http://freeswmm.ddns.net

→ Python

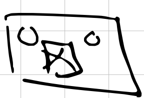
Network (general)

Hydraulics

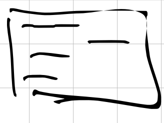
Python

Intro to EPANET

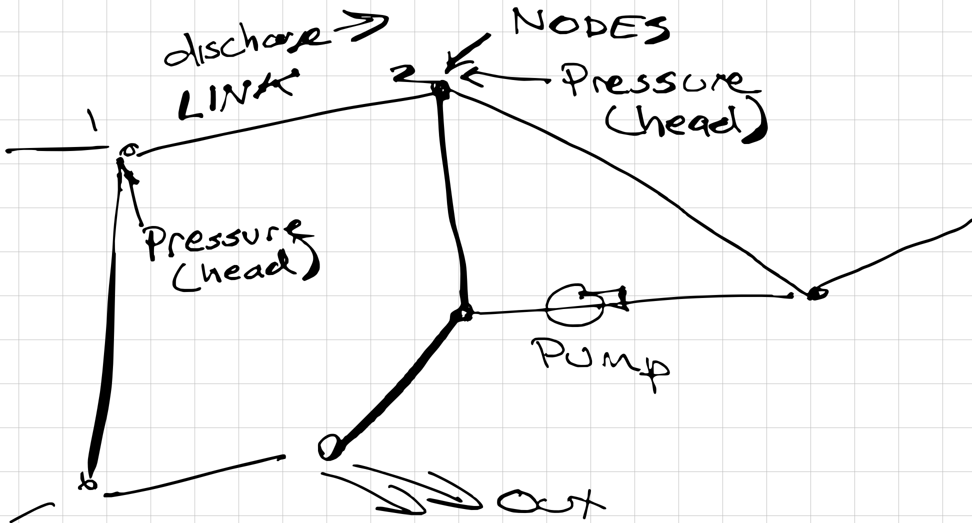
EPANET

 GUI

TOOLKIT



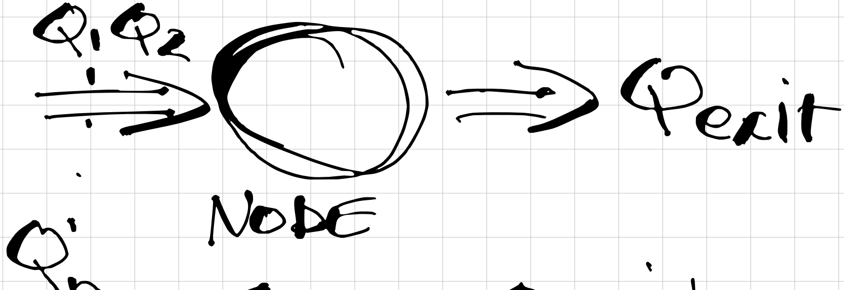
Topology



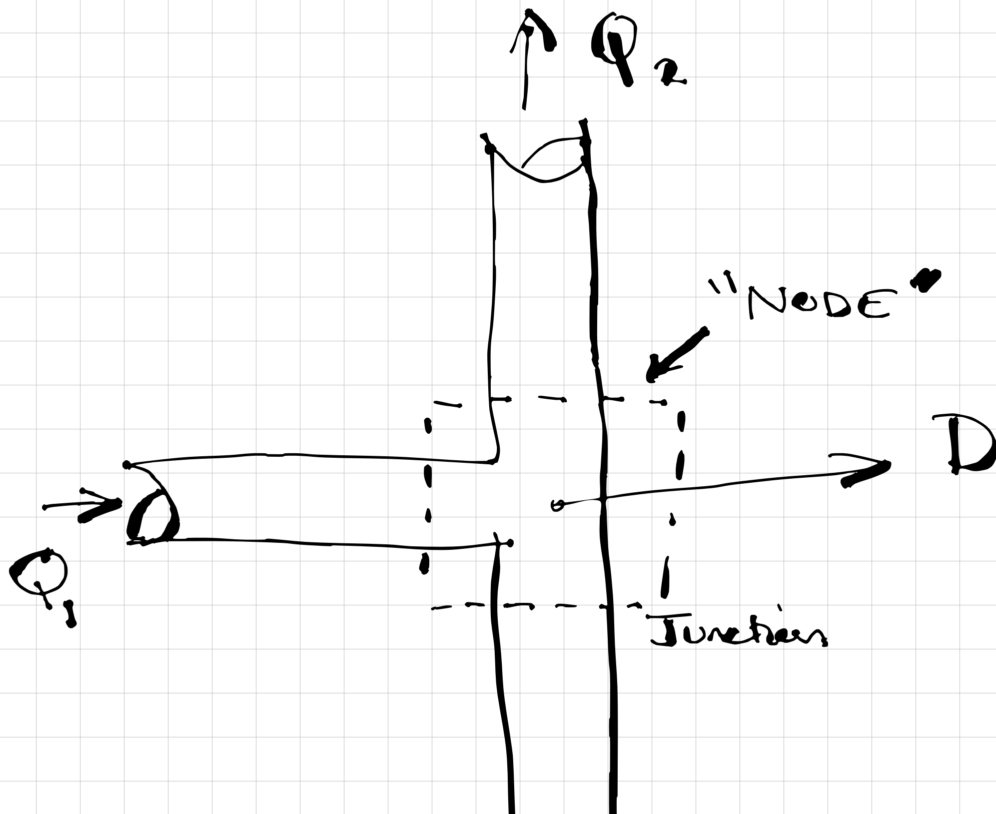
pressures (min required by statute)

