Custom Kernel GPR for Modeling Heat Equation Dynamics

By Hasanat Hasan

Heat conduction, the transfer of thermal energy through a material due to a temperature gradient, is a fundamental physical process described by the heat equation. This partial differential equation (PDE) models how temperature distribution evolves over time, driven by the tendency of systems to reach thermal equilibrium. It is a cornerstone of transport phenomena and serves as a prototypical parabolic PDE, finding analogies in diverse fields such as particle diffusion, fluid dynamics (viscous flow), quantum mechanics (the Schrödinger equation in imaginary time), and mathematical finance (e.g., the Black-Scholes model). In this project, we investigate the one-dimensional heat equation on a finite spatial domain $x \in [0, L]$:

$$\frac{\partial T(x,t)}{\partial t} = \alpha \frac{\partial^2 T(x,t)}{\partial x^2} \tag{1}$$

Here, $T(x,t)$ represents the temperature at position x and time t, and α is the thermal diffusivity, a material property indicating how quickly heat diffuses. We impose zero Dirichlet boundary conditions, $T(0,t) = T(L,t) = 0$, signifying that the ends of the domain are held at a constant zero temperature. The evolution is determined by an initial temperature profile $T(x,0) = f(x)$. For these conditions, the solution can be elegantly expressed using the method of separation of variables, leading to a Fourier sine series:

$$T(x,t) = \sum_{n=1}^{\infty} c_n \exp\left(-\alpha \left(\frac{n\pi}{L}\right)^2 t\right) \sin\left(\frac{n\pi x}{L}\right) \tag{2}$$

The coefficients c_n are determined by the initial condition $f(x)$ via the Fourier sine transform:

$$c_n = \frac{2}{L} \int_0^L f(x) \sin\left(\frac{n\pi x}{L}\right) dx \tag{3}$$

Equation 2 decomposes the solution into a superposition of spatial modes $\sin(n\pi x/L)$, each decaying exponentially in time at a rate determined by the eigenvalue $\lambda_n = \alpha(n\pi/L)^2$. Higher-frequency modes (larger n) decay more rapidly, leading to a progressive smoothing of the temperature profile as time evolves. These modes capture the diffusion of heat throughout the region, enabling a detailed understanding of how the temperature profile evolves under the given initial and boundary conditions.

Goal

The primary objective of this project is to develop and evaluate a data-driven model for predicting the spatial-temporal dynamics of the heat equation using Gaussian Process Regression (GPR). While the Introduction focused on the 1D heat equation (Equation 8) and its solution

(Equation 2) for illustrative purposes, a central aim of this work is to extend these concepts to two spatial dimensions.

Instead of relying solely on numerical discretization or generic regression techniques, we leverage the known analytical structures of the heat equation's solutions in both 1D and 2D to construct custom, physics-informed GPR kernels [1]. Building upon previous work that predicted spatial temperature profiles at a fixed time, this project significantly extends the approach by:

- Developing a kernel for the 1D case that incorporates both spatial (x) and temporal (t) coordinates, enabling the prediction of the full $T(t, x)$ field from sparse measurements (t_i, x_i, T_i).
- Deriving an analogous custom kernel for the 2D heat equation on a square domain, designed to handle sparse spatial-temporal measurements (t_i, x_i, y_i, T_i) and reconstruct the 2D temperature field $T(t, x, y)$.

We aim to demonstrate that by using sparse measurements sampled across space and time, the GPR models equipped with these tailored kernels can accurately reconstruct the temperature dynamics in both one and two dimensions. The custom kernels explicitly encode the correlations implied by the respective Fourier series solutions, particularly the exponential decay. The subsequent sections detail the derivation of these 1D and 2D kernels and present results demonstrating the predictive capabilities and limitations of this physics-informed GPR approach in both scenarios.

1D Temporal Reconstruction

Previous 1D Spatial Reconstruction

In assignment 10, we imbued the knowledge that

- The time evolution of the heat equation exponentially decays. We can encode this information into our prior-covariance matrix:

$$A^{-1}_{ss'} = \exp\left(-\alpha\left(\frac{n\pi}{L}\right)^2 t\right)\delta_{s,s'}$$

- The kernel/covariance function is a heuristic function representing how function values at different points are correlated and fully encodes our prior information.
- In the 1D heat equation case, we constructed a kernel using temperature measurements at a fixed time, t, for some $\vec{X} = [x_1, \ldots, x_N]$. It was supplied a dataset $[T(t, x_1), \ldots, T(t, x_N)]$. The kernel is given as follows:

$$K(X, X') = \langle f(X)f(X')\rangle = \sum_{s,s'=1}^{S} A^{-1}_{ss'} T_s(X) T_{s'}(X'),$$

  where $x \in (x_1, \ldots, x_N)$.

