1. Clean data
   - Sugar tenders at the G5
   - Remove data points
   - ask if statements

2. data vector is a linear combo of the two columns and what we need to put in those columns

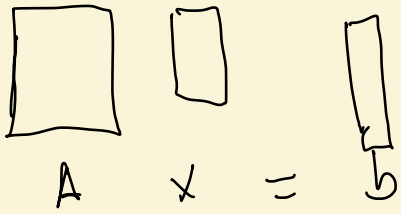
$$A = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix} \quad m = \begin{bmatrix} M \\ C \end{bmatrix} \quad d = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

3. Creating matrix

$$\{mx_1 + c\}$$

$$\{mx_2 + c\}$$

- Time = np.array
- ones = for hundrit (np.ones)
- A = (np.vstack(x, 1)
- Co_2 = np.array(
- d = Co2.reshape(-1, 1)



coeff

Responds predictively
So single input parameter
double output model.

Fault double slip Model ... 2x slip 2x displacement

$$x = (A^T A)^{-1} A^T b$$

normal equations - fit data
in a least square sense
norm will be minimized
w/ this solution.

Non linearity of the model ...

- Best model is not trivial
-

mitigate by setting up

- misfit or penalty statistic
- Choose lowest misfit form models

Parameter Space ...

lowest point on the parameter surface is the lowest input misfit of the

Problem of Non linear problems

- Want the Global misfit

More detail...

Time

data measurement

$$A \begin{bmatrix} \quad \end{bmatrix} x \begin{bmatrix} \quad \end{bmatrix} = b \begin{bmatrix} \quad \end{bmatrix}$$

So... $\text{data}(t) \approx m_1 t + m_0$

m_1 = slope

m_0 = intercept

① $\text{time} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$ or np.array(time)

② $\text{ones} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ or np.ones(time.shape)

The constant. ones are like the scale tells model to use same intercept every time ($y_1 = mx_1 + b \cdot 1$)

③ stack them $\begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix}$ or np.vstack((time, ones))

this create the A vector for matrix 2x2

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix}$$

④ $\text{data} = \begin{bmatrix} 400 \\ 401 \\ 402 \\ 403 \\ 404 \\ 405 \end{bmatrix}$ or data.reshape(-1, 1)

Creates a vertical vector

The linear model

$$d \approx A m$$

or

$$\begin{bmatrix} 400 \\ 401 \\ 402 \\ 403 \\ 404 \end{bmatrix} \approx \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} \text{Yint} \\ \text{slope} \end{bmatrix} = \begin{bmatrix} \text{slope} \cdot 0 + \text{yint} \\ \text{slope} \cdot 1 + \text{yint} \\ \vdots \\ \vdots \end{bmatrix}$$

This allows us to fit the data with a linear formula to the data points given

(5) Compute the $A^TA = \begin{array}{c} a \\ c \end{array}\begin{bmatrix} \sum t_i^2 & \sum t_i \\ \sum t_i & N \end{bmatrix}\begin{array}{c} b \\ d \end{array}$

or $\begin{bmatrix} 30 & 7 \\ 7 & 5 \end{bmatrix}$

a = Sum of t^2

b = Sum of t

c = Sum of t

d = number of data points

(6) Now $A^Td = \begin{array}{c} a \\ b \end{array}\begin{bmatrix} \sum t_i d_i \\ \sum d_i \end{bmatrix}$

$a = t_1 \cdot d_1$ or $400 \cdot 0 = 0$

$t_2 \cdot d_2 = 401$

$\cdots = 804$

$\cdots = 1209$

$\cdots = 1616$

$+$

3020

or

$$A^Td = \begin{bmatrix} 3020 \\ 2010 \end{bmatrix}$$

7. Solve the system of equations

$$(A^T A)m = A^T d \quad \text{or} \quad m = \text{np.matmul(np.linalg.inv(ATA), ATd)}$$

So...
$$\begin{bmatrix} 36 & 7 \\ 7 & 5 \end{bmatrix} \cdot \begin{bmatrix} \text{slope} \\ \text{yint} \end{bmatrix} = \begin{bmatrix} 3020 \\ 2010 \end{bmatrix}$$

Solving 2x2 linear system w/ normal equations

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

$A \cdot m + B \cdot b = d_1$ (eq1)

$C \cdot m + D \cdot b = d_2$ (eq2)

D and B are the coeff

Find the LCM for D & B

What $\cdot$ D = LCM

What $\cdot$ B = LCM

$$\text{What}(A \cdot m + B \cdot b) = \text{What}(d_1)$$

$$\text{What}(C \cdot m + D \cdot b) = \text{What}(d_2)$$

now..

$$(A+C) \cdot m + (D+B) \cdot b = d_1 + d_2$$

where $D+b=0$

solve for m then sub m back in for b

$m = np.linalg.inv(A^TA)$

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{if } ad - bc = \emptyset$$

inverse is ...

