

# Computational physics Final project

Fin analysis using finite element method

정준상, 김영훈

# Presentation context

1. Basic of FEM with fin analysis
2. FEM with solid conduction heat transfer
3. FEM with given fin mesh
4. Result in 2D & 3D
5. Analysis

# 1. Basic of FEM with fin analysis

a. What is Finite Element Method?

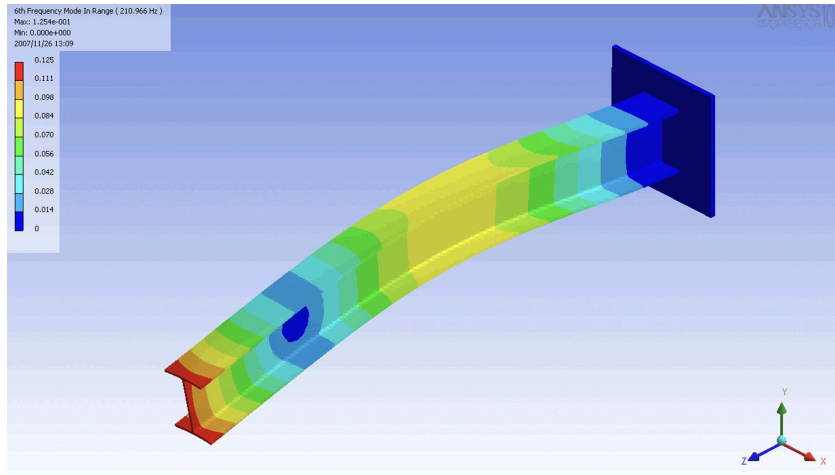
b. What is fin?

c. How can we do this?

# 1. Basic of FEM with fin analysis -- (a)

What is Finite Element Method?

- FEM is an analytic method to find solution by calculating mesh.

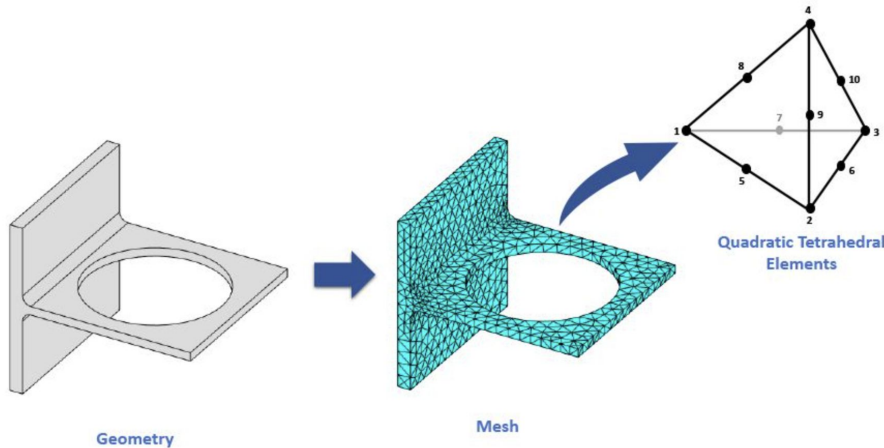


For example, it can be used analyzing critical stress or temperature of material, deflection of beam.

# 1. Basic of FEM with fin analysis -- (a)

What is Finite Element Method?

- FEM is an analytic method to find solution by calculating mesh.



**We slice given geometry to “element”.**

**We call the set of every elements, “mesh”.**

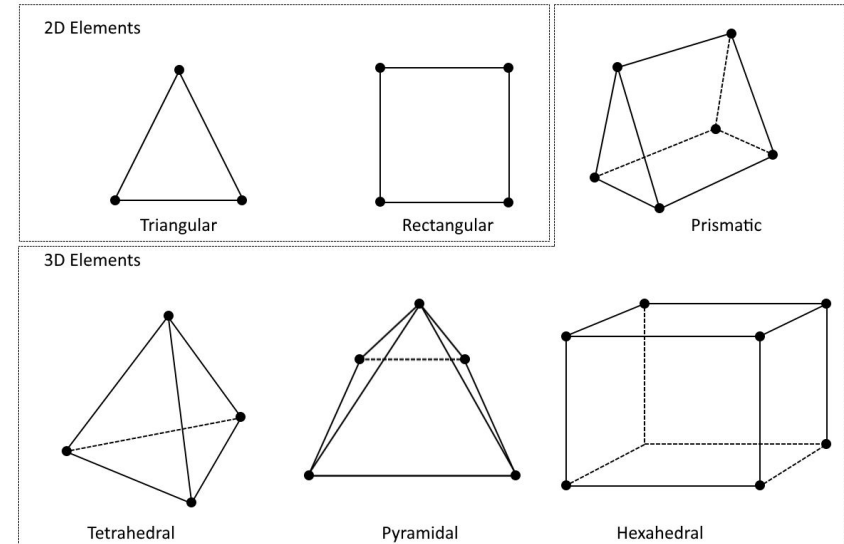
# 1. Basic of FEM with fin analysis -- (a)

What is Finite Element Method?

- FEM is an analytic method to find solution by calculating mesh.

