

# High-Resolution 3D Thermal Modeling with Fortran Solver and Machine Learning Surrogate

Tommer Weizmann

*University of York, Heslington, York, YO10 5DD, , United Kingdom*

---

## Abstract

This study introduces an open, modular framework for simulating the inhomogeneous 3D heat equation with a laser source term under general boundary conditions. The solver, implemented in Fortran for computational efficiency, models realistic Gaussian laser pulses as volumetric heat sources in cylindrical coordinates. Validation against analytical solutions confirms energy conservation, numerical stability, and first-order convergence Evans (2010).

To enable rapid exploration of high-dimensional parameter spaces, we develop an XGBoost-based surrogate model trained on simulation outputs with automated hyperparameter optimization. The surrogate achieves high accuracy on test data ( $R^2 = 0.994$ , RMSE = 134 K), and shows reasonable generalization to three of five previously unseen material types (transition metals, alkali metals, and alloys), where the relative error remained at or below  $\sim 30\%$ .

The complete workflow—including the solver, surrogate framework, and datasets—is openly released to facilitate reproducibility and extension to other geometries and multi-physics problems. By combining high-fidelity simulations with efficient surrogate modeling, this work provides a practical computational tool for high-resolution thermal transport studies, particularly suited for large-scale parameter sweeps and materials optimization where physics-informed neural network approaches may be computationally prohibitive.

*Keywords:* Heat transfer, Surrogate modeling, XGBoost, Finite difference methods, Gaussian laser pulses

---

## 1. Introduction

Partial differential equations (PDEs) underpin many physical and engineering phenomena, including heat conduction, fluid flow, wave propagation, and diffusion. Although numerical methods for the heat equation are well established, temperature predictions at

laboratory scales often rely on simplified, uniform, or one-dimensional formulations Tsega (2022); Dehghan (2004); Evans (2010). Analytical solutions exist Evans (2010), but they become impractical for realistic three-dimensional, nonuniform systems, creating a gap between tractable academic examples and the multiphysics scenarios encountered in practice. This limitation is particularly critical in high-resolution thermal transport studies, such as laser–material interactions, thermal management design, and materials processing, where accurate temperature predictions are required across a wide parameter space. A central challenge in applying machine learning to PDE-based problems is the generation of reliable high-resolution datasets. Physics-informed neural networks (PINNs) can solve PDEs without explicit data, but they are computationally demanding and often restricted in resolution or parameter coverage, limiting their use in large-scale parameter sweeps. An alternative is to generate data directly from efficient PDE solvers and train regression models as surrogates. In this work, a surrogate is developed to predict the maximum temperature from simulation parameters, offering rapid and accurate predictions, with the option of extending to full-field approximations in later studies. The solver, implemented in Fortran for computational efficiency, supports general boundary conditions, spatially varying material properties (thermal conductivity, density, heat capacity, absorption coefficient), and realistic volumetric sources. Demonstrated here with a Gaussian laser pulse in cylindrical coordinates, the solver modular design enables a straightforward extension to other geometries. High-resolution data sets ( $100 \times 100$  grids) are generated efficiently, since the maximum temperature is reached on short timescales, reducing the simulation cost compared to long-time PDE integration. Trained in these datasets, an XGBoost-based surrogate with automated hyperparameter optimization achieves high accuracy across a broad parameter range. XGBoost was chosen due to its high accuracy and efficient training time, as demonstrated in various previous studies Bentéjac et al. (2019); M’hamdi et al. (2024), as it often outperforms alternative methods. Compared with direct PDE solves or PINNs, the approach balances computational efficiency with resolution and predictive accuracy, making it well suited for parameter sweeps, experimental design, and optimization studies in thermal transport. Key contributions include: (i) efficient generation of high-resolution data for three-dimensional heat transport, (ii) a modular solver that can accommodate different geometries and source terms, and (iii) a reproducible workflow for building surrogate models capable of predicting maximum temperature.

## 2. Theory

### 2.1. Numerical Method

The transient heat equation is solved numerically in cylindrical coordinates using an explicit finite-difference scheme (FTCS) Smith (1985); Patankar (1980); Tsega (2022):

$$T_{r_i, z_k}^{n+1} = T_{r_i, z_k}^n + \Delta t \left( \alpha \nabla^2 T_{r_i, z_k}^n + f(r_i, z_k, t_n) \right), \quad (1)$$

where the source term models the Gaussian laser pulse as a volumetric heat source Pantawane et al. (2020); Li et al. (2023); Jalili et al. (2024):

$$f(r_i, z_k, t_n) = \frac{\mu_a}{\rho c} I_0 \exp\left(-\frac{r_i^2}{w^2}\right) \exp\left(-\frac{z_k}{\delta}\right) \exp\left(-\frac{(t_n - t_0)^2}{2\sigma^2}\right), \quad (2)$$

with standard SI units for absorption coefficient  $\mu_a$ , density  $\rho$ , specific heat  $c$ , laser intensity  $I_0$ , beam radius  $w$ , optical penetration depth  $\delta$ , simulation time  $t$ , pulse center  $t_0$ , and pulse duration  $\sigma$ .

Convective cooling on the top surface ( $z = 0$ ) is incorporated as Patankar (1980); Smith (1985):

$$T_{i,k}^{n+1} = T_{i,k}^n + \Delta t \left( \alpha \nabla^2 T_{i,k}^n + f(r_i, z_k, t_n) - \frac{h}{\rho c \Delta z} (T_{i,k}^n - T_\infty) \chi_{z=0} \right), \quad (3)$$

where  $h$  is the convective heat transfer coefficient,  $T_\infty$  the ambient temperature, and  $\chi_{z=0}$  an indicator function for the top surface.

For axisymmetric problems, the Laplacian in cylindrical coordinates is given by Evans (2010); Tsega (2022):

$$\nabla^2 T = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial T}{\partial r} \right) + \frac{\partial^2 T}{\partial z^2}, \quad (4)$$

with the singularity at  $r = 0$  handled via

$$\left. \frac{\partial T}{\partial r} \right|_{r=0} = 0 \quad \Rightarrow \quad \left. \frac{\partial^2 T}{\partial r^2} \right|_{r=0} \approx \frac{2(T_2 - T_1)}{(\Delta r)^2} Tsega (2022). \quad (5)$$

For internal nodes ( $i = 2, \dots, n_r - 1$ ):

$$\nabla^2 T_{i,k} \approx \frac{T_{i+1,k} - 2T_{i,k} + T_{i-1,k}}{(\Delta r)^2} + \frac{1}{r_i} \frac{T_{i+1,k} - T_{i-1,k}}{2\Delta r} + \frac{T_{i,k+1} - 2T_{i,k} + T_{i,k-1}}{(\Delta z)^2} Tsega (2022); Dehghan (2004). \quad (6)$$

Boundary conditions include:

- Neumann (insulated/symmetric):  $\frac{\partial T}{\partial \mathbf{n}} = 0$  Smith (1985),
- Convective (top surface  $z = 0$ ):  $-k \frac{\partial T}{\partial z} = h(T - T_\infty)$  Patankar (1980).

The FTCS scheme satisfies the stability condition Konangi et al. (2018); Smith (1985):

$$\alpha \Delta t \left( \frac{1}{\Delta r^2} + \frac{1}{\Delta z^2} \right) \leq \frac{1}{2} \quad \Rightarrow \quad \Delta t \leq \frac{\Delta h^2}{4\alpha} \quad (\text{if } \Delta r = \Delta z = \Delta h). \quad (7)$$

Grid convergence is assessed by comparing the current solution to a reference finest-grid solution.

$$\varepsilon = |T_{\text{ref}} - T_{\text{current}}|, \quad \varepsilon \propto h^p \Rightarrow \ln(\varepsilon) = p \ln(h) + C, \quad (8)$$

with  $p$  giving the convergence rate Shen et al. (2020).

## 2.2. Machine Learning Approach

Tree-based models learn by recursively splitting the input space based on feature values to capture non-linear patterns and interactions Breiman (2001); Shahbazi and Azadeh-Fard (2025). They are robust to outliers, require minimal preprocessing, and excel in scenarios with limited or noisy data Shahbazi and Azadeh-Fard (2025).

