Lecture 5

A Priori Information and Weighted Least Squared

Syllabus

Lecture 01	Describing Inverse Problems
Lecture 02	Probability and Measurement Error, Part 1
Lecture 03	Probability and Measurement Error, Part 2
Lecture 04	The L_2 Norm and Simple Least Squares
Lecture 05	**A Priori Information and Weighted Least Squared**
Lecture 06	Resolution and Generalized Inverses
Lecture 07	Backus-Gilbert Inverse and the Trade Off of Resolution and Variance
Lecture 08	The Principle of Maximum Likelihood
Lecture 09	Inexact Theories
Lecture 10	Nonuniqueness and Localized Averages
Lecture 11	Vector Spaces and Singular Value Decomposition
Lecture 12	Equality and Inequality Constraints
Lecture 13	L_1 , L_∞ Norm Problems and Linear Programming
Lecture 14	Nonlinear Problems: Grid and Monte Carlo Searches
Lecture 15	Nonlinear Problems: Newton's Method
Lecture 16	Nonlinear Problems: Simulated Annealing and Bootstrap Confidence Intervals
Lecture 17	Factor Analysis
Lecture 18	Varimax Factors, Empircal Orthogonal Functions
Lecture 19	Backus-Gilbert Theory for Continuous Problems; Radon's Problem
Lecture 20	Linear Operators and Their Adjoints
Lecture 21	Fréchet Derivatives
Lecture 22	Exemplary Inverse Problems, incl. Filter Design
Lecture 23	Exemplary Inverse Problems, incl. Earthquake Location
Lecture 24	Exemplary Inverse Problems, incl. Vibrational Problems

Purpose of the Lecture

Classify Inverse Problems as Overdetermined, Underdetermined and Mixed-Determined

Further Develop the Notion of A Priori Information

Apply A Priori Information to Solving Inverse Problems

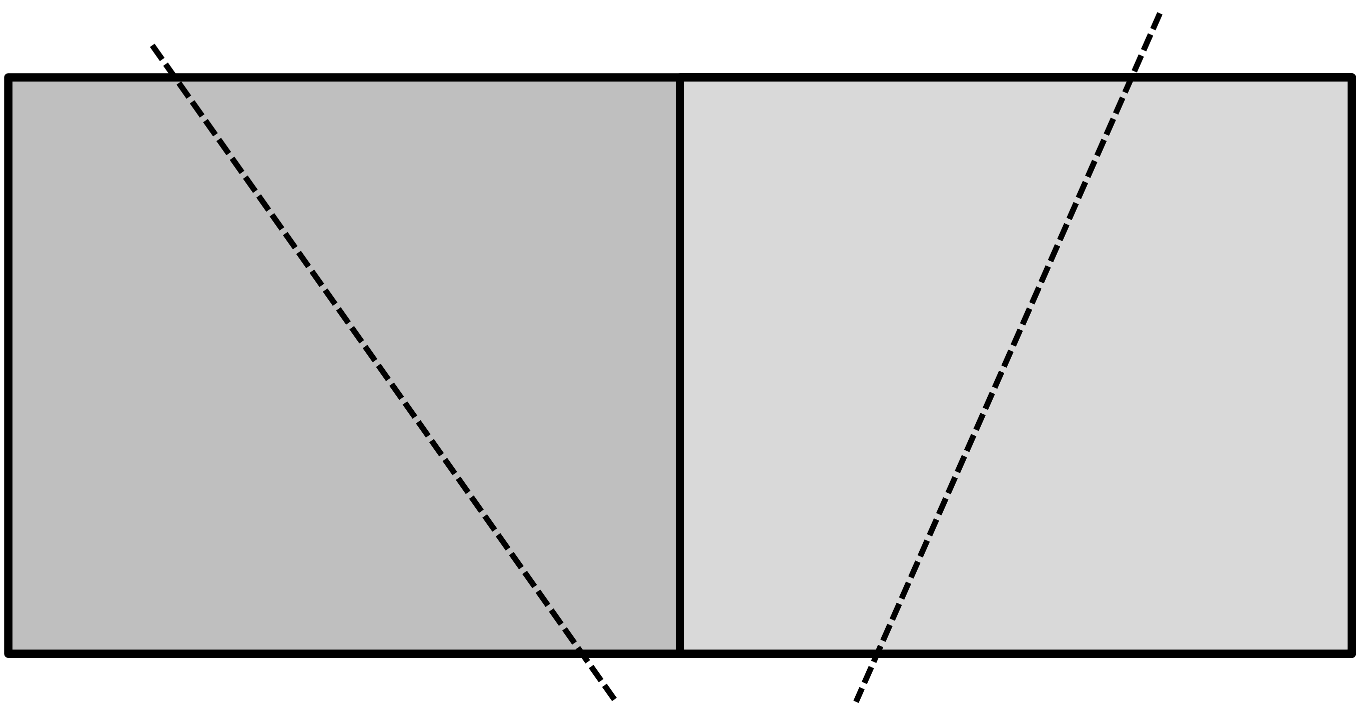
Part 1

Classification of Inverse Problems on the Basis of Information Content

"Even Determined"

