

Linear Algebra

Matrix $\underline{\underline{A}}$ with elements a_{ij}

i : row
j : column

$$\underline{\underline{A}} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & \ddots & & \\ a_{31} & & \ddots & \end{pmatrix}$$

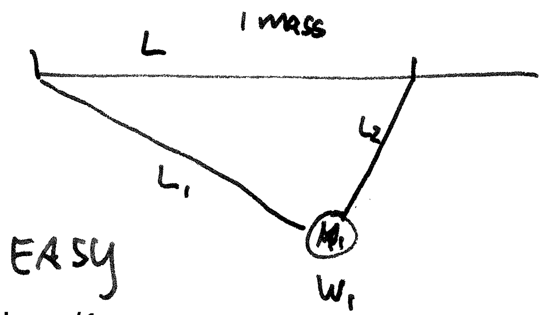
$$\underline{\underline{A}} \cdot \underline{x} = \underline{b}$$

$$\underline{\underline{A}} \cdot \underline{x} = \lambda \underline{x}$$

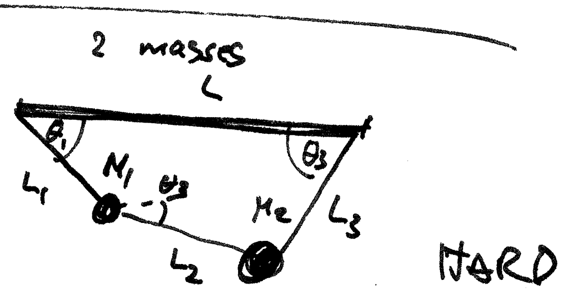
$$\underline{\underline{A}} \cdot \underline{\underline{A}}^{-1} = \mathbb{1}$$ inverse (does not always exist, $\underline{\underline{A}}$ must be $N \times N$ square)

$\det(\underline{\underline{A}})$: determinant

3 Strings/2 Masses Problem



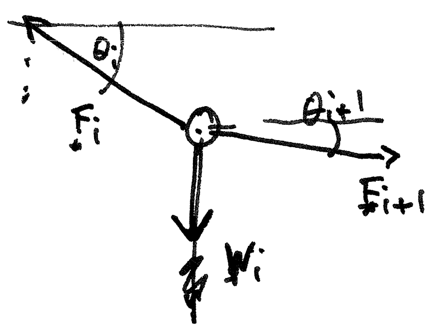
2 strings/1 mass



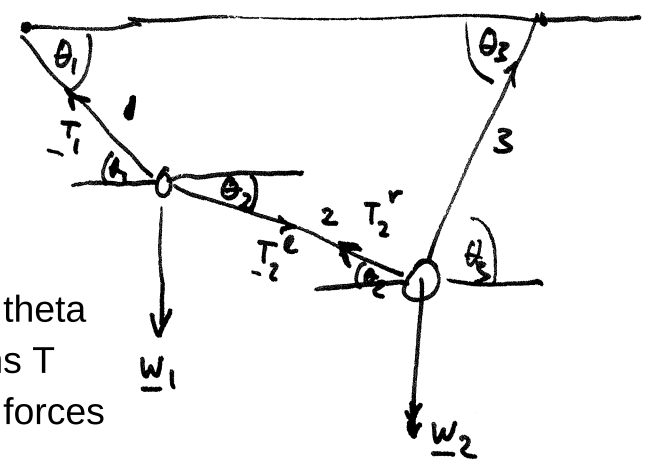
3 strings/2 masses

fixed by L_1, L_2

$\theta_1, \theta_2, \theta_3$ can adjust according to W_1 and W_2, $W_i = M_i g$



$$\underline{F}_i + \underline{F}_{i+1} + \underline{W}_i = 0$$



unknowns
- three angles theta
- three tensions T
(magnitude of forces along strings)

$$\underline{F}_1 = T_1 \begin{pmatrix} -\cos\theta_1 \\ \sin\theta_1 \end{pmatrix}$$

$$\underline{F}_2^l = T_2 \begin{pmatrix} \cos\theta_2 \\ -\sin\theta_2 \end{pmatrix} \qquad \underline{F}_2^r = -\underline{F}_2^l$$

$$\underline{F}_3 = T_3 \begin{pmatrix} \cos\theta_3 \\ \sin\theta_3 \end{pmatrix}$$

$$\underline{W}_1 = \begin{pmatrix} 0 \\ -W_1 \end{pmatrix}$$

$$\underline{W}_2 = \begin{pmatrix} 0 \\ -W_2 \end{pmatrix}$$

Both masses are at rest:

