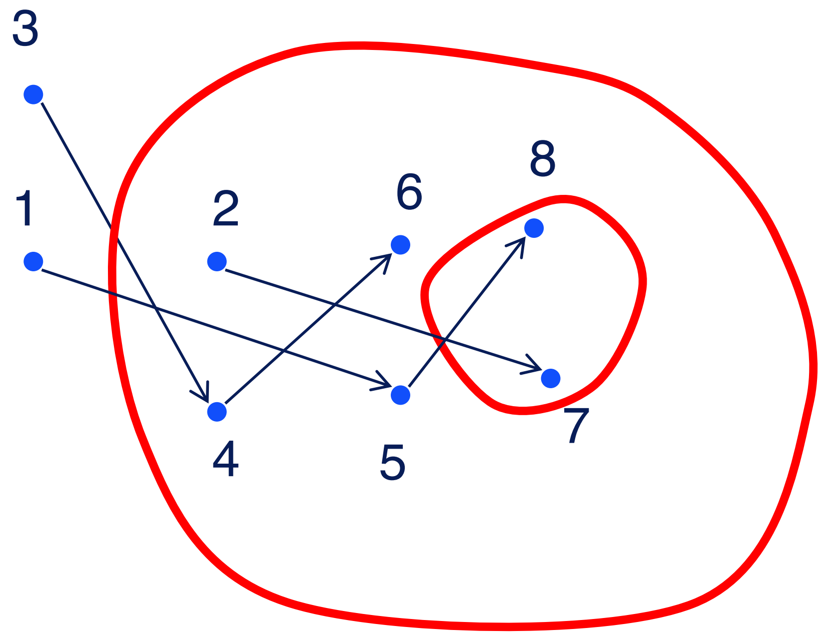
(**Levenberg-Marquadt** method: add constant to Hessian diagonal to prevent over-stepping. See e.g. Numerical Recipes.

Method 2: Amoeba (Downhill Simplex)

Amoeba (downhill simplex)

Simplex = cluster of *M+1* points in the *M*-dimensional parameter space.

1. Evaluate χ^2 at each node.
2. Pick node with highest χ^2, move it on a line thru the centroid of the other *M* nodes, using simple rules to find new place with lower χ^2 .
3. Repeat until converged.



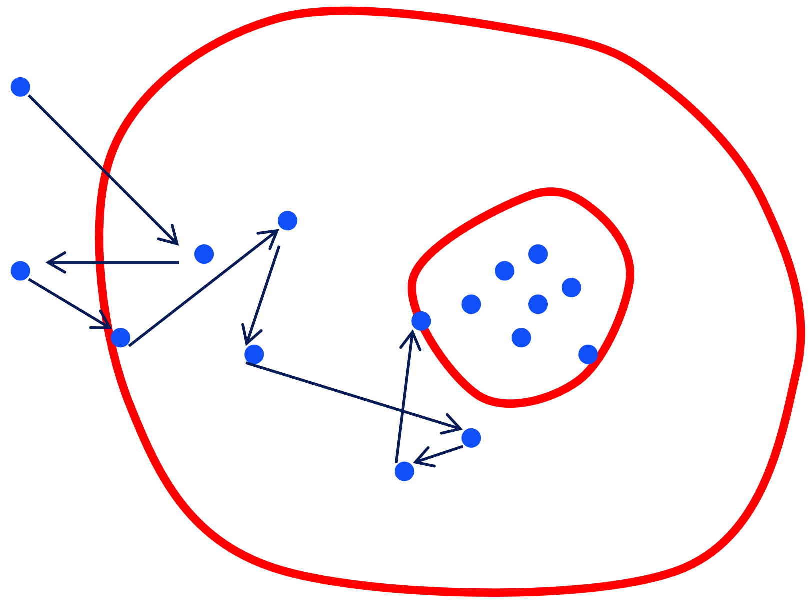
Amoeba requires no derivatives ☺

Amoeba "crawls" downhill, adjusting shape to match the χ^2 landscape, then shrinks down onto a local minimum.

See e.g. Numerical Recipes for full description.