Exactly enough data is available to determine the model parameters

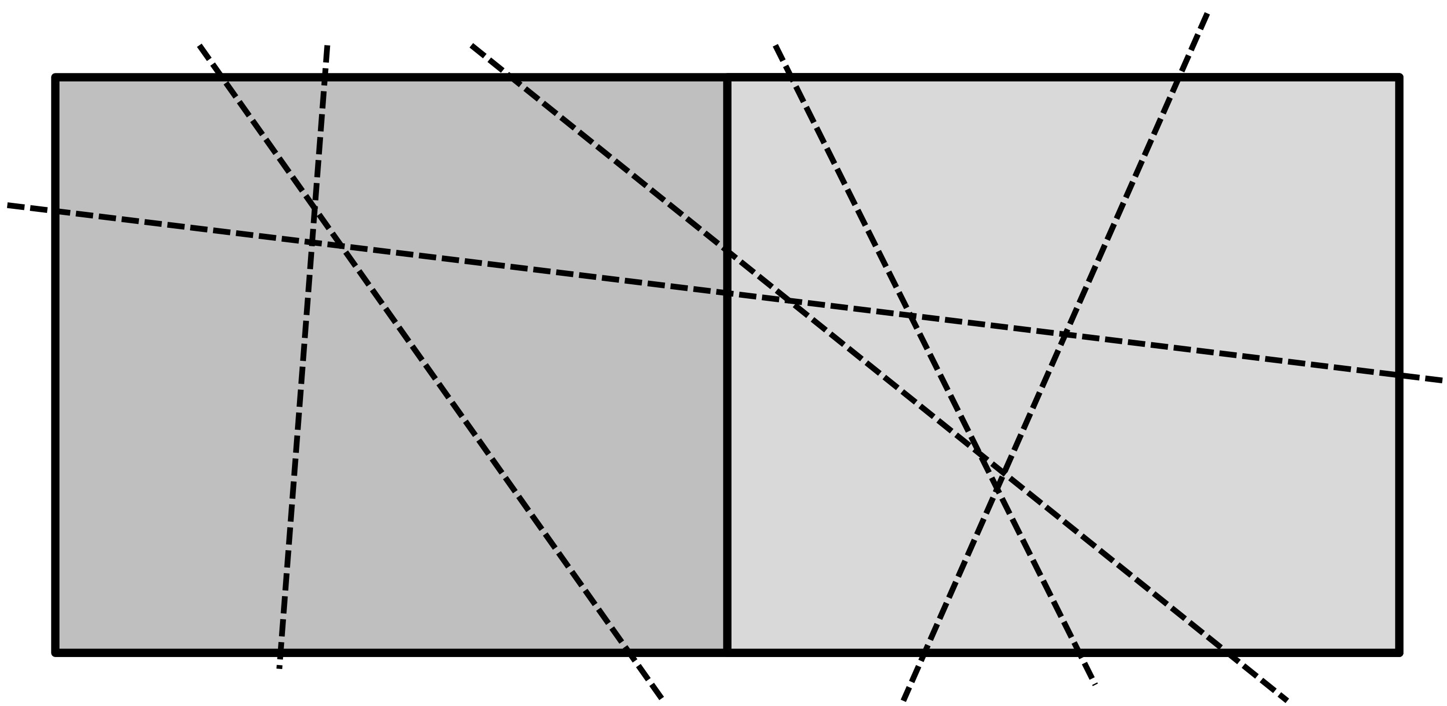
E=0 and solution unique
(a rare case)



"Over Determined"