**There could be multiple ways to generating mesh.**

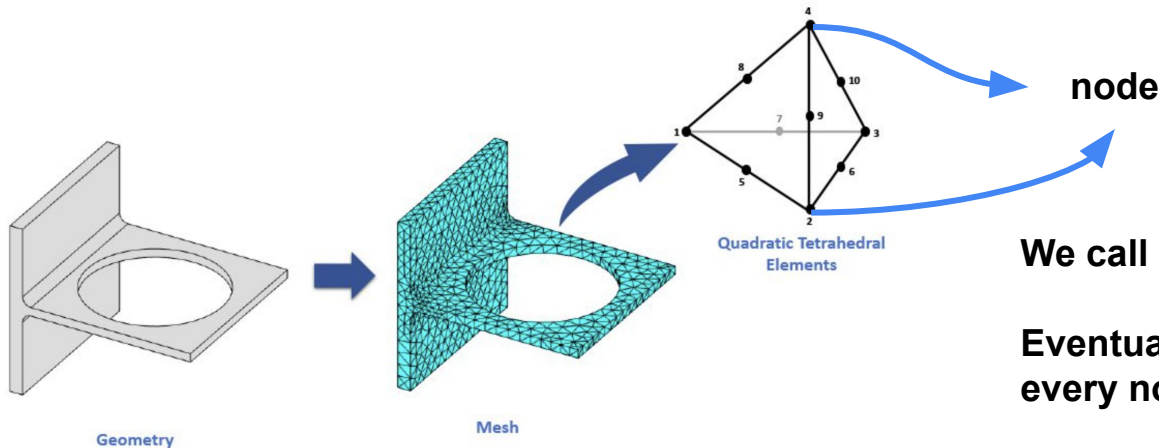
**The geometry of mesh may affect the result.**



# 1. Basic of FEM with fin analysis -- (a)

What is Finite Element Method?

- FEM is an analytic method to find solution by calculating mesh.



**We call every vertex of mesh “node”.**

**Eventually, FEM is a way that calculates every nodes via govern equation.**

# 1. Basic of FEM with fin analysis -- (b)

What is fin?



Fin is a surface that extend from an hot object to increase the heat transfer rate.

The amount of conduction, convection and radiation will affect the heat transfer.

In short, if the fin can transfer heat faster, the CPU can cooler.



# 1. Basic of FEM with fin analysis -- (c)

How can we do this?

- There are several important coefficient about transient fin heat transfer.

1. Biot number ( $Bi$ ) 
$$Bi = \frac{hL_c}{k}$$

2. Fourier number ( $Fo$ ) 
$$Fo = \frac{\alpha \Delta t}{dx * dy}$$

# 1. Basic of FEM with fin analysis -- (c)

How can we do this?

- Biot number ( $Bi$ )  
$$Bi = \frac{hL_c}{k}$$

$L_c$  : characteristic length  
 $h$  : convection heat transfer coefficient  
 $k$  : conduction heat transfer coefficient
- Biot number is a coefficient that shows contribution of convection versus conduction in transient heat transfer.
- If Biot number  $\gg 1$ , convection will be the dominant factor that contributes heat transfer
- If Biot number  $\ll 1$ , conduction will be the dominant factor that contributes heat transfer

# 1. Basic of FEM with fin analysis -- (c)

How can we do this?

- Biot number ( $Bi$ )  
$$Bi = \frac{hL_c}{k}$$

$L_c$  : characteristic length  
 $h$  : convection heat transfer coefficient  
 $k$  : conduction heat transfer coefficient
- characteristic length is approximated length that describes “outer region VS inner region”

$$L_c \approx \frac{\text{Surface}}{\text{Volume}} \text{ or } \frac{\text{Circumference}}{\text{Surface}}$$

# 1. Basic of FEM with fin analysis -- (c)

How can we do this?

- Fourier number ( $Fo$ )  $Fo = \frac{\alpha \Delta t}{dx * dy}$ 
  - $\alpha$  : thermal diffusivity
  - $\Delta t$  : unit time step
  - $dx$  and  $dy$  : unit length of mesh
- Fourier number is a coefficient that describes “how much heat will travel during the unit time step”.

# 1. Basic of FEM with fin analysis -- (c)

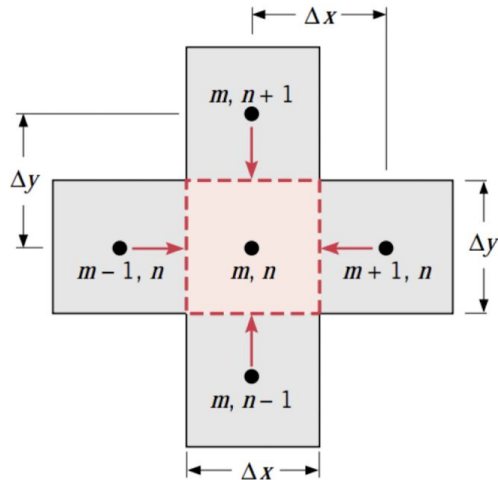
How can we do this?

- Fourier number ( $Fo$ )  $Fo = \frac{\alpha \Delta t}{dx * dy}$ 
  - $\alpha$  : thermal diffusivity
  - $\Delta t$  : unit time step
  - $dx$  and  $dy$  : unit length of mesh
- Larger the unit time step becomes, unit mesh length has to be larger too.
- If we set smaller unit mesh, FEM will collapse, since heat will pass through current mesh element and affect the others.

# 1. Basic of FEM with fin analysis -- (c)

How can we do this?

