2CM4 Design Project

Shayaan Siddiqui,400247500,sidds41

Currently non-renewable energy sources cause great negative impacts to our environment and future damages to the economy. Before the damages people have caused to our planet by using these non-renewable sources, scientists are trying to find the best method of producing a renewable source of energy. Solar panels have been a relatively new innovation that consistently get remodelled to maximize the efficiency of the output energy generation. Solar panels use the photoelectric effect to produce a voltage.

Solar panels consist of multiple materials to ensure the deformation under heat is minimized. Models generally consist of an electrode, protective films, and semiconductor photovoltaic materials. The semi conducting material can be the same material or multiple.

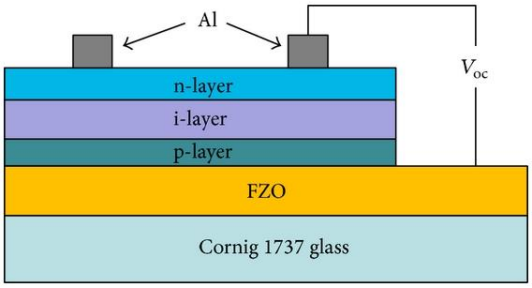


Figure 1

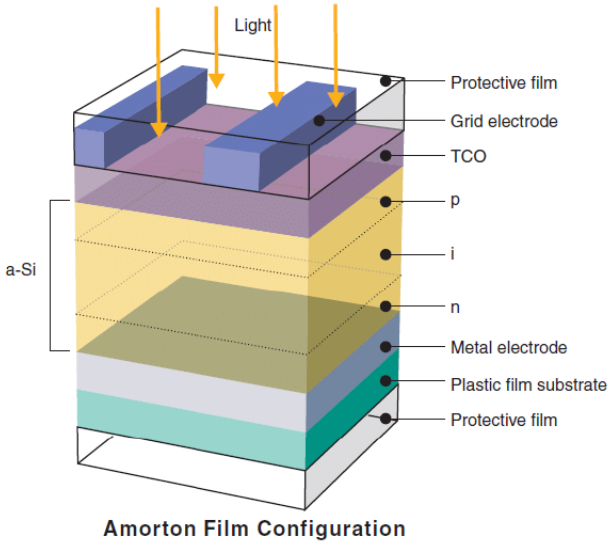
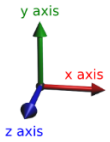


Figure 2



By changing the model of the solar panel, the semiconducting material is the most prominent material that changes the features of the solar panel.

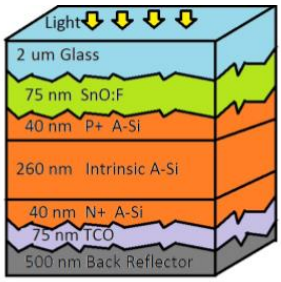


Figure 2

Using the above dimensions (although better fitted for Flex PDE), I will be modeling the deformation of a solar cell under the conditions of the heat produced from the surroundings. For the purposes of this design project, the solar cell will be a bimorph consisting of the semiconduction material, and a transparent conducting oxide (TCO). The TCO will stay constant. The following chart will present all material parameters.

	Young's Modulus $E\ (GPa)$	Poisson's Ratio ν	Coefficient of Thermal Expansion $\alpha\ (\frac{1}{K})$	Density ρ $(\frac{kg}{m^3})$	Specific Heat Capacity $C_p\ (\frac{J}{kgK})$	Thermal Conductivity k $(\frac{W}{mK})$	Ly (μm)
Cadmium telluride (Semi-conductor)	36.52	0.41	5.9×10^{-6}	5850	210	6	34000
Indium Tin Oxide (TCO)	116	0.35	8.4×10^{-6}	6800	178 from $C_p = (\frac{k}{\rho\alpha})$	10.2	7500

Lz is an arbitrary term so the code will function in 2 space and Lx can be set to 50000 μm although the value is not a variable that should affect the data results as this will also be held constant.

