[LPL2] Thermal CAE Modify for T_J

2017-09-20, 김동호

Model

Software : Salome Platform, Python Lang
Auto Geometry Generating Script Program : Heatsink4LED.py

https://github.com/dymaxionkim/ElmerFEM_Examples/tree/master/20170920_Salome_Script_Heatsink4LED

- LED PACKAGE MODEL

SLUG_X = 0.005
SLUG_Y = 0.005
SLUG_Z = 0.0004

LED_X = 0.0005
LED_Y = 0.0015
LED_Z = 0.000015

QTY_X = 6
QTY_Y = 20

PITCH_X = 0.014
PITCH_Y = 0.0128

- PCB & TIM MODEL

MARGIN_X = 0.0075
MARGIN_Y = 0.0075
PCB_X = (QTY_X-1)*PITCH_X+2*MARGIN_X
PCB_Y = (QTY_Y-1)*PITCH_Y+2*MARGIN_Y
PCB_Z = 0.0016

TIM_X = PCB_X
TIM_Y = PCB_Y
TIM_Z = 0.0005

- HEATSINK MODEL

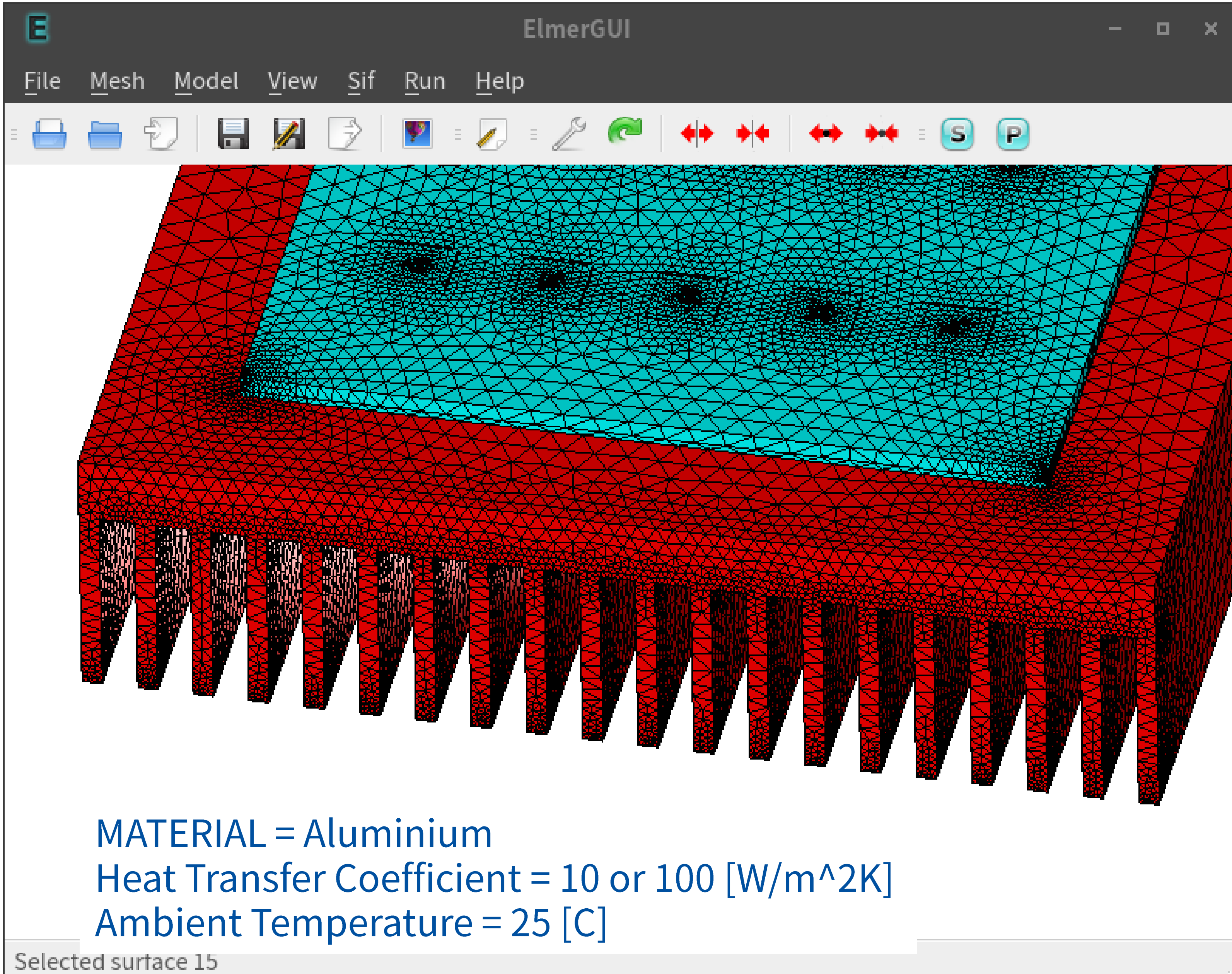
MARGIN_BASE_X = 0.009
MARGIN_BASE_Y = 0.02
BASE_X = PCB_X+2*MARGIN_BASE_X
BASE_Y = PCB_Y+2*MARGIN_BASE_Y
BASE_Z = 0.004

