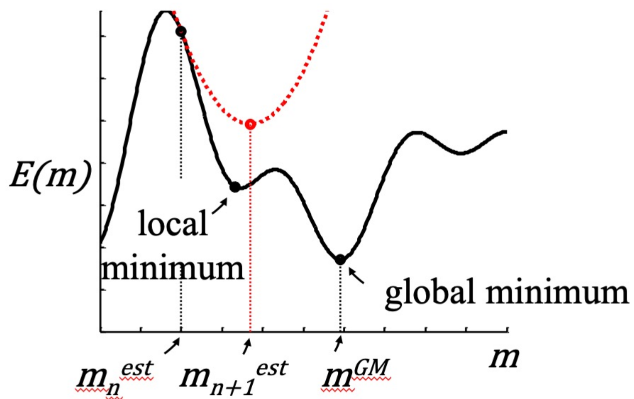
GEOL/PHYS 6670
Geophysical Inverse Theory

Lecture 5, September 21



Note to self
Don't forget to
Start recording

Prof. Anne Sheehan
Anne.Sheehan @ colorado.edu
Office: Benson Rm 440A

Office Hours M 3:30-4:20, Tues 11-11:50

Homework 4 – due now

fitting a parabola with least squares

Homework 5 – for next week

fitting a parabola, this time constraining it to go through a point

Class presentations of papers from the literature

Today – Shane, Bayesian Monte Carlo Inversion

next week - Jonah, Optimal estimation

Term paper – handout on Canvas. Let me know if you have any questions. Paragraph on topic due on Oct 19.

Last time –
Chapter 3 –

Least squares, Weighted Least squares,
Damped least squares, Constrained least squares

Today (lecture 5)

Review of last time, *especially constrained least squares since that is on this week's homework*

Nonlinear inversion preview, Taylor series example (Ch 9)

Ch 4 – Generalized inverse (data resolution, model resolution)

Shane – Bayesian Monte Carlo Inversion

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

Simple least squares

$$\mathbf{m}^{est} = \mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-1}\mathbf{d}$$

Minimum Length

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G} + \varepsilon^{2}\mathbf{I}]^{-1}\,\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

Damped least squares

$$\mathbf{m}_{\mathrm{WLS}} = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_{e}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{W}_{e}\mathbf{d}$$

Weighted least squares

Weighted damped least squares

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_{e}\mathbf{G} + \varepsilon^{2}\mathbf{W}_{m}]^{-1}\,[\mathbf{G}^{\mathrm{T}}\mathbf{W}_{e}\,\mathbf{d} + \varepsilon^{2}\mathbf{W}_{m}\langle\mathbf{m}\rangle\,]$$

Weighted least squares
minimize E where

$$E = \mathbf{e}^{\mathrm{T}} \mathbf{W}_e \mathbf{e}$$

Weighted least squares solution

$$\mathbf{m}_{\mathrm{WLS}} = [\mathbf{G}^{\mathrm{T}} \mathbf{W}_e \mathbf{G}]^{-1} \mathbf{G}^{\mathrm{T}} \mathbf{W}_e \mathbf{d}$$

$\mathbf{W_e}$ error weight matrix, can represent one data type being more accurate than another

example
when d_3 is more accurately measured than the other data

$$\mathbf{W}_e = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 2 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

weighting matrix,

often use

$$W_e = \sigma^{-1} I$$

$$= \begin{pmatrix} 1/\sigma_1 & & & & 0 \\ & 1/\sigma_2 & & & \\ & & 1/\sigma_3 & & \\ & 0 & & \ddots & \\ & & & & 1/\sigma_n \end{pmatrix}$$

ie small variance, higher weighting

example of a common weighting matrix

Damped least squares, minimize

$$\Phi(\mathbf{m}) = E + \varepsilon^2 L = \mathbf{e}^{\mathrm{T}}\mathbf{e} + \varepsilon^2 \mathbf{m}^{\mathrm{T}}\mathbf{m}$$

Leads to

damped least-squares solution

$$\mathbf{m}^{\mathrm{est}} = \ [\mathbf{G}^{\mathrm{T}}\mathbf{G} + \varepsilon^2 \mathbf{I}]\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

Very similar to least-squares

$$\mathbf{m}^{\mathrm{est}} = \ [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

Just add ε^2 to diagonal of $\mathbf{G}^{\mathrm{T}}\mathbf{G}$

'L-curve'

To examine tradeoff between data misfit and model length with changing damping parameter

(following from last time, 2)

tradeoff curve

1b

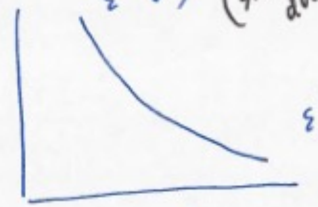
Damped LS

reminder of what damping parameter ε^2 does

follow-up to last time

ε^2 big → m^Tm being small matters the most (smooth models, doesn't fit data)

data misfit e^Te



ε^2 small → e^Te being small matters the most (rough models, small misfit)

m^Tm

damped LS minimize

$$\phi = E + \varepsilon^2 L$$

$$\phi = e^Te + \varepsilon^2 m^Tm$$

$$\boxed{m = (G^TG + \varepsilon^2 I)^{-1} G^T d}$$

if G^TG singular this will still work

aside:

