

Article

Not peer-reviewed version

Thermal Field Reconstruction on Microcontrollers: A Physics-Informed Digital Twin Using Laplace Equation and Real-Time Sensor Data

[Victor H. Benitez](#)^{*}, [Jesus Pacheco](#), [Agustin Brau-Avila](#)

Posted Date: 11 July 2025

doi: 10.20944/preprints202507.0797.v1

Keywords: physics-Informed digital twin; thermal field estimation; embedded systems; finite difference method; real-time sensing



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Thermal Field Reconstruction on Microcontrollers: A Physics-Informed Digital Twin Using Laplace Equation and Real-Time Sensor Data

Victor H. Benitez *, Jesus Pacheco and Agustin Brau

Department of Industrial Engineering, Universidad de Sonora, Hermosillo 83000, Mexico

* Correspondence: victor.benitez@unison.mx

Highlights

- A physics-informed digital twin is implemented on an embedded system for real-time thermal monitoring.
- The system solves the steady-state 2D Laplace equation using a finite difference method directly in embedded hardware.
- Boundary conditions are obtained from calibrated thermistors using the Steinhart–Hart model and processed via DMA-enabled ADCs.
- The reconstructed thermal field is transmitted through UART and visualized as a live heatmap using a Python interface.
- Demonstrates real-time PDE simulation on low-cost hardware without external computation or cloud processing.
- Enables compact, scalable, and physically consistent thermal diagnostics suitable for education and edge computing applications.

What are the main findings?

- The embedded system accurately reconstructs internal temperature distributions from sparse boundary sensor data in real time.
- The proposed architecture achieves stable and low-latency thermal simulation using physics-based modeling on resource-constrained microcontrollers.

What is the implication of the main finding?

- Enables the deployment of physics-based digital twins in low-power edge devices without reliance on cloud infrastructure.
- Facilitates affordable and interpretable thermal diagnostics and monitoring in educational, prototyping, and industrial environments.

Abstract

This paper presents a physics-informed digital twin designed for real-time thermal monitoring and visualization of a metallic plate. The system comprises a physical layer consisting of an aluminum plate equipped with thermistors to capture boundary conditions, a computational layer that implements the steady-state Laplace equation using the finite difference method, and an embedded execution framework deployed on a microcontroller that utilizes Direct Memory Access-driven ADC for efficient concurrent acquisition. The computed thermal field is transmitted through a serial interface and displayed in real time using a Python-based visualization interface. The Steinhart–Hart model was used to experimentally characterize the sensors, ensuring accuracy in the boundary condition acquisition. Results demonstrate that the proposed architecture can simulate and reconstruct internal thermal states with high spatial resolution in resource-constrained hardware. This affordable, real-time approach is suitable for educational applications, industrial training, industrial thermal diagnostics, and the prototyping of embedded digital twin systems. Future extensions include adaptive mesh refinement, dynamic thermal modeling, and integration with control systems.

Keywords: physics-Informed digital twin; thermal field estimation; embedded systems; finite difference method; real-time sensing

1. Introduction

The two-dimensional Laplace equation is fundamental for simulating steady-state physical phenomena such as heat conduction, electrostatics, and incompressible fluid flow. Numerical solutions of this equation yield spatial temperature distributions that are vital for understanding and managing thermal behavior in engineered systems. Until a few years ago, solving this equation in real-time required powerful desktop computers or workstations. However, recent advancements in embedded computing have fostered the development of innovative strategies for executing these solvers on low-power, cost-effective microcontrollers.

This study presents an innovative embedded implementation of a physics-informed digital twin, designed to monitor and visualize the temperature distribution across a metallic plate governed by the two-dimensional Laplace equation. Unlike conventional data-driven digital twins, which often rely on black-box machine learning models and extensive datasets, this implementation is based on first-principles modeling (a physically informed system).

The physical layer consists of a one mm-thick aluminum plate equipped with thermistors distributed along its boundaries. The thermistors are experimentally calibrated using the nonlinear Steinhart–Hart model, which requires the experimental measurement of three known resistance-temperature pairs to solve a system of equations and obtain a precise conversion from resistance to temperature.

A Python-based graphical interface communicates with the microcontroller via serial communication to visualize the results. The interface renders the computed temperature distribution as a dynamic thermal map, enabling intuitive interpretation and real-time monitoring.

This architecture enables physics-informed digital twins in constrained environments, with potential applications in education, rapid prototyping of thermal systems, and real-time monitoring of photovoltaic modules, as well as thermal testing of construction materials and embedded thermal diagnostics. The system eliminates the need for external computation and significantly reduces data transmission overhead by sending only the computed temperature matrix, rather than raw sensor data, thereby bringing physical modeling directly to the edge. The proposed approach enhances scalability, minimizes latency, and enables the efficient and autonomous operation of the digital twin in embedded contexts. The system exemplifies how embedded digital twins can reconstruct spatial fields from sparse sensor data and can be adapted to other domains involving steady-state partial differential equations (PDEs).

The remainder of the paper is structured as follows: Section 2 reviews related work and the state of the art. Section 3 introduces the theoretical model and system architecture. Section 4 describes the experimental implementation, including sensor calibration, hardware/software configuration, and visualization strategy. Section 5 discusses implications, limitations, and future directions. Finally, Section 6 presents the conclusions.

2. Related Work

The concept of digital twins (DTs) has gained significant traction in recent years, in the context of thermal system monitoring and control. References, such as [1], [2], [3], [4] can be found, where several approaches are presented, all of which rely on high computing resources. DTs are virtual replicas of physical systems that enable real-time monitoring, simulation, and optimization. They are broadly categorized into data-driven, physics-based, and hybrid approaches [5].

2.1. Data-Driven Digital Twins

Data-driven digital twins depend on machine learning algorithms trained on historical and real-time data to predict system behavior. For example, artificial neural networks (ANNs) and recurrent neural networks (RNNs) have been utilized to forecast thermal load profiles in buildings and industrial systems [6]. Support vector machines (SVMs) and decision trees are utilized in digital twin frameworks to identify anomalies in HVAC systems and enhance energy consumption efficiency [7]. These applications demonstrate how data-driven approaches can identify intricate thermal behavior patterns, providing real-time monitoring and control solutions.

2.2. *Physics-Based Digital Twins*

Physics-based digital twins (PBDTs) employ mathematical models rooted in fundamental physical principles to simulate system behavior. These models frequently require solving partial differential equations (PDEs) that represent the underlying physical phenomena. For example, the Laplace equation has been applied in robotics for path planning, improving its properties to model potential fields that guide robots through environments while avoiding obstacles [8]. In industrial settings, PBDTs are used to simulate and manage heating processes. By integrating the principles of heat transfer, such as conduction and radiation, these DTs can model temperature distributions within heating systems like furnaces. This allows for precise control of the heating process, enhancing both efficiency and product quality. For example, a study developed a physics-informed DT for industrial heating processes, integrating physical models with machine learning techniques to improve temperature prediction accuracy [9].

2.3. *Hybrid Approaches for Digital Twins*

Hybrid digital twins combine data-driven and physics-based modeling methods to power their strengths. Although physics-based models ensure interpretability and comply with fundamental principles, data-driven models demonstrate adaptability and can manage complex datasets. Combining these approaches facilitates enhanced prediction accuracy, robustness, and generalization, particularly in dynamic environments where either approach alone may be insufficient.

Recent studies have explored this synergy in the context of thermal systems. For example, a hybrid digital twin framework was developed to model and control transient thermal systems, where deep neural networks were trained on stationary data to estimate heat transfer coefficients in real time. These coefficients were then used within a finite element (FE) physical model to regulate maximum temperatures under dynamic heat fluxes, ensuring operational safety and thermal efficiency in 1D, 2D, and 3D domains [10]. In aerospace applications, a hybrid digital twin architecture is proposed to enhance the prediction of satellite temperature fields. This approach merges data-driven models with physical field simulations, ensuring high fidelity across both spatial and temporal dimensions [11].

Recent technical literature and industry discussions indicate that developing hybrid DTs improves modeling accuracy and decision-making in complex systems. These systems face considerable challenges due to uncertain boundary conditions, time-varying dynamics, and nonlinearities [12].

2.4. *Numerical Solutions of PDEs on Embedded Systems*

Solving Laplace's equation on embedded systems is challenging due to the intensive computation and memory required for numerical solutions. Traditionally, powerful processors handle such PDEs, but recent advances have enabled embedded solutions. For instance, researchers have developed custom low-power accelerators that directly solve discretized Laplace/Poisson equations in silicon [13]. On the other hand, the authors of the reference [14] have explored analog computing approaches. They developed an analog finite-difference time-domain solver chip for a nonlinear shock-tube PDE, with a microcontroller used to calibrate the system. These researchers

demonstrate advanced PDE solvers on embedded platforms, utilizing digital and analog hardware acceleration to manage large linear systems from Laplace's equation.