To have a changing temperature, the function of temperature dependant on time will be:

$$T(t) = 8t$$

Physics Concepts

This question will incorporate concepts of the heat equation as thermal flux and also thermal expansion properties of materials. The differential equation describing the heat flux is presented as the following formula.

$$\rho C_p \frac{\partial T}{\partial t} = \dot{q}_V + \nabla \cdot (k \nabla T)$$

This is a time dependant differential. On the left side, this shows a time dependant enthalpic equation, mcT and on the ride side, we have defined the heat flow and the gradient of the change in heat. The right side effectively describes the change in heat and shows the flux of the materials. This is because it uses the thermal conductivity and change in temperature to describe this property of change. Since the

temperature is time dependant, we can see the change in heat and this will show the flow of heat will be directed towards the location where the temperature and heat is the lowest.

$$\boldsymbol{\varepsilon_T} = \boldsymbol{\alpha}\Delta T$$

The above formula describes the property of thermal strain for a material. This is dependant on the coefficient of thermal expansion and the change in temperature.

We can then describe the thermal strain and the thermal expansion coefficient as line matrices describing the axial and shear components.

$$\boldsymbol{\varepsilon_T} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{bmatrix}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ \alpha_{yz} \\ \alpha_{xz} \\ \alpha_{xy} \end{bmatrix} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The thermal strain only effects the axial components, so the shear components remain 0.

Since the relationships are time dependant, the temperature change can be described as the following formula.

$$\Delta T = T(t) - T_0$$

Finally, we can use the deformation formulas to describe how the thermal flux will effect the shape of the solar panel bimorph.

$$EI = \iint E(y - \bar{y})^2 dA$$

The flexural rigidity is dependant on the modulus of elasticity and weighted centroid. Integrating this in respect to the surface of focus will output the desired result.

$$\bar{y} = \frac{\iint EydA}{\iint EdA}$$

We can determine the weighted centroid by integrating along the cross section.

$$M = -\frac{\Delta T}{EI}\iint E(y-\bar{y})\alpha dA$$

The moment incorporates the temperature change and the thermal expansion coefficient as this is how the beam deform.

The following formulas will allow us to solve for the tip displacement using the curvature relationships.

$$c \equiv \frac{\partial^2 v}{\partial x^2}$$

$$v_{tip} = \frac{1}{2}cL^2$$

Note that Flex PDE will not need all of this data as it runs and updates, unlike maple, a simulation in respect to time will output the desired data.

Flex Pde solution

To simulate this problem, we can operate in 2 dimensions since the Lz dimensions never change. This allows us to simplify the problem to effectively output data.

```
DEFINITIONS
!Gridplot Magnification Variable
   mag=0.1*globalmax(magnitude(x,y))/globalmax(magnitude(U,V))

!Material Parameters
   k                !Thermal Conductivity
   Cp               !Specific Heat Capacity
   rho              !Density
   Temp0            !Initial Temperature
   E                !Young's Modulus
   nu               !Poisson's Ratio
   alpha            !Coefficient of Thermal Expansion

!Strip Dimentions
   Lx = 0.05        !Length of the strip
   L_Top = 0.0075     !Thickness of the Indium Tin Oxide  section
   L_Bottom = 0.034 !Thickness of the Cadmium Telluride section
```

The first process is to initialize all terms. Then we can add time dependencies and the stiffness and compliance relationships.

```
!Temperature of Surroundings
   TSurroundings = 8*t

!Temperature Difference
   DeltaTemp = Temp - Temp0

!Internal Heat Generation
   qdotvol = 0

!Shear Modulus
   G = E/(2*(1+nu))

!Stiffness Matrix Components
   C11 =E*(1-nu)/(1+nu)/(1-2*nu)
   C22 = C11
   C33 = C11
   C12 = E*nu/(1+nu)/(1-2*nu)
   C13 = C12
   C21 = C12
   C23 = C12
   C31 = C12
   C32 = C12

!Axial Strain Definitions
   EpsilonX = dx(u)
   EpsilonY = dy(v)

!Shear Strain Definitions
   GammaXY = (dx(v) + dy(u))

!Mechanical Strain Definitions
   EpsilonXm = EpsilonX - alpha*DeltaTemp
   EpsilonYm = EpsilonY - alpha*DeltaTemp

!Axial Stress-Strain
   SigmaX = C11*EpsilonXm + C12*EpsilonYm
   SigmaY = C21*EpsilonXm + C22*EpsilonYm

!Shear Stress-Strain
   TauXY = G*GammaXY
```