flows (physics)

Continuity (at nodes)



$$\Sigma Q_i = Q_{exit}$$



CONTINUNITY

$$\boxed{Q_1 - Q_2 - Q_3 - D = 0}$$

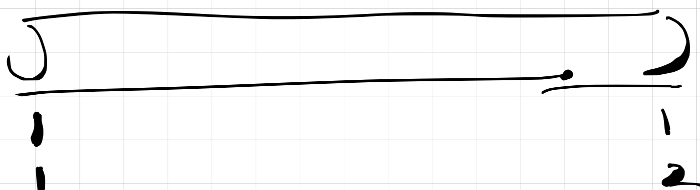
$$Q_1 - Q_2 - Q_3 = D$$

By Convention

"D" Demand

- Demand always at nodes
+ Demand is <u>leaving</u>

Pipe



$$\frac{p_1}{\gamma} + \boxed{\frac{V_1^2}{2g}} + z_1 = \frac{p_2}{\gamma} + \boxed{\frac{V_2^2}{2g}} + z_2 + h_f$$

Modified bernoulli for pipe

$$\frac{p_1 - p_2}{\gamma} = h_f$$

model of frictional losses

- Chezy-Manning (not used much in pipes)
- Hazen-Williams (not used much outside USA)
- Darcy-Weisbach

$$h_f = f \frac{L}{D} \underbrace{\left[\frac{V^2}{2g}\right]}_{\text{Velocity head in the link}}$$

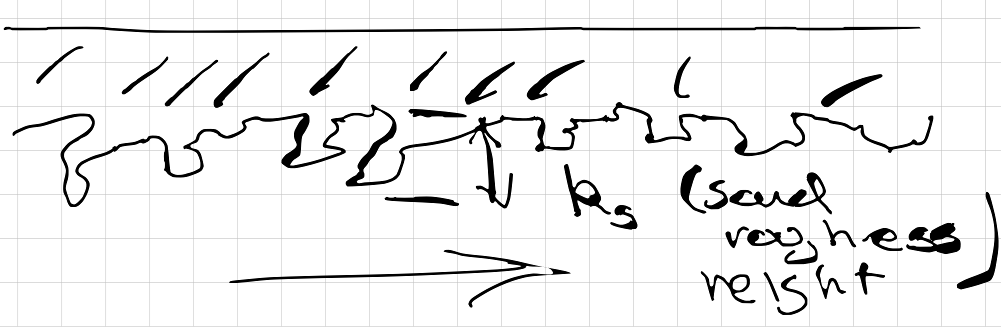
f - "friction factor"

$f \propto Re$, material

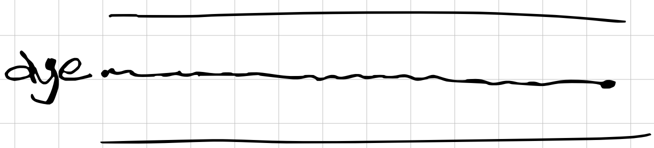
Re: "fluid property"