Useful matrix identities

$(A+B)^T = A^T + B^T$

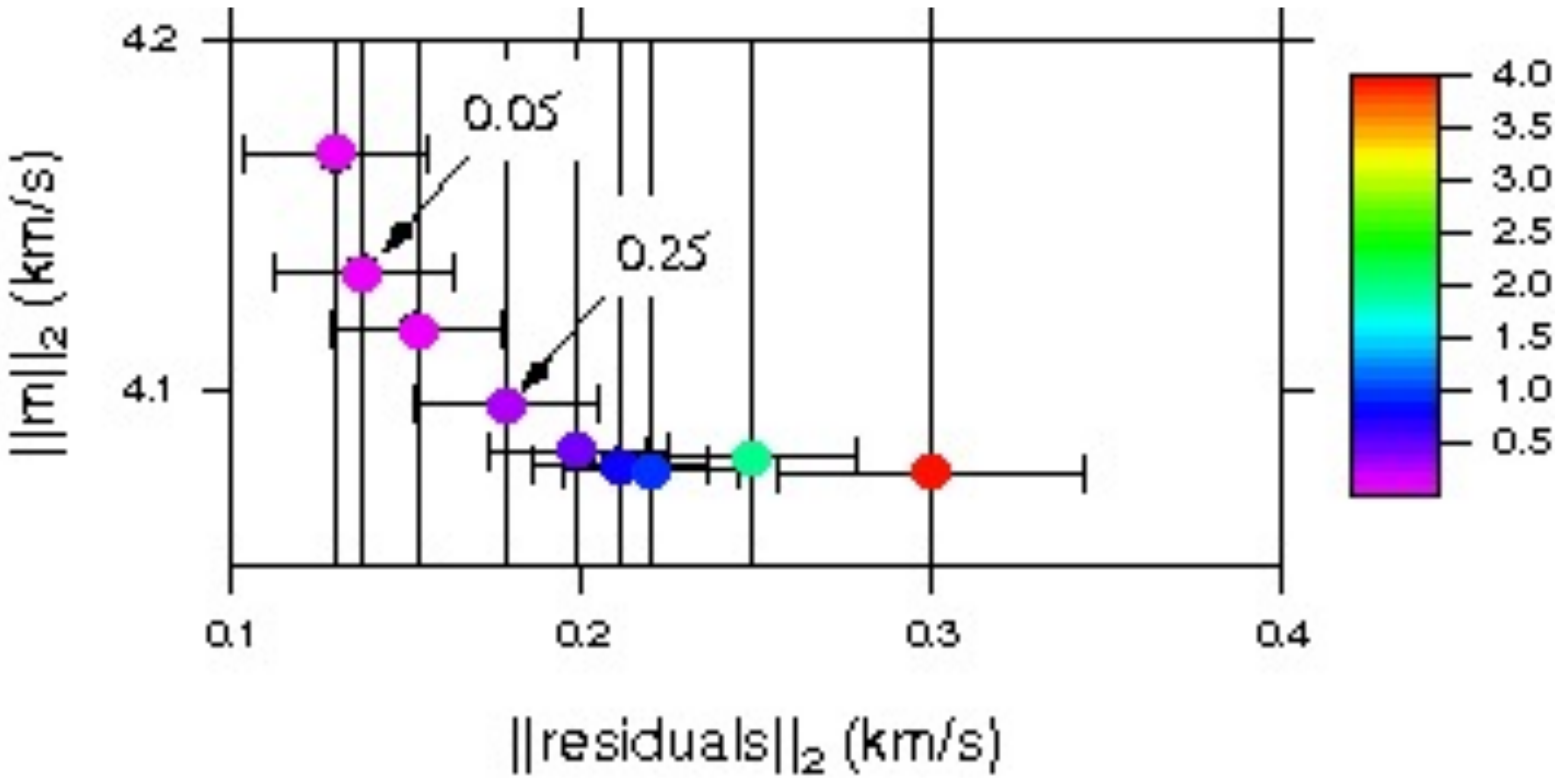
$(AB)^T = B^T A^T$

$(A^TA)^T = A^TA$

$(B^TAB)^T = B^TAB$

$(AB)^{-1} = B^{-1}A^{-1}$

Example of an ‘L-curve’



L-curve between the L-2 norm of the residuals and the L-2 norm of model (model energy) for different damping parameters (color-coded) for grid nodes in the central Sierras. Two particular damping parameters are noted (0.05 and 0.25). The model with damping of 0.05 is preferred because the resulting model reveals velocity structures and gradients more consistent with prior knowledge, in particular the negative velocity gradients found at about 80 km depth in the Frassetto Pds

http://research.flyrok.org/snep.html

Constrained inversion

Solve $Gm=d$ with constraint that $Fm=h$

Constrained inversion

Solve Gm=d with constraint that Hm=h

Example, require a model parameter, m_k, to have a certain value

$$\mathbf{Hm} = [0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0] \begin{bmatrix} m_1 \\ \vdots \\ m_k \\ \vdots \\ m_M \end{bmatrix} = [h_k] = \mathbf{h} \tag{3.60}$$

Another method of implementing the constraints is through the use of Lagrange multipliers. One minimizes $E = \mathbf{e}^{\mathrm{T}}\mathbf{e}$ with the constraint that $\mathbf{Hm} - \mathbf{h} = 0$ by forming the function

$$\Phi(m) = \sum_{i=1}^{N}\left[\sum_{j=1}^{M} G_{ij}m_j - d_i\right]^2 + 2\sum_{i=1}^{p}\lambda_i\left[\sum_{j=1}^{M} H_{ij}m_j - h_i\right] \tag{3.61}$$

(where there are p constraints and $2\lambda_i$ are the Lagrange multipliers) and setting its derivatives with respect to the model parameters to zero as

$$\frac{\partial\Phi(\mathbf{m})}{\partial m_q} = 2\sum_{i=1}^{M} m_i \sum_{j=1}^{N} G_{jq}G_{ji} - 2\sum_{i=1}^{N} G_{iq}d_i - 2\sum_{i=1}^{p}\lambda_i H_{iq} \tag{3.62}$$

This equation must be solved simultaneously with the constraint equations $\mathbf{Hm} = \mathbf{h}$ to yield the estimated solution. These equations, in matrix form, are

$$\begin{bmatrix} \mathbf{G}^{\mathrm{T}}\mathbf{G} & \mathbf{H}^{\mathrm{T}} \\ \mathbf{H} & 0 \end{bmatrix}\begin{bmatrix} \mathbf{m} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{G}^{\mathrm{T}}\mathbf{d} \\ \mathbf{h} \end{bmatrix} \tag{3.63}$$

Although these equations can be manipulated to yield an explicit formula for $\mathbf{m}^{\mathrm{est}}$, it is often more convenient to solve directly this $M + p$ system of equations for M estimates of model parameters and p Lagrange multipliers by premultiplying by the inverse of the square matrix.