The next step is to add in the heat equation so that flex PDE can simulate the deformation properly.

```
EQUATIONS
   u: dx(SigmaX)+dy(TauXY) = 0                      !Static Equilibrium in x direction
   v: dx(TauXY)+dy(SigmaY) = 0                      !Static Equilibrium in y direction
   Temp: rho*Cp*dt(Temp) = qdotvol - div(-k*grad(Temp))   !Heat Equation
```

To separate the material properties, in the boundaries, we can initialize individual regions that will have their own unique dependencies.

```
REGION 'Cadmium Telluride'
  E = 36.52e9
  nu = 0.41
  alpha = 5.9e-6
  rho = 5850
  Temp0 = 0
  Cp = 210
  k = 6
start (0,0) !y = 0 surface
  load(u) = 0
  load(v) = 0
line to (Lx,0) !x = Lx surface
  load(u) = 0
  load(v) = 0
  value(Temp) = TSurroundings
line to (Lx,-L_Top) !y = Ly surface
  load(u) = 0
  load(v) = 0
  value(Temp) = 0
line to (0,-L_Top) !x = 0 surface
  value(u) = 0
  value(v) = 0
  value(Temp) = TSurroundings
line to close
```

```
REGION 'Indium Tin Oxide '
  E = 116e9
  nu = 0.35
  alpha = 8.4e-6
  rho = 6800
  Temp0 = 0
  Cp = 178
  k = 10.2
start (0,0) !y = 0 surface:
  load(u) = 0
  load(v) = 0
line to (Lx,0) !x = Lx surface
  load(u) = 0
  load(v) = 0
  value(Temp) = TSurroundings
line to (Lx,L_Bottom) !y = Ly surface
  load(u) = 0
  load(v) = 0
  value(Temp) = TSurroundings
line to (0,L_Bottom) !x = 0 surface
  value(u) = 0
  value(v) = 0
  value(Temp) = TSurroundings
line to close
```

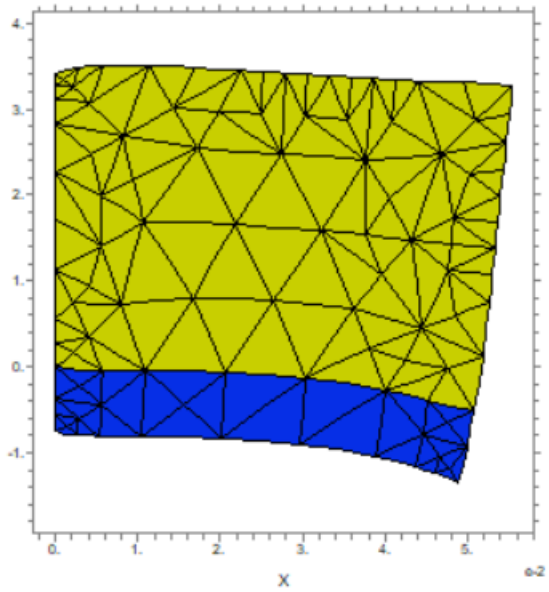
Finally, we can run this simulation for 5s as realistically the temperature surrounding will not rise above 40 degrees Celcius. We can then report the displacement and the average temperature to give us the data we are looking for.

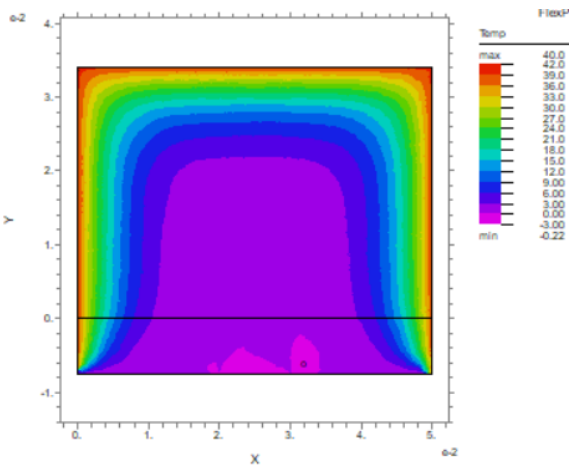
```
SUMMARY
  report val(v,Lx,0) as 'Tip Displacement'

  report (integral (Temp,'Cadmium Telluride') + integral (Temp,'Indium Tin Oxide '))/( integral (1,'Cadmium Telluride')+ integral (1,'Indium Tin Oxide ')) as "Average Temperature"
```

Plots will help visualize how the displacement takes place.

```
PLOTS
  for t = endtime
    grid(x+mag*u, y+mag*v)
    contour (Temp) painted
    history (val(v,Lx,0))
```

The following images provide us how the flux and the deformation occurs in the material given the specific material parameters.