QTY_FIN = 8
FIN_X = 0.006
FIN_Y = BASE_Y
FIN_Z = 0.008

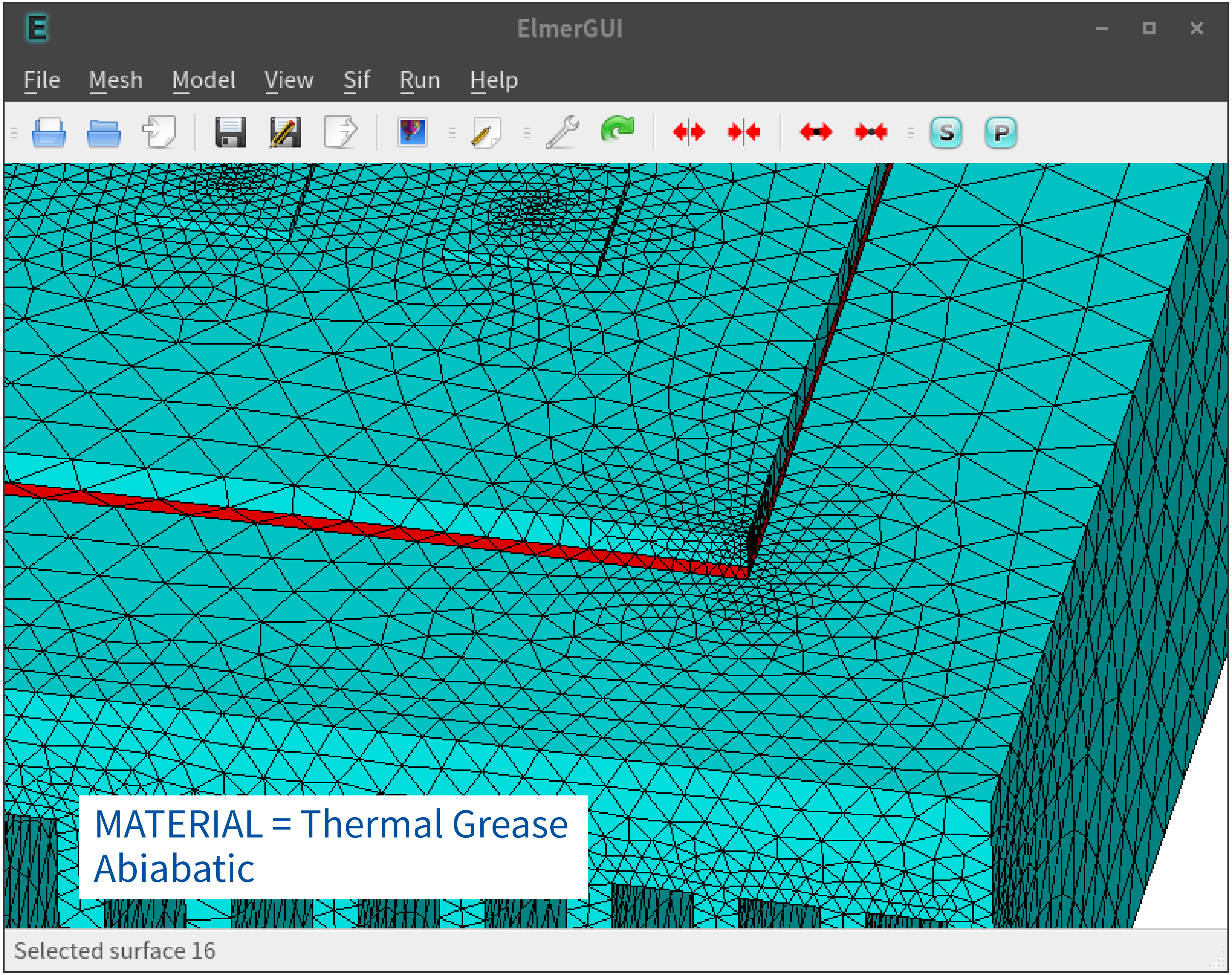
- NETGEN Parameters

```
MinMeshSize = 0.0000001 # specify in m
MaxMeshSize = 0.003 # specify in m
#MeshSecondOrder = float(raw_input("SetSecondOrder[0,1] : "))
MeshSecondOrder = 1
#SetFiness ::: 0=VeryCoarse, 1=Coarse, 2=Moderate, 3=Fine,
4=VeryFine, 5=Custom“
MeshFineness = 5
if MeshFineness==5:
        MeshSegPerEdge = 5
        MeshSegPerRadius = 5
        MeshGrowthRate = 0.1
```

