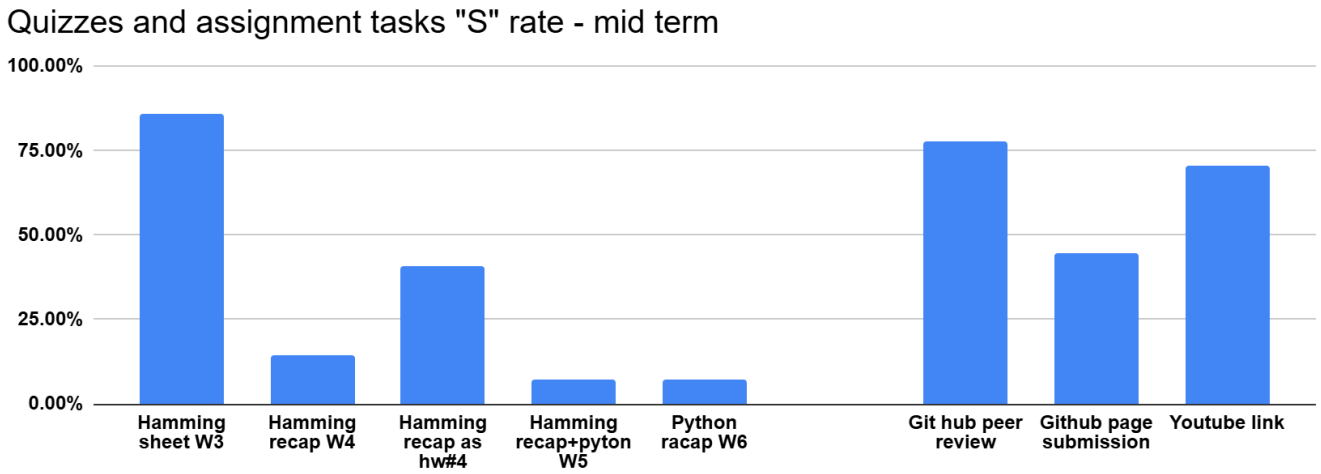
ENR145 Computational Methods:
Mid-term review

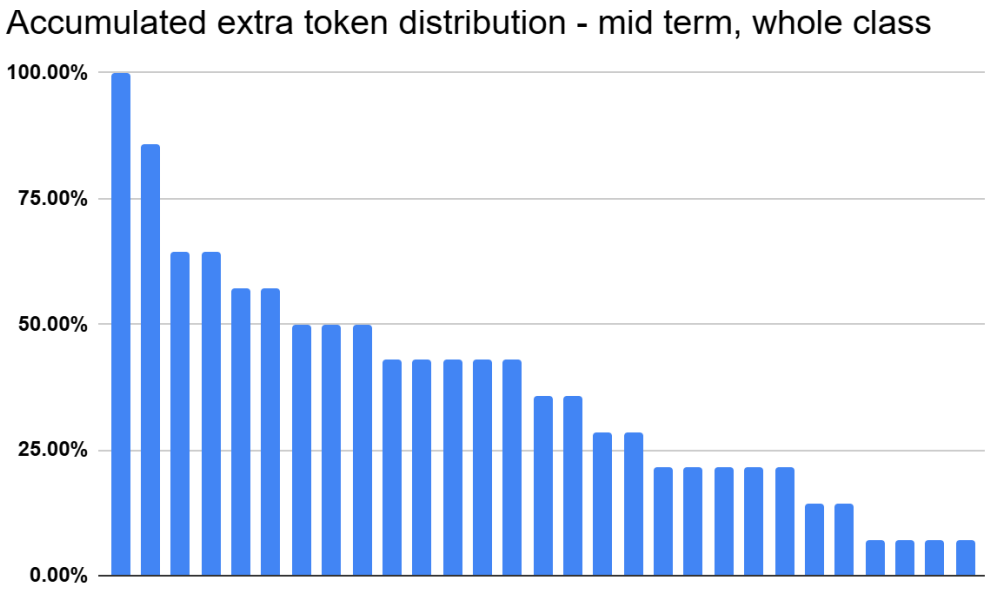
Xiang Li

Spring 2026

COE COLLEGE

ENR 145 has coding issues?