Solar Panel Nyeah Ey

SUMMARY

Tip Displacement= -8.980219e-6
Average Temperature= 10.85309

The tip displacement is very minimal. In an actual solar panel, these materials are the most thermally conductive and even so, the deformation is still very minimal. This is very desirable and makes the solar panel retain it's efficiency. For this model of the solar panel, the semiconducting material was the upper layer and the TCO is the bottom layer. Models can vary depending on desired output. To minimize material used, we will start to optimize the minimum width of the semiconducting material before the deformation is greater than 10e-6m.

Optimization

To optimize for this scenario, we will be changing the semi conducting dimensions. We will output a graph where the tip displacement is the y axis and the x axis is the Ly dimensions. After ensuring that the graph is outputted, we will find the value of Ly that reaches the exact value of 10e-6 m tip displacement.

To make sure that we can output plots and store values properly we input libraries that allow this. We would then input the flex pde code, replacing the value that we want to optimize with %s.

```
import subprocess
import scipy
import matplotlib.pyplot
```

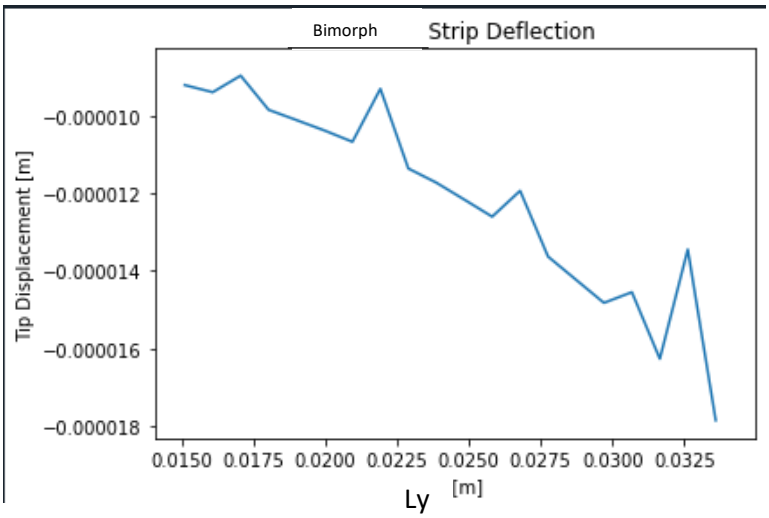
```
!Strip Dimentions
    Lx = 0.05                           !Length of the strip
    L_Top = 0.0075                  !Thickness of the Indium Tin Oxide  section
    L_Bottom = %s               !Thickness of the Cadmium Telluride section
```

We would now lower the Ly of the semi conducting material and then analyze the results.

Then we need to create an ordered list of all values and an if statement that would store these displacement values, then finally plot them in respect to the y distance.

```
Tip_Displacement_List = []
Ly_List = []

Ly_Range = scipy.arange(0.034,0.015,-0.001)

for i in Ly_Range:
    FlexPDE_File_Name = "Bimmorph.pde"
    with open(FlexPDE_File_Name,"w") as f:
        print(FlexPDECode%i,file = f)

    subprocess.run(["C:\FlexPDE6student\FlexPDE6s","-S",FlexPDE_File_Name])

    with open(FlexPDE_File_Name,"r") as f:
        FlexPDE_Data = scipy.loadtxt('Bimmorph.txt',skiprows = 7)

    #t = FlexPDE_Data[:,0]
    v = FlexPDE_Data[:,1]
    #Temp = FlexPDE_Data[:,2]

    if v[-1] < -0.00001 and i < Ly_min:
        Ly_min = i
        Tip_Displacement = v[-1]
    else:
        pass

    print(i) #To show the code is running

    Tip_Displacement_List.append(v[-1])
    Ly_List.append(i)
```

The result of the following code outputs this graph.