More than enough data is available to determine the model parameters

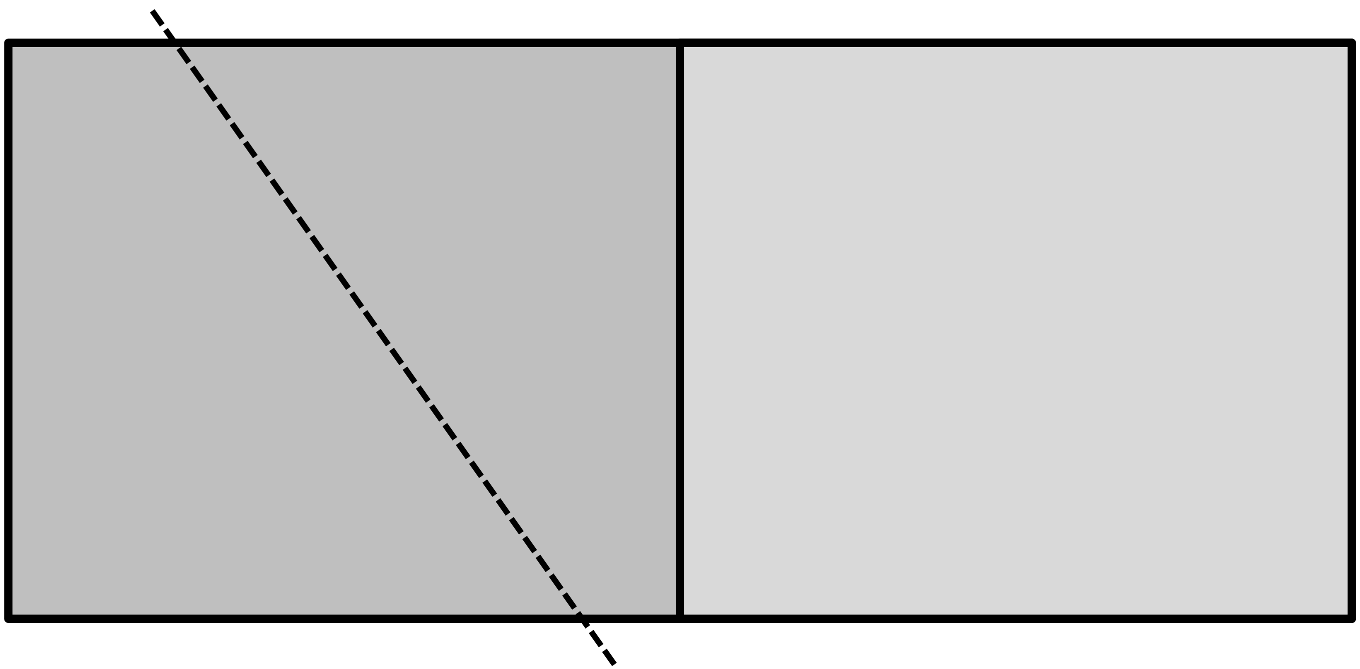
$E>0$ and solution unique



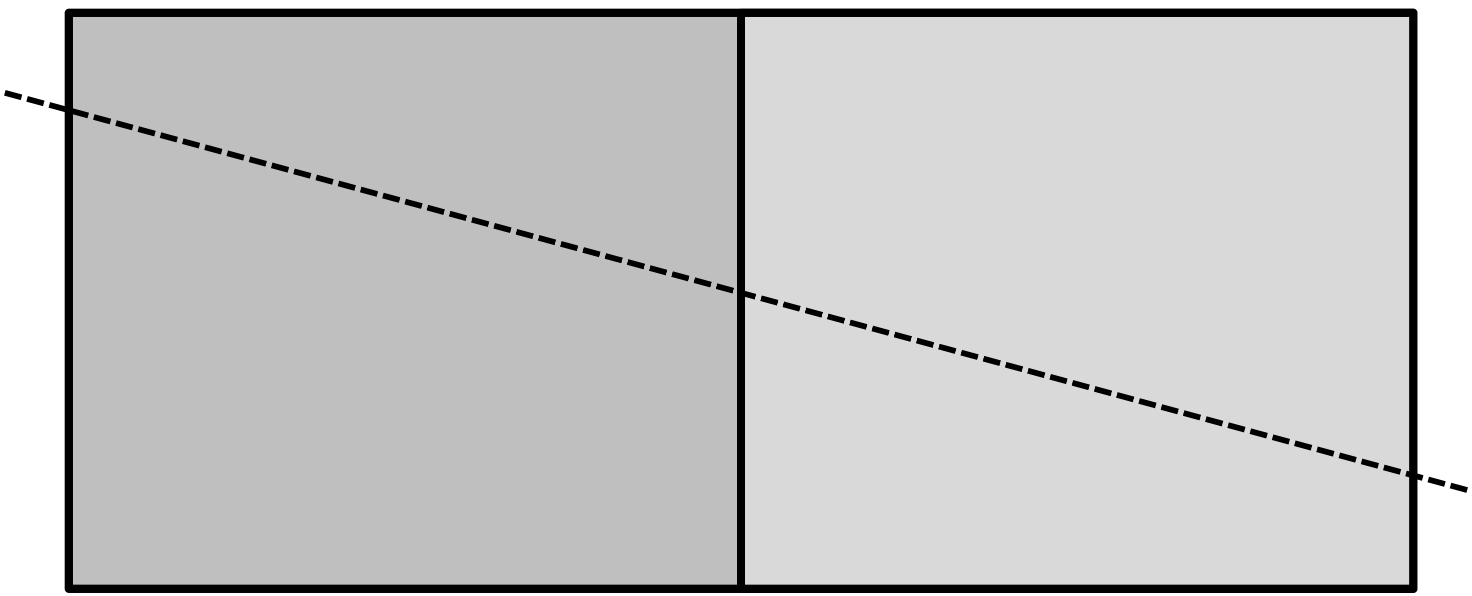
"Under Determined"