The goal of this project is, instead of sampling different temperatures at a fixed time to predict the entire temperature distribution at some time t, we will sample at different positions and times to predict the entire time evolution of temperature at a point. Now we will extend it to find the correlation between temperatures with different time measurements, $T(t_i, x_i)$ and $T(t_j, x_j)$. This means the prior no longer has an exponential term and should be 1 to represent maximum variance:

$$A^{-1}_{ss'} = \delta_{s,s'}.$$

Because $A^{-1}_{ss'}$ is diagonal, it is isomorphic to vectors in $\mathbf{R^N}$, so we can represent $Z = [A^{-1}_{1,1'}, \ldots, A^{-1}_{N,N'}]$. We want our kernel function to capture the correlation of the temperature measurements at (t, x) with $Z = [A^{-1}_{1,1'}, \ldots, A^{-1}_{N,N'}]$. Let us define $X = (t, x)$ and $X' = (t', x')$:

$$K(X, X') = K((t, x), (t', x')) = \langle f(X) f(X') \rangle \tag{4}$$

$$= \sum_{s,s'=1}^{S} Z_s \exp\left(-\alpha \left(\frac{s\pi}{L}\right)^2 t\right) \exp\left(-\alpha \left(\frac{s\pi}{L}\right)^2 t'\right) \sin\left(\frac{s\pi x}{L}\right) \sin\left(\frac{s\pi x'}{L}\right) \tag{5}$$

$$= \sum_{s=1}^{S} Z_s \exp\left(-\alpha \left(\frac{s\pi}{L}\right)^2 (t + t')\right) \sin\left(\frac{s\pi x}{L}\right) \sin\left(\frac{s\pi x'}{L}\right). \tag{6}$$

Thus, our resulting kernel function is given by:

$$K(X, X') = \sum_{s=1}^{S} \exp\left(-\alpha \left(\frac{s\pi}{L}\right)^2 (t + t')\right) \sin\left(\frac{s\pi x}{L}\right) \sin\left(\frac{s\pi x'}{L}\right).$$

Given the dataset $(t_\alpha, x_\alpha, T(t_\alpha, x_\alpha), \sigma^2_\alpha)$, the entry of the kernel matrix given the data is:

$$\bar{K}_{i,j} = K(X, X') + \sigma^2_x \delta_{i,j} + \sigma^2_t \delta_{i,j} = K(X, X') + \sigma^2 \delta_{i,j},$$

where we combined the noise: $\sigma^2 = \sigma^2_x + \sigma^2_t$. We therefore have our posterior expectation of our measurement:

$$\langle T(X) \rangle = \sum_{i,j} K(X, X'_i)[\bar{K}]^{-1}_{i,j} T_j.$$

and the posterior variance is given by:

$$\sigma^2_{\text{post}}(X) = K(X, X) + \sum_{i,j} K(X, X'_i) \bar{K}^{-1}_{i,j} K(X'_j, X)$$

$$\sigma^2_{\text{post}}((t, x)) = K((t, x), (t, x)) + \sum_{i,j} K((t, x), (t_i, x_i)) \bar{K}^{-1}_{i,j} K((t_j, x_j), (t, x)) \tag{7}$$

Program Manual: GPR Implementation

This section outlines the computational algorithm and core Python functions used to implement the Gaussian Process Regression model with custom kernels for predicting heat equation dynamics in both 1D and 2D. These implementations are provided in their respective files, `1D_Temp_dynamics.ipynb` and `2D_Temp_dynamics.ipynb`. The implementation relies heavily on NumPy for efficient vectorized operations.

Algorithm Overview

The general workflow for generating predictions using the physics-informed GPR model is as follows:

1. **Parameter Definition:** Define physical parameters (L, α), GPR model parameters (number of modes N, noise variance σ^2), and simulation parameters (time range, number of points, etc.).

2. **Measurement Data Generation (Simulation):**

   - Select sparse measurement coordinates $X_i = (t_i, x_i)$ (for 1D) or $X_i = (t_i, x_i, y_i)$ (for 2D).
   - Define coefficients (c_n or B_{nm}) for the analytical solution to represent a "true" underlying temperature field.
   - Calculate the exact temperature $T_{true}(X_i)$ at the measurement coordinates using the analytical solution (e.g., via `evolve_temp` or `compute_heat_solution_2d`).
   - Generate measurement data $T_i = T_{true}(X_i)$

3. **Kernel Matrix Computation:**

   - Calculate the prior covariance matrix K, where $K_{ij} = K(X_i, X_j)$, using the appropriate custom kernel function (`kernel_matrix` for 1D, `kernel_matrix_2d` for 2D) applied to the set of measurement coordinates $X_1, ..., X_P$.
   - Compute the data covariance matrix $\bar{K} = K + \sigma^2\mathbf{I}$ using the `data_covar` (or `data_covar_2d`) function. This matrix incorporates the measurement noise.

