Learning Green's Functions Using Neural Operators

Jérémy PAWLUS

Université de Strasbourg

August 29, 2024

Table of Contents

Preliminary remarks

- My internship took place at the University of Strasbourg, in the Institut de Recherche Mathématique Avancée (IRMA) from February 19 to July 31, 2024 under the supervision of Dr. Emmanuel FRANCK, Dr. Victor MICHEL-DANSAC and Dr. Antoine DELEFORGE.
- It is a Scientific Machine Learning (SciML) project about learning Green's functions using neural networks.

About the research center and the team

Figure 1: Université de Strasbourg

Figure 2: *Institut de Recherche Mathématique Avancée* (IRMA)

Context and Objectives

Context:

- **General:** SciML integrates math, physics, and ML for numerical analysis neural operators offer a potential replacement for traditional numerical methods.
- **Specific Context:** Use Green Neural Operators to study Green functions for the Helmholtz equation.

Objectives:

- **Academical:** Apply skills in a professional setting; complete report, presentation, and portfolio.
- **Scientific:** Understand SciML and Green's functions; develop neural network for Green function kernel; implementing data generation and training.
- **Technical:** Use Scimba library to solve second-order one-dimensional partial differential equation (PDEs) (**therefore ODEs**), compute errors, and plot results.

Nonhomogeneous Second-Order Differential Equation:

$$\mathcal{L}[u](x) = f(x) \quad \text{in } \Omega =]a, b[, \tag{1}$$

where $\mathcal{L}$ is a differential operator, Ω is the domain, and u satisfies the boundary conditions $u(a) = \alpha$ and $u(b) = \beta$.

Green's Function Definition [1, 2]:

$$\mathcal{L}[G(x, \xi)] = \delta(x - \xi) \quad \text{in } \Omega = [a, b], \tag{2}$$

The Green's function $G(x, \xi)$ is the solution to the differential equation with a Dirac delta function $\delta(x - \xi)$ as the source term, subject to zero boundary conditions $G(a, \xi) = 0$ and $G(b, \xi) = 0$ on $\partial\Omega$.

Remarks on Green's Functions

Key Idea: This formulation allows the solution $u(x)$ to be expressed as a convolution of $G(x, \xi)$ with the source term $f(\xi)$.
Boundary Conditions: The Dirichlet boundary conditions were the only ones used in the project.

Green's Function under the Sturm-Liouville Operator

Formulation of the problem [3]:

$$\mathcal{L}_{\mathrm{SL}}[u](x) = \frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + q(x)u(x) = f(x), \quad a < x < b,$$

with boundary conditions $u(a) = 0$ and $u(b) = 0$. $\mathcal{L}_{\mathrm{SL}}$ is the Sturm-Liouville operator.
Solution:

$$G(x,\xi) = \begin{cases} \frac{u_1(\xi)u_2(x)}{p(\xi)W(\xi)}, & a \le \xi \le x, \\ \frac{u_1(x)u_2(\xi)}{p(\xi)W(\xi)}, & x \le \xi \le b, \end{cases} \tag{3}$$

where $u_1(x)$ and $u_2(x)$ are solutions of the homogeneous problem with $u_1(a) = 0$, $u_2(b) = 0$, $u_1(b) \neq 0$, $u_2(a) \neq 0$, and $W(\xi)$ is the Wronskian:

$$W(u_1, u_2)(\xi) = \begin{vmatrix} u_1(\xi) & u_2(\xi) \\ u_1'(\xi) & u_2'(\xi) \end{vmatrix}. \tag{4}$$

Relevancy: such a framework is useful for determining the exact Green's function on a given problem compatible with the Sturm-Liouville operator.

Complete formulation of a PDE solution using Green's Function

Preliminary Problem: Suppose the PDE follows the form in equation 1
Solution:

$$u(x) = u_0(x) + \int_a^b G(x,\xi) f(\xi)\, d\xi, \tag{5}$$

where $u_0(x)$ is the homogeneous solution and $G(x,\xi)$ is the Green's function.
Corresponding Green's Function:

$$\mathcal{L}[G(x,\xi)] = \delta(x-\xi), \quad a < x, \xi < b,$$
$$G(a,\xi) = G(b,\xi) = 0$$

Paving the way to neural networks: the Green's function potential in solving PDEs might be leveraged by neural networks to approximate a solution of a given PDE.

About Neural Operators

- Neural-networks that learn mappings between infinite-dimensional function spaces. [4]
- Universally approximate continuous operators, overcoming traditional method challenges.
- Robust to variation of source terms when applied to PDE numerical solving.

General Elliptic PDE forumulation for the Neural Operator

$$\begin{cases} L_\mu u(x) = \nabla \cdot (A(x,\mu)\nabla u(x)) + B(x,\mu)u(x) = f(x), & x \in \Omega, \mu \in \mathcal{M} \\ a\partial_x u + bu = g(x), & x \in \partial\Omega \end{cases}$$

where μ are parameters, L_μ is a linear differential operator, $A(x,\mu)$ and $B(x,\mu)$ are coefficient functions, and $f(x)$ is the source term.

Deep Green Neural Operators

Approach:

- Learn a neural network N_κ with a kernel κ that approximates the Green's function. [5]
- Train N_κ to minimize the mean squared error between the predicted and true solutions.

Neural Network Equation:

$$L_\mu\left(\kappa(x,y)\right) = \delta(x-y) \tag{6}$$

with appropriate boundary conditions.
PDE Solution (inspired from equation (5)):

$$u_{\text{NO}}(x) = \int_\Omega \kappa(x,y) f(y)\, dy + u_{\text{hom, NO}}(x) \tag{7}$$

where $u_{\text{hom, NO}}(x)$ is the homogeneous solution.

Neural Network Architecture

The figure below, is sourced from Nicolas Boullé's thesis [6]:

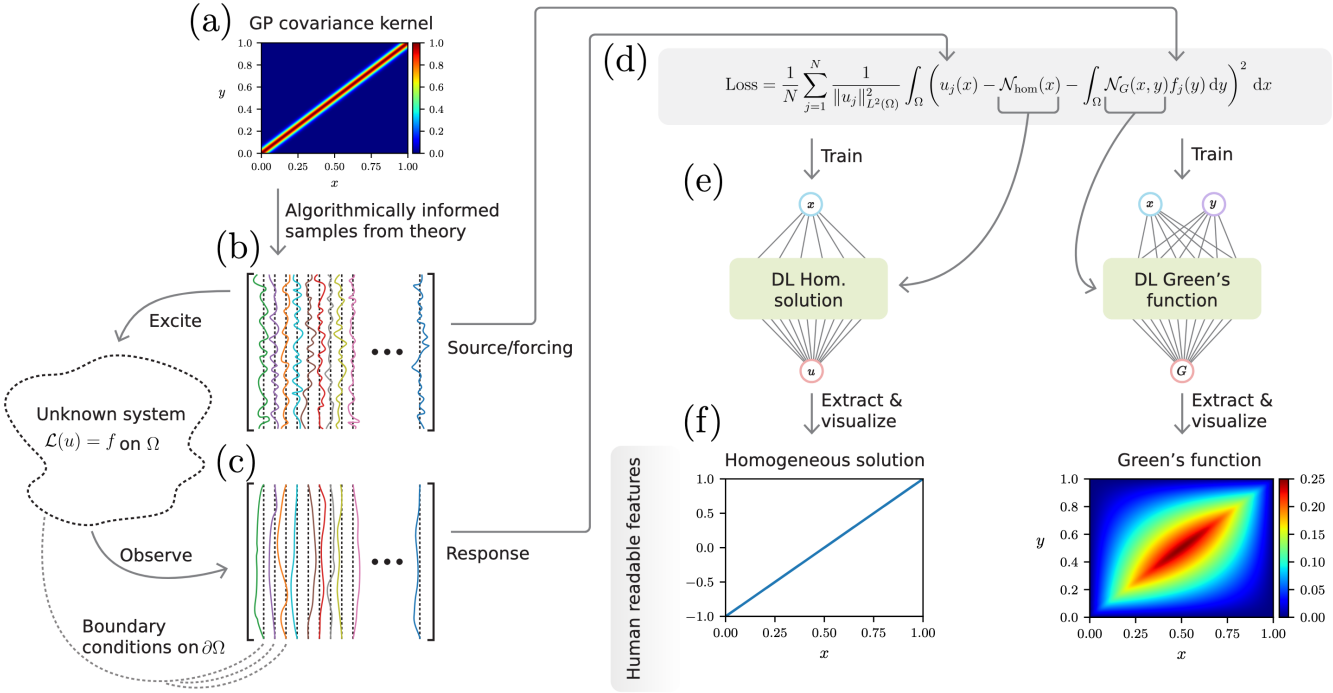


Figure 3: Architecture of the neural network used in the implementation

Network Roles and Architecture

Role of N_G (Green's Function Network):