Insufficient data is available to determine all the model parameters

$E=0$ and solution non-unique



This configuration is also underdetermined since only the mean is determined

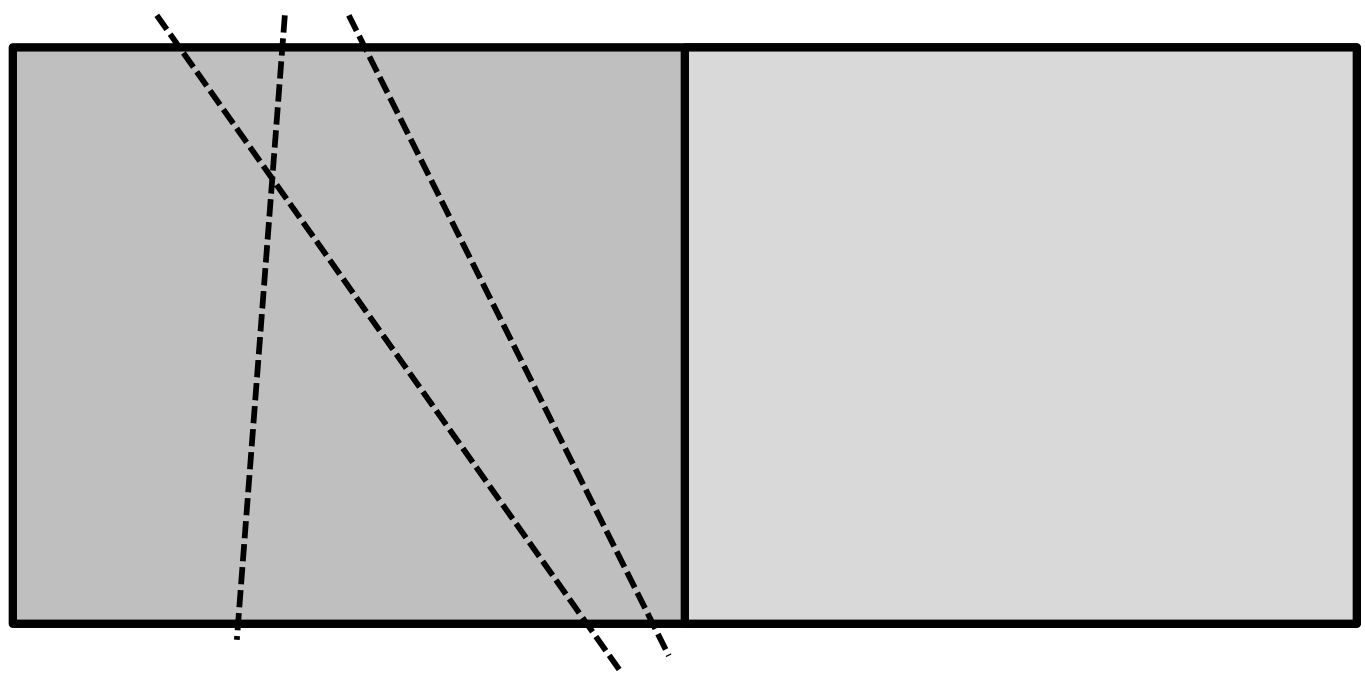


“Mixed Determined”

More than enough data is available to constrain some the model parameters

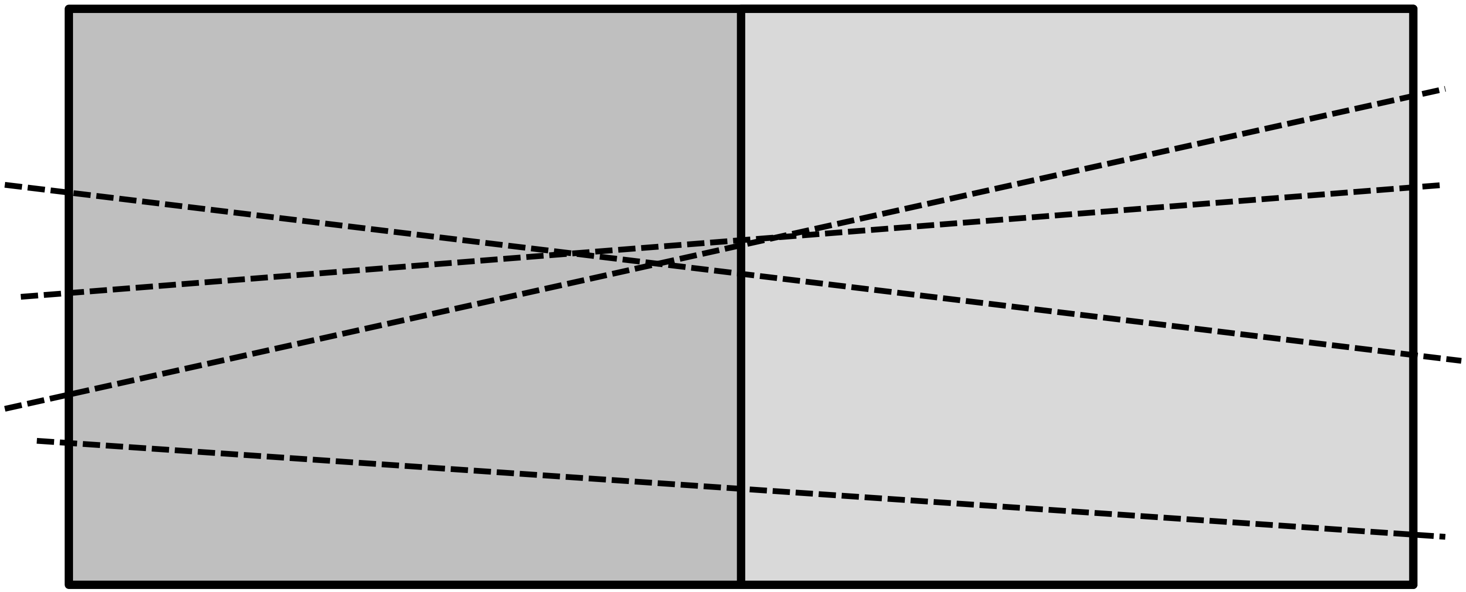
Insufficient data is available to constrain other model parameters