Even low-cost microcontrollers, though with coarse resolution, can solve Laplace's equation (or similar PDEs) in real time. Simple, memory-local iterative methods like Jacobi or Gauss-Seidel relaxation are implementable on microcontroller units (MCUs) to approximate Laplace's solution on a grid. For instance, an Arduino demo solved a 2D fluid flow equation using a finite difference scheme on a 25×25 grid [15]. Although performance is limited, integrating a PDE solver into a microcontroller enables autonomous sensor and actuator devices that can, for example, compute temperature or potential fields on-device without relying on cloud or PC resources.

2.5. Finite Difference Methods on Microcontrollers

Among numerical techniques for PDEs, the finite difference method (FDM) is particularly suitable for implementation on microcontrollers due to its straightforward and iterative nature. FDM replaces the continuous Laplace operator $\nabla^2 u = 0$ with discrete differences on a mesh, resulting in linear equations that relate each point to its neighbors [13]. FDM stencils, such as the five-point stencil in 2D Laplace problems, require each grid node update to be a simple arithmetic average or weighted sum of neighbor values—operations manageable by a modest 32-bit MCU. Implementing an FDM solver involves using an iterative solver (like Jacobi or Gauss-Seidel) to relax the solution across the grid. By applying local updates, these algorithms can halt when the change in the solution is below a specified threshold, ensuring that they conform to the limited throughput of microcontrollers.

Over the past ten years, academic and industrial research has demonstrated that microcontrollers and embedded processors can perform FDM computations for PDEs in real-time, given specific constraints. On the more advanced side, custom hardware accelerators combine numerous simple processing elements to update grid points simultaneously [16]. In reference [16], Mu and Kim presented a reconfigurable-bit-width 21×21 processing-element lattice that maps directly onto a 2D grid for FDM computations. Each tiny processing element handles the update of one cell of the discretized domain and communicates with its four neighbors, enabling massive parallelism for iterative methods.

2.6. Computing Thermal Boundary Conditions

It is necessary to apply thermal boundary conditions to address heat distribution challenges, particularly in embedded systems where Laplace's equation serves as a specific steady-state model. Thermistors are a popular choice for temperature sensors at boundaries due to their small size and high sensitivity. In embedded PDE applications, thermistor measurements can be utilized to set Dirichlet boundary values for fixed temperatures or Neumann conditions at the edges of a simulated domain [17]. However, raw thermistor signals are highly nonlinear with respect to temperature due to the semiconductor nature of the material from which the thermistor is made. The Steinhart-Hart method is a widely used empirical model for converting thermistor resistance into an accurate temperature reading [18]. This model expresses the temperature as a cubic polynomial as

$$\frac{1}{T} = A + B \ln(R) + C (\ln(R))^3 \quad (1)$$

with A, B, and C as sensor-specific coefficients; T refers to the temperature in Kelvin, and R denotes the electrical resistance.

3. Theoretical Background

3.1. Laplace's Equation in Heat Transfer

Laplace's equation is a second-order linear partial differential equation (PDE) that describes the behavior of scalar fields in the absence of sources or sinks. In two dimensions, it takes the form

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \quad (2)$$

This formulation is based on Fourier's law and the conservation of energy, assuming the system has reached thermal equilibrium. In the context of digital twins, this PDE serves as the physics-informed backbone for estimating the internal temperature field from limited boundary measurements.

To implement this numerically on embedded hardware, the domain (a 2D metal plate) is discretized into a grid. Using the finite difference method, the second-order partial derivatives are approximated as:

$$\begin{aligned} \frac{\partial^2 T}{\partial x^2} &\approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} \\ \frac{\partial^2 T}{\partial y^2} &\approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} \end{aligned} \quad (3)$$

Assuming a square mesh, $\Delta x = \Delta y$, and combining both approximations, the discrete Laplace operator is obtained as

$$T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1} - 4T_{i,j} = 0 \quad (4)$$

This algebraic expression is valid for all interior nodes. Boundary conditions are imposed using real-time sensor data from thermistors distributed along the edges of the plate. These readings serve as Dirichlet boundary conditions.

3.2. Embedded Implementation and Convergence

The discrete system of equations is solved iteratively using Liebmann's method. To compute the update rule for interior nodes the equation (5) is applied

$$T_{i,j}^{(n+1)} = \frac{1}{4} \left(T_{i+1,j}^{(n)} + T_{i-1,j}^{(n)} + T_{i,j+1}^{(n)} + T_{i,j-1}^{(n)} \right) \quad (5)$$

The $(n+1)$ term is solved iteratively for $j = 1$ to n and $i = 1$ to m . Since Eq. (4) is diagonally dominant, this procedure will ultimately converge on a stable solution. To enhance the convergence rate, overrelaxation is employed by applying (6) after each iteration.

$$T_{i,j}^{(n)} = \lambda T_{i,j}^{(n)} + (1 - \lambda) T_{i,j}^{(n-1)} \quad (6)$$

where λ is a weighting factor that is set between 1 and 2. Convergence is determined when the maximum absolute difference between iterations falls below a tolerance criterion

$$\left| (\mathcal{E}_{tol})_{i,j} \right| = \left| \frac{T_{i,j}^{(n)} - T_{i,j}^{(n-1)}}{T_{i,j}^{(n)}} \right| \times 100 \quad (7)$$

This tolerance, combined with uniform grid spacing and proper floating-point management, ensures numerical stability even on resource-constrained microcontrollers. The physics-informed nature of the model ensures interpretability, physical consistency, and robustness in reconstructing internal states from limited boundary measurements, qualifying the system as a true digital twin.

3.3. Embedded Systems Architecture for Real-Time Thermal Monitoring

Implementing DTs in embedded systems presents several challenges due to limitations in computational resources, memory availability, and power consumption. However, recent

advancements in microcontroller design have significantly enhanced the feasibility of deploying real-time monitoring and control solutions at the edge.

The STM32 microcontroller series, based on the ARM Cortex-M architecture, spans a wide range of performance classes. From the ultra-low-power Cortex-M0/M0+ (STM32F0/L0 series), suitable for simple sensing and control, to the high-performance Cortex-M7 cores (e.g., STM32F7, STM32H7), capable of DSP-level operations and floating-point arithmetic, the STM32 ecosystem offers a scalable platform for embedded implementations. Mid-range variants, such as the Cortex-M3/M4 (STM32F1/F3/F4 series), offer a balance between computational capability, peripheral variety, and energy efficiency [19]. Additionally, STM32 technology includes advanced features like dual ADCs, Direct Memory Access (DMA), floating-point units (FPU), and digital signal processing (DSP) instructions embedded in a single core, which are critical for real-time signal acquisition and numerical computations [20].

Compared to other providers, such as Microchip's PIC microcontrollers, STM32 MCUs offer higher clock speeds, broader 32-bit architecture support, and more advanced peripherals. While SoCs (e.g., Raspberry Pi) and dedicated DSPs (e.g., TI C6000 series) can provide greater processing capabilities, they often exceed power, cost, and complexity constraints for embedded applications. STM32 MCUs bridge this gap by offering DSP-like performance in a low-power, suitable for field-deployable systems [21,22].

For instance, the STM32F3XX microcontroller family, equipped with dual high-speed ADCs and integrated DMA controllers, has been successfully employed in thermal applications requiring synchronized multichannel sampling. Coupling such hardware capabilities with numerical methods like the finite difference method enables the implementation of compact, low-latency embedded systems that simulate physical heat transfer phenomena in real-time [23].

4. Methodology

The methodology is constituted by four key components: (i) physical system and boundary sensing, (ii) embedded numerical solver, (iii) communication and data exchange, and (iv) visualization interface. The core computational framework is embedded in a low-power microcontroller that integrates sensor acquisition, finite-difference numerical modeling, and a real-time thermal block.

4.1. Embedded Systems Architecture for Real-Time Thermal Monitoring

The physical system consists of a planar metallic surface acting as a 2D conductive domain governed by Laplace's equation under steady-state conditions

$$\Delta^2 T(x, y) = 0 \quad (8)$$

To emulate physical boundary conditions, sensors are positioned along the edges of the medium to monitor fixed temperature. The thermistor sensors are of the NTC type and are characterized using the Steinhart–Hart model, as shown in equation (1). These sensors connect to the analog inputs of the microcontroller through precision voltage dividers. Each reading is sampled via dual 12-bit ADCs operating in DMA mode on the STM32F303, enabling efficient and parallel acquisition of temperature boundary values.

