Improved Physics-Informed Neural Network for Thermal–Fluid Coupling in Czochralski Silicon Crystal Growth

1 Introduction

This document summarizes the governing equations, modeling assumptions, training methodology, and mathematical structure of the *improved physics-informed neural network (PINN)* proposed in:

> Research on the thermal–fluid coupling in the growth process of Czochralski silicon single crystals based on an improved physics-informed neural network,
> *AIP Advances*, 15, 105202 (2025).

The model targets steady, incompressible, thermally driven melt flow during Czochralski (CZ) silicon crystal growth and introduces spatial-information enhancement and adaptive loss balancing.

2 Governing Equations

The melt is modeled as a steady, incompressible Newtonian fluid under the Boussinesq approximation. A two-dimensional axisymmetric domain is assumed.

2.1 Continuity Equation

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

In Cartesian form (as used in the paper's 2D numerical implementation):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2}$$

2.2 Momentum Equations

The steady incompressible Navier–Stokes equations with buoyancy coupling are given by

$$\rho(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \rho \boldsymbol{g} \beta (T - T_0). \tag{3}$$

Component-wise:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \tag{4}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + g\beta(T - T_0). \tag{5}$$

2.3 Energy Equation

$$\rho c_p(\boldsymbol{u} \cdot \nabla T) = k\nabla^2 T \tag{6}$$

or equivalently,

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \tag{7}$$

where $\alpha = k/(\rho c_p)$.

3 Modeling Assumptions

- **Incompressibility**: density is constant except in the buoyancy term.
- **Boussinesq approximation**: thermal expansion drives buoyancy.
- **Laminar flow**: turbulence is neglected.
- **Steady-state**: time derivatives are omitted.
- **Axisymmetry**: azimuthal velocity is neglected.
- **Constant properties**: ν, k, c_p are constant.

4 Dimensionless Formulation

Let characteristic scales be:

$$L, \quad U, \quad \Delta T.$$

Define dimensionless variables:

$$x^* = \frac{x}{L}, \quad \boldsymbol{u}^* = \frac{\boldsymbol{u}}{U}, \quad T^* = \frac{T - T_0}{\Delta T}, \quad p^* = \frac{p}{\rho U^2}. \tag{8}$$

The dimensionless equations become:

4.1 Continuity

$$\nabla^* \cdot \boldsymbol{u}^* = 0 \tag{9}$$

4.2 Momentum

$$(\boldsymbol{u}^* \cdot \nabla^*)\boldsymbol{u}^* = -\nabla^* p^* + \frac{1}{\mathrm{Re}}\nabla^{*2}\boldsymbol{u}^* + \mathrm{Ra}\,\mathrm{Pr}^{-1}T^*\boldsymbol{e}_g \tag{10}$$

4.3 Energy

$$(\boldsymbol{u}^* \cdot \nabla^*)T^* = \frac{1}{\mathrm{Re}\,\mathrm{Pr}}\nabla^{*2}T^* \tag{11}$$

where

$$\mathrm{Re} = \frac{UL}{\nu}, \tag{12}$$

$$\mathrm{Pr} = \frac{\nu}{\alpha}, \tag{13}$$

$$\mathrm{Ra} = \frac{g\beta\Delta T L^3}{\nu\alpha}. \tag{14}$$

5 PINN Approximation

The neural network approximates:

$$\mathcal{N}_\theta(x, y) \to \{u, v, p, T\}. \tag{15}$$

Automatic differentiation is used to compute all spatial derivatives.

6 Improved PINN vs Standard PINN

6.1 Standard PINN

A standard PINN minimizes:

$$\mathcal{L}_{\text{std}} = \sum_i \|\mathcal{R}_i\|^2 + \sum_j \|\mathcal{B}_j\|^2, \tag{16}$$

with fixed loss weights and spatial coordinates only entering at the input layer.

6.2 Improved PINN (SI–LB PINN)

The improved PINN introduces:

Spatial Information Embedding

$$h_\ell = \sigma\left(W_\ell h_{\ell-1} + Z_\ell \odot M(x, y) + (1 - Z_\ell) \odot N(x, y)\right), \tag{17}$$

which preserves geometric information in deep layers.