The chosen model is XGBoost (Extreme Gradient Boosting) Chen and Guestrin (2016), which sequentially builds a set of decision trees. Each tree corrects the errors of its predecessor by optimizing a differentiable loss function via gradient descent. Regularization prevents overfitting and parallelization improves computational efficiency Shahbazi and Azadeh-Fard (2025). In this study, the coarsegrid solver output is used to train XGBoost to capture physical trends and average behaviors, rather than exact numerical values, allowing fast and reliable predictions across multiple materials.

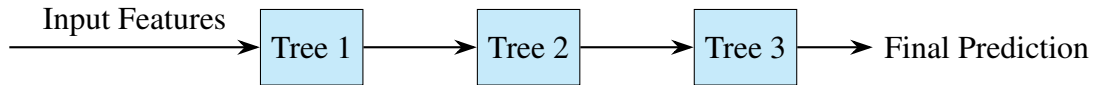


Figure 1: Schematic of a tree-based machine learning ensemble. Each tree independently processes the input features, and the final prediction is obtained by averaging the outputs of all trees.

## 3. Solver Implementation

Simulations were implemented in Fortran90 (gfortran, VSCode) using a modular framework that allows independent modification of geometry, material properties, and boundary conditions Patankar (1980); Smith (1985). The material properties—density

( $\rho$ ), specific heat ( $c$ ), thermal conductivity ( $k$ ) and extinction coefficient ( $\epsilon$ )—were compiled for 31 metals, providing a diverse data set (see Appendix Appendix B) Dehghan (2004).

Neumann boundary conditions (zero flux) were applied to insulated or symmetric models surfaces Smith (1985), while convective losses were considered on the irradiated surface Patankar (1980).

The transient heat equation was solved in cylindrical coordinates using an explicit FTCS scheme Smith (1985); Konangi et al. (2018). A discretization of  $100 \times 100$  steps in the  $r$ - $z$  domain was used, with uniform spacing  $\Delta r = \Delta z = h = 0.01$  m. The time step was set to  $\Delta t = h^2/(6\alpha)$  s. Simulations were run for 100 s during initial testing to ensure that all materials, including slow diffusers, reached their maximum temperature and exhibited full diffusion behavior.

The solver outputs the temperature at the irradiated center ( $r = 0, z = 0$ ), which serves as the scalar value for prediction. The general workflow for data generation, including test loops, is illustrated in Fig. 2. For reproducibility, all variables required to rerun the solver are listed in the appendix Appendix C.

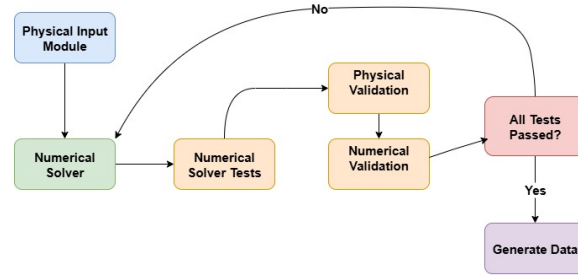


Figure 2: Workflow of the numerical solver, illustrating iterative testing and verification steps.

#### 4. Physical and Numerical Validation

The computational model was validated against the analytical solution for Aluminum ( $k = 237 \text{ W m}^{-1} \text{ K}^{-1}$ ,  $\rho = 2700 \text{ kg m}^{-3}$ ,  $c = 897 \text{ J kg}^{-1} \text{ K}^{-1}$ ). Simulations were conducted for 50 s on an  $80 \times 80$  spatial grid to assess time-step stability and grid convergence Smith (1985); Konangi et al. (2018), with a  $100 \times 100$  grid used as the reference solution for the evaluation of errors Shen et al. (2020). Energy residuals, defined as the difference between input, stored, and lost energy, confirm conservation of energy to numerical precision Patankar (1980).

Comparison with the analytical solution from Evans (2010), which assumes an infinite

3D domain, shows good agreement when adjusted for the finite computational domain size. To account for the fact that the analytical solution distributes heat over an unbounded space, we compared it to a reduced effective domain (0.004 m vs. 0.01 m computational domain). This scaling aligns peak magnitudes but does not affect the temporal evolution, which remains consistent. Agreement is particularly strong in the early-time regime before boundary effects in the finite computational domain become significant. Energy residuals peak during the initial laser pulse and decay to zero within 0.3 s (Figures 3–4, Table 1).

Table 1: Errors are reported relative to the analytical solution. The table shows peak error, RMSE, MAE, and peak temperatures and times. Note that the earlier peak observed in the computational model (0.010 s vs. 0.060 s analytically).

Metric	Value
Peak error	0.23 %
RMSE	$1.37 \times 10^2$ K
MAE	$6.65 \times 10^1$ K
Analytical peak time	0.060 s
Simulation peak time	0.010 s
Peak temperature	$1.74 \times 10^3$ K

The earlier peak observed in the computational model (0.010 s vs. 0.060 s analytically) is due to the analytical solution being based on the Green’s function for an infinite domain, which distributes heat more broadly and delays the temperature rise compared to the localized energy deposition in the finite computational model.

The residual energy was calculated as

$$\text{Residual} = E_{\text{input}} - E_{\text{stored}} - E_{\text{loss}},$$

where

$$E_{\text{stored}} = \sum \rho c T \Delta V, \quad E_{\text{input}} = P \sigma, \quad E_{\text{loss}} = hA(T - T_{\infty})\Delta t.$$

Here,  $\Delta V = 2\pi r \Delta r \Delta z$  represents the control volume and  $A = \pi(w^2 + 2z^2)$  is the relevant surface area.

Time-step refinement tests demonstrate that smaller  $\Delta t$  values reduce peak temperature error and improve stability (Figure 6). Grid refinement confirms convergence: the absolute error  $\varepsilon$  decreases with decreasing step size  $h$ , showing an initial convergence rate of approximately 1.5 for coarse grids and stabilizing near 1 for finer grids, consistent with first-order theory (Smith (1985) (Figure 5)).

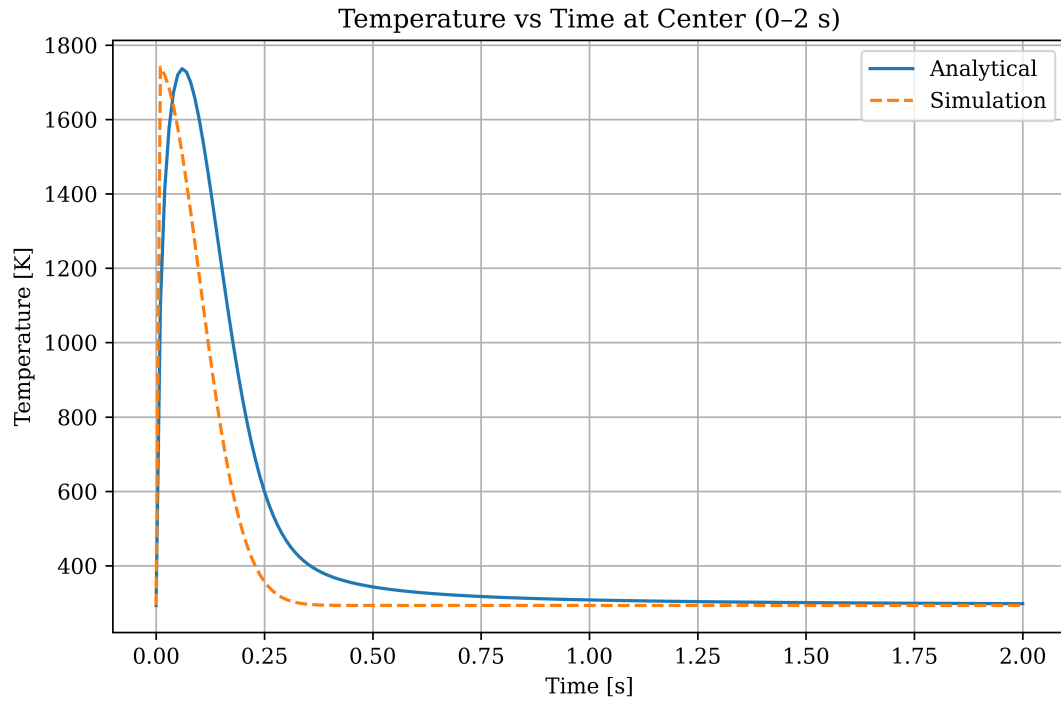


Figure 3: Comparison of the computational model with the analytical solution. Peak magnitudes are aligned by using a reduced effective analytical domain (0.004 m vs. 0.01 m computational domain). Good agreement is observed in the early-time regime, confirming that the simulation reproduces the expected trend from the analytical solution. The earlier peak in the simulation arises from the Green's function nature of the analytical solution.