- FEM is based on energy equilibrium.  $\dot{E}_{in} + \dot{E}_{generate} = \dot{E}_{out}$



(Heat flows to  $(m,n)$ ) + (Heat that generated by  $(m,n)$ )  
= (Heat that  $(m,n)$  emit)

Assume there's no heat generated by material, ( $q = 0$ )  
and apply this condition to Heat diffusion equation

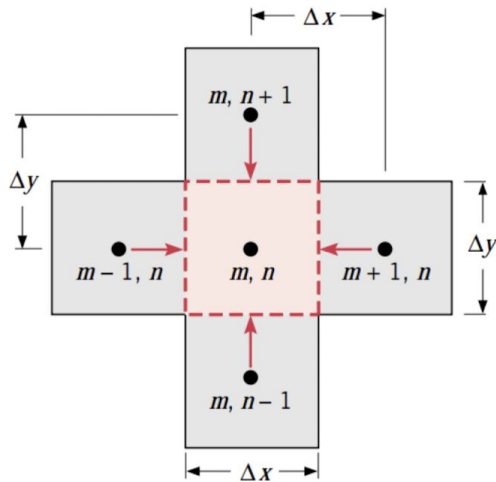
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{q}}{k} = \frac{1}{\alpha} \frac{\partial T}{\partial t}$$

We can use finite difference approximation to  
calculate second derivative.

# 1. Basic of FEM with fin analysis -- (c)

How can we do this?

- Eventually, we can describe the node like below



$$T_{m,n}^{p+1} = Fo(T_{m+1,n}^p + T_{m-1,n}^p + T_{m,n+1}^p + T_{m,n-1}^p) + (1 - 4Fo)T_{m,n}^p$$

$T_{m,n}^p$  : temperature of node (m,n) in time (p)

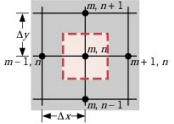
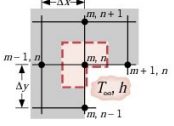
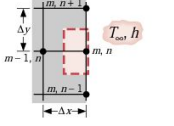
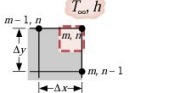
**Note that in only applies 2-dimensional, closed mesh  
(4 nodes are adjacent with node (m,n))**

# 1. Basic of FEM with fin analysis -- (c)

## - Other mesh case can be calculated like this

Bergman, T. L., and Frank P. Incropera. *Fundamentals of Heat and Mass Transfer*. Seventh edition. Wiley, 2011.

**TABLE 5.3** Transient, two-dimensional finite-difference equations ( $\Delta x = \Delta y$ )

Configuration	(a) Explicit Method		(b) Implicit Method
	Finite-Difference Equation	Stability Criterion	
 <p>1. Interior node</p>	$T_{m,n}^{p+1} = Fo(T_{m+1,n}^p + T_{m-1,n}^p + T_{m,n+1}^p + T_{m,n-1}^p) + (1 - 4Fo)T_{m,n}^p \quad (5.76)$	$Fo \leq \frac{1}{4} \quad (5.80)$	$(1 + 4Fo)T_{m,n}^{p+1} - Fo(T_{m+1,n}^{p+1} + T_{m-1,n}^{p+1} + T_{m,n+1}^{p+1} + T_{m,n-1}^{p+1}) = T_{m,n}^p \quad (5.92)$
 <p>2. Node at interior corner with convection</p>	$T_{m,n}^{p+1} = \frac{2}{3}Fo(T_{m+1,n}^p + 2T_{m-1,n}^p + 2T_{m,n+1}^p + T_{m,n-1}^p + 2BiT_{\infty}) + (1 - 4Fo - \frac{4}{3}BiFo)T_{m,n}^p \quad (5.85)$	$Fo(3 + Bi) \leq \frac{3}{4} \quad (5.86)$	$(1 + 4Fo(1 + \frac{1}{3}Bi))T_{m,n}^{p+1} - \frac{2}{3}Fo \cdot (T_{m+1,n}^{p+1} + 2T_{m-1,n}^{p+1} + 2T_{m,n+1}^{p+1} + T_{m,n-1}^{p+1}) = T_{m,n}^p + \frac{4}{3}BiFoT_{\infty} \quad (5.95)$
 <p>3. Node at plane surface with convection<sup>a</sup></p>	$T_{m,n}^{p+1} = Fo(2T_{m-1,n}^p + T_{m,n+1}^p + T_{m,n-1}^p + 2BiT_{\infty}) + (1 - 4Fo - 2BiFo)T_{m,n}^p \quad (5.87)$	$Fo(2 + Bi) \leq \frac{1}{2} \quad (5.88)$	$(1 + 2Fo(2 + Bi))T_{m,n}^{p+1} - Fo(2T_{m-1,n}^{p+1} + T_{m,n+1}^{p+1} + T_{m,n-1}^{p+1}) = T_{m,n}^p + 2BiFoT_{\infty} \quad (5.96)$
 <p>4. Node at exterior corner with convection</p>	$T_{m,n}^{p+1} = 2Fo(T_{m-1,n}^p + T_{m,n+1}^p + 2BiT_{\infty}) + (1 - 4Fo - 4BiFo)T_{m,n}^p \quad (5.89)$	$Fo(1 + Bi) \leq \frac{1}{4} \quad (5.90)$	$(1 + 4Fo(1 + Bi))T_{m,n}^{p+1} - 2Fo(T_{m-1,n}^{p+1} + T_{m,n+1}^{p+1}) = T_{m,n}^p + 4BiFoT_{\infty} \quad (5.97)$

<sup>a</sup>To obtain the finite-difference equation and/or stability criterion for an adiabatic surface (or surface of symmetry), simply set  $Bi$  equal to zero.