From numbers, math to codes:

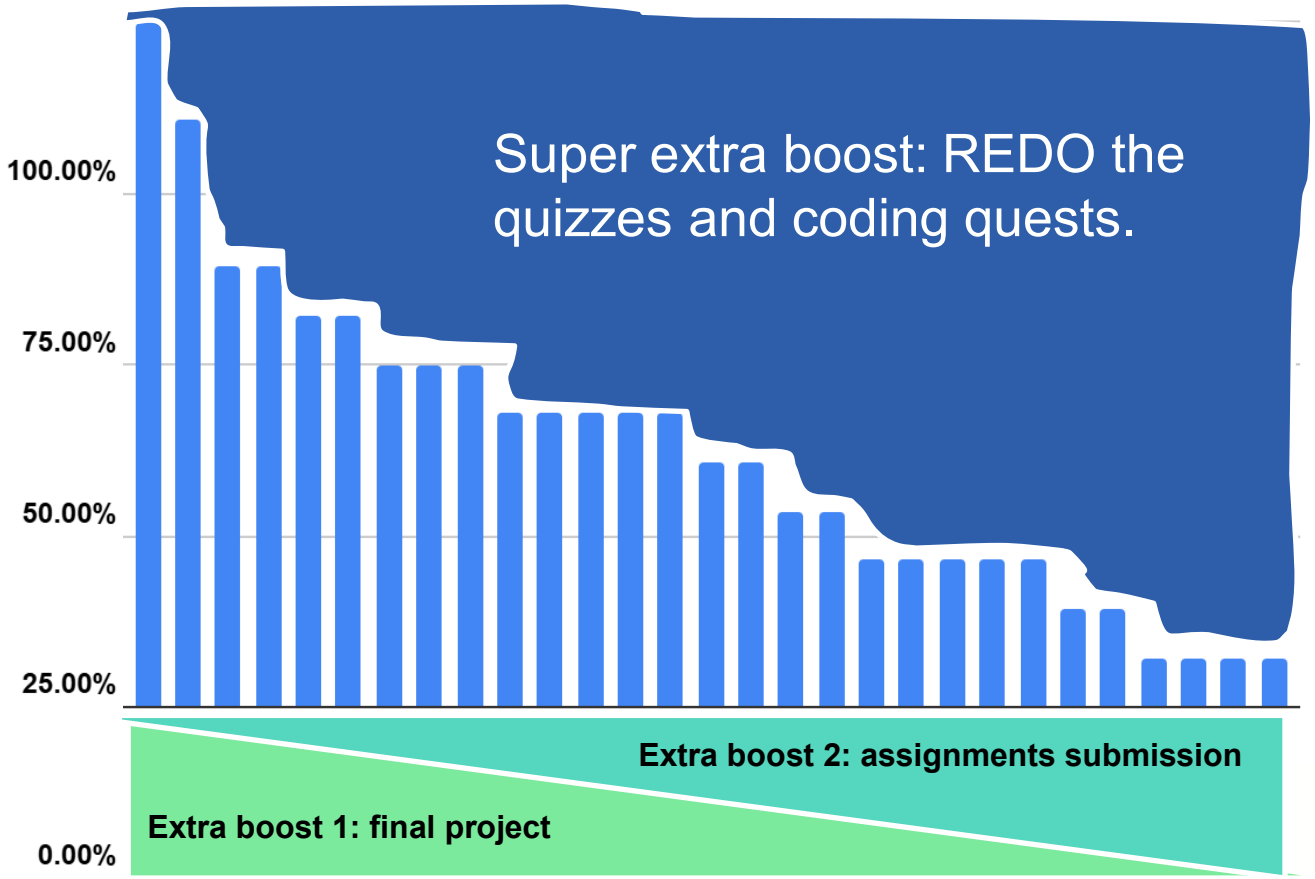
- Decreased “S” completion of subject matter.

- Accumulated token bump and toolbox bump won’t generate enough A grade (to my taste).

