POLITECNICO DI TORINO

Master's Degree in Mathematical Engineering

Master's Degree Thesis

Nonlinear reduced-order modeling with a Graph Convolutional Autoencoder for time-domain electromagnetics

Supervisors

Prof. Stefano BERRONE

Dr. Maria STRAZZULLO

Prof. Stéphane LANTERI

Dr. Federico PICHI

Candidate

Carlotta FILIPPIN

Academic year 2023-2024

Summary

In the present work, we provide insights into the realm of computational electromagnetics, with a particular focus on time-domain electromagnetism. Numerical modeling plays a crucial role in revealing the behavior of light and matter interactions at the nanoscale, exploiting computational schemes, such as Finite-Differences Time-Domain and, as in our case, Discontinuous Galerkin methods. Since the choice of the basis elements is fundamental to enhance particularly interesting features, in the following we will consider nodal basis, thus leading to the Nodal discontinuous Galerkin. Furthermore, we will introduce Reduced Order Modelling (ROM) strategies, as a consequence of the pressing need for more efficient and accurate models capable of handling parameterized electromagnetic problems. Traditional ROM techniques like Proper Orthogonal Decomposition (POD) and the Greedy algorithm have already been investigated in the literature, along with their inherent limitations in effectively capturing nonlinear phenomena. Indeed, subsequently, we will introduce a particular deep learning-based ROM, the Graph Convolutional Autoencoder (GCA) method. The GCA method serves as a nonlinear extension of POD compression, harnessing the power of Graph Neural Networks (GNNs) to retain geometric structures within unstructured meshes.

Ai miei Nonni

Acknowledgements

I want to express my deepest gratitude to my supervisors, Prof. Stefano Berrone and Prof. Maria Strazzullo, for their invaluable support throughout my thesis work. Their guidance, especially regarding bureaucratic aspects, thesis writing, and innovative ideas, was essential for the successful completion of this project. I am thankful for their time, dedication, and insightful feedback.

I am grateful to Prof. Federico Pichi for his invaluable assistance with the coding aspects of this project and for his insightful guidance in deepening my understanding of the key characteristics of the Thesis's subject. His expertise was crucial to the success of this work.

I want to express my deep gratitude to Prof. Stéphane Lanteri and the ATLANTIS team at INRIA for giving me the amazing opportunity to spend six months with them. I also want to thank Prof. Lanteri for his support, inspiring ideas, and the chance to engage in stimulating discussions about exciting future projects. His mentorship and the welcoming environment of the team greatly enriched my experience.

Sono grata per le persone che ho incontrato e ritrovato in questi anni. A partire da Silvia e Giada, che sono state amiche, casa e famiglia a Torino. Stefano, Amin, Nico e Lorenzo, amici e colleghi da cui ho imparato tantissimo e con cui ho condiviso interminabili lavori di gruppo, budini e notti insonni. Thiago, trovato per caso alla seconda e ultima esercitazione di informatica, e gli altri compagni del primo anno Diego e Riccardo. Gli amici monregalesi che hanno animato molteplici avventure, in particolare Fabri e Luca. Isabella che mi ha ospitato presso airbnb Nakamura molte notti e cucinato per me anche con una gamba ingessata. Roberta, che ha iniziato e condiviso con me questo viaggio torinese. Le sgherlZ Sara, Nicoletta e Ilaria sempre presenti anche da lontano. Lorenzo, che mi ha supportato con tanti pasticcini e merende.

Infine, un sincero ringraziamento va alla mia famiglia. Il loro costante incoraggiamento è stato una fonte preziosa di motivazione, spingendomi continuamente a puntare al meglio e a non perdere di vista i miei obiettivi.

Table of Contents

List of Tables

List of Figures

Introduction

Nanophotonics has a pivotal role in the contemporary technological research field since the understanding and the application of light-matter interactions at the nanoscale emerge as a powerful tool to enhance technological innovation and societal advancement. The advancements in nanophotonics are aimed at further developing solar energy control with nanostructured solar cells, enhancing digital imaging sensitivity through nanostructured Complementary Metal-Oxide-Semiconductor (CMOS) image sensors [1], and optimizing light extraction and emission in optoelectronic devices such as microLED displays. Moreover, it contributes to medical applications, like nanoparticle-based therapies and virus detection biosensors. One of the crucial elements in the development of nanophotonics is numerical modeling. Numerical modeling is of great importance, providing researchers with a computational laboratory wherein they can simulate and dissect the intricate dynamics governing light-matter interactions, but also for tailoring or harnessing these interactions guided by specific objectives in the context of inverse design studies. In all generality, numerical modeling for nanophotonics is based on the system of time-domain or frequency-domain Maxwell's equations coupled with differential equations modeling the behavior of propagation media at optical frequencies. In this context several Discontinuous Galerkin (DG) type methods have been developed [2, 3]. Complementing numerical modeling, the field of nanophotonics has witnessed the emergence of reduced order modeling (ROM) techniques, such as the Greedy algorithm [4] and the proper orthogonal decomposition with Cubic Spline Interpolation (POD-CSI) method [5] and even nonlinear ROMs, introduced to handle the hyperbolic nature of the underlying PDE system.

Such a system often involves parameters like geometric features, boundary conditions, and physical properties. Solving these models accurately is crucial, especially when solutions are needed for numerous parameter variations. ROMs are particularly valuable in simulations where full-order models are computationally expensive and time-consuming to solve. The goal of a ROM is then to approximate the solution manifold, that is the set of all PDE solutions, when the parameters vary in the parameter space, through a suitable, approximated trial manifold, thus simplifying computationally complex systems by reducing their dimensionality and selecting the essential features while maintaining acceptable accuracy. Simultaneously, simulations can often be solved just for a fixed number of parameters or combinations to build the ROM space. The evaluated solutions are called snapshots and form the dataset used to identify the system's most significant modes or features. Once the meaningful features are extracted, the reduced space can be built. All these steps - dataset formation, features extraction, and low-dimensional space construction - are part of the offline phase. This initial stage is meant to build the reduced model assembling all the parameters independent quantities. Subsequently, the reduced model achieves a faster and cheaper evaluation of the solution associated with the desired parameters. This is referred to as the online phase. Furthermore, ROMs can be classified into intrusive and non-intrusive. In the intrusive ROMs, the dimensionality reduction is combined with the Galerkin projection of the full-order model onto the lower-dimensional space. Meanwhile, when the projection procedure turns out to be quite expensive, as in the case of non-linear and non-affine problems, non-intrusive ROMs are preferred, since they treat the original high-fidelity model as a black box and construct the reduced-order model using input-output data from simulations. These methods are developed with machine learning, surrogate modeling, and data-driven approaches, like POD combined with interpolation methods such as Cubic Spline Interpolation (CSI) or Neural Networks. These methodologies empower researchers to efficiently explore parameterized time-domain electromagnetic scattering problems, extracting essential insights into system behavior while facilitating the optimization of design parameters. However, in the context of nanophotonics simulations, it is common to have an initial full-order model with a discretization based on an unstructured mesh due to the complexity of the geometry, thus the new approach exploiting graph neural networks (GNNs) fulfills the request of handling the geometrical information enclosed in the mesh. In this thesis we will exploit and adapt the Graph convolutional Autoencoder (GCA) architecture presented in [6]. The novelty that we will introduce consists in considering a dataset built with the DG method, instead of the classical FEM method. In Chapter 1, we will introduce in detail the context of time-domain electromagnetism.

Thus, we present the mathematical modeling of Maxwell's equations in Chapter 2. The computational scheme used to build the dataset for our ROM method is detailed in Chapter 3, and, finally, in Chapter 4, we will discuss the peculiarities of a particular nonlinear ROM technique, the GCA and its application to our physical context.

Chapter 1

Time-domain electromagnetism

Maxwell's equations have been studied extensively for many decades and have led to practical applications in our everyday lives, including wireless communications, optical fibers, and medical imaging. These devices rely on specialized materials and geometries to control electromagnetic wave propagation. With the advancement of lithography[1] techniques in recent decades, it has become possible to fabricate geometrical structures at the nanometer scale. This has led to the discovery of numerous new phenomena arising from interactions between light and matter at such small dimensions. These effects typically manifest when the size of the device is comparable to or smaller than the wavelength of the incident field. In this chapter we will briefly discuss the development of the most common simulation techniques used in time-domain electromagnetism, focusing in particular on the Discontinuous Galerkin (DG) approach. Finally, we will present the state-of-the-art of reduced-order modeling for nonlinear problems and our physical context.

1.1 Computational electromagnetics

The large variety of phenomena exhibited by nano-optic systems, coupled with their dependency on numerous parameters and the complexity of most fabrication processes, prevents physicists from relying solely on experiments. However, apart from very specific cases involving geometries, and for which electromagnetic fields can be expressed as closed forms, solutions to Maxwell's equations are out of

[1] Semiconductor lithography or photolithography is a patterning process in chip manufacturing. The process involves transferring a pattern from a photomask to a substrate.

reach of hand calculations. Hence simulations emerge as a valuable appropriate complementary tool to physical experiments and can be exploited in various ways. Indeed, it can be used to scan a large number of configurations to pinpoint the most optimal set of parameters. Various techniques are available to solve nano-optics problems: among these techniques, the Finite-Difference Time-Domain method stands out as the most widely used due to its simple implementation and high computational efficiency. However, it has its limitations, as it can suffer from accuracy and convergence problems. To overcome these issues, the DG methods were developed. Alternatively, Finite Elements are more commonly employed in frequency-domain problems, providing an alternative approach to tackle nano-optics challenges. We will use the DG method because of its effectiveness in handling discontinuities that occur at material interfaces and in dealing with complex geometries, which is common in many nanophotonic devices.

1.2 The Discontinuous Galerkin method

Discontinuous Galerkin methods were first introduced by Reed and Hill in 1973 [7]. They have been widely used in computational fluid dynamics, but their application to the time-domain Maxwell's equations is a more recent development [8].

DG methods can be seen as an extension of the traditional Finite Element (FE) methods, where global continuity of the approximation is not mandatory. As in FE methods, the unknowns are approximated using a finite set of basis functions. However, in DG, the basis functions are limited to a single discretization cell. This implies that the solution obtained by a DG method could be discontinuous. Consequently, DG methods can effectively handle material and field discontinuities, and the weak formulation remains local to each element, eliminating the necessity for large mass matrix inversions during the solving process. Nonetheless, these methods often require more memory than standard FE methods. In addition, connections between cells are re-established using a numerical flux, which is similar to finite volume methods. The choice of numerical flux significantly affects the mathematical properties of the DG discretization. For example, the choice of centered fluxes will lead to a non-dissipative method, which is a fundamental property if we consider wave propagation in a closed cavity. Unlike its centered counterpart, the jump term of the upwind introduces dissipation in the DG scheme, which can be very helpful in situations where instabilities might occur, since it helps in damping nonphysical modes. A formal description of the DG method will be presented in Chapter 3.

1.3 Reduced-Order Modeling

In the context of electromagnetic wave propagation problems, a variety of physical systems require considerable computational effort, especially in the case of parameterized PDEs. One example of encountering such a situation is when studying problems in complex geometries for various input parameters. These parameters include frequency, directional incidence of waves, geometrical dimensions, and material properties. As mentioned in the previous sections, the discontinuous Galerkin time-domain (DGTD) method has several attractive features, such as easy adaptation to complex geometries and material composition, local approximation order strategy, and easy parallelization, and it does not require the inversion of the global mass matrix when combined with a full explicit time scheme.