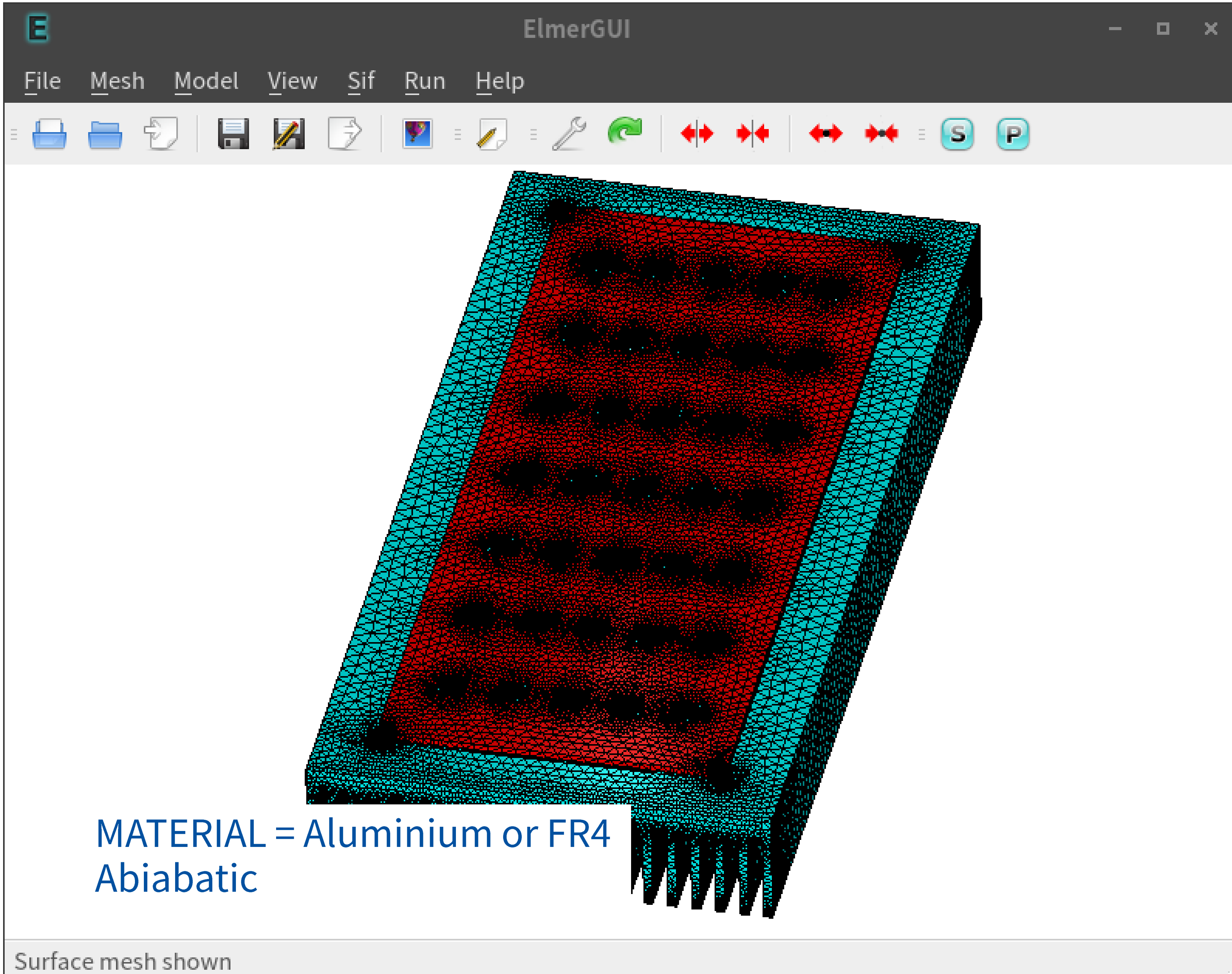
HEATSINK Boundary



MATERIAL = Aluminium
Heat Transfer Coefficient = 10 or 100 [W/m^2K]
Ambient Temperature = 25 [C]

