Laser additive manufacturing Thermal Field Prediction (LTFP)

Jun Fan, Weihao Liu

NERS 570 Final Project

December 4, 2023

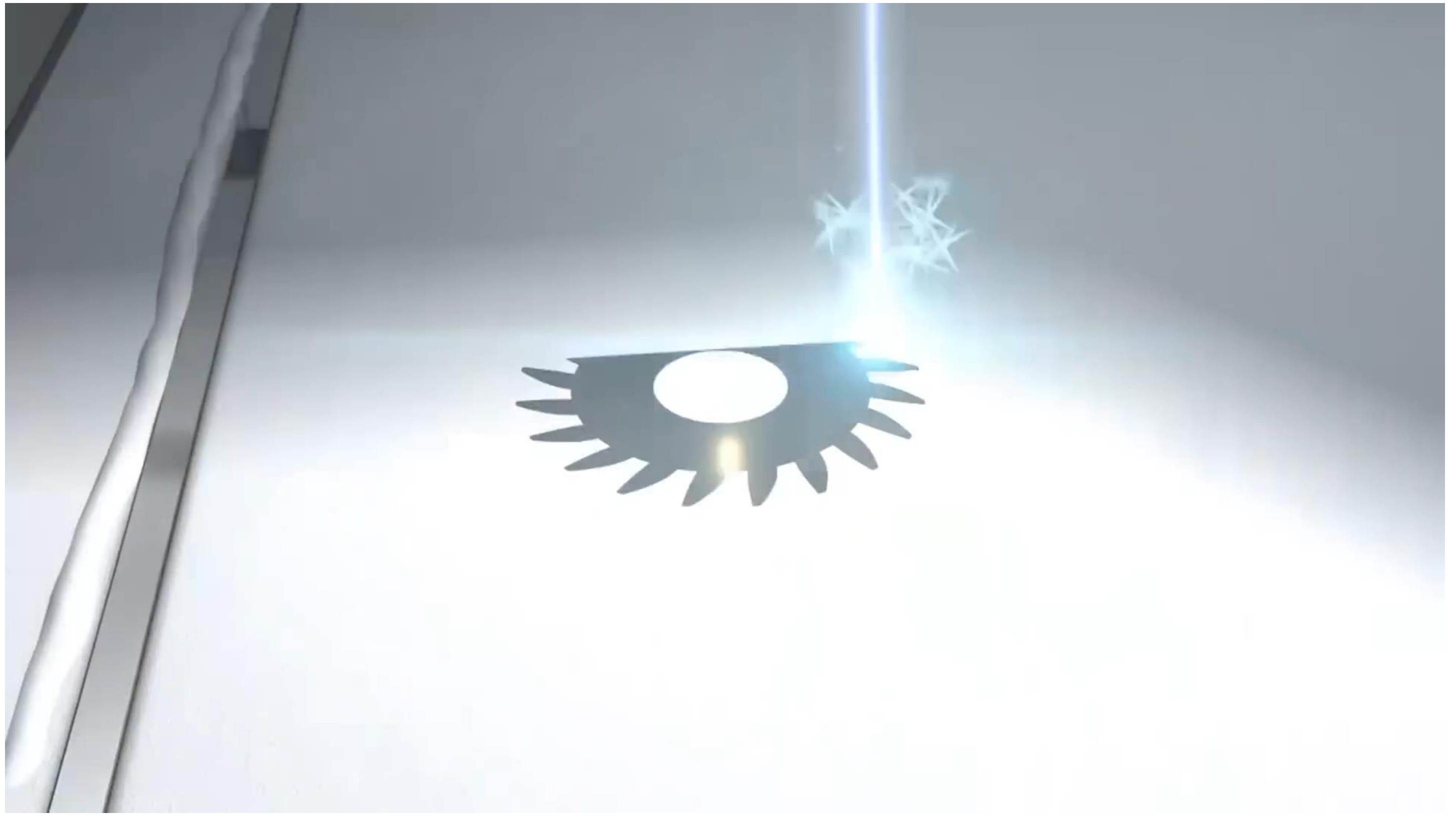
MECHANICAL ENGINEERING
UNIVERSITY OF MICHIGAN

Content

- Background
- Methods
- Programming
- Result and Discussion
- Summary and Future Work

Background

- **<u>L</u>**aser **<u>A</u>**dditive **<u>M</u>**etal **<u>M</u>**anufacturing (LAMM):
  - Involves a set of metal additive manufacturing processes
  - Harness high power density of lasers to melt and fuse the metal powder and build the structure layer by layer
  - Unprecedented geometry flexibility & rapid prototyping capability
- Laser Beam Scans Pattern:
  - Various scan patterns are developed trying to improve product quality

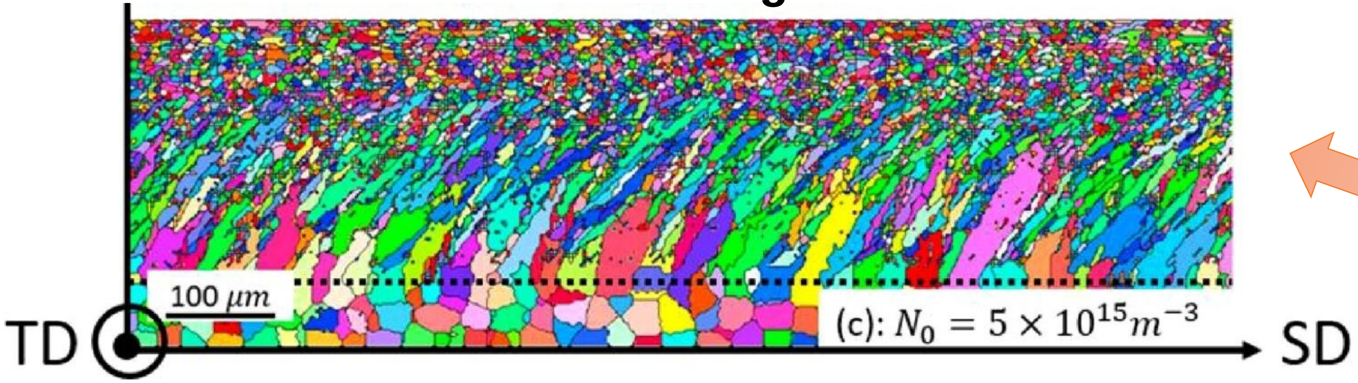


Source: https://www.youtube.com/watch?v=bgQvqVq-SQU

Background

- ❖ Different scan pattern leads to different temperature distribution in the material, leading to different grain growth
- ❖ Grain growth is difficult to monitor during the printing
  - ➢ Simulation is essential to revealing the grain formation process
  - ➢ Numerical models can be used to improve the scan path design by predicting the grain distribution before part is actually printed

Grain distribution of a single laser line scan



Product geometry

Scan path design

Numerical model

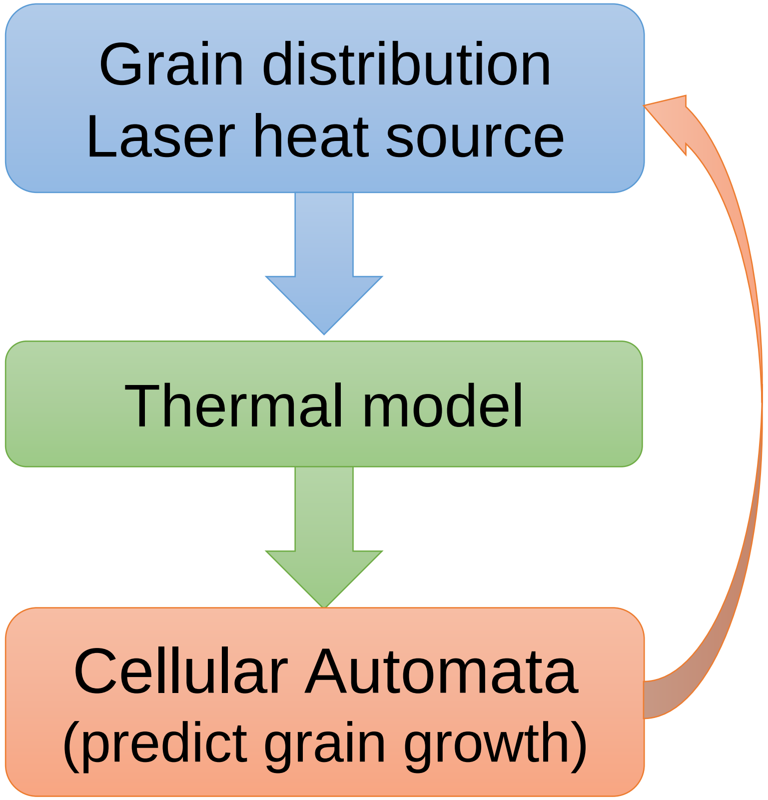
- ➢ Residual stress concentration
- ➢ Micro-defects
- ➢ Deformation of the printed part
- ➢ Grain formation

Xiao, Zhen, et al. "Recent progress on microstructure manipulation of aluminium alloys manufactured via laser powder bed fusion." *Virtual and Physical Prototyping* 18.1 (2023): e2125880.