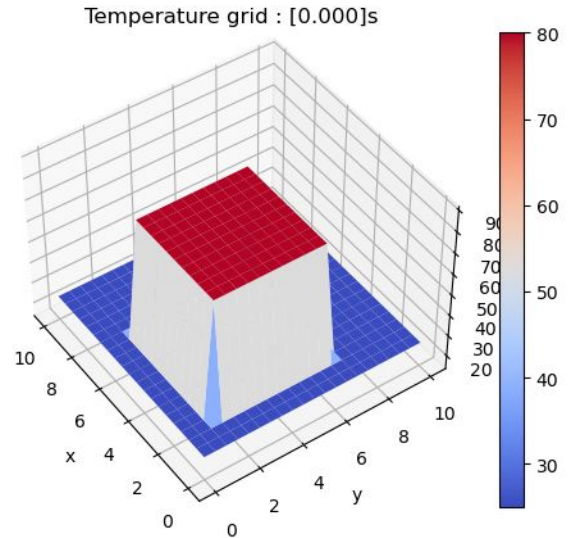


## 2. FEM with solid conduction heat transfer

We made first model using numpy, matplotlib, and OpenCV to export animation

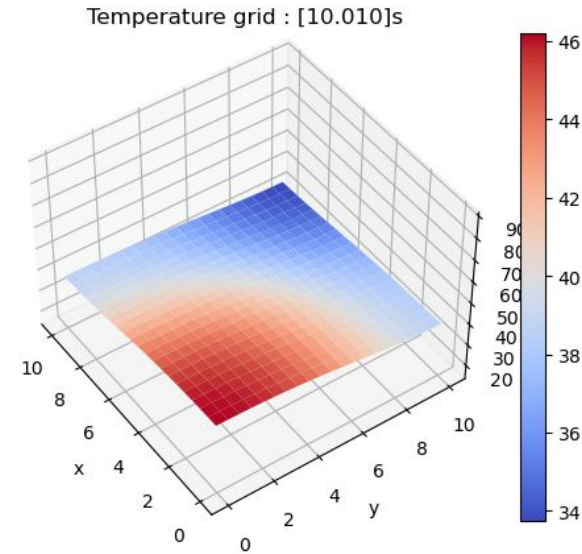
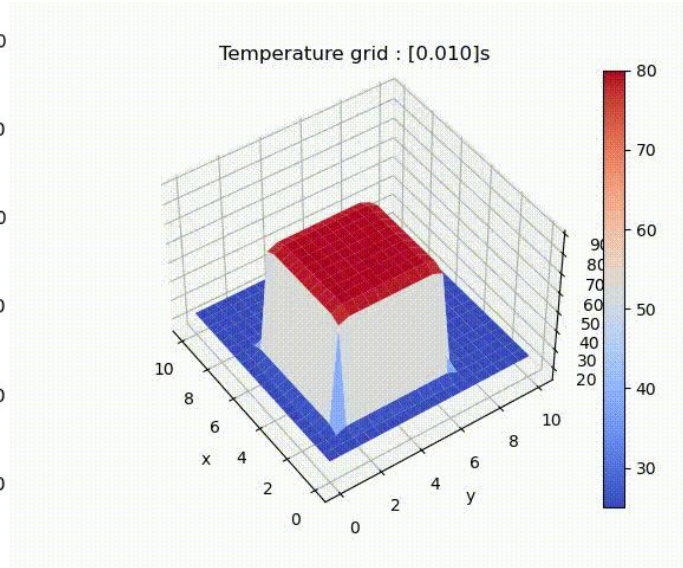
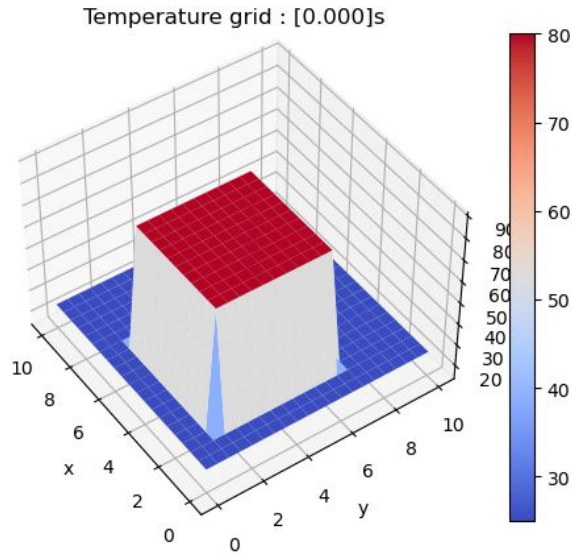
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import cv2
4 from matplotlib.cm import coolwarm
5 from matplotlib.backends.backend_agg import FigureCanvasAgg
6
7 class FEM_2d :
8 > def __init__(self, alpha, X, Y, T, dx, dy) :--
29
30 > def return_index(self, x, y) : --
38
39 > def set_ic_rect(self, T, pt1 : list, pt2 : list) :          # pt1[x0, y0] ~ pt2[x1, y1]--
57
58 > def set_bc(self, T, X_axis : list=None, Y_axis : list=None) : --
91
92 > def return_plot(self, elev=45, azim=145) : --
107
108 > def plot_status(self, elev=45, azim=145) : --
123
124 > def print_status(self) : --
163
164 > def build(self, dt) : --
178
179 > def compute(self, t_end, verbose=False, save_gif=False, elev=45, azim=145) : --
277
278 > def save_animation(self, t_end) : --
✓ 2.8s
```

Python



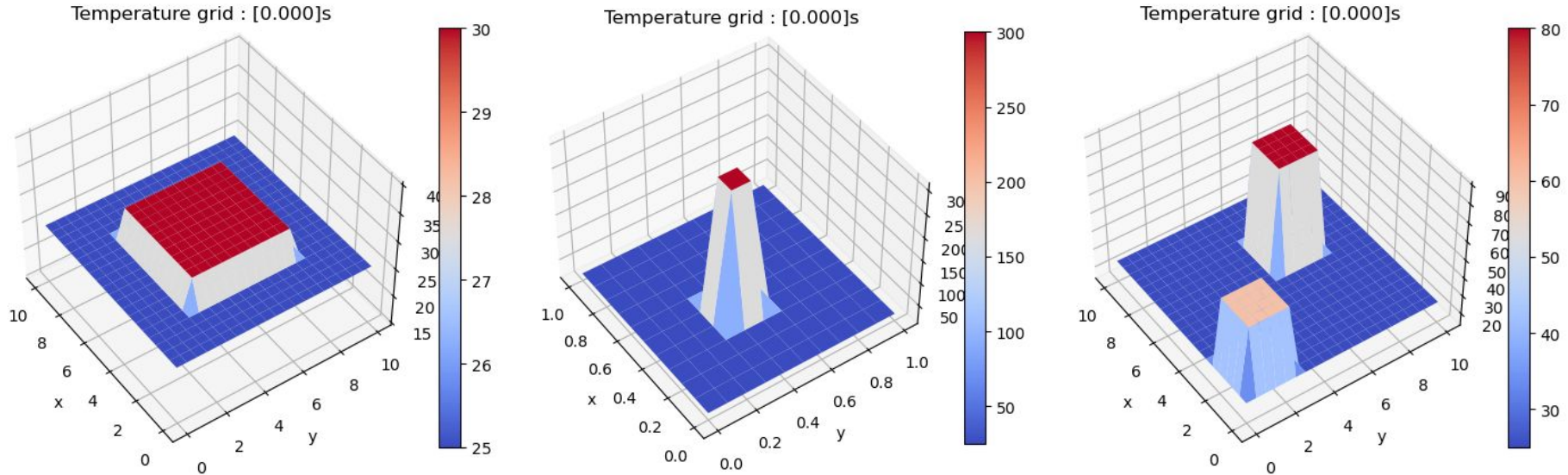
## 2. FEM with solid conduction heat transfer

It's a model for rectangular fin, only affected by conduction and boundary condition.



## 2. FEM with solid conduction heat transfer

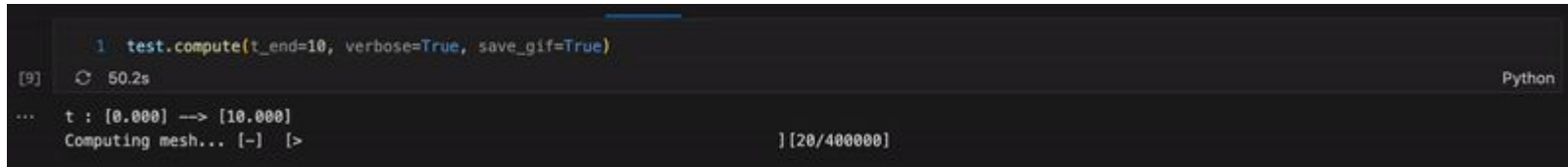
You can adjust mesh size, initial conditions and boundary conditions



## 2. FEM with solid conduction heat transfer

There are disadvantage with this model

1. There's too many meshes to calculate.
2. Using OpenCV to animate process wasn't good idea.
3. There's no utils for computing mesh until steady-state is reached.



