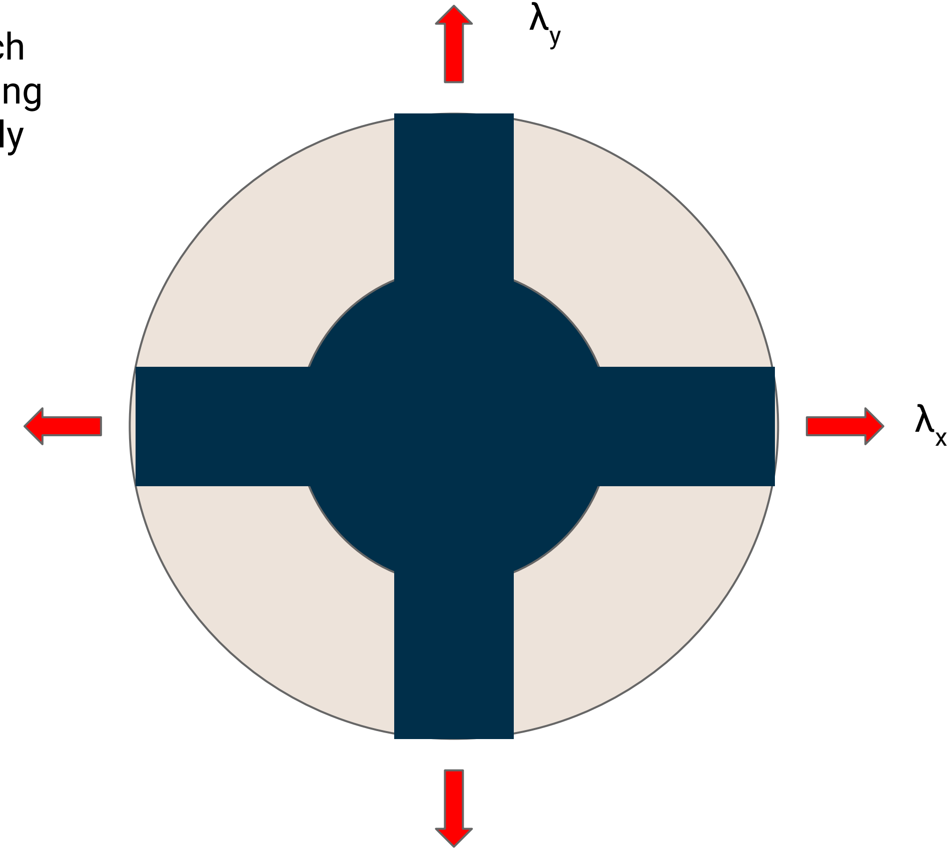
Bilayer Shell Simulation

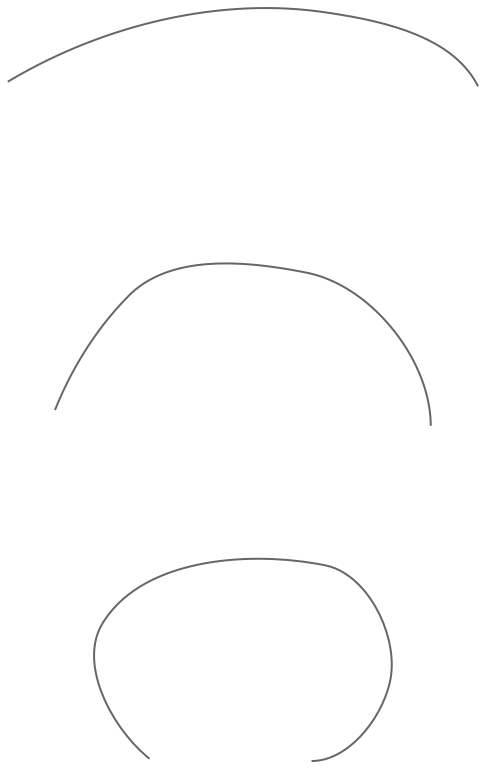
Progress Report

Vishal Kackar, Yunbo Wang, Shyan Shokrzadeh

- $\lambda = 1$ -> no stretch
- Will be considering radial stretch only



Overview of the problem we are trying to solve



- The curvature of the shell is dependent on the amount of pre-stretch applied.
- A larger pre-stretch corresponds to a 3D shape that looks more like a sphere.
- These are just 2D cross-sections of the shell

How does the 3D shape of the shell change with the amount of pre-stretch applied?

Initial Conditions

We give the simulation the relevant material properties and the amount of pre-stretch applied to the substrate layer.

We then calculate the total energy of the system using the stretching energy of the substrate layer before the pre-stretch is released.

$$\text{Energy Initial} = \tfrac{1}{2} * E_s * A_s * (\lambda-1)^2 * L$$

Where: $A_s = w * t_s$

L = reference length

Reference Length

$\mathbf{L}_0$ = reference length at time = 0

$\mathbf{L}_{inf}$ = reference length after a long time

- Make the simulation a function of **L**.
- Start at $\mathbf{L}_0$ and slowly step **L** to $\mathbf{L}_{inf}$
  - $\mathbf{L}_0$ = length of edges initially (pre-stretched)
  - $\mathbf{L}_{inf}$ = original length of the edges

Natural Curvature

The kirigami + substrate layer will develop a natural curvature to account for the pre-stretch.