- Approximates the Green's function $G(x, y)$ with $\kappa(x, y)$ (in equation (5)).
- Learns system response to forcing terms.
- Implemented with `green_kernel_x` and `kernel_sublayer_x`.

Role of N_{hom} (Homogeneous Solution Network):

- Learns the homogeneous solution $u_{\text{hom,NO}}(x)$ (in equation (5)).
- Ensures boundary conditions are met.
- Implemented with `local_sublayer_x`.

Conclusion:

- N_G handles external forces, N_{hom} ensures boundary conditions.
- Joint optimization captures PDE characteristics.
- Implemented with `deep_operator_x` for simultaneous training.

Data Generation and Training

Preliminary:

- Domain Ω (e.g., $[0, 1]$) and parameter domain $\mathcal{M}$.
- N_{simu} simulations with $N_{\text{collocation}}$ collocation points $\{x_i\}$, N_{sensor} sensor points $\{y_i\}$ and N_{simu} distinct instances of parameters μ_j. Here, $N_{\text{collocation}} = N_{\text{sensor}}$.

Steps:

- Sample N_{simu} forcing terms $\{f_j\}$ from a Gaussian process (relevancy showcased in [7]).
- Solve PDE $Lu_j = f_j$ numerically to obtain responses $\{u_j\}$ using either finite differences methods or SciPy's Boundary Value Problem (BVP) solver.

Gaussian Process:

- Uses RBF kernel $K_{RBF}(x, y)$ with length-scale ℓ.
- Normalized length-scale $\lambda = \ell/(b - a)$.

Conclusion:

- Ensures training data captures essential PDE characteristics.

Loss Functions

Green Kernel Loss:

$$\mathcal{J}_{\mathrm{G}}(x) = \frac{1}{N_{\mathrm{simu}}} \sum_{j=1}^{N_{\mathrm{simu}}} \left(\int_{\Omega} \kappa(x, y; \mu_j) f_j(y)\, dy - u_j(x) \right)^2 dx$$

$$\approx \frac{1}{N_{\mathrm{simu}}} \sum_{j=1}^{N_{\mathrm{simu}}} \left(\frac{\mathrm{vol}(\Omega)}{N_{\mathrm{sensor}}} \sum_{i=1}^{N_{\mathrm{sensor}}} \kappa(x, y_i; \mu_j) f_j(y_i) - u_j(x) \right)^2 dx,$$

where N_{simu} is the number of simulations, N_{sensor} is the number of sensors, and $\mathrm{vol}(\Omega)$ is the volume of the domain Ω.

Homogeneous Solution Loss:

$$\mathcal{J}_{\mathrm{hom}}(x) = \frac{1}{N_{\mathrm{simu}}} \sum_{j=1}^{N_{\mathrm{simu}}} \left(u_{\mathrm{hom}}(x; \mu_j) - u_{j_{\mathrm{hom}}}(x) \right)^2,$$

where $u_{j_{\mathrm{hom}}}(x; \mu_j)$ represents the true homogeneous solution obtained from the numerical solvers corresponding to the source term f_j.

The test cases involved

Laplace (or Poisson) Equation:

- Models potential fields in mechanical and thermal equilibria.
- Equation:

$$\Delta u = f \quad \text{in} \quad [0,1]$$

- Boundary conditions:

$$u(0) = 0, \quad u(1) = 0$$

Advection-Diffusion Equation:

- Models transport and diffusion of substances.
- Equation:

$$Du'' + 2u' + u = f \quad \text{in} \quad [0,1] and D \text{in} \quad [0.5, 1.5],$$

- Boundary conditions:

$$u(0) = 1, \quad u(1) = -2$$

Helmholtz Equation:

- Arises in wave propagation and vibration analysis.
- Equation:

$$\Delta u + k^2 u = f \quad \text{in} \quad [0,1] \text{ with } k = 8$$

- Boundary conditions:

$$u(0) = 0, \quad u(1) = 0$$

Neural Network Configuration for the Helmholtz Equation (with $k = 8$)

Neural Network Configuration	**Value**
Fourier Features	70
Physical Parameter dependance of the kernel	$\mu = k = 8$
Layer Format	[50, 50, 50, 50, 50]
Local Layer Activation	Yes
Local Layer Format	[50, 50, 50, 50, 50]
Optimizer	Adam (100 iterations) + LBFGS (2400 iterations)
Batch Size (Collocation and sensor points)	200
Gaussian Sampling	RBF Kernel with $\ell = 0.03$
Standard Deviation of Fourier Features	2.0
Number of Data/Collocation Points (Batch Size)	200
Number of Simulations	100
Activation Function	Rational

Table 1: Neural Network Configuration and Test Cases for the Helmholtz Equation with a Fixed Wavenumber $k = 8$

Predicted Green's Function for the Helmholtz Equation with $k = 8$

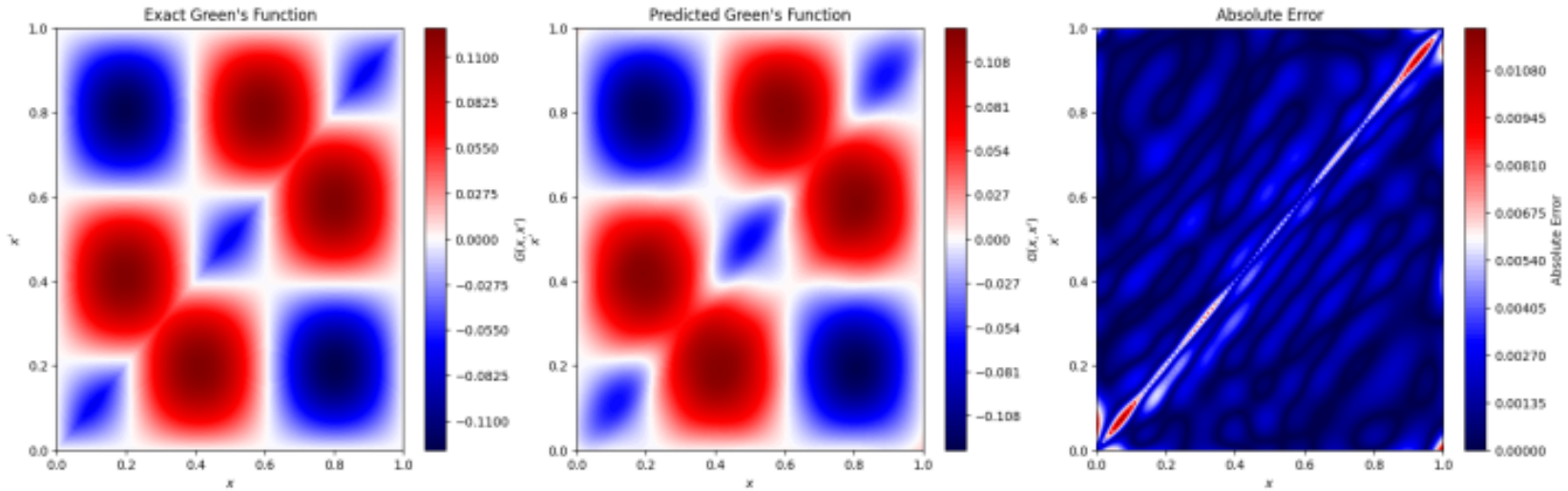


Figure 4: Predicted Green's Function for the Helmholtz Equation with $k = 8$

Error Analysis for the Helmholtz Equation (with $k = 8$, part 1)