From the graph we can see that a Ly of 0.0175-0.02m gives a displacement that ends up reaching a values of 10e-6 m. This is the limit of the dimensions of the semiconducting material.

Conclusion

After doing this design project, I have learned how to optimize for a desired parameter. Although, since many variable were not considered about the solar panel, this is not completely accurate and there are errors brought upon the optimization. A few would be the fact that not all materials were considered and the fact that material parameters are functions of the temperature as well. Having material parameters that are functions of another variable make the simulation much more complex but would output more realistic data points. Solar panels do not undergo much thermal deformation, and this makes is very feasible, even if it were to be put in the hottest areas of our planet. Most solar panels would consist of at least 5-10 materials, again depending on the complexity of the design. The glass is generally made so that the thermal conductivity is very low and even the thickness of this material is very small. The materials used in solar panels are quite complex and will be doped so that it will create certain types of semi conductors such that the photoelectric effect will produce the best amount of energy output. To find the efficiency of the solar panels, another python code could be created that would input the flux at each time, then multiply this heat flow by time to get energy. The surrounding heat would be caused by the sun's irradiation. You could then take the hf of the frequency the surrounding radiation. Then subtract the 2 values and this is all the available energy for the solar panel to gain and using known efficiencies of the panels design, we could optimize for another variable that would output the most amount of energy. Python is an effective tool that can be used to optimize for certain values, especially to ensure that a specific efficiency to produced from the data sets. Undergoing 2CM4 was quite entertaining as there were a lot of new concepts I learned, such as tunneling and piezoelectric. Understanding physics concepts helps perception of the functionalities of surrounding tools and how I could use a computer code to describe these aspects.

Flex Pde

```
TITLE 'Solar Panel Nyeah Ey'

COORDINATES
        cartesian2

SELECT
        ngrid = 5
```

VARIABLES

u
!Displacement in x

v
!Displacement in y

Temp(threshold 1e-6) !Temperature

DEFINITIONS

!Gridplot Magnification Variable

mag=0.1*globalmax(magnitude(x,y))/globalmax(magnitude(U,V))

!Material Parameters

k
!Thermal Conductivity

Cp !Specific Heat Capacity

rho !Density

Temp0 !Initial Temperature

E
!Young's Modulus

nu !Poisson's Ratio

alpha !Coefficient of Thermal Expansion

!Strip Dimentions

Lx = 0.05 !Length of the strip

L_Top = 0.0075 !Thickness of the Indium Tin Oxide section

L_Bottom = 0.034 !Thickness of the Cadmium Telluride section

!Temperature of Surroundings

TSurroundings = 8*t

!Temperature Difference

DeltaTemp = Temp - Temp0

!Internal Heat Generation

qdotvol = 0

!Shear Modulus

G = E/(2*(1+nu))

!Stiffness Matrix Components

C11 =E*(1-nu)/(1+nu)/(1-2*nu)

C22 = C11

C33 = C11

C12 = E*nu/(1+nu)/(1-2*nu)

C13 = C12

C21 = C12

C23 = C12

C31 = C12

C32 = C12

!Axial Strain Definitions

EpsilonX = dx(u)

EpsilonY = dy(v)

!Shear Strain Definitions

GammaXY = (dx(v) + dy(u))