material: pipe material property

→ roughness



Laminar Flow

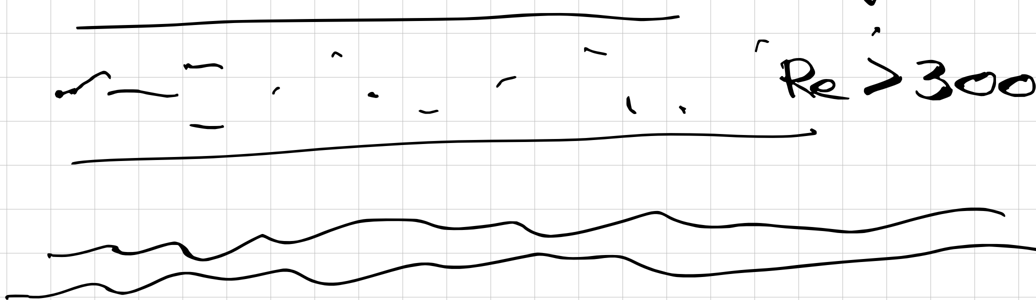


Re < 2000

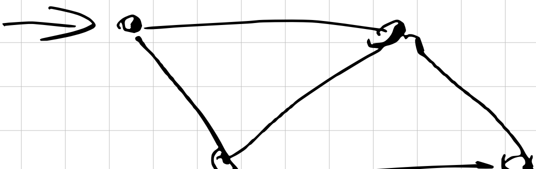
transition

Re > 3000

Turbulent



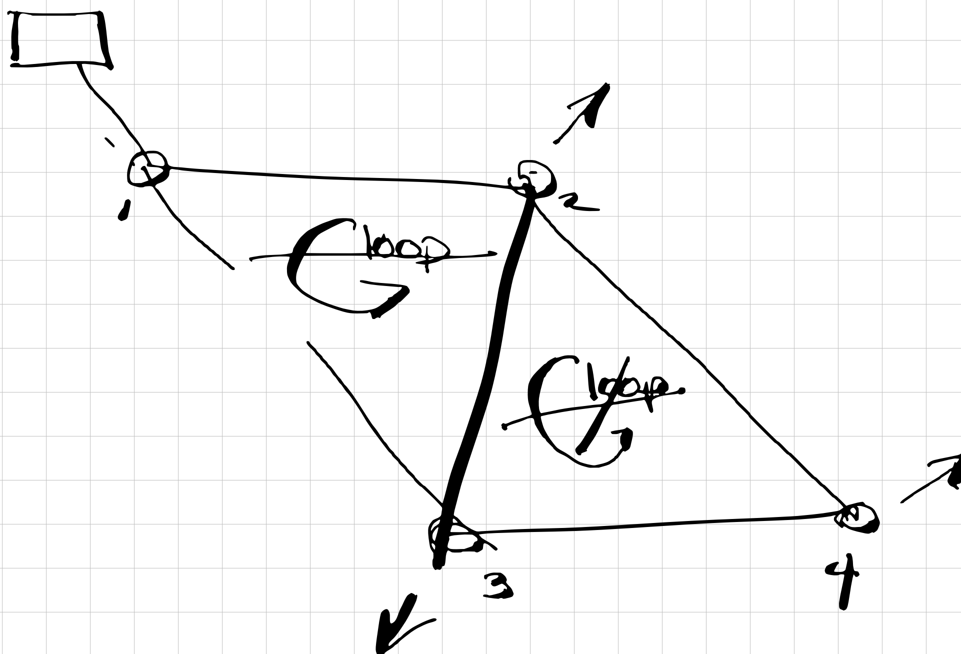
"solve" for pressures at each node & flows in each pipe



Hardy-Cross (Structural engrs.)

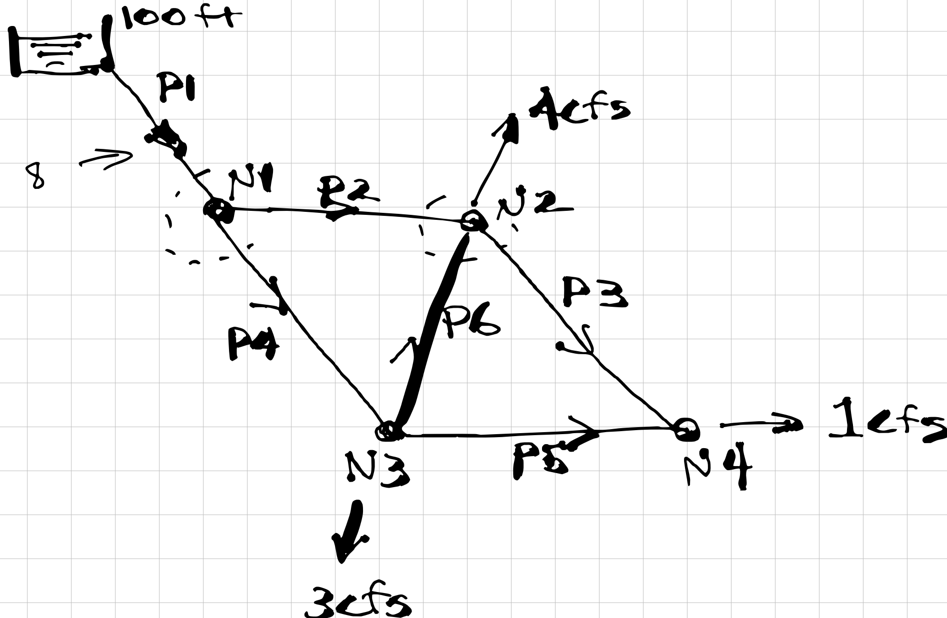
Newton Raphson

Hamam & ~~Ba~~ameller (hybrid)