M_1:

$$\boxed{\underline{F}_1 + \underline{F}_2^l + \underline{W}_1 = 0}$$

(1) $-T_1\cos\theta_1 + T_2\cos\theta_2 = 0$

(2) $T_1\sin\theta_1 - T_2\sin\theta_2 - W_1 = 0$ ✓

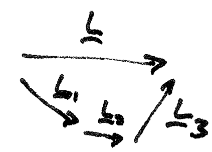
M_2:

$$\boxed{\underline{F}_2^r + \underline{F}_3 + \underline{W}_2 = 0}$$

(3) $-T_2\cos\theta_2 + T_3\cos\theta_3 = 0$

(4) $T_2\sin\theta_2 + T_3\sin\theta_3 - W_2 = 0$

$$\underline{L} = \begin{pmatrix} L \\ 0 \end{pmatrix}$$



$$\underline{L}_1 = L_1 \begin{pmatrix} \cos\theta_1 \\ -\sin\theta_1 \end{pmatrix}$$

$$\underline{L}_2 = L_2 \begin{pmatrix} \cos\theta_2 \\ -\sin\theta_2 \end{pmatrix}$$

$$\underline{L}_3 = L_3 \begin{pmatrix} \cos\theta_3 \\ \sin\theta_3 \end{pmatrix}$$

Geometry:

$$\underline{L} = \underline{L}_1 + \underline{L}_2 + \underline{L}_3$$

(5) $L_1\cos\theta_1 + L_2\cos\theta_2 + L_3\cos\theta_3 = L$

(6) $-L_1\sin\theta_1 - L_2\sin\theta_2 + L_3\sin\theta_3 = 0$

$\cos^2\theta_1 + \sin^2\theta_1 = 1$ (7)

$\cos^2\theta_2 + \sin^2\theta_2 = 1$ (8)

$\cos^2\theta_3 + \sin^2\theta_3 = 1$ (9)

Nine equations for 9 unknowns: $T_1, T_2, T_3, \sin\theta_{1..3}, \cos\theta_{1..3}$

$$\underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_9 \end{pmatrix} = \begin{pmatrix} \sin\theta_1 \\ \sin\theta_2 \\ \vdots \\ T_1 \\ T_2 \\ T_3 \end{pmatrix} \qquad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{array}{c} \begin{pmatrix} \sin\theta_1 \\ \sin\theta_2 \\ \sin\theta_3 \\ \cos\theta_1 \\ \cos\theta_2 \\ \cos\theta_3 \\ T_1 \\ T_2 \\ T_3 \end{pmatrix} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}$$

Write equations as

$$f_i(x_1, x_2 \ldots x_9) = 0 \qquad i = 1 \ldots N$$

$$\underline{f}(\underline{x}) = \underline{0}$$

$$f_1(\underline{x}) = -x_7 x_4 + x_8 x_5 = 0$$
$$f_2(\underline{x}) = x_7 x_1 - x_7 x_2 - W_1 = 0$$
$$f_3(\underline{x}) = -x_8 x_5 + x_9 x_6 = 0$$
$$f_4(\underline{x}) = x_8 x_2 + x_9 x_3 - W_2 = 0$$
$$f_5(\underline{x}) = L_1 x_4 + L_2 x_5 + L_3 x_6 - L = 0$$
$$f_6(\underline{x}) = -L_1 x_1 - L_2 x_2 + L_3 x_3 = 0$$

$$\left.\begin{aligned} f_7(\underline{x}) &= x_4^2 + x_1^2 - 1 = 0 \\ f_8(\underline{x}) &= x_5^2 + x_2^2 - 1 = 0 \\ f_9(\underline{x}) &= x_6^2 + x_3^2 - 1 = 0 \end{aligned}\right\} \text{non-linear!}$$

$$\underline{f}(\underline{x}) = \underline{0}$$

→ root finding!