Despite its high accuracy, the DGTD method suffers from a major drawback: its high dimensional structure due to duplicating the degrees of freedom, which is related to their local definition in each element. Thus, often the method turns out to be expensive in terms of both CPU time and memory demands for computing high-fidelity solutions. Especially, when dealing with complex problems that require solving over a large number of parameter values, cost reduction is often necessary. To meet this need, reduced-order modeling (ROM) methods have been developed. The final objective of ROMs is to construct a system with substantially smaller dimensions compared to the replaced full-order one, also called the high-fidelity system. The selected decrease in the computational costs is linked to a threshold of controlled loss of accuracy. ROMs have become a well-established class of methodologies based on solid mathematical foundations due to increasing interest and efforts over the last few decades. Among them, the Reduced Basis (RB) method [9, 10] enables fast and reliable evaluations of the solution for new parameter values. One usually exploits linear techniques, such as POD or the Greedy algorithm to build the reduced space, which allows for these efficient computations. POD is an SVD-based method, where the selection of the basis is based on the extraction of the principal components over a properly selected set of numerical solutions for certain values in the parameter space, called snapshots, while the Greedy algorithm, iteratively augments the space with the basis corresponding to the worst approximation in the parametric space with respect to an error estimator between the high fidelity solution and the ROM one. These methods enable the separation of computation into two phases: offline and online, yielding reliable and consistent accuracy with minimal computational overhead during the online phase. The decoupling of the two stages is fully admissible only when the dependence on the parameters is affine. Even though projection-based RB methods are effective in terms of accuracy, they do not offer any computational advantage compared to a direct approach for complex nonlinear problems with a non-affine dependence on the parameters. This is a result of the cost involved in computing the projection coefficients, which depends on the dimension of the full-order model. To address this, one can use the empirical interpolation method (EIM) [11] or its variations to obtain a linear expansion of the differential operator. However, for general nonlinear problems, this is far from trivial.

To address these weaknesses, one approach is to incorporate deep learning tools into the ROM architecture [12, 13, 14]. These tools consist of neural networks with numerous layers, designed to imitate the *highly connected and highly parallel* structure of the brain. An intriguing aspect of deep learning algorithms is that they do not need prior knowledge of the data structure. By introducing deep learning tools, thus utilizing non-intrusive model order reduction techniques, we can overcome some of the limitations of traditional linear approaches. Nonlinear machine learning methodologies can be particularly helpful in achieving a low-dimensional description of the solution varying with respect to time and/or parameters in terms of latent subspace, which represents the solution manifold. This allows for efficient capture of feature correlations and optimal capacity for pattern learning.

For problems characterized by coherent structures that evolve, such as transport, wave, or convection-dominated phenomena, the RB method might produce reduced-order models that are inefficient. To overcome this limitation, deep learning algorithms have been proposed. In particular, starting from the proper orthogonal decomposition-neural network (POD-NN) [15], the coupling of classical POD, or randomized POD (r-POD), with deep learning algorithms, has been investigated [16, 13]. This approach leverages two key strategies: (i) dimensionality reduction of snapshots from the full-order model via r-POD, treating it as the first layer of the convolutional autoencoder. This differs from traditional POD-Galerkin ROMs, where r-POD is used to generate the linear trial solution manifold; (ii) employing a multi-fidelity pretraining stage to iteratively initialize network parameters by combining different models. The resulting strategy embodies an effective fusion of the most advantageous aspects of deep learning algorithms and POD. Specifically, it leverages the non-intrusive nature of deep learning alongside the simplicity and robust mathematical foundations of POD.

In [5], a non-intrusive model order reduction for the solution of parameterized electromagnetic scattering problems is presented and, as in the case of the present work, the snapshot vectors are generated by a high-order discontinuous Galerkin time-domain solver. The approach introduced here is based on the extraction of time- and parameter-independent POD basis functions. By using the SVD method, the principal components of the projection coefficient matrices of full-order solutions onto the RB subspace are extracted. A cubic spline interpolation-based approach is proposed to approximate the dominating modes without resorting to Galerkin projection.

A notable enhancement in Deep Learning (DL) based ROMs is the autoencoder architecture, which offers a nonlinear extension of the POD linear compression method. This architecture allows for encoding the main information into a latent set of variables while extracting their key features. Indeed, some architecture presented previously [16, 12], have been improved by the introduction of a Convolutional Neural Network (CNN) [13, 17]. However, all the cited methods, and in particular classical autoencoders based on CNNs, are not suited for problems based on unstructured meshes, since they rely on a Cartesian representation of the data and do not embody geometrical features in the learning process. Thus, geometric deep learning emerges as a unifying theory for analyzing data by leveraging information about its geometry [18, 19].

A particularly promising approach, exhibiting greater capabilities in handling advection-dominated phenomena compared to classical POD, has been presented in [6]. This method, inspired by [12, 13], employs a non-intrusive and data-driven nonlinear reduction technique based on GNNs to encode the reduced manifold and facilitate rapid evaluation of parameterized PDEs [20, 21, 22]. A key feature of GNNs, achieved by assigning geometrical information to the edges of the graph, is the possibility of overcoming the limitation of the Cartesian representation of CNNs. Encouraged by these results, we aim to explore the extension of this method to the context of discontinuous Galerkin discretization applied to time-domain Maxwell's equations in the following sections.

Chapter 2

Mathematical modelling

In this chapter, we introduce the mathematical models of simple test cases, that will be used in the present work to generate numerical results in Chapter 4. First, we focus on a brief introduction to the classical formulation of Maxwell's equations and constitutive relations, then we show two formulations for two simple 2D cases, characterized by the absence of an internal source or an incident wave. The particularity of these tests is the possibility of having an analytical solution, allowing us to evaluate the error with respect to the numerical solution.

2.1 Maxwell's equations

The electric charge is the fundamental property of matter that causes electromagnetic interaction. A particle of charge q and speed $\mathbf{v}$ is subject to the Lorentz force:

$$\mathbf{F} = q\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right) \tag{2.1}$$

where $\mathbf{E}$ and $\mathbf{B}$ are respectively the electric field and the magnetic induction vectors, one shall see that these are related, through constitutive relations, to the electric displacement $\mathbf{D}$ and the magnetic field $\mathbf{H}$. We also introduce the density of free electric charges ρ, and the free electric current density $\mathbf{J}$. All these quantities depend on position $\mathbf{x} = (x, y, z)^T$ and time t, we omit the dependency for the sake of notation. We can now write Maxwell's equations in SI units:

$$\begin{cases} \nabla \times \mathbf{E} & = -\dfrac{\partial \mathbf{B}}{\partial t}, \\ \nabla \times \mathbf{H} & = \dfrac{\partial \mathbf{D}}{\partial t} + \mathbf{J}, \\ \nabla \cdot \mathbf{D} & = \rho, \\ \nabla \cdot \mathbf{B} & = 0. \end{cases} \tag{2.2}$$

along with the continuity equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0. \tag{2.3}$$

The two curl equations are considered the "fundamental" equations, while the two divergence equations are known as the "auxiliary" equations. It is apparent that the last two equations of the system (2.2) do not evolve in the sense that they do not contain any time derivative. Indeed, they only impose constraints on the solution of the first two equations of (2.2). Taking the divergence of the two curl equations, and combining with (2.3), one obtains:

$$\frac{\partial}{\partial t}\left(\nabla \cdot \mathbf{D} - \rho\right) = 0, \tag{2.4}$$

$$\frac{\partial}{\partial t}\left(\nabla \cdot \mathbf{B}\right) = 0. \tag{2.5}$$

Therefore, if the divergence conditions are satisfied for the initial state, they should also hold for any future state. Hence, we can assume that the divergence conditions are satisfied for all considered initial states.

2.1.1 Constitutive relations

Upon examining the system constituted by the curl equations of (2.2), we can observe that it contains 12 scalar unknowns but only 6 scalar equations. As a result, the system is not closed and is therefore unsuitable for solving. In order to close the system, we need to establish relationships between $(\mathbf{E}, \mathbf{B})$ and $(\mathbf{D}, \mathbf{H})$, through the introduction of the permittivity $\bar{\varepsilon}$ and permeability $\bar{\mu}$ tensors. Permittivity is a measurement of a medium's resistance to producing an electric field, meanwhile, permeability is the capacity by which a material allows magnetic lines to pass through it. In the most general case, the constitutive relations are:

$$\mathbf{D} = \bar{\varepsilon}\, \mathbf{E}, \tag{2.6}$$

$$\mathbf{B} = \bar{\mu}\, \mathbf{H}, \tag{2.7}$$

where $\bar{\varepsilon}$ and $\bar{\mu}$ are tensors depending on $\mathbf{x}$, t, $\mathbf{E}$ and $\mathbf{B}$. To simplify the notation, a few assumptions can be made:

- the considered materials are linear, thus $\bar{\varepsilon}$ and $\bar{\mu}$ are independent of **E** and **B**;
- materials are isotropic, which means $\bar{\varepsilon} \equiv \varepsilon \, \mathbb{I}_3$ and $\bar{\mu} \equiv \mu \, \mathbb{I}_3$;
- materials are homogeneous, i.e. ε and μ are constant in a give material;
- ε and μ are independent of time.

Hence, in such a material with constant permittivity ε and permeability μ, (2.6) and (2.7) become:

$$\mathbf{D} = \varepsilon \, \mathbf{E},$$
$$\mathbf{B} = \mu \, \mathbf{H}.$$

The vacuum permittivity and permeability ε_0 and μ_0 can be introduced, as well as the relative permittivity and permeability, ε_r and μ_r, of the considered material. The previous equations read as:

$$\mathbf{D} = \varepsilon_0 \varepsilon_r \, \mathbf{E},$$
$$\mathbf{B} = \mu_0 \mu_r \, \mathbf{H}.$$

Then is straightforward to obtain Maxwell's equations for linear, homogeneous, isotropic, nondispersive materials:

$$\begin{cases} \nabla \times \mathbf{E} & = -\mu_0 \mu_r \, \frac{\partial \mathbf{H}}{\partial t}, \\ \nabla \times \mathbf{H} & = \varepsilon_0 \varepsilon_r \, \frac{\partial \mathbf{E}}{\partial t} + \mathbf{J}. \end{cases} \tag{2.8}$$

This system can now be solved with the appropriate initial and boundary conditions.

2.2 Analytical solutions

Analytical solutions to electromagnetic propagation problems play a crucial role in validating numerical implementations of electromagnetic solvers. In this section, we present solutions to two elementary propagation problems. These solutions will serve as reference benchmarks in the subsequent chapters.

Vacuum-filled perfect electric conductor cavity

As a first example, let us consider the solution of the two-dimensional vacuum

Maxwell's equations in what is known as transverse magnetic form (TM). These are given as

$$\begin{aligned}\bar{\mu}\frac{\partial \tilde{H}^x}{\partial \tilde{t}} &= -\frac{\partial \tilde{E}^z}{\partial \tilde{y}},\\ \bar{\mu}\frac{\partial \tilde{H}^y}{\partial \tilde{t}} &= -\frac{\partial \tilde{E}^z}{\partial \tilde{x}},\\ \bar{\varepsilon}\frac{\partial \tilde{E}^z}{\partial \tilde{t}} &= \frac{\partial \tilde{H}^y}{\partial \tilde{x}} - \frac{\partial \tilde{H}^x}{\partial \tilde{y}}.\end{aligned} \tag{2.9}$$

Here, we have tow magnetic fields, $\left(\tilde{H}^x, \tilde{H}^y\right)$, and the electric field, $\tilde{E}^z$, all functions of $\left(\tilde{x}, \tilde{y}, \tilde{t}\right)$. All fields and units are dimensional. Furthermore, we have the magnetic permeability, $\bar{\mu}$, and the electric permittivity, $\bar{\varepsilon}$, which reflect the material coefficients.

In the following, we wish to model a metallic air-filled cavity, $\Omega = [-1{,}1]^2$, with perfect electrical conductor (PEC) boundary conditions, as depicted in Figure 2.1.

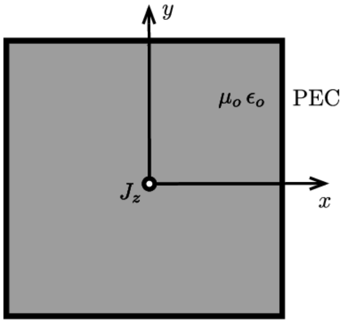


Figure 2.1: Metallic air-filled cavity $\Omega = [-1{,}1]^2$.