4 nodes

5 pipes



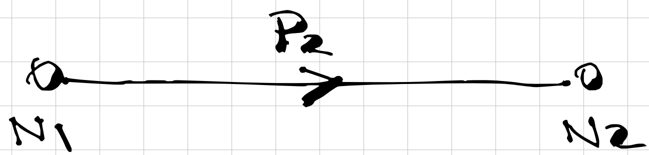
$$[Q_1, Q_2 \cdots \cdot Q_6, H_1, H_2, H_3, H_4] = \underline{x}$$

Continuity for node 1.

$$Q_1 - Q_2 \qquad - Q_4 = 0$$

$$Q_2 - Q_3 \qquad Q_6 = 4$$

$\vdots$

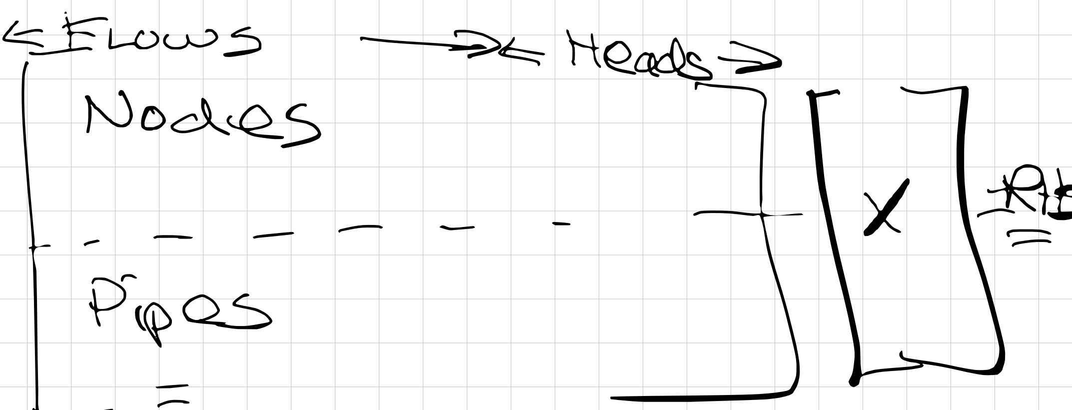


$$H_1 - f\frac{L}{D}\frac{V^2}{2g} = H_2$$

$\vdots$

$$H_1 - \underbrace{f\frac{8L_2}{\pi^2 D_2^5 g}|Q_2|}_{L(Q)}Q_2 = H_2$$

$$-L_2 Q_2 + H_1 - H_2 = 0$$



$$\frac{V^2}{2g}$$

$$Q = VA$$

$$Q = V \frac{\pi D^2}{4}$$

$$Q^2 = \frac{V^2 \pi^2 D^4}{16}$$

$$\frac{16 Q^2}{\pi^2 D^4} = V^2$$

$$f \frac{L}{D} \frac{V^2}{2g} = f \frac{L}{D} \frac{\overset{8}{\cancel{16}} Q^2}{\pi^2 D^4} \frac{1}{\cancel{2} g}$$

$$= f \frac{L}{\cancel{D}} \frac{8 Q^2}{\pi^2 D^5 g}$$

unknown

$$L_1 = \frac{8 f l_1}{\pi^2 g D_1^5} |Q_1|$$

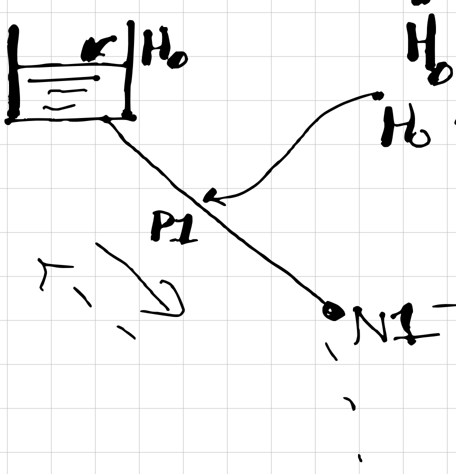
$$L_2 = \frac{8 f l_2 |Q_2|}{\pi^2 g D_2^5}$$

"length"