$E>0$ and solution non-unique



this configuration is also mixed-determined

the average of the two blocks is over-determined
and
the difference between the two blocks is under-determined



mixed-determined

some linear combinations of model parameters are not determined by the data

(very common)

what to do?

add a priori information that supplement observations

Part 2

a priori information

a priori information

preconceptions about the behavior of the model parameters

example of a priori information

model parameters are:

small
near a given value
have a known average value
smoothly varying with position
solve a known differential equation
positive
etc.

dangerous?

perhaps …

but we have a lot of experience about the world in general, so why not put that experience to work

one approach to solving
a mixed-determined problem

"of all the solutions that minimize $E=||\mathbf{e}||^2$ choose the one with minimum $L=||\mathbf{m}||^2$"

“of all the solutions that minimize E choose the one with minimum L”

turns out to be hard to do, since you have to know how to divide up the model parameters into two groups

one over-determined
one under-determined

next best thing

"of all the solutions that minimize E choose the one with minimum L"

"choose the solutions that minimizes $E+\varepsilon^2 L$"

minimize

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

when ε^2 is chosen to be small,
the E will be approximately minimized and the solution will be small

minimize

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

damped least-squares solution

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

minimize

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

damped least-squares solution

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

Very similar to least-squares

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

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

Part 3

Using Prior Information to Solve Inverse Problems

$\mathbf{m}$ is small

minimize

$$L = \mathbf{m}^{\mathrm{T}}\mathbf{m}$$

$\mathbf{m}$ is close to $<\mathbf{m}>$

minimize

$$L = (\mathbf{m} - \langle \mathbf{m} \rangle)^{\mathrm{T}} (\mathbf{m} - \langle \mathbf{m} \rangle)$$

m varies slowly with position
(**m** is flat)

characterize steepness with
first-difference

$$\mathbf{l} = \begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix} = \mathbf{Dm}$$

approximation
for dm/dx

m varies smoothly with position
(**m** is smooth)

characterize roughness with second-difference

$$\mathbf{l} = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_M \end{bmatrix} = \mathbf{Dm}$$

approximation for d^2m/dx^2

$\mathbf{m}$ varies slowly/smoothly with position

minimize

$$L = \mathbf{l}^{\mathrm{T}}\mathbf{l} = [\mathbf{Dm}]^{\mathrm{T}}[\mathbf{Dm}] = \mathbf{m}^{\mathrm{T}}\mathbf{D}^{\mathrm{T}}\mathbf{Dm} = \mathbf{m}^{\mathrm{T}}\mathbf{W}_{m}\mathbf{m}$$

with $\mathbf{W}_{\mathrm{m}} = \mathbf{D}^{\mathrm{T}}\mathbf{D}$

Suppose that some data are more accurately determined than others

minimize

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

example when d_3 is more accurately measured than the other data

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

weighted damped least squares

minimize $E+\varepsilon^2 L$

with

$$L = [\mathbf{m} - \langle \mathbf{m} \rangle]^{\mathrm{T}} \mathbf{W}_m [\mathbf{m} - \langle \mathbf{m} \rangle]$$

and

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

weighted damped least squares

solution

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

weighted damped least squares

solution

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

a bit complicated, but …

equivalent to solving

$$\mathbf{F}\,\mathbf{m}^{est} = \mathbf{f} \quad \text{with} \quad \mathbf{F} = \begin{bmatrix} \mathbf{W}_e{}^{1/2}\mathbf{G} \\ \epsilon\mathbf{D} \end{bmatrix} \quad \text{and} \quad \mathbf{f} = \begin{bmatrix} \mathbf{W}_e{}^{1/2}\mathbf{d} \\ \epsilon\mathbf{D}\langle\mathbf{m}\rangle \end{bmatrix}$$