COE COLLEGE

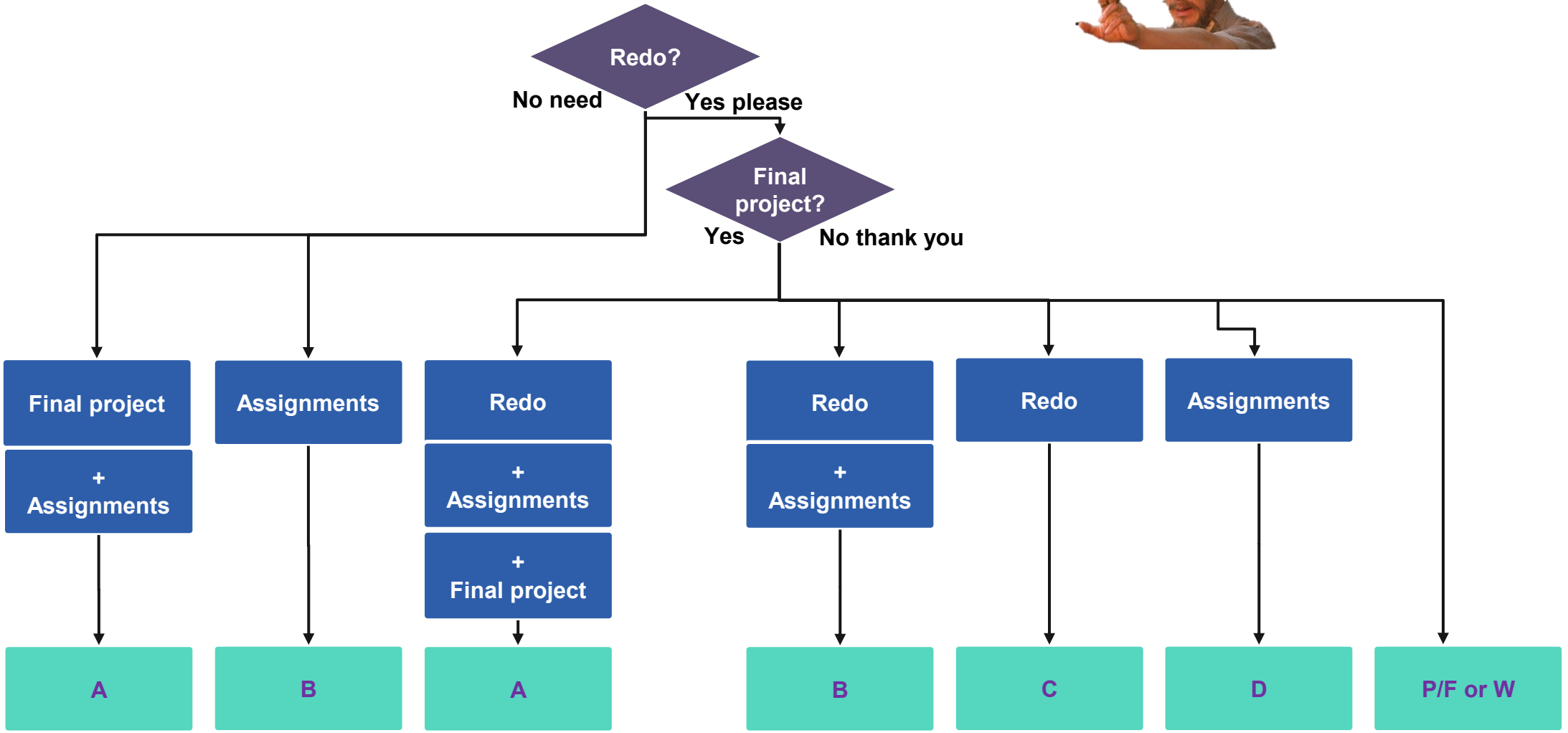
Guys we can fix this:

- *Easy assignments from now on.
- Chance to redo the quizzes and coding tasks.
- Final project optional.



*Easy as no “skill checks” in quizzes. But submission deadline rules still apply.

Possible outcome?



COE COLLEGE

When redo?

1. Individual appointment
2. Bit sized task sessions (Hamming math, Hamming python, Hamming numpy… etc)
3. The tasks won't be the same as the old quizzes, it's a "skill check".

And finally: class room engagement issue: we need to talk about the cellphone use.

We will still have a quiz this week about numpy and “hamming numpy”

$$M3 = M1 \times M2 = \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} \times \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} = \begin{bmatrix} a_1 \times a_2 + b_1 \times c_2 & a_1 \times b_2 + b_1 \times d_2 \\ c_1 \times a_2 + d_1 \times c_2 & c_1 \times b_2 + d_1 \times d_2 \end{bmatrix}$$

So, a [2,3] x [3,4] = [2,4]

```python
I_1 = np.eye(11, dtype=int) # we define data tyep to int so we save some storage (default number type in numpy is over-kill for this application)
## TODO: encoding P1-P4, and P0 based on our theory, our SoS matrix:
# P1 checks: D1, D2, D4, D5, D7, D9, D11
# P2 checks: D1, D3, D4, D6, D7, D10, D11
# P3 checks: D2, D3, D4, D8, D9, D10, D11
# P4 checks: D5, D6, D7, D8, D9, D10, D11
P_1 = np.array([
  # P1 P2 P3 P4
    [1, 1, 0, 0],  #D1
    [1, 0, 1, 0],  #D2
    [0, 1, 1, 0],  #D3
    [1, 1, 1, 0],  #D4
    [1, 0, 0, 1],  #D5
    [0, 1, 0, 1],  #D6
    [1, 1, 0, 1],  #D7
    [0, 0, 1, 1],  #D8
    [1, 0, 1, 1],  #D9
    [0, 1, 1, 1],  #D10
    [1, 1, 1, 1],  #D11
])
G_1 = np.concatenate((I_1, P_1), axis=1)  # this is just a simple function to stich two matrix together
# print (G_1)
```