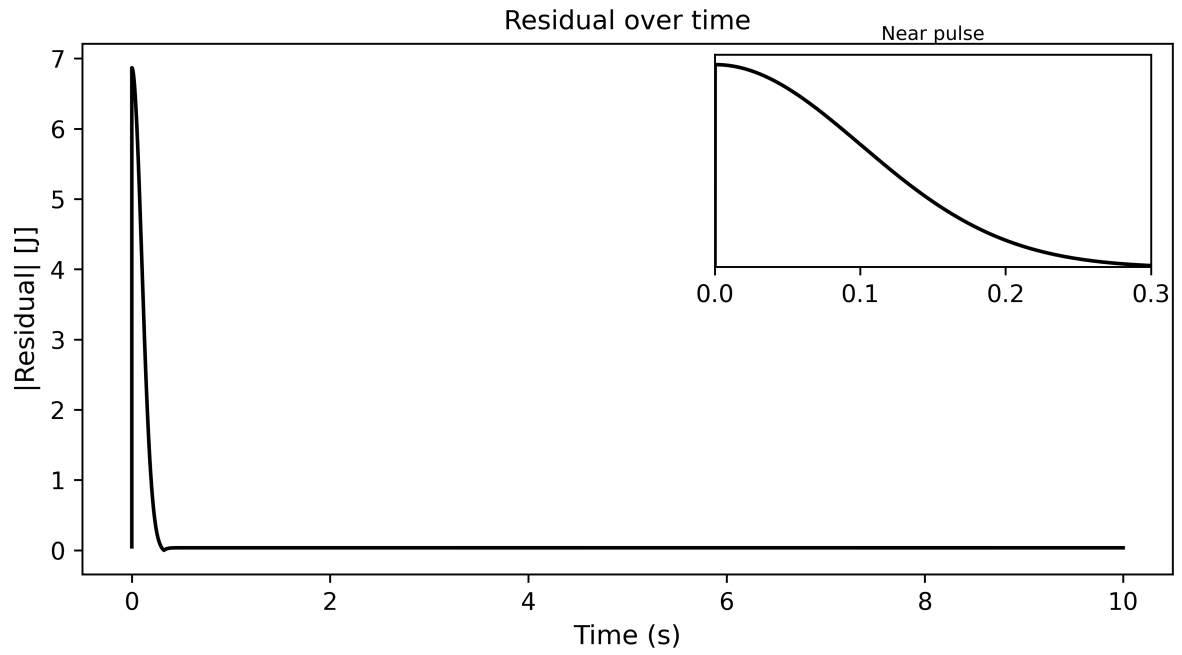


Figure 4: Time evolution of the energy residual, calculated as the difference between input, stored, and lost energy. The residual peaks during the initial laser pulse and decays smoothly to zero within 0.3 s, confirming conservation of energy to numerical precision. The inset in the top right shows a zoomed view around the pulse, highlighting the gradual reduction of residuals toward zero as expected from physical theory by the end of the simulation period.



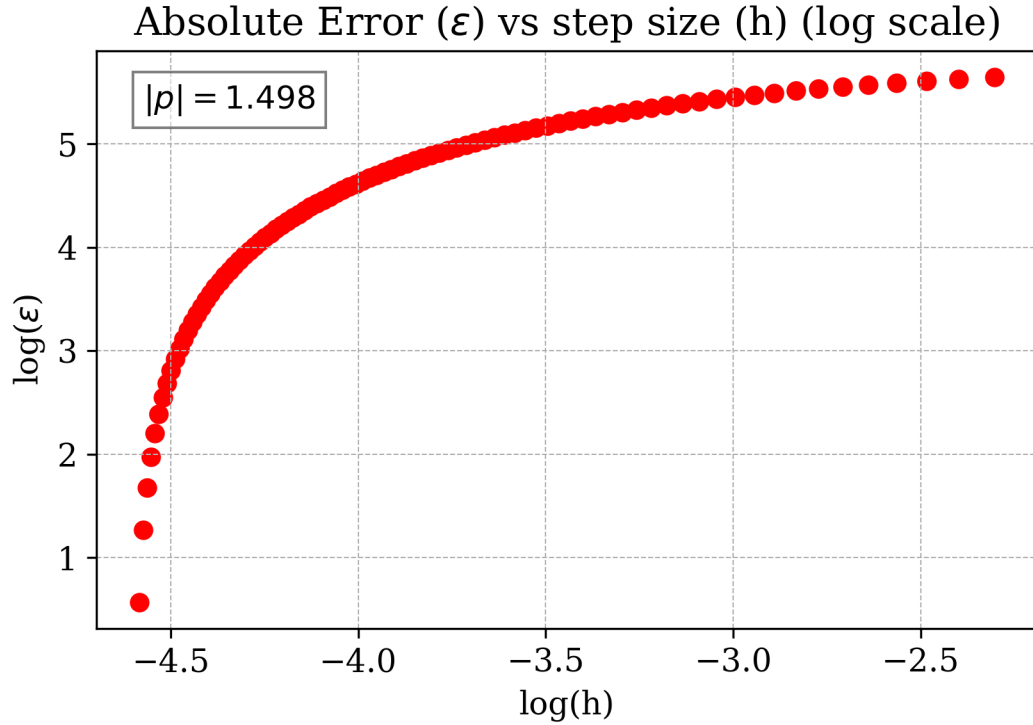
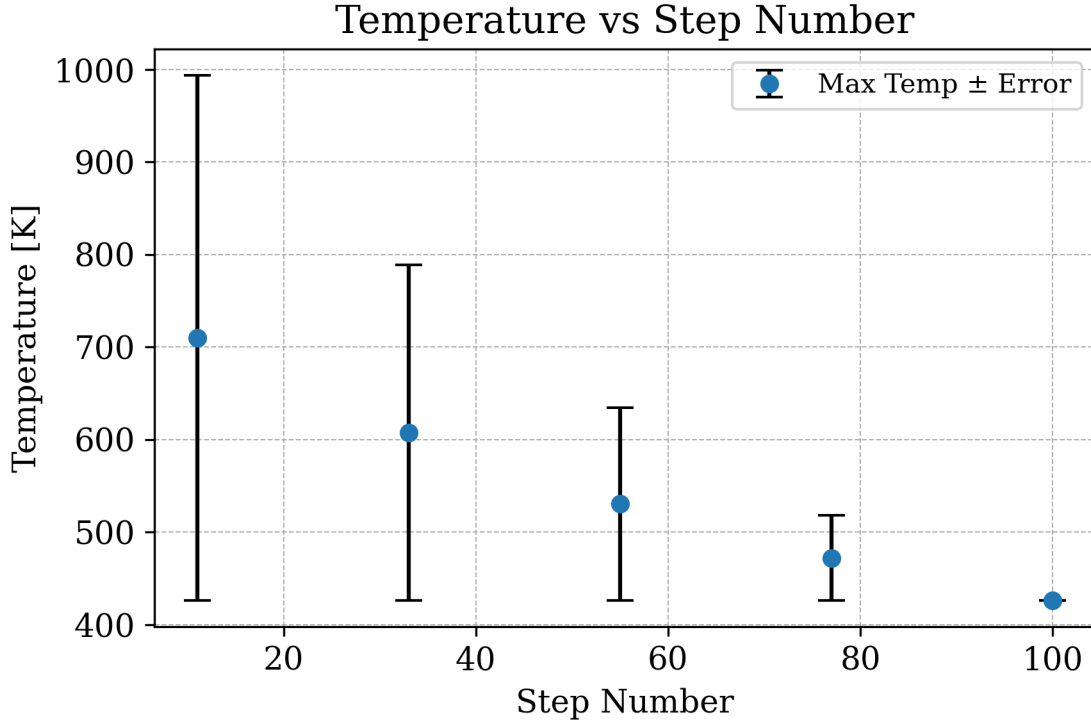


Figure 5: Top plot: Maximum temperature as a function of step number. The computed maximum temperature decreases as the step number increases, indicating that coarser grids overestimate peak values. As the number of steps per spatial interval increases, numerical accuracy improves and errors decrease. Bottom plot: Absolute error as a function of spatial grid size  $h = \frac{\text{length}}{\text{step\_number}-1}$ . Coarse grids show an initial convergence rate of approximately 1.5, while finer grids stabilize near 1, consistent with first-order theory. This demonstrates the effect of spatial grid refinement on the accuracy of the computational model.