3.10.1 Example: Constrained Fitting of a Straight Line

Consider the problem of fitting the straight line $d_i = m_1 + m_2 z_i$ to data, where one has prior information that the line must pass through the point (z', d') (Fig. 3.13). There are two model parameters: intercept m_1 and slope m_2. The $p = 1$ constraint is that $d' = m_1 + m_2 z'$, or

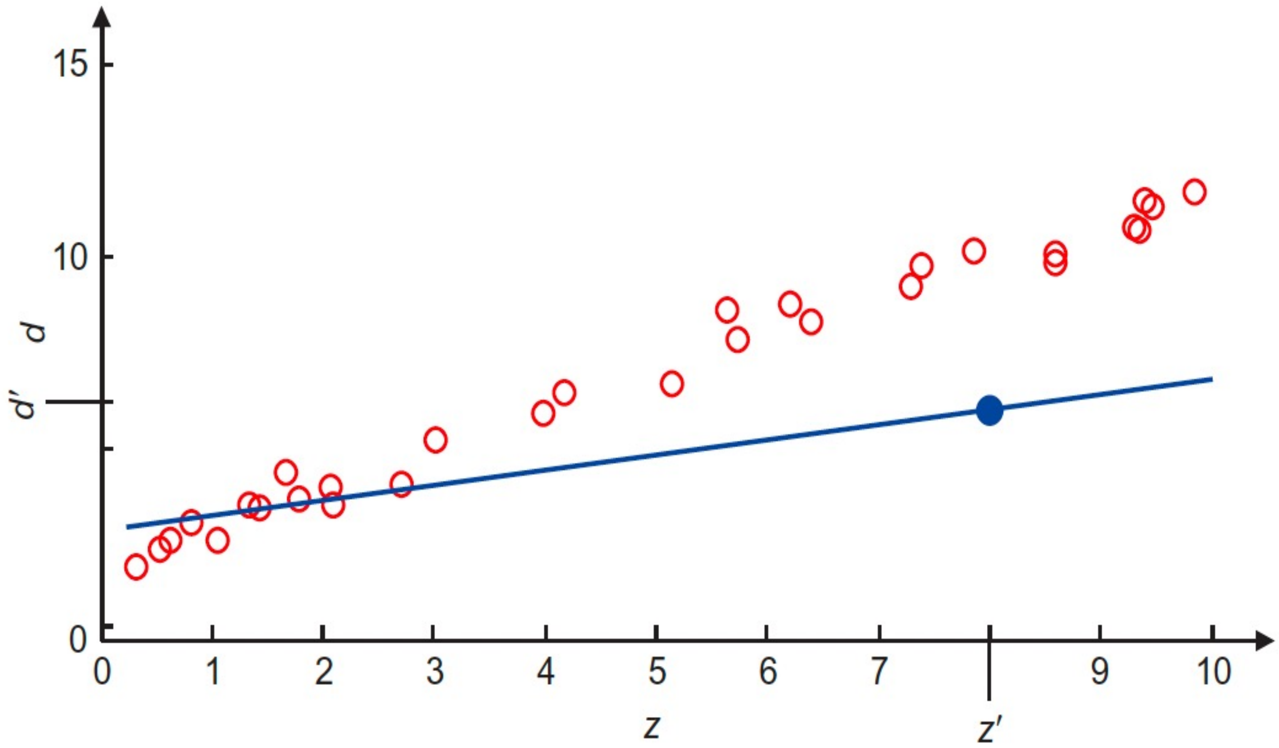


FIG. 3.13 Least squares fitting of a straight line to (z,d) data, where the line is constrained to pass through the point $(z',d') = (8,6)$. *MatLab* script gda03_11.

This might be useful for the homework

3.10.1 Example: Constrained Fitting of a Straight Line

$$\mathbf{H}\mathbf{m} = [1 \;\; z'] \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = [d'] = \mathbf{h} \tag{3.64}$$

Using the $\mathbf{G}^{\mathrm{T}}\mathbf{G}$ and $\mathbf{G}^{\mathrm{T}}\mathbf{d}$ computed in Section 3.5.1, the solution is

$$\begin{bmatrix} m_1^{\text{est}} \\ m_2^{\text{est}} \\ \lambda_1 \end{bmatrix} = \begin{bmatrix} N & \sum_{i=1}^{N} z_i & 1 \\ \sum_{i=1}^{N} z_i & \sum_{i=1}^{N} z_i^2 & z' \\ 1 & z' & 0 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{N} d_i \\ \sum_{i=1}^{N} z_i d_i \\ d' \end{bmatrix} \tag{3.65}$$

3.10.1 Example: Constrained Fitting of a Straight Line

constrained line fit, w/ #5

2e

$$y = a + bx$$

constrain inversion to go through, ~~point 3,~~ x-y

point 3,4

x y

so

$$y = a + bx$$

$$4 = a + 3b$$

$$F\,m = \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = h_1 = \cancel{3}\,4$$

3.10.1 Example: Constrained Fitting of a Straight Line

$$F\,m = \begin{bmatrix} 1 & 3 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = h_1 = \cancel{3}\;4$$

$$\begin{bmatrix} G^T G & F^T \\ F & 0 \end{bmatrix} \begin{bmatrix} m \\ \lambda \end{bmatrix} = \begin{bmatrix} G^T d \\ h \end{bmatrix}$$

$$\underset{3\times 3}{\begin{bmatrix} \overset{2\times 2 \text{ matrix}}{\boxed{G^T G}} & \begin{matrix} 1 \\ 3 \end{matrix} \\ \begin{matrix} 1 & 3 \end{matrix} & 0 \end{bmatrix}} \underset{3\times 1}{\begin{bmatrix} a \\ b \\ \lambda \end{bmatrix}} = \underset{3\times 1}{\begin{bmatrix} \overset{2\times 1 \text{ vector}}{\begin{bmatrix} G^T d \end{bmatrix}} \\ h \end{bmatrix}}$$

Solving TWO systems of equations, both Gm=d and Fm=h

Preview of nonlinear inverse problems

Approaches for solving nonlinear inverse problems

- Reparameterize

or

- Grid search – brute force solve a ton of forward problems (or fancy randomized grid search such as Genetic Algorithms)

or

- Linearize about an initial guess via Taylor series

For grid search to be effective

The total number of model parameters are small, say $M<7$. The grid is M-dimensional, so the number of trial solution is proportional to L^M, where L is the number of trial solutions along each dimension of the grid.

The solution is known to lie within a specific range of values, which can be used to define the limits of the grid.

The forward problem $\mathbf{d}=\mathbf{g}(\mathbf{m})$ can be computed rapidly enough that the time needed to compute L^M of them is not prohibitive.

The error function $E(\mathbf{m})$ is smooth over the scale of the grid spacing, Δm, so that the minimum is not missed through the grid spacing being too coarse.