COE COLLEGE

We will still have a quiz this week about numpy and hamming numpy 

```
#3.2 now let's do a [15,16] "G_2 matrix" (I2|P0)
I_2 = np.eye(15, dtype=int)
# print (I_2)
P_0 = np.ones((15, 1), dtype=int) # total parity check P0 is going to Sum all the bits (d1-d11, P1-P4)
G_2 = np.concatenate((I_2, P_0), axis=1)
# print (G_2)

#3.2 the ultimate generator matrix [11,5] X [15,16] = [11,16]
G = G_1 @ G_2 %2
```

```
# this error checking code, is basically your P matrix sitting on top of a (5x5) "eye" matrix

#We just need the "non-eye" part of the generator matrix (why?)
#                      -------------
#                      |This part  |
#[[1 0 0 0 0 0 0 0 0 0 0|1 1 0 0 1] |
# [0 1 0 0 0 0 0 0 0 0 0|1 0 1 0 1] |
# [0 0 1 0 0 0 0 0 0 0 0|0 1 1 0 1] |
# [0 0 0 1 0 0 0 0 0 0 0|1 1 1 0 0] |
# [0 0 0 0 1 0 0 0 0 0 0|1 0 0 1 1] |
# [0 0 0 0 0 1 0 0 0 0 0|0 1 0 1 1] |
# [0 0 0 0 0 0 1 0 0 0 0|1 1 0 1 0] |
# [0 0 0 0 0 0 0 1 0 0 0|0 0 1 1 1] |
# [0 0 0 0 0 0 0 0 1 0 0|1 0 1 1 0] |
# [0 0 0 0 0 0 0 0 0 1 0|0 1 1 1 0] |
# [0 0 0 0 0 0 0 0 0 0 1|1 1 1 1 1]]|
#                      |This part  |
#                      -------------
```

COE COLLEGE

We will still have a quiz this week about numpy and hamming numpy 🙀

```
# next we just repeat the hamming coding trick, but on [1,16] output
# output_code: [[1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1]]
# you can inject 1 bit error below, anywhere:
received_msg = [[1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1]]
Error_msg = received_msg @ H_T % 2
print ("the error code:")
print (Error_msg)

## TODO: wait, the postion code how?
```

```
the error code:
[[1 0 0 1 1]]
```

COE COLLEGE

If you can do this, you have the power:

Array to array conversion will be so easy:

This is math:

$$P_{i,j} = \sum_{k=1}^{n} Q_{i,k} \times R_{k,j}$$

This is math application:

[2,3] x [3,4] = [2,4]

This is Engineering application:

11 bits in

16 bits out

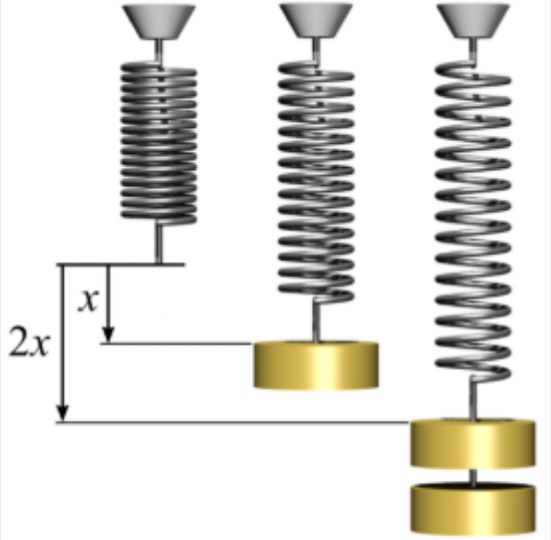
So instead of double for loops, we can do matrix operation to get it in one go?

[1,11] x [?] = [1,16]?