4. **Prediction:**

   - Define a set of prediction coordinates X_* where temperature estimates are desired.
   - Calculate the covariance matrix from data K_*, where $[K_*]_{pi} = K(X_{*,p}, X_i)$, between prediction points and measurement points using the relevant kernel matrix function.
   - Compute the mean posterior prediction $\langle T(X_*)\rangle = K_*\bar{K}^{-1}\,T$ using the `mean_prediction` (or `mean_prediction_2d`) function, where T is the vector of measurements T_i.

5. **Uncertainty Quantification:** Calculate the posterior variance $\sigma^2_{\text{post}}(X_*)$ at the prediction points X_* to quantify confidence in the mean prediction. The calculation involves:

   - **Prior Variance:** Compute the prior variance at each prediction point,

     $$K(X_{*,p}, X_{*,p}),$$

     using the appropriate kernel function (`kernel_func` or `kernel_func_2d`). This requires a kernel function that evaluates individual pairs of points, not just matrices. Let K_{**} denote the vector containing all diagonal elements $K(X_{*,p}, X_{*,p})$.

- **Variance Reduction:** Compute the reduction in variance due to measurement information. This term is derived from the diagonal elements of the matrix:

$$K_* \bar{K}^{-1} K_*^T,$$

where K_* is the cross-covariance matrix and $\bar{K}^{-1}$ is the inverse of the data covariance matrix. In the implementation (`GPR_prediction`), this is efficiently computed as:

`variance_reduction = np.sum((K_star_matrix @ Kbar_inv) * K_star_matrix, axis=1)`,

where `K_star_matrix` corresponds to K_* and `Kbar_inv` to $\bar{K}^{-1}$.

- **Posterior Variance:** The posterior variance is given by:

$$\sigma^2_{\text{post}}(X_{*,p}) = k_p - [K_* \bar{K}^{-1} K_*^T]_{pp}.$$

Note: While the function `GPR_prediction` performs both mean and variance calculations, the core functions described (`mean_prediction`, etc.) separate these steps for clarity in presentation and implementation. The posterior standard deviation is the square root of the posterior variance:

$$\sigma_{\text{post}}(X_*) = \sqrt{\sigma^2_{\text{post}}(X_*)}.$$

Core Function Descriptions

The key Python functions implementing this algorithm are:

- `kernel_matrix(Length, alpha, modes, coord1, coord2)` (1D) / `kernel_matrix_2d(Length, alpha, modes, coord1, coord2)` (2D):
  - **Purpose:** Computes the kernel matrix K between two sets of input coordinates.
  - **Equation (1D):**

$$[K]_{ij} = \sum_{s=1}^{N} \exp\left(-\alpha \left(\frac{s\pi}{L}\right)^2 (t_i + t_j)\right) \sin\left(\frac{s\pi x_i}{L}\right) \sin\left(\frac{s\pi x_j}{L}\right)$$

  - **Equation (2D):**

$$[K]_{ij} = \sum_{n=1}^{N} \sum_{m=1}^{M} \exp\left(-\alpha \lambda_{nm} (t_i + t_j)\right) \sin\left(\frac{n\pi x_i}{L}\right) \sin\left(\frac{m\pi y_i}{L}\right) \sin\left(\frac{n\pi x_j}{L}\right) \sin\left(\frac{m\pi y_j}{L}\right)$$

    where $\lambda_{nm} = \left(\frac{\pi}{L}\right)^2 (n^2 + m^2)$.
  - **Notes:** Accepts lists of coordinate arrays (e.g., `[t_array, x_array]`) as input. Uses NumPy broadcasting for efficient computation over all pairs of points and summation over modes.