Motivation

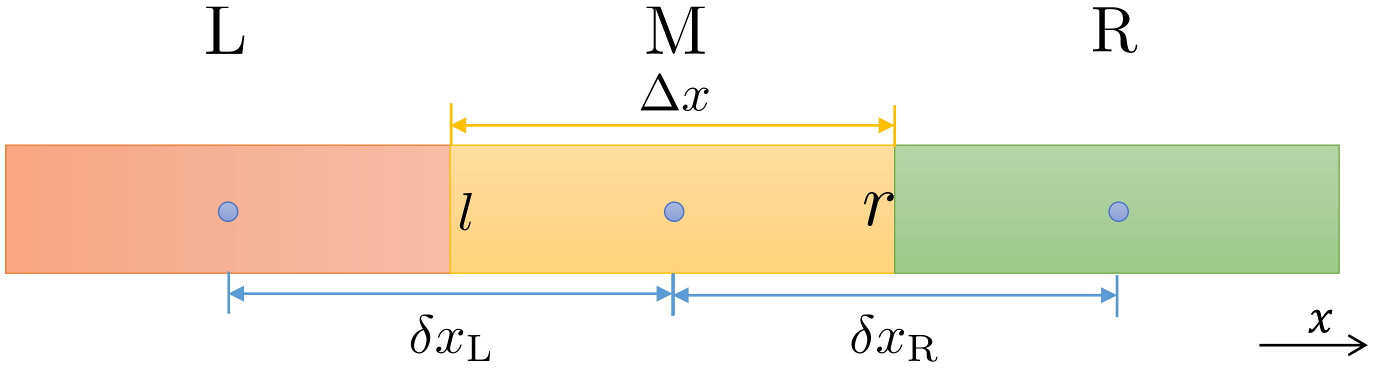
- Thermal field is essential to the prediction of grain growth and other phenomena during LAMM
- Cellular Automata (CA) model
  - Relays on precomputed thermal field
  - Lack of interaction with CA model
  - Low flexibility
- Laser scan Thermal Field Prediction (LTFP)
  - Interact with the CA model
  - Various boundary conditions and domain increment
  - High efficiency (Parallelization)

Grain growth prediction workflow



Method

- **<u>F</u>inite <u>V</u>olume <u>M</u>ethod (FVM):**
  - Computational domain is divided into non-overlapping control volumes
  - The state is stored in each cell, typically the mean value
- Diffusion Equation:
  - 1D Case: $\dfrac{\partial u}{\partial t} = \dfrac{k}{\rho}\dfrac{\partial^2 T}{\partial x^2} + \dot{q}$
  - *u(T)*: specific internal energy dependent on temperature



$$\int_l^r \frac{\partial u}{\partial t}\,\mathrm{d}x = \int_l^r \frac{k}{\rho}\frac{\partial^2 T}{\partial x^2}\,\mathrm{d}x + \int_l^r \dot{q}\,\mathrm{d}x \Rightarrow \frac{\mathrm{d}\bar{u}}{\mathrm{d}t} = \frac{1}{\rho\Delta x}\underbrace{\left(k\left.\frac{\partial T}{\partial x}\right|_r - k\left.\frac{\partial T}{\partial x}\right|_l\right)}_{\text{Internal Flux}} + \underbrace{\bar{\dot{q}}}_{\text{External Source}}$$

- Define: $\bar{X} \equiv \dfrac{1}{\Delta x}\displaystyle\int_l^r X\,\mathrm{d}x$

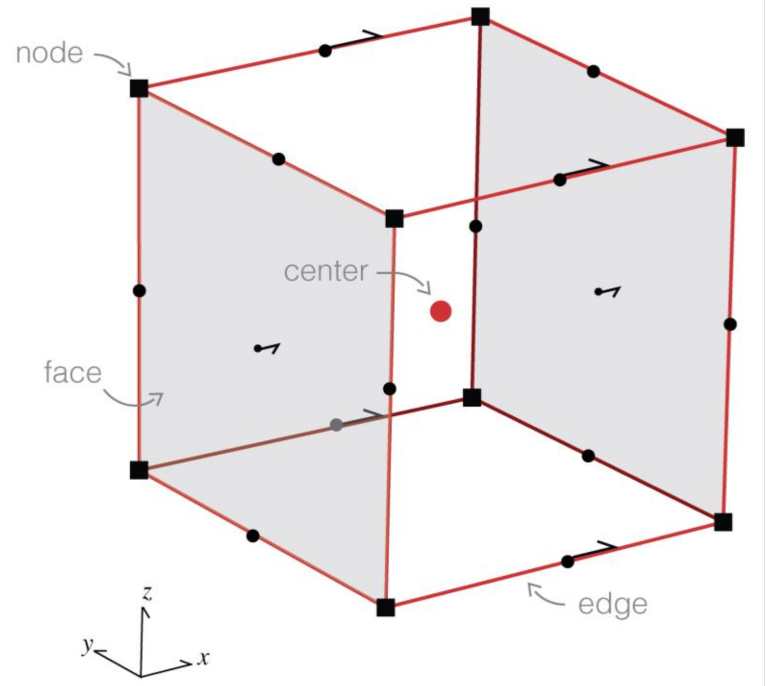
$$\frac{\mathrm{d}\bar{u}}{\mathrm{d}t} = \frac{1}{\rho\Delta x}\left[\frac{k_r\left(T_{\mathrm{R}} - T_{\mathrm{M}}\right)}{\delta x_{\mathrm{R}}} - \frac{k_l\left(T_{\mathrm{M}} - T_{\mathrm{L}}\right)}{\delta x_{\mathrm{L}}}\right] + \bar{\dot{q}}$$

Method

- **F**inite **V**olume **M**ethod (FVM) in 3D:
- Governing equation: $$\frac{\partial u}{\partial t} = \frac{k}{\rho}\nabla^2 T + \dot{q}$$
- Also define: $$\bar{X} \equiv \frac{1}{V}\int_\Omega X \,\mathrm{d}V$$
- Recall 1D case: $$\frac{\mathrm{d}\bar{u}}{\mathrm{d}t} = \frac{1}{\rho\Delta x}\left(k\left.\frac{\partial T}{\partial x}\right|_r - k\left.\frac{\partial T}{\partial x}\right|_l\right) + \bar{\dot{q}}$$
- Similarly, we have (with central discretization):

$$\begin{aligned}\frac{\mathrm{d}\bar{u}}{\mathrm{d}t} = \frac{1}{\rho\Delta V}\Bigg[&\frac{k_{x^+}\left(T_{x^+} - T_{\mathrm{M}}\right)}{\delta x_+}\Delta y\Delta z + \frac{k_{x^-}\left(T_{x^-} - T_{\mathrm{M}}\right)}{\delta x_-}\Delta y\Delta z \\ &+ \frac{k_{y^+}\left(T_{y^+} - T_{\mathrm{M}}\right)}{\delta y_+}\Delta z\Delta x + \frac{k_{y^-}\left(T_{y^-} - T_{\mathrm{M}}\right)}{\delta y_-}\Delta z\Delta x \\ &+ \frac{k_{z^+}\left(T_{z^+} - T_{\mathrm{M}}\right)}{\delta z_+}\Delta x\Delta y + \frac{k_{z^-}\left(T_{z^-} - T_{\mathrm{M}}\right)}{\delta z_-}\Delta x\Delta y\Bigg] + \bar{\dot{q}}\end{aligned}$$

$$\Longrightarrow \quad \frac{\mathrm{d}\bar{u}}{\mathrm{d}t} = \cdots \equiv f(T)$$