Method 3: Markov Chain Monte Carlo (MCMC)

1. Start somewhere in the *M*-dimensional parameter space. Guess parameters α_i



2. Estimate σ_i for each parameter (e.g. covariance matrix from last *n* points).

3. Take a **random step**, e.g. using a Gaussian random number with same σ_i (and covariances) as "recent" points.

$$\Delta\alpha_i \sim G(0,\sigma_i^{\,2})$$

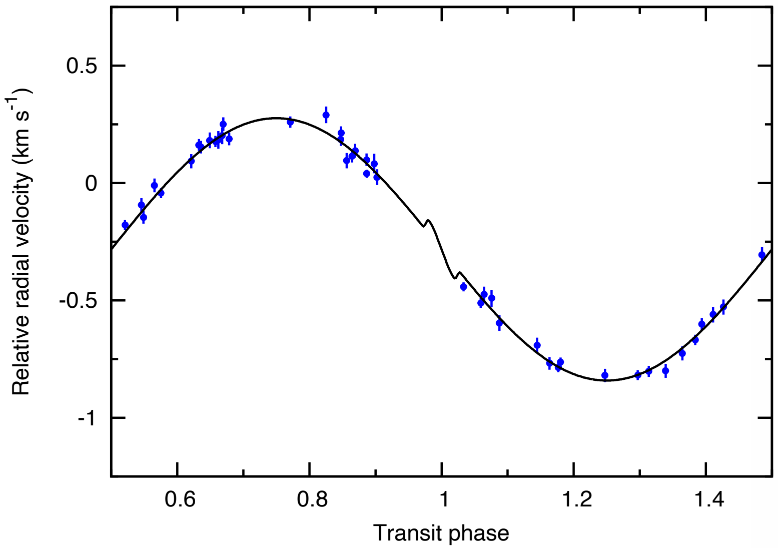
4. Evaluate $\Delta\chi^2 = \chi^2_{new} - \chi^2_{old}$ and keep the step with probability $P = \min\left[\,1,\ \exp\left(-\Delta\chi^2/2\right)\,\right]$

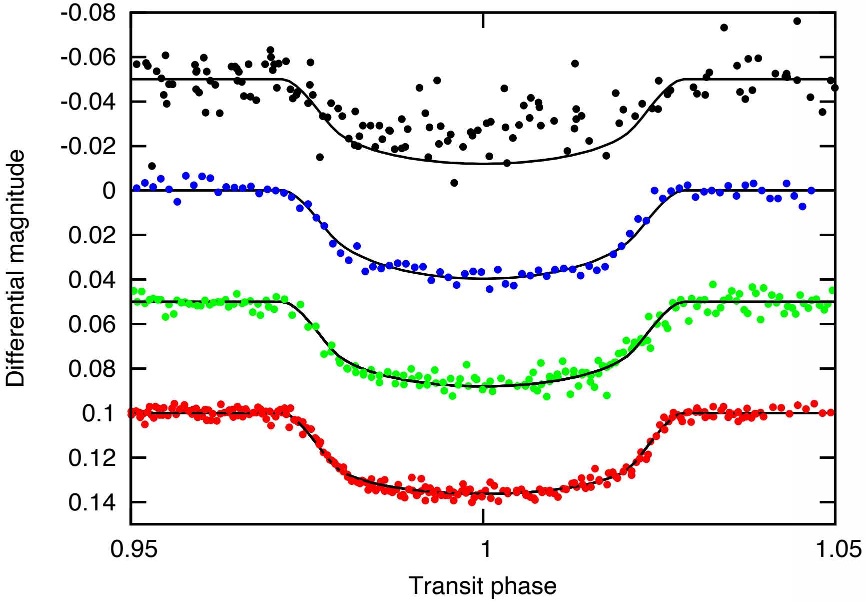
5. Iterate steps 2-4 until "convergence".