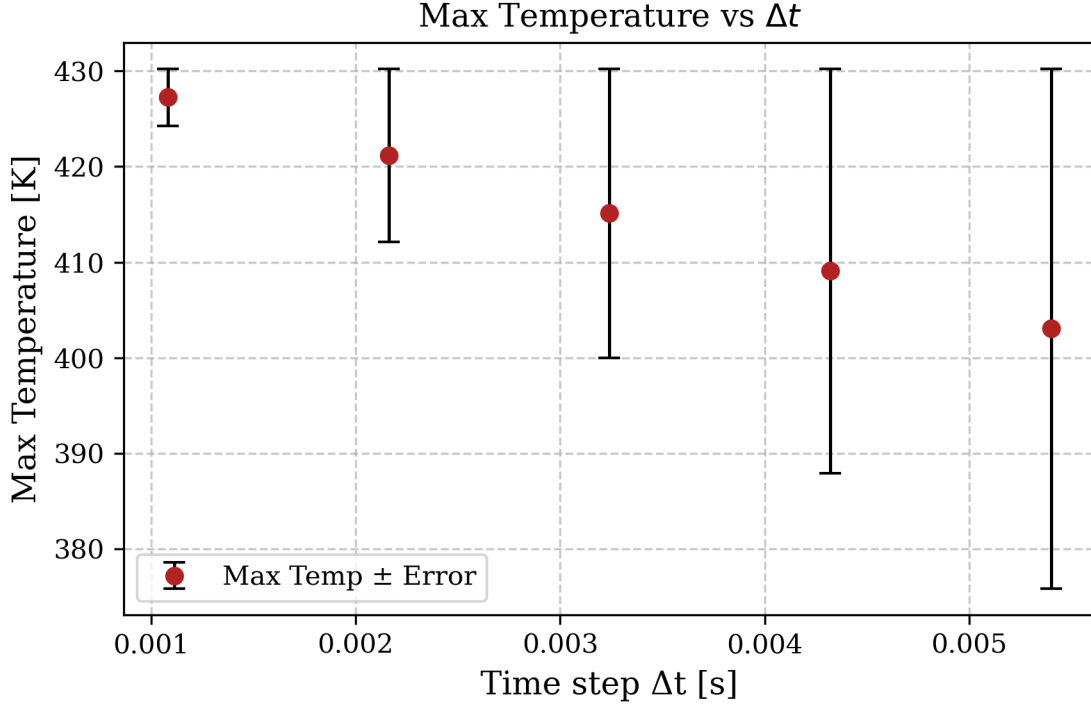


Figure 6: Peak temperature and absolute error as a function of time-step size. Smaller  $\Delta t$  values reduce the peak temperature error and improve numerical stability, confirming time-step refinement behavior.

## 5. XGBoost Modeling, Evaluation, and Generalization

An XGBoost regression model was developed to predict the maximum temperature resulting from laser–material interactions. The input features included both physical and engineered variables. Physical features consisted of laser power, wavelength, beam radius, pulse duration, and material thermal properties (thermal conductivity, density, specific heat capacity, and emissivity). To account for the large variation in magnitude, beam radius and pulse duration were transformed using base-10 logarithms.

Engineered features were introduced to represent energy deposition characteristics. These included the laser intensity,  $\log(\sum \text{source\_term})$ ,  $\log(\max(\text{source\_term}))$ , and  $\log(\|\text{source\_term}\|_2)$ , where  $\|\text{source\_term}\|_2$  denotes the L2 norm of the source term array. Logarithmic transformations ensured stable learning across quantities that differed by several orders of magnitude. This enabled the model to capture nonlinear dependencies while reducing the need for interpolation across widely differing scales.

The dataset was generated from simulations on a  $100 \times 100$  spatial grid, with temperatures restricted to the range 293.15–10,000 K. Laser parameters were systematically varied as follows: beam radius  $1 \times 10^{-6}$ – $1 \times 10^{-2}$  m, power 0.1–0.5 W, wavelength 400–700 nm, and pulse duration  $1 \times 10^{-6}$ – $1 \times 10^{-2}$  s. All metals listed in Appendix B.3 were included in the training set to ensure broad material coverage.

Hyperparameters were optimized using 5-fold randomized cross-validation with early stopping. The best configuration minimized mean RMSE with low variance (Table 2). The learning curve (Figure 7) shows RMSE convergence and the point at which additional boosting iterations no longer improve performance, confirming the effectiveness of early stopping.

Feature importance analysis consistently identified thermal conductivity and wavelength as the most influential predictors. Overall, the model captured physically meaningful relationships across all features, as illustrated in Figure 9. This highlights the dominant roles of energy deposition geometry and material transport properties in determining maximum temperature.

Prediction errors exhibited only weak scaling with temperature ( $T \propto \text{Error}^r$ , where  $r < 1$  in most cases). Predicted versus true temperatures (Figure 8) demonstrated strong agreement across the dataset, with larger deviations occurring primarily at higher temperatures ( $T \geq 6000$  K). Residual analysis (Figure 10) confirmed that errors remained centered near zero for most of the temperature range, increasing beyond 6000 K, consistent with the trends observed in Figure 8. On the independent test set, the model achieved RMSE = 134 K, MAE = 44 K, MAPE = 3.1%, and  $R^2 = 0.994$ , indicating excellent generalization within the tested parameter space.

To test generalization, new materials not included in the training set were evaluated. The materials were grouped by type, with three new metals per type: transition metals (Vanadium, Zirconium, Hafnium), poor metals (Gallium, Indium, Thallium), alloys (Inconel, Duralumin, Constantan), alkali metals (Rubidium, Cesium, Francium), and actinides (Thorium, Plutonium, Neptunium). Their material properties and types are listed in Appendix F.5. The model’s performance on these new materials, quantified by Mean Absolute Percentage Error (MAPE), is shown in Figure 11. The model performs reasonably well for transition metals, alkali metals, and alloys, with MAPE values within acceptable limits for practical applications. Higher errors were observed for poor metals and actinides, reflecting the challenges in extrapolating to materials with properties significantly different from the training set.

All modeling was implemented using XGBoost 3.0.2, scikit-learn 1.7.1, pandas 2.3.1, and NumPy 1.26.4.

Table 2: Optimized XGBoost hyperparameters.