4.2. Embedded Numerical Solver

The thermal field inside the conductive medium is computed using a finite difference method (FDM), where the Laplacian operator is discretized over a uniform grid using Equation (5).

The domain is modeled as a 2D array of nodes, with each node representing a discretized temperature point. Boundary values are updated dynamically based on thermistor readings, while interior nodes are iteratively updated until the steady-state convergence criteria are met. The iterative solver is optimized for low memory usage and computational latency, employing in-place array

updates and integer arithmetic where feasible. Convergence thresholds are pre-calibrated to strike a balance between resolution and computational load.

The real-time performance of the proposed digital twin is enabled by the internal architecture of the STM32F303RET6 microcontroller, which integrates a high-speed Arm Cortex-M4 core with an advanced bus matrix and dual DMA controllers. Figure 1 illustrates a multi-layer AHB bus matrix that interconnects peripherals, memory blocks, and system buses, enabling high-throughput, low-latency communication throughout the system.

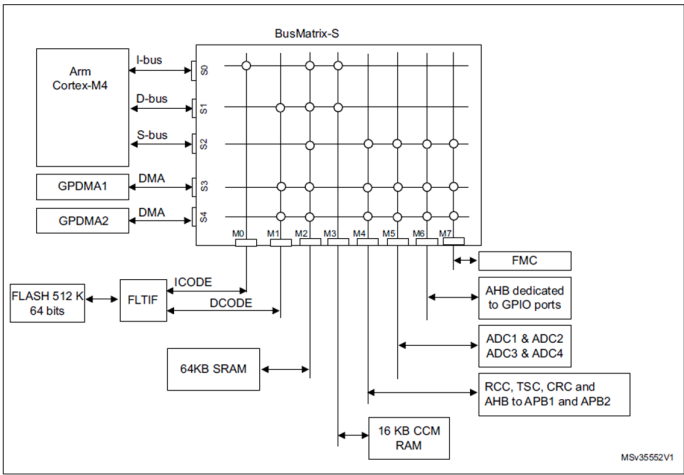


Figure 1. BusMatrix architecture depicting the DMA component and its integration to transfer peripheral data to memory in the ARM Cortex-M4 embedded systems.

A key architectural aspect of this implementation is the presence of two analog-to-digital converters (ADC1 and ADC2), both of which can function in independent scan mode. These ADCs are linked to DMA1 and DMA2 controllers, respectively, enabling simultaneous data collection from various sensors without CPU intervention.

The system ensures continuous, non-blocking acquisition of boundary temperature data by configuring each ADC to operate in direct DMA mode. The DMA engines autonomously transfer data from the ADC data registers to memory buffers, freeing the CPU to execute the finite-difference Laplace solver concurrently. This architectural parallelism minimizes system latency and is critical for the physical consistency of the digital twin.

4.3. Communication and Data Exchange

The estimated thermal field is transmitted via a UART interface for real-time monitoring. The STM32F303 MCU employs a communication protocol to serialize the two-dimensional temperature matrix into byte streams, thereby ensuring synchronization and fault-tolerant transmission. For the purposes of debugging and calibration, the microcontroller also records boundary sensor values and solver convergence metrics at fixed intervals. Additionally, this interface supports optional I²C expansion for configurations involving multiple sensors.

4.4. Communication and Data Exchange

The communication phase of the digital twin architecture methodology is based on the efficient coupling between the embedded system and the external digital representation. After reconstructing the internal thermal field from boundary data using the embedded Laplace solver, the result must be transferred to a host system for visualization and further analysis. This stage closes the feedback loop between the physical and digital layers, preserving the synchronicity of the digital twin.

The communication protocol utilizes a low-latency transmission scheme, employing a serial interface to transmit the computed temperature matrix. In formal terms, let $T(x,y)$ be the

continuous solution that is approximated numerically as a discrete grid $T_{i,j}$. Once convergence of the numerical scheme is reached, the matrix $T_{i,j}$ is linearly serialized into a vector form using a linear transformation given by equation (9)

$$\mathbf{T}_{\text{vec}} = \text{vec}(T_{i,j}) \in \mathbb{R}^{N^2} \quad (9)$$

where N^2 indicates a square grid of sensors. This vector contains the temperature values of the interior mesh nodes (excluding boundary points, which are physically measured) and is transmitted over a reliable and lightweight point-to-point protocol to the host system.

The design prioritizes minimal communication overhead, aligning with the embedded constraints of limited buffer sizes and energy efficiency. No external memory or higher-layer communication stacks (e.g., TCP/IP) are employed, ensuring that data transfer latency remains bounded and predictable.

On the host side, a software interface reconstructs the 2D thermal field from the received vector using the inverse mapping

$$T_{i,j} = \text{vec}^{-1}(\mathbf{T}_{\text{vec}}) \quad (10)$$

Equation (10) implements a reconstruction vector that feeds into a visualization layer, which interprets the temperature distribution as a color-coded heatmap, simulating a virtual thermal camera. Visualization is a fundamental element of the digital twin architecture, providing real-time insights into the evolution of the physical system's state based on sensed data.

From a methodological perspective, the communication layer provides the observability of the physical system, translating raw sensor data and internal model computations into an intelligible spatial representation. Moreover, since the model relies on physical laws rather than empirical mappings, the resulting visualization is grounded in physically consistent behavior, enhancing its interpretability.

This layer completes the digital twin cycle by creating a two-way channel: the embedded system communicates with the digital interface, and visual feedback can influence operational or experimental decisions in the physical environment. The proposed architectural modularity also allows for future enhancements, such as feedback-based control, anomaly detection, and integration into broader cyber-physical systems.

Figure 2 displays the methodology employed in this study.

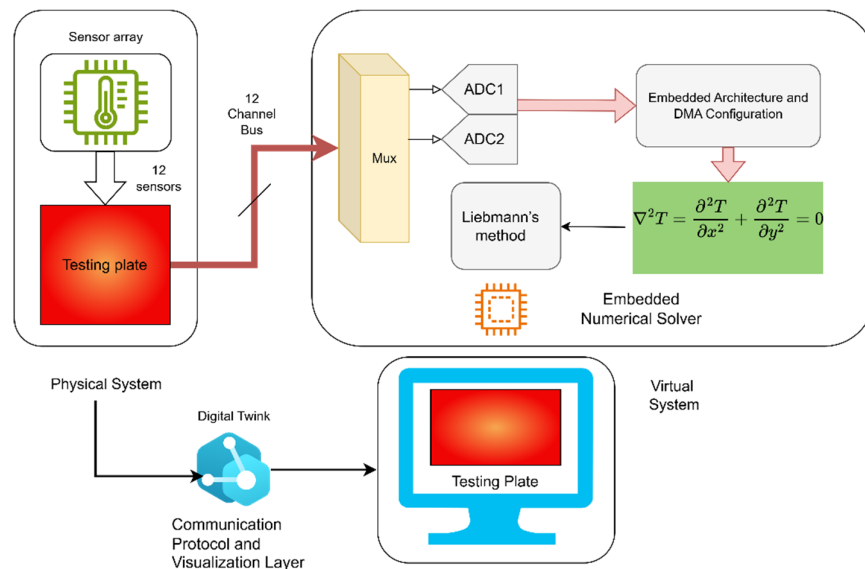


Figure 2. Components of the proposed methodology.

5. Results

5.1. System Setup and Execution

The entire system was installed on a custom fixture that supports a 1 mm aluminum plate equipped with 12 thermistors (three on each edge). These sensors measure real-time temperatures and are used as Dirichlet boundary conditions for the numerical solver implemented within the embedded system. The spatial domain of the plate was discretized into a uniform $N \times N$ grid, with $N = 5$, resulting in 25 nodes. Figure 3 displays the conceptual and implemented grid array of sensors deployed on the plate.