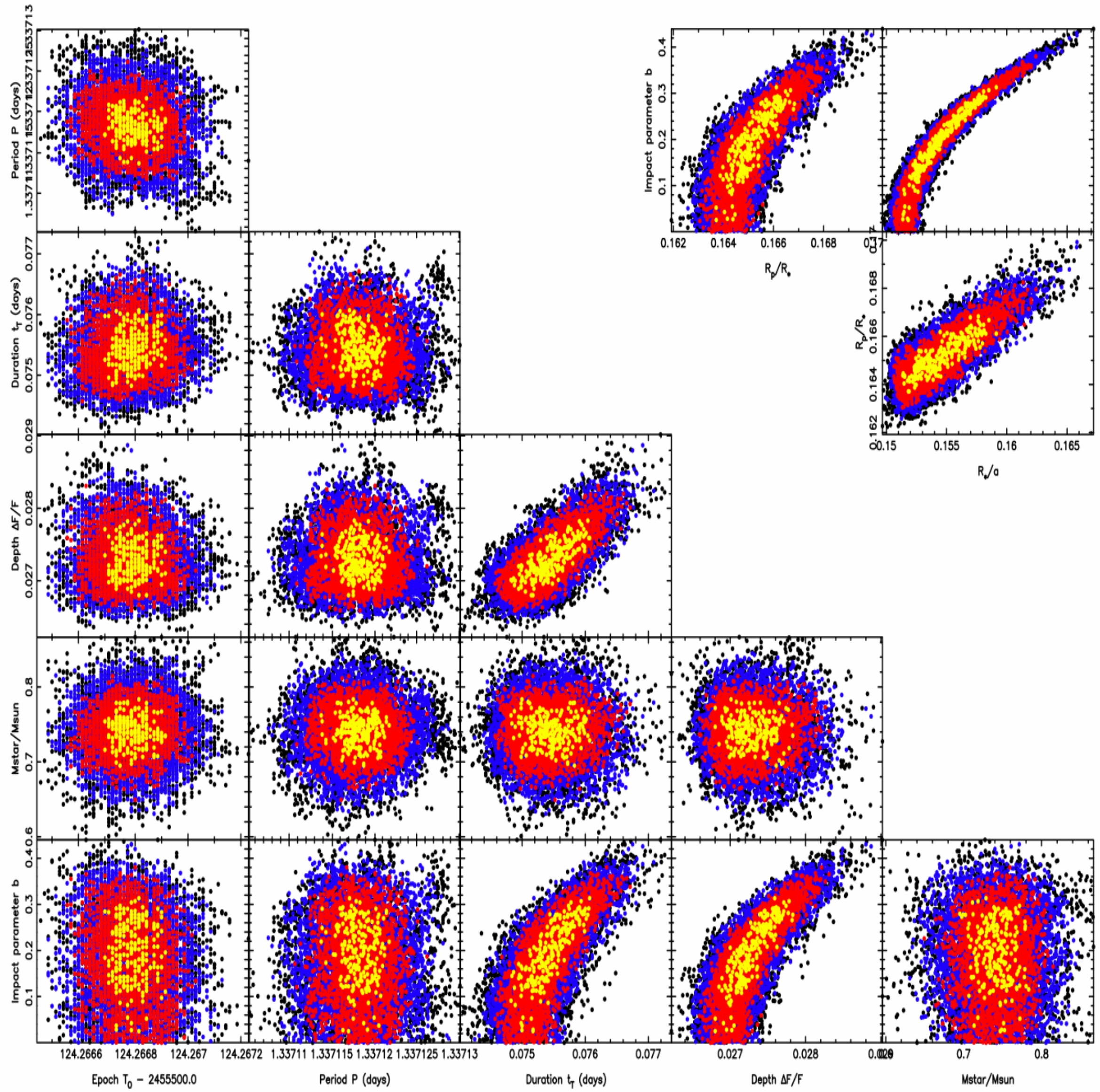
MCMC requires no derivatives ☺ Easy to code ☺

MCMC generates a "chain" of points tending to move downhill, then settling into a pattern matching the full **posterior distribution** of the parameters. ☺
Can escape from local minima. ☺
Can also include prior distributions on the parameters.

Example: MCMC fit of exoplanet model to transit lightcurves and radial velocity curve data.







Fini -- ADA 09