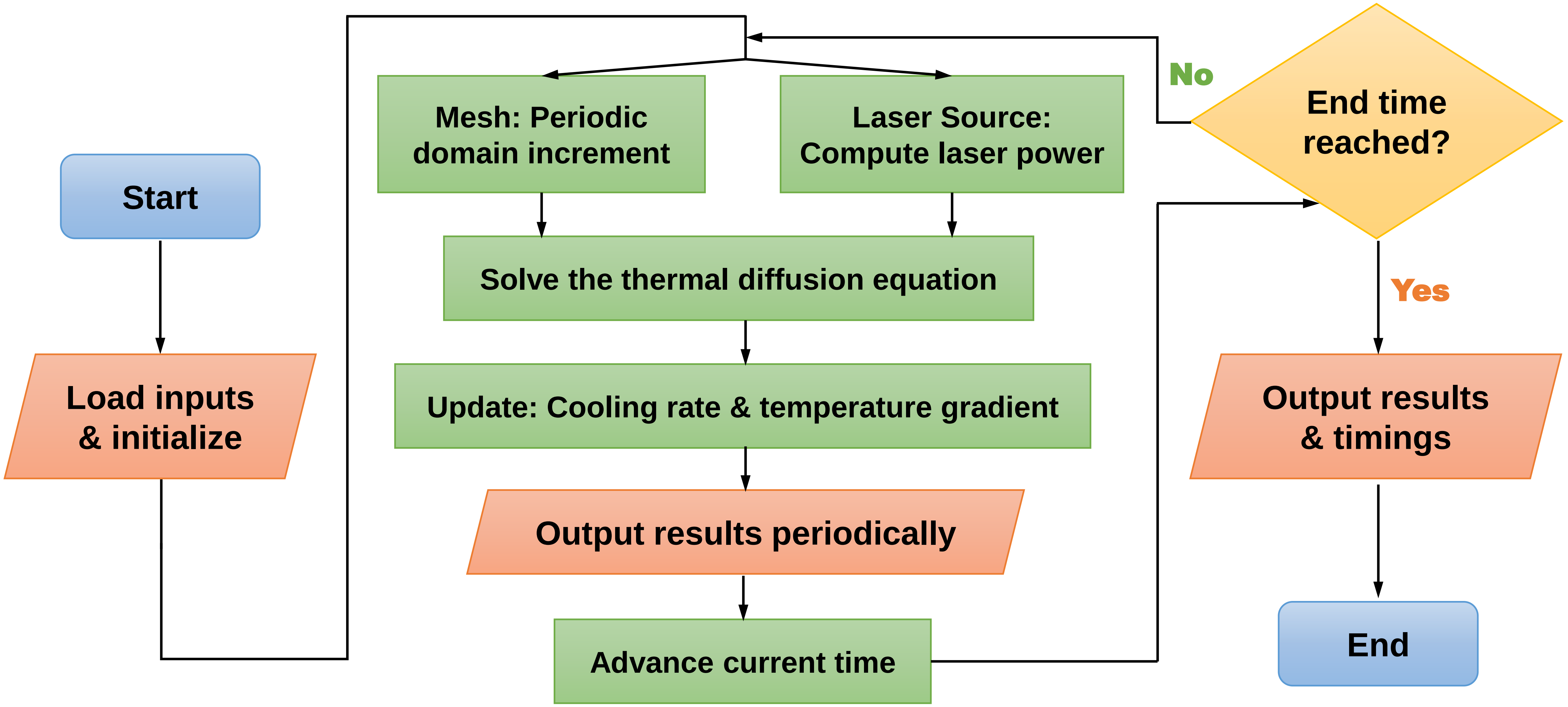
Source: https://row1.ca/pixels-and-their-neighbors

Method

- Time stepping: $\dfrac{\mathrm{d}\bar{u}}{\mathrm{d}t} = \cdots \equiv f(T)$
  - **F**orward **E**uler (FE): $\dfrac{\mathrm{d}\bar{u}}{\mathrm{d}t} = f(T) \implies \boxed{\bar{u}_{i,j,k}^{(n+1)} = \bar{u}_{i,j,k}^{(n)} + \Delta t_{i,j,k}^{(n)} f\left(T_{i,j,k}^{(n)}\right)}$
  - RK2: $\begin{cases} u_{i,j,k}^{(n+1/2)} = \bar{u}_{i,j,k}^{(n)} + \Delta t_{i,j,k}^{(n)} f\left(T_{i,j,k}^{(n)}\right) \\ u_{i,j,k}^{(n+1)} = \bar{u}_{i,j,k}^{(n)} + \dfrac{\Delta t_{i,j,k}^{(n)}}{2}\left[f\left(T_{i,j,k}^{(n)}\right) + f\left(T_{i,j,k}^{(n+1/2)}\right)\right] \end{cases}$
  - Von Neumann stability Criterion: $\Delta t_{i,j,k}^{(n)} \leqslant \min\left[\dfrac{(\Delta x)^2}{6k^{(n)}/\rho}, \dfrac{(\Delta y)^2}{6k^{(n)}/\rho}, \dfrac{(\Delta z)^2}{6k^{(n)}/\rho}\right]$
- **Cooling rate** and **temperature gradient** are needed for the grain growth model:

$$-\frac{\partial T^{(n)}}{\partial t} = \frac{T^{(n-1)} - T^{(n)}}{\Delta t}, \quad \nabla T^{(n)} = \left(\frac{T_{i+1,j,k}^{(n)} - T_{i-1,j,k}^{(n)}}{2\Delta x}, \frac{T_{i,j+1}^{(n)} - T_{i,j-1}^{(n)}}{2\Delta y}, \frac{T_{i,j,k+1}^{(n)} - T_{i,j,k-1}^{(n)}}{2\Delta z}\right)$$

Flowchart

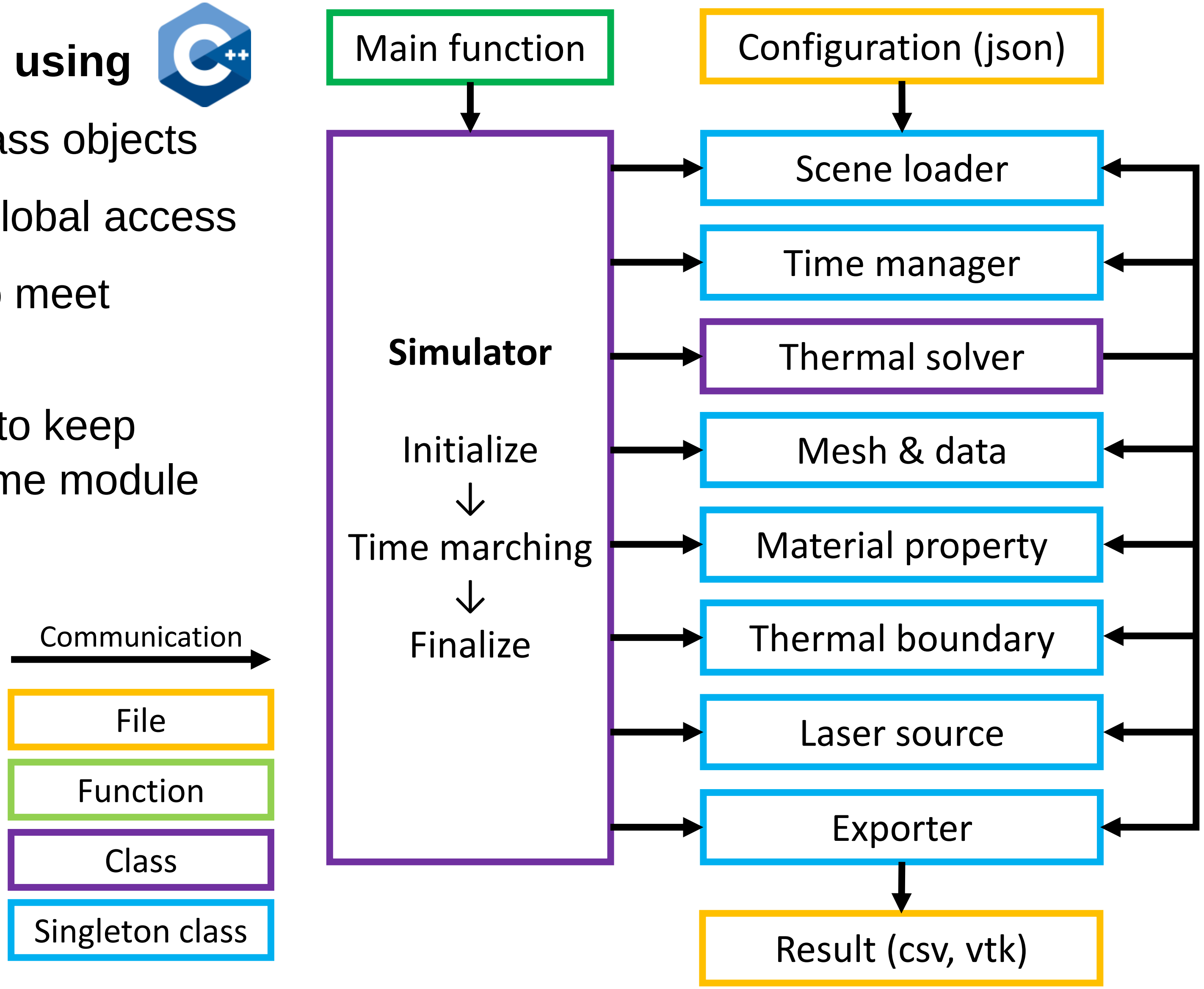


Start
Load inputs & initialize
Mesh: Periodic domain increment
Laser Source: Compute laser power
Solve the thermal diffusion equation
Update: Cooling rate & temperature gradient
Output results periodically
Advance current time
End time reached?
No
Yes
Output results & timings
End

Code Algorithm

Object-oriented design and programming using C++

- All modules of simulation are written as class objects
- Make most modules singleton to provide global access
- Modules can be reassembled differently to meet different simulation needs
- Uses <u>inherited class</u> + <u>singleton manager</u> to keep track of different types of objects in the same module and provide unified interface