Hyperparameter	Value
Subsample	0.7
$\lambda$ (reg_lambda)	1.0
$\alpha$ (reg_alpha)	0.5
Min child weight	3
Max depth	8
Learning rate	0.1
Gamma	0.5
Colsample by tree	1.0

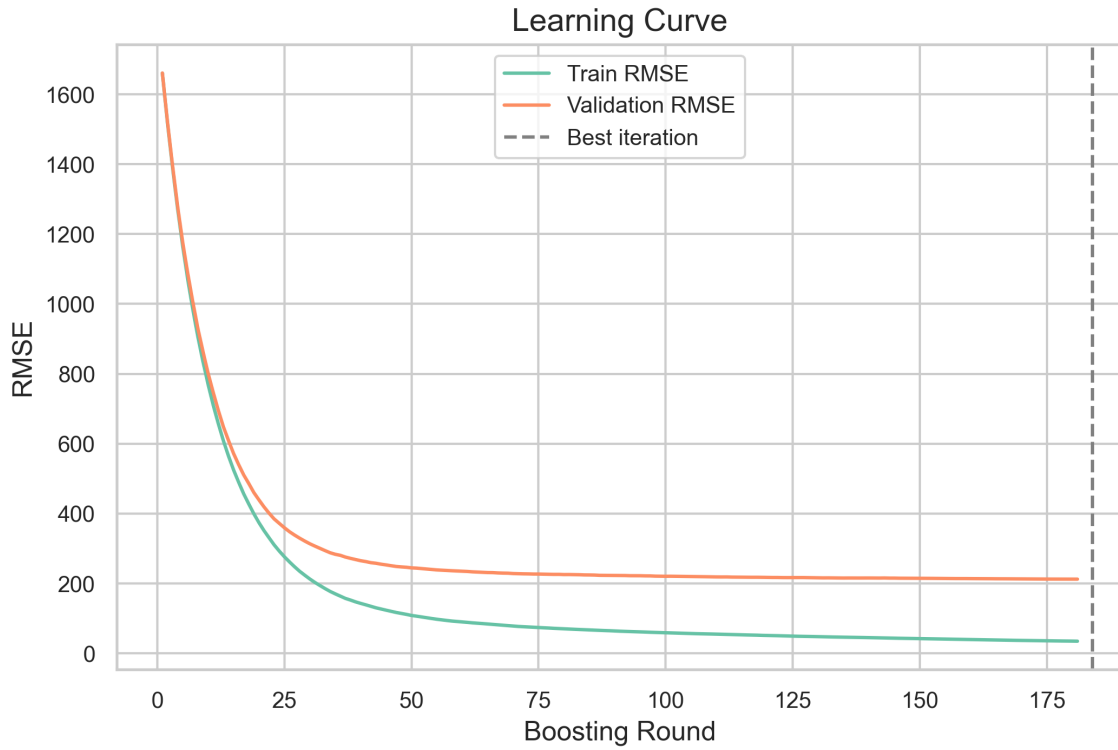


Figure 7: Learning curve showing RMSE convergence with early stopping. The curve illustrates how additional boosting iterations reduce error initially, then plateau once performance improvement ceases. Early stopping prevents overfitting and confirms that the selected number of iterations balances bias and variance.

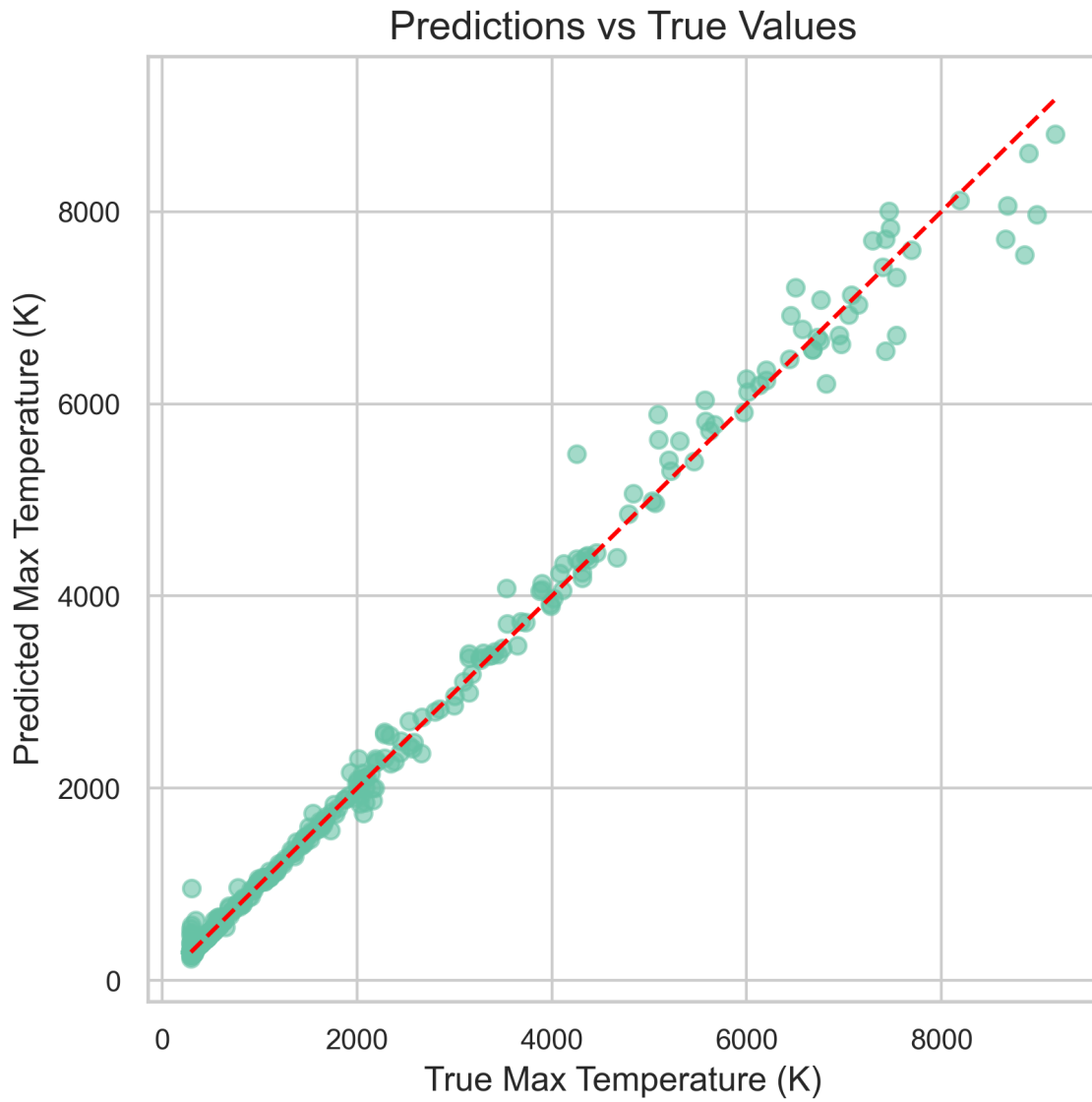


Figure 8: Scatter plot of predicted versus true maximum temperatures. Strong agreement is observed across most of the temperature range, with larger deviations appearing at higher temperatures ( $T_{\text{true}} \geq 6000 \text{ K}$ ). This highlights that the model captures the general trend accurately while exhibiting slightly higher uncertainty at extreme values.).