We can simplify the equations in this case, since $\bar{\mu} = \mu_0$ and $\bar{\varepsilon} = \varepsilon_0$ are the constant vacuum values. If we introduced the vacuum speed of light defined as

$$c_0 = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \simeq 3 \times 10^8 \text{m/s},$$

we can consider the normalized system of equations in the form

$$\begin{cases} \dfrac{\partial H^x}{\partial t} &= -\dfrac{\partial E^z}{\partial y}, \\ \dfrac{\partial H^y}{\partial t} &= -\dfrac{\partial E^z}{\partial x}, \\ \dfrac{\partial E^z}{\partial t} &= \dfrac{\partial H^y}{\partial x} - \dfrac{\partial H^x}{\partial y}, \end{cases}$$

where the unit-free variables are obtained as

$$t = \frac{c_0 \tilde{t}}{L}, \quad \mathbf{x} = \frac{\tilde{\mathbf{x}}}{L}, \quad \mathbf{H} = \frac{\tilde{\mathbf{H}}}{H_0}, \quad E^z = (Z_0)^{-1} \frac{\tilde{E}^z}{H_0}.$$

Here, H_0 is a unit magnetic field strength, $Z_0 = \sqrt{\mu_0/\varepsilon_0} \simeq 120\pi$ Ohms is the vacuum impedance, and L is some reference length, typically the wavelength of the phenomena of interest. For the boundary conditions, we assume that the walls of the cavity are perfectly electrically conducting such that the tangential component of the electric field, E^z, vanishes at the wall, i.e., $E^z = 0$ on the boundary of the domain. The exact cavity solution is given as

$$\begin{aligned} H^x(x,y,t) &= -\frac{\pi n}{\omega} \sin(m\pi x)\cos(n\pi y)\sin(\omega t), \\ H^y(x,y,t) &= \frac{\pi m}{\omega} \cos(m\pi x)\sin(n\pi y)\sin(\omega t), \\ E^z(x,y,t) &= \sin(m\pi x)\sin(n\pi y)\cos(\omega t), \end{aligned} \tag{2.10}$$

where the resonance frequencies, ω, are given as

$$\omega = \pi\sqrt{m^2 + n^2}, \quad (m,n) \geq 0.$$

Perfect electric conductor cavity with two material interfaces

In [23], a lossless dielectric with a relative permittivity ε_1 is enclosed by air in the x direction. The media are nonmagnetic and homogeneous along the y direction, as shown in Figure 2.2. The computational domain $\Omega = [-1,1]^2$ is bounded by PEC walls. The permittivity is given as $\varepsilon = \varepsilon_0$ if $1/2 \leq |x| \leq 1$ and $|y| \leq 1$, and $\varepsilon = \varepsilon_1$ if $|x| \leq 1/2$ and $|y| \leq 1$, where $\varepsilon_0 = 1$ and $\varepsilon_1 = 2.25$. The exact time-domain solution is [24]

$$E^z = \begin{cases} \sin\left(\frac{\omega_2}{2}\right)\sin\left(\omega_1(x+1)\right)\sin\left(\omega_y y\right)\cos\left(\omega t\right), & -1 \leq x < -1/2, \quad |y| \leq 1, \\ -\sin\left(\frac{\omega_1}{2}\right)\sin\left(\omega_2 x\right)\sin\left(\omega_y y\right)\cos\left(\omega t\right), & -1/2 \leq x \leq 1/2, \quad |y| \leq 1, \\ \sin\left(\frac{\omega_2}{2}\right)\sin\left(\omega_1(x-1)\right)\sin\left(\omega_y y\right)\cos\left(\omega t\right), & 1/2 \leq x \leq 1, \quad |y| \leq 1, \end{cases}$$

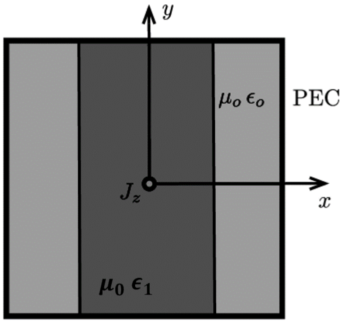


Figure 2.2: Metallic dielectric cavity $\Omega = [-1,1]^2$.

$$H^x = \begin{cases} -\frac{\omega_y}{\omega}\sin\left(\frac{\omega_2}{2}\right)\sin\left(\omega_1(x+1)\right)\cos\left(\omega_y y\right)\sin\left(\omega t\right), & -1 \le x < -1/2,\ |y| \le 1, \\ \frac{\omega_y}{\omega}\sin\left(\frac{\omega_1}{2}\right)\sin\left(\omega_2 x\right)\cos\left(\omega_y y\right)\sin\left(\omega t\right), & -1/2 \le x \le 1/2,\ |y| \le 1, \\ -\frac{\omega_y}{\omega}\sin\left(\frac{\omega_2}{2}\right)\sin\left(\omega_1(x-1)\right)\cos\left(\omega_y y\right)\sin\left(\omega t\right), & 1/2 \le x \le 1, \quad |y| \le 1, \end{cases}$$

$$H^y = \begin{cases} \frac{\omega_y}{\omega}\sin\left(\frac{\omega_2}{2}\right)\cos\left(\omega_1(x+1)\right)\sin\left(\omega_y y\right)\sin\left(\omega t\right), & -1 \le x < -1/2, \quad |y| \le 1, \\ -\frac{\omega_y}{\omega}\sin\left(\frac{\omega_1}{2}\right)\cos\left(\omega_2 x\right)\sin\left(\omega_y y\right)\sin\left(\omega t\right), & -1/2 \le x \le 1/2, \quad |y| \le 1, \\ \frac{\omega_y}{\omega}\sin\left(\frac{\omega_2}{2}\right)\cos\left(\omega_1(x-1)\right)\sin\left(\omega_y y\right)\sin\left(\omega t\right), & 1/2 \le x \le 1, \quad |y| \le 1, \end{cases}$$

where $\omega_1^2 + \omega_y^2 = \varepsilon_0\omega^2$ and $\omega_2^2 + \omega_y^2 = \varepsilon_1\omega^2$. The value of ω_y is determined according to the relation:

$$\sqrt{\varepsilon_1\omega^2 - \omega_y^2}\,\tan\left(\frac{\sqrt{\varepsilon_0\omega^2 - \omega_y^2}}{2}\right) = \sqrt{\varepsilon_0\omega^2 - \omega_y^2}\,\tan\left(-\frac{\sqrt{\varepsilon_1\omega^2 - \omega_y^2}}{2}\right). \tag{2.11}$$

The aforementioned relations are obtained thanks to the solution of the wave equation with the separation of variable method. The latter is a resonance condition emerging when continuity conditions are considered for the tangential electric and magnetic fields at the dielectric interfaces. As in [24] we choose $\omega_y = 2\pi$ to satisfy the PEC conditions on $y = \pm 1$ which leads to $\omega = 9.07716175885174$. Across the dielectric interface, the E^z and H^x components, their derivatives, and their first y derivative are continuous while their first x derivative is discontinuous. To generate the datasets exploited in Chapter 4 we will sample different values of ε_1 and, thus, solve the relation 2.11 with the dichotomy method discussed in Appendix A.

Chapter 3

Numerical discretization

In this chapter, we will be discussing the discretization method used in this study, which is the Discontinuous Galerkin method (DG). DG combines the finite element method (FEM) and finite volume method (FVM), using a space of basis and test functions similar to the finite element method. However, DG satisfies the considered equation in a way that is closer to the FVM, resulting in several advantageous properties.

3.1 Discontinuous Galerkin method

As presented in [25], time-dependant wave-dominated problems, e.g. time-domain Maxwell's equations, emerge as the main candidates for problems where the DG approach is advantageous. We remark that, even though the structure of the DG method is very similar to that of the FEM, there are several fundamental differences.

In particular, the DG method features a local mass matrix rather than a global one, allowing for less expensive inversion and resulting in an explicit semidiscrete scheme. Furthermore, by carefully designing the numerical flux to capture the underlying dynamics, one can achieve greater flexibility compared to classical FEM in ensuring stability for wave-dominated problems. Unlike the FVM, the DG method overcomes the main limitation of obtaining high-order accuracy on general grids by enabling this through the local element-based basis. All this is achieved while retaining advantages such as local conservation and flexibility in choosing the numerical flux. On the other hand, with the DG method, one has to face the challenge of an increase in the total degrees of freedom, which is a direct consequence of the decoupling of the elements.

Weak formulation

Let us consider a domain $\Omega \subset \mathbb{R}^2$, and let $\mathbf{n}$ be the unitary outward normal to its boundary $\partial\Omega$. We introduce a discretization of Ω, Ω_h, relying on a quasi-uniform triangulation $\mathcal{T}_h$ verifying $\mathcal{T}_h = \bigcup_{i=1}^{N} T_i$, where N is the number of mesh elements. The internal faces of the discretization are denoted as $a_{ik} = T_i \cap T_k$ if T_i and T_k are adjacent cells, and $\mathbf{n}_{ik}$ are defined as the unit normal vectors to the face a_{ik}, oriented from T_i to T_k. By taking the L^2 scalar product of each component with a vector test function ψ, the system (2.8), in the adimensionalised form, can be recast into the following variational problem: Find $(\mathbf{E},\mathbf{H}) \in H_0(\mathbf{curl},\Omega_h) \times H(\mathbf{curl},\Omega_h)$ such that $\forall \psi \in H(\mathbf{curl},\Omega_h)$,

$$\begin{aligned}\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \psi \,\mathrm{d}\Omega + \int_{T_i} \nabla \times \mathbf{E} \cdot \psi \,\mathrm{d}\Omega &= \mathbf{0},\\ \int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \psi \,\mathrm{d}\Omega - \int_{T_i} \nabla \times \mathbf{H} \cdot \psi \,\mathrm{d}\Omega &= -\int_{T_i} \mathbf{J} \cdot \psi \,\mathrm{d}\Omega.\end{aligned}$$

When we formally rewrite the previously mentioned equalities using classical vector calculus and Green's formulas, we get:

$$\begin{aligned}\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \psi \,\mathrm{d}\Omega + \int_{T_i} \mathbf{E} \cdot \nabla \times \psi \,\mathrm{d}\Omega &= \int_{\partial T_i} (\psi \times \mathbf{E}) \cdot \mathbf{n}_i \,\mathrm{d}\Gamma,\\ \int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \psi \,\mathrm{d}\Omega - \int_{T_i} \mathbf{H} \cdot \nabla \times \psi \,\mathrm{d}\Omega &= -\int_{T_i} \mathbf{J} \cdot \psi \,\mathrm{d}\Omega - \int_{\partial T_i} (\psi \times \mathbf{H}) \cdot \mathbf{n}_i \,\mathrm{d}\Gamma.\end{aligned}$$

Considering the properties of the mixed product, the latter becomes:

$$(\psi \times \mathbf{E}) \cdot \mathbf{n}_i = (\mathbf{E} \times \mathbf{n}_i) \cdot \psi.$$

Hence, $\forall\, T_i,\ \forall\, \psi \in H^1(\Omega_h)$,

$$\begin{aligned}\int_{T_i} \mu_r \frac{\partial \mathbf{H}}{\partial t} \cdot \psi \,\mathrm{d}\Omega + \int_{T_i} \mathbf{E} \cdot \nabla \times \psi \,\mathrm{d}\Omega &= \int_{\partial T_i} (\mathbf{E} \times \mathbf{n}_i) \cdot \psi \,\mathrm{d}\Gamma,\\ \int_{T_i} \varepsilon_r \frac{\partial \mathbf{E}}{\partial t} \cdot \psi \,\mathrm{d}\Omega - \int_{T_i} \mathbf{H} \cdot \nabla \times \psi \,\mathrm{d}\Omega &= -\int_{T_i} \mathbf{J} \cdot \psi \,\mathrm{d}\Omega - \int_{\partial T_i} (\mathbf{H} \times \mathbf{n}_i) \cdot \psi \,\mathrm{d}\Gamma.\end{aligned}$$