```
1 test.compute(t_end=10, verbose=True, save_gif=True)
[9] 50.2s Python
... t : [0.000] --> [10.000]
Computing mesh... [-] [>] [20/400000]
```

The screenshot shows a Jupyter Notebook interface with a dark theme. A code cell is selected, showing a single line of Python code: `test.compute(t_end=10, verbose=True, save_gif=True)`. Below the code, the output is displayed, indicating the execution time as 50.2 seconds. The output also shows a progress bar for the mesh computation, with the current time at 0.000 and the target time at 10.000. The progress bar is currently at 20 out of 400,000 steps.

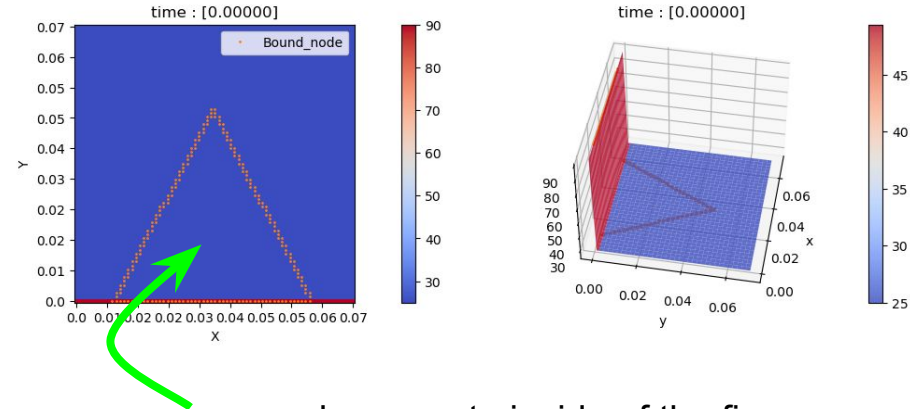
### 3. FEM with given fin mesh

We've made a second model that decrease computation, and uses FuncAnimation of matplotlib to animate process.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import time, sys
4
5 from numpy import pi, sin, cos
6 from math import sqrt
7 from functools import partial
8 from matplotlib.cm import coolwarm
9 from matplotlib.animation import FuncAnimation
10
11 > class Bound_node : ...
13
14 > class Mesh : ...
246
247 > class FEM_2D() : ...
952
```

[1]

To decrease computation, we only compute the mesh inside the fin.