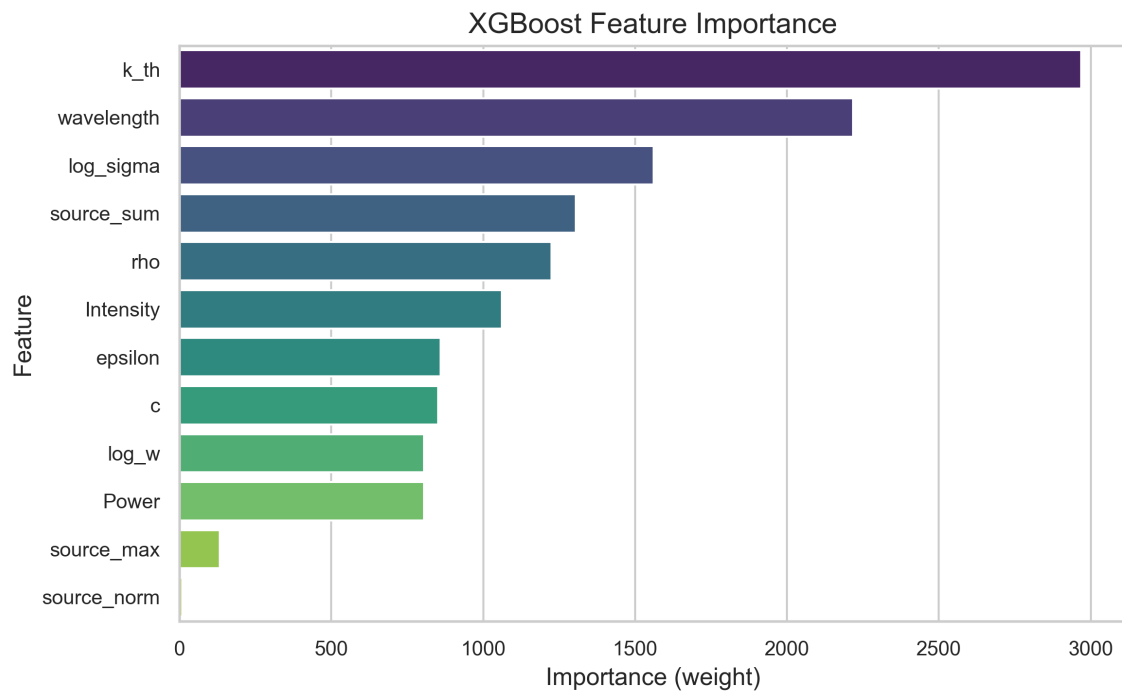


Figure 9: Ranking of input feature importance in the XGBoost model. Thermal conductivity and wavelength are identified as the most influential predictors, indicating that both material transport properties and laser energy deposition geometry dominate the determination of maximum temperature..

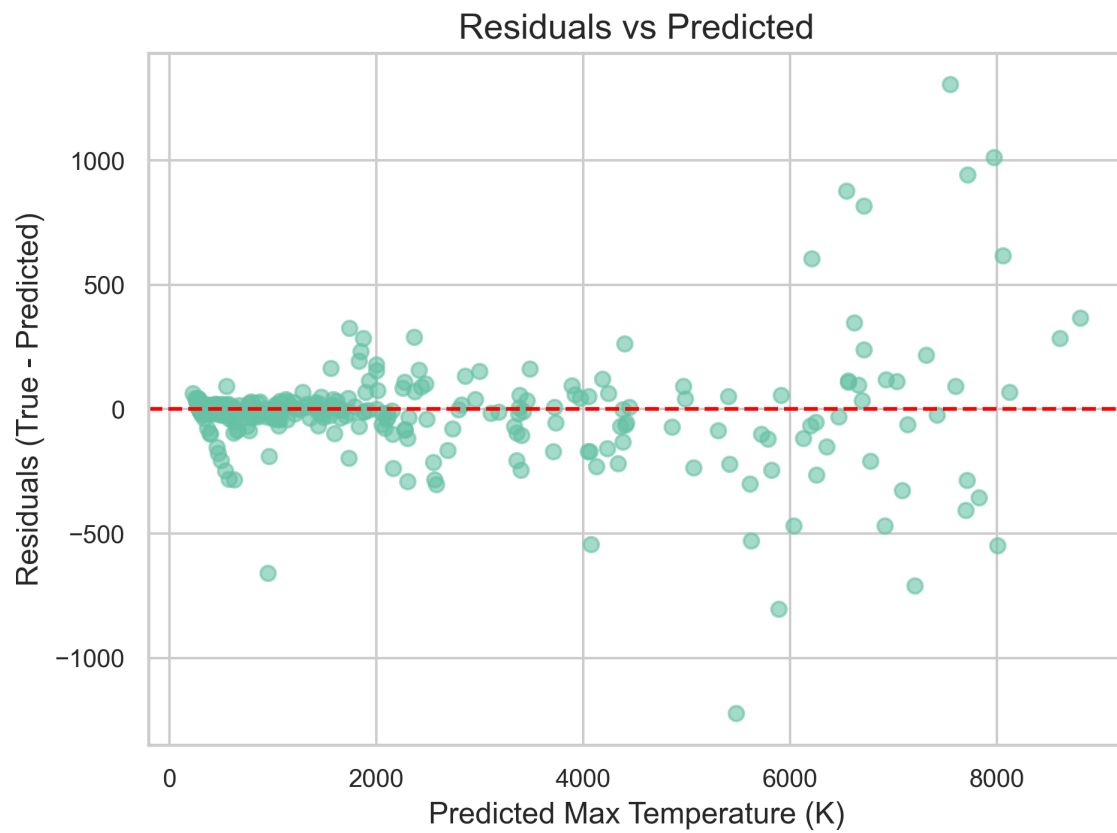


Figure 10: Residuals plotted against predicted maximum temperature. Most residuals are centered around zero, indicating unbiased predictions across the typical range. Larger deviations occur beyond 6000 K, consistent with Figure 8, reflecting increased difficulty in predicting extreme temperatures

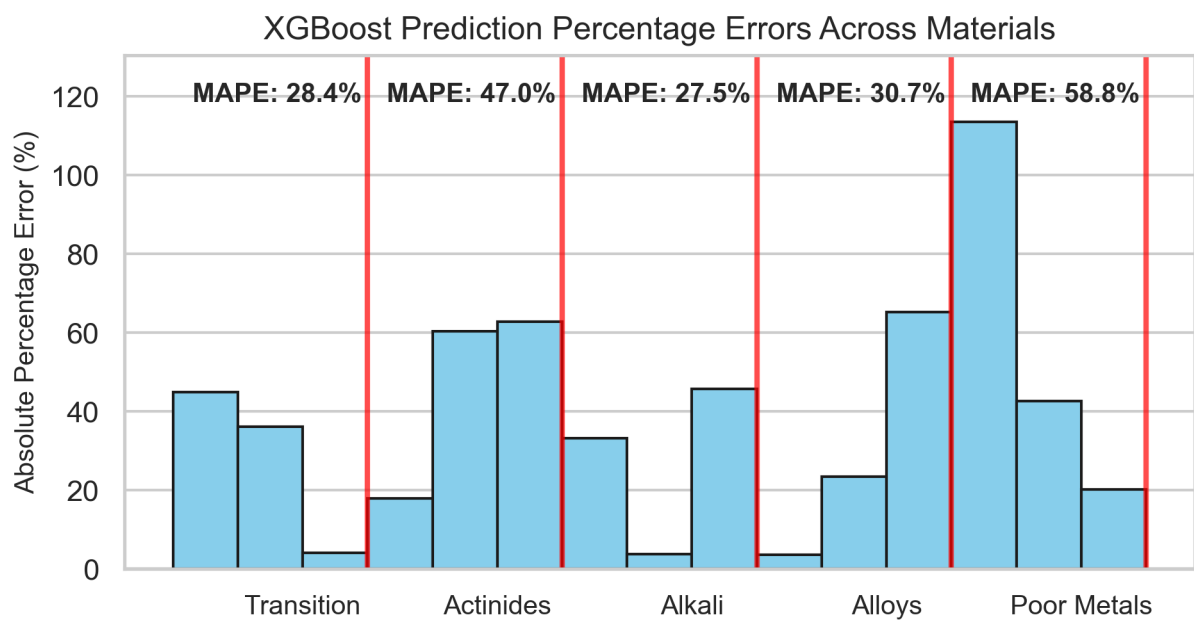


Figure 11: Generalization performance of the XGBoost model on unseen materials, reported as mean absolute percentage error (MAPE) averaged by material type. Within each type, MAPE varied across individual materials.



## 6. Discussion

This study presents a modular workflow for solving the 3D heat equation in cylindrical coordinates, implemented in Fortran for computational efficiency, and designed for extensibility to new materials and geometries. The framework and all supporting code are openly available, ensuring reproducibility and adaptability. The solver was validated against analytical solutions with excellent agreement (maximum temperature difference 0.23%, Table 1), physically consistent energy residuals (Fig. 4), and convergence of order  $\sim 1.5$  (Fig. 5). These results confirm the robustness and physical reliability of the numerical framework.

An XGBoost surrogate model with automated hyperparameter optimization was developed (Table 2), achieving high test precision ( $R^2 = 0.994$ ) and importance of physically interpretable characteristics (Fig. 9). Generalization tests on unseen materials (Figure 11) showed mean absolute percentage errors of  $\sim 30\%$  for transition metals, alkali metals, and alloys. For other groups, performance was weaker but still reasonable: most errors remained below 60%, with an outlier above 100%, probably due to limited representation of the training set rather than a systematic deficiency.