Space discretization

First, we define the following approximation space V_h:

$$V_h = \left\{ v \in (L^2(\Omega))^2,\ v|_{T_i} \in (\mathbb{P}_p(T_i))^2,\ \forall\, T_i \in \mathcal{T}_h \right\}$$

where $\mathbb{P}_p(T_i)$ is the space of polynomials of maximum degree p on T_i. The semi-discrete fields sought in space V_h, are denoted $(\mathbf{H}_h, \mathbf{E}_h, \mathbf{J}_h)$, and on each cell we define the restrictions $(\mathbf{H}_i, \mathbf{E}_i, \mathbf{J}_i)$. A set of scalar basis functions $(\phi_{ik})_{1 \le k \le d_i}$ is defined for each T_i, where d_i is the number of degrees of freedom per dimension. We notice that, in a $2D$ system, $\mathbf{E}_i$ is actually a vector that has two components:

$$\mathbf{E}_i = \left[E_i^x, E_i^y\right]^T,$$

each of which is locally expanded on the chosen set of basis functions:

$$E_i^v = \sum_{j=1}^{d_i} E_{ij}^v(t)\phi_{ij}(x) = \sum_{j=1}^{d_i} \hat{E_{ij}^v}(x_{ij}, t)\ell_{ij}(x), \quad v \in \{x, y\}. \tag{3.1}$$

That is the first step to get a matrix-vector form of the system. In our case, instead of using a local polynomial basis, we will express the polynomial through the associated interpolated Lagrange polynomial $\ell_{ij}(x)$, as in [25]. The connection between the modal and nodal representations is achieved through a generalized Vandermnonde matrix, as $\mathcal{V}E = \hat{E}$. The same procedure can be done for the $\mathbf{H}$ evolution equation.

Numerical fluxes

Given that the test functions are now allowed to be discontinuous at the interfaces between cells, it is important to notice that the surface integrals, such as:

$$\int_{a_{il}} (\mathbf{E}_h \times \mathbf{n}_{il}) \cdot \psi \, \mathrm{d}\Gamma,$$

and

$$\int_{a_{il}} (\mathbf{H}_h \times \mathbf{n}_{il}) \cdot \psi \, \mathrm{d}\Gamma$$

are not well-defined. The introduction of a numerical flux facilitates the restoration of a proper definition of surface integrals and is crucial for connecting field values between neighboring cells. It is important to note, however, that there is not a singular choice for fluxes. In the context of a set of linear equations, various selections can yield stable and convergent discrete schemes. Consequently, the surface integrals mentioned above are replaced with the following expressions:

$$\int_{a_{il}} (\mathbf{E}_* \times \mathbf{n}_{il}) \cdot \psi \, \mathrm{d}\Gamma$$

and

$$\int_{a_{il}} (\mathbf{H}_* \times \mathbf{n}_{il}) \cdot \psi \, \mathrm{d}\Gamma.$$

As demonstrated in [2], the flux can be interpreted as the solution of a Riemann problem at cell interfaces. In the subsequent discussion, we will utilize the centered flux, which is expressed as:

$$\mathbf{E}_* = \frac{\mathbf{E}_i + \mathbf{E}_l}{2}, \quad \mathbf{H}_* = \frac{\mathbf{H}_i + \mathbf{H}_l}{2}. \tag{3.2}$$

This flux is non-dissipative, and coupled with a non-dissipative time-integration scheme, will lead to a non-dissipative Discontinuous Galerkin time domain scheme. This property of the scheme is of fundamental importance since our test cases are closed cavities and we want to ensure mass conservation.
All this choices, as detailed in [2], lead to the following semidiscrete scheme :

$$\begin{aligned}
\bar{\mathbb{M}}_i^{\mu_r} \frac{\partial \bar{\mathbf{H}}_i}{\partial t} &= -\bar{\mathbb{K}}_i \times \bar{\mathbf{E}}_i + \sum_{l \in \mathcal{V}_i} \bar{\mathbb{S}}_{il} (\bar{\mathbf{E}}_* \times \mathbf{n}_{il}), \\
\bar{\mathbb{M}}_i^{\varepsilon_r} \frac{\partial \bar{\mathbf{E}}_i}{\partial t} &= -\bar{\mathbb{K}}_i \times \bar{\mathbf{H}}_i + \sum_{l \in \mathcal{V}_i} \bar{\mathbb{S}}_{il} (\bar{\mathbf{H}}_* \times \mathbf{n}_{il}) - \bar{\mathbb{M}}_i \bar{\mathbf{J}}_i,
\end{aligned}$$

where $\bar{\mathbb{M}}$, $\bar{\mathbb{K}}$ and $\bar{\mathbb{S}}$, are respectively the extended mass matrix, the stiffness matrix, and the flux matrix, that are defined as presented in the following:

$$\bar{\mathbb{M}}_i^u = \begin{bmatrix} \mathbb{M}_i^u & \mathbf{0}_{d_i\times d_i} & \mathbf{0}_{d_i\times d_i} \\ \mathbf{0}_{d_i\times d_i} & \mathbb{M}_i^u & \mathbf{0}_{d_i\times d_i} \\ \mathbf{0}_{d_i\times d_i} & \mathbf{0}_{d_i\times d_i} & \mathbb{M}_i^u \end{bmatrix}, \quad \bar{\mathbb{S}}_{il} = \begin{bmatrix} \mathbb{S}_{il} & \mathbf{0}_{d_i\times d_i} & \mathbf{0}_{d_i\times d_i} \\ \mathbf{0}_{d_i\times d_i} & \mathbb{S}_{il} & \mathbf{0}_{d_i\times d_i} \\ \mathbf{0}_{d_i\times d_i} & \mathbf{0}_{d_i\times d_i} & \mathbb{S}_{il} \end{bmatrix} \text{ and } \quad \bar{\mathbb{K}}_i = \begin{bmatrix} \mathbb{K}_i^x \\ \mathbb{K}_i^y \\ \mathbb{K}_i^z \end{bmatrix},$$

while the matrices used to construct the blocks of the extended matrices are defined as follows:

$$\begin{aligned} (\mathbb{M}_i^{\varepsilon_r})_{jk} &= \int_{T_i} \varepsilon_r \phi_{ij}\phi_{jk}\, \mathrm{d}\Omega, \\ (\mathbb{K}_i^v)_{jk} &= \int_{T_i} \phi_{ij}\frac{\partial \phi_{ik}}{\partial v}\, \mathrm{d}\Omega \quad \text{for } v \in \{x, y, z\}, \\ (\mathbb{S}_{il})_{jk} &= \int_{a_{il}} \phi_{ij}\phi_{jk}\, \mathrm{d}\Gamma \end{aligned}$$

with $(j, k) \in [1, d_i]2$.

3.2 Time advancing scheme

So far, we have considered the spatial dimension and a discrete representation of the latter. This reflects a method-of-lines approach where space and time are discretized. Once the semi-discrete system in space is obtained, one can use some standard techniques to solve these ordinary differential equations.

As in [25], a low-storage explicit Runge-Kutta (LSERK) method is exploited for discretization in time. LSERK is an alternative to the explicit Runge-Kutta, that allows to reduce the memory usage, requiring only one additional storage level. On the other hand, this comes at the price of an additional function evaluation, as the low-storage version has more stages. At first, it would seem that the additional stage makes the low-storage approach less interesting due to added cost, however, this is offset by allowing a larger stable timestep, Δt. One of the most widely used LSERK schemes in computational electromagnetics is the 5-stage fourth-order algorithm proposed by Kennedy and Carpenter [26]. We consider a generalized semidiscrete problem

$$\frac{\mathrm{d}\mathbf{u}_h}{\mathrm{d}t} = \mathcal{L}_h(\mathbf{u}_h, t),$$

where $\mathbf{u}_h$ is the vector of the unknowns, the n^{th} step of the algorithm has the following form:

$$\begin{aligned} &\mathbf{p}^{(0)} = \mathbf{u}_h^n, \\ &i \in [1, ..., 5] : \begin{cases} \mathbf{k}^{(i)} = a_i\mathbf{k}^{(i-1)} + \Delta t\mathcal{L}_h(\mathbf{p}^{(i-1)}, t^n + c_i\Delta t), \\ \mathbf{p}^{(i)} = \mathbf{p}^{(i-1)} + b_i\mathbf{k}^{(i)}, \end{cases} \\ &\mathbf{u}_h^{n+1} = \mathbf{p}^{(5)}, \end{aligned}$$

where a_i , b_i and c_i are constant coefficients.

3.3 Mesh convergence

To generate the snapshots dataset, for the test cases presented in Chapter 2, an adaptation of the MATLAB scripts in the text on Nodal Discontinuous Galerkin methods by Jan S. Hestaven and Tim Wartburton has been used [25, 27]. The meshes have been generated with the MATLAB package PDE modeler, and their features are reported in Table 3.1. In Figure 3.1 we show the convergence of the electric field, E^z, under both element and order refinement. We show the results computed using central flux. Similar convergence behavior can be observed for the other field components. When comparing the results, in Table 3.2, we see that there are indications of an even-odd pattern with the accuracy being $\mathcal{O}(h^{N+1})$ for N even and $\mathcal{O}(h^N)$ for N odd; such behavior is often observed when central fluxes are used.

	Nodes	Elements	h_{min}
Mesh 1	177	312	0.2
Mesh 2	665	1248	0.09
Mesh 3	2577	4992	0.045

Table 3.1: Features of the grids.

$N+1$	$h=0.2$	$h=0.09$	$h=0.045$	Estimated order
2	1.46e-01	3.66e-02	9.23e-3	1.85
3	8.91e-03	1.13e-03	1.42e-04	2.77
4	4.40e-04	3.24e-05	2.89e-06	3.37

Table 3.2: h-Convergence results for the PEC cavity problem with a centered fluxes DG discretization in the time interval [0,1].

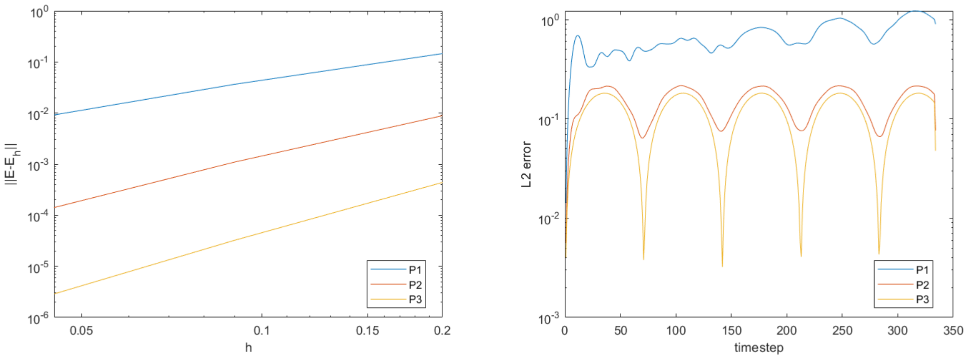


Figure 3.1: *h*-Convergence results for the PEC cavity problem with a centered fluxes DG discretization in the time interval [0,1] (left) and evolution of the L_2 error in time (right).

Chapter 4

Reduced order modelling

Parameterized partial differential equation models are frequently encountered in engineering and applied sciences, where the parameters include geometric features, boundary conditions, and physical properties. These parameterized models implicitly establish connections between input parameters and outputs of interest.

While developing accurate computational tools to solve such problems is of broad interest, our focus lies on scenarios where solutions are sought for a multitude of parameters. Typical applications of relevance include optimization, control, design, uncertainty quantification, real-time query, and others. In such cases it is not only the accuracy of the model that matters, but also the computational efficiency of the model.

With applications characterized by parameterized problems that require repeated evaluation, it is clear that we need to seek alternatives to simply solve the full-order problem many times. This is exactly the place where reduced models need to be developed.