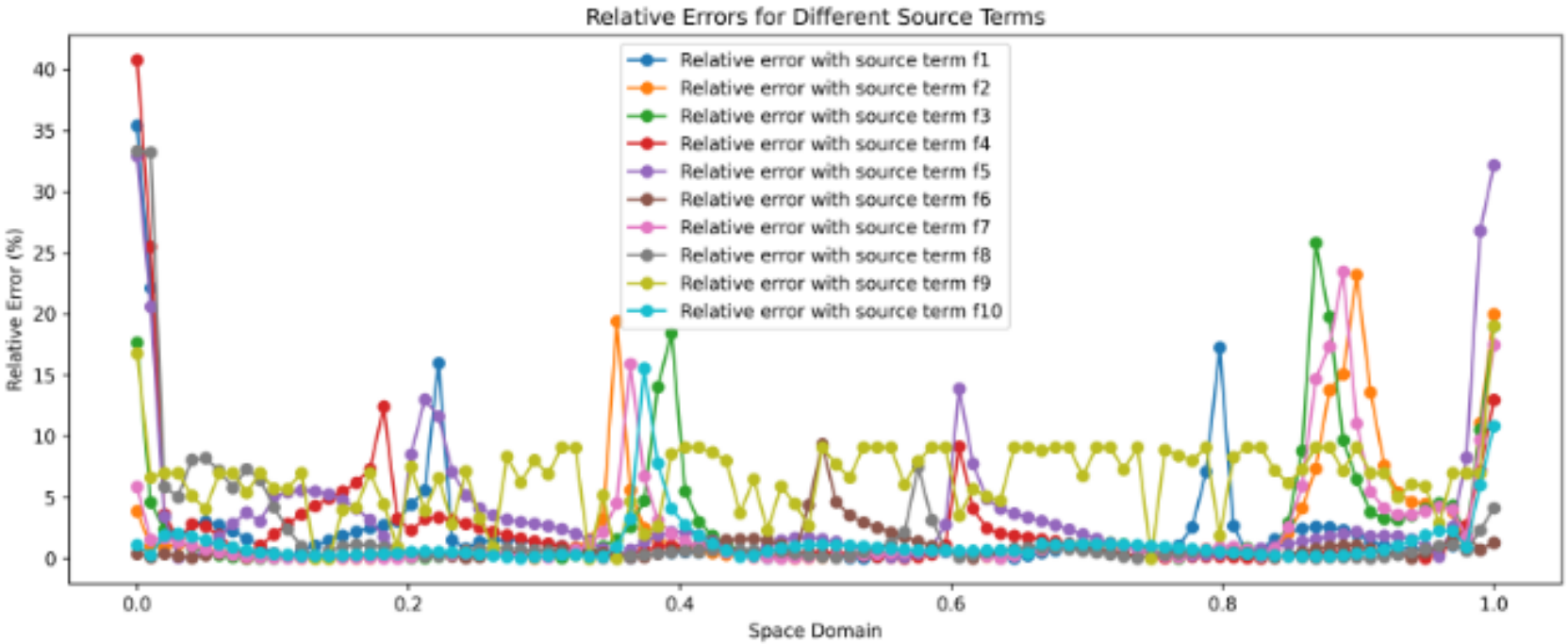


Figure 5: Analysis of relative errors for different source terms in the Helmholtz equation as a function of the space domain.

Error Analysis for the Helmholtz Equation with $k = 8$ (part 2)

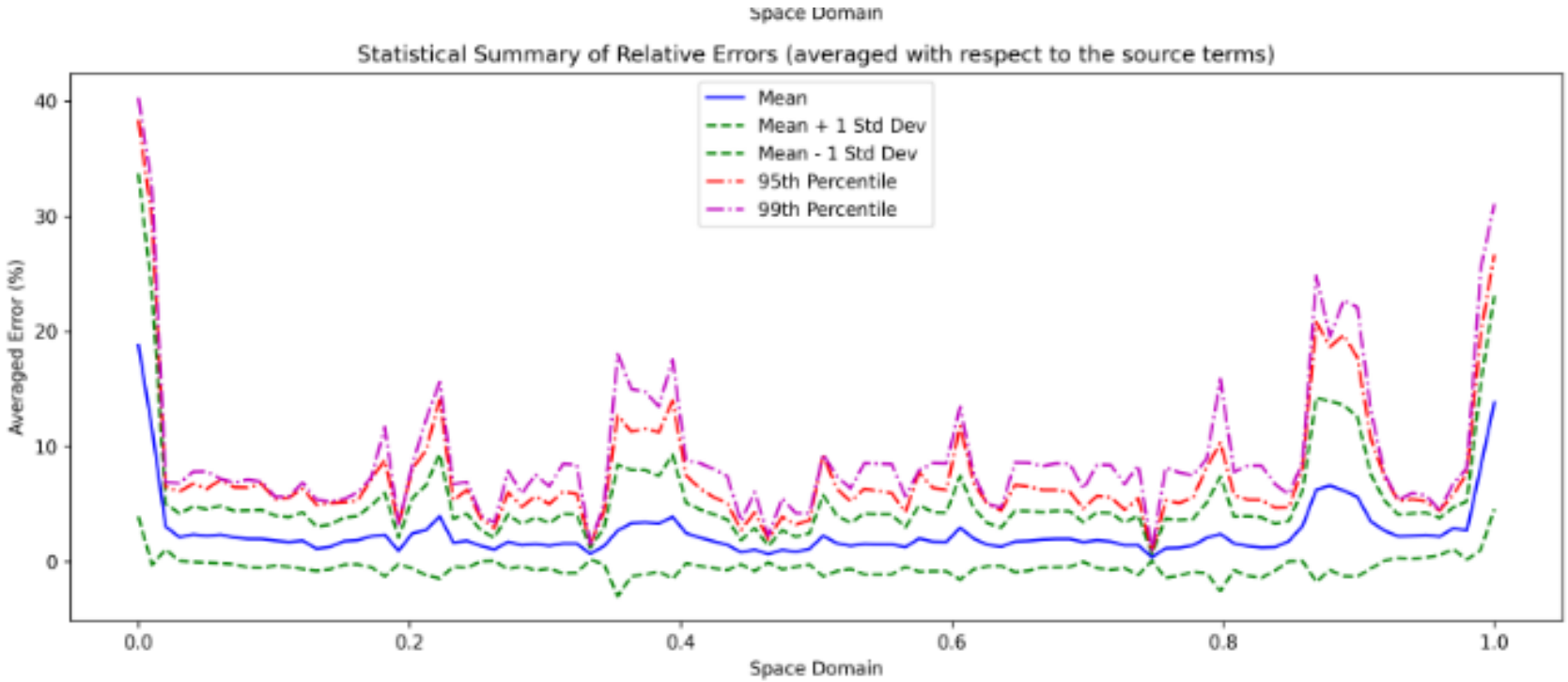


Figure 6: Analysis of relative errors for different source terms in the Helmholtz equation as a function of the space domain.