!Mechanical Strain Definitions

```
        EpsilonXm = EpsilonX - alpha*DeltaTemp

        EpsilonYm = EpsilonY - alpha*DeltaTemp


!Axial Stress-Strain

        SigmaX = C11*EpsilonXm + C12*EpsilonYm

        SigmaY = C21*EpsilonXm + C22*EpsilonYm


!Shear Stress-Strain

        TauXY = G*GammaXY


INITIAL VALUES

        Temp = Temp0


EQUATIONS

        u: dx(SigmaX)+dy(TauXY) = 0
                                !Static Equilibrium in x direction

        v: dx(TauXY)+dy(SigmaY) = 0
                                !Static Equilibrium in y direction

        Temp: rho*Cp*dt(Temp) = qdotvol - div(-k*grad(Temp)) !Heat Equation


BOUNDARIES


REGION 'Cadmium Telluride'

        E = 36.52e9

        nu = 0.41

        alpha = 5.9e-6

        rho = 5850

        Temp0 = 0

        Cp = 210
```

```
        k = 6
start (0,0) !y = 0 surface
        load(u) = 0
        load(v) = 0
line to (Lx,0) !x = Lx surface
        load(u) = 0
        load(v) = 0
        value(Temp) = TSurroundings
line to (Lx,-L_Top) !y = Ly surface
        load(u) = 0
        load(v) = 0
        value(Temp) = 0
line to (0,-L_Top) !x = 0 surface
        value(u) = 0
        value(v) = 0
        value(Temp) = TSurroundings
line to close

REGION 'Indium Tin Oxide '

        E = 116e9
        nu = 0.35
        alpha = 8.4e-6
        rho = 6800
        Temp0 = 0
        Cp = 178
        k = 10.2
start (0,0) !y = 0 surface:
        load(u) = 0
```

```
        load(v) = 0
line to (Lx,0) !x = Lx surface
        load(u) = 0
        load(v) = 0
        value(Temp) = TSurroundings
line to (Lx,L_Bottom) !y = Ly surface
        load(u) = 0
        load(v) = 0
        value(Temp) = TSurroundings
line to (0,L_Bottom) !x = 0 surface
        value(u) = 0
        value(v) = 0
        value(Temp) = TSurroundings
line to close

TIME
        0 to 5

PLOTS
        for t = endtime
                grid(x+mag*u, y+mag*v)
                contour (Temp) painted
                history (val(v,Lx,0))

SUMMARY
        report val(v,Lx,0) as 'Tip Displacement'

        report (integral (Temp,'Cadmium Telluride') + integral (Temp,'Indium Tin Oxide '))/( integral
(1,'Cadmium Telluride')+ integral (1,'Indium Tin Oxide ')) as ''Average Temperature''
```

```
END
```

Python

```
import subprocess

import scipy

import matplotlib.pyplot


FlexPDECode = """


TITLE 'Solar Panel Nyeah Ey'


COORDINATES

        cartesian2


SELECT

        ngrid = 5


VARIABLES

        u
        !Displacement in x

        v
        !Displacement in y

        Temp(threshold 1e-6)    !Temperature


DEFINITIONS

!Gridplot Magnification Variable

        mag=0.1*globalmax(magnitude(x,y))/globalmax(magnitude(U,V))
```

```
!Material Parameters

        k
        !Thermal Conductivity

        Cp                                                      !Specific Heat
Capacity

        rho                                                     !Density

        Temp0                                           !Initial Temperature

        E
        !Young's Modulus

        nu                                                      !Poisson's Ratio

        alpha                                           !Coefficient of Thermal
Expansion


!Strip Dimentions

        Lx = 0.05                                               !Length of the strip

        L_Top = 0.0075                                  !Thickness of the Indium Tin Oxide  section

        L_Bottom = %s                   !Thickness of the Cadmium Telluride section


!Temperature of Surroundings

        TSurroundings = 8*t


!Temperature Difference

        DeltaTemp = Temp - Temp0


!Internal Heat Generation

        qdotvol = 0


!Shear Modulus

        G = E/(2*(1+nu))
```

```
!Stiffness Matrix Components
        C11 =E*(1-nu)/(1+nu)/(1-2*nu)
        C22 = C11
        C33 = C11
        C12 = E*nu/(1+nu)/(1-2*nu)
        C13 = C12
        C21 = C12
        C23 = C12
        C31 = C12
        C32 = C12

!Axial Strain Definitions
        EpsilonX = dx(u)
        EpsilonY = dy(v)

!Shear Strain Definitions
        GammaXY = (dx(v) + dy(u))

!Mechanical Strain Definitions
        EpsilonXm = EpsilonX - alpha*DeltaTemp
        EpsilonYm = EpsilonY - alpha*DeltaTemp

!Axial Stress-Strain
        SigmaX = C11*EpsilonXm + C12*EpsilonYm
        SigmaY = C21*EpsilonXm + C22*EpsilonYm

!Shear Stress-Strain
        TauXY = G*GammaXY
```

```
INITIAL VALUES
        Temp = Temp0

EQUATIONS
        u: dx(SigmaX)+dy(TauXY) = 0
                                !Static Equilibrium in x direction
        v: dx(TauXY)+dy(SigmaY) = 0
                                !Static Equilibrium in y direction
        Temp: rho*Cp*dt(Temp) = qdotvol - div(-k*grad(Temp)) !Heat Equation

BOUNDARIES

REGION 'Cadmium Telluride'
        E = 36.52e9
        nu = 0.41
        alpha = 5.9e-6
        rho = 5850
        Temp0 = 0
        Cp = 210
        k = 6
start (0,0) !y = 0 surface
        load(u) = 0
        load(v) = 0
line to (Lx,0) !x = Lx surface
        load(u) = 0
        load(v) = 0
        value(Temp) = TSurroundings
line to (Lx,-L_Top) !y = Ly surface
```