Solving a nonlinear inverse problem via Taylor Series expansion

(a way to linearize and solve for perturbations relative to an initial guess)

Solving a nonlinear inverse problem via Taylor Series expansion

(a way to linearize and solve for perturbations relative to an initial guess)

Taylor series expansion
f(x) can be approximated by

$$f(x) =$$

$$f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \ldots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \ldots.$$

Taylor Series:

$$f(m_0)\ +$$

$$d = f(m) = \left.\frac{df}{dm_1}\right|(m - m_0)$$

$$+ \left.\frac{df}{dm_2}\right|(m - m_0)$$

(partial derivatives)

$$+ \quad \frac{d^2 f}{dm_1^2}(m - m_0)^2 + \cdots$$

$$= f(m_0) + \sum_{j=1}^{m} \frac{df(m)}{dm_j}(m - m_0)$$

$$(f(m_0) = initial\ guess)$$

$$= f(m_0) + G_0(m - m_0)$$

$$f(m) - f(m_0) = G_0(m - m_0)$$

$$\Delta d = G\Delta m \quad iterate\ with\ new\ m_0$$

Steps for nonlinear iterative inversions:

1) make initial guess of model parameters, m_0
2) calculate partial derivatives evaluated at m_0, this gives the matrix G
3) Since $G\Delta m = \Delta d$

   solve for $\Delta m = (G^T G)^{-1} G^T \Delta d$
4) solve $m_1 = m_0 + \Delta m$

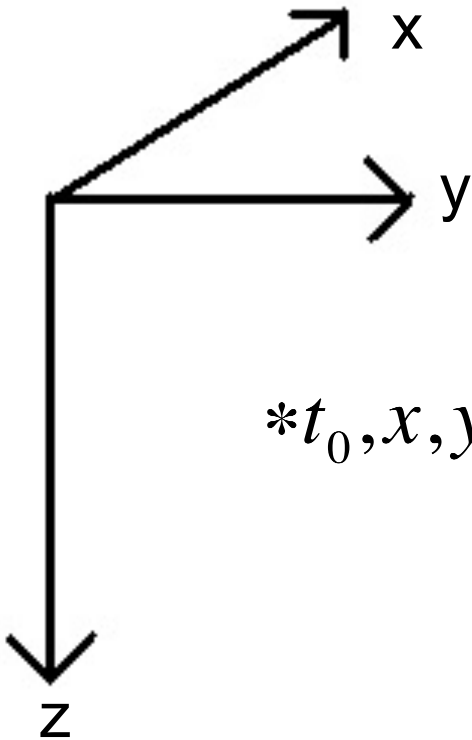
   let this be the new model, go to step (1) and iterate

   stop when residual $\sum \left(d_{pred} - d_{obs}\right)^2$

   is small enough.

Nonlinear example:

<u>earthquake hypocenter location</u>



$(x_i, y_i) \rightarrow t_i$ *is travel time at* i^{th} *station*

$*t_0, x, y, z$

(1) Time @ i^{th} station

$$t_i = t_0 + \frac{r_i}{\alpha}$$

Assuming station elevation is zero

$$= t_0 + \frac{\left[(x - x_i)^2 + (y - y_i)^2 + z^2\right]^{\frac{1}{2}}}{\alpha}$$

Want to solve for t_0, x, y, z

nonlinear

guess location $(\hat{x}, \hat{y}, \hat{z}, \hat{t}_0) = \ell_0$

this guess implies time t_i at station i

(2)

$$t_i = \hat{t}_0 + \left.\frac{d\hat{t}_i}{dt_0}\right|_{\ell_0}(t_0 - \hat{t}_0)$$

$$+ \left.\frac{d\hat{t}_i}{dx}\right|_{\ell_0}(x - \hat{x}) + \left.\frac{d\hat{t}_i}{dy}\right|_{\ell_0}(y - \hat{y})$$

$$+ \left.\frac{d\hat{t}_i}{dz}\right|_{\ell_0}(z - \hat{z})$$

$$t_i = t_0 + \frac{r_i}{\alpha}$$

$$= t_0 + \frac{\left[\left(x - x_i\right)^2 + \left(y - y_i\right)^2 + z^2\right]^{\frac{1}{2}}}{\alpha}$$

(3) Take derivative of equation (1)

$$\left.\frac{dt_i}{dt_0}\right|_{\ell_0} = 1$$

$$\left.\frac{dt_i}{dx}\right|_{\ell_0} = \frac{1}{\alpha} \left.\frac{\hat{x} - x_i}{r_i}\right|_{\ell_0} \quad etc.$$

Observations are arrival times at *n* stations

In form Gm = d,

$$\underbrace{\begin{pmatrix} 1 & \frac{1}{\alpha}\frac{\hat{x} - x_i}{r_i} & \frac{1}{\alpha}\frac{\hat{y} - y_i}{r_i} & \frac{1}{\alpha}\frac{\hat{z} - z_i}{r_i} \\ 1 & & & \\ 1 & & & \\ 1 & & & \end{pmatrix}}_{G} \underbrace{\begin{pmatrix} t_0 - \hat{t}_0 \\ x - \hat{x} \\ y - \hat{y} \\ z - \hat{z} \end{pmatrix}}_{m} = \underbrace{\begin{pmatrix} t_1 - \hat{t}_1 \\ t_2 - \hat{t}_2 \\ \\ t_n - \hat{t}_n \end{pmatrix}}_{d}$$

Steps for nonlinear iterative inversions:

1) make initial guess of model parameters, m_0
2) calculate partial derivatives evaluated at m_0, this gives the matrix G
3) Since $G\Delta m = \Delta d$