$$L_3 = \frac{8 f l_3 |Q_3|}{\pi^2 g D_3^5}$$

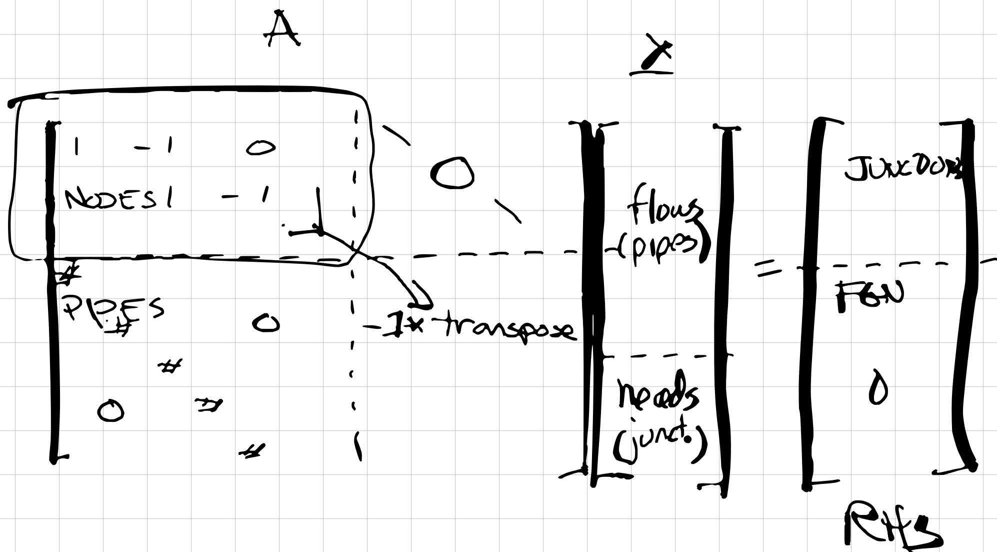
$\vdots$

head (pressure + elev.)



$$H_0 - L_1 Q_1 = H_1$$

$$H_0 - \frac{8 f l_1 |Q_1| Q_1}{\pi^2 g D_1^5} = H_1$$



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

$$\underline{x} = A^{-1}\,RHS$$

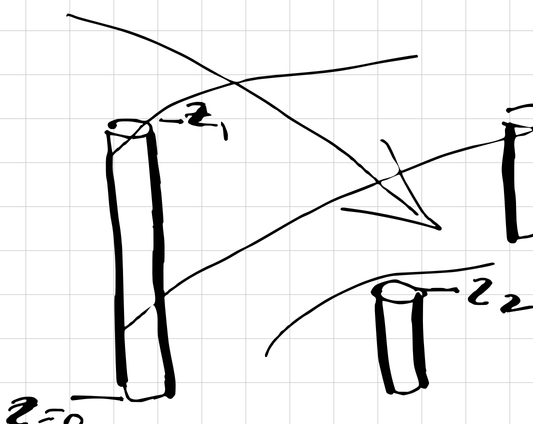
$$A = F(Q)$$

Hydraulics in Python

$$[A][x] = [\underline{rhs}] \quad \text{Non-linear.}$$

multivariate Newton.

- Script (program)(function)

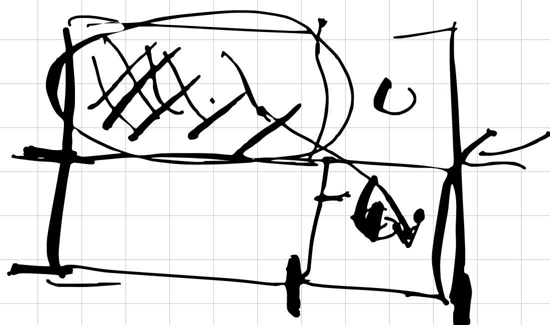


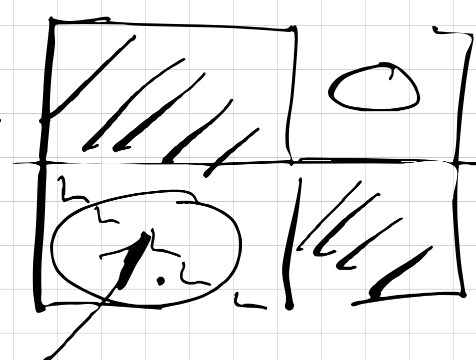
k_s

$$Ax - rhs = \boxed{f(x) = 0}$$

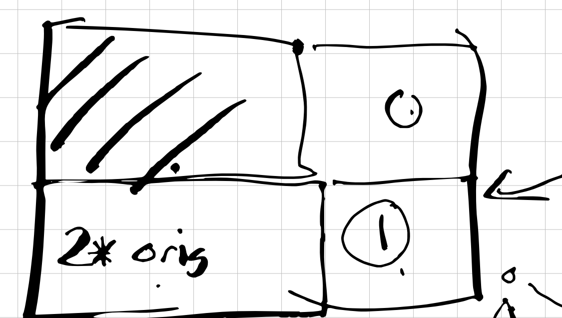
$$\underline{x}_{k+1} = \underline{x}_k - \frac{f(x_k)}{\left.\frac{df}{dx}\right|_{x_k}} \qquad f(x_k) \Rightarrow 0$$

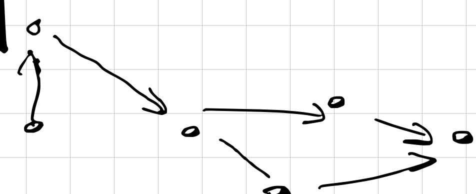
orig.