load(u) = 0

load(v) = 0

value(Temp) = 0

line to (0,-L_Top) !x = 0 surface

value(u) = 0

value(v) = 0

value(Temp) = TSurroundings

line to close

REGION 'Indium Tin Oxide '

E = 116e9

nu = 0.35

alpha = 8.4e-6

rho = 6800

Temp0 = 0

Cp = 178

k = 10.2

start (0,0) !y = 0 surface:

load(u) = 0

load(v) = 0

line to (Lx,0) !x = Lx surface

load(u) = 0

load(v) = 0

value(Temp) = TSurroundings

line to (Lx,L_Bottom) !y = Ly surface

load(u) = 0

load(v) = 0

value(Temp) = TSurroundings

```
line to (0,L_Bottom) !x = 0 surface
        value(u) = 0
        value(v) = 0
        value(Temp) = TSurroundings
line to close

TIME
        0 to 5

PLOTS
        for t = endtime

    history(v,Temp) at (Lx,0) PrintOnly Export Format '#t#b#1#b#2' file = 'Bimmorph.txt'

END
"""

Tip_Displacement = -100
Ly_min = 100

Tip_Displacement_List = []
Ly_List = []

Ly_Range = scipy.arange(0.034,0.015,-0.001)

for i in Ly_Range:
   FlexPDE_File_Name = "Bimmorph.pde"
   with open(FlexPDE_File_Name,"w") as f:
      print(FlexPDECode%i,file = f)
```

```
    subprocess.run(["C:\FlexPDE6student\FlexPDE6s","-S",FlexPDE_File_Name])

    with open(FlexPDE_File_Name,"r") as f:
        FlexPDE_Data = scipy.loadtxt('Bimmorph.txt',skiprows = 7)

    #t = FlexPDE_Data[:,0]
    v = FlexPDE_Data[:,1]
    #Temp = FlexPDE_Data[:,2]

    if v[-1] < -0.00001 and i < Ly_min:
        Ly_min = i
        Tip_Displacement = v[-1]
    else:
        pass

    print(i) #To show the code is running

    Tip_Displacement_List.append(v[-1])
    Ly_List.append(i)

matplotlib.pyplot.plot(Ly_List,Tip_Displacement_List)
matplotlib.pyplot.xlabel('Ly [m]')
matplotlib.pyplot.ylabel('Tip Displacement [m]')
matplotlib.pyplot.title('BiMorph Strip Deflection')
matplotlib.pyplot.show

print('Ly Min =', Ly_min)
print('Tip Displacement = ', Tip_Displacement)
```

Bibliography

Crystran, “Cadmium Telluride (CdTe),” *Cadmium Telluride Optical Material*. [Online]. Available: https://www.crystran.co.uk/optical-materials/cadmium-telluride-cdte. [Accessed: 16-Apr-2021].

J. Deng, L. Zhang, L. Hui, X. Jin, and B. Ma, “Indium Tin Oxide Thin-Film Thermocouple Probe Based on Sapphire Microrod,” *Sensors (Basel, Switzerland)*, 27-Feb-2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7085781/. [Accessed: 16-Apr-2021].

Indium Tin Oxide. [Online]. Available: http://www.mit.edu/~6.777/matprops/ito.htm. [Accessed: 16-Apr-2021].