The reduced order modeling is based on a two-stage procedure, consisting of an offline and online stage. In the potentially very costly offline stage, one empirically explores the solution manifold to construct an approximation of the latter, exploiting a finite, and as small as possible, number of high-fidelity solutions. The online stage consists of the *fast* evaluation of the ROM solution, with a varying set of parameters $\mu \in \mathbb{P}$. In this stage, one can explore the parameter space $\mathbb{P}$ at a substantially reduced cost, ideally at a cost that is independent of the dimension of the high-fidelity model.

As presented in Chapter 1, the intrusive nature of projection-based model order reduction can compromise the efficiency of traditional reduced order models when dealing with nonlinear and non-affine problems. Non-intrusive and data-driven ROMs offer an alternative approach to these techniques by reducing computational complexity without needing to project the governing equation. The first step in constructing such ROMs involves a dataset built up from the previously introduced snapshots. Various linear and non-linear methodologies have been developed and studied for this purpose. Moreover, machine learning approaches are beneficial in non-intrusive methods, such as Neural Networks, CNNs, and GNNs. In particular, the GCA showed good performances, with respect to classical linear and intrusive approaches, when dealing with advection dominated phenomena, which is the case in our physical context.

In this chapter, we will present the main features of the GCA architecture [6], and discuss its application to the context of Maxwell's equations with a DGTD scheme.

4.1 Reduced Order Models

Full-order models are designed to solve PDEs by utilizing high-fidelity systems of equations. These systems are characterized by a large number of degrees of freedom, denoted by N_h, which makes the numerical solution process computationally expensive. This computational cost becomes particularly significant in the context of parameterized PDEs, where the goal is to obtain real-time evaluations for a variety of different physical or geometrical configurations. Each configuration is linked to a unique parameter vector μ, resulting in a parameter-dependent high-fidelity solution $\mathbf{u}_h(\mu)$.

In practical applications, where real-time solution recovery is required, solving such large-scale systems repeatedly for different parameter values becomes infeasible due to the high computational cost. To overcome this challenge, ROMs are employed. ROMs aim to create and solve a much less expensive reduced system, which is derived from the original full-order model but has a significantly lower number of degrees of freedom, denoted as N, where $N \ll N_h$. This reduced system yields the reduced order solution $\mathbf{u}_N(\mu)$, which approximates the behavior of the full system but at a fraction of the computational cost.

Once the reduced order solution $\mathbf{u}_N(\mu)$ is obtained, it can be used to estimate the high-fidelity solution $\mathbf{u}_h(\mu)$ through a transformation, typically expressed as $\mathbf{u}_h(\mu) \approx \phi(\mathbf{u}_N(\mu))$. The function ϕ represents a mapping that can be either linear or nonlinear, depending on the ROM technique being used. This mapping ensures that the reduced model captures the essential dynamics of the full-order model, thereby providing an accurate approximation of the high-fidelity solution.

ROMs can be developed using either intrusive or non-intrusive approaches. Intrusive methods require direct access to and manipulation of the full-order model's governing equations. This means that detailed knowledge of the high-fidelity system is necessary to construct the reduced system and to compute the reduced order solution $\mathbf{u}_N(\mu)$. In contrast, non-intrusive methods do not require such access; instead, they rely on external data, such as snapshots of high-fidelity solutions for various parameters, to build the ROM. These non-intrusive methods often employ machine learning or statistical techniques to learn the relationship between the parameters and the solution, making them more flexible but sometimes less accurate than intrusive methods.

Overall, ROMs [28, 29] provide a powerful mean of reducing computational costs in the solution of parameterized PDEs, enabling real-time simulation and optimization in complex systems. By carefully selecting the appropriate ROM technique and mapping function ϕ, it is possible to achieve a balance between accuracy and efficiency, making ROMs an essential tool in many engineering and scientific applications.

4.1.1 Intrusive Model Order Reduction

Intrusive model order reduction (MOR) methods require knowledge of the high-fidelity system they aim to simplify. These methods directly interact with the system's governing equations, making them highly dependent on the precise structure and properties of the original model. To illustrate the application of these methods, let us focus on a linear high-fidelity system characterized by the following equation:

$$\mathbf{A}_h(\mu)\mathbf{u}_h(\mu) = \mathbf{f}_h(\mu).$$

In this equation, $\mu \in \mathbb{R}^{N_\mu}$ represents the vector of parameters that influence the system, such as physical properties or boundary conditions. The matrix $\mathbf{A}_h \in \mathbb{R}^{N_h \times N_h}$ is the stiffness matrix, which encapsulates the system's structural properties and interactions. The vector $\mathbf{u}_h \in \mathbb{R}^{N_h}$ is the high-fidelity solution, which provides a detailed and accurate representation of the system's response to the forcing term $\mathbf{f}_h \in \mathbb{R}^{N_h}$, the latter representing external influences or sources acting on the system.

To make the problem more tractable, linear MOR techniques seek to approximate the high-fidelity solution $\tilde{\mathbf{u}}_h$ using a reduced set of basis functions. These basis functions are chosen to efficiently capture the dominant features of the solution across the parameter space. Specifically, $\tilde{\mathbf{u}}_h$ is expressed as a linear expansion over these basis functions $\{\psi_i\}_{i=1}^{N}$, where each $\psi_i \in \mathbb{R}^{N_h}$ represents a basis vector that spans the reduced subspace. By introducing the matrix $\mathbf{V} = [\psi_1|\dots|\psi_N]$, which contains these basis vectors as columns, the approximation can be written as $\mathbf{u}_h \approx \mathbf{V}\mathbf{u}_N$, where $\mathbf{u}_N$ are the coefficients in the reduced space.

POD is a prevalent technique for constructing the basis $\{\psi_i\}$. POD works by analyzing a series of high-fidelity solutions, known as snapshots, which are computed for different parameter values. The objective of POD is to extract the principal components or modes that capture the most significant variations in the system's behavior. These modes form an optimal rank-N subspace that approximates the snapshot data in the least squares sense. This process ensures that the reduced model retains the most critical dynamics of the original high-fidelity model, while significantly reducing computational complexity.

Once the basis functions have been selected through POD or another method, the next step is to determine the reduced coefficients $\mathbf{u}_N$, which represent the system's behavior within the reduced subspace. Intrusive projection-based MOR approaches leverage the structure of the high-fidelity system by imposing a condition that the residual, i.e., the difference between the actual and approximated solutions when projected onto the reduced basis $\mathbf{V}$, should be zero. This condition leads to the following system of equations that must be solved for $\mathbf{u}_N$:

$$\mathbf{V}^T(\mathbf{f}_h - \mathbf{A}_h\mathbf{V}\mathbf{u}_N) = 0.$$

Solving this system yields the coefficients $\mathbf{u}_N$, which can then be used to reconstruct an approximate solution to the original high-fidelity problem. However, despite their effectiveness in certain scenarios, intrusive methods can be inefficient or challenging to apply in cases where the system exhibits non-affine parameter dependencies, meaning that the parameters do not influence the system in a straightforward linear manner. Additionally, these methods often struggle with complex or nonlinear problems, where the underlying assumptions of linearity and superposition may no longer hold. Consequently, while intrusive methods are powerful tools in MOR, their application is often limited to scenarios where the system's behavior is well-understood and sufficiently linear to permit accurate approximation.

4.1.2 Non-Intrusive Model Order Reduction

Non-intrusive methods offer a way to reduce computational complexity compared to projection-based approaches. Traditional methods, such as the RB [9] approach, rely on projecting the governing equations onto a reduced basis, which requires solving a potentially nonlinear reduced system to obtain the reduced coefficient vector $\mathbf{u}_N(\mu)$, whereas non-intrusive methods use different approaches to avoid this step. Examples of such methods include POD-NN [15] and Proper Orthogonal Decomposition with Interpolation (PODI) [30, 31]. POD-NN projects snapshots onto a reduced POD basis and then uses a neural network to predict the system's behavior, while PODI interpolates the reduced coefficients, enabling efficient recovery of approximate solutions. Another example of POD with a particular kind of interpolation is POD-CSI [5]. These approaches approximate high-fidelity coefficients relying on a linear scheme. Despite their efficiency, linear approaches struggle with complex systems where phenomena exhibit slow decay in the Kolmogorov n-width, such as in advection-dominated problems, where a large number of modes is required for accurate approximation [32]. To overcome these limitations, nonlinear reduction techniques have been developed, introducing nonlinear mappings to improve approximation quality.

To address this limitation, nonlinear approaches involving a nonlinear mapping Ψ, such that $\mathbf{u}_h \approx \Psi(\mathbf{u}_N)$, have been explored. For example, methods based on kernel principal component analysis [33], shifted POD [34], and nonlinear autoencoders aim to map reduced solutions through nonlinear transformations, thereby capturing the underlying dynamics more effectively. Autoencoders, in particular, have gained attention for their ability to generalize the POD technique, offering superior accuracy in problems with complex, nonlinear behavior.

For instance, in [12], the authors propose a nonlinear manifold least-squares Petrov-Galerkin method that utilizes convolutional autoencoders, while [17] introduces a hyper-reduced extension that leverages physical insights. The DL-ROM approach developed in [13] enhances the training process through a supervised task, allowing a feedforward neural network to effectively learn the bottleneck. Additionally, a POD preprocessing step is considered in [16] to streamline the network's size. Notably, the reconstruction of hyper-reduction operators is explored in [35], and dynamic integration informed by thermodynamic principles utilizes autoencoders to establish mappings to low-dimensional manifolds, as discussed in [36, 37]. Other non-intrusive methodologies, such as Gaussian Process Regression [38] and Operator Inference [39], have also gained traction in recent literature. Furthermore, Neural Operators have demonstrated success in approximating mappings between function spaces, moving beyond direct function approximations. These approaches commonly exploit both discretization-invariance and universal approximation properties [40, 41, 42]. In light of these developments, graph convolutional autoencoders emerge as a promising avenue for furthering nonlinear model order reduction techniques, integrating machine learning with classical PDE frameworks, particularly for solutions defined on unstructured grids. These approaches allow for encoding geometric features and handling more sophisticated physical systems, opening up new possibilities for the development of nonlinear MOR techniques. By combining the geometric flexibility of graph-based models with the efficiency of autoencoders, these methods represent a promising direction for improving both accuracy and scalability in MOR, making them well-suited for high-dimensional, parameterized problems in fields such as fluid dynamics, structural mechanics, and multiphysics simulations.

4.2 Graph Convolutional Autoencoder

In their work [6], the authors introduce a framework for non-linear model order reduction using a GCA. This autoencoder architecture extends the POD compression method in a non-linear way. On the other hand, GNNs provide a natural framework for analyzing PDE solutions on unstructured meshes.

The autoencoder is constituted by nonlinear encoding and decoding structures connected through a bottleneck. This bottleneck layer identifies the latent dimension and plays the role of the reduced space, or the approximation of the solution manifold. This setup exemplifies an unsupervised learning task. To construct the autoencoder architecture, both Fully Connected Neural Networks (FCNNs) [43] and CNNs have been investigated. CNNs have been studied for their spatial-related properties [44, 45, 12, 13], in order to incorporate geometrical features in the learning process, even though their natural context of application is a structured dataset and the aim in this case is to obtain a more versatile architecture, able to deal with unstructured meshes. In this setting, GNNs are employed to preserve the underlying geometric structure of the data.

CNN-based autoencoder architectures serve as the primary inspiration for the GCA.