- `data_covar(..., meas_points, sigma, kernel_matrix_func)` (1D) / `data_covar_2d(...)` (2D):
  - **Purpose:** Computes the data covariance matrix $\bar{K}$ by adding noise variance to the diagonal of the Gram matrix.
  - **Equation:**
    $$\bar{K} = K + \sigma^2 \mathbf{I},$$
    where K is computed by calling the passed `kernel_matrix_func` with `meas_points` as both input coordinate sets.
  - **Notes:** Takes the measurement noise standard deviation `sigma` as input.
- `mean_prediction(..., prediction_coords, measurement_coords, Kbar, meas_temps, kernel_matrix_func)` (1D) / `mean_prediction_2d(...)` (2D):
  - **Purpose:** Computes the mean posterior prediction $\langle T(X_*) \rangle$.
  - **Equation:**
    $$\langle T(X_*) \rangle = K_* \bar{K}^{-1} \mathbf{T}$$
    where K_* is the cross-covariance matrix between the prediction and measurement coordinates, $\bar{K}$ is the data covariance matrix, and $\mathbf{T}$ is the vector of measured temperatures.
  - **Notes:** Calculates the cross-covariance K_* using the passed `kernel_matrix_func` between prediction and measurement coordinates. Performs matrix inversion of $\bar{K}$ (or solves the equivalent linear system) and matrix multiplication to get the final predictions.
- `evolve_temp(coeffs_list, modes, Positions, times, length, alpha)` (1D Analytical Solution) / `compute_heat_solution_2d(...)` (2D Analytical Solution):
  - **Purpose:** Calculates the analytical solution to the heat equation based on the Fourier series expansion.
  - **Equation (1D):** $T(x,t) = \sum_{n=1}^{N} c_n \exp\left(-\alpha \lambda_n t\right) \sin\left(\frac{n\pi x}{L}\right)$
  - **Equation (2D):** $T(x,y,t) = \sum_{n=1}^{N} \sum_{m=1}^{M} B_{nm} \exp\left(-\alpha \lambda_{nm} t\right) \sin(\frac{n\pi x}{L}) \sin(\frac{m\pi y}{L})$.
  - **Notes:** Used primarily for generating synthetic ground truth data and for comparison with GPR predictions. Vectorized using NumPy for efficiency. The `evolve_temp` function has helper functions (`heat_decay`, `heat_field`, `heat_single_point`) for different input types.

Subtlties Between 1 and 2 Dimensions

The core GPR algorithm (`data_covar`, `mean_prediction`) remains the same for both 1D and 2D problems. The primary difference lies in:

- The input coordinates passed to the functions (e.g., ‘[t, x]‘ vs ‘[t, x, y]‘).
- The specific kernel function used (`kernel_matrix` vs `kernel_matrix_2d`), which implements the mathematically distinct kernel derivation for each dimensionality.
- The analytical solution function used for data generation/comparison (`evolve_temp` vs `compute_heat_solution_2d`).

The Python code is structured to easily switch between dimensions by calling the appropriate kernel and analytical solution functions.

Results and Analysis: 1D Temperature Reconstruction

The effectiveness of the custom kernel GPR model for reconstructing the spatial-temporal dynamics of the 1D heat equation is evaluated using sparse measurements. Figure 1 illustrates the performance using 20 randomly sampled data points across the space-time domain $[0, L] \times [0, t_{max}]$ where $L = 1.0$, $t_{max} = 2$ seconds, thermal diffusivity $\alpha = 0.01$, and 50 modes in the kernel definition. The figure presents four key visualizations: the GPR predicted temperature field, the corresponding analytical solution, the posterior standard deviation field from the GPR, and the absolute error field between the prediction and the analytical solution.