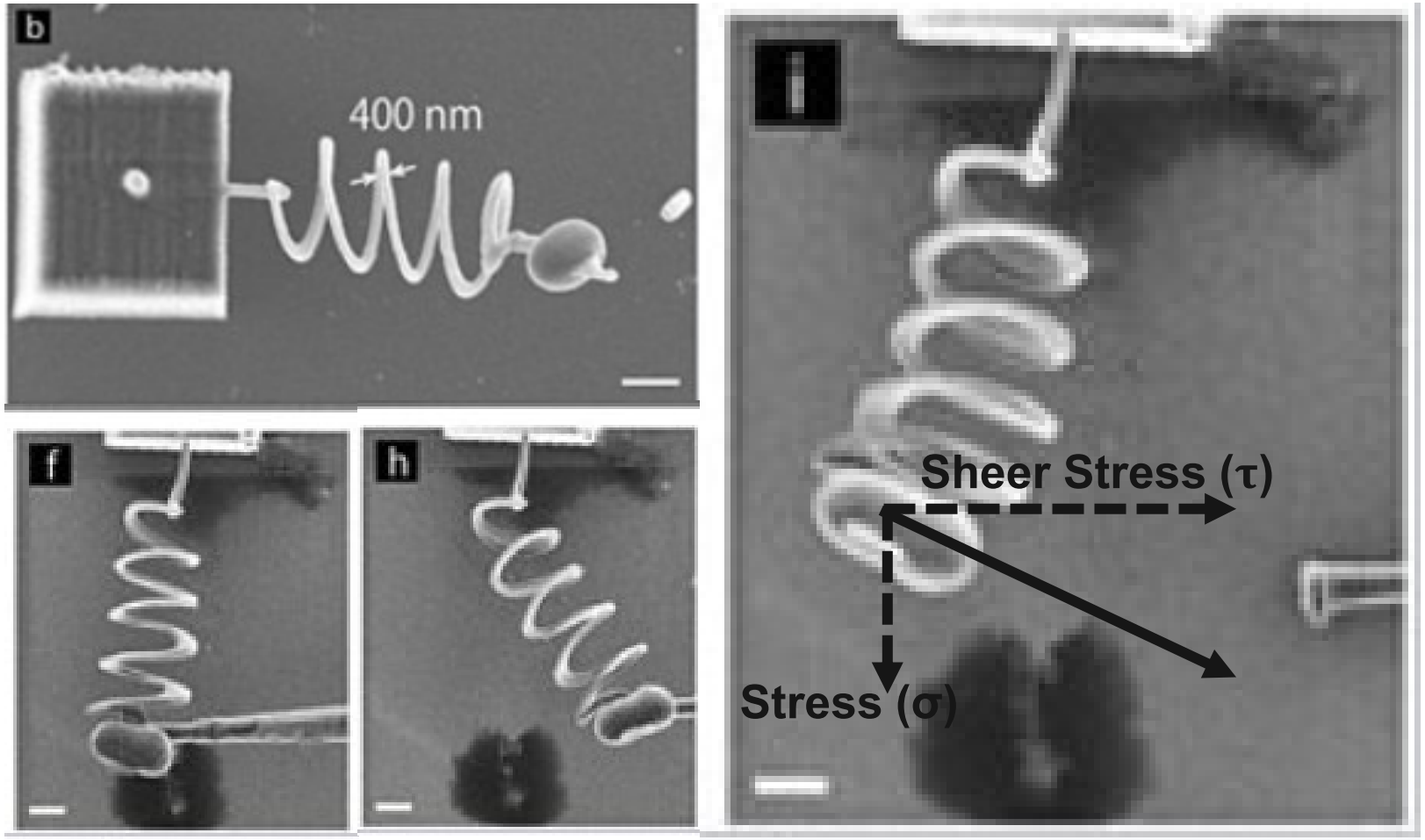
One example to skill check your math power:

Hooke's law

Article Talk



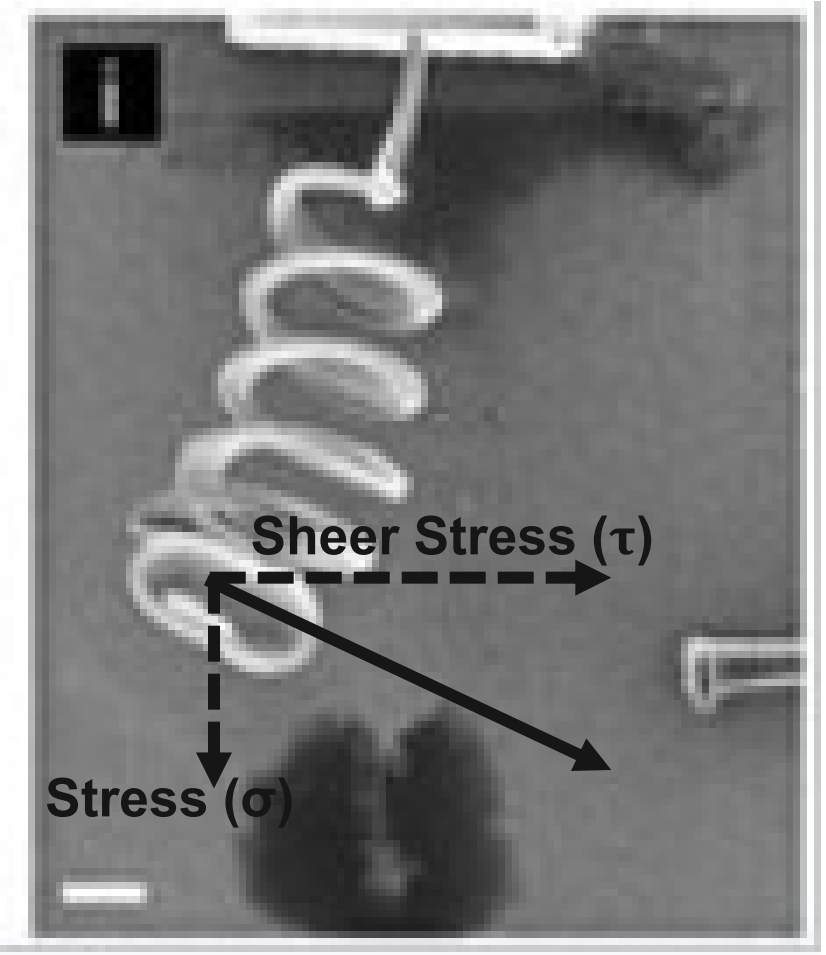
Hooke's law: the force is proportional to the extension $F_s = kx$



COE COLLEGE

https://en.wikipedia.org/wiki/Hooke%27s_law

Yes, math is power (1/3)



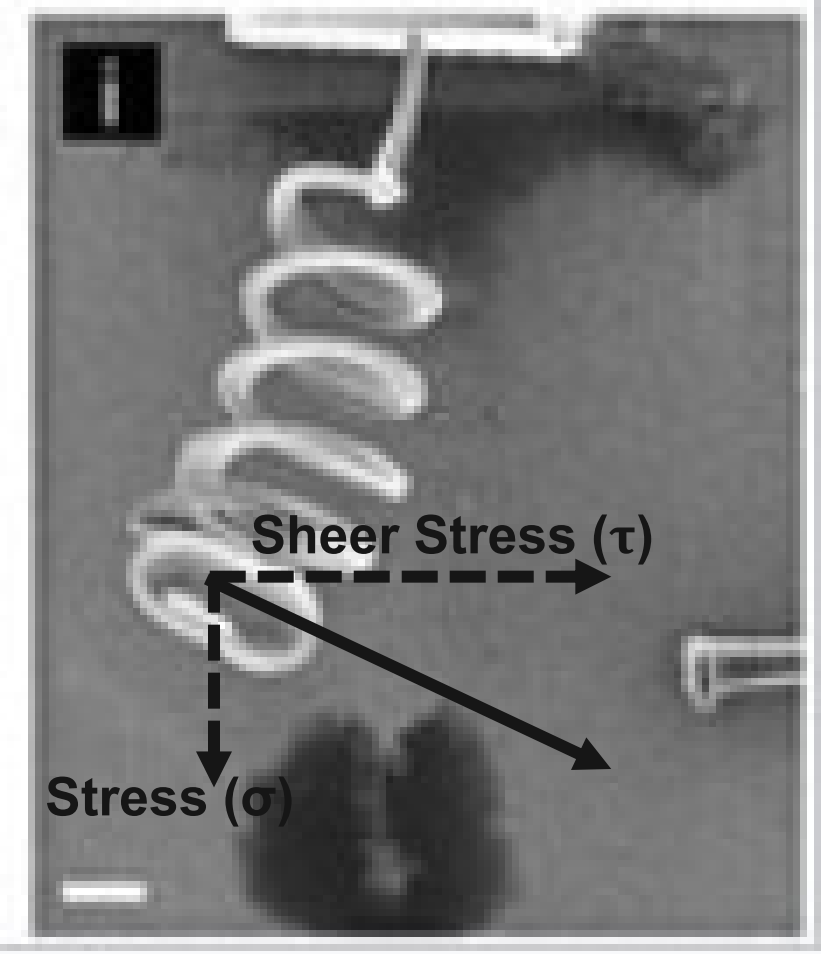
If there's stress (σ), then there's deformation , aka strain (ε)

If 1D, just like $F_s = kx$ We can have σ = Eε, or ε = $\frac{1}{E}$σ

Just like k, E is a constant to deal with force and deformation ratio.