Adaptive Loss Balancing

$$\mathcal{L} = \sum_m \left(e^{-\log \sigma_m^2} \mathcal{L}_m + \log \sigma_m\right), \tag{18}$$

where σ_m are trainable parameters.

This improves conditioning for multi-physics coupling.

7 Mapping Governing Equations to PyTorch Code

7.1 Continuity Residual

```
u_x = autograd.grad(u, x, grad_outputs=ones, create_graph=True)[0]
v_y = autograd.grad(v, y, grad_outputs=ones, create_graph=True)[0]
R_cont = u_x + v_y
```

7.2 Momentum Residual (x-direction)

```
u_xx = autograd.grad(u_x, x, grad_outputs=ones, create_graph=True)[0]
u_yy = autograd.grad(u_y, y, grad_outputs=ones, create_graph=True)[0]
p_x  = autograd.grad(p, x, grad_outputs=ones, create_graph=True)[0]

R_mom_x = u*u_x + v*u_y + p_x - nu*(u_xx + u_yy)
```

7.3 Momentum Residual (y-direction)

```
p_y = autograd.grad(p, y, grad_outputs=ones, create_graph=True)[0]

R_mom_y = u*v_x + v*v_y + p_y - nu*(v_xx + v_yy) - g*beta*(T-T0)
```

7.4 Energy Residual

```
T_xx = autograd.grad(T_x, x, grad_outputs=ones, create_graph=True)[0]
T_yy = autograd.grad(T_y, y, grad_outputs=ones, create_graph=True)[0]

R_energy = u*T_x + v*T_y - alpha*(T_xx + T_yy)
```

8 Training Process

- Collocation points sampled inside the domain
- Boundary condition points enforced via penalty loss
- Adam optimizer for training
- No labeled CFD or experimental data required
- COMSOL solutions used only for validation

9 Axisymmetric Cylindrical Coordinate Formulation

The Czochralski crystal growth process is inherently axisymmetric. We therefore adopt cylindrical coordinates (r, z, θ) and assume $\partial/\partial\theta = 0$ and $u_\theta = 0$.

The velocity field is:

$$\boldsymbol{u} = (u_r(r,z),\, u_z(r,z))$$

All governing equations include geometric source terms arising from the cylindrical coordinate system.

—

9.1 Continuity Equation

The incompressible continuity equation in axisymmetric coordinates is:

$$\frac{1}{r}\frac{\partial}{\partial r}(ru_r) + \frac{\partial u_z}{\partial z} = 0 \tag{19}$$

This term is critical and is a common source of error in nave PINN implementations.

—

9.2 Momentum Equations

9.2.1 Radial Momentum

$$u_r\frac{\partial u_r}{\partial r} + u_z\frac{\partial u_r}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial r} + \nu\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_r}{\partial r}\right) + \frac{\partial^2 u_r}{\partial z^2} - \frac{u_r}{r^2}\right) \tag{20}$$

—

9.2.2 Axial Momentum

$$u_r \frac{\partial u_z}{\partial r} + u_z \frac{\partial u_z}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \nu \left(\frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial u_z}{\partial r} \right) + \frac{\partial^2 u_z}{\partial z^2} \right) + g\beta(T - T_0) \tag{21}$$

—

9.3 Energy Equation

The heat transport equation in axisymmetric form is:

$$u_r \frac{\partial T}{\partial r} + u_z \frac{\partial T}{\partial z} = \alpha \left(\frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial T}{\partial r} \right) + \frac{\partial^2 T}{\partial z^2} \right) \tag{22}$$

—

10 Dimensionless Axisymmetric Form

Using characteristic scales $(L, U, \Delta T)$:

$$r^* = \frac{r}{L}, \quad z^* = \frac{z}{L}, \quad u^* = \frac{u}{U}, \quad T^* = \frac{T - T_0}{\Delta T}$$

The dimensionless equations become:

—

10.1 Continuity

$$\frac{1}{r^*}\frac{\partial}{\partial r^*}(r^* u_r^*) + \frac{\partial u_z^*}{\partial z^*} = 0 \tag{23}$$