Loss Function Convergence for the Helmholtz Equation with $k = 8$

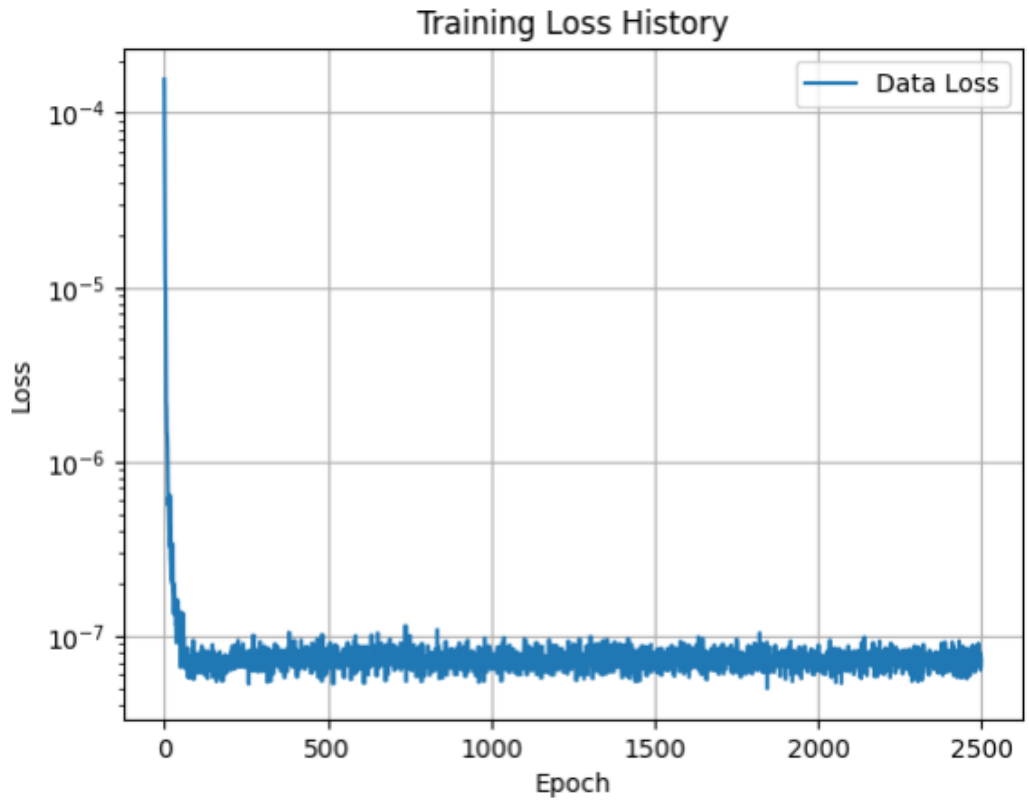


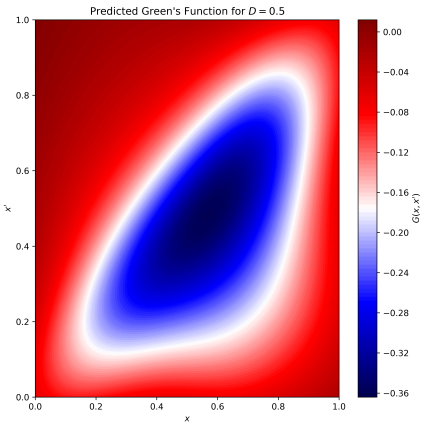
Figure 7: Convergence of the Loss Function for the Helmholtz Equation with $k = 8$

Neural Network Configuration for the the Parametric Advection-Diffusion Equation (with D in $[0.5, 1.5]$)

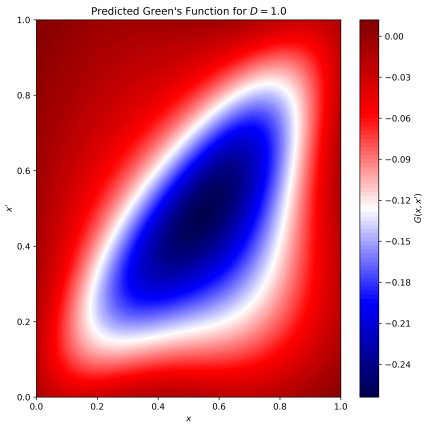
Neural Network Configuration	**Value**
Fourier Features	70
Physical Parameter dependance of the kernel	$\mu = D$ **(varies between 0.5 and 1.5)**
Layer Format	[50, 50, 50, 50, 50]
Local Layer Activation	Yes
Local Layer Format	[50, 50, 50, 50, 50]
Optimizer	Adam (100 iterations) + LBFGS (1000 iterations)
Batch Size (Collocation and sensor points)	200
Gaussian Sampling	RBF Kernel with $\ell = 0.03$
Standard Deviation of Fourier Features	1.0
Number of Data/Collocation Points (Batch Size)	200
Number of Simulations	100
Activation Function	Rational

Table 2: Neural Network Configuration and Test Cases (with D varying from 0.5 to 1.5 in the training)

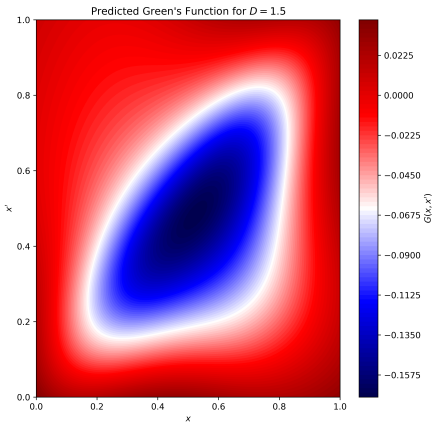
Predicted Green's Function for the Parametric Advection-Diffusion Equation (with D in $[0.5, 1.5]$)



(a) Predicted Green's Function for $D = 0.5$



(b) Predicted Green's Function for $D = 1.0$



(c) Predicted Green's Function for $D = 1.5$

Remark: The exact Green's function cannot be computed for the Parametric Parametric Advection-Diffusion equation, yet the neural network highlights how it looks like.

Space Error Analysis for the homogeneous solution of the Parametric Advection-Diffusion Equation (with D in $[0.5, 1.5]$, part 1)