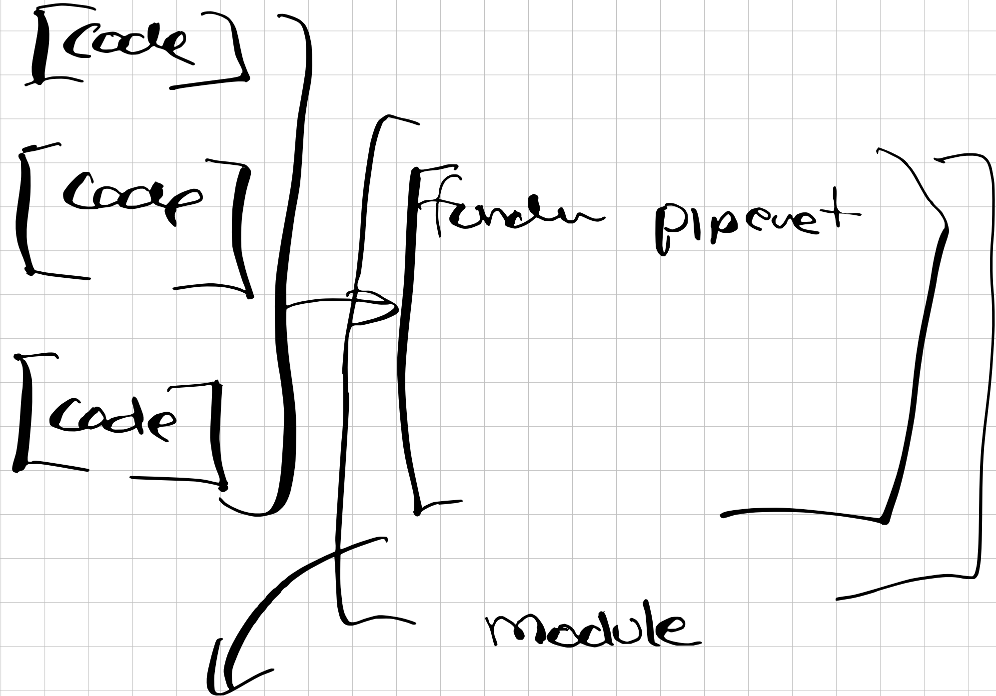




Jacobian







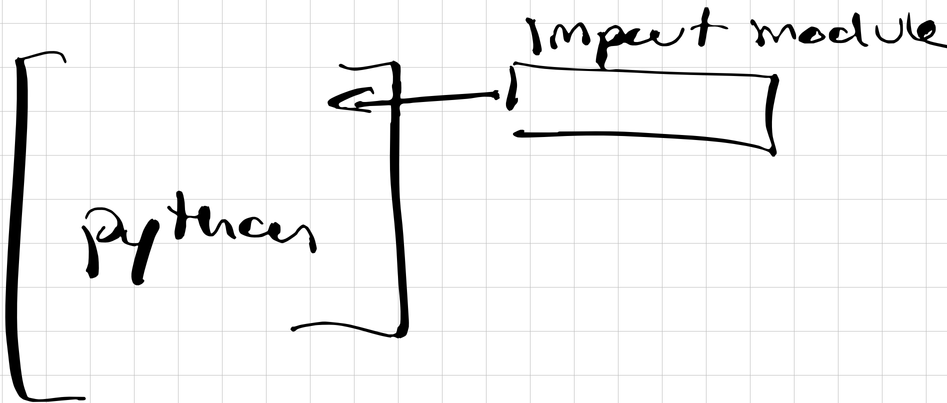
function name (filename)