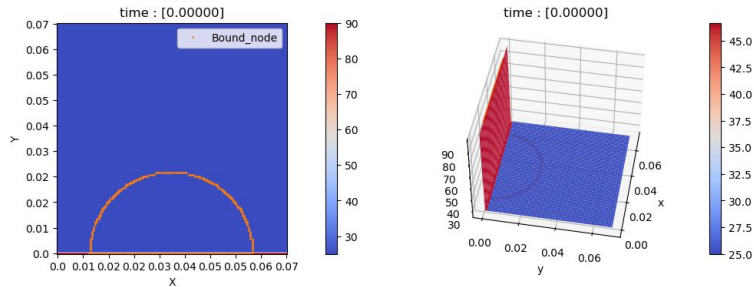


we only compute inside of the fin

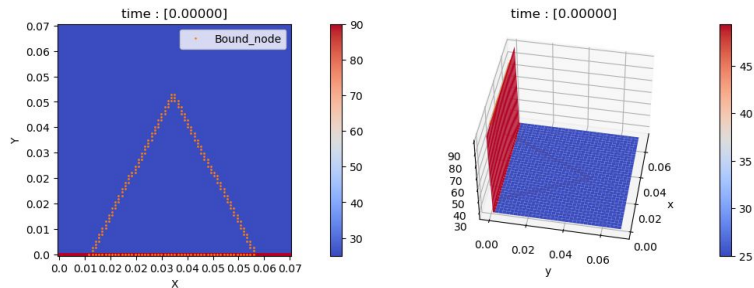
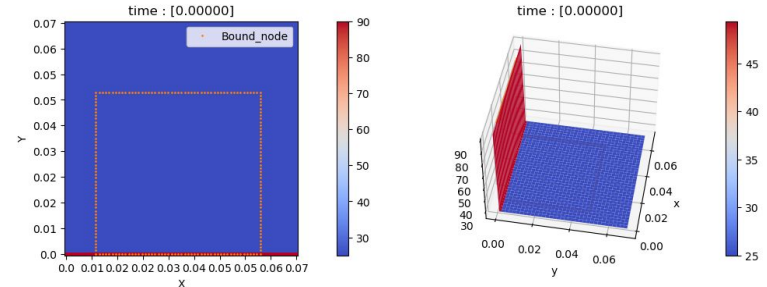
### 3. FEM with given fin mesh

There are three geometry of second model

Half-circle mesh



Rectangular mesh

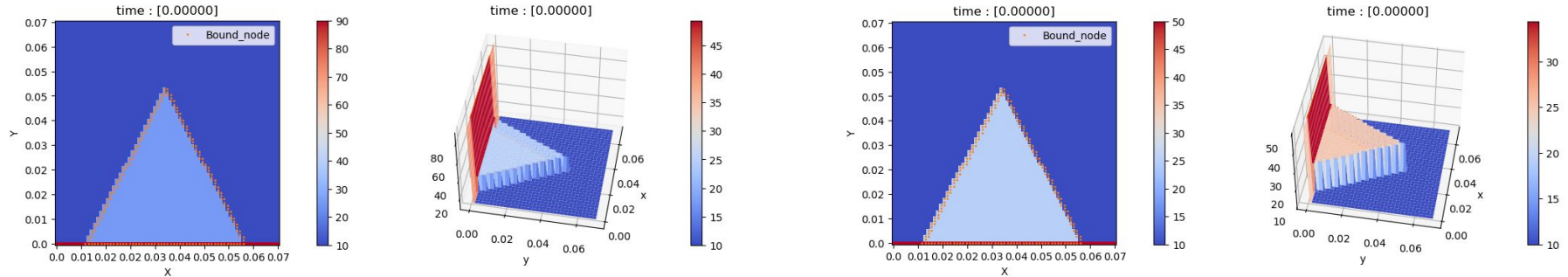


Triangular mesh

### 3. FEM with given fin mesh

Model can adjust boundary condition, initial condition, and mesh size.

This model includes the convection heat transfer via surrounding air with given coefficient and temperature



You can see both the air and boundary temperature are different

### 3. FEM with given fin mesh

We made a utils that compute mesh until steady-state.