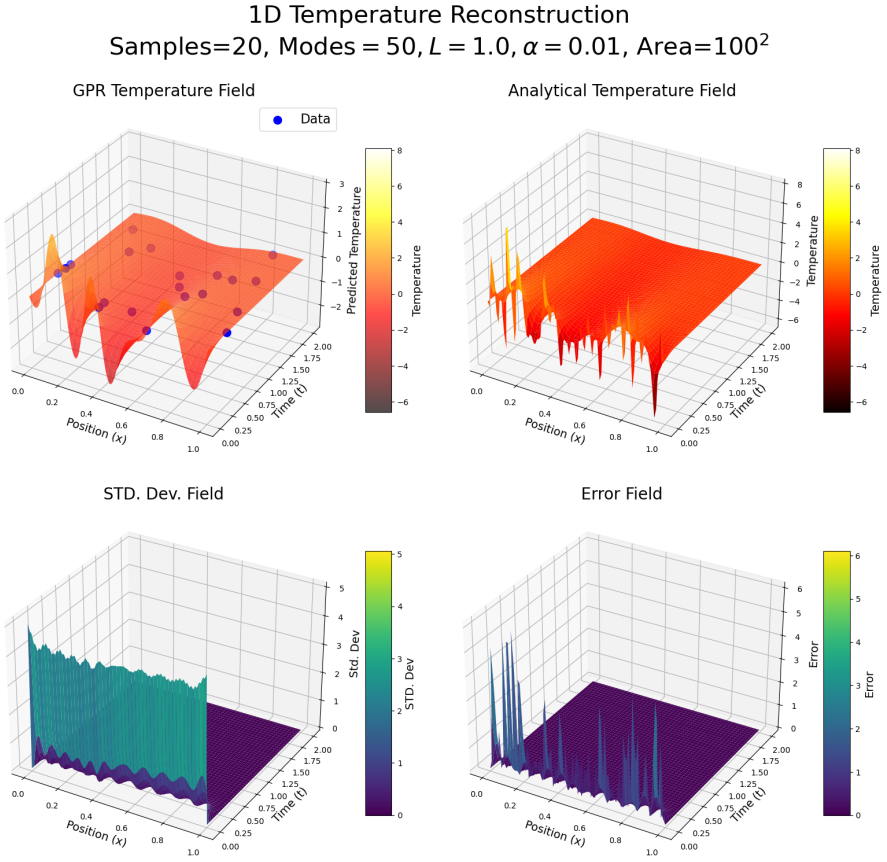

Figure 1: GPR reconstruction of 1D heat equation dynamics using 20 sparse measurements. Parameters: Modes= 50, $L = 1.0$, $\alpha = 0.01$. Top-left: GPR mean prediction with data points (blue). Top-right: Analytical solution. Bottom-left: GPR posterior standard deviation. Bottom-right: Absolute error field.

The top left panel displays the mean posterior prediction surface generated by the GPR model. The overlaid blue markers represent the 20 sparse (t, x) locations where temperature measurements were provided as input. The predicted surface demonstrates a smooth interpolation between these data points, constrained by the physics embedded within the custom kernel. Visually, the model captures the essential characteristics of heat diffusion: the temperature profile

decays over time, and the spatial structure exhibits the sinusoidal form inherent to the heat equation solution with zero boundary conditions.

The success of the model in reconstructing a plausible field from sparse data stems directly from the physics-informed kernel. The kernel (Equation 5) is constructed as a sum over modes, explicitly incorporating the knowledge that the solution consists of sinusoidal basis functions $\sin(s\pi x/L)$ and that each mode decays exponentially in time via the factor $\exp(-\alpha(s\pi/L)^2 t)$. This structure encodes strong prior knowledge about the solution space. By correlating points (t, x) and (t', x') using the term $\exp(-\alpha(s\pi/L)^2(t + t'))$, the kernel inherently understands the thermalization process – the tendency for high-frequency spatial modes (large s) to decay rapidly. However, the largest discrepancy between the prediction and the true solution (as seen in the Error Field, bottom-right) occurs near $t = 0$. This is because the exponential decay factor in the kernel is close to 1 when t and t' are small. Near $t = 0$, the kernel does not yet strongly suppress high-frequency modes based on time evolution alone. This allows a wider range of possible superpositions of modes (consistent with the kernel structure) that could potentially fit the sparse data points located near $t = 0$. The GPR has less temporal constraint here, leading to higher uncertainty (bottom-left) and potentially larger errors if the true initial condition has significant high-frequency components not fully captured by the sparse samples. As time progresses ($t > 0$), the exponential term in the kernel rapidly dampens high-frequency contributions, significantly reducing the space of plausible functions consistent with both the measurements and the physics. This forces the GPR prediction to converge quickly towards the smoother, thermalized solution observed at later times. The model successfully integrates the information from spatially and temporally scattered points, guided strongly by the encoded physics, to form a coherent spatio-temporal field.

Summary: Overall, Figure 1 demonstrates that the GPR model equipped with the physics-informed kernel can effectively learn the underlying dynamics of the 1D heat equation from sparse space-time data. It successfully captures the global diffusion behavior and spatial structure. The primary discrepancies arise in accurately reproducing very high-frequency components or sharp gradients present in the true solution, which is a common challenge when reconstructing complex fields from limited data. The model also provides valuable uncertainty estimates that correctly identify regions of lower confidence in the prediction. The performance suggests that incorporating the analytical structure of the solution into the kernel is a promising approach for data-driven modeling of PDEs.