output

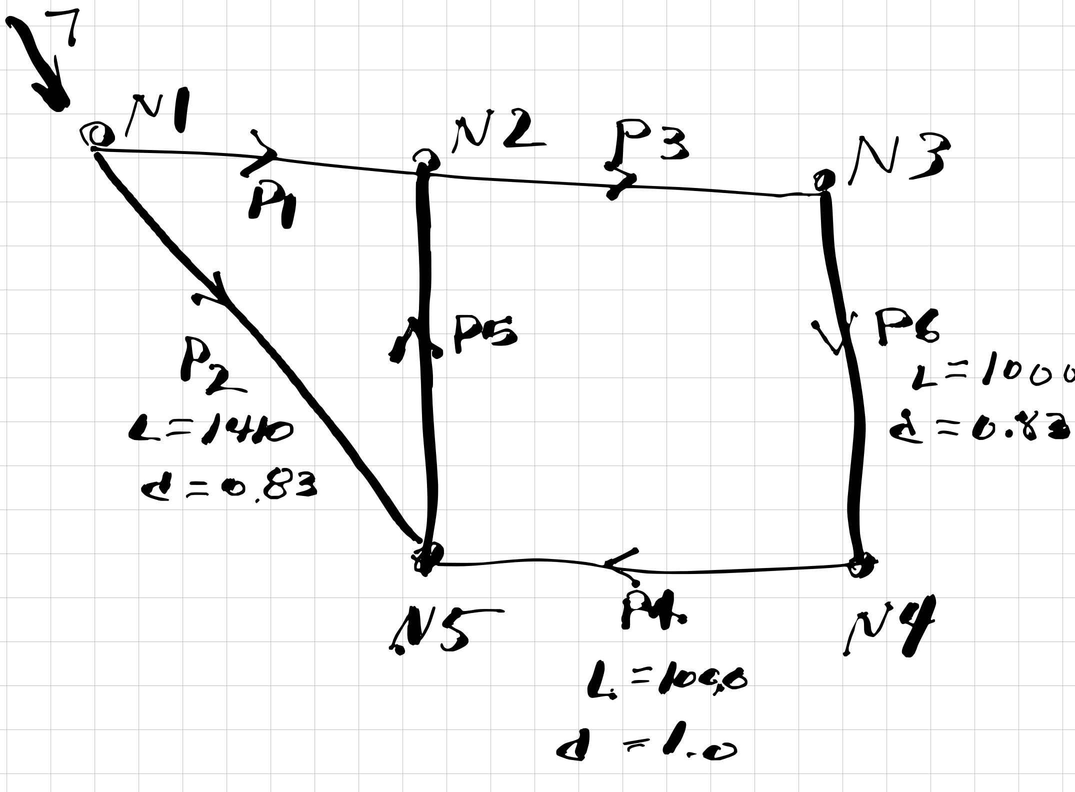
module

module store as file.py

Importing



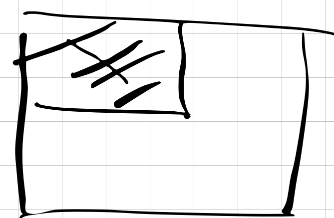
module name.function (argum)



LINKS →

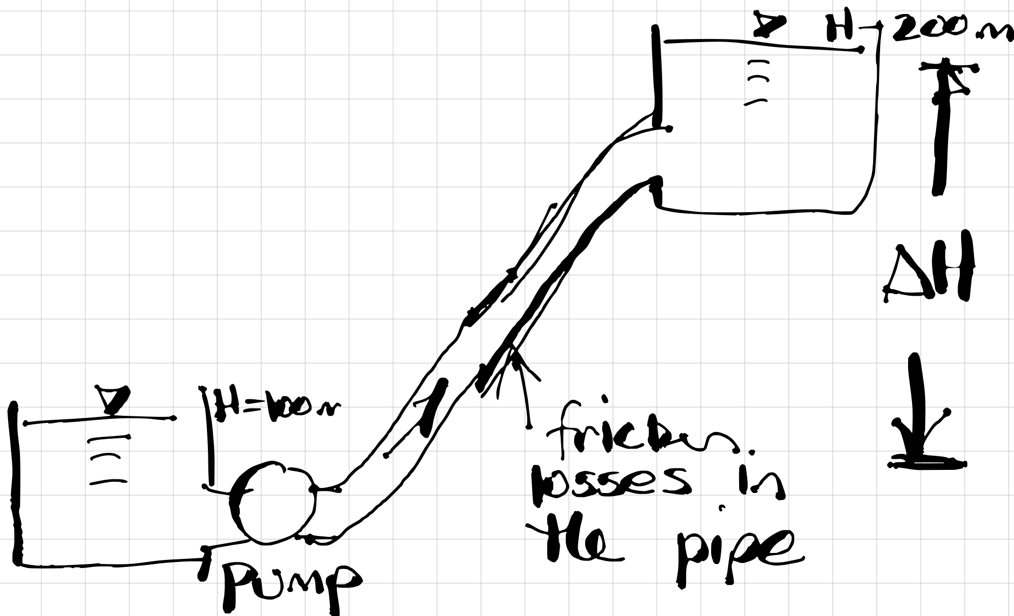
NODE	1	2	3	4	5	6	7	RHS
1	-1	-1	0	0	0	0	1	0
2	1	0	-1	0	1	0	0	0
3	0	0	1	0	0	-1	0	10
4	0	0	0	-1	0	1	0	5
5	0	1	0	1	-1	0	0	0

- introduced/refreshed
  bernoulli's eqn for pipe
  continuity for node
- assemble node-arc matrix