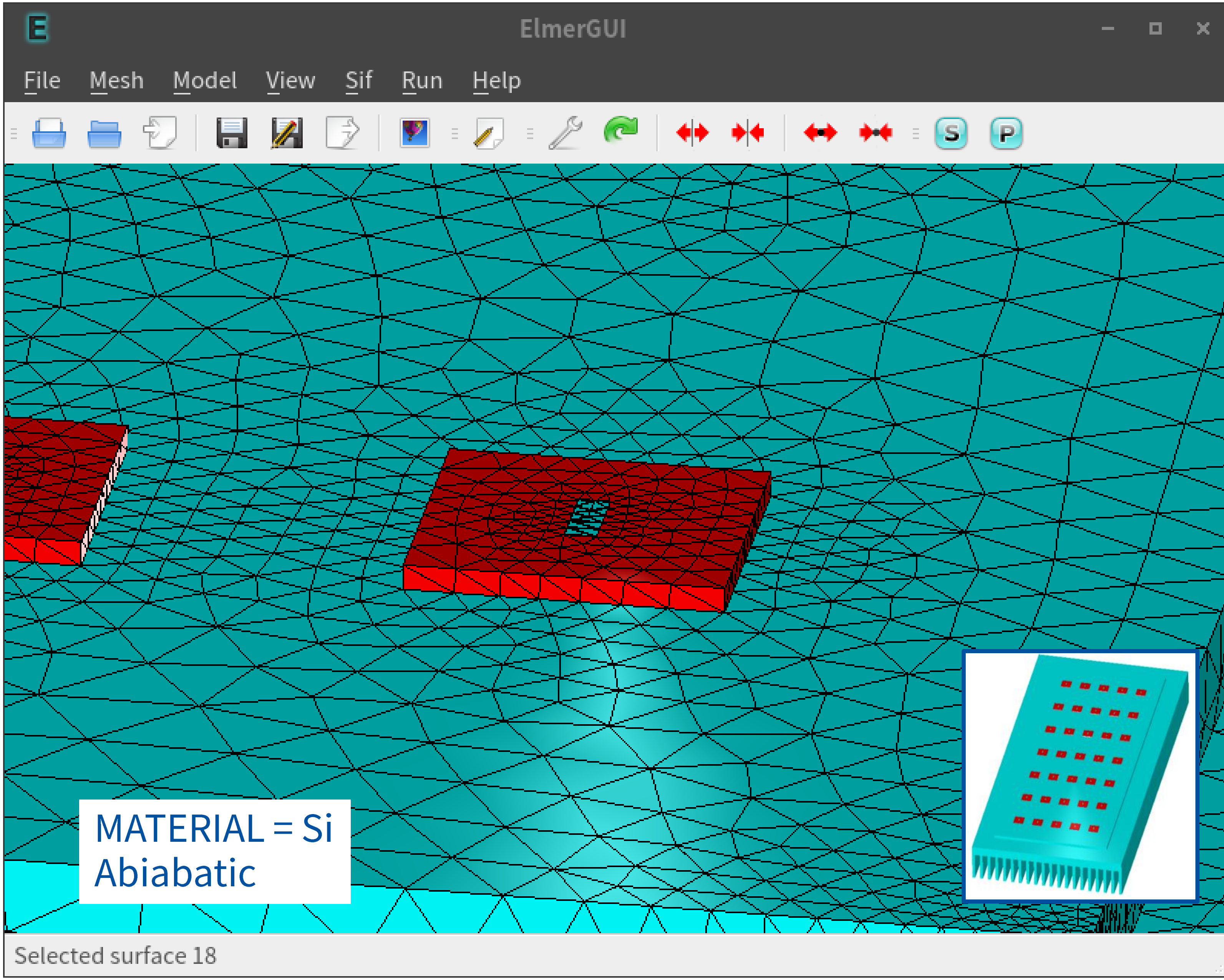

ElmerGUI
File
Mesh
Model
View
Sif
Run
Help
MATERIAL = Thermal Grease
Abiabatic
Selected surface 16