Extension to 2D Heat Equation on a Square Domain

We now consider the 2D heat equation:

$$\frac{\partial T(t, x, y)}{\partial t} = \alpha \nabla^2 T(t, x, y) \tag{8}$$

with a square domain $[0, L] \times [0, L]$ with zero Dirichlet boundary conditions: $T(t, x, 0) = T(t, x, L) = T(t, 0, y) = T(t, L, y) = 0$. The analytical solution is found using separation of

variables:

$$T(t,x,y) = \sum_{n=1}^{\infty}\sum_{m=1}^{\infty} B_{nm} \exp\left(-\alpha\lambda_{nm}t\right) \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{m\pi y}{L}\right)$$

where the eigenvalues are $\lambda_{nm} = \left(\frac{\pi}{L}\right)^2 (n^2+m^2)$. Our goal is to predict the temperature field $T(t,x,y)$ using sparse measurements at different space-time coordinates (t_i, x_i, y_i). Let the input vector be $X = (t,x,y)$. The function we are modeling is $f(X) = T(t,x,y)$. We represent the function using its basis expansion:

$$f(X) = \sum_{n=1}^{N}\sum_{m=1}^{M} B_{nm}\phi_{nm}(X)$$

where N, M are truncation limits for the series and $N = M$ (due to boundary conditions), and the basis functions are:

$$\phi_{nm}(X) = \phi_{nm}(t,x,y) = \exp\left(-\alpha\left(\frac{\pi}{L}\right)^2 (n^2+m^2)t\right) \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{m\pi y}{L}\right)$$

We place a prior on the coefficients B_{nm}. Following the 1D case logic, we assume the coefficients are uncorrelated a priori with uniform variance. Let the prior covariance be:

$$\langle B_{nm} B_{n'm'} \rangle = A^{-1}_{(nm),(n'm')} = Z_{nm}\delta_{nn'}\delta_{mm'}$$

Similar to the final step in the 1D derivation, we set the prior variance $Z_{nm} = 1$ for all n, m. This implies that our prior belief assumes equal contribution potential from all modes before seeing data, and the structure (decay rates, spatial oscillations) is encoded entirely within the basis functions ϕ_{nm}. Now we compute the kernel function $K(X, X')$, where $X = (t,x,y)$ and $X' = (t',x',y')$. The kernel represents the expected covariance between function values at these two points under the prior:

$$\begin{aligned}
K(X,X') &= \langle f(X) f(X') \rangle \\
&= \left\langle \left(\sum_{n=1}^{N}\sum_{m=1}^{M} B_{nm}\phi_{nm}(X)\right)\left(\sum_{n'=1}^{N}\sum_{m'=1}^{M} B_{n'm'}\phi_{n'm'}(X')\right)\right\rangle \\
&= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{n'=1}^{N}\sum_{m'=1}^{M} \langle B_{nm}B_{n'm'}\rangle \phi_{nm}(X)\phi_{n'm'}(X') \\
&= \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{n'=1}^{N}\sum_{m'=1}^{M} (1\cdot\delta_{nn'}\delta_{mm'})\phi_{nm}(X)\phi_{n'm'}(X') \\
&= \sum_{n=1}^{N}\sum_{m=1}^{M} \phi_{nm}(X)\phi_{nm}(X')
\end{aligned}$$

Substituting the expression for $\phi_{nm}(X)$:

$$\begin{aligned} K(X, X') &= \sum_{n=1}^{N}\sum_{m=1}^{M}\left[\exp\left(-\alpha\lambda_{nm}t\right)\sin\left(\frac{n\pi x}{L}\right)\sin\left(\frac{m\pi y}{L}\right)\right] \\ &\quad\times\left[\exp\left(-\alpha\lambda_{nm}t'\right)\sin\left(\frac{n\pi x'}{L}\right)\sin\left(\frac{m\pi y'}{L}\right)\right] \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M}\exp\left(-\alpha\lambda_{nm}(t+t')\right)\sin\left(\frac{n\pi x}{L}\right)\sin\left(\frac{m\pi y}{L}\right)\sin\left(\frac{n\pi x'}{L}\right)\sin\left(\frac{m\pi y'}{L}\right) \end{aligned}$$

Substituting $\lambda_{nm} = \left(\frac{\pi}{L}\right)^2(n^2+m^2)$, the final kernel is:

$$K(X, X') = \sum_{n=1}^{N}\sum_{m=1}^{M}\exp\left(-\alpha\left(\frac{\pi}{L}\right)^2(n^2+m^2)(t+t')\right)\sin\left(\frac{n\pi x}{L}\right)\sin\left(\frac{m\pi y}{L}\right)\sin\left(\frac{n\pi x'}{L}\right)\sin\left(\frac{m\pi y'}{L}\right)$$

where $X = (t, x, y)$ and $X' = (t', x', y')$.