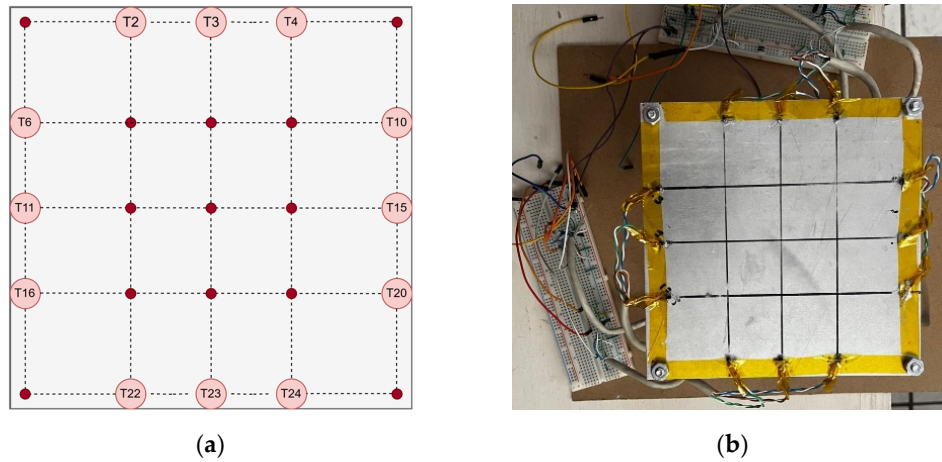


Figure 3. (a) Conceptual representation of the sensor grid array; (b) sensor deployment on the real plate.

Figure 3 (a) shows the estimated temperature (red dots) calculated by the algorithm, along with the actual temperatures (circled values) measured by the thermistors. This is represented as a matrix, with the plate divided into sections as

$$\begin{bmatrix} T_1 & T_2 & T_3 & T_4 & T_5 \\ T_6 & T_7 & T_8 & T_9 & T_{10} \\ T_{11} & T_{12} & T_{13} & T_{14} & T_{15} \\ T_{16} & T_{17} & T_{18} & T_{19} & T_{20} \\ T_{21} & T_{22} & T_{23} & T_{24} & T_{25} \end{bmatrix}$$

Notice that T_{13} is the inner estimated temperature. The resolution (grid size) was selected to balance the computational load of the embedded processor with the spatial resolution needed to capture meaningful gradients.

Sensors were linked to the ADC1 and ADC2 peripherals of the microcontroller, which were configured for DMA-based sampling to avoid any interruptions in the numerical computation loop. The grid size was determined based on the available RAM and processing power of the MCU. Each node of the internal domain was updated using a Jacobi method, and the maximum difference between iterations was monitored to evaluate convergence.

The embedded system implements the Liebmann method, a variant of the Jacobi iterative solver, to compute the steady-state temperature distribution. At each iteration, every interior node $T_{i,j}$ is updated by averaging the temperature values of its four immediate neighbors according to equation (5).

To accelerate convergence, a successive over-relaxation technique is applied as described by equations (6) and (7). The iterative process continues until the maximum change between iterations falls below a predefined threshold \mathcal{E}_{tol} , ensuring numerical stability.

Once convergence is achieved, the temperature matrix is flattened and transmitted over UART to a Python-based visualization layer, enabling real-time rendering of the heat distribution. The system performs acquisition, computation, and communication autonomously, without external processing.

5.2. Boundary Condition Measurement

Boundary conditions play a critical role in ensuring the accuracy of the numerical model implementation. Each of the twelve thermistors attached to the edges of the metallic plate was experimentally characterized using the Steinhart–Hart model, represented by equation (1). The resistance R_i of each sensor is calculated from the ADC value using a voltage divider with a known series resistor, and temperature is estimated using equation (1). The unknowns A, B, and C are the Steinhart–Hart coefficients that uniquely characterize each thermistor. These coefficients must be obtained through experimental calibration by measuring the resistance of the thermistor at three known temperatures.

Let the three resistance R and temperature T calibration points be (T1, R1), (T2, R2), and (T3, R3). For each point, equation (11) is applied. This results in a system of three nonlinear equations, which can be expressed in matrix form as

$$\begin{bmatrix} 1 & \ln(R_1) & \ln^3(R_1) \\ 1 & \ln(R_2) & \ln^3(R_2) \\ 1 & \ln(R_3) & \ln^3(R_3) \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \frac{1}{T_1} \\ \frac{1}{T_2} \\ \frac{1}{T_3} \end{bmatrix} \quad (11)$$

Solving this system yields coefficients A, B, and C, enabling the embedded system to estimate real-time temperatures from ADC computations accurately. In Figure 4, an example of sensor calibration is displayed with the coefficients computed.

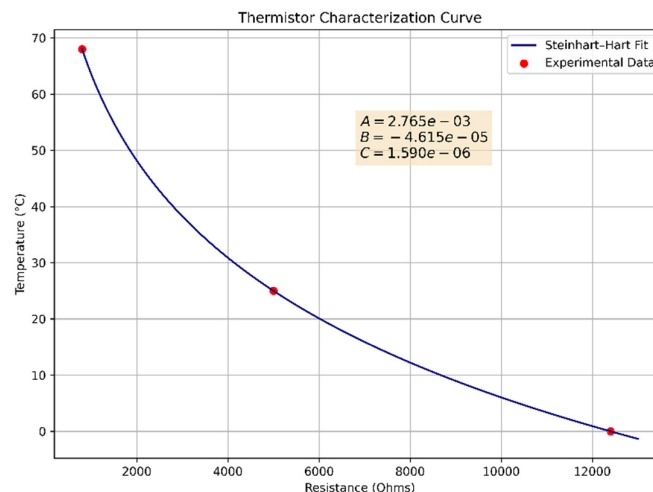


Figure 4. Experimental characterization curve of one thermistor.

5.3. Real-Time Temperature Field Computation

This phase entails running the numerical model integrated within the microcontroller to calculate the steady-state temperature distribution over the metallic plate. The previously outlined

architecture enables the effective collection of boundary values using dual ADCs in DMA mode, which supply the real-time simulation engine.

The temperature computation relies on a finite-difference solution to the two-dimensional Laplace equation, whose physical model is determined by equation (1). The spatial domain of the plate is discretized into a 5×5 grid. Boundary conditions at the outermost points are updated with thermistor values, while internal nodes are updated iteratively using Liebmann’s method while computing equations (7) to (9). It is worth noting that $\lambda = 1.5$ was experimentally determined to balance convergence speed and numerical stability.

Figure 5 shows the experimental setup used during the real-time tests. Figure 5 (a) depicts a close-up of the aluminum plate, which was equipped with calibrated thermistors distributed at the outer edges, secured with thermal tape, and wired to dual ADC channels configured in DMA mode. These sensors captured the boundary conditions in real time, while the interior temperature values were calculated iteratively on the microcontroller. A dedicated thermistor, not used by the solver, was placed in the center of the plate to act as a reference or witness sensor. This element enabled direct comparison between the computed and actual temperatures at the core of the domain.

To create temperature gradients and assess how the system responds to disturbance, the plate was covered with thermal tape as shown in Figure 3 (b) and exposed to external heat sources using alcohol lamps. This disturbance helped validate the solver’s ability to reach thermal equilibrium and accurately reconstruct the temperature field despite uneven and dynamic heating. Figure 4 (b) and (c) show the thermal output images captured by a thermal camera and by the digital twin.

After convergence was achieved, the data were visualized using a custom Python graphical interface that receives the reconstructed temperature vector from the microcontroller. Figure 13 shows the resulting thermal field, highlighting the interpolated internal temperatures and comparing the actual reference value ($31.19\text{ }^{\circ}\text{C}$) with the estimated center temperature ($29.43\text{ }^{\circ}\text{C}$), which results in an observed error of $+1.76\text{ }^{\circ}\text{C}$. These initial results confirm that the embedded solver can reliably approximate the spatial thermal gradient using only boundary data, validating the methodology for further testing and adaptation of domain-specific applications.