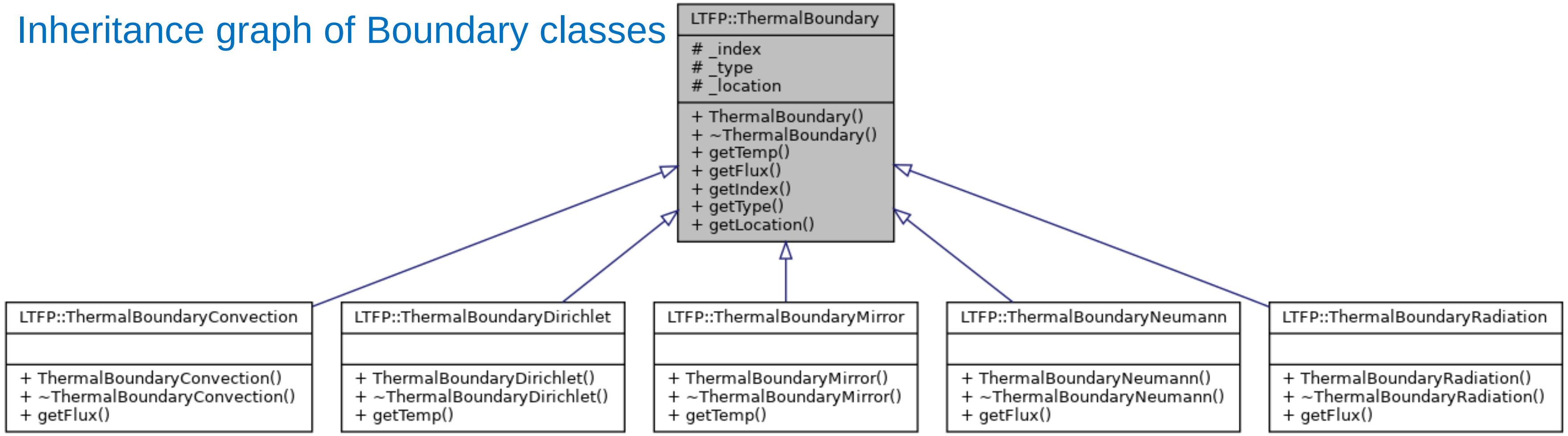


Code Algorithm

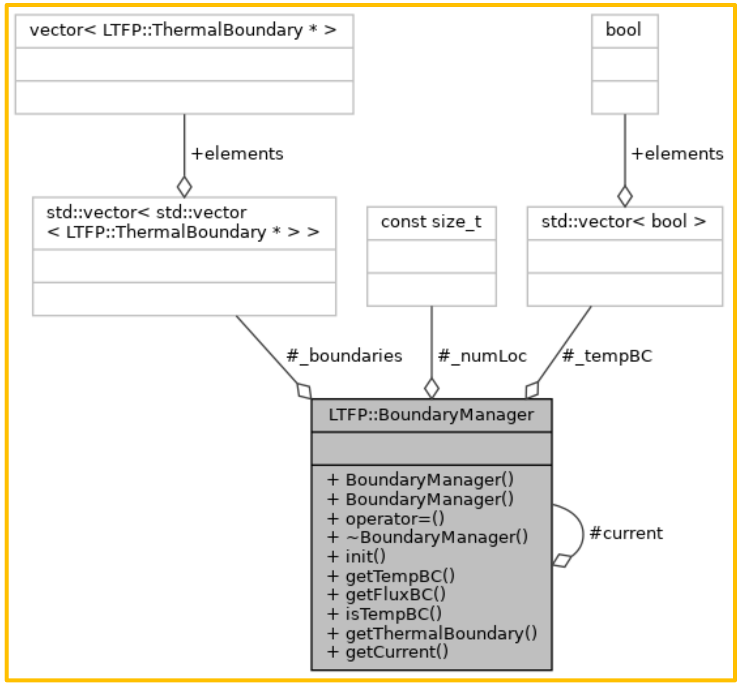
11 MAKE THE WORLD WORK BETTER

❖ Structure of thermal boundary module (diagrams generated by Doxygen)