—

10.2 Momentum

$$u_r^* \partial_{r^*} u_r^* + u_z^* \partial_{z^*} u_r^* = -\partial_{r^*} p^* + \frac{1}{\mathrm{Re}} \left(\frac{1}{r^*} \partial_{r^*}(r^* \partial_{r^*} u_r^*) + \partial_{z^* z^*} u_r^* - \frac{u_r^*}{r^{*2}} \right), \tag{24}$$

$$u_r^* \partial_{r^*} u_z^* + u_z^* \partial_{z^*} u_z^* = -\partial_{z^*} p^* + \frac{1}{\mathrm{Re}} \left(\frac{1}{r^*} \partial_{r^*}(r^* \partial_{r^*} u_z^*) + \partial_{z^* z^*} u_z^* \right) + \frac{\mathrm{Ra}}{\mathrm{Re}\,\mathrm{Pr}} T^* \tag{25}$$

—

10.3 Energy

$$u_r^* \partial_{r^*} T^* + u_z^* \partial_{z^*} T^* = \frac{1}{\mathrm{Re}\,\mathrm{Pr}} \left(\frac{1}{r^*} \partial_{r^*}(r^* \partial_{r^*} T^*) + \partial_{z^* z^*} T^* \right) \tag{26}$$

—

11 Weak Formulation (COMSOL-Consistent)

COMSOL solves the **weak (variational) form** of the governing equations. Let v_r, v_z, q, w be admissible test functions.

—

11.1 Continuity Weak Form

$$\int_\Omega \left(\frac{1}{r}\frac{\partial}{\partial r}(r u_r) + \frac{\partial u_z}{\partial z} \right) q\, r\, dr\, dz = 0 \tag{27}$$

Note the **Jacobian factor** r, which is essential in axisymmetric COMSOL models.

—

11.2 Radial Momentum Weak Form

$$\int_\Omega \Big[(u_r \partial_r u_r + u_z \partial_z u_r) v_r - p \partial_r v_r + \nu(\nabla u_r \cdot \nabla v_r) - \nu \frac{u_r}{r^2} v_r \Big]\, r\, dr\, dz = 0 \tag{28}$$

—

11.3 Axial Momentum Weak Form

$$\int_\Omega \left[(u_r \partial_r u_z + u_z \partial_z u_z) v_z - p \partial_z v_z + \nu(\nabla u_z \cdot \nabla v_z) + g\beta(T - T_0) v_z \right] r\, dr\, dz = 0 \tag{29}$$

—

11.4 Energy Weak Form

$$\int_\Omega \left[(u_r \partial_r T + u_z \partial_z T) w + \alpha(\nabla T \cdot \nabla w) \right] r\, dr\, dz = 0 \tag{30}$$

—

12 PINN Residuals vs COMSOL Weak Residuals

Key Correspondence

- **PINN**: Enforces *strong-form residuals* at collocation points
- **COMSOL**: Enforces *weak-form residuals* over finite elements

Mathematically:

$$\mathcal{R}_{\text{PINN}}(x_i) \approx 0 \quad \Longleftrightarrow \quad \int_\Omega \mathcal{R}_{\text{weak}}\, d\Omega = 0$$

XPINN partially bridges this gap by enforcing local subdomain consistency, which mimics element-wise weak enforcement.

—

13 Mapping to PyTorch Residuals (Axisymmetric)

```
# Continuity
R_cont = (u_r + r*u_r_r)/r + u_z_z

# Radial momentum
R_ur = u_r*u_r_r + u_z*u_r_z + p_r \
       - nu*((u_r_r + r*u_r_rr)/r + u_r_zz - u_r/r**2)
```

```
# Axial momentum
R_uz = u_r*u_z_r + u_z*u_z_z + p_z \
       - nu*((u_z_r + r*u_z_rr)/r + u_z_zz) \
       - g*beta*(T - T0)

# Energy
R_T = u_r*T_r + u_z*T_z \
      - alpha*((T_r + r*T_rr)/r + T_zz)
```

—

14 Remarks

- Axisymmetric geometric terms are essential for physical accuracy
- COMSOL weak-form includes the Jacobian factor r
- PINNs approximate the strong form but converge to the same solution
- XPINNs improve conditioning by mimicking FEM locality

15 Conclusion

The improved SI–LB PINN provides a mathematically well-conditioned framework for solving coupled thermal–fluid PDEs in Czochralski crystal growth, significantly improving convergence and accuracy over standard PINNs.