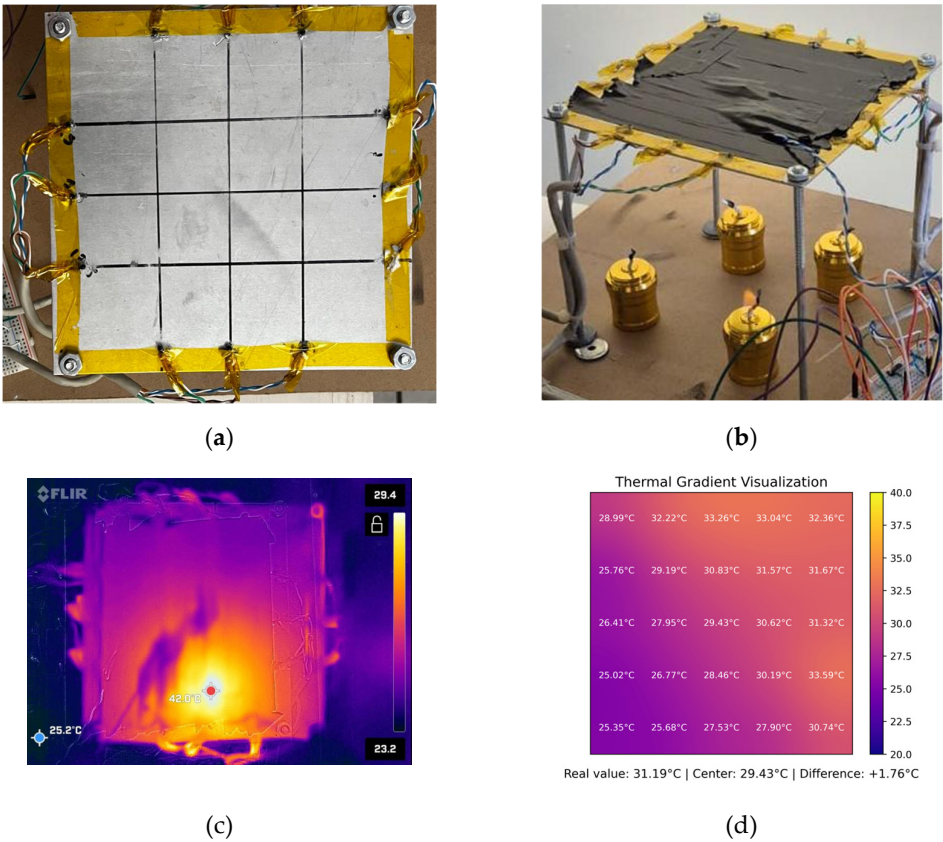


Figure 5. Experimental setup used during real-time tests; (a) close-up view of the sensor array; (b) external heat sources applied to the plate; (c) thermal images captured by a camera to assess the temperature gradient as an initial estimate, with a control sensor located at the center of the plate; and (d) gradient image computed by the digital twin.

5.4. Embedded Schedule Logic

Once both ADCs complete their DMA transfers, flags are raised, triggering the reconstruction of boundary temperature values from the acquired voltages using the Steinhart–Hart model. These values are mapped onto the external edges of the discretized domain. The interior of the grid is then updated according to the numerical solver logic.

This real-time loop is executed within the main application thread and consists of the following logical steps:

1. DMA Completion Detection, where the embedded system waits for both ADC1 and ADC2 to complete DMA transfers.
2. Boundary Assignment, where raw ADC values are converted into temperatures and assigned to border positions.
3. Finite Difference Iteration, where the temperature of internal nodes is updated via the iterative solver.
4. UART Transmission, where the resulting 5×5 temperature grid is serialized and transmitted to a host interface.
5. DMA Restart, where the ADC readings are re-triggered to maintain a continuous acquisition cycle.

The previously mentioned logic sequence steps are expressed in Algorithm 1 as follows.

Algorithm 1. Embedded Schedule Logic

WHILE True:

IF adc1_done AND adc2_done:

boundary_values = convert_adc_to_temp(adc1_data, adc2_data)

assign_boundary_conditions(grid, boundary_values)

FOR i = 1 to L-2:

FOR j = 1 to L-2:

T_new = average_neighbor_temperatures(grid, i, j)

T_relaxed = $\lambda T_{\text{new}} + (1-\lambda)T_{\text{old}}[i][j]$

grid[i][j] = T_relaxed

send_grid_via_UART(grid)

restart_ADC_DMA()

5.5. Real-Time Thermal Visualization Interface

5.5.1. Symmetrical Perturbations Experiment

To enable real-time visual feedback of the computed temperature field, a host-side visualization interface was implemented using Python and the Matplotlib library, using an interface that communicates with the embedded system via UART over a serial connection. The embedded system transmits a linear array of 25 temperature values, corresponding to a 5×5 mesh representation of the metallic plate, at each iteration. This resolution can be modified depending on the number of boundary sensors used and the resolution required to represent the full internal temperature field estimated by the microcontroller.

Upon receiving a complete set of values, the Python program reshapes the flat vector into a 2D array, applies a light Gaussian filter for spatial smoothing, and then updates the heatmap in real time.

Each cell in the mesh displays the temperature value with a white label overlay, providing both qualitative and quantitative insight into the gradient evolution. Algorithm 2 executes the logic behind the visualization interface.

Algorithm 2. Real-Time Thermal Visualization Interface

Initialize serial port and 5×5 temperature matrix
Initialize interactive heatmap using matplotlib

Loop indefinitely:
Wait for 25 valid float values from serial input
If complete and valid data received:
Reshape into 5×5 matrix
Apply Gaussian smoothing
Update color map and text labels on plot

To assess the fidelity of the internal temperature estimation computed by the embedded system, a temporal analysis was conducted by comparing the simulated temperature at the center of the plate (T_{13}) against a calibrated reference sensor placed at the same location. In the experimental setup, four flame sources were applied symmetrically at the edges of the metallic plate for 10 minutes, establishing boundary conditions that propagated inward. The evolution of T_{13} over time was plotted alongside the reference temperature, revealing that the embedded solver effectively captured the general heating trend and dynamic response of the plate (Figure 6a). However, discrepancies emerged after the initial transient phase, with the absolute error stabilizing around 3–4°C once the temperature field reached a steady state. The real-time error was also calculated and visualized, providing a quantitative measure to evaluate the accuracy of the internal node approximation compared to the physical sensor (Figure 7).

Further insight is shown in Figure 6c and Figure 6d by the reconstructed thermal gradient visualizations generated by the digital twin system at three representative time points. After 31.5 seconds from the start of the experiment (Figure 6a and 6c, sample 63), the plate exhibited relatively low and homogeneous temperatures, with the estimated center temperature (26.71 °C) closely matching the reference value (27.46 °C), resulting in a minimal absolute error of 0.75 °C (Figure 7). As the heat propagated from the boundaries (sample 660), the internal field developed a stronger gradient, and the estimated center value (56.06 °C) began to deviate from the measured 60.82 °C, yielding a larger error discrepancy of 4.76 °C (Figure 7). By the final stage (sample 1190), the system approached thermal equilibrium, and although the center temperature estimate improved to 58.73 °C, a residual error of 3.58 °C persisted with respect to the reference (62.31 °C) as displayed by Figures 6a and 6d and validated by the error performance behavior.

To further understand the dynamics of thermal conduction across the plate, a focused analysis was conducted along a linear path from the top edge (T2) toward the center (T13), passing through intermediate points (T7, T12). The temporal evolution of these points revealed the progressive inward diffusion of heat, where each subsequent point exhibited a delayed rise in temperature corresponding to its distance from the boundary. This edge-to-center propagation curve not only validates the expected physical behavior dictated by Fourier’s law but also serves as a diagnostic trajectory to assess how well the model captures the transient performance of thermal conduction (Figure 6b). The embedded system successfully mirrored this progression, though with minor temporal offsets and underestimations at intermediate points, highlighting the limitations of coarse spatial discretization. Nevertheless, the shape and sequence of thermal response closely matched expectations, reinforcing the validity of the model.

It is important to note that the numerical solver implemented in the embedded system is based on Laplace’s equation in two dimensions, as expressed in Equation (8). This formulation assumes steady-state heat conduction, where no internal heat generation or time-dependent changes are present. As such, the model inherently neglects transient thermal behaviors during the early stages of heating and converges toward an equilibrium solution as the system stabilizes. This theoretical foundation explains the high fidelity of the spatial distribution at later times, but also the initial discrepancies observed when the plate is still undergoing dynamic thermal transitions. The fidelity of the embedded approximation thus depends not only on spatial resolution and boundary accuracy, but also on the validity of the steady-state assumption over the timescales of interest.

Moreover, the experiment was deliberately conducted in a real laboratory environment subject to uncontrolled disturbances, such as external airflow from opened doors and exposure to air conditioning currents. These perturbations were intentionally preserved during the trials to evaluate the model’s robustness under non-ideal conditions, as typically encountered in industrial environments. While more accurate results could be obtained under strictly controlled thermal conditions, the current findings highlight the resilience of the embedded model in the presence of realistic environmental noise. This characteristic is essential for practical deployment, where such fluctuations are unavoidable. The ability to accurately reflect thermal propagation and gradient formation under such disturbances further demonstrates the robustness and applicability of the proposed system in real-world settings. The thermal gradients corresponding to the beginning and the end of the 10-minute test are displayed by the digital twin and are shown in Figure 6 (c)–(d). Figure 7 illustrates the error performance throughout the experiment duration.