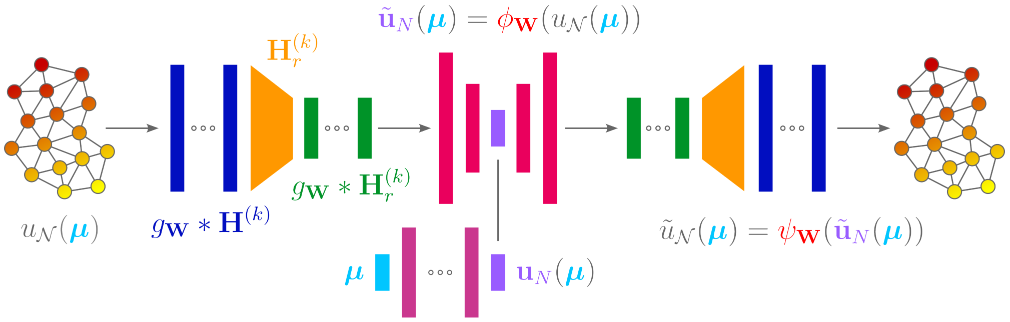
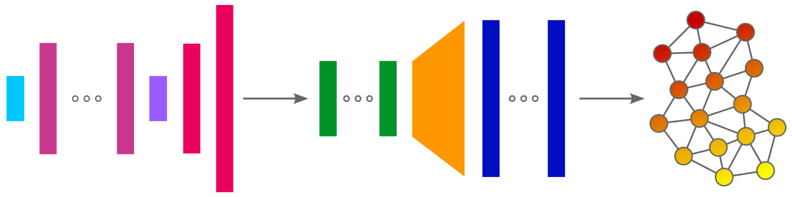

$\tilde{\mathbf{u}}_N(\boldsymbol{\mu}) = \phi_{\mathbf{W}}(u_{\mathcal{N}}(\boldsymbol{\mu}))$
$\mathbf{H}_r^{(k)}$
$u_{\mathcal{N}}(\boldsymbol{\mu})$
$g_{\mathbf{W}} * \mathbf{H}^{(k)}$
$g_{\mathbf{W}} * \mathbf{H}_r^{(k)}$
$\tilde{u}_{\mathcal{N}}(\boldsymbol{\mu}) = \psi_{\mathbf{W}}(\tilde{\mathbf{u}}_N(\boldsymbol{\mu}))$
$\boldsymbol{\mu}$
$\mathbf{u}_N(\boldsymbol{\mu})$

Figure 4.1: Offline phase for the GCA-ROM architecture [6].

These approaches, as mentioned earlier, are particularly effective when dealing with structured meshes, which can be seen as images with a fixed number of neighboring pixels. Consequently, a similar architecture has been explored to extend their applicability in a geometrically consistent manner to unstructured meshes defined over complex domains, where a Cartesian representation is no longer feasible. Machine learning approaches mirror the standard ROM setting, featuring an offline phase for dataset formation and reduced model construction via neural network training, and an online phase for real-time



$$\boldsymbol{\mu} \dashrightarrow \mathbf{u}_N(\boldsymbol{\mu}) \dashrightarrow \tilde{u}_{\mathcal{N}}(\boldsymbol{\mu}) = \psi_{\mathbf{W}}(\mathbf{u}_N(\boldsymbol{\mu}))$$

Figure 4.2: Online phase for the GCA-ROM architecture [6].

evaluation of the field of interest.

Graph Neural Networks

The GNN model [46] was first presented as an extension of the already existing neural network method for processing the data represented in graph domains. The idea behind GNNs is to encode the underlying graph structured data using the topological relationships among the nodes of the graph, to incorporate graph-structured information in the data-processing step. GNNs are based on an information diffusion mechanism. A graph is processed by a set of units, each one corresponding to a node of the graph, which are linked according to the graph connectivity. The units update their states and exchange information until they reach a stable equilibrium.

Let's consider a mesh $\mathcal{T}(\mathcal{V}, \mathcal{E}, \mathcal{F})$, which can be seen as a simple, undirected, and connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The nodes have associated features $\mathbf{u}$, that represent the evaluation of a set of state variables at the vertices of the mesh. Starting from the defined graph structure, a graph neural network is obtained by defining a set of optimizable operations that act on all attributes of the graph. The basic operations of many GNNs are (i) message-passing framework, (ii) convolutional layers, and (iii) down-sampling and up-sampling procedures.

We consider the graph dataset $\Xi = \left\{\mathbf{u}_{\mathcal{N}}(\mu^i),\, \Omega_{\mathcal{N}}(\mu^i)_{i=1}^{N_s}\right\}$ formed by N_s solutions $\mathbf{u}_{\mathcal{N}}(\mu^i)$ of a parameterized PDE defined over unstructured meshes $\Omega_{\mathcal{N}}(\mu^i)$. The architecture for the offline training is composed of an autoencoder and a multi-layer perceptron (MLP). The former seeks to approximate the identity map while encoding the information into low-dimensional space expressed by the bottleneck or latent space.

Message passing framework

The main idea is to propagate information to the local neighborhood of each node u, which is denoted by $N(u)$ and defines the computation graph with degree $|N(u)|$. Such messages are exchanged between nodes at different k-th hops computing the hidden embedding $\mathbf{h}_u^{(k)}$ of the node u. At a node $u \in \mathcal{V}$, one assemble the messages to be sent through the operation $\mathfrak{m}^{(k)}$ as

$$\mathbf{m}_v^{(k)} = \mathfrak{m}^{(k)}(\mathbf{h}_v^{(k-1)}), \quad \text{from each node}\, v \in N(u)$$

aggregates them with $\mathfrak{a}^{(k)}$ in

$$\mathbf{m}_{N(u)}^{(k)} = \mathfrak{a}^{(k)}(\{\mathbf{m}_v^{(k)}, \forall v \in N(u)\}),$$

and finally updates the hidden embedding through the function $\mathfrak{u}^{(k)}$

$$\mathbf{h}_u^{(k)} = \mathfrak{u}^{(k)}(\mathbf{m}_{N(u)}^{(k)}).$$

Initialization is defined by $\mathbf{h}_{v_j}^{(0)} = u_j$. Since the previous information must be preserved with each update of the embeddings, we also include ghost self-edges, meaning that we consider u as part of its neighborhood $\mathcal{N}(u)$.

Convolutional layers in the non-Euclidean setting

The graph convolutional network (GCN) learns to combine the hidden embeddings by defining convolutional operations able to capture the relationships between the nodes and optimizing a given loss function. MoNet [47] is used, which can be interpreted as a Gaussian Mixture Model, and thus a general class for convolutions in non-Euclidean domains. MoNet builds a set of pseudo-coordinates $\mathbf{e}$ used to define the weights of an optimizable Gaussian kernel with $\mathcal{Q}$ filters, through the iteration procedure

$$\mathbf{h}_u = \frac{1}{|N(u)|} \sum_{v \in N(u)} \frac{1}{Q} \sum_{q=1}^{Q} \omega^q(e_u) \odot \mathbf{W}^q \mathbf{h}_v,$$

where the weighting function ω^q is defined in terms of a trainable mean vector μ_q and a diagonal covariance matrix $\mathbf{\Sigma}_q$ as

$$\omega^q(\mathbf{e}_u) = \exp\left(-\frac{1}{2}(\mathbf{e}_u - \mu_q)^T \mathbf{\Sigma}_q^{-1}(\mathbf{e}_u - \mu_q)\right).$$

In practice, MoNet considers pseudo-coordinates as the edge attributes given by the distance between two connected nodes, introducing a geometric bias in the learning process.

Down-sampling and up-sampling procedures

Pooling is the most widely used technique to down-sample the size of the input by aggregating information from multiple nodes and edges. This results in a more manageable graph, improving generalization and performance, but there is no natural hierarchy in node importance. Un-pooling is still an open research challenge. PointNet++ [48]

proposes a k-NN interpolation of the points to up-sample by considering the position and the features of the nodes in the down-sampled coarser configuration. In particular, given a node to position $\mathbf{x}_i$, we define its feature vector $\mathbf{u}_i$ as the weighted interpolation

$$\mathbf{u}_i = \frac{\sum_{j=1}^{N(i)} \xi(\mathbf{x}_j \mathbf{u}_j)}{\sum_{j=1}^{N(i)} \xi(\mathbf{x}_j)},$$
$$\xi(\mathbf{x}_j) = \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j)^2}.$$

In our test cases, the pooling step will not be considered, as the information embedded in each node will be not just related to the value of a single state variable.

GCA architecture

We can now present the graph convolutional autoencoder for model order reduction applications. As detailed before, this approach, like standard ROMs, relies on two stages, the offline and the online stage. The architecture for offline learning (Figure 4.1) is composed of an autoencoder and a multi-layer perceptron (MLP). The former aims to mimic the identity map while encoding the information into a low-dimensional space represented by the bottleneck. The encoder module takes as input the graph data Ξ, exploiting MoNet [47] as the message passing algorithm. Employing the convolutions, the most meaningful information between the nodes and their evolution is extracted with respect to the samples. To reduce dimensionality, one can consider the additional module of down-sampling and a second batch of convolutional layers can be used to encode the latent space further. After these steps, an FCNN is used to connect the graph structure with the bottleneck. Thus, the initial information is encoded in the latent vector $\tilde{\mathbf{u}}_N(\mu) \in \mathbb{R}^N$.

The decoding structure consists of the same operations but in a reversed order. Thanks to the decoding process, the output data, which is a reconstruction $\tilde{\mathbf{u}}_\mathcal{N}$ of the input, can be confronted with the latter in terms of a loss function for the unsupervised learning task, defined as

$$\mathcal{L}_{MSE} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} ||\tilde{\mathbf{u}}_\mathcal{N}(\mu^i) - \mathbf{u}_\mathcal{N}(\mu^i)||_2^2,$$

where N_{tr} is the number of samples in the training set. Another term can be taken into account when evaluating the loss function, in fact, the supervised learning task, confronting the dataset and the latent vector, can be added, and also balanced by the hyperparameter λ. Hence we have the following loss functions

$$\mathcal{L}_{BTT} = \frac{1}{N_{tr}} \sum_{i=1}^{N_{tr}} ||\mathbf{u}_N(\mu^i) - \tilde{\mathbf{u}}_N(\mu^i)||_2^2,$$
$$\mathcal{L} = \mathcal{L}_{MSE} + \lambda \mathcal{L}_{BTT},$$

which guides the training procedure through the Adam optimizer. During the online phase (Figure 4.2), the bottleneck can be directly evaluated using the MLP to process a new parameter, μ, and then successfully decompressed through the graph decoder to recover the corresponding field defined over its geometry.

Regarding the preprocessing of the dataset, we will consider different normalization procedures. In addition to [6], we will also explore the non-normalized dataset.

4.3 Adaptation to the Discontinuous Galerkin framework

The GCA-ROM architecture, as described in [6], deals with a graph input. The data information used consists of nodal values obtained from classical FEM simulations as described in the previous sections. In the case of P_1 elements, each node on the mesh or graph has only one degree of freedom when creating the dataset. When dealing with P_2 elements, such as in test cases for Navier-Stokes equations, only the degrees of freedom at the mesh nodes were considered during the training step. It is worth mentioning that, despite neglecting information linked to the degrees of freedom on the edges of the elements, the results obtained are still satisfactory.

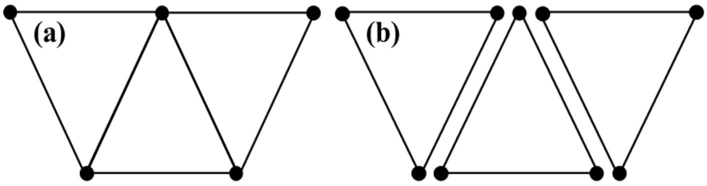


Figure 4.3: Representation of a classical Galerkin method (a), and representation of the DG method (b), for P_1 elements.

To make the method applicable to datasets obtained from DG numerical solutions, and to retain information for all degrees of freedom, a new definition for the input graph needs to be introduced. This is necessary because, e.g. in the case of P_1 elements, more than one value is defined in each node of the mesh, as depicted in Figure 4.3. Also, we might consider retaining the values defined on the edges of each element, since in our physical context, an optimal characterization of discontinuities in our quantity of interest is crucial.

