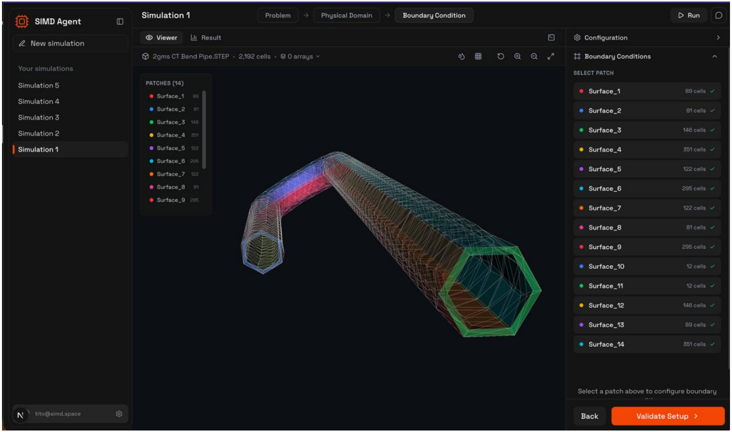
physics-AI For CFD Simulations: SciML

January 26, 2026

Outline

1. Adapted Physics-Informed Neural Networks for Crystal Growth
2. Modeling of Cryogenic Systems-H2, N2
3. Dyad Agent- Jüiahüb
   - Cryogenic H2/N2
   - Dyad agent
   - Czochralski Crystal Growth Process
   - Modeling Assumptions and Domain
   - Governing Equations
   - Dimensionless Analysis
   - Improved PINN Methodology
   - Equation-to-Loss Function Mapping
   - Training and Validation


SIMD Agent
New simulation
Your simulations
Simulation 5
Simulation 4
Simulation 3
Simulation 2
Simulation 1
tito@simd.space
Simulation 1
Problem
Physical Domain
Boundary Condition
Run
Viewer
Result
2gms CT Bend Pipe.STEP · 2,192 cells · 0 arrays
PATCHES (14)
Surface_1
Surface_2
Surface_3
Surface_4
Surface_5
Surface_6
Surface_7
Surface_8
Surface_9
Configuration
Boundary Conditions
SELECT PATCH
Surface_1 89 cells
Surface_2 81 cells
Surface_3 146 cells
Surface_4 351 cells
Surface_5 122 cells
Surface_6 295 cells
Surface_7 122 cells
Surface_8 81 cells
Surface_9 295 cells
Surface_10 12 cells
Surface_11 12 cells
Surface_12 146 cells
Surface_13 89 cells
Surface_14 351 cells
Select a patch above to configure boundary
Back
Validate Setup

Dyad Search Ctrl K Installation Getting Started Tutorials Manual Analysis Libraries Component Libraries dev •••

Dyad

Hardware Engineering at the Speed of Software

Get Started | Get Dyad Studio | Get Help

Dyad is the modeling system for next-generation engineers.
Dyad combines the latest methods for differentiable programming, scientific machine learning, and generative AI with the techniques of traditional modeling environments, e.g. acausal physical models, controls analysis, and embedded code generation.
Using a single-artifact philosophy ensures that the same file can be used for both developing models in a GUI and within textual environments, such as VS Code. Modern design principles of Dyad include a package manager for sharing composable models, test-driven development CI/CD integration, and deployment to real-world hardware.

What is the Czochralski (CZ) Process?

- Dominant method for single-crystal silicon growth
- Used in semiconductor and photovoltaic industries

CZ Furnace Schematic

CZ Furnace: Crucible, Melt, Heater, Crystal Pulling

Crystal Growth Physics

- Solidification at melt–crystal interface
- Strong heat extraction through crystal
- Melt flow affects dopant and defect transport

Why Modeling is Difficult

- Strong thermal–fluid coupling
- High Rayleigh number convection
- Thin boundary layers
- Multi-physics and multi-scale nature

Axisymmetric Approximation

- Nearly rotationally symmetric geometry
- Reduces 3D problem to 2D (r, z)

Computational Domain

Axisymmetric Melt Domain in (r, z)

Boundary Identification

Crucible Wall, Free Surface, Crystal Interface

Physical Assumptions

- Incompressible Newtonian melt
- Laminar, steady-state flow
- Constant material properties
- Boussinesq approximation

Continuity Equation

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

Radial Momentum Equation

$$u_r \partial_r u_r + u_z \partial_z u_r = -\frac{1}{\rho} \partial_r p + \nu \left(\nabla^2 u_r - \frac{u_r}{r^2} \right)$$

Axial Momentum Equation

$$u_r \partial_r u_z + u_z \partial_z u_z = -\frac{1}{\rho} \partial_z p + \nu \nabla^2 u_z + g \beta (T - T_0)$$

Energy Equation

$$u_r \partial_r T + u_z \partial_z T = \alpha \nabla^2 T$$

Thermal–Fluid Coupling

- Temperature $\rightarrow$ buoyancy force
- Velocity $\rightarrow$ heat advection
- Strong nonlinear feedback loop

Rayleigh Number

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

- Measures strength of natural convection

High Rayleigh Number Regime

- Thin thermal and velocity boundary layers
- Strong nonlinear advection
- Major source of instability for PINNs

Standard PINN Concept

- Neural network approximates PDE solution
- Physics enforced via residual minimization

PINN Inputs and Outputs

- Inputs: (r, z)
- Outputs: u_r, u_z, p, T

Spatial Information (SI) Embedding

- Coordinates injected into hidden layers
- Preserves geometric sensitivity
- Improves boundary-layer resolution

Adaptive Loss Balancing

- Separate losses for each equation
- Trainable weights
- Prevents gradient domination

PINN Loss Function Structure

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_m \mathcal{L}_m + \lambda_e \mathcal{L}_e + \lambda_b \mathcal{L}_b$$

Continuity Loss

$$\mathcal{L}_c = \frac{1}{N} \sum_i \left| \frac{1}{r} \partial_r (r u_r) + \partial_z u_z \right|^2$$

Momentum Loss

$$\mathcal{L}_m = \left\|\mathcal{R}_{u_r}\right\|^2 + \left\|\mathcal{R}_{u_z}\right\|^2$$

- Enforces Navier–Stokes equations

Energy Loss

$$\mathcal{L}_e = \frac{1}{N}\sum_i \left| u_r \partial_r T + u_z \partial_z T - \alpha \nabla^2 T \right|^2$$

Boundary Condition Loss

$$\mathcal{L}_b = \|u - u_{BC}\|^2 + \|T - T_{BC}\|^2$$

Why Adaptive Weighting Matters

- Different PDEs have different stiffness
- High-Ra momentum residuals dominate gradients
- Learned weights improve conditioning

Training Data

- No experimental or CFD data
- Physics-only collocation points

Reference Solution

- COMSOL Multiphysics
- Weak-form finite element solver

Velocity Comparison

COMSOL vs PINN Velocity Contours

Temperature Comparison

COMSOL vs PINN Temperature Contours

Error Metrics

- Relative L_2 norms
- Improved PINN outperforms standard PINN

Work

- Transient growth
- Rotation and Marangoni effects
- Real-time process optimization

Thank You