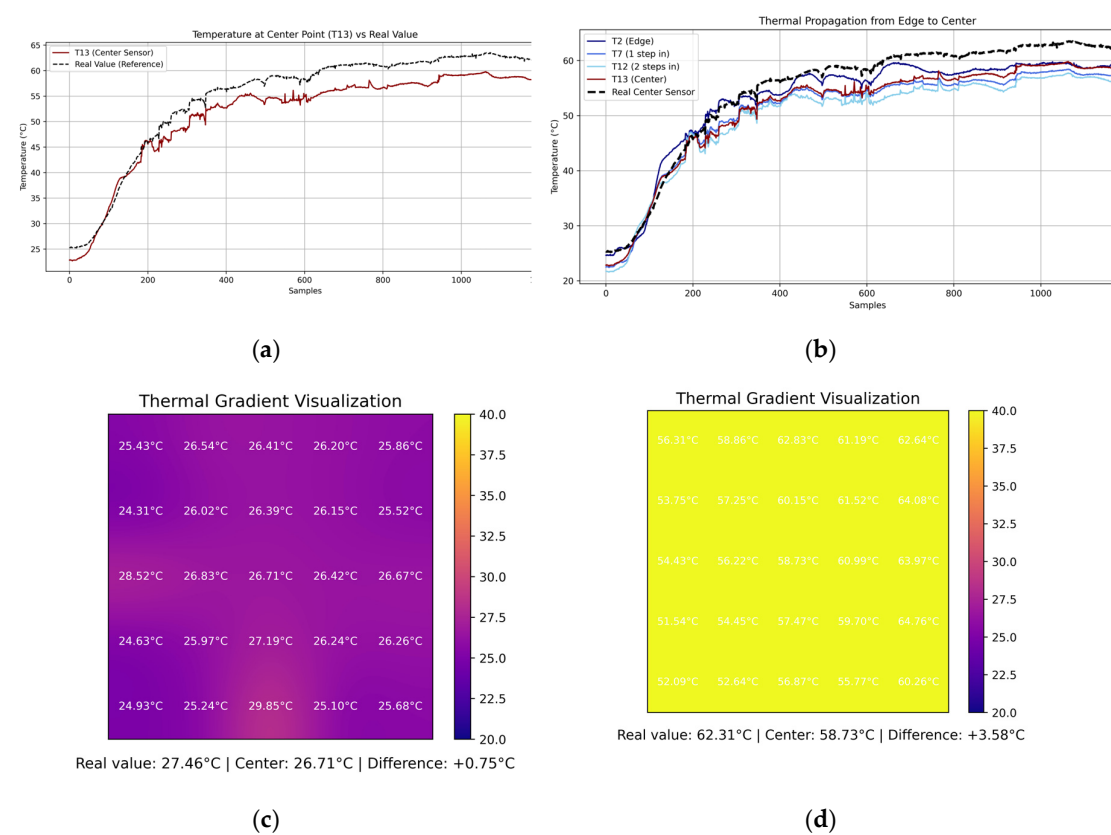


Figure 6. Symmetrical Perturbations Experiment.

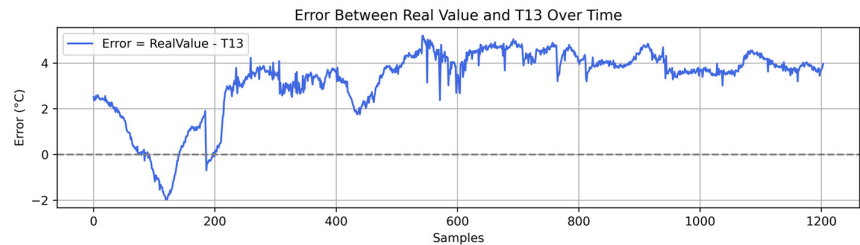


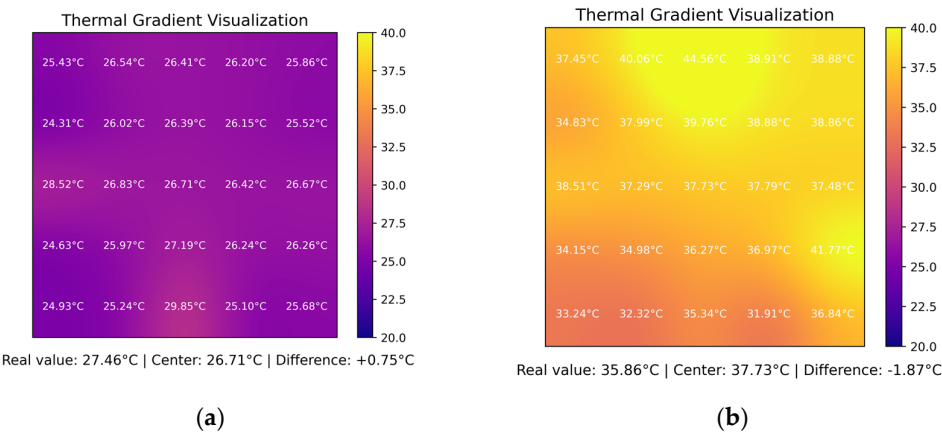
Figure 7. Error performance between the estimated temperatures and the real value at the center of the plate.

Figure 8 presents the output of the digital twin graphic interface as a series of thermal snapshots captured during the first three minutes of the experiment under symmetric heating conditions. In this configuration, four alcohol flame sources were placed at each edge of the metallic plate, simulating an evenly distributed thermal excitation.

Each frame corresponds to a specific time mark: 30 s (sample 60), 60 s (sample 120), 90 s (sample 180), and 120 s (sample 240) and shows the estimated temperature field across the 5×5 mesh as reconstructed by the embedded system solver. The color scale is fixed between 20 °C and 40 °C for consistency, while each cell displays its corresponding temperature in degrees Celsius. Additionally, the real sensor reading at the physical center of the plate is displayed below each frame, along with the computed estimate (T13) and the instantaneous estimation error.

The sequence reveals a gradual and symmetric thermal propagation from the edges toward the center of the plate. At 30 seconds, the boundary nodes are already responding to the applied heat, while the central region remains cooler. As time progresses, the isothermal curves become more centralized and rounded, indicating a diffusive propagation pattern consistent with two-dimensional conduction. By the 150-second mark, the internal temperature field is approaching a quasi-stationary state, with reduced thermal gradients between adjacent nodes.

The comparison between the estimated central value and the physical sensor confirms that, during this interval, the embedded digital twin provides a reasonably accurate approximation of the true thermal state, with deviations generally below ± 3 °C. These visualizations validate the ability of the system to reconstruct the internal thermal profile from boundary-only measurements, highlighting the relevance of this approach for real-time diagnostics and heat distribution monitoring in embedded systems.



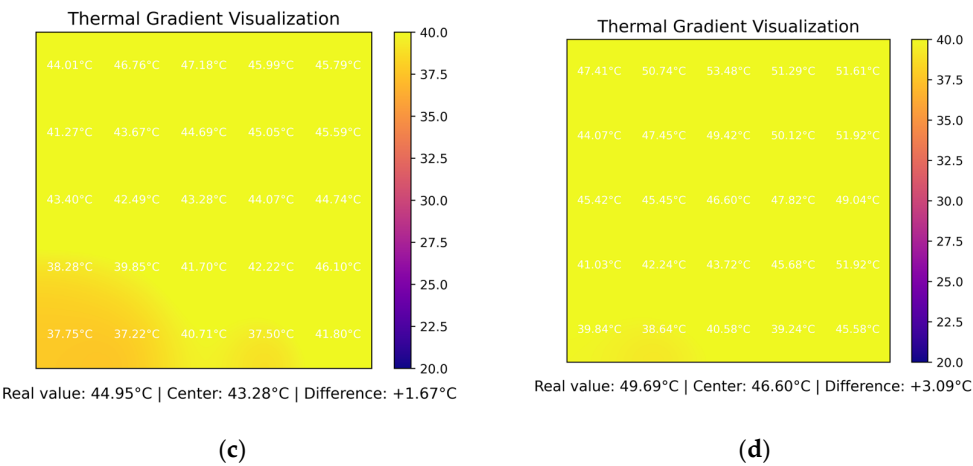


Figure 8. Snapshots of the digital twin graphical interface captured during the initial three minutes of the experiment.

5.5.2. Asymmetrical Perturbations Experiment

To investigate the response of the thermal estimation system under asymmetric excitation conditions, an experimental run was conducted in which only a single heat source was applied to one edge of the metallic plate. This setup was designed to evaluate how the internal estimation model, based on the Laplace equation, reacts to directional and non-uniform heat propagation.

In this experiment, the thermal evolution at the center of the plate was analyzed by tracking both the estimated temperature value computed from the boundary conditions (Figure 9a) and the physically measured value acquired through an embedded thermistor located at the geometric center of the plate. Both values were monitored continuously over a ten-minute interval, with uniform sampling assumed across the duration. Two snapshots were taken from the digital twin output to visualize the thermal performance, which are shown in Figures 9c and 9d.