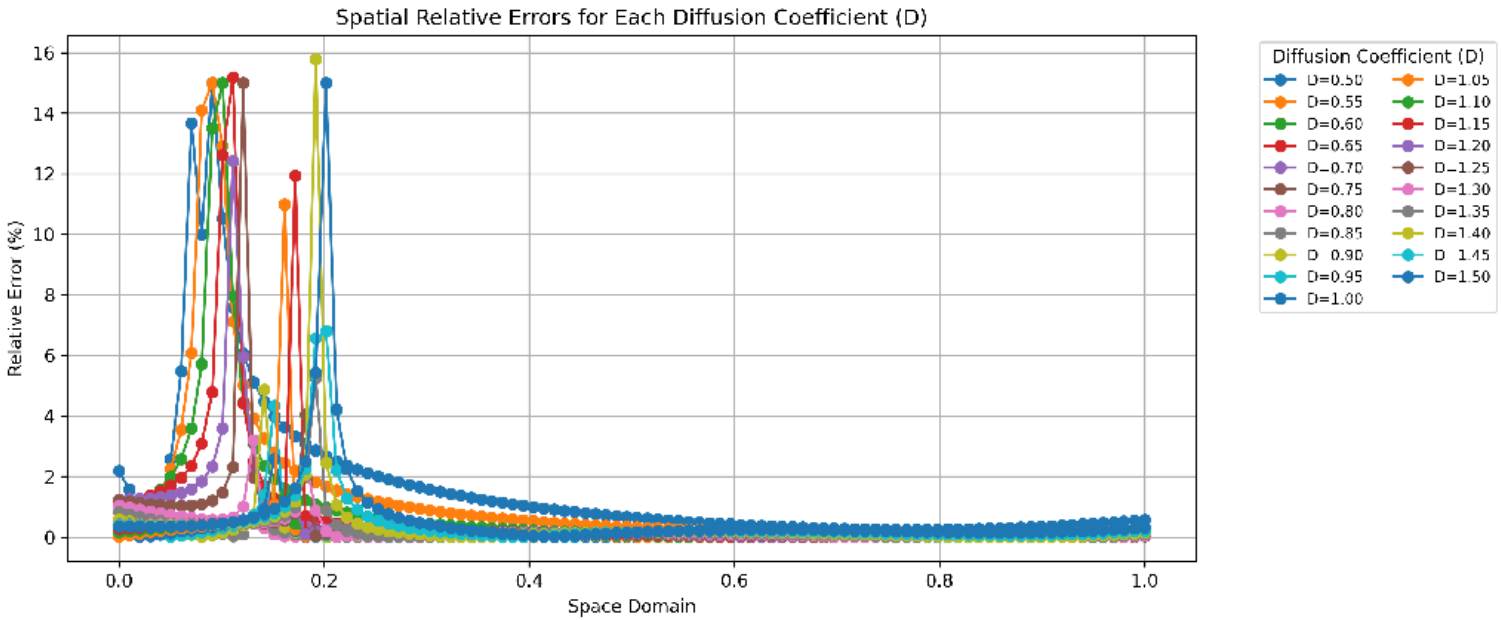


Figure 9: Analysis of relative errors for different source terms in the Parametric Advection-Diffusion equation as a function of the space domain

Space Error Analysis for the homogeneous solution of the Parametric Advection-Diffusion Equation (with D in $[0.5, 1.5]$, part 2)

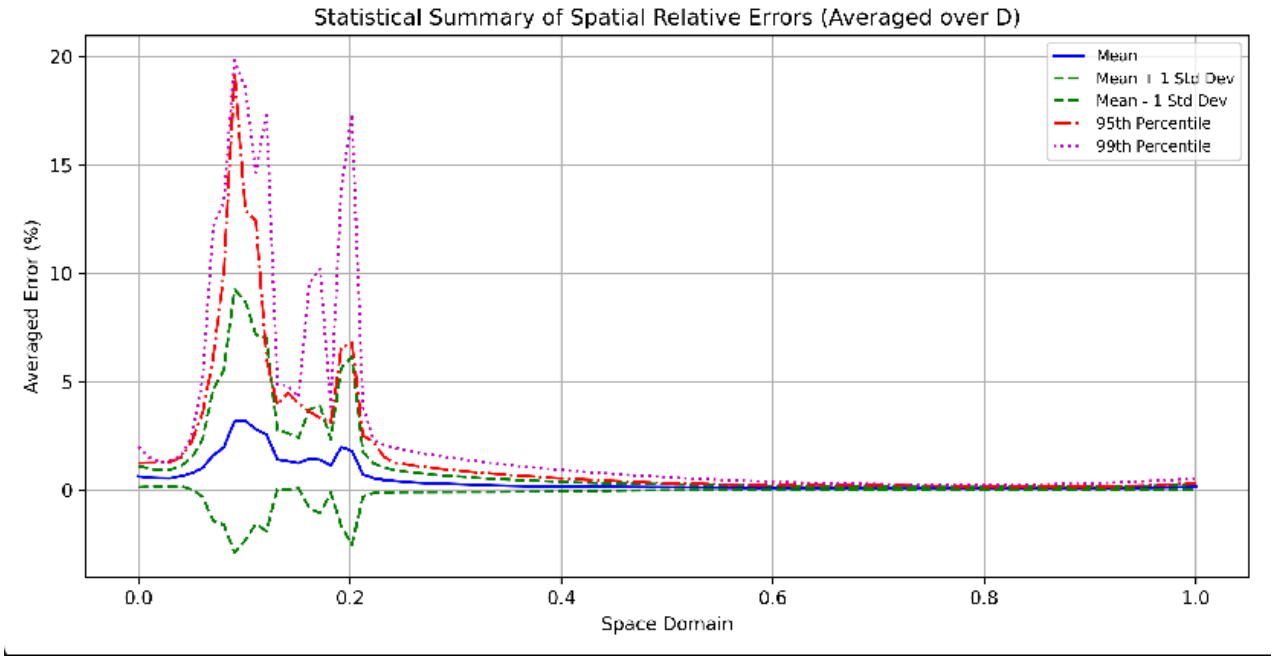


Figure 10: Analysis of relative errors for different source terms in the Parametric Advection-Diffusion equation as a function of the space domain

Error Analysis for the homogeneous solution of the Parametric Advection-Diffusion with respect to the diffusion coefficient D (part 1)

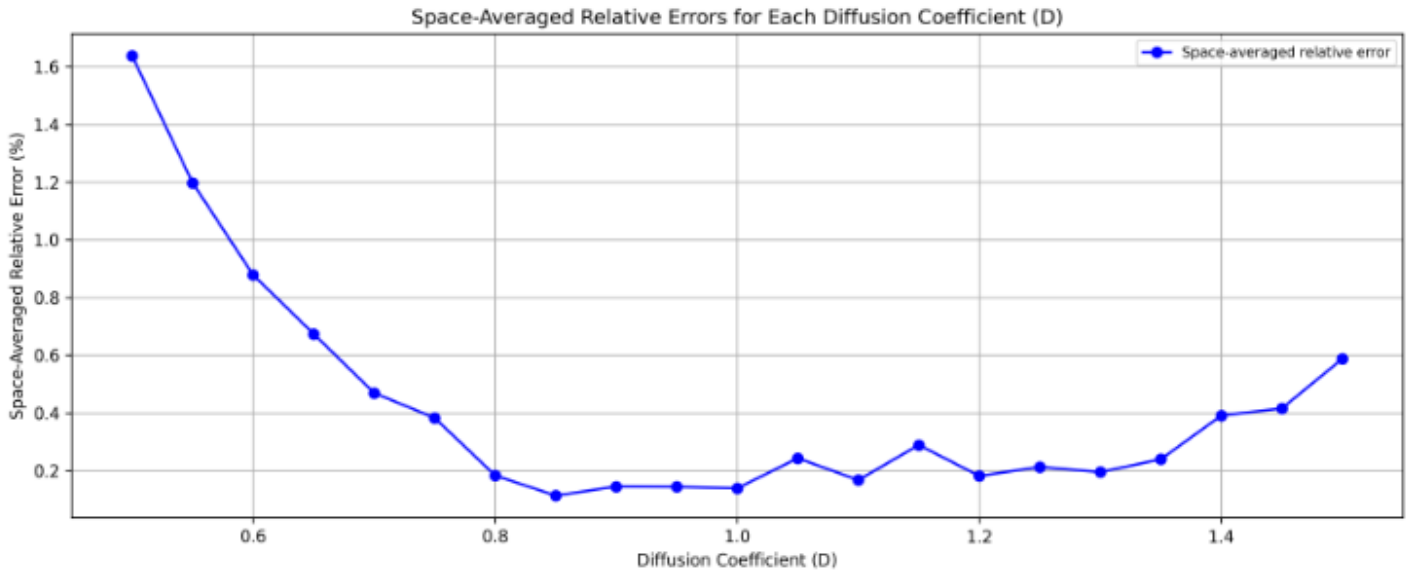


Figure 11: Analysis of relative errors for different source terms in the Parametric Advection-Diffusion equation as a function of the Diffusion coefficient D

Error Analysis for the homogeneous solution of the Parametric Advection-Diffusion with respect to the diffusion coefficient D (part 2)

Figure 12: Analysis of relative errors for different source terms in the Parametric Advection-Diffusion equation as a function of the Diffusion coefficient D

Space Relative Error of the overall solution of the Parametric Advection-Diffusion Equation (part 1)