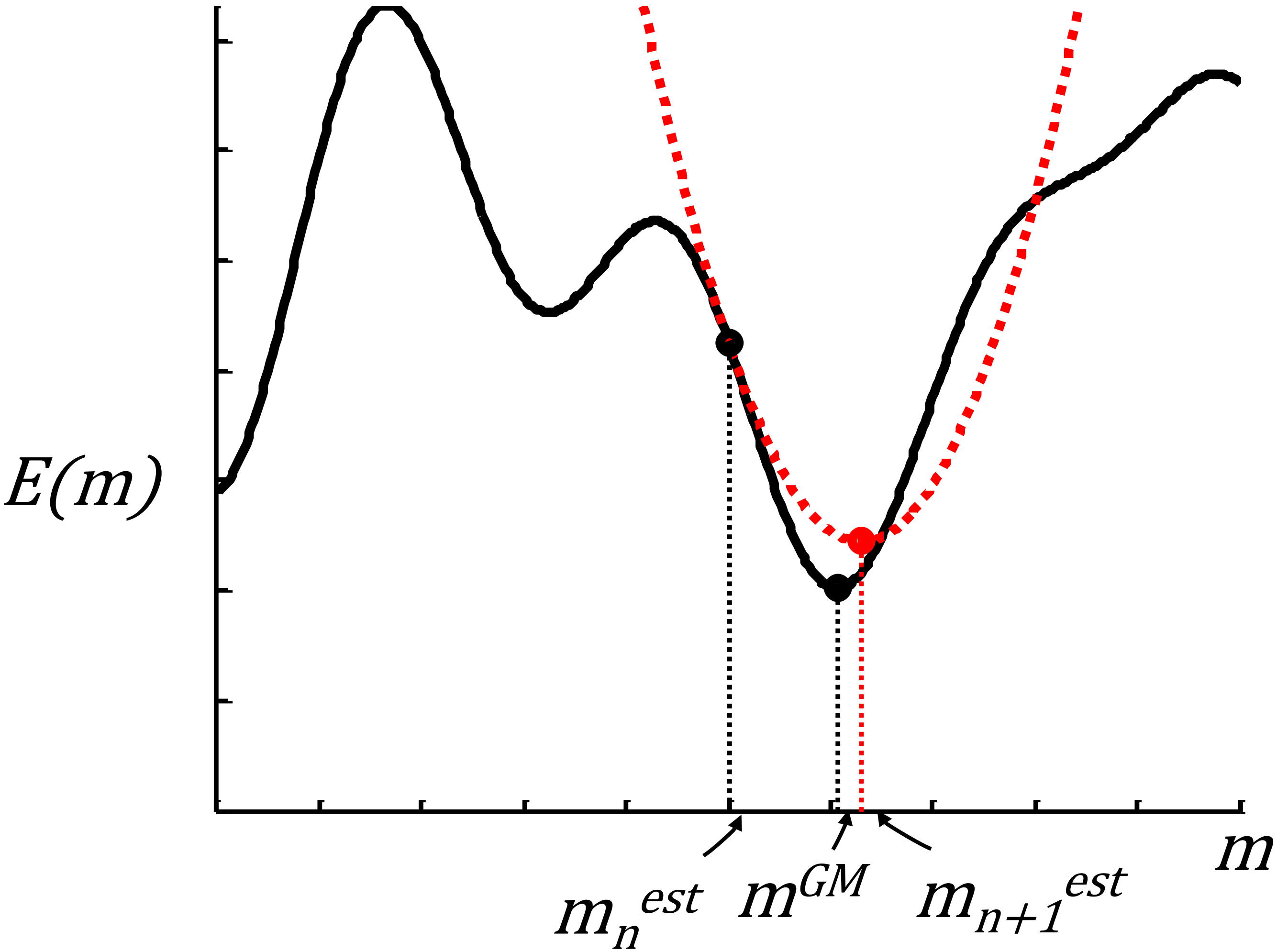
   solve for $\Delta m = (G^T G)^{-1} G^T \Delta d$

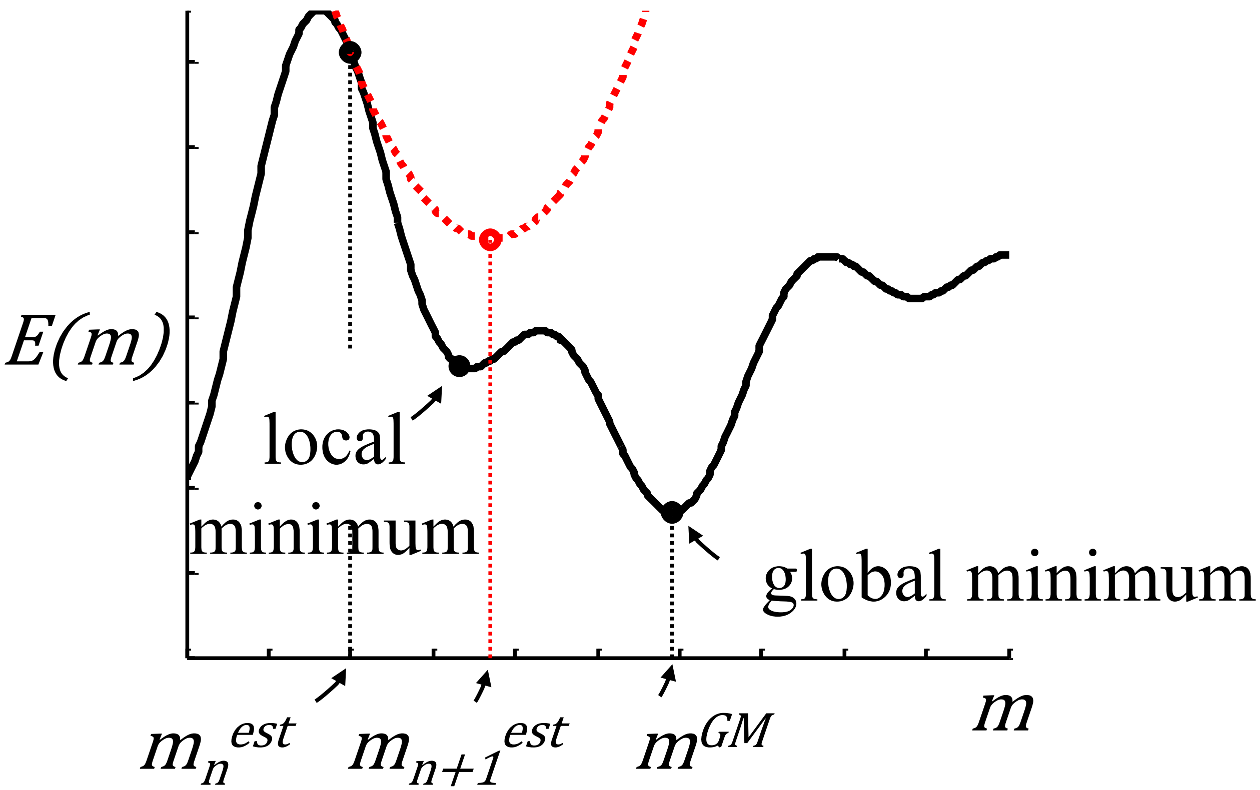
4) solve $m_1 = m_0 + \Delta m$

   let this be the new model, go to step (1) and iterate

   stop when residual $\sum \left(d_{pred} - d_{obs}\right)^2$

   is small enough.