by simple least squares

$$\mathbf{F}\,\mathbf{m}^{est} = \mathbf{f}$$

$$\begin{bmatrix} \mathbf{W}_e{}^{1/2}\mathbf{G} \\ \varepsilon\mathbf{D} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{W}_e{}^{1/2}\mathbf{d} \\ \varepsilon\mathbf{D}\langle\mathbf{m}\rangle \end{bmatrix}$$

$$\mathbf{F}\,\mathbf{m}^{est} = \mathbf{f}$$

$$\begin{bmatrix} \mathbf{W}_e^{1/2}\mathbf{G} \\ \varepsilon\mathbf{D} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{W}_e^{1/2}\mathbf{d} \\ \varepsilon\mathbf{D}\langle\mathbf{m}\rangle \end{bmatrix}$$

top rows
data equation **Gm=d**
weighted by $\mathbf{W}_e^{1/2}$

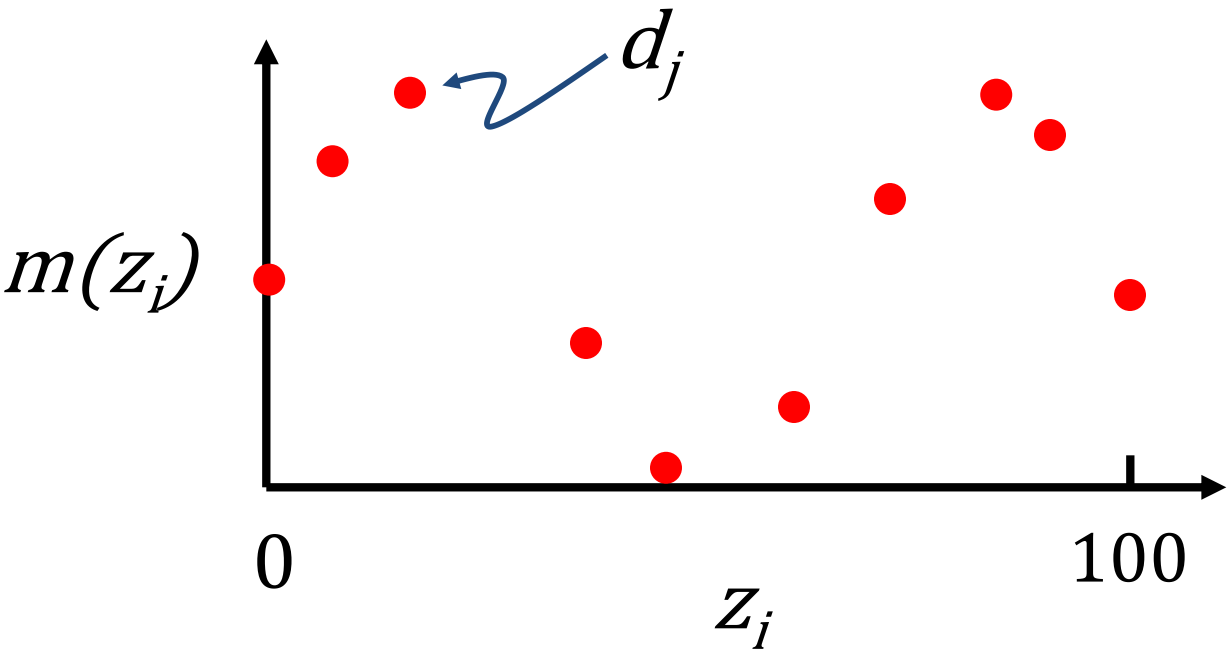
$$\mathbf{F}\,\mathbf{m}^{est} = \mathbf{f}$$

$$\begin{bmatrix} \mathbf{W}_e{}^{1/2}\mathbf{G} \\ \varepsilon\mathbf{D} \end{bmatrix} \mathbf{m} = \begin{bmatrix} \mathbf{W}_e{}^{1/2}\mathbf{d} \\ \varepsilon\mathbf{D}\langle\mathbf{m}\rangle \end{bmatrix}$$

top rows
a priori equation
m=<m>
weighted by ε **D**

you can even use this equation to implement constraints just by making ε very large

example
fill in missing data
of
discretized version of *m(z)*



set up

M=100 model parameters

N<M data

data, when available, gives values of model parameter

$$d_i = m_j$$

$$\mathbf{d} = \mathbf{Gm}$$

associates datum with corresponding model parameter each row has M-1 zeros and a single one

a priori information of smoothness in interior x's flatness at ends

$\mathbf{F\,m = f}$

$$\mathbf{F} = \begin{bmatrix} & 1 & & & & & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & & 1 & & & & \\ - & - & - & - & - & - & - & - & - & - & - \\ \varepsilon & -2\varepsilon & \varepsilon & & & & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & & & & \varepsilon & -2\varepsilon & \varepsilon \\ - & - & - & - & - & - & - & - & - & - & - \\ -\varepsilon & \varepsilon & & & & & & & & & \\ & & & & & & & & & -\varepsilon & \varepsilon \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \\ - \\ 0 \\ \vdots \\ 0 \\ - \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{F}\,\mathbf{m} = \mathbf{f}$$