PCB Boundary

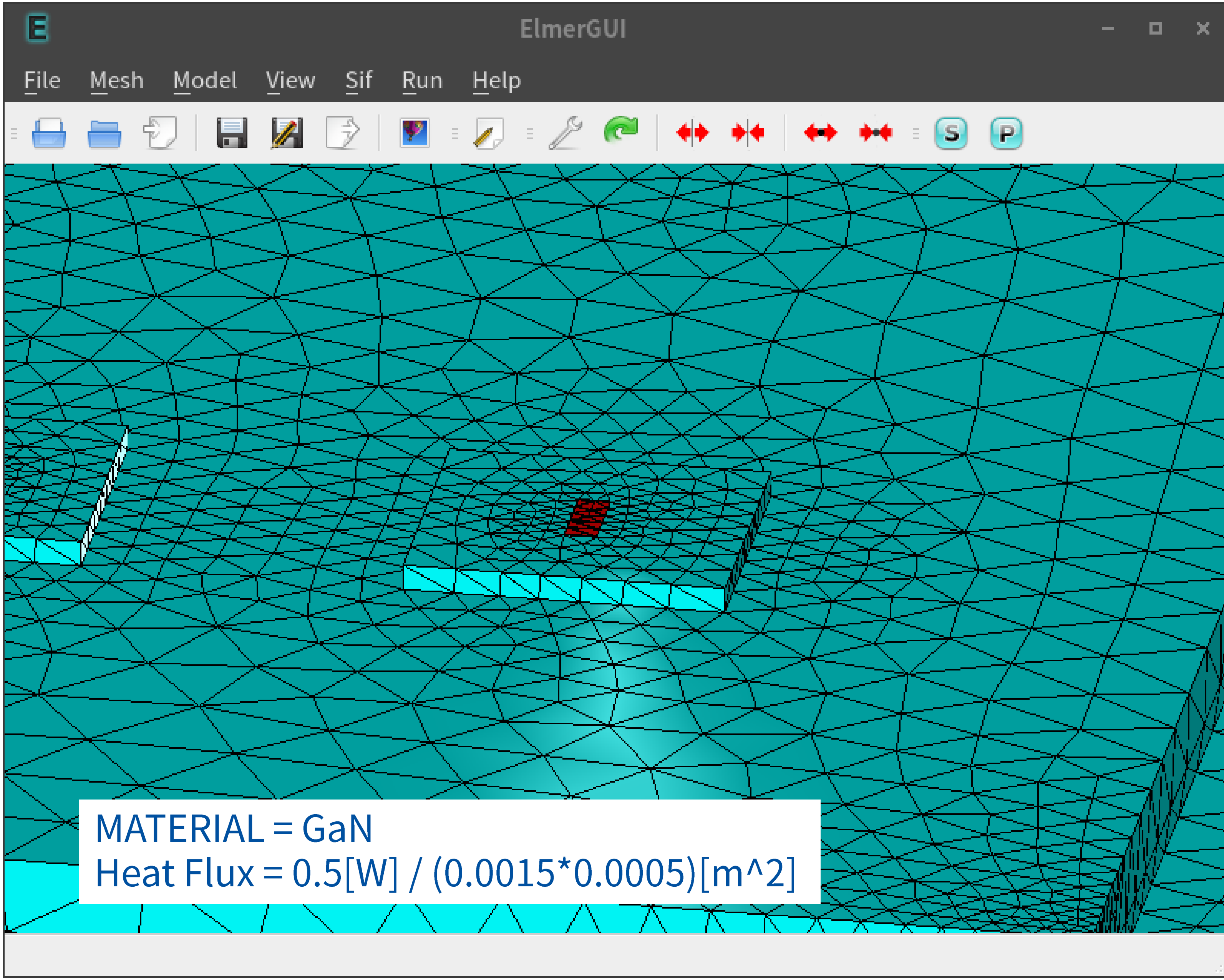


MATERIAL = Aluminium or FR4
Abiabatic

SLUG Boundary



LED Boundary



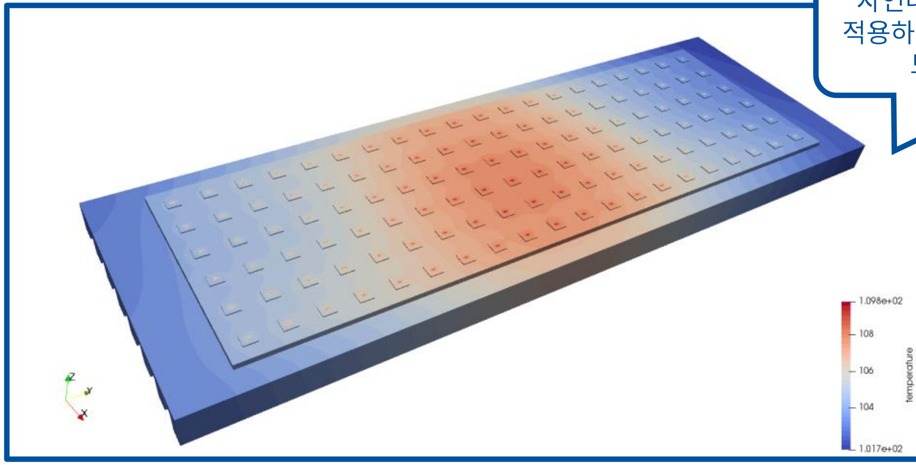
MATERIAL = GaN

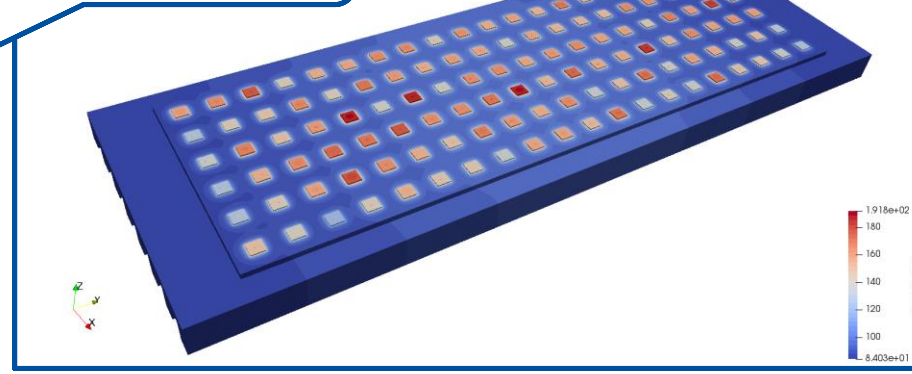
Heat Flux = 0.5[W] / (0.0015*0.0005)[m^2]

Result

Temperature

자연대류상태에서, 메탈코어PCB를
적용하면 82[C] 가량의 LED 접합부 온
도 저하를 기대할 수 있다.