COE COLLEGE

Yes, math is power (2/3)



If there's stress (σ), then there's deformation , aka strain (ε)

If 1D, just like $F_s = kx$ We can have σ = Eε, or ε = $\frac{1}{E}$σ

Just like k, E is a constant to deal with force and deformation ratio.

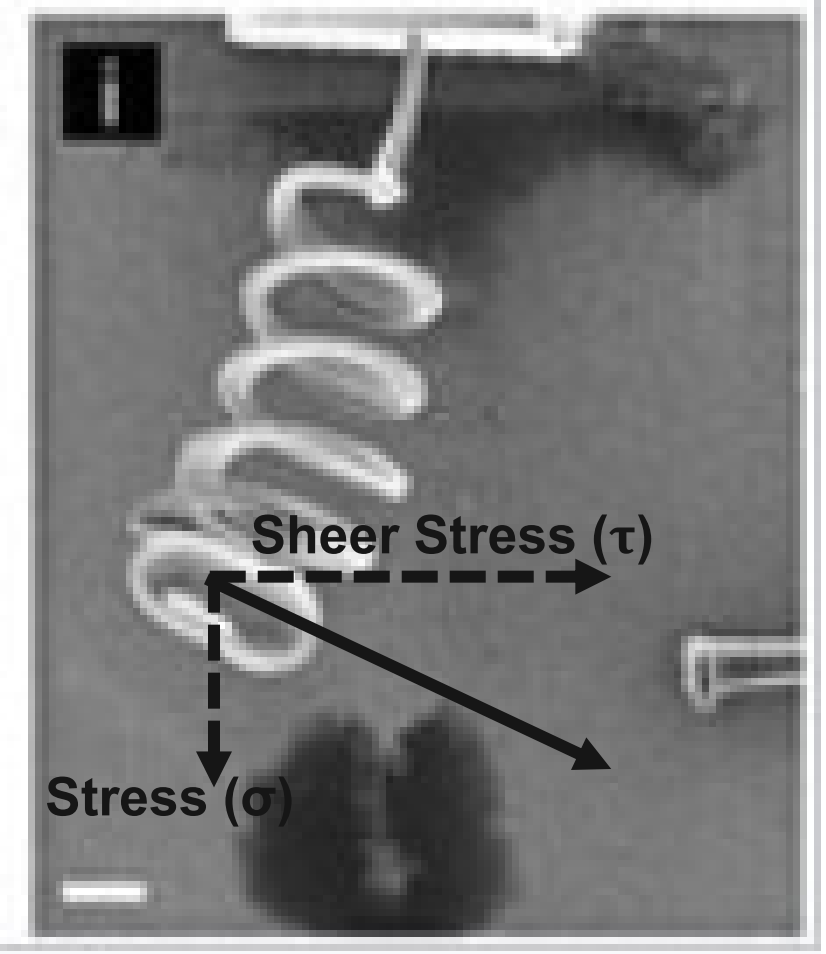
If 2D plane:

$$\begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{pmatrix} = \frac{1}{E} \begin{pmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 1+\nu \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}$$

ν is just another constant to deal with material property.

COE COLLEGE

Yes, math is power (3/3)



If there's stress (σ), then there's deformation , aka strain (ε)

If 1D, just like $F_s = kx$ We can have σ = Eε, or ε = $\frac{1}{E}$σ

Just like k, E is a constant to deal with force and deformation ratio.