E(m)
m
m_n^{est}
m^{GM}
m_{n+1}^{est}


E(m)
local
minimum
global minimum
m
m_n^{est}
m_{n+1}^{est}
m^{GM}

Chapter 4

Generalized Inverse

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

$$\mathbf{m}^{est} = \mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-1}\mathbf{d}$$

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G} + \varepsilon^2\mathbf{I}]^{-1}\,\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_e\mathbf{G} + \varepsilon^2\mathbf{W}_m]^{-1}\,[\mathbf{G}^{\mathrm{T}}\mathbf{W}_e\,\mathbf{d} + \varepsilon^2\mathbf{W}_m\langle\mathbf{m}\rangle\,]$$

all are of the form

$$\mathbf{m}^{\mathrm{est}} = \mathbf{M}\mathbf{d} + \mathbf{v}$$

$$\mathbf{m}^{\text{est}} = \mathbf{M}\mathbf{d} + \mathbf{v}$$

let's focus on
this matrix

$$\mathbf{m}^{est} = \mathbf{G}^{-g}\mathbf{d} + \mathbf{v}$$

rename it the "generalized inverse" and use the symbol $\mathbf{G}^{-g}$

Generalized Inverse $\mathbf{G}^{-g}$

operates on the data to give an estimate of the model parameters

Generalized Inverse $\mathbf{G}^{-g}$

$$\mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d}^{\text{obs}}$$

looks like a matrix inverse
except

$M \times N$, not necessarily square
and
$\mathbf{G}\mathbf{G}^{-g}$ and $\mathbf{G}^{-g}\mathbf{G}$ don't have to $= \mathrm{I}$

Data Resolution Matrix, $\mathbf{N}$

$$\mathbf{d}^{\text{pre}} = \mathbf{N}\mathbf{d}^{\text{obs}}$$

Describes how well the predictions match the data

If N=I, observed data and predicted data match perfectly

$$\mathbf{d}^{\mathbf{pre}} = \mathbf{d}^{\mathbf{obs}}$$

$$d_i^{pre} = d_i^{obs}$$

Data Resolution Matrix, $\mathbf{N}$

$$\mathbf{d}^{\text{pre}} = \mathbf{G}\mathbf{m}^{\text{est}} \quad \text{and} \quad \mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d}^{\text{obs}}$$

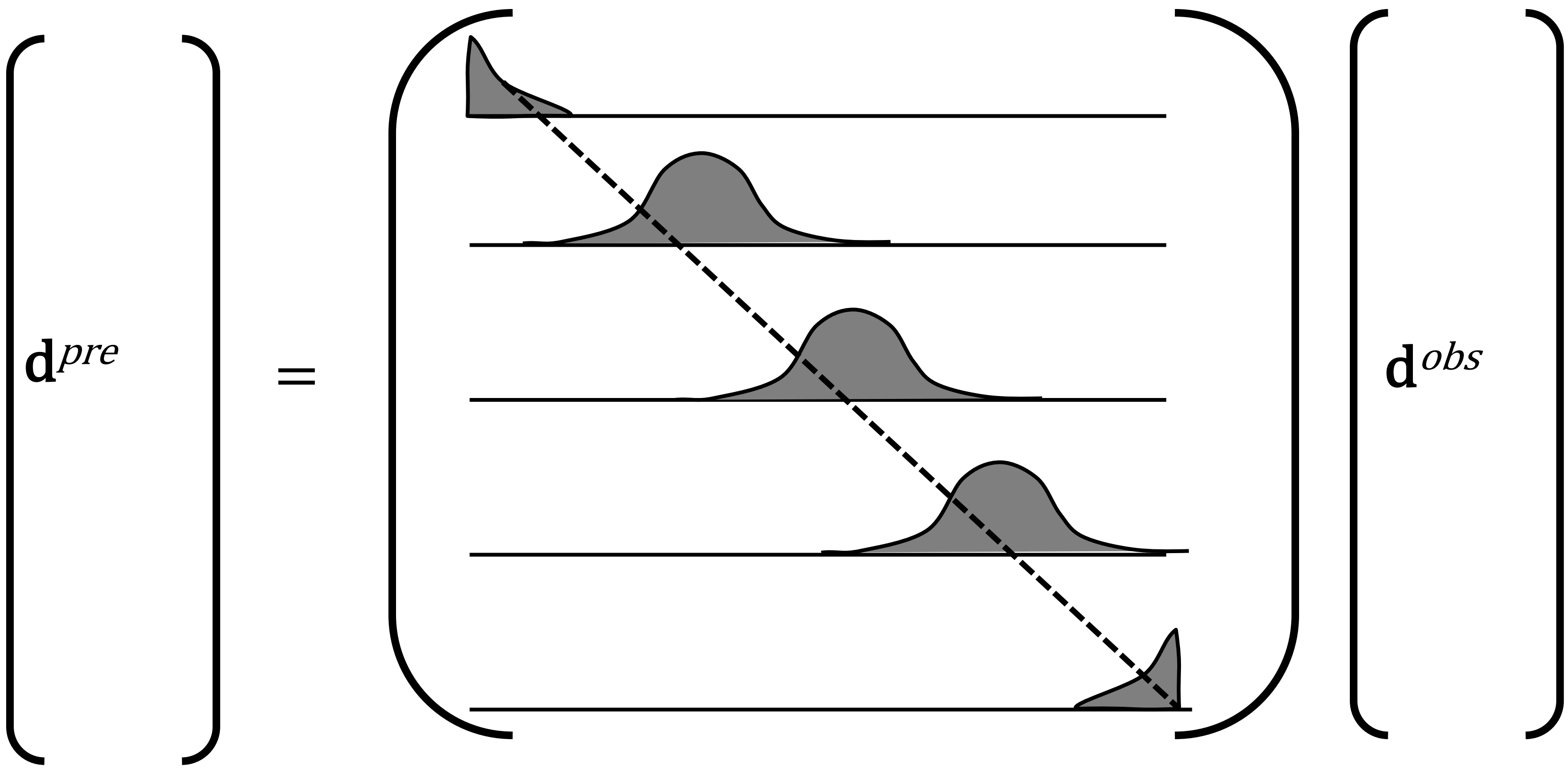
$$\mathbf{d}^{\text{pre}} = \mathbf{G}\mathbf{m}^{\text{est}} = \mathbf{G}\left[\mathbf{G}^{-g}\mathbf{d}^{\text{obs}}\right] = \left[\mathbf{G}\mathbf{G}^{-g}\right]\mathbf{d}^{\text{obs}} = \mathbf{N}\mathbf{d}^{\text{obs}}$$

$$\mathbf{d}^{\text{pre}} = \mathbf{N}\mathbf{d}^{\text{obs}} \quad \text{with} \quad \mathbf{N} = \mathbf{G}\mathbf{G}^{-g}$$

"data resolution matrix"

$$\mathbf{d}^{\mathbf{pre}} = \mathbf{N}\mathbf{d}^{\mathbf{obs}}$$

(A)