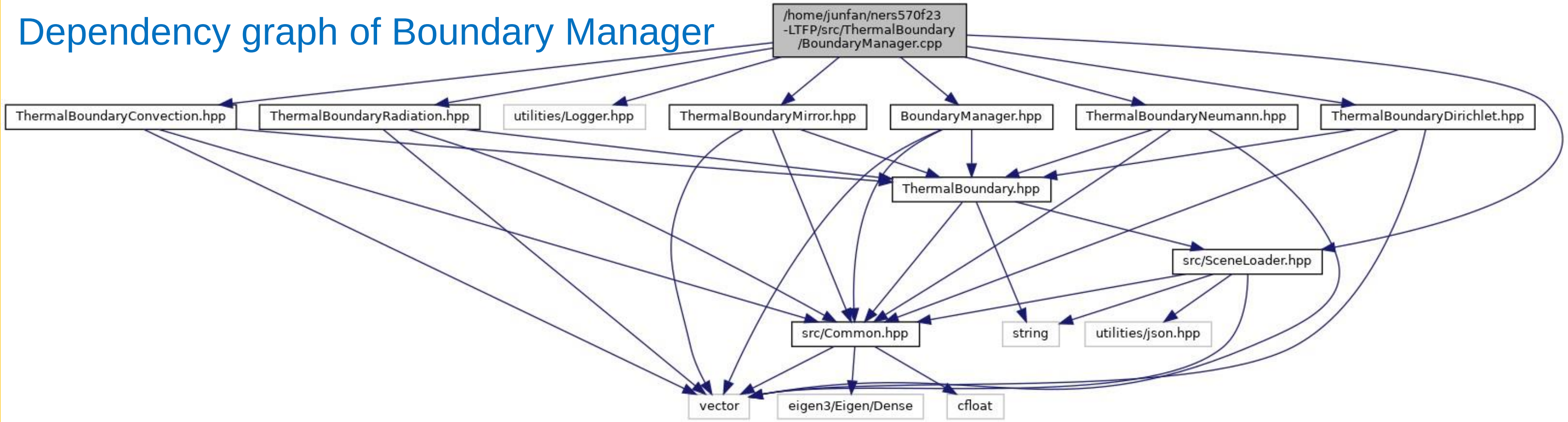
Inheritance graph of Boundary classes



UML diagram of Boundary Manager object



Dependency graph of Boundary Manager

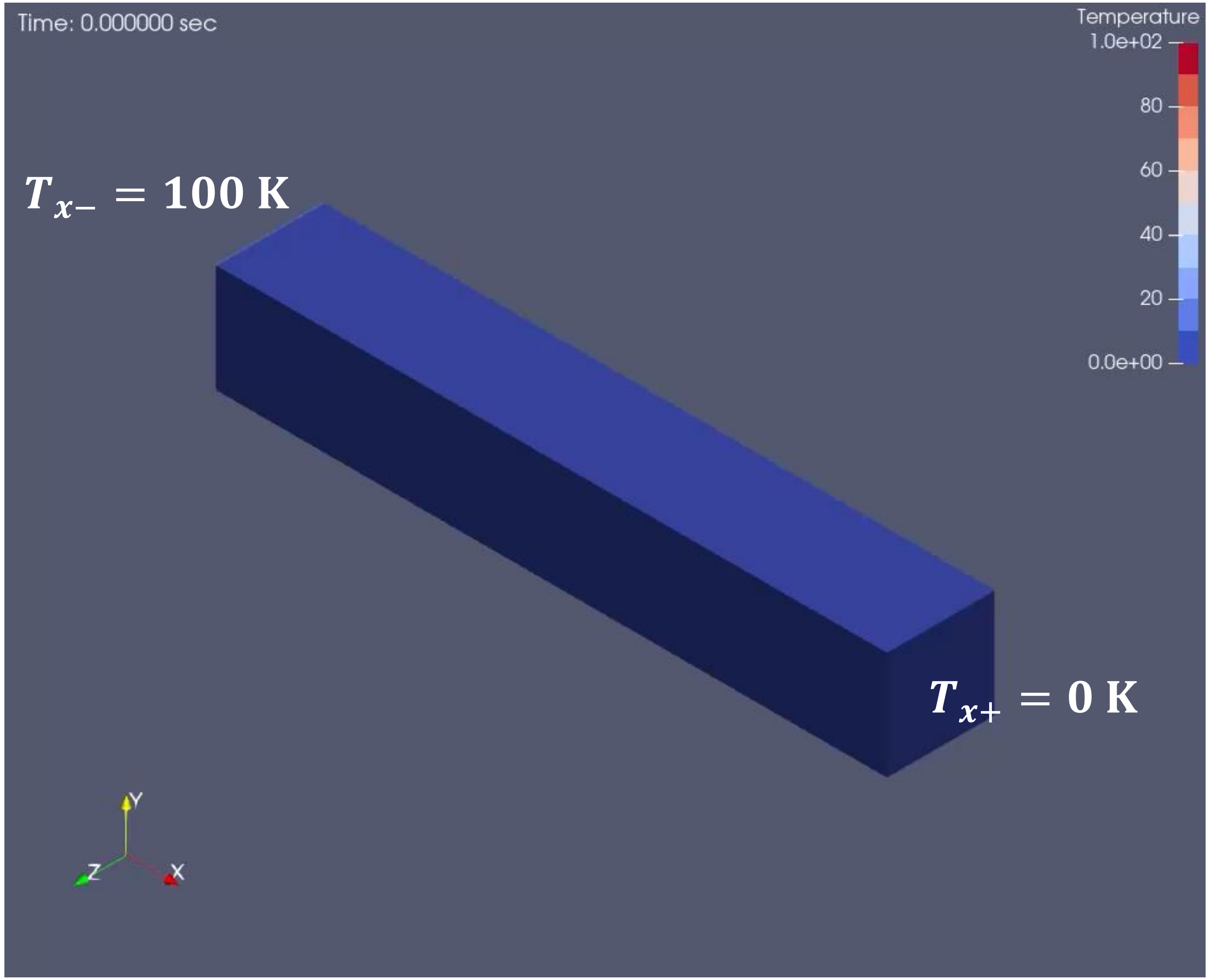


Simulation result – 1D diffusion

1D diffusion test

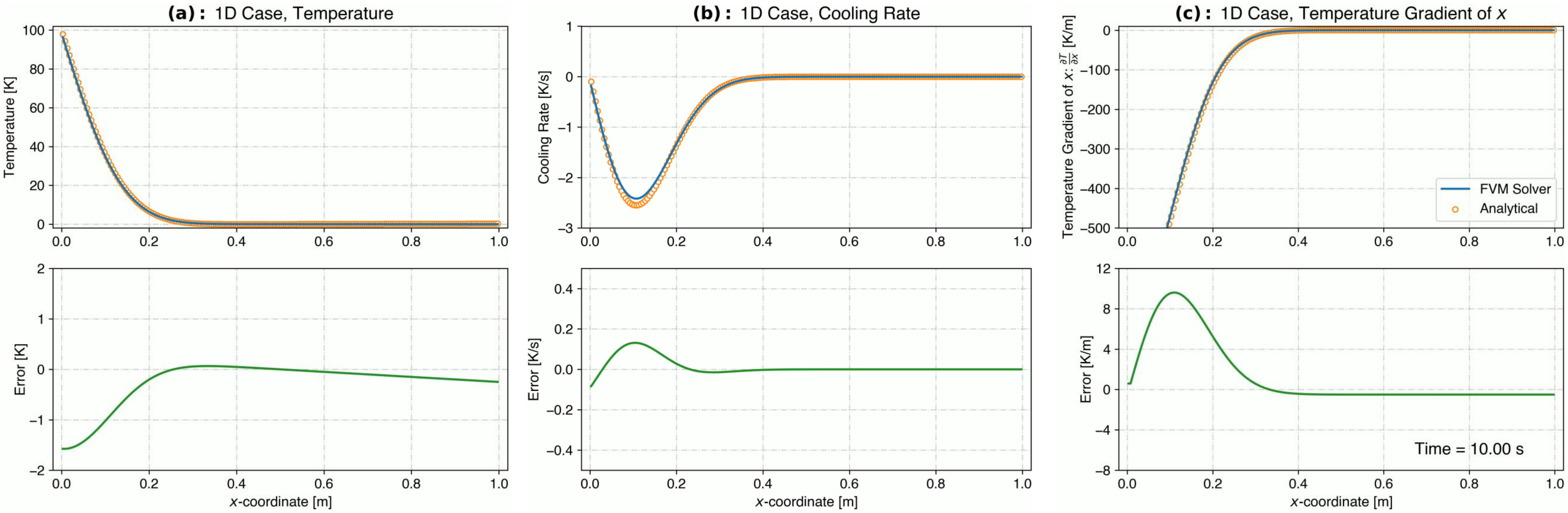
- Constant diffusion coefficient
- Initial temperature: 0 K
- Dirichlet boundary at
  - x^+: 0 K
  - x^-: 100 K
- Analytical solution:

$$T^*(t,x) = \sum_{n=1}^{+\infty} \left\{ \frac{200}{n\pi} \left[\frac{(-1)^n}{n\pi} - 1 \right] \sin(n\pi x) \times \mathrm{e}^{-\frac{k}{c_p \rho}(n\pi)^2 t} \right\} + 100 - 100x$$



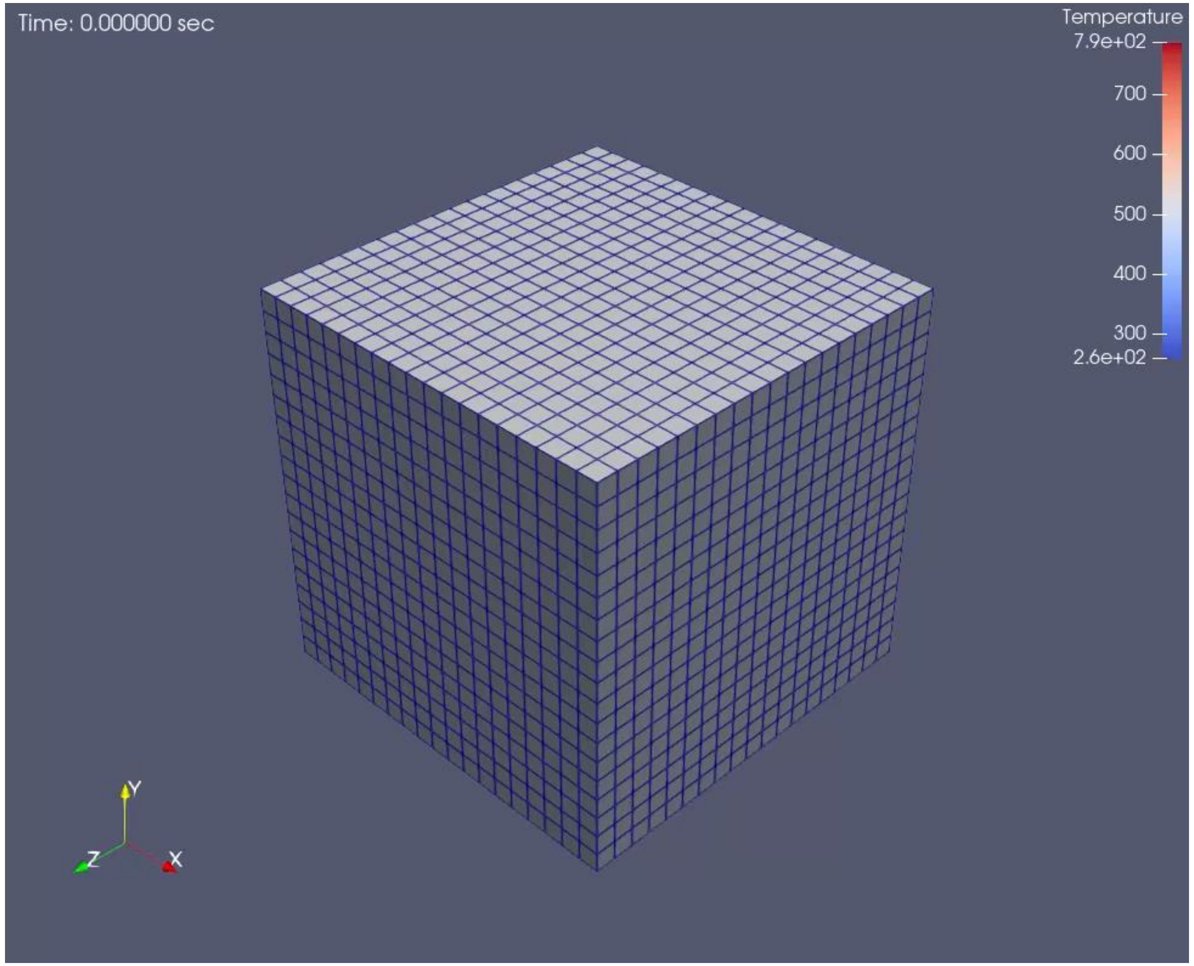
Simulation result – 1D diffusion

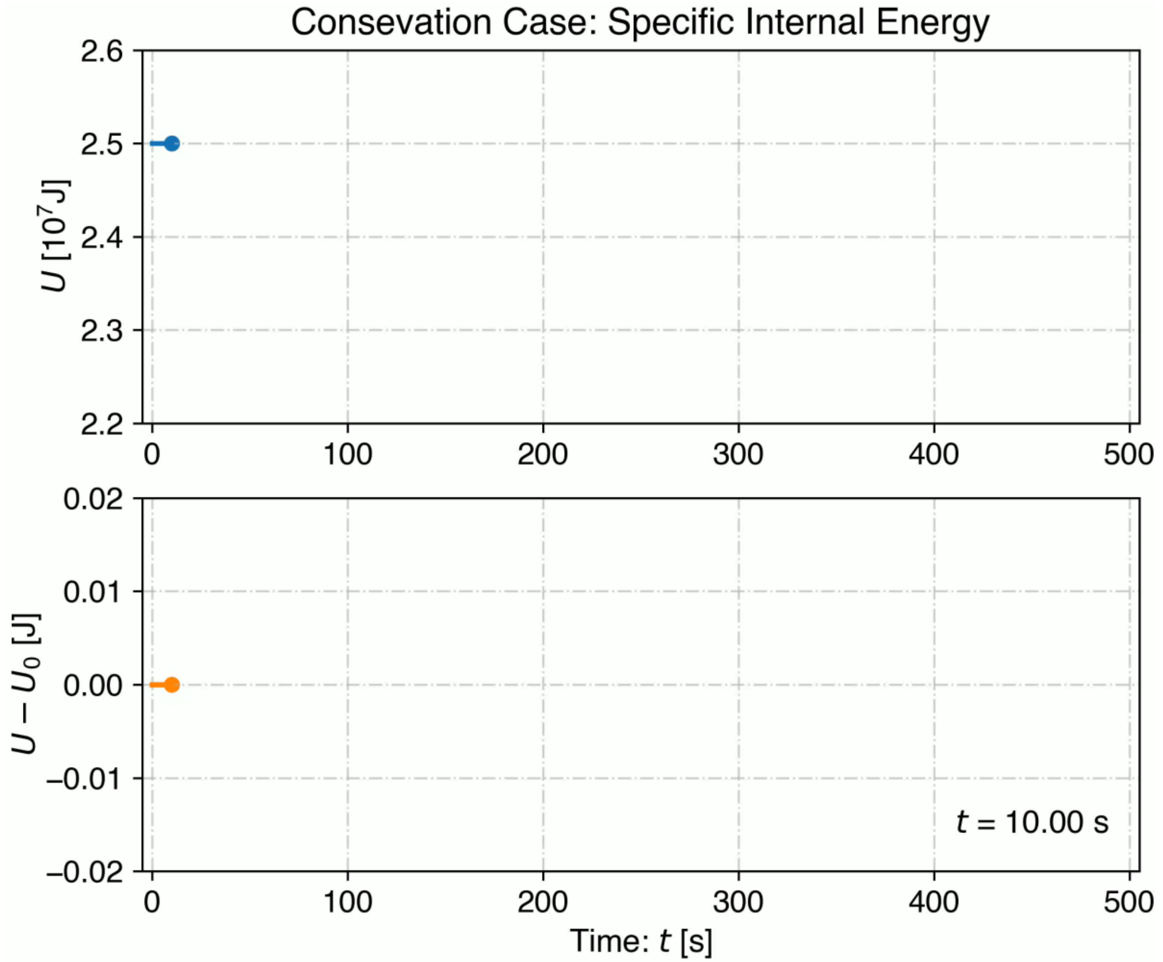
- Temperature, cooling rate and temperature gradient with respective to the x-axis (diffusion direction)
- Cooling rate and temperature gradient behave as expected