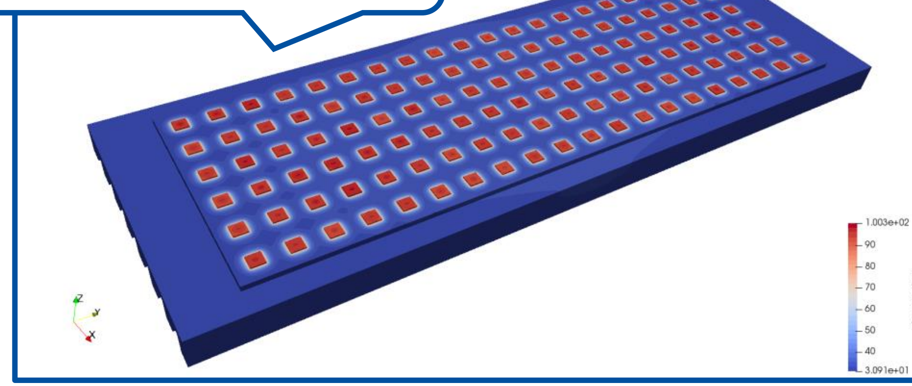




- Case01 (Metal PCB, Natural Convection)
  Max 109.8[C], Min 101.7[C]
- Case02 (FR4 PCB, Natural Convection)
  Max 191.8[C], Min 84.03[C]

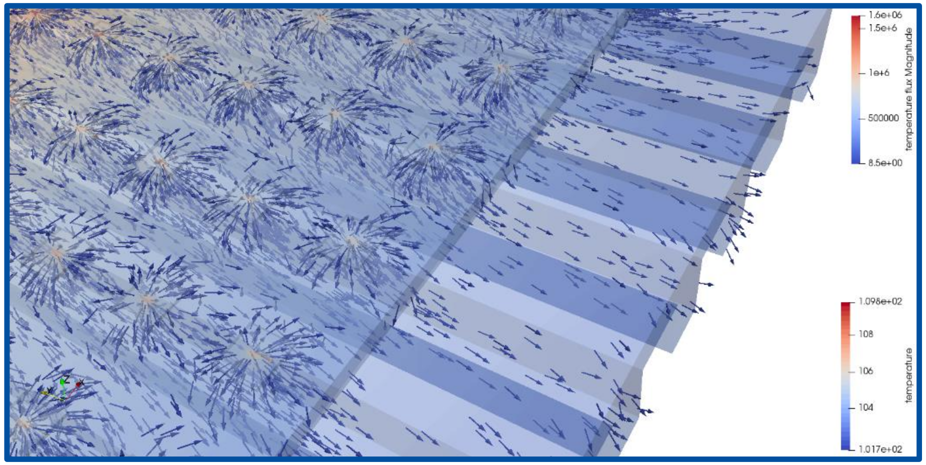
강제대류를 적용하면 73[C] 가량의
LED 접합부 온도 저하를 기대할 수 있다.

강제대류를 적용하면 91.5[C] 가량의
LED 접합부 온도 저하를 기대할 수 있다.

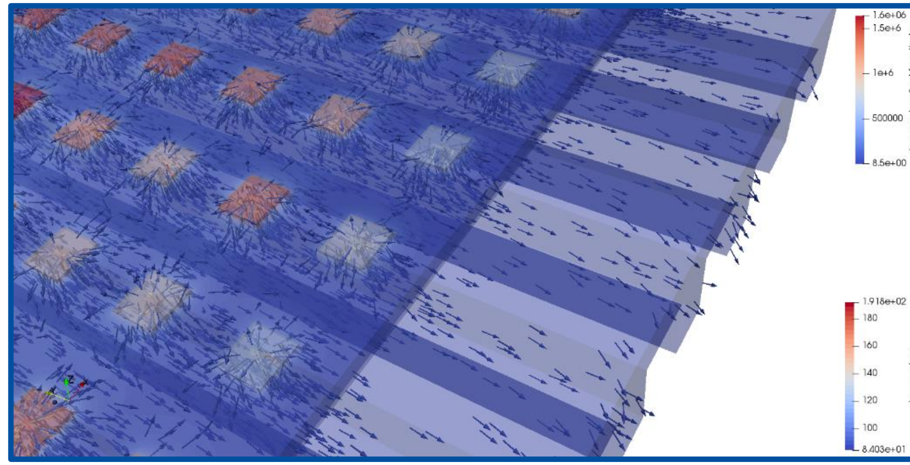


- Case03 (Metal PCB, Forced Convection)
  Max 36.33[C], Min 31.88[C]
- Case04 (FR4 PCB, Forced Convection)
  Max 100.3[C], Min 30.91[C]