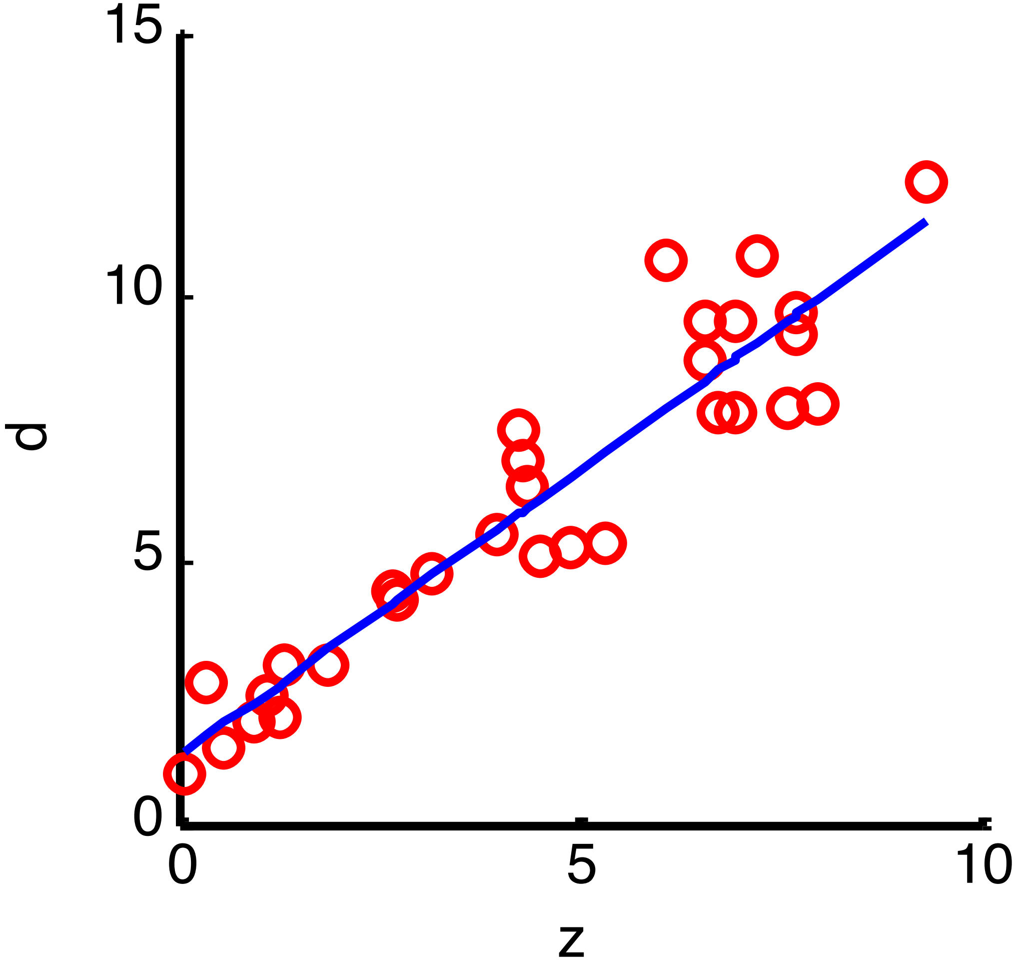
The closer **N** is to **I**, the closer d_i^{pre} is to d_i^{obs}

Consider the kth element of the predicted data $\hat{d}_k$

$$\hat{d}_k = \begin{bmatrix} & \\ \hline \text{kth row of } \mathbf{N} \\ \hline & \\ \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}$$

The rows of **N** are "windows" through which the observed data are viewed. If the kth row of **N** has a 1 in the kth column and zeroes elsewhere, the kth observation is perfectly resolved. We

straight line problem



$$\mathbf{N} = \mathbf{G}\mathbf{G}^{-g}$$

N does *not* depend on **d**. Just depends on the data kernel **G** and any a priori information applied to problem (weighting, smoothing, etc.)

Thus **N** can be computed and studied without actually performing the experiment, and can be a useful tool in experiment design.

$$\mathbf{N} = \mathbf{G}\mathbf{G}^{-g}$$

N does *not* depend on **d**. Just depends on the data kernel **G** and any a priori information applied to problem (weighting, smoothing, etc.)

Thus **N** can be computed and studied without actually performing the experiment, and can be a useful tool in experiment design.

$$\mathbf{m}^{est} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{d}$$

$$\mathbf{m}^{est} = \mathbf{G}^T[\mathbf{G}\mathbf{G}^T]^{-1}\mathbf{d}$$

$$\mathbf{m}^{est} = [\mathbf{G}^T\mathbf{G} + \varepsilon^2\mathbf{I}]^{-1}\mathbf{G}^T\mathbf{d}$$

Model Resolution Matrix, $\mathbf{R}$

$$\mathbf{m}^{\mathbf{est}} = \mathbf{R}\mathbf{m}^{\mathbf{true}}$$

Shows how well m_i^{est} compares to m_i^{true}

if

R=I

$$\mathbf{m}^{\text{est}} = \mathbf{m}^{\text{true}}$$

$$m_i^{est} = m_i^{true}$$

else if

R≠I

$$m_i^{est} =$$

$$... + R_{i,i\text{-}1} m_{i\text{-}1}^{true} + R_{i,i} m_i^{true} + R_{i,i+1} m_{i+1}^{true} + ...$$

Model Resolution Matrix, $\mathbf{R}$

$$\mathbf{d}^{\text{obs}} = \mathbf{G}\mathbf{m}^{\text{true}} \quad \text{and} \quad \mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d}^{\text{obs}}$$

$$\mathbf{m}^{\text{est}} = \mathbf{G}^{-g}\mathbf{d}^{\text{obs}} = \mathbf{G}^{-g}[\mathbf{G}\mathbf{m}^{\text{true}}] = [\mathbf{G}^{-g}\mathbf{G}]\mathbf{m}^{\text{true}} = \mathbf{R}\mathbf{m}^{\text{true}}$$

$$\mathbf{m}^{\text{est}} = \mathbf{R}\mathbf{m}^{\text{true}} \quad \text{with} \quad \mathbf{R} = \mathbf{G}^{-g}\mathbf{G}$$

"model resolution matrix"

$$m_k^{\text{est}} = \begin{bmatrix} \\ \hline k\text{th row of } \mathbf{R} \\ \hline \\ \\ \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix}^{\text{true}}$$