Given a dataset of P measurements $\{(X_i, T_i, \sigma_i^2)\}_{i=1}^{P}$, where $X_i = (t_i, x_i, y_i)$, $T_i = T(t_i, x_i, y_i)$ is the measured temperature, and σ_i^2 is the measurement noise variance at point i. We assume independent noise for each measurement. The kernel matrix $\bar{K}$ for the observed data points is constructed as:

$$[\bar{K}]_{ij} = K(X_i, X_j) + \sigma_i^2\delta_{ij}$$

Note: If we assume a uniform noise variance σ^2 for all measurements, the matrix becomes $\bar{K} = K + \sigma^2\mathbf{I}$, where $[K]_{ij} = K(X_i, X_j)$ and $\mathbf{I}$ is the identity matrix. The posterior expectation (prediction) of the temperature at a new point $X_* = (t_*, x_*, y_*)$ is given by:

$$\langle T(X_*)\rangle = K_*^T\bar{K}^{-1}\mathbf{T}$$

where:

- $\mathbf{T} = [T_1, T_2, ..., T_P]^T$ is the vector of observed temperatures.
- K_* is the vector of covariances between the test point X_* and the training points X_i:

$$[K_*]_i = K(X_*, X_i)$$

Explicitly written using summation notation:

$$\langle T(X_*)\rangle = \sum_{i,j=1}^{P} K(X_*, X_i)[\bar{K}^{-1}]_{ij}T_j$$

This framework allows predicting the full spatial-temporal temperature field $T(t, x, y)$ based on sparse measurements by leveraging the known analytical structure of the heat equation solution embedded within the custom kernel.

Results and Analysis: 2D Temperature Reconstruction

Having derived the custom kernel for the 2D heat equation, we now evaluate its performance in reconstructing the spatio-temporal temperature field $T(x, y, t)$ from sparse measurements. For this demonstration, 60 measurements were generated by sampling the analytical solution at 20 unique spatial coordinates (x_i, y_i), each measured at 3 distinct time points. These 60 data points (t_i, x_i, y_i, T_i) were then used to train the GPR model. The kernel incorporated $N = M = 50$ modes, with domain size $L = 1.0$ and thermal diffusivity $\alpha = 0.01$.

Figure 2 presents snapshots of the reconstruction at three different times: $t = 0.00$, $t = 1.00$, and $t = 2.00$. Each row corresponds to a time slice and displays four panels: the analytical temperature field (ground truth), the GPR mean prediction, the GPR posterior standard deviation (uncertainty), and the absolute error between the GPR mean and the analytical solution.