In the following, first, we will consider the Averaged Discontinuous Galerkin Graph Convolutional Autoencoder (AVG-DG-GCA), an approach that considers the same architecture exploited with the FEM structure, so for each node, we will consider an average value of all the degrees of freedom linked to it, then we will present the results with the DG-GCA, obtained preserving all the degrees of freedom of the DG scheme, with the new graph structure. In this new graph structure, each node represents a single element of the triangulation, while the edges of the graph are no longer the edges of the elements, but the distance between the barycenters of the corresponding adjacent elements. In this case, a different preprocessing should be done, since we are no longer associating the degrees of freedom to a specific coordinate. Still, we will associate them with the corresponding element. Thus the resulting snapshot matrix will consist of column vectors with components $u_{l,m}(\mu^k)$, with $l = 1 \dots N$, $k = 1 \dots N_s$, $m = 1 \dots K$, with K being the number of the degrees of freedom per element associated to the chosen method order. Another path, not explored in this work, could be to exploit the properties of the DG scheme and consider just the value of our quantity of interest in the barycentre, which can be obtained knowing the values of the degrees of freedom and the basis functions.

Moreover, in our 2D test cases, we have different components for the electric and the magnetic fields, thus we are interested in having a GCA architecture able to handle all our state variables at once. This feature is encountered also in the Navier-Stokes example since we have both velocity and pressure, but in [6] the two fields are treated separately.

Chapter 5

Numerical results

This chapter presents results for the test cases introduced in Chapter 2. The first section will focus on the test case with a single value for the permittivity, and the only parameter will be time. In the second section, we will treat a slightly more complex problem, taking into account a variation in the composition of the materials inside the PEC cavity, thus we will consider different values for the permittivity.

5.1 Time as parameter

As a first step, we will deal with the problem of the vacuum-filled perfect electric conductor cavity presented in Chapter 2. This toy problem lacks practical applications but is valuable for understanding qualitative behavior in our physical context.

In order to study the behavior of the AVG-DG-GCA method using our physical case, we initially conducted an analysis of its performance while changing the dimensions of the dataset. This involved varying the number of snapshots, the size of the considered meshes, and the dataset partitioning into training and testing subsets. As shown in Table 5.1, better results are obtained with finer meshes, as we expected since the more nodes a network has, the more complex and diverse features of the solution field it can learn. It is important to note that certain parameters of the architecture need to be adjusted when working with coarse meshes. In compressing procedures, the input nodes of the MLP may have a higher number of nodes than the triangulation vertices. As a result, the GCA architecture attempts to compress the information in a higher dimension, thus leading to an inaccurate representation of the input dataset.

One of the promising aspects of the GCA architecture is that it can obtain good results by exploiting just the 30% of the dataset for the training process. In Table 5.2, we see that a quite good error is obtained with the smaller training dataset and that increasing it leads to possible overfitting problems. After this first examination of the behavior of the architecture depending on the chosen hyperparameters, we are able to obtain quite satisfactory results for all three components of our electromagnetic field, as shown in Figure 5.1, using a dataset of 336 snapshots and a learning rate $l_r = 0.1$.

One important tool to detect critical points in the learning procedure is to plot the relative error in function of the parameter space. In fact, in Figure 5.3, the evolution of our numerical solution reaches a critical point at half a period. The explanation for the presence of this peak is that the related snapshots approximate a solution that is close to zero. The problem is not just that we are considering the relative error, thus dividing by a quantity close to zero, but also that in the scaling process, we are considering a standard scaling that takes into account all the snapshots. So, part of the difficulties in the learning process here relies also on the physics of the problem what we have is a solution, that preserves its shapes, while varying sign and amplitude, in fact, training over 3 periods leads to an increase in the mean relative error $\bar{e}_r = 5.99e-01$ and qualitatively much worse results (Figure 5.2), while if we consider a dataset of snapshots ranging on half a period we obtain a mean relative error that decrease of two orders of magnitude. A similar behavior is present also for both the components of the magnetic field, in particular, in this case, we have a zero initial condition that strongly impacts the learning process. Thus, we also removed the initial condition snapshot from the dataset. In Figure 5.4 it is displayed how the information encoded in the latent space is flattened in the case of standard scaling, while more variance is captured when imposing the identity scaling.

nodes	1220	317	90
max relative error	1.04e+00	7.61e-01	5.13e-01
mean relative error	2.50e-02	8.54e-02	7.08e-02
min relative error	2.76e-03	5.94e-02	3.41e-02

Table 5.1: Relative errors with fine and coarse mesh (from left to right) with $r_t = 30\%$ and $l_r = 10$, obtained with the AVG-DG-GCA method.

r_t	30%	60%	70%
max relative error	1.044e+00	9.04e-01	4.42e-01
mean relative error	2.50e-02	1.75e-01	1.48e-01
min relative error	2.76e-03	1.36e-01	1.29e-01

Table 5.2: Relative errors obtained increasing the dimension of the training set, with 1220 nodes and $l_r = 10$, obtained with the AVG-DG-GCA method.

All the considerations done with the AVG-DG-CGA are a useful starting point to set our framework of parameters in the case of the DG-GCA. Indeed we will neglect the initial condition and train with the 30% of the initial dataset over half a period. Since we are now considering all the degrees of freedom, our dataset will have a greater size, thus leading to more information to be incorporated in the learning phase, in fact, in Table 5.4 we see that the training time for the DG-GCA with P_1 elements is twice with respect to the AVG-DG-CGA training time, while for grids with 1220 nodes it is almost 20 minutes, and for the AVG-DG-GCA it is about 8 minutes. Furthermore, increasing the order of the DG method does not significantly affect the training time, but it allows us to work with a more accurate dataset of snapshots. Comparing Figure 5.5 and Figure 5.4 we

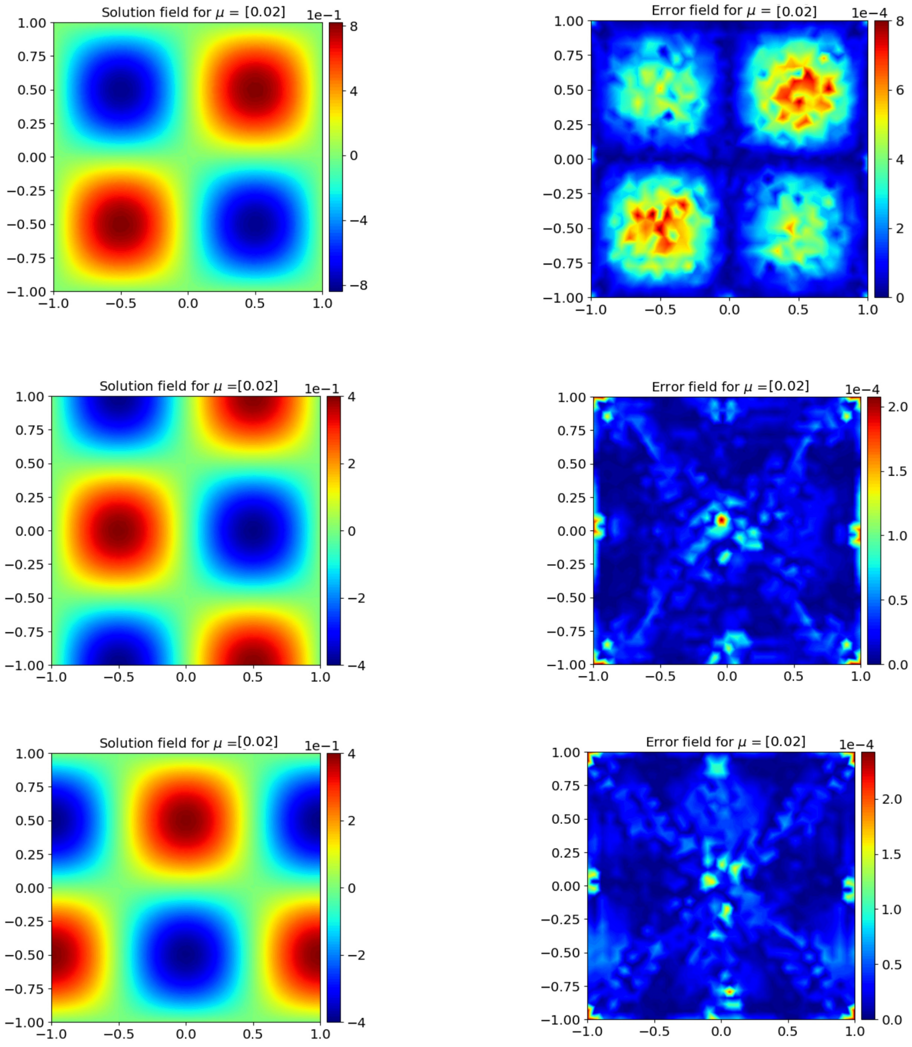


Figure 5.1: AVG-DG-GCA solution and error fields for the electric field (top) and x and y components of the magnetic field, at the instant of time $t = 0.02$, on a mesh with 1220 nodes.

notice that applying the DG-GCA method with the same grid, which has 2310 elements, especially with P_2 elements, allows the loss to have a steepest decrease. Confronting the error field in Figures 5.1 and 5.6 for the electric field, we can see that we achieve a better and uniform accuracy with the DG-GCA method.

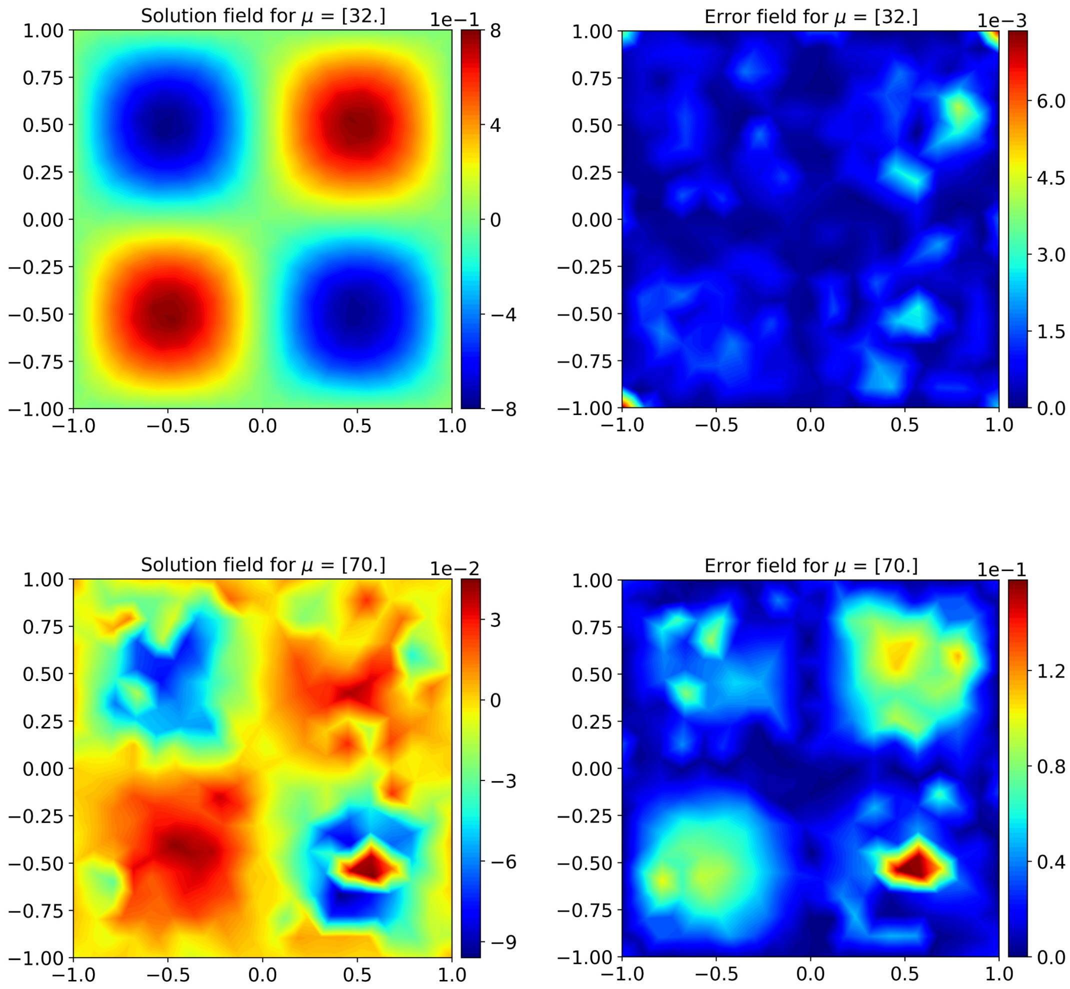


Figure 5.2: AVG-DG-GCA solution fields and error field for the electric field obtained training over a time of three periods.