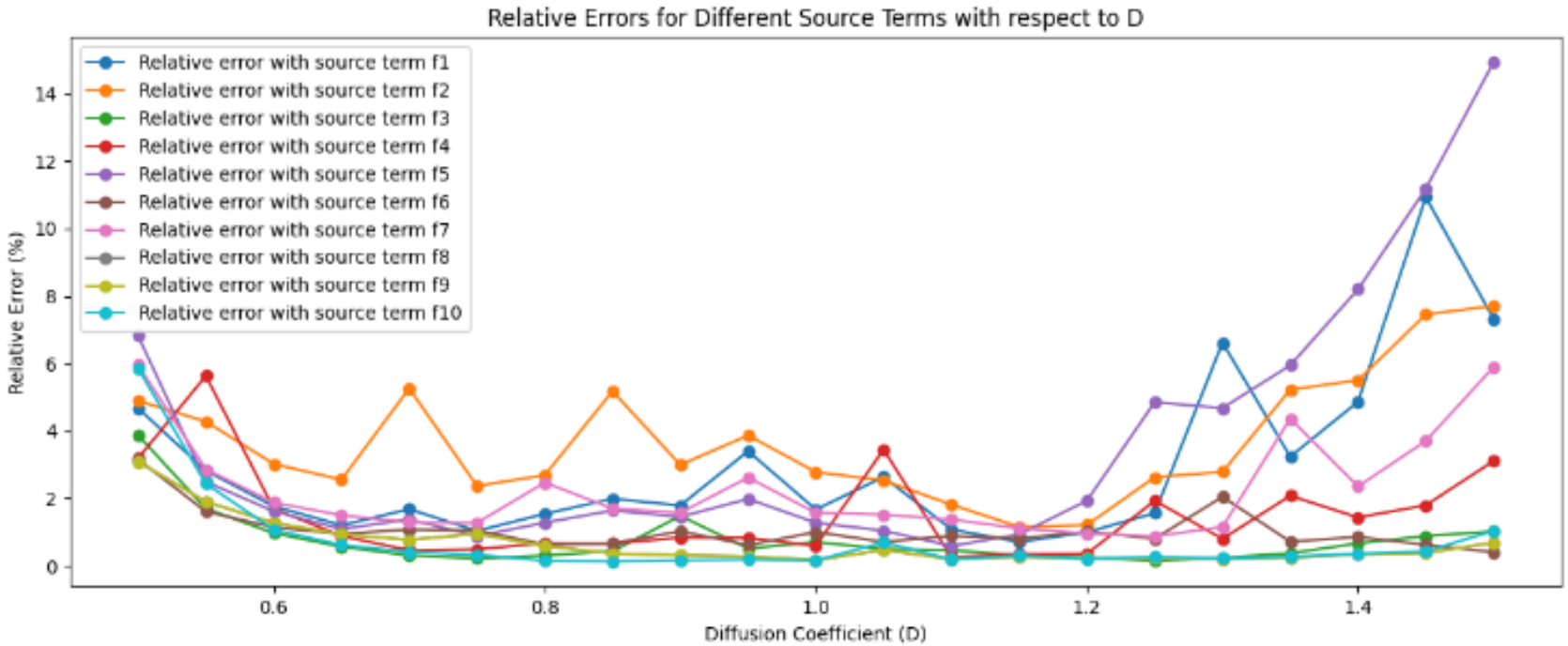


Figure 13: Analysis of the relative errors for the overall solution of the Parametric Advection-Diffusion Equation

Space Relative Error of the overall solution of the Parametric Advection-Diffusion Equation (part 2)

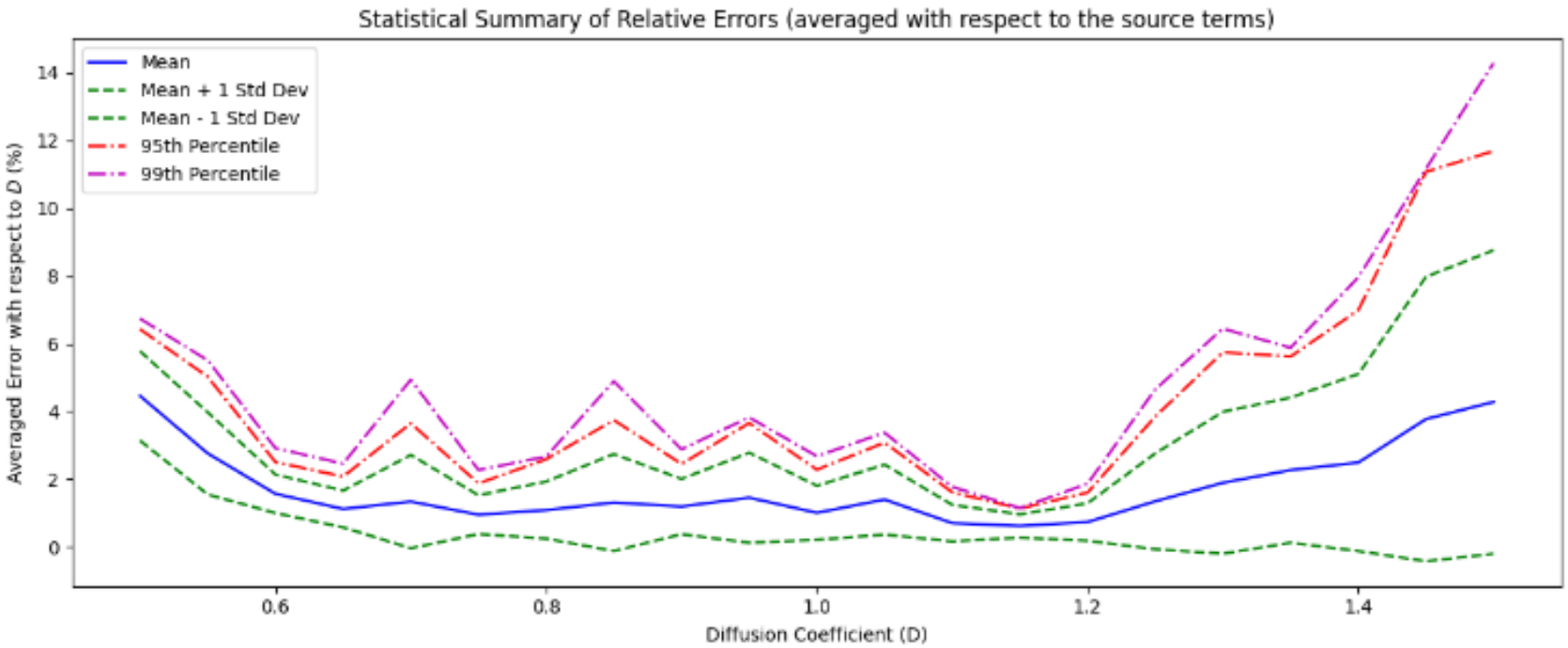


Figure 14: Analysis of the relative errors for the overall solution of the Parametric Advection-Diffusion Equation