Although Yan et al. (2024) showed that physical constraints can improve XGBoost, our preliminary tests with physics-informed loss functions significantly reduced predictive accuracy. A promising alternative, not explored here, is to enrich the dataset with physically meaningful features (energy residuals) that implicitly encode such constraints, combining efficiency with physical interpretability.

The surrogate framework is general; the hyperparameter optimization and prediction workflow can be applied to any simulation dataset, enabling reuse beyond the present heat-transfer problem. Together, the results demonstrate both the reliability of the solver and the potential of machine learning surrogates to accelerate predictions across wide parameter spaces. Limitations remain: error scales with maximum temperature (Fig. 8), which reduces reliability under extreme thermal conditions, and the model cannot extrapolate beyond the training domain. These issues can be mitigated by expanding the dataset, retraining with additional materials, and leveraging the modular workflow for ongoing improvement.

Future work may explore approximating full temperature fields on coarse grids using efficient corrections, for example:

$$T(r, z) \approx T_{\text{pred}} \times T_{\text{coarse}}(r, z) T_{\text{norm}}, \quad (9)$$

and comparing this approach with neural network-based methods. If comparable errors are achieved, this would provide an efficient strategy for computing full-temperature fields from scalar predictions.

## 7. Conclusion

We have developed and validated a modular 3D heat equation solver, implemented in Fortran, that supports various materials, general boundary conditions, and realistic volumetric source term. Validation against analytical solutions confirmed numerical stability, energy conservation, and convergence consistent with first-order finite-difference theory. Building on this, we constructed an XGBoost-based surrogate model that achieved high accuracy ( $R^2 = 0.994$ ) within the training domain and extended reasonably well to several classes of unseen materials (transition metals, alkali metals, and alloys, with mean errors  $\sim 30\%$ ), while showing larger deviations for poor metals and actinides. This demonstrates both the promise of the surrogate framework and the need to expand the training data for challenging regimes. The complete workflow, including solver and surrogate framework openly released to enable reproducibility and straightforward extension to other geometries and multiphysics problems. This combination of reliable numerical modeling and efficient surrogate construction offers a practical computational tool for high-resolution temperature predictions, particularly suited for large-scale parameter sweeps and materials optimization where PINN approaches may be computationally prohibitive.

## References

- Bentéjac, C., Csörgő, A., Martínez-Muñoz, G., 2019. A comparative analysis of xgboost. doi:10.48550/arXiv.1911.01914.
- Breiman, L., 2001. Random forests. *Machine Learning* 45, 5–32. doi:10.1023/A:1010933404324.
- Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system. *arXiv preprint arXiv:1603.02754*.
- Dehghan, M., 2004. Weighted finite difference techniques for the one-dimensional advection–diffusion equation. *Applied Mathematics and Computation* 147, 307–319.
- Evans, L.C., 2010. *Partial Differential Equations*. 2nd ed., American Mathematical Society. Section 2.3, p. 49.
- Jalili, D., Jang, S., Jadidi, M., Giustini, G., Keshmiri, A., Mahmoudi, Y., 2024. Physics-informed neural networks for heat transfer prediction in two-phase flows. *International Journal of Heat and Mass Transfer* 221, 125089. doi:10.1016/j.ijheatmasstransfer.2023.125089.
- Konangi, S., Palakurthi, N.K., Ghia, U., 2018. von neumann stability analysis of first-order accurate discretization schemes for one-dimensional (1d) and two-dimensional (2d) fluid flow equations. *Computers & Mathematics with Applications* 75, 643–665. URL: <https://doi.org/10.1016/j.camwa.2017.09.040>, doi:10.1016/j.camwa.2017.09.040.
- Li, S., Wang, G., Di, Y., Wang, L., Wang, H., Zhou, Q., 2023. A physics-informed neural network framework to predict 3d temperature field without labeled data in process of laser metal deposition. *Engineering Applications of Artificial Intelligence* 120, 105908. URL: <https://doi.org/10.1016/j.engappai.2023.105908>, doi:10.1016/j.engappai.2023.105908.
- M’hamdi, O., Takács, S., Palotás, G., Ilahy, R., Helyes, L., Pék, Z., 2024. A comparative analysis of xgboost and neural network models for predicting some tomato fruit quality traits from environmental and meteorological data. *Plants* 13, 746. doi:10.3390/plants13050746.
- Pantawane, M., Robertson, W., Chipper, R., Khan, J., Fick, D., Dahotre, N., 2020. Evolution of surface morphology of er:yag laser-machined human bone. *Lasers in Medical Science* 35, 1–9. doi:10.1007/s10103-019-02927-w.

- Patankar, S.V., 1980. Numerical Heat Transfer and Fluid Flow. Taylor & Francis.
- Shahbazi, M.A., Azadeh-Fard, N., 2025. Hierarchical data modeling: A systematic comparison of statistical, tree-based, and neural network approaches. Machine Learning with Applications 21, 100688. URL: <https://doi.org/10.1016/j.mlwa.2025.100688>, doi:10.1016/j.mlwa.2025.100688. open access under a Creative Commons license.
- Shen, H., Shao, J., Zuo, X., An, S., 2020. Indoor visible light localization based on compressed sensing under matrix filling recovery. Optical and Quantum Electronics 52. doi:10.1007/s11082-020-02322-8.
- Smith, G.D., 1985. Numerical solution of partial differential equations: finite difference methods. Oxford University Press.
- Tsega, E.G., 2022. Numerical solution of three-dimensional transient heat conduction equation in cylindrical coordinates. Journal of Applied Mathematics 2022, 1–8. URL: <https://doi.org/10.1155/2022/1993151>, doi:10.1155/2022/1993151.
- Yan, B., Xu, Z., Gudala, M., Tariq, Z., Sun, S., Finkbeiner, T., 2024. Physics-informed machine learning for reservoir management of enhanced geothermal systems. Geoenergy Science and Engineering 234, 212663. doi:10.1016/j.geoen.2024.212663.

## Appendix A. Analytical Solution from Evans

For completeness, the full analytical solution for the temperature distribution in a three-dimensional infinite medium with a Gaussian laser source is given in cylindrical coordinates  $(r, z)$  as:

$$T(r, z, t) = \int_{\mathbb{R}^3} G(r - r', z - z', t) T_0(r', z') dV' + \int_0^t \int_{\mathbb{R}^3} G(r - r', z - z', t - t') f(r', z', t') dV' dt', \quad (\text{A.1})$$

where the Green's function for 3D heat conduction is

$$G(r, z, t) = \frac{1}{(4\pi\alpha t)^{3/2}} \exp\left(-\frac{r^2 + z^2}{4\alpha t}\right), \quad (\text{A.2})$$

and  $f(r, z, t)$  represents the Gaussian laser pulse source term as described in Equation 2. This solution assumes an infinite domain, which is why the simulation domain was reduced by 40% to match physically meaningful results.