Simulation result – Conservation

- Conservation test case with Neumann boundary on all surfaces. All flux is summed up to zero.
- Applied temperature-dependent thermal conductivity and specific heat.

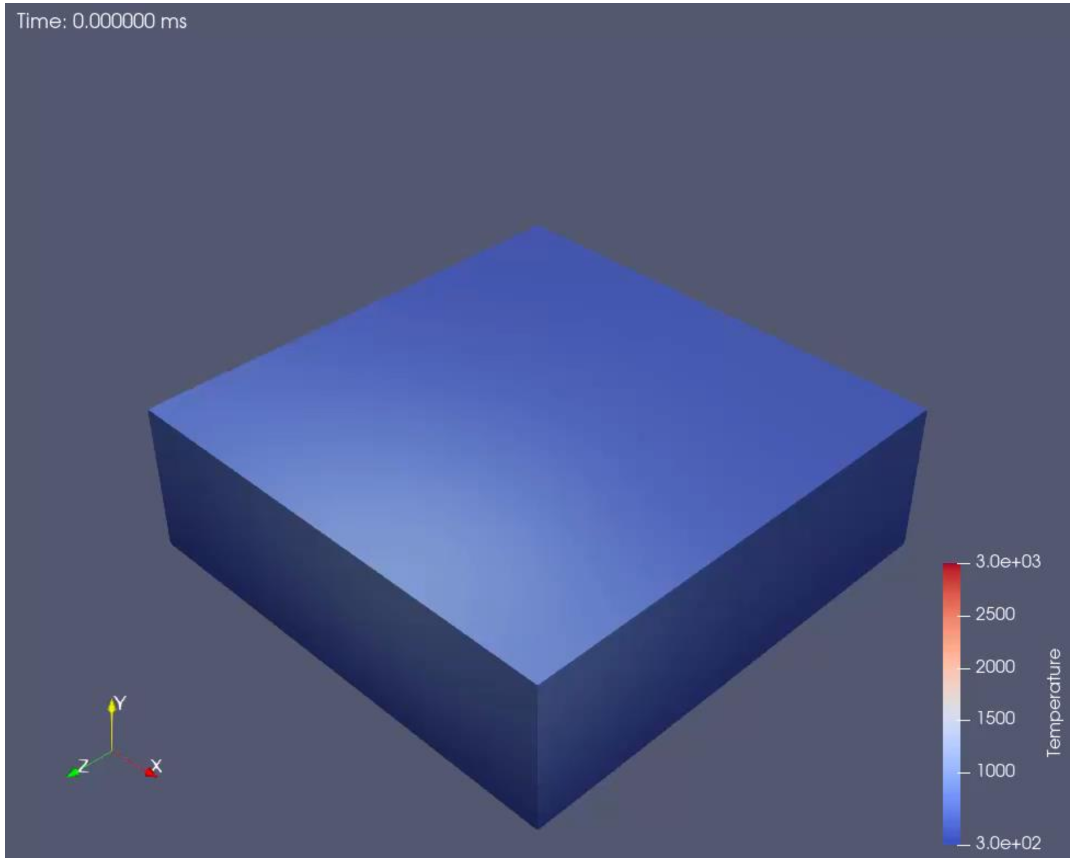




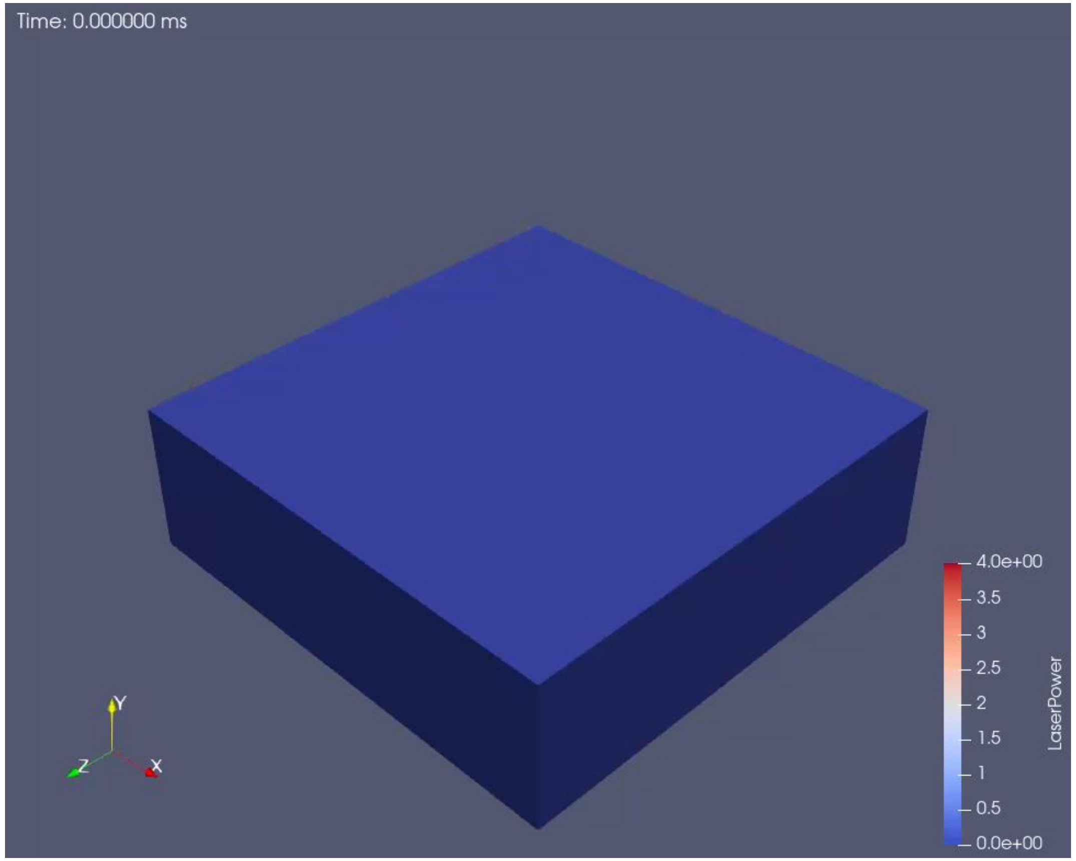
Simulation result –Multi-layer scan test

❖ Multi-layer scan on a block steel with domain increment

Temperature [K]