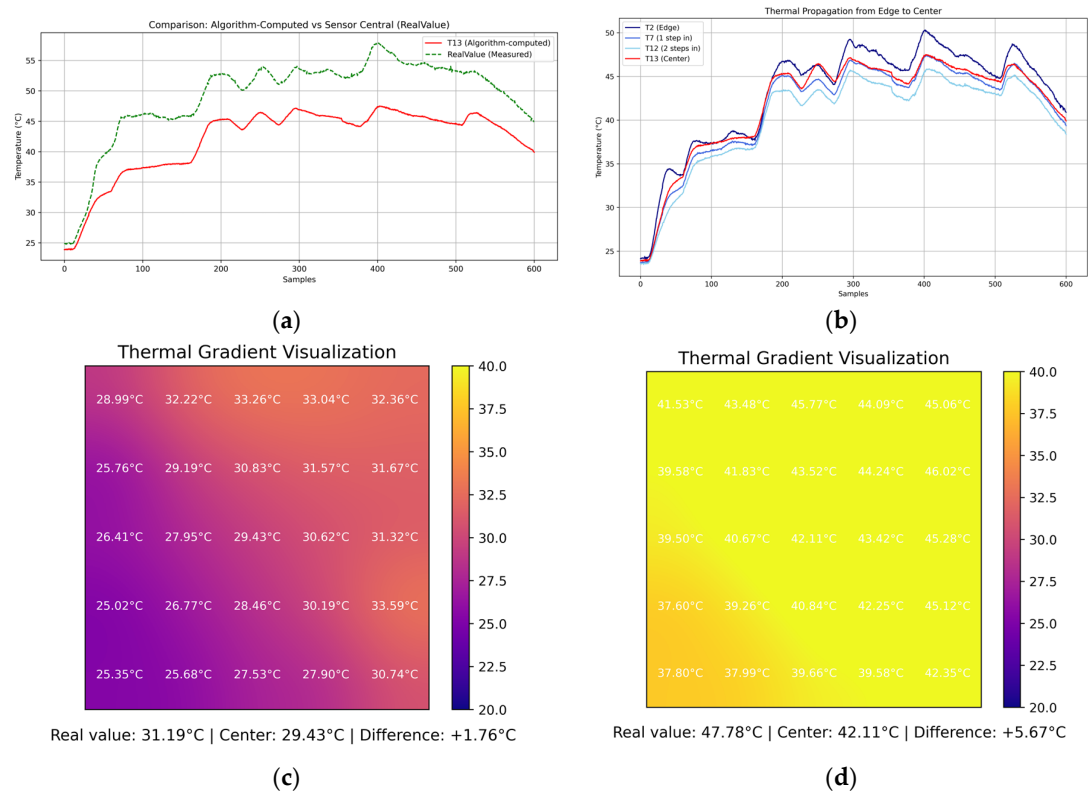


Figure 9. Snapshots of the digital twin graphical interface captured during the initial three minutes of the experiment.

The comparative analysis revealed that, while the Laplacian model provided a reasonable early-stage approximation, it increasingly underestimated the central temperature as the experiment progressed. This discrepancy is attributable to the inability of the stationary Laplace model to capture the dynamic accumulation of heat in the central region, which is influenced by the sustained unidirectional energy input from the single heat source.

Additionally, the thermal propagation from the edge to the center was quantified by examining the temporal evolution of a sequence of internal nodes aligned along the heat flow path, from the heated edge T2 through intermediate positions T7, T12, to the center T13 (Figure 9b). The resulting temperature profiles confirmed a delayed and diffused response consistent with one-dimensional heat conduction through a conductive medium, validating the system's ability to visualize directional thermal gradients.

This asymmetric heating scenario demonstrates the utility of the embedded estimator not only for real-time visualization but also as a diagnostic tool to study spatiotemporal thermal behavior under non-ideal conditions, offering insight into the limitations of steady-state models when exposed to dynamic or localized stimuli.

Finally, a vector field representation of the thermal gradient is included in Figure 10 to further enhance the approach's capability by showing both the direction and coherence of heat flow across the plate. In cases of asymmetric excitation, the vector field illustrates how thermal energy diffuses unevenly, deviating from perfect radial symmetry and converging toward the center in a skewed pattern. This directional insight, which is not obvious from scalar heatmaps alone, allows for a more detailed understanding of how localized boundary conditions influence the internal thermal landscape. Furthermore, the convergence of vector trajectories toward the sensor positioned at the center offers an intuitive validation of the spatial consistency of the embedded estimator. Additionally, it indicates potential future applications in thermal source localization and anomaly detection, particularly under irregular heating conditions.

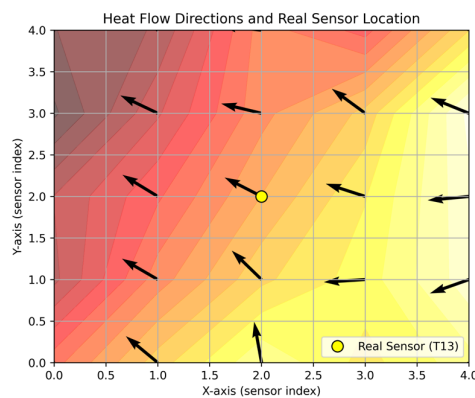


Figure 10. Vector field of normalized heat gradients under asymmetric edge heating. Arrows indicate the direction of heat flow, revealing anisotropic propagation toward the center sensor T13, highlighted in yellow.

6. Discussion

The implementation of a physics-informed digital twin for estimating real-time thermal distribution in metallic structures demonstrates both the feasibility and educational value of the proposed approach using mainstream microcontrollers. The embedded system can reconstruct internal thermal states by utilizing the steady-state Laplace equation and Dirichlet boundary conditions obtained from calibrated thermistors, achieving high temporal resolution with low computational overhead.

One of the advantages of this approach lies in its reliance on first-principles modeling, which ensures the physical consistency and interpretability of results without needing extensive training datasets typical of purely data-driven methods. The Laplace solver employs the Jacobi method with over-relaxation, achieving stable solutions in a limited number of iterations, making it suitable for real-time applications.

The visualization layer built in Python allows users to see results similar to how they would through a thermal camera, turning a physical setup into an interactive, real-time platform. This pairing of actual hardware with its simulated version creates a digital twin, effectively linking theoretical ideas with practical application.

However, several challenges can be pointed out. First, the spatial resolution is inherently constrained by the number of boundary sensors and the grid size that can be implemented within the available SRAM memory. Although a 5×5 grid proved sufficient for demonstrating thermal gradients, finer discretization would require optimization in memory allocation or the use of external RAM. Additionally, using floating-point arithmetic on microcontrollers can lead to precision and performance issues in certain setups. This work used a microcontroller with built-in hardware support for floating-point operations, which avoided these limitations. However, other platforms may not offer this support, and in such cases, convergence and accuracy may suffer, particularly when dealing with sharp thermal gradients or meeting industrial accuracy standards.

Sensor calibration with the Steinhart–Hart model was performed effectively in this laboratory setup to maintain accuracy. However, thermistor nonlinearity and potential thermal lag at the boundaries may result in minor deviations in the reconstructed data values. Operating directly at the edge computing level, where sensing, computation, and decision-making occur on the embedded platform itself, minimizes latency and reduces data transmission requirements. However, this architecture also imposes strict constraints on synchronization and timing. While utilizing DMA significantly reduces CPU overhead during ADC data acquisition, it presents timing challenges that must be addressed to maintain coherence between measurement and computation. In this implementation, interrupt-driven synchronization routines were required to precisely manage the capture and update cycles of both DMA and ADC peripherals, ensuring reliable real-time performance within the embedded environment.

From a systems perspective, the modular design of the embedded digital twin, which encompasses acquisition, computation, communication, and visualization components, presents opportunities for future enhancements. The anticipated improvements may involve real-time actuation (e.g., cooling systems), cloud-based logging, or hybrid physics-informed/machine learning models that adapt parameters over time.

From the practitioner's point of view, the embedded digital twin framework presented in this study validates the practicality of real-time simulation of partial differential equations on resource-constrained platforms. Its affordability, ability to replicate, and notable experimental value make it a valuable resource for industrial diagnostics and engineering education.