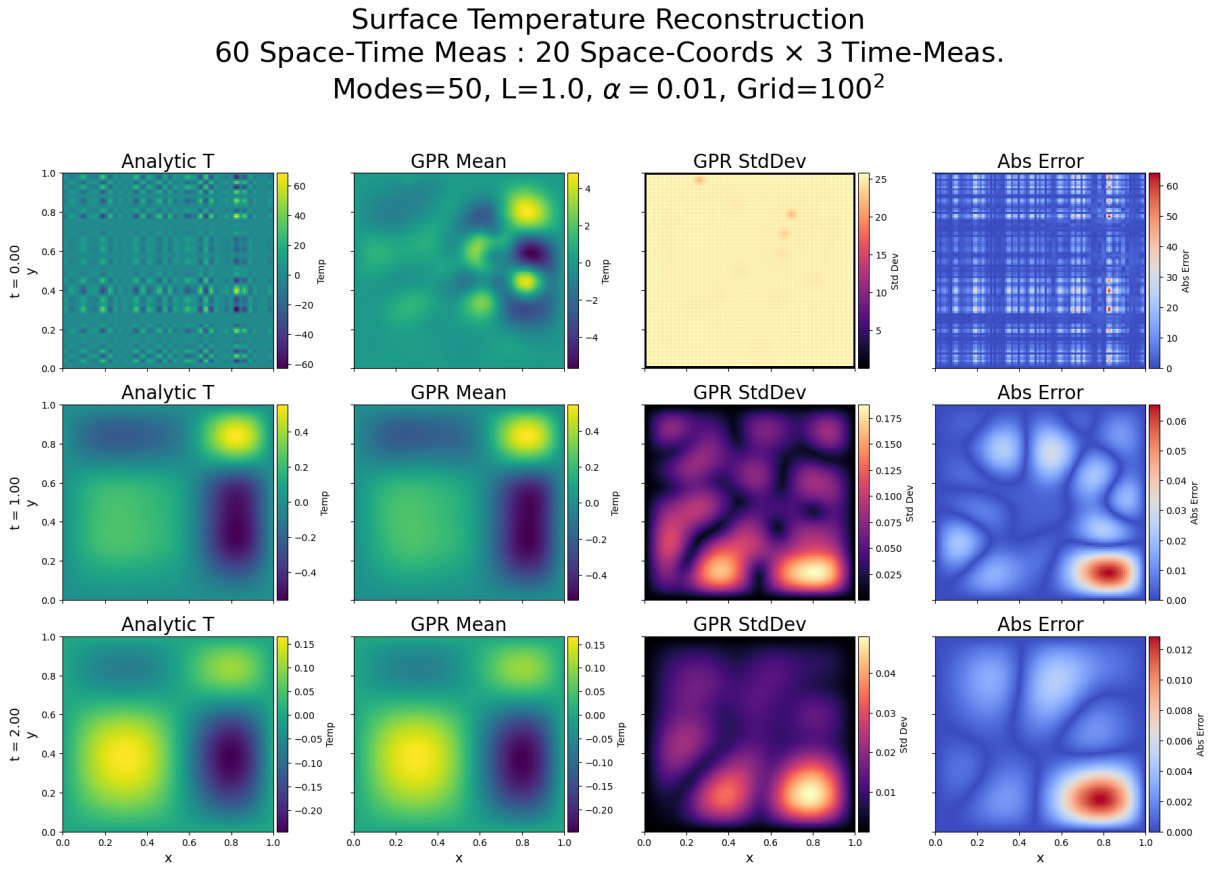

Figure 2: GPR reconstruction of 2D heat equation dynamics using 60 sparse measurements (20 spatial locations x 3 time points). Parameters:Modes= 50 (X-Y), $L = 1.0$, $\alpha = 0.01$. Each row shows a time snapshot (t=0.00, 1.00, 2.00). Columns display: Analytical Temperature, GPR Mean Prediction, GPR Standard Deviation, and Absolute Error.

Analysis of Time Snapshots:

t = 0.00 (Top Row): The analytical solution exhibits a complex, high-frequency spatial structure, characteristic of an initial condition composed of many modes. The GPR mean prediction captures the larger-scale features but presents a significantly smoothed version, unable to resolve the fine details from the sparse data. This smoothing is reflected in the Absolute Error plot, which shows very high errors across the domain, indicating a poor pointwise reconstruction of the initial state. The Standard Deviation is also high and relatively uniform, consistent with the kernel providing minimal temporal damping near $t = 0$. As explained in the 1D case, the kernel's exponential factor $\exp(-\alpha\lambda_{nm}(t+t'))$ is near 1 for $t, t' \approx 0$, allowing many high-frequency functions (consistent with the kernel's spatial structure) to potentially fit the data, leading to high uncertainty and error.

t = 1.00 (Middle Row): Significant thermal diffusion has occurred. The analytical solution is now much smoother, dominated by lower-frequency modes. The GPR mean prediction matches the analytical solution remarkably well, capturing the location, shape, and magnitude of the remaining temperature variations. The Standard Deviation is drastically reduced compared to $t = 0$, with clear hotspots indicating regions farther from the underlying 20 spatial measurement locations. Correspondingly, the Absolute Error is orders of magnitude smaller than at $t = 0$, with the largest errors correlating spatially with the regions of highest standard deviation. This demonstrates the effectiveness of the kernel's time-decay component in constraining the solution as time progresses.

t = 2.00 (Bottom Row): The system has further thermalized. Both the analytical solution and the GPR mean prediction show further decay and smoothing. The agreement between them remains excellent. The Standard Deviation continues to decrease, although the spatial pattern related to data sparsity persists. The Absolute Error is further reduced, confirming the model's ability to accurately track the long-term dynamics dictated by the physics embedded in the kernel.

Overall Performance: The results clearly demonstrate the power of the physics-informed GPR approach for the 2D heat equation. Despite using only 60 measurements spread across space and time, the model successfully reconstructs the dominant dynamics of the system, particularly after the initial high-frequency components have decayed. The custom 2D kernel, encoding the modal structure and exponential decay rates (λ_{nm}), provides a strong inductive bias that enables accurate predictions far from the initial time ($t = 0$) where generic methods would likely fail with such sparse data. The standard deviation maps provide valuable spatial information about prediction confidence, highlighting areas potentially requiring more data. The primary limitation, consistent with the 1D case, is the difficulty in accurately resolving complex, high-frequency initial conditions ($t = 0$) with limited data, a regime where the kernel's temporal constraints are weakest.

References

[1] Tao Jiang, John Rogers, Marius S. Frank, Ove Christiansen, Yong-Xin Yao, and Nicola Lanatà. Error mitigation in variational quantum eigensolvers using tailored probabilistic machine learning. *Phys. Rev. Res.*, 6:033069, Jul 2024.