associates datum with corresponding model parameter

$$\mathbf{F} = \begin{bmatrix}
 & 1 & & & & & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & & & 1 & & & & \\
- & - & - & - & - & - & - & - & - & - & - \\
\varepsilon & -2\varepsilon & \varepsilon & & & & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & & & & & \varepsilon & -2\varepsilon & \varepsilon \\
- & - & - & - & - & - & - & - & - & - & - \\
-\varepsilon & \varepsilon & & & & & & & & & \\
 & & & & & & & & & -\varepsilon & \varepsilon
\end{bmatrix}
\qquad
\mathbf{f} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \\ - \\ 0 \\ \vdots \\ 0 \\ - \\ 0 \\ 0 \end{bmatrix}$$

$\mathbf{F\,m = f}$

roughness in interior

$$\mathbf{F} = \begin{bmatrix} & 1 & & & & & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & & 1 & & & & \\ - & - & - & - & - & - & - & - & - & - & - \\ \varepsilon & -2\varepsilon & \varepsilon & & & & & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & & & & & & \varepsilon & -2\varepsilon & \varepsilon \\ - & - & - & - & - & - & - & - & - & - & - \\ -\varepsilon & \varepsilon & & & & & & & & & \\ & & & & & & & & & -\varepsilon & \varepsilon \end{bmatrix} \qquad \mathbf{f} = \begin{bmatrix} d_1 \\ \vdots \\ d_N \\ - \\ 0 \\ \vdots \\ 0 \\ - \\ 0 \\ 0 \end{bmatrix}$$

$\mathbf{F}\,\mathbf{m} = \mathbf{f}$

steepness at left

$$
\mathbf{F} = \begin{bmatrix}
 & 1 & & & & & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & & & 1 & & & & \\
- & - & - & - & - & - & - & - & - & - & - \\
\varepsilon & -2\varepsilon & \varepsilon & & & & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & & & & & \varepsilon & -2\varepsilon & \varepsilon \\
- & - & - & - & - & - & - & - & - & - & - \\
-\varepsilon & \varepsilon & & & & & & & & & \\
 & & & & & & & & & -\varepsilon & \varepsilon
\end{bmatrix}
\qquad
\mathbf{f} = \begin{bmatrix}
d_1 \\ \vdots \\ d_N \\ - \\ 0 \\ \vdots \\ 0 \\ - \\ 0 \\ 0
\end{bmatrix}
$$

$\mathbf{F\,m = f}$

steepness at right

$$
\mathbf{F} = \begin{bmatrix}
 & 1 & & & & & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & & & 1 & & & & \\
- & - & - & - & - & - & - & - & - & - & - \\
\varepsilon & -2\varepsilon & \varepsilon & & & & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & & & & & \varepsilon & -2\varepsilon & \varepsilon \\
- & - & - & - & - & - & - & - & - & - & - \\
-\varepsilon & \varepsilon & & & & & & & & & \\
 & & & & & & & & & -\varepsilon & \varepsilon
\end{bmatrix}
\qquad
\mathbf{f} = \begin{bmatrix}
d_1 \\ \vdots \\ d_N \\ - \\ 0 \\ \vdots \\ 0 \\ - \\ 0 \\ 0
\end{bmatrix}
$$

computational efficiency

1. Use sparse matrices

2. Use solver that does **not** require forming $\mathbf{F}^T\mathbf{F}$

1. Sparse matrices

```
F = spalloc(N, M, 3*N);
```

2A. Use Biconjugate Gradient Algorithm (an iterative method to solve $\mathbf{F}^T\mathbf{F}\mathbf{m} = \mathbf{F}^T\mathbf{f}$)

```
clear F;
global F;
- - -
tol = 1e-6;
maxit = 3*M;
mest = bicg( @weightedleastsquaresfcn, F'*f, tol, maxit );
```

stop iterations
when solution is good enough

2A. Use Biconjugate Gradient Algorithm (an iterative method to solve $\mathbf{F}^T\mathbf{F}\mathbf{m} = \mathbf{F}^T\mathbf{f}$)

```
clear F;
global F;
- - -
tol = 1e-6;
maxit = 3*M;
mest = bicg( @weightedleastsquaresfcn, F'*f, tol, maxit );
```

stop iterations after maximum iteration is reached, regardless of whether it is good enough

2A. Use Biconjugate Gradient Algorithm (an iterative method to solve $\mathbf{F}^T\mathbf{F}\mathbf{m} = \mathbf{F}^T\mathbf{f}$)

```
clear F;
global F;
- - -
tol = 1e-6;
maxit = 3*M;
mest = bicg( @weightedleastsquaresfcn, F'*f, tol, maxit );
```