→ apply Newton-Raphson:

Newton-Raphson in n-D

1D: $f(x) = 0$

$$x \to x + \Delta x$$

$$\Delta x = -\frac{1}{f'} f = -(f')^{-1} f$$

n-D:

Start with $\underline{\tilde{x}}$ and get correction $\Delta \underline{x}$ so that $\underline{f}(\underline{\tilde{x}} + \Delta\underline{x}) = \underline{0}$

Assume $\underline{\tilde{x}}$ is close: expand

$$f_i(\underline{\tilde{x}} + \Delta\underline{x}) \approx f_i(\underline{\tilde{x}}) + \sum_{j=1}^{N} \left.\frac{\partial f_i}{\partial x_j}\right|_{\underline{\tilde{x}}} \Delta x_j + \mathcal{O}(\Delta x^2)$$

$$\underline{f}(\underline{\tilde{x}} + \Delta\underline{x}) \approx \underline{f}(\underline{\tilde{x}}) + \underline{\underline{J}}\, \Delta\underline{x} = \underline{f}(\underline{\tilde{x}}) + \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{\underline{\tilde{x}}} \Delta\underline{x}$$

$$(\underline{\underline{J}})_{ij} = \frac{\partial f_i}{\partial x_j} \qquad \underline{\underline{J}} = \frac{\partial \underline{f}}{\partial \underline{x}}$$

Jacobian

Solve $\underline{f}(\underline{x} + \Delta\underline{x}) = \underline{f}(\underline{x}) + \underline{\underline{J}}(\underline{x})\, \Delta\underline{x} = \underline{0}$ (dropped $\underline{\tilde{x}}$ and just write $\underline{x}$)

Matrix equation: 9 unknowns Δx_i, 9 equations:

$$\underline{f} + \underline{\underline{J}}\, \Delta\underline{x} = \underline{0}$$

or

$$\underline{\underline{J}}\, \Delta\underline{x} = -\underline{f}$$

Formally: solve with inverse $\underline{\underline{J}}^{-1}$ ($\underline{\underline{J}}\,\underline{\underline{J}}^{-1} = \underline{\underline{1}}$):

$$\Delta\underline{x} = -\underline{\underline{J}}^{-1} \underline{f} \qquad \left(\text{compare to } \Delta x = -(f')^{-1} f \ !\right)$$

Use solver for

$$\underline{\underline{A}}\, \underline{x} = \underline{b}$$

- numpy.linalg.solve(), dot() (or ~~declare~~ as matrices)
- test solution by evaluating $\underline{\underline{A}}\,\underline{x} - \underline{b}$

Jacobian

$$(\underline{\underline{J}})_{ij} = \frac{\partial f_i}{\partial x_j} \quad \text{write} \quad \underline{f} = \begin{pmatrix} f_1(x_1, x_2, \dots, x_N) \\ f_2(x_1, \dots) \\ \vdots \\ f_M(\dots) \end{pmatrix} = \underline{f}(\underline{x})$$

$$\frac{\partial f_i}{\partial x_j} \approx \frac{f_i(x_1, \dots, x_j + \frac{h}{2}, \dots) - f_i(x_1, \dots, x_j - \frac{h}{2}, \dots)}{h} + \mathcal{O}(h^2)$$

Calculate a partial derivative by only changing the variable at position j and hold all others fixed. Use central difference algorithm.

$$\underline{h}_j := (0, 0, 0, \dots \underset{\substack{\uparrow \\ j}}{h}, \dots)$$

$$\underline{x} + \tfrac{1}{2}\underline{h}_j = (x_1, \dots, x_j + \tfrac{h}{2}, \dots)$$

$$\frac{\partial f_i}{\partial x_j} \approx \frac{f_i(\underline{x} + \frac{1}{2}\underline{h}_j) - f_i(\underline{x} - \frac{1}{2}\underline{h}_j)}{h}$$

$$f_{ij} \equiv \frac{\partial f_i}{\partial f_j}$$

$$\underline{\underline{J}} = \begin{bmatrix} f_{11} & f_{12} & f_{13} & \dots \\ f_{21} & f_{22} & f_{23} & \dots \\ \vdots & & & \end{bmatrix} \leftarrow$$

In numpy: f(x + hj/2) will produce a whole column (all the i for one fixed j) in one operation.

$$= \left[\begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \end{pmatrix} \begin{pmatrix} \\ \\ \\ \end{pmatrix} \dots \right]$$

$$\frac{\underline{f}(\underline{x} + \frac{1}{2}\underline{h}_1) - \underline{f}(\underline{x} - \frac{1}{2}\underline{h}_1)}{h}$$

```
hj = np.zeros(N)
hj[j] = h
J[:, j] = (f(x+hj/2) - f(x-hj/2))/h
```