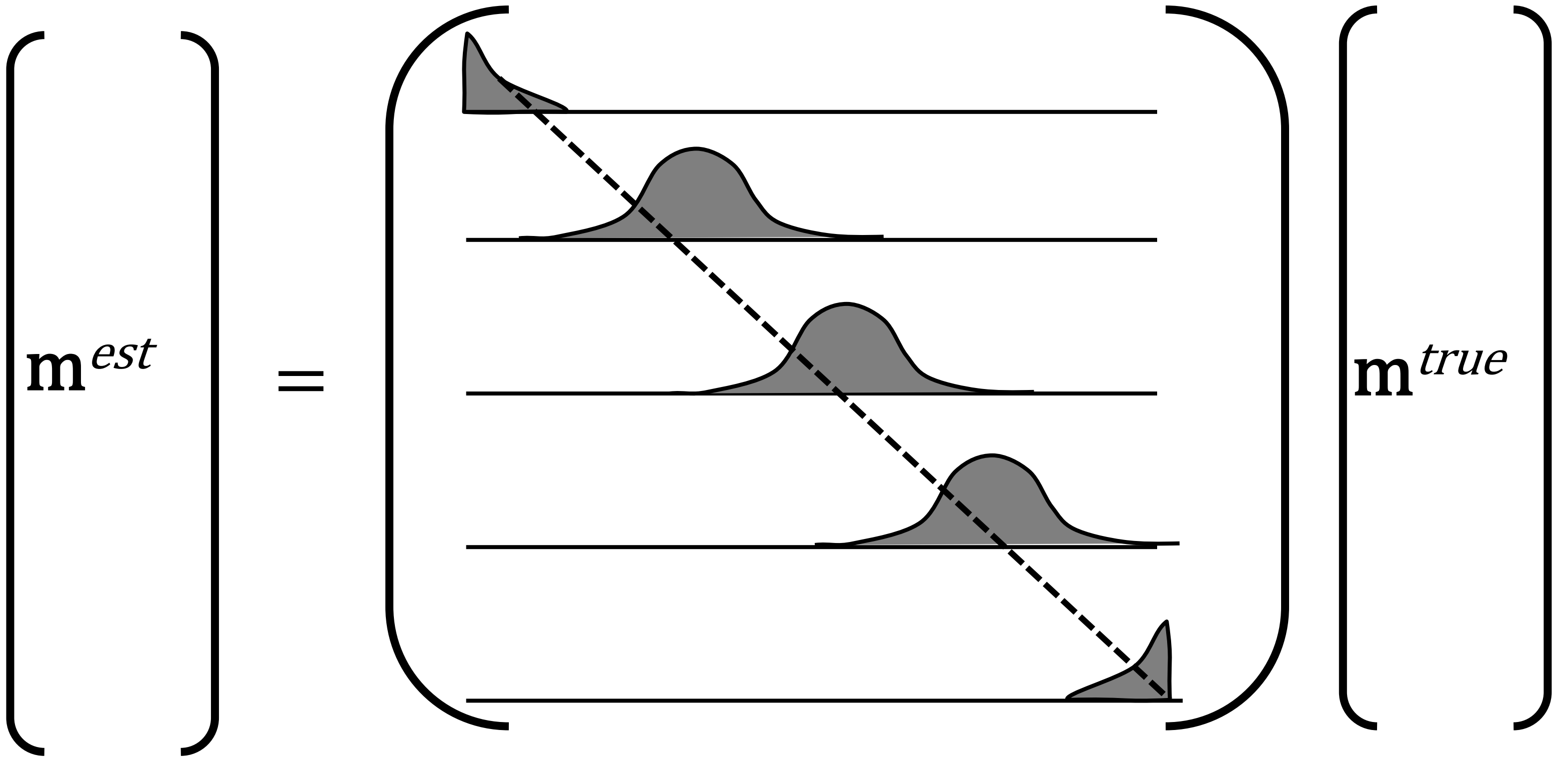
$$m_k^{\text{est}} = \begin{bmatrix} \rule{0pt}{1em} \\ \hline k\text{th row of } \mathbf{R} \\ \hline \\ \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix}^{\text{true}}$$

consider

$$\mathbf{R}_k = [0, 0, \ldots, 0, \underset{\substack{\Uparrow \\ k\text{th column}}}{1}, 0, \ldots, 0, 0]$$

versus

$$\mathbf{R}_k = [0, \ldots, 0, 0.1, 0.3, \underset{\substack{\Uparrow \\ k\text{th column}}}{0.8}, 0.4, 0.2, \ldots, 0]$$



The closer $\mathbf{R}$ is to $\mathbf{I}$, the more m_i^{est} reflects only m_i^{true}

$$\mathbf{R} = \mathbf{G}^{-g}\mathbf{G}$$

Similar to the data resolution matrix – can be used in experiment design – can calculate it without actual values of data

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G}]^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

$$\mathbf{m}^{est} = \mathbf{G}^{\mathrm{T}}[\mathbf{G}\mathbf{G}^{\mathrm{T}}]^{-1}\mathbf{d}$$

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{G} + \varepsilon^{2}\mathbf{I}]^{-1}\,\mathbf{G}^{\mathrm{T}}\mathbf{d}$$

$$\mathbf{m}^{\mathrm{est}} = [\mathbf{G}^{\mathrm{T}}\mathbf{W}_{e}\mathbf{G} + \varepsilon^{2}\mathbf{W}_{m}]^{-1}\,[\mathbf{G}^{\mathrm{T}}\mathbf{W}_{e}\,\mathbf{d} + \varepsilon^{2}\mathbf{W}_{m}\langle\mathbf{m}\rangle\,]$$

Another measure of goodness of fit
Spread Functions

$$\text{spread}(\mathbf{N}) = \|\mathbf{N} - \mathbf{I}\|_2^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\left[N_{ij} - \delta_{ij}\right]^2$$

$$\text{spread}(\mathbf{R}) = \|\mathbf{R} - \mathbf{I}\|_2^2 = \sum_{i=1}^{M}\sum_{j=1}^{M}\left[R_{ij} - \delta_{ij}\right]^2$$

If R = I, Spread (R) = 0

sometimes called "Dirichlet" Spread Functions

over-determined case

note that for
simple least squares

$$\mathbf{G}^{-g} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T$$

model resolution

$$\mathbf{R}=\mathbf{G}^{-g}\mathbf{G} = [\mathbf{G}^T\mathbf{G}]^{-1}\mathbf{G}^T\mathbf{G}=\mathbf{I}$$

always the identity matrix

try to minimize the spread of the data resolution matrix, **N**

find $\mathbf{G}^{-g}$ that minimizes spread($\mathbf{N}$)

the simple least squares solution
minimizes the spread of data resolution
and
has zero spread of the model resolution

the minimum length solution
minimizes the spread of model resolution
and
has zero spread of the data resolution