Loss Function Convergence for the Parametric Advection-Diffusion Equation

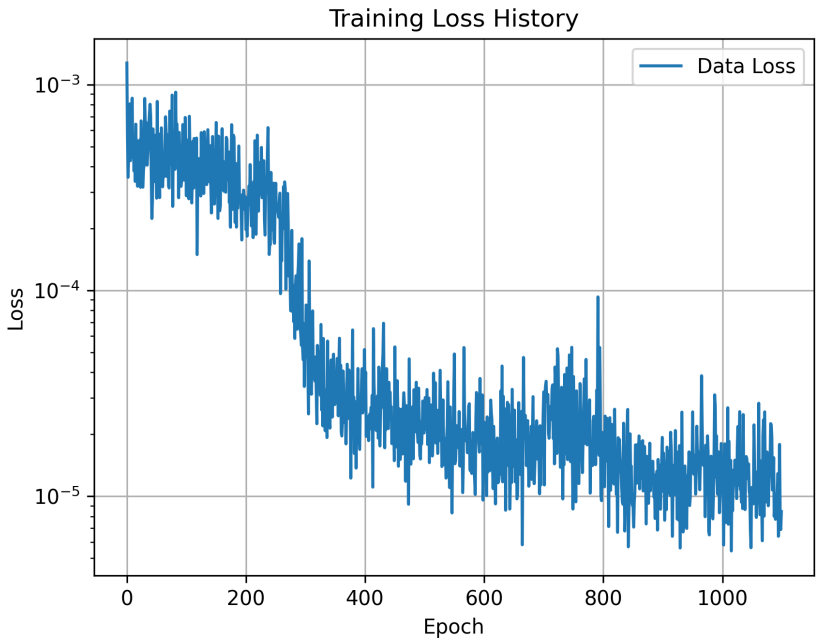


Figure 15: Convergence of the Loss Function for the Parametric Advection Diffusion Equation

Performance Comparison: Neural Network vs Finite Differences

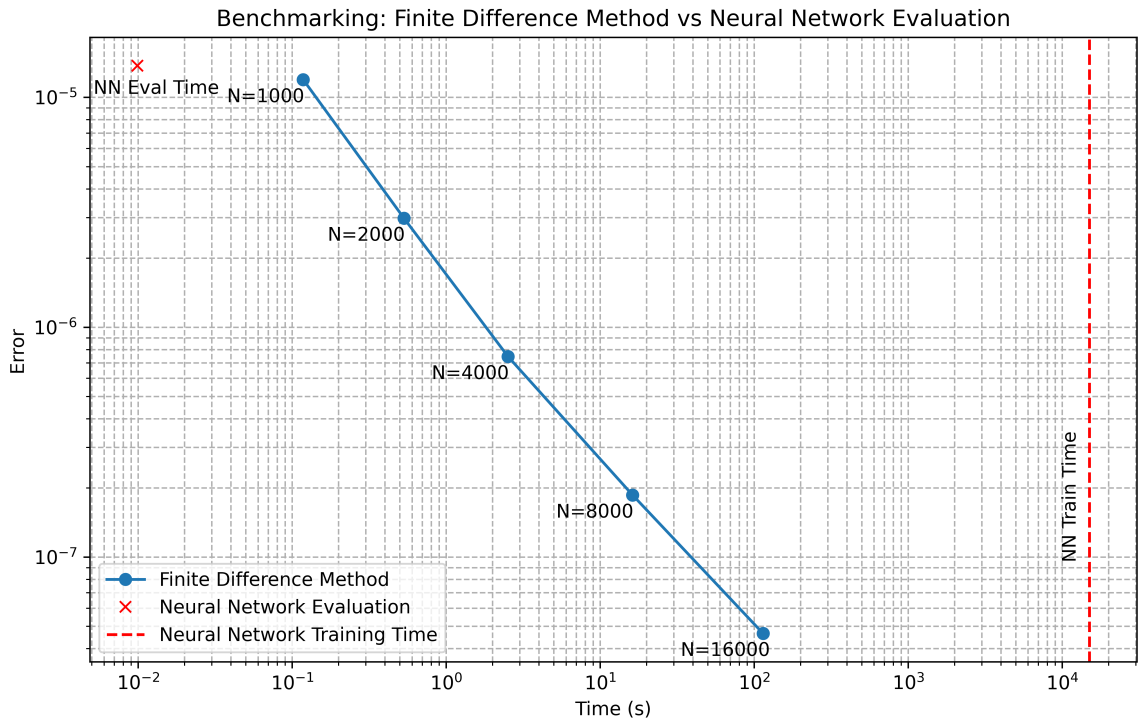


Figure 16: Performance Comparison: Neural Network vs Finite Differences

Unsatisfactory Results for the Parametric Helmholtz Equation

Eigenmodes and Green's Function:

- Eigenmodes occur at wavenumbers that are multiples of π.
- Green's function has a singularity at the source point.

$$G(x,\xi;k) = \begin{cases} -\frac{\sin(k\xi)\sin(k(1-x))}{k\sin(k)}, & 0 \leq \xi \leq x, \\ -\frac{\sin(kx)\sin(k(1-\xi))}{k\sin(k)}, & x \leq \xi \leq 1, \end{cases}$$

Neural Network Configurations:

- First network: parameter k training domain $[2, 3]$.
- Second network: parameter k training domain $[2, 3.1]$.
- Third network: parameter k training domain $[2, 3.14]$.

Predicted Green's Function for the Parametric Helmholtz Equation with $k_{\max} = 3$

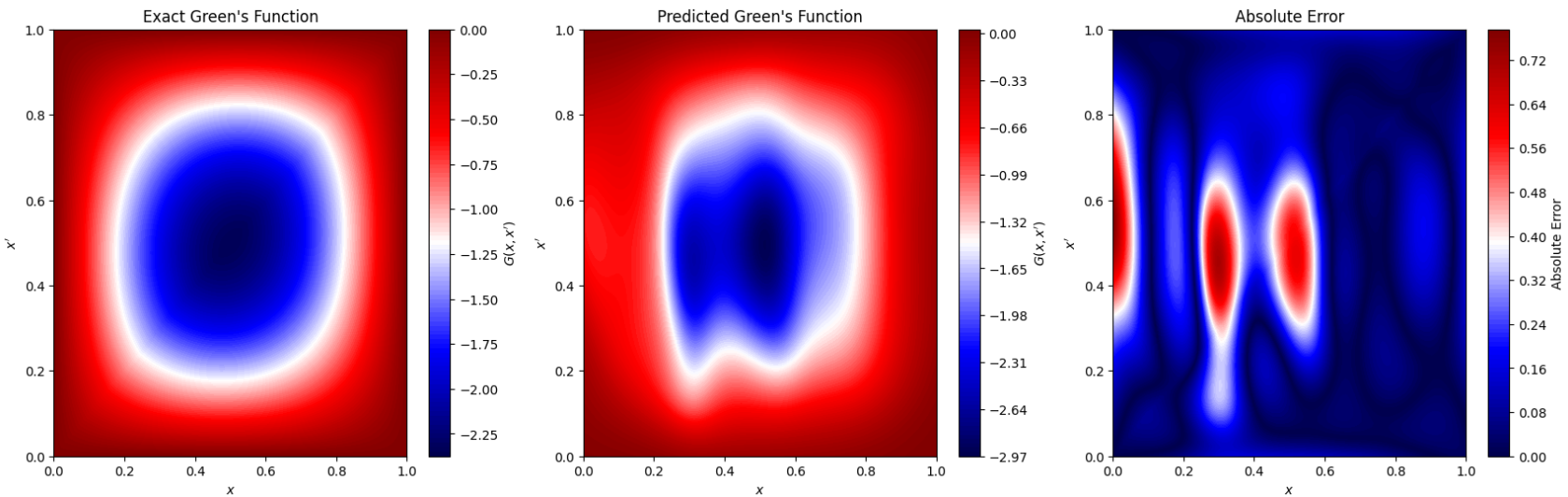


Figure 17: Predicted Green's Function for the Helmholtz Equation with $k_{\max} = 3$

Predicted Green's Function for the Parametric Helmholtz Equation with $k_{\max} = 3.1$