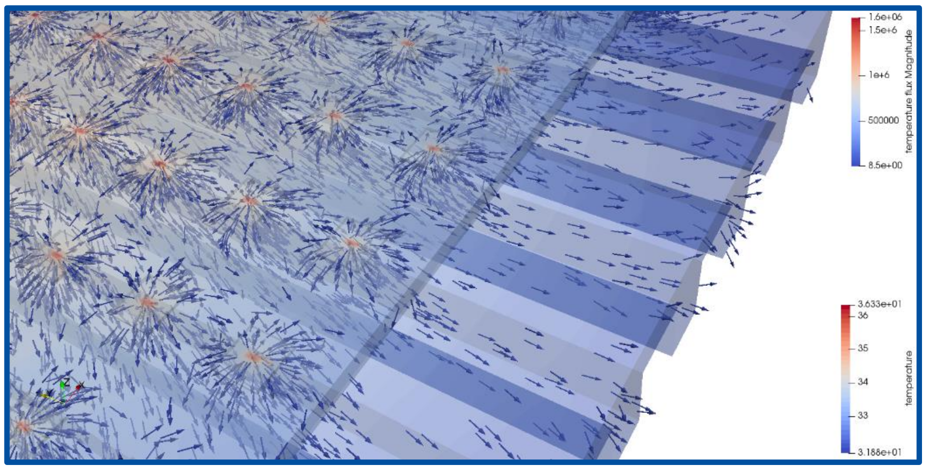
Temperature Flux (around LED)



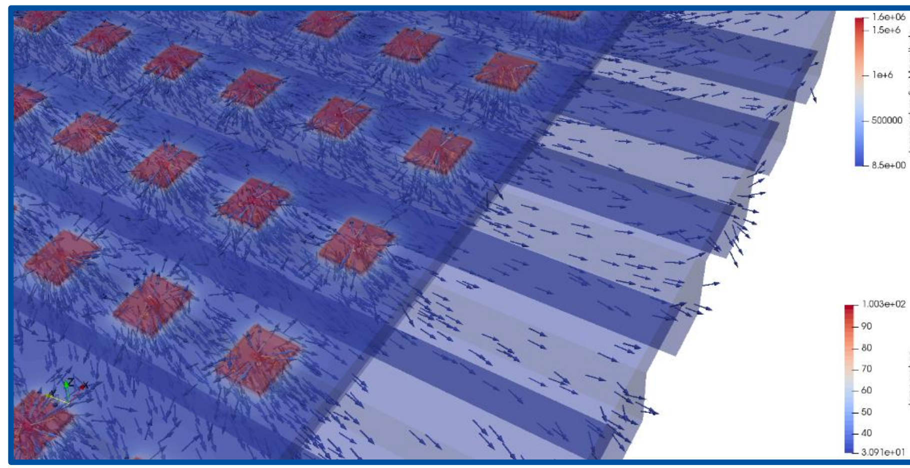
- Case01 (Metal PCB, Natural Convection)



- Case02 (FR4 PCB, Natural Convection)



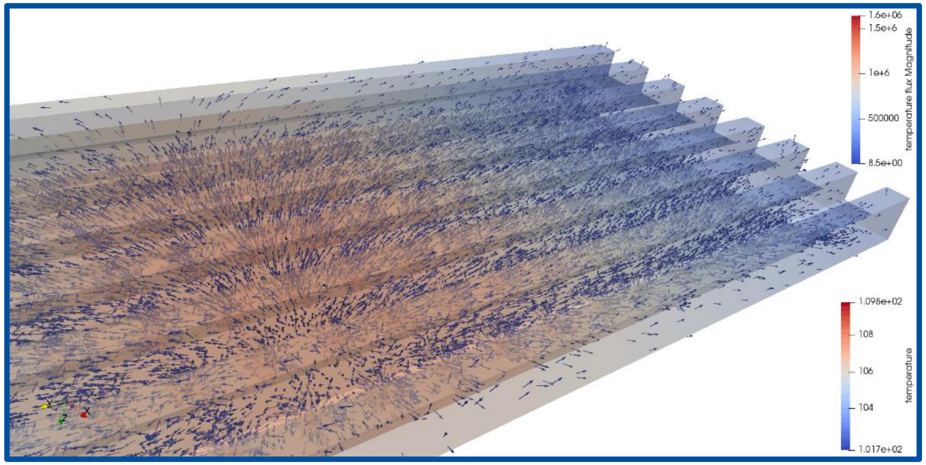
- Case03 (Metal PCB, Forced Convection)