$$M^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

So $A^{T^{-1}}A = \begin{bmatrix} 5 & -7 \\ -7 & 30 \end{bmatrix}$

$$M^{-1} = \frac{1}{(30 \cdot 5) - (7 \cdot 7)} \begin{bmatrix} 5 & -7 \\ -7 & 30 \end{bmatrix} = \frac{1}{101} \begin{array}{c} a \\ c \end{array} \begin{bmatrix} 5 & -7 \\ -7 & 30 \end{bmatrix} \begin{array}{c} b \\ d \end{array} \begin{bmatrix} 3020 \\ 2010 \end{bmatrix} \begin{array}{l} \square \\ \star \end{array}$$

So ...

$$5 \cdot 3020 - 7 \cdot 2010 = 1030$$

or $a \cdot \square + b \cdot \star =$

So now

$$-7 \cdot 3020 + 30 \cdot 2010 = 39160$$

or

$$\begin{matrix} 1030 / 101 \\ 39160 / 101 \end{matrix} = \begin{bmatrix} 10 \\ 387 \end{bmatrix} \begin{matrix} = m_1 \\ m_0 \end{matrix}$$

Now we found $\begin{bmatrix} m \\ b \end{bmatrix}$ or $\begin{bmatrix} \text{slope} \\ \text{yint} \end{bmatrix}$ or $\begin{bmatrix} m_1 \\ m_0 \end{bmatrix}$

now we can bueing the line, m, with A. (time)

$$\hat{data} = slope \cdot t_0 + yint$$

or

$$387 = 10 \cdot 0 + 387$$

where $t = \sum_{t_0}^{n} t_i$

Strategies for Non Known Inv.

Monte Carlo - random Sampling

Optimization methods - "gradient decent"

Markov Chain Monte Carlo - "Random walk throughout"

Grid Search -

1. Fixed param values
2. Used to make grid Search

Pro	Con
	↑ Param ↑ computation

1D Strike Slip Model - Measuring the Beardon of liquid elastic later

V = Vel

S = Slip

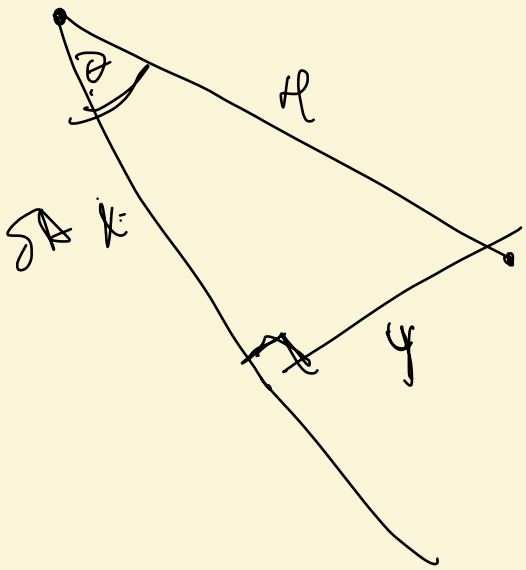
D = locking Depth L

V_{Bulk} = Shift in velocity

Spectral Deriv, Smooth Scale

↓ lock depth = Closer to step function

Changes farfield and gradient in the middle



Scree geo map

Cavines → SCRL

if

89

124

P050	-3.1 km
C616	8.65
TWR2	1.48
NAPO	6.72
LASY	2.83
P807	21.2
GOUD	14.3
FIBR	37.8
CAUV	35.2
O504	-20.9
SALZ	-15
MADC	30.7
THLI	55.7
LAMO	-64.3
ORES	-70.6
GRSY	-80
ALUA	-105
RUSI	-106
VNDP	-107
LOSP	-82.2
O613	79.2
O611	141

North vel

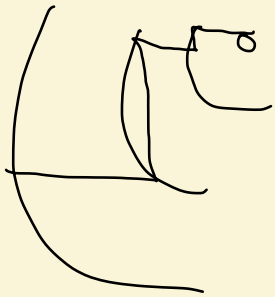
East vel

E	N	x	fault
1	3		7

down hill method

- low points
  - gradient decent

- Taxi Cab Method
  - steps in a direction
  - how many steps?

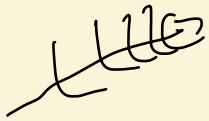


Slow when

- parameter space

Penalty function not aligned with parameter axis

Powels Methods uses a vector for decent in Parameter space



probability is?

- related to
  $-2 \times$ normal^2 misfit

- have misfit higher value
  (less negative,
  Higher value more
  negative)

Scipy. minimize

- function

- X0 model parameters

- method

- bounds

- constraints

fun - lowest value of penalty function

direct - Hooke's directions

nfev - How much evaluated

State - V
D

don't get the same because its non linear and starting direction dependent

X = resol. x[6] .. ..

V = slipd .. .. .. , X

Scipy optimization penalty function

- Can be used to solve the input for this strategy

MCMC -

Most probable model that fits data well

$$\left[y - (\theta_1 x + \theta_0)\right]^2$$

total square misfit

-

Photo Hunting

Stepsigma – How many paragraphs could make

Git Hub

Keeeeeen

add

- commit

- push