7. Conclusions and Future Work

This work presented a physics-informed digital twin architecture capable of estimating and visualizing the steady-state temperature distribution over a metallic plate in real-time. The system integrates boundary temperature measurements obtained through experimentally calibrated thermistors, a numerical solver based on the Laplace equation using the finite difference method, and an embedded implementation on a low-power mainstream microcontroller using DMA-driven ADC acquisition. The computed thermal map is transmitted and visualized through a Python interface, allowing an intuitive interpretation of the results.

The findings showed that real-time physics-based modeling can be effectively implemented on resource-limited embedded systems while maintaining accuracy and responsiveness. The Jacobi method's use of overrelaxation enabled rapid convergence while maintaining simplicity of

implementation. This approach is modular, extensible, and low-cost, making it ideal for educational settings and preliminary prototyping of thermal systems.

Several future directions are anticipated. First, increasing spatial resolution through adaptive meshing or memory optimization techniques would enhance model fidelity. Second, expanding the framework to manage dynamic thermal behavior would enable modeling of heat propagation and thermal inertia. Third, integrating actuation mechanisms based on feedback control from the reconstructed thermal field would close the loop and facilitate autonomous thermal regulation. Finally, hybrid approaches that combine physical modeling with lightweight machine learning techniques could improve robustness against sensor noise and parameter uncertainty.

Author Contributions: Conceptualization, V.H.B.; methodology, V.H.B.; software, V.H.B. and J.P.; validation, V.H.B. and J.P.; formal analysis, V.H.B.; investigation, V.H.B.; resources, V.H.B. and A.B.; data curation, V.H.B. and J.P.; writing—original draft preparation, V.H.B.; writing—review and editing, J.P. and A.B.; visualization, V.H.B. and J.P.; supervision, V.H.B. and A.B.; project administration, V.H.B. and A.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study are available from the corresponding author upon reasonable request. Due to the nature of the experimental setup and institutional policies, the dataset is not publicly archived but can be shared for research and academic purposes upon direct communication with the authors.

Acknowledgments: The authors would like to express their sincere gratitude to the technical and academic staff of the Control Laboratory at the Universidad de Sonora for their valuable support during the experimental setup and execution stages of this study. In particular, the contributions of Leonardo Sandoval, Isaac Samaniego, Rodrigo Reyes, and Manuel Plumedá, who participated as part of their social service commitment, were essential for the successful implementation, calibration, and testing of the embedded thermal monitoring system. Their assistance in hardware integration and experimental validation significantly contributed to the quality of the results obtained.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. D. Menges, F. Stadtmann, H. Jordheim, and A. Rasheed, "Predictive Digital Twin for Condition Monitoring Using Thermal Imaging," Nov. 2024, Accessed: Apr. 15, 2025. [Online]. Available: <https://arxiv.org/abs/2411.05887v1>
2. M. Shahinmoghdam, W. Natephra, and A. Motamedi, "BIM- and IoT-based virtual reality tool for real-time thermal comfort assessment in building enclosures," *Build Environ*, vol. 199, p. 107905, Jul. 2021, doi: 10.1016/j.BUILDENV.2021.107905.
3. A. Papacharalampopoulos and P. Stavropoulos, "Towards a Digital Twin for Thermal Processes: Control-centric approach," *Procedia CIRP*, vol. 86, pp. 110–115, Jan. 2019, doi: 10.1016/j.PROCIR.2020.01.015.
4. M. Boukaf, F. Fadli, and N. Meskin, "A Comprehensive Review of Digital Twin Technology in Building Energy Consumption Forecasting," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3498107.
5. H. Zhang, S. Jiang, D. Gao, Y. Sun, and W. Bai, "A Review of Physics-Based, Data-Driven, and Hybrid Models for Tool Wear Monitoring," *Machines* 2024, Vol. 12, Page 833, vol. 12, no. 12, p. 833, Nov. 2024, doi: 10.3390/MACHINES12120833.
6. R. and W. R. and W. Z. Wang Zhijun and Cong, "Digital Twin for Power Load Forecasting," in *Proceedings of the 2023 International Conference on Wireless Communications, Networking and Applications*, M. A. and C. S. K. S. and L. X. Siarry Patrick and Jabbar, Ed., Singapore: Springer Nature Singapore, 2025, pp. 366–374.

7. P. Norouzi, S. Maalej, and R. Mora, "Applicability of Deep Learning Algorithms for Predicting Indoor Temperatures: Towards the Development of Digital Twin HVAC Systems," *Buildings* 2023, Vol. 13, Page 1542, vol. 13, no. 6, p. 1542, Jun. 2023, doi: 10.3390/BUILDINGS13061542.
8. A. A. Dahalan, A. Saudi, A. A. Dahalan, and A. Saudi, "An iterative technique for solving path planning in identified environments by using a skewed block accelerated algorithm," *AIMS Mathematics* 2023 3:5725, vol. 8, no. 3, pp. 5725–5744, 2023, doi: 10.3934/MATH.2023288.
9. U. K. Dutta, A. Lipani, C. Wang, and Y. Hu, "Application of Zone Method based Physics-Informed Neural Networks in Reheating Furnaces," Aug. 2023, Accessed: Apr. 15, 2025. [Online]. Available: <https://arxiv.org/abs/2308.16089v2>
10. A. Di Meglio, N. Massarotti, and P. Nithiarasu, "A physics-driven and machine learning-based digital twinning approach to transient thermal systems," *Int J Numer Methods Heat Fluid Flow*, vol. 34, no. 6, pp. 2229–2256, Jul. 2024, doi: 10.1108/HFF-10-2023-0616/FULL/PDF.
11. Y. Xie, W. Yao, X. Li, N. Wang, X. Zheng, and X. Chen, "Hybrid digital twin for satellite temperature field perception and attitude control," *Advanced Engineering Informatics*, vol. 60, p. 102405, Apr. 2024, doi: 10.1016/J.AEI.2024.102405.
12. H. Zhang, S. Jiang, D. Gao, Y. Sun, and W. Bai, "A Review of Physics-Based, Data-Driven, and Hybrid Models for Tool Wear Monitoring," *Machines* 2024, Vol. 12, Page 833, vol. 12, no. 12, p. 833, Nov. 2024, doi: 10.3390/MACHINES12120833.
13. T. Chen, J. Botimer, T. Chou, and Z. Zhang, "A 1.87-mm² 56.9-GOPS Accelerator for Solving Partial Differential Equations," *IEEE J Solid-State Circuits*, vol. 55, no. 6, pp. 1709–1718, Jun. 2020, doi: 10.1109/JSSC.2019.2963591.
14. H. Malavipathirana, S. Mandal, N. Udayanga, Y. Wang, S. I. Hariharan, and A. Madanayake, "Analog Computing for Nonlinear Shock Tube PDE Models: Test and Measurement of CMOS Chip," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3524500.
15. "2-D Boundary Layer CFD Finite Difference Implicit Solver with Arduino | Arduino Project Hub." Accessed: Apr. 16, 2025. [Online]. Available: https://projecthub.arduino.cc/adrianos_botis/2-d-boundary-layer-cfd-finite-difference-implicit-solver-with-arduino-f7b4f1
16. J. Mu and B. Kim, "A Dynamic-Precision Bit-Serial Computing Hardware Accelerator for Solving Partial Differential Equations Using Finite Difference Method," *IEEE J Solid-State Circuits*, vol. 58, no. 2, pp. 543–553, Feb. 2023, doi: 10.1109/JSSC.2022.3174354.
17. Richard. Haberman, *Applied partial differential equations with fourier series and boundary value problems*, 5th ed. Pearson, 2014.
18. M. Grossi and M. Omaña, "Accuracy of NTC Thermistor Measurements Using the Sensor to Microcontroller Direct Interface," *Engineering Proceedings* 2024, Vol. 82, Page 12, vol. 82, no. 1, p. 12, Nov. 2024, doi: 10.3390/ECSA-11-20527.
19. "Releasing your creativity Discover the STM32 family of microcontrollers & microprocessors".
20. "AN4195 Application note STM32F30x ADC modes and application", Accessed: Apr. 19, 2025. [Online]. Available: www.st.com
21. "STM32 vs PIC32: Best Microcontroller for Manufacturing?—Titoma." Accessed: Apr. 20, 2025. [Online]. Available: https://titoma.com/blog/stm32-pic32-manufacturing/?utm_source=chatgpt.com
22. M. Samiullah, M. Z. Irfan, and A. Rafique, "Microcontrollers: A Comprehensive Overview and Comparative Analysis of Diverse Types," Sep. 2023, doi: 10.31224/3228.
23. "STM32F3 Mixed-Signal Microcontrollers (MCU)—STMicroelectronics." Accessed: Apr. 20, 2025. [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32f3-series.html?utm_source=chatgpt.com

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.