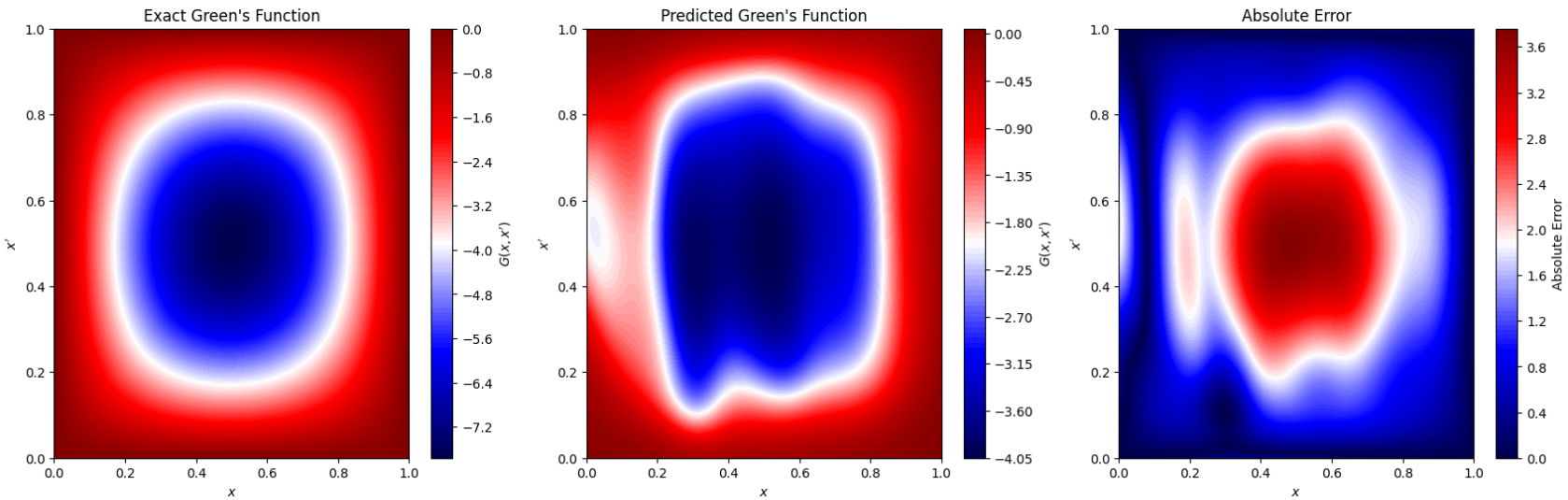


Figure 18: Predicted Green's Function for the Parametric Helmholtz Equation with $k_{\max} = 3.1$

Predicted Green's Function for the Parametric Helmholtz Equation with $k_{\max} = 3.14$

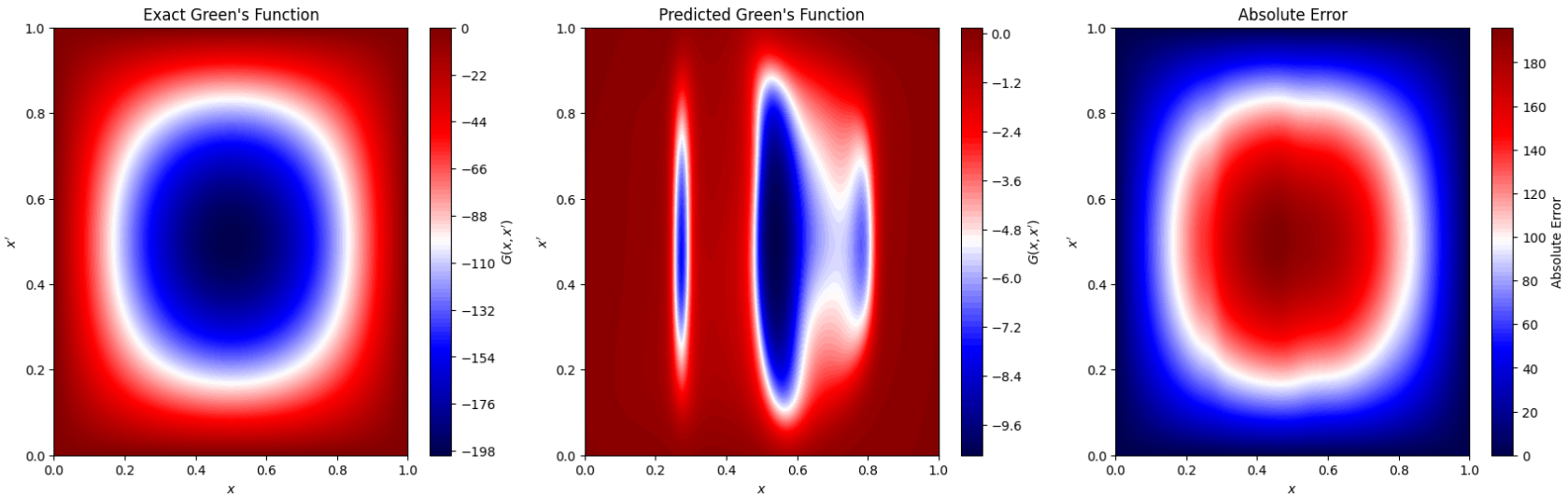


Figure 19: Predicted Green's Function for the Parametric Helmholtz Equation with $k_{\max} = 3.14$

Conclusion and Future Work

Summary:

- Implementation and simulation of Green's functions for the Laplace, Advection-Diffusion and Helmholtz equations.
- Analysis of relative errors for different source terms and parameter variations.
- Comparison of neural network performance with traditional methods like finite differences.

Outlook:

- This research can be extended, given the difficulties encountered with the Helmholtz equation in the vicinity of the eigenmodes, to potentially relying on deep learning methods to detect singularities in space domains.
- On my own, I am pursuing in a CIFRE thesis with the company *Axessim* to work on various mutidiciplinary projects involving deep learning, applied mathematics and physics.

Acknowledgements:

- I would like to thank my supervisor, Dr. Emmanuel FRANCK, Dr. Victor MICHEL-DANSAC and Dr. Antoine DELEFORGE for their guidance and support throughout this project.

Retrospective

Success: Implementing and simulating Green Neural Equations for many test cases.
Difficulties:

- Long time (Three Months) elapsed before obtaining first fruitful results with Green Neural Operators.
- Handling singularities in Green's functions and eigenmodes in the Helmholtz equation.

Acquired Knowledge and Skills:

- Overall skills in functional analysis, deep-learning and SciML.
- Gained general skills in code design, test elaboration and processing and overall methodicity.

Axes of Improvements:

- Work more linearly (step by step) on sequenced tasks.
- Accepting the fact that results might become relevant later than expected and being more patient about it.

References


D. G. Duffy. (2001). *Green's Functions with Applications*. CRC Press. https://doi.org/10.1201/9781420034790. ISBN: 9780429142697.


S. Hassani. (2013). *Mathematical Physics: A Modern Introduction to Its Foundations* (2nd ed.). Springer, Cham. https://doi.org/10.1007/978-3-319-01195-0. ISBN: 978-3-319-01194-3.


R. L. Herman. (2024). *Introduction to Partial Differential Equations*. Self-published. Reformatted using a modification of the Tufte-book document class in LaTeX. Available at: https://people.uncw.edu/hermanr/pde1/PDE1notes/PDE1_Main.pdf.


Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, A. Stuart, A. Anandkumar. (2024). *Neural Operator: Learning Maps Between Function Spaces with Applications to PDEs*. Available at: https://arxiv.org/pdf/2108.08481.


N. Boullé, S. Kim, T. Shi, A. Townsend. (2023). *Learning Green's Functions Associated with Time-Dependent Partial Differential Equations*. Available at: https://arxiv.org/pdf/2301.12345.


N. Boullé. (2022). *Data-Driven Discovery of Green's Functions*. (Doctoral thesis, University of Oxford). Available at: https://nboulle.github.io/files/thesis.pdf.


G. Pang, L. Yang, and G. E. Karniadakis, *Neural-net-induced Gaussian process regression for function approximation and PDE solution*, J. Comput. Phys., 384 (2019), pp. 270–288.