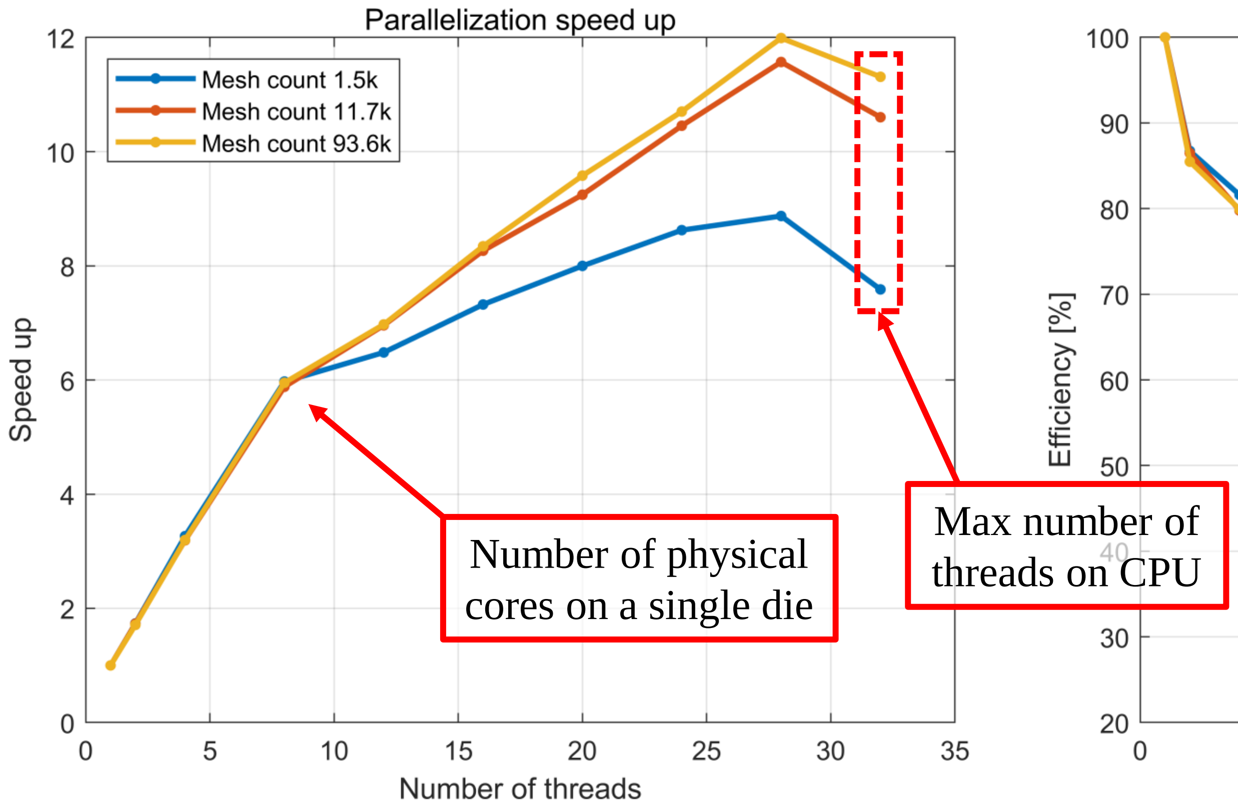


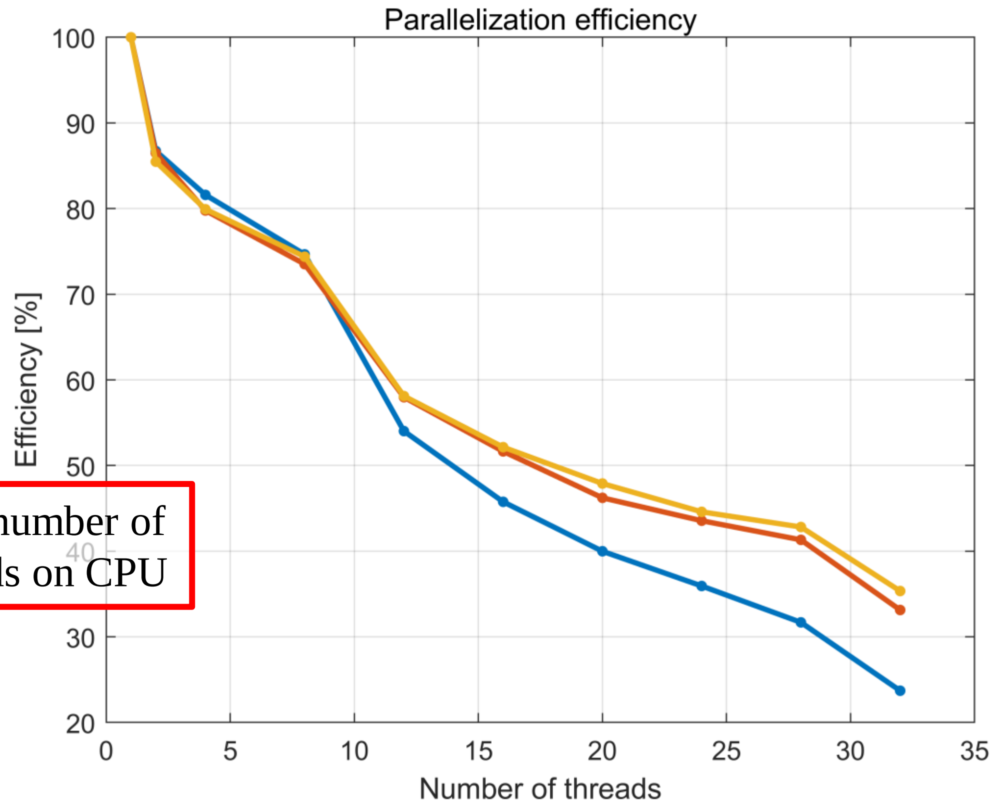
Laser power distribution [W]



Parallelization performance

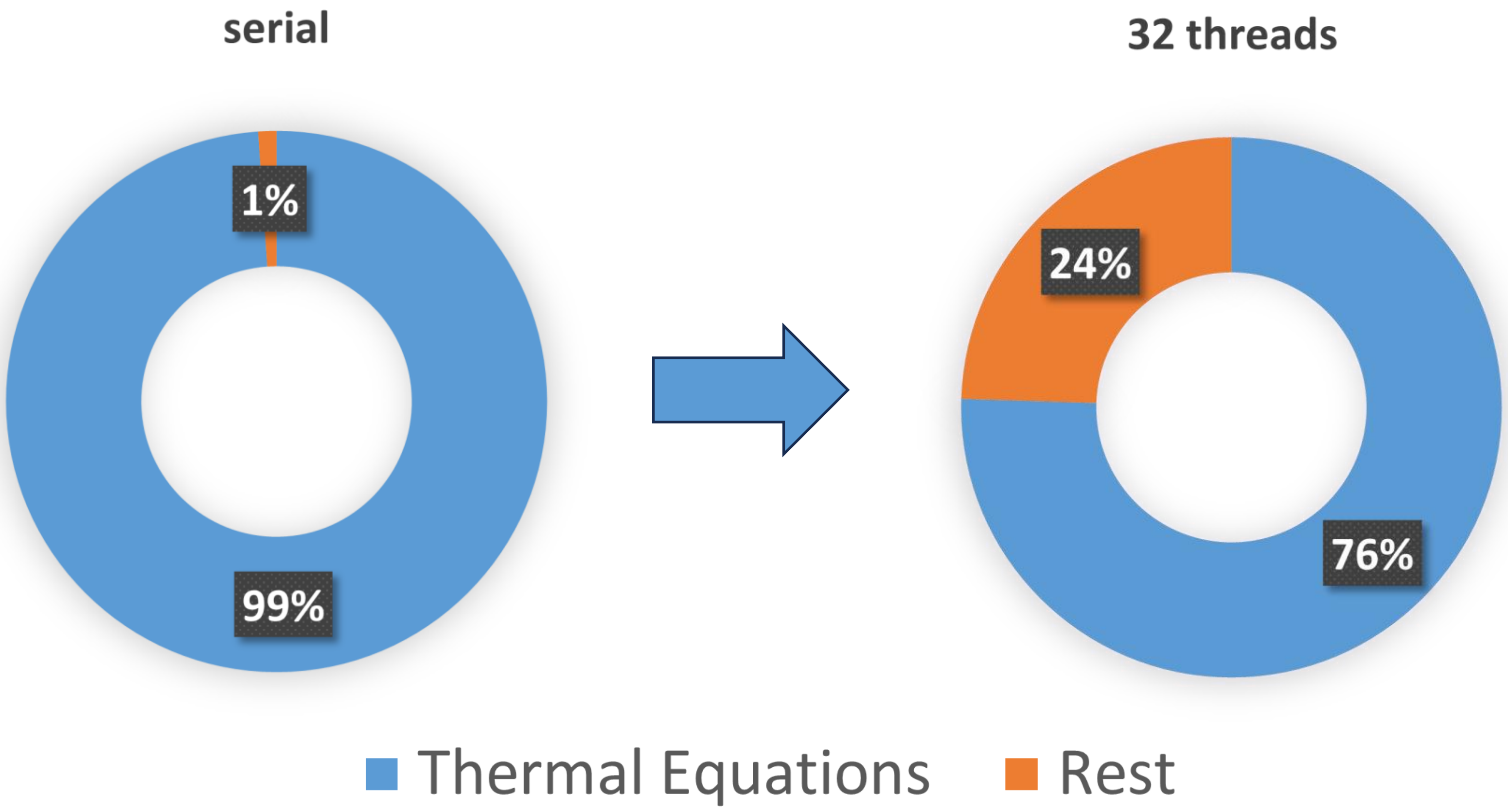
- Loops are parallelized with OpenMP
- The multilayer scan case is timed in WSL running on Ryzen 7950X, 16c32t, @5.4GHz





Parallelization performance

- Time consumption of solving thermal equations is greatly reduced after parallelization, but still takes a significant part of comptutation