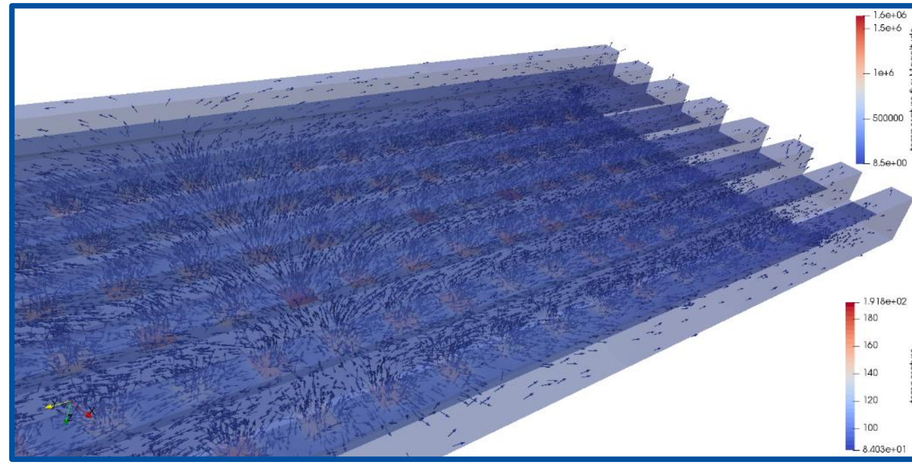
- Case04 (FR4 PCB, Forced Convection)

Temperature Flux 방향은 거의 동일하게 나타난다.

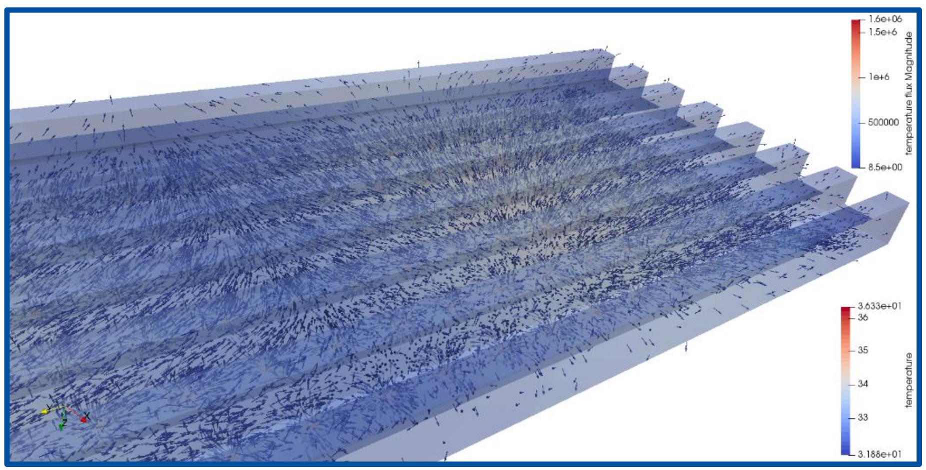
Temperature Flux (around HEATSINK)



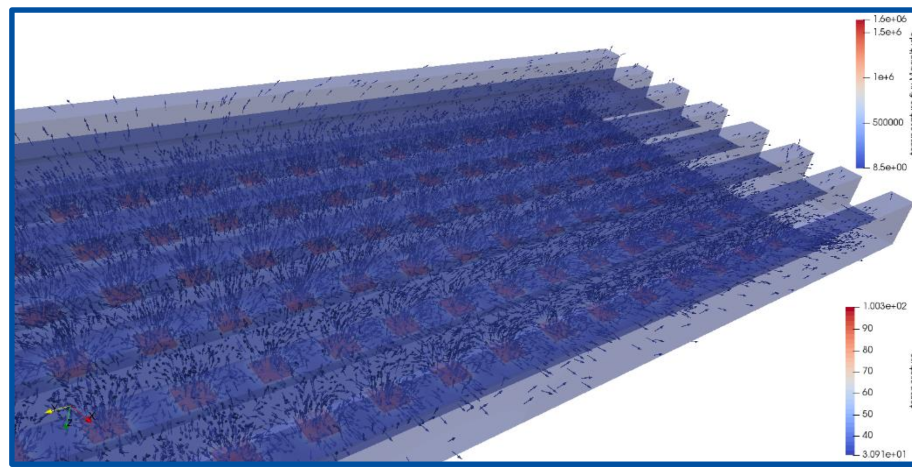
- Case01 (Metal PCB, Natural Convection)



- Case02 (FR4 PCB, Natural Convection)



- Case03 (Metal PCB, Forced Convection)



- Case04 (FR4 PCB, Forced Convection)

Temperature Flux 방향은 거의 동일하게 나타난다.

Summary

1. LED의 T_J 온도를 근접하게 예측할 수 있는 유한요소모델 해석 조건을 찾아내었다.

2. 모델링 및 해석 과정을 자동화할 수 있는 스크립트 코드를 작성하여 적용하였다.

3. 메탈코어PCB 적용시 큰 효과를 볼 수 있다.

4. 일반 FR4 PCB 적용시에는, 강제대류를 해 주면 메탈코어PCB 자연대류 상황과 비슷한 수준의 T_J를 얻을 수 있다.

5. 메탈코어PCB와 강제대류를 동시 적용시에는 비약적인 효과를 볼 수 있다.

6. 단, 본 모델에서는 TIM의 두께가 0.5mm로 매우 두껍게 설정되어 있어 실제보다 온도 수준이 높게 나타난 경우이다. (매우 보수적인 해석 조건)

fin