$\mathbf{F}^T\mathbf{f}$

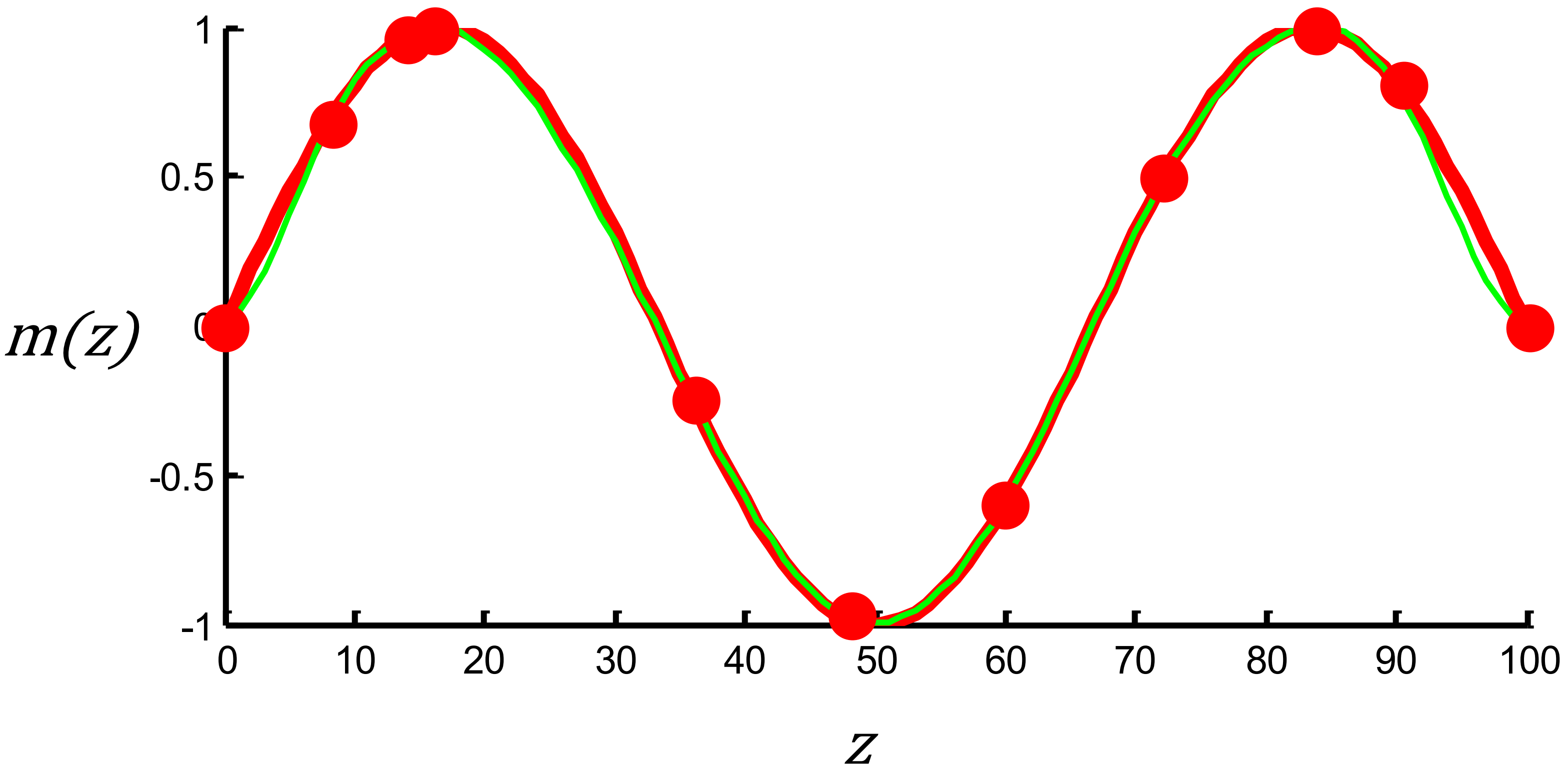
2A. Use Biconjugate Gradient Algorithm (an iterative method to solve $\mathbf{F}^T\mathbf{F}\mathbf{m} = \mathbf{F}^T\mathbf{f}$)

```
clear F;
global F;
- - -
tol = 1e-6;
maxit = 3*M;
mest = bicg( @weightedleastsquaresfcn, F'*f, tol, maxit );
```

function that calculates $\mathbf{F}^T\mathbf{F}$ times a vector $\mathbf{v}$

2B. Function to multiply a vector by $\mathbf{F}^T\mathbf{F}$ do as $\mathbf{y} = \mathbf{F}^T(\mathbf{Fv})$ so $\mathbf{F}^T\mathbf{F}$ never calculated

```
function y = weightedleastsquaresfcn(v,transp_flag)
global F;
temp = F*v;
y = F'*temp;
return
```



m(z)
1
0.5
0
-0.5
-1
0
10
20
30
40
50
60
70
80
90
100
z