Documentation

❖ Documentation is generated using **Doxygen**

LTFP Laser additive manufacturing Thermal Field Prediction
1.0.0

Main Page | Namespaces ▾ | Classes ▾ | Files ▾

- LTFP
  - Namespaces
    - Namespace List
      - LTFP
        - Exporter
        - ExporterCsvMesh
        - ExporterVtkMesh
        - ExportManager
        - LaserSource
        - MaterialProperty
        - MeshData
        - SceneLoader
        - Simulator
        - Solver
        - BoundaryManager
        - ThermalBoundary
        - ThermalBoundaryConvectio
        - ThermalBoundaryDirichlet
        - ThermalBoundaryMirror
        - ThermalBoundaryNeumann
        - ThermalBoundaryRadiation
        - TimeManager
        - MeshReal
        - MeshVector
        - PiecewisePoly
        - Real
        - Table
        - Vector2i
        - Vector2r
        - Vector3i

Class List

Here are the classes, structs, unions and interfaces with brief de

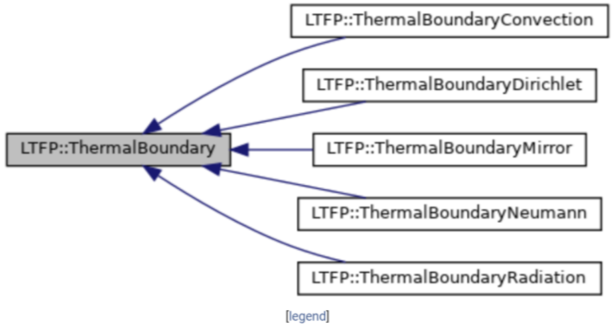
N LTFP	
C Exporter	Base class of exporters
C ExporterCsvMesh	Class that exports mes
C ExporterVtkMesh	Class that exports mes
C ExportManager	Class managing data e
C LaserSource	Class managing laser h
C MaterialProperty	Class managing materia properties
C MeshData	Class managing mesh and data
C SceneLoader	Class reads the scene file and store the configurations
C Simulator	Class managing simulator work flow
C Solver	Base class of solvers
C BoundaryManager	Class managing thermal boundary objects
C ThermalBoundary	Base class of thermal boundaries
C ThermalBoundaryConvection	Class of Convection thermal boundary
C ThermalBoundaryDirichlet	Class of Dirichlet thermal boundary
C ThermalBoundaryMirror	Class of Mirror thermal boundary
C ThermalBoundaryNeumann	Class of Dirichlet thermal boundary
C ThermalBoundaryRadiation	Class of Radiation thermal boundary
C TimeManager	Class managing current simulation time and time step size

LTFP::ThermalBoundary Class Reference

Base class of thermal boundaries. More...

`#include <ThermalBoundary.hpp>`

Inheritance diagram for LTFP::ThermalBoundary:



Public Member Functions

	ThermalBoundary (Config *config)
virtual	~ThermalBoundary ()
virtual Real	getTemp (const Vector3r &pos, const Real &temp) Get boundary temperature. More...
virtual Real	getFlux (const Vector3r &pos, const Real &temp) Compute the simple flux thought the boundary. More...
int	getIndex () const
BoundaryType	getType () const
BoundaryLocation	getLocation () const

Member Function Documentation

◆ getFlux()

Real LTFP::ThermalBoundary::getFlux (const Vector3r & pos,
const Real & temp
)

Compute the simple flux thought the boundary.

Parameters

- **pos** Position of the boundary neighboring cell
- **temp** Temperature of boundary neighboring cell

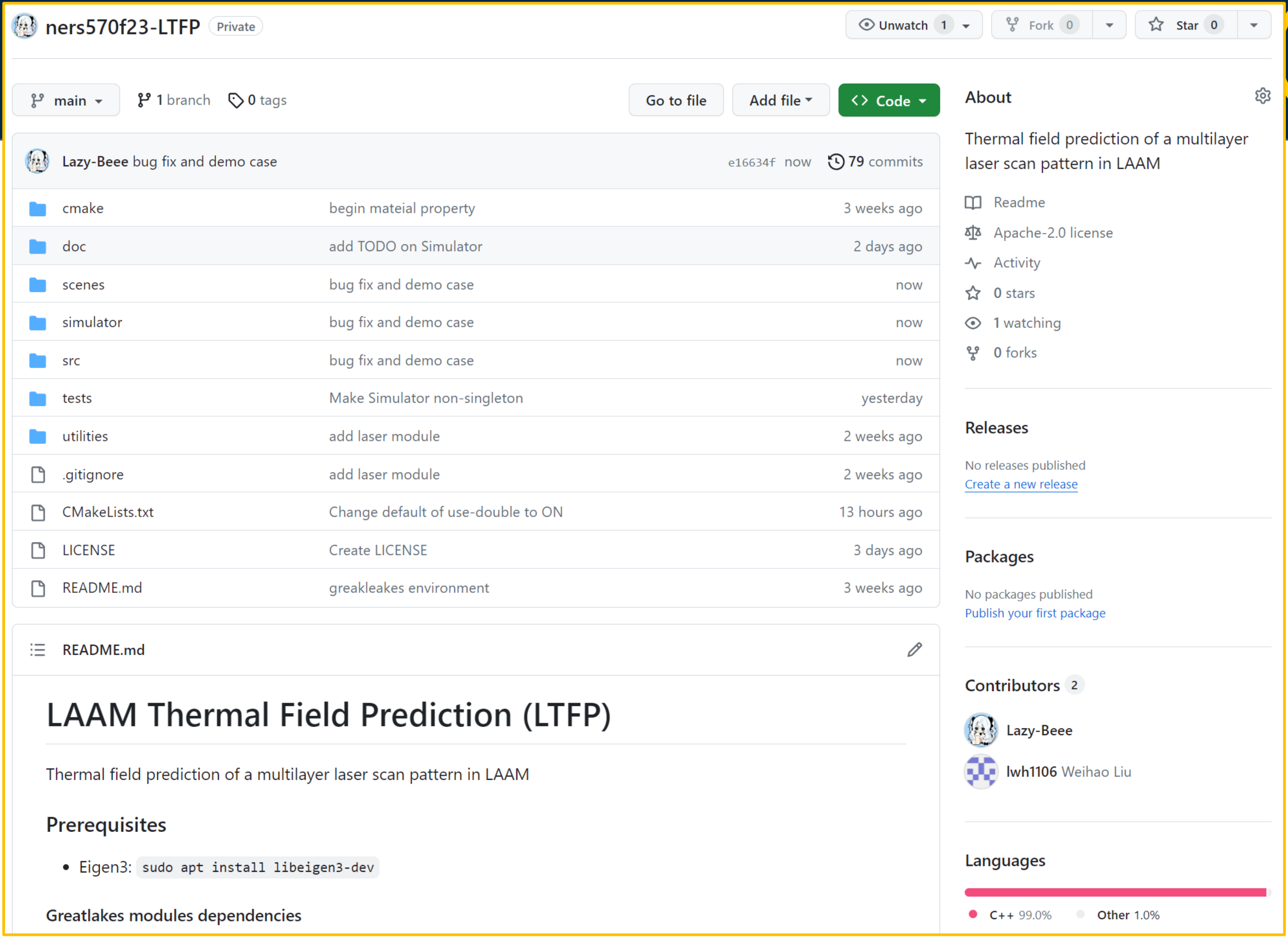
Returns

Flux though the boundary

Reimplemented in LTFP::ThermalBoundaryRadiation, LTFP::ThermalBou

GitHub

- Code is managed on **GitHub**



Summary

- Designed algorithm and completed coding of LTFP
  - Using the object-oriented programming language
  - Combined with parallelization
- Preliminarily tested the model
  - 1D case
  - Energy-conserved case
  - Functional test
- Generated git repo and documentation

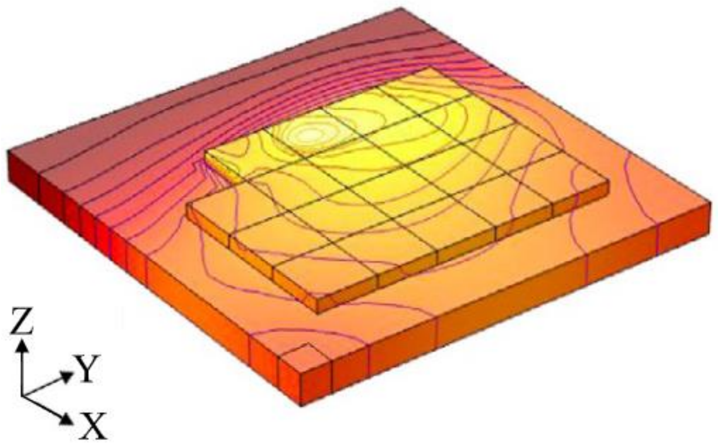
Future work

Work in progress

- ❖ Add more advanced and stable **solvers**: RK2, RK4, …
- ❖ Analysis and improve **parallelization** performance
- ❖ More **analysis** on generated thermal field and different scan patterns

Future work

- ❖ Validation using experiment data
- ❖ Integrate LTFP into the grain prediction model
- ❖ Uneven domain increment
- ❖ Z-ordered data storage



Ren, K., et al. "Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning." *Computer Methods in Applied Mechanics and Engineering* 362 (2020): 112734.