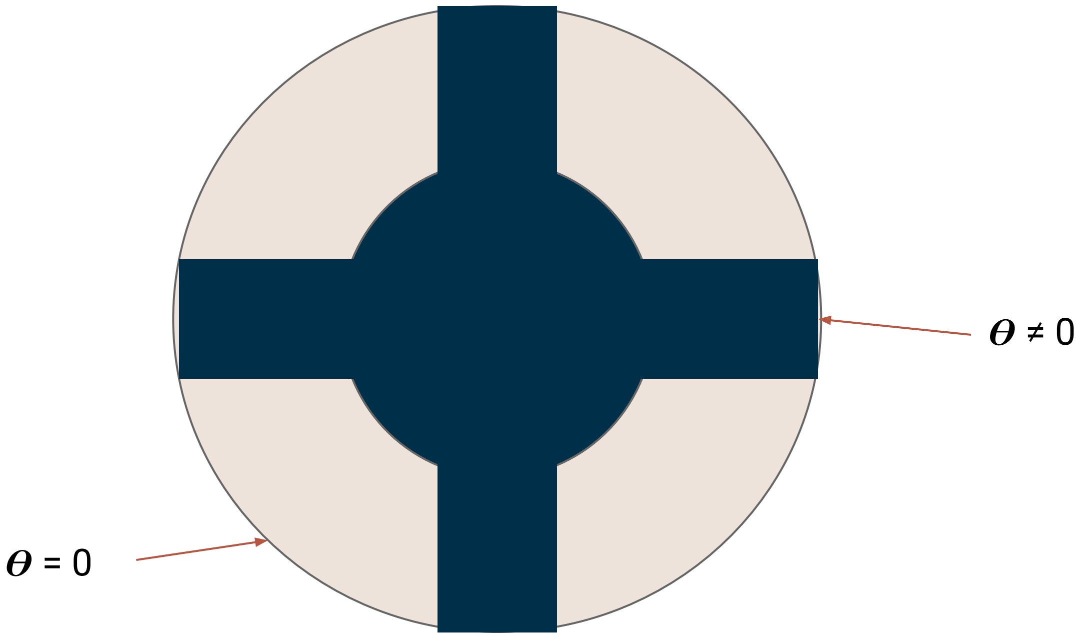
This is where the complicated part of the simulation lies.

Need to account for thickness changing as a result of stretching and the effective I of the two layers (156A).

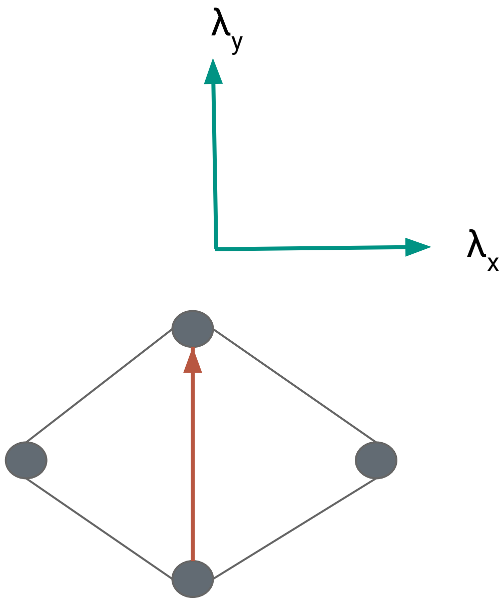
$$\text{Bending Energy} = \tfrac{1}{2} * k_b * \theta^2$$

$$k_b = 2/\sqrt{(3)} * E * I_{eff}$$

Natural Curvature cont.



Theta Bar function



Theta bar is a function of:

- Lambda vector
- Hinge vector
- Material properties
- Time
  - Can't start simulation with the full theta bar

Putting It All Together

computeThetaBar:

- Hinge nodes
- E1, E2
- t1, t2
- Lambda

Gives the theta bar for a particular hinge section

Function of **L**

computeThetaBar then goes into a function similar to gradEb_hessEb_Shell to calculate the total bending energy

Generally:

- More complex mesh than before
- Need to separate kirigami layers from only substrate layers
  - For the moment of inertia
  - For the theta bar
- Computation time can get large
- Compare against Abaqus FEA to test accuracy

Future Work

- Have a more sophisticated weight and mass calculation
- Slowly decrease gravity with time
  - We need gravity to make sure the simulation doesn’t get stuck at a local minimum
- Update the thickness of the substrate layer as a function of time
  - Poisson’s ratio = 0.5
  - The substrate layer starts thin, and slowly increases in thickness as **L** decreases
  - This also affects I_{eff} of the kirigami + substrate region
- Simulation does not converge right now

Framework for the simulation is in place, just need to fine tune it and fix the mistakes