If 2D plane:

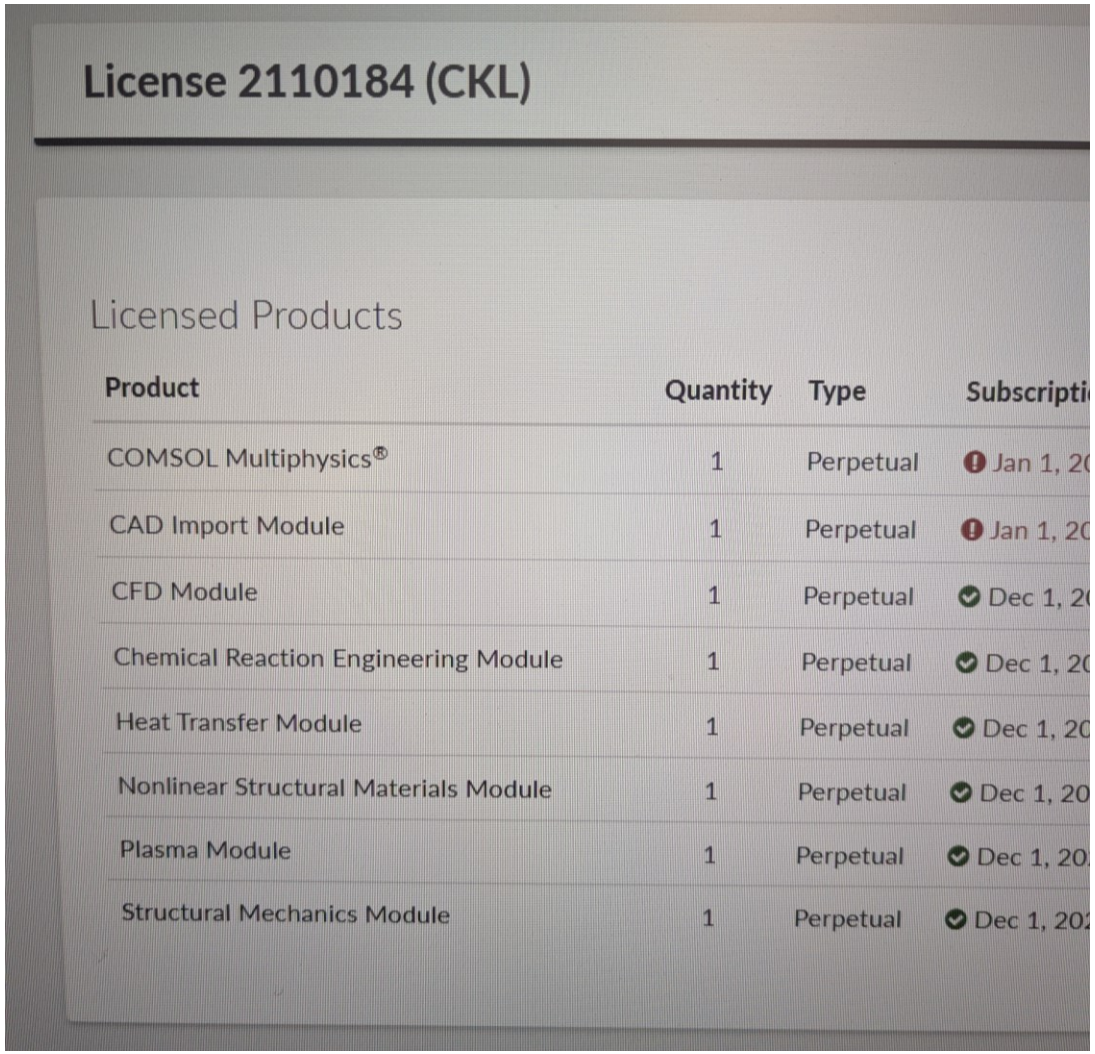
$$\begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{pmatrix} = \frac{1}{E} \begin{pmatrix} 1 & -\nu & 0 \\ -\nu & 1 & 0 \\ 0 & 0 & 1+\nu \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{pmatrix}$$

ν is just another constant to deal with material property.
What if ν = 0?

With the power of ***matrix multiply***, you can understand mechanical engineering without learning about mechanical engineering.

COE COLLEGE

Before we go, I need your help to check the COMSOL on your desktop:



License 2110184 (CKL)

Licensed Products

Product	Quantity	Type	Subscripti
COMSOL Multiphysics®	1	Perpetual	Jan 1, 2(
CAD Import Module	1	Perpetual	Jan 1, 2(
CFD Module	1	Perpetual	Dec 1, 2(
Chemical Reaction Engineering Module	1	Perpetual	Dec 1, 2(
Heat Transfer Module	1	Perpetual	Dec 1, 2(
Nonlinear Structural Materials Module	1	Perpetual	Dec 1, 20
Plasma Module	1	Perpetual	Dec 1, 20
Structural Mechanics Module	1	Perpetual	Dec 1, 202

- **You need to let me know which PC doesn't have access to COMSOL**
- **Which PC doesn't have access to the following packages.**
- **Maybe leave the info in ENR 145 channel?**

COE COLLEGE