If the model reaches steady-state in max iteration, it will shows the precision of convergence and average temperature. Otherwise, it will stop

```
Estimate computing time

> 1 dt = 0.0025
  2 t_end = dt * 500
  3 Triangle_fin.is_good(dt, t_end)

[]

1 Triangle_fin.compute_steady_state(dt=0.0025, max_iter=1.0e+4)

[]

1
```

```
> 1 Triangle_fin.compute_steady_state(dt=0.0025, max_iter=1.0e+4)

[]

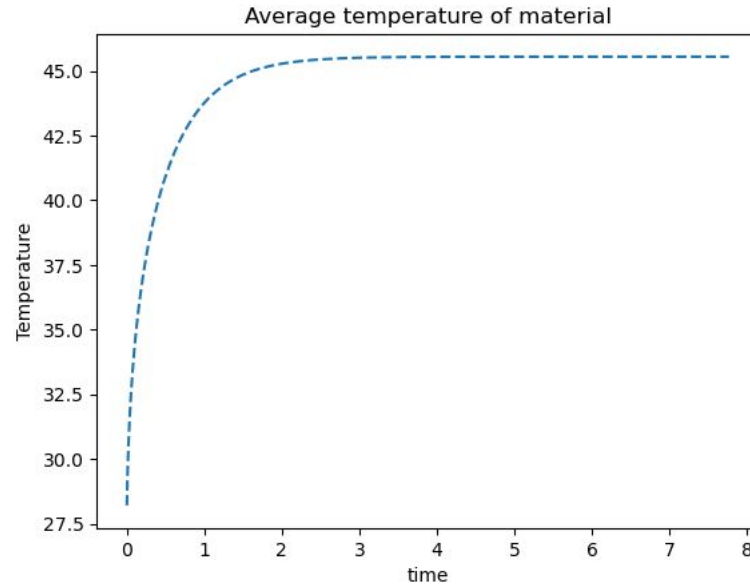
1
```



### 3. FEM with given fin mesh

You can watch the average temperature difference via time.

```
> ~  
1 Triangle_fin.compute_steady_state(dt=0.0025, max_iter=1.0e4)  
2 Triangle_fin.plot_process("--")  
[ ]  
  
1  
[ ]
```

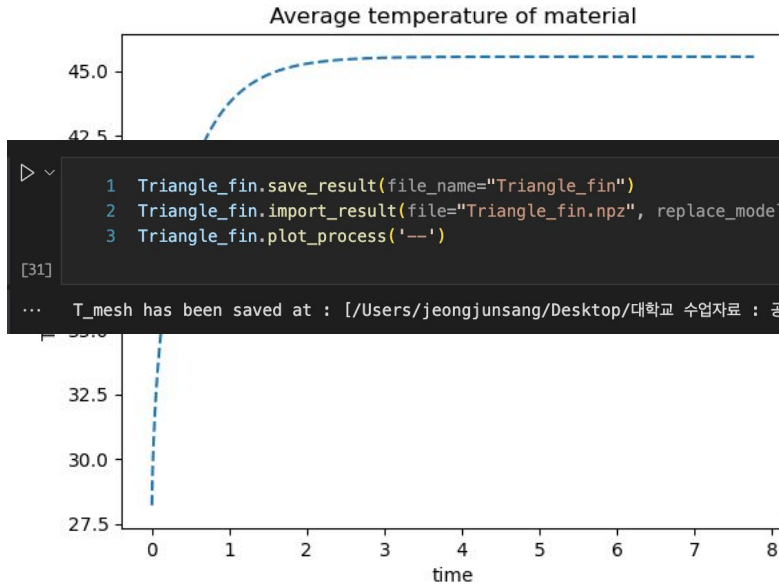


And you can save or import computing result.

### 3. FEM with given fin mesh

You can watch the average temperature difference via time.

```
> ~  
1 Triangle_fin.compute_steady_state(dt=0.0025, max_iter=1.0e4)  
2 Triangle_fin.plot_process('--')  
[ ]  
  
1  
[ ]
```

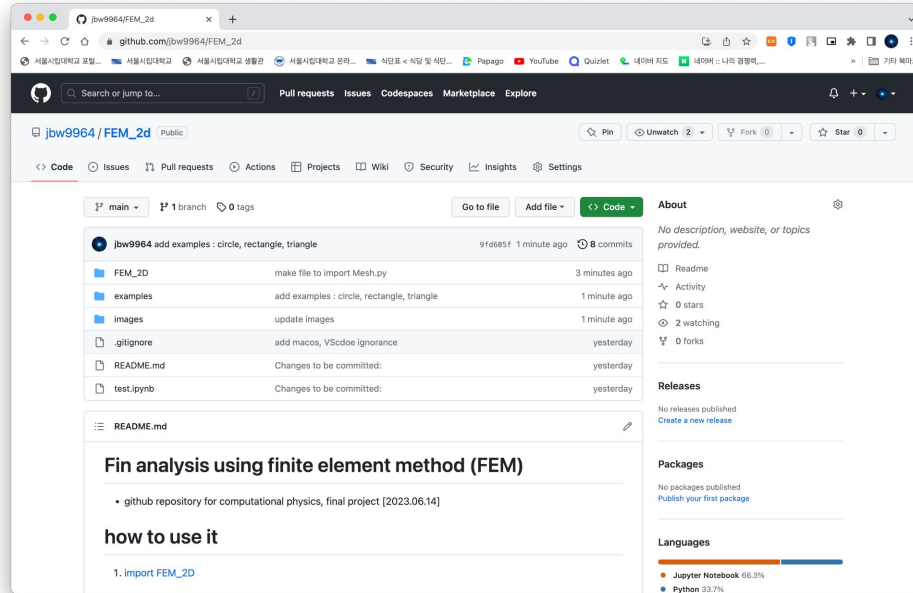


And you can save or import computing result.

```
> ~  
1 Triangle_fin.save_result(file_name="Triangle_fin")  
2 Triangle_fin.import_result(file="Triangle_fin.npz", replace_model=True)  
3 Triangle_fin.plot_process('--')  
[31]  
... T_mesh has been saved at : [/Users/jeongjunsang/Desktop/대학교 수업자료 : 공부/4-1/기타/전산물리/기말 프로젝트/test/Triangle_fin.npz]
```

### 3. FEM with given fin mesh

More details at github repository.