- Newton-Raphson technique to solve the non-linear system.
- Examples

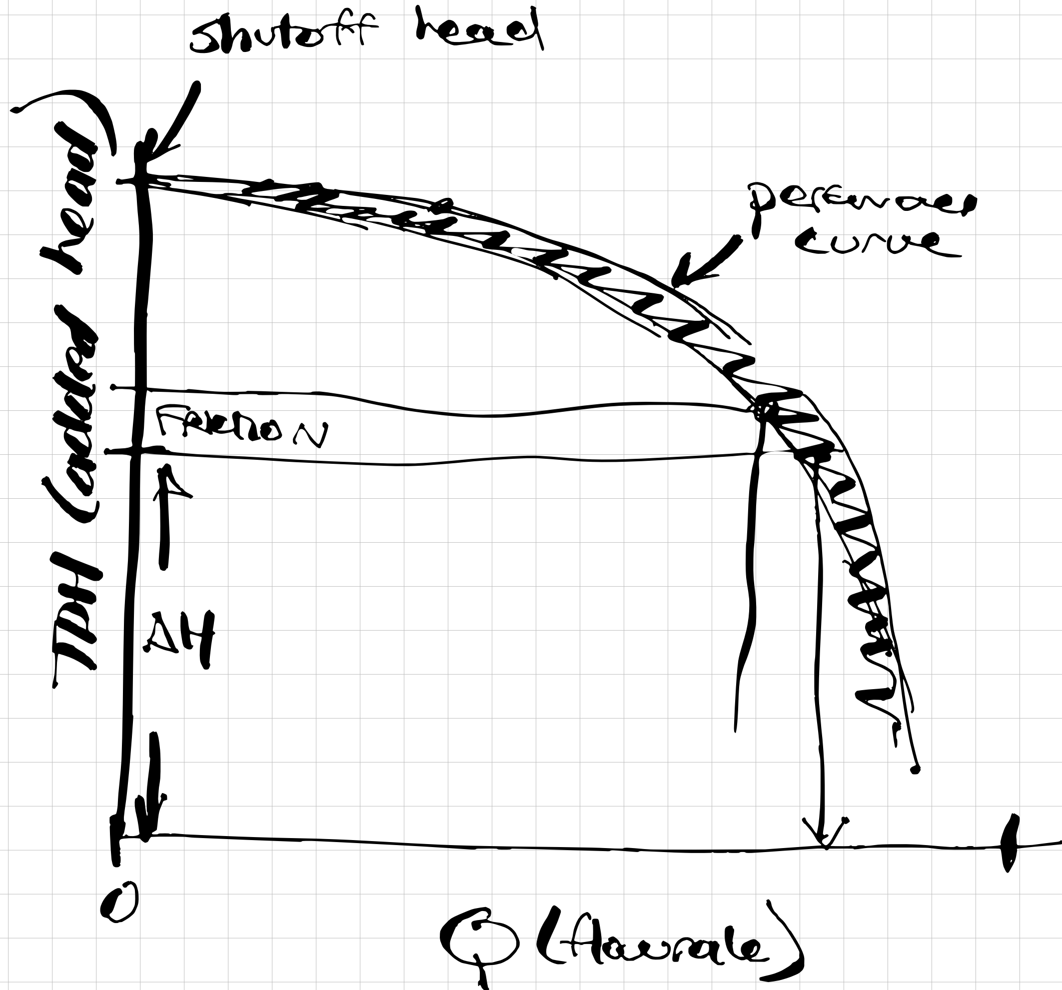
Pump?



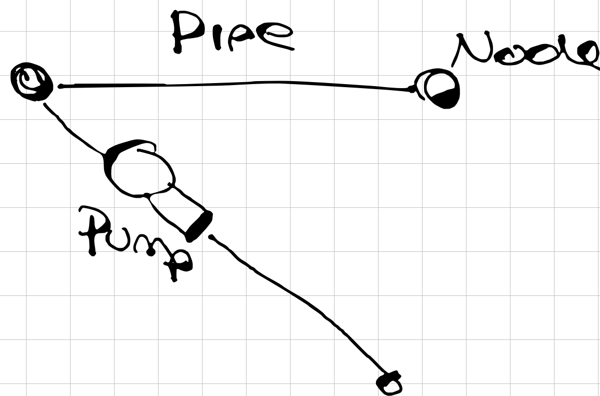
Pump "adds" head to
(energy)

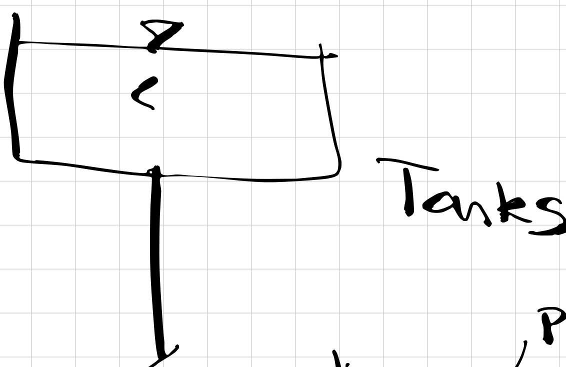
move water uphill.

- Follow a pump performance curve



Shutoff head
performance curve
TDH (added head)
FRICTION
ΔH
0
Q (flowrate)





links
- pipes
- valves
- pumps

nodes
- junctions
- reservoirs/ tanks

OMA - metropolice