## Appendix B. Material Properties

Table B.3: Material Properties Database

Material	$k$ (W/m·K)	$\rho$ (kg/m <sup>3</sup> )	$c$ (J/kg·K)	$\epsilon$
Aluminum	237.0	2700	897	8.4
Copper	401.0	8960	385	2.6
Gold	318.0	19300	129	5.4
Iron	80.0	7874	449	3.1
Lead	35.0	11340	128	1.6
Nickel	90.9	8908	444	3.3
Silver	429.0	10490	235	4.0
Steel	50.0	7850	486	2.5
Titanium	21.9	4507	522	3.5
Tungsten	173.0	19300	134	3.4
Zinc	116.0	7140	388	1.0
Magnesium	156.0	1740	1023	1.2
Platinum	71.6	21450	133	4.2
Chromium	93.9	7190	448	3.3
Brass	109.0	8530	380	2.0
Bronze	60.0	8800	380	2.0
StainlessSteel	16.0	8000	500	2.5
Molybdenum	138.0	10220	251	3.0
Beryllium	200.0	1850	1825	2.0
Lithium	85.0	534	3582	1.5
Sodium	142.0	968	1230	1.5
Potassium	102.0	856	757	1.5
Uranium	27.0	19050	116	2.5
Mercury	8.3	13534	140	1.0
Cadmium	96.0	8650	230	2.0
Palladium	72.0	12023	240	3.5
Rhodium	150.0	12450	240	3.5
Iridium	147.0	22560	131	3.5
Osmium	87.0	22590	130	3.5
Bismuth	7.9	9780	122	1.5
Cobalt	100.0	8900	420	3.0

## Appendix C. Solver Parameters

Table C.4: Key Variables and Parameters for Cylindrical Heat Equation Solver

Category	Variable	Description / Units
Input Parameters	$w$	Laser beam radius (m)
	$P$	Laser power (W)
	$\sigma$	Temporal standard deviation of pulse (s)
	wavelength	Laser wavelength (m)
	material_name	Name of material (string)
Material Properties	$k$	Thermal conductivity (W/m·K)
	$\rho$	Density (kg/m <sup>3</sup> )
	$c$	Specific heat capacity (J/kg·K)
	$\varepsilon$	Extinction coefficient
	$\alpha$	Thermal diffusivity $k/(\rho c)$ (m <sup>2</sup> /s)
Simulation Parameters	$T_0$	Initial temperature (K)
	length	Side length of domain (m)
	step_number	Grid resolution (number of points per axis)
	$nr, nz$	Number of radial / axial points
	$dr, dz$	Grid spacing in r and z (m)
	$dt$	Time step (s)
	$n\_steps$	Number of time steps
	$h$	Convective heat transfer coefficient (W/m <sup>2</sup> ·K)
	$T_{\text{ambient}}$	Ambient temperature (K)
Derived / Computed	$\mu_a = \frac{4\pi\varepsilon}{\lambda}$	Absorption coefficient (1/m)
	$\delta = \frac{1}{\mu_a}$	Penetration depth (m)
	$I_0 = \frac{P}{\pi w^2}$	Laser intensity (W/m <sup>2</sup> )
	$R, Z$	Grid coordinate arrays (m)
	Temp_field	Temperature field at current timestep (K)
	T_new	Updated temperature field (K)
	source_term	Heat source term (K/s)
	laplacian	Laplacian of temperature field (K/m <sup>2</sup> )
	center_T	Maximum temperature at center (K)
	times	Array of time values (s)
	center_temps	Array of center temperatures (K)

## Appendix D. Implementation of the model

### Appendix D.1. Physical Input Module

This module should include all the material properties used, and can be adjusted for different PDE's that require certain physical inputs that are system dependent.

```
! =====
! Material Module (simplified)
! =====
module material_module
  implicit none
  public :: get_material, material
  type :: material
    real :: k, rho, c, epsilon
  end type material

  function get_material(name) result(mat)
    character(len=*), intent(in) :: name
    type(material) :: mat

    select case (trim(adjustl(name)))
    case ("Aluminum")
      mat = material(237, 2700.0, 897.0, 8.4)
    case ("Copper")
      mat = material(401, 8960.0, 385.0, 2.6)
    ! ... Other materials follow same pattern
    end select
  end function get_material
end module material_module
```

### Appendix D.2. Boundary Conditions

Subroutine boundary\_conditions\_cylindrical(T\_new, nr, nz)

1. Symmetry at  $r = 0$  (Neumann BC):  
T\_new(1,:) = T\_new(2,)
2. Insulated outer radial edge  $r = R$  (Neumann BC):  
T\_new(nr,:) = T\_new(nr-1,)

```

3. Insulated bottom surface  $z = H$  (Neumann BC):
   For i = 1 to nr:
       T_new(i,nz) = T_new(i,nz-1)

4. Convective boundary at top surface  $z = 0$  (optional):
   For i = 1 to nr:
       T_new(i,1) -= (h * dt / (rho * dz)) * (T_new(i,1) - T_ambient)
       [Note: commented out in code and used as part of the solver]
End Subroutine

```

### *Appendix D.3. Cylindrical Solver*

```

1. Load material properties:
   mat = get_material(material_name)
   alpha = mat.k / (mat.rho * mat.c)

2. Set up grid and time step:
   nr, nz = step_number
   dr = dz = length / (step_number-1)
   dt = dz^2 / (6 * alpha)
   n_steps = total_time / dt

3. Initialize temperature arrays:
   Temp_field(:, :) = T0
   T_new(:, :) = T0
   source_term(:, :) = 0
   laplacian(:, :) = 0

4. Compute cylindrical coordinates:
   R(i,j) = (i-1)*dr
   Z(i,j) = (j-1)*dz

5. Time-stepping loop:
   for time = 1 to n_steps
       current_time = time*dt

       5a. Compute source term for laser:
           source_term(i,j) = (I0*mu_a)/(rho*c) * exp(-R(i,j)^2 / w^2)
                               * exp(-Z(i,j)/delta)

```



```

                                * exp(-current_time^2 / (2*sigma^2))

5b. Compute Laplacian (finite differences in cylindrical coords):
    laplacian(1,kk) = special case at r=0
    laplacian(2:nr-1,2:nz-1) = general case

5c. Update temperature field:
    T_new(i,j) = Temp_field(i,j) + dt*(alpha*laplacian(i,j) + source_term(i,j))

5d. Apply boundary conditions:
    call boundary_conditions_cylindrical(T_new, nr, nz)

5e. Apply convective loss at surface z=0:
    T_new(i,1) -= h*dt/(rho*c*dz) * (Temp_field(i,1) - T_ambient)

5f. Track maximum center temperature:
    if T_new(1,1) > center_T: center_T = T_new(1,1)

5g. Store results for this timestep:
    times(time) = current_time
    center_temps(time) = T_new(1,1)

5h. Update Temp_field for next step:
    Temp_field = T_new

6. Return results (center temperature, time series, optional field)

```

## Appendix E. Main Program Pseudocode

Program main\_heat\_sim

1. Initialize general parameters:
  - Cylinder length, grid resolution
  - Material list
  - Laser parameter lists: w, P, sigma, wavelength
  - Test parameters for a single run (optional)
2. Open output file for writing results

```

3. Loop over materials:
  For each material in material_names:
    - Fetch material properties (k, rho, c, epsilon)

  Loop over laser powers P:
    Loop over beam radii w:
      Loop over wavelengths:
        Loop over pulse widths sigma:

          - Call heat_equation_solver_cylindrical(w, P, sigma,&
            wavelength,material_name)

          - Safety check: ensure temperature array is allocated

          - Write results to CSV file:
            material, P, w, wavelength, sigma, material properties,&
            max temperature

4. Close output file

End Program

```

## Appendix F. New material properties

Table F.5: Properties of the new materials used for generalization testing.

Material	Type	$k$ (W/m·K)	$\rho$ (kg/m <sup>3</sup> )	$c$ (J/kg·K)	$\epsilon$
Vanadium	Transition Metal	50.0	6100	490	3.0
Zirconium	Transition Metal	23.0	6500	280	2.5
Hafnium	Transition Metal	23.0	13300	144	2.5
Gallium	Poor Metal	29.0	5900	370	1.9
Indium	Poor Metal	81.8	7330	233	2.0
Thallium	Poor Metal	46.0	11850	129	1.8
Inconel	Alloy	11.0	8300	435	2.0
Duralumin	Alloy	120.0	2780	870	1.6
Constantan	Alloy	21.0	8900	390	1.5
Rubidium	Alkali Metal	58.0	1530	376	1.5
Cesium	Alkali Metal	35.0	1870	242	1.5
Francium	Alkali Metal	20.0	2100	200	1.5
Calcium	Alkaline Earth Metal	200.0	1550	647	1.2
Strontium	Alkaline Earth Metal	35.0	2190	420	1.2
Barium	Alkaline Earth Metal	18.0	3500	204	1.2
Thorium	Actinide	50.0	11700	120	2.5
Plutonium	Actinide	6.7	19800	120	2.0
Neptunium	Actinide	6.5	20700	115	2.0