[https://github.com/jbw9964/FEM\\_2d.git](https://github.com/jbw9964/FEM_2d.git)

## 4. Result 2D & 3D

Shapes?

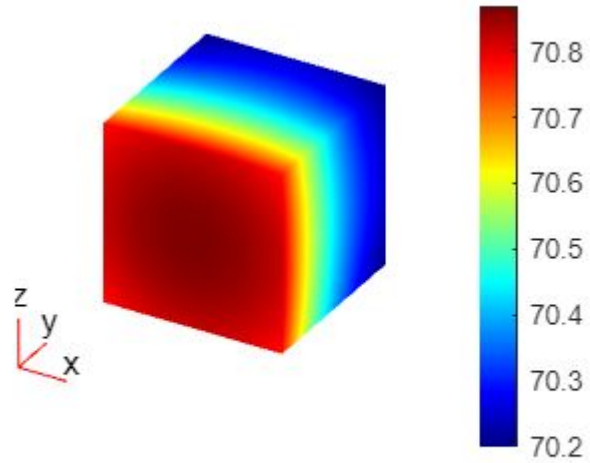
## 4-1. 2D results





## 4-2. 3D results

steady state

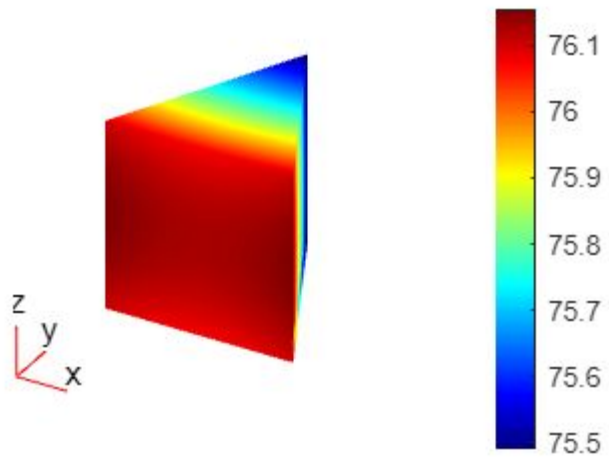


Average Temperature: 70.4851

transient video



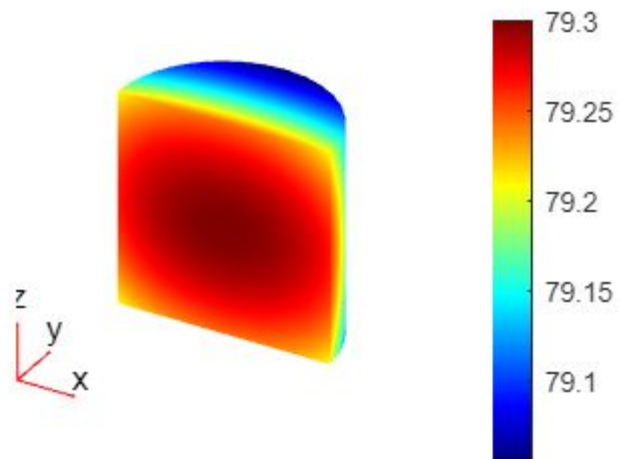
steady state



Average Temperature: 75.8988

transient video

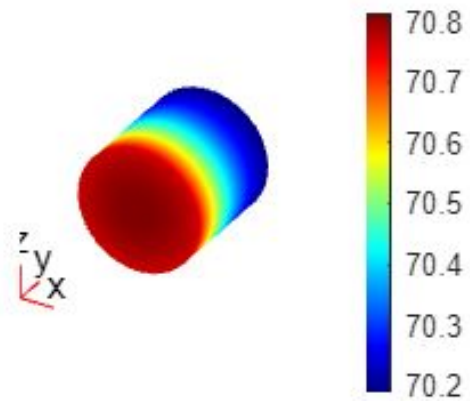
steady state



Average Temperature: 79.1791

transient video

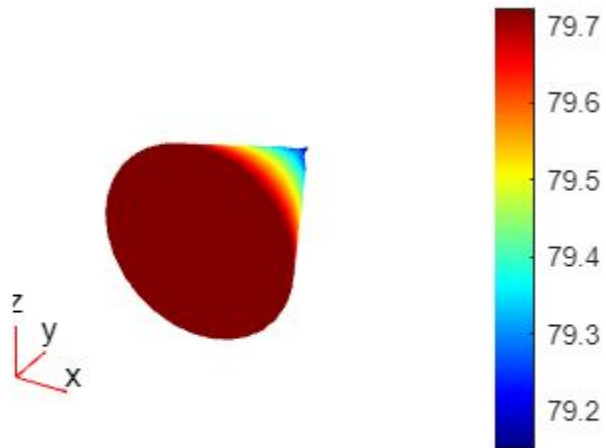
steady state



Average Temperature: 70.437

transient video

steady state



Average Temperature: 79.5971

transient video

transient video



## 5. Analysis

heat sink재료: 알루미늄 합금.. 접촉부분 Cu, 나머지(fin부분 등) Al사용 경우 多