	E^z	H^x	H^y
max relative error	1.75e-02	1.45e-03	1.25e-03
mean relative error	1.68e-03	3.97e-04	2.76e-04
min relative error	1.81e-04	5.24e-05	6.93e-05

Table 5.3: Relative errors obtained training over half a period with the AVG-DG-GCA method.

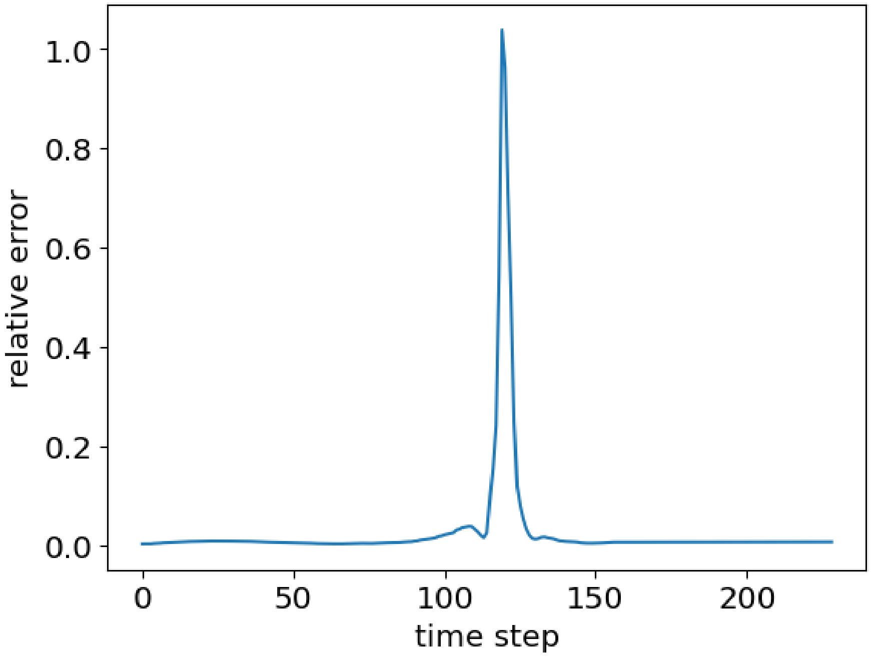


Figure 5.3: Relative error over one period for the electric field component obtained with the AVG-DG-GCA method.

	AVG	P_1	P_2
training time (min)	3.32	7.50	8.41
mean error	0.0653	0.0642	0.0505

Table 5.4: Training time and accuracy for the original and the new definition of the graph structure for a mesh with 317 nodes and 568 elements.

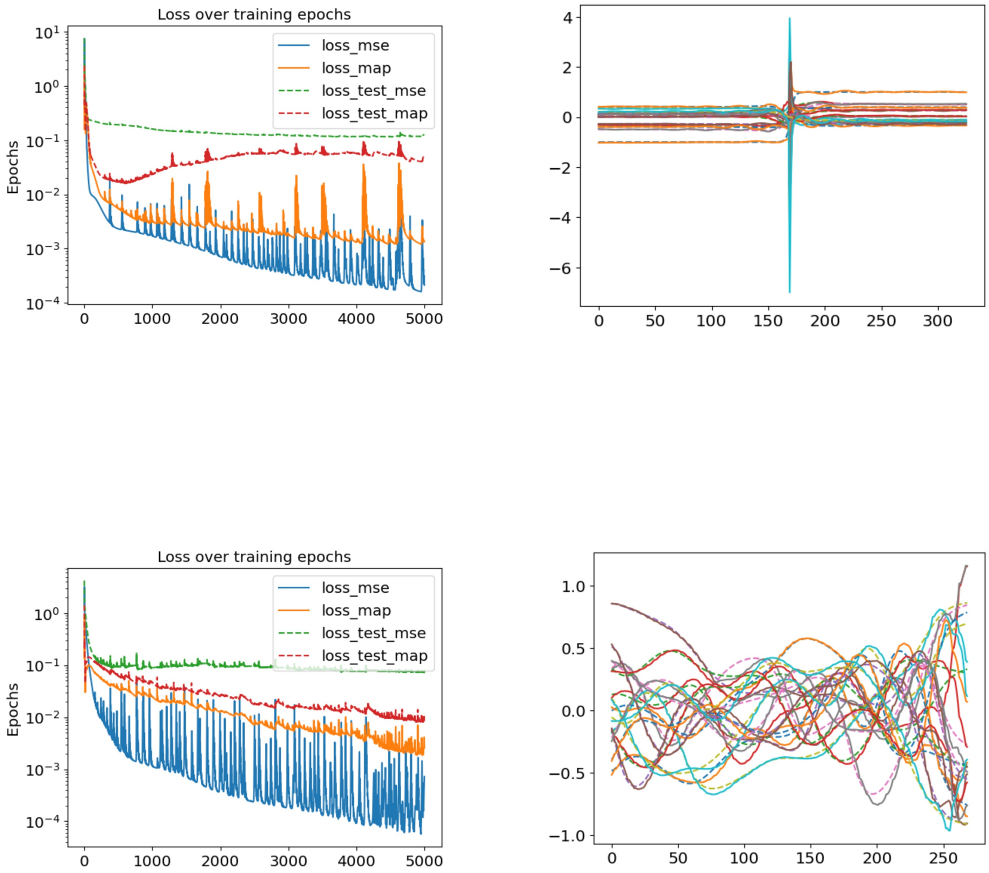


Figure 5.4: Top row: Loss and latent space for the training of the AVG-DG-GCA method with standard scaling over one period. Bottom row: Without scaling, over half a period, the training exhibits more stable loss convergence (left) and more information captured in the latent space (right).

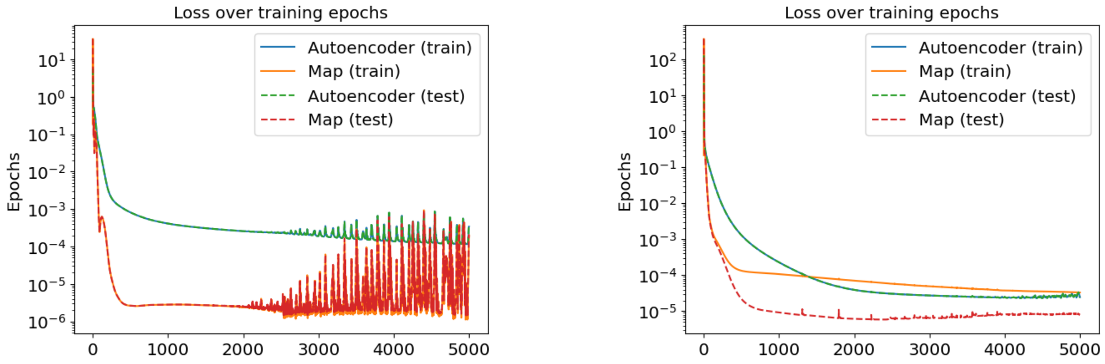


Figure 5.5: Loss evolution for P_1 and P_2 elements (2310 graph nodes), with MinMax scaling function (sample scaling) for the DG-GCA method.

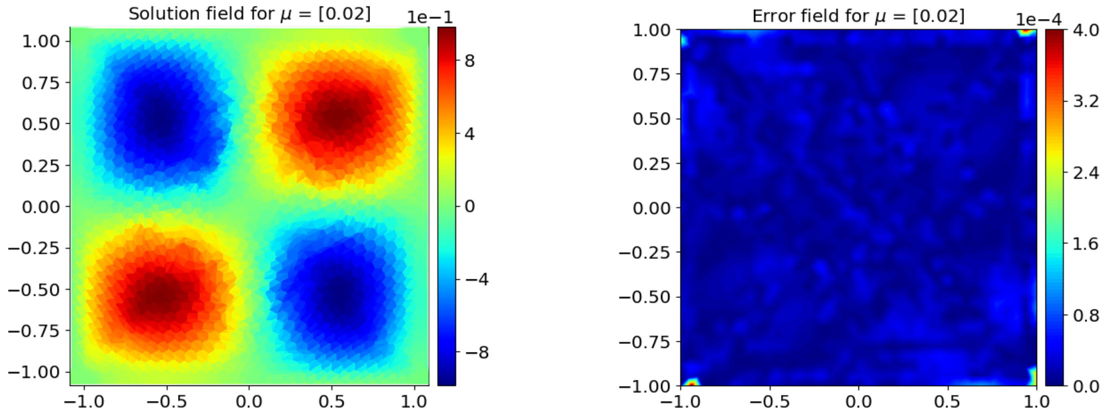


Figure 5.6: Solution field and error field obtained with the DC-GCA method with P_1 elements for the component E_Z at time $t = 0.02$.

5.2 Variation in the permittivity

We will now consider a slight modification of the previous physical setting by introducing a thin slab of a certain conducting material. This leads to the second problem presented in Chapter 2, the perfect electric conductor cavity with two material interfaces.

As a first attempt to build a dataset, we selected 10 values for the permittivity parameter $\varepsilon_1 \in [1.25\,,\, 3.50]$ (see Appendix A) and sampled 300 time steps, thus having 3000 snapshots.

In Figure 5.7, we see that, due to the presence of a higher number of modes, for the AVG-DG-GCA method it is harder to capture the shape of the solutions. This low accuracy might be due to the presence of some critical snapshots with values of the permittivity $\varepsilon_1 = \{1.25\,,\, 1.75\,,\, 2.75,\, 3.0\}$, as displayed in Figures 5.8 and 5.9. For these values of the parameter ε_1, the numerical solutions exhibit a qualitative behavior that is different from the others, which is related to the nonlinearity of the relation (2.11). Indeed the dichotomy method does not guarantee the selection of the most appropriate value for ω_y.

To avoid feeding the relative error with the contributions of those snapshots we tried to train the architecture on the parameter space $\mathcal{P} = [1.75{,}1{,}3.5] \times [0{,}1]$, notwithstanding the resulting error fields are not improving. An explanation of this could be that during the evolution of the electromagnetic wave in $t = [0{,}1]$ the zero solution is encountered three times and, more importantly, that in the dielectric slab, the number of minima and maxima vary a lot depending on the permittivity value (see Figure 5.7). All these features of the physical problem impose to better investigate the hyperparameters of the architecture and the dataset construction part.

In Figure 5.9 we can compare the results obtained with the three sampling of the parameter space, i.e. with equispaced nodes sampling over $\varepsilon_1 \in [1.25,\, 3.5]$ and with the sampling based on the snapshots selection. The mean relative error is still high, but we can achieve some good results for $\varepsilon_1 \in [2.75,\, 3.5]$ as shown in Figure 5.10, where the magnitude of the error is decreased by one order. The small improvement in the learning process is also visible confronting the behaviors of the losses in Figure 5.11.

Moreover, as said before, a source of error is due to periodicity of the solution. Thus, we tried to train the architecture over $t = [0,\, 0.2]$, corresponding to half a period, augmenting the sampled values in the parameter space for time. Figure 5.12 exhibits that, as in the first test case, the relative error increases significantly when approaching the zero solution at $t = 0.2$ and close to the initial condition. In between these values of time, the architecture seems quite able to recover the expected solution, as shown in Figure 5.13. Though improving, the obtained results are still not satisfactory, and a better way to choose the parameter space sampling should be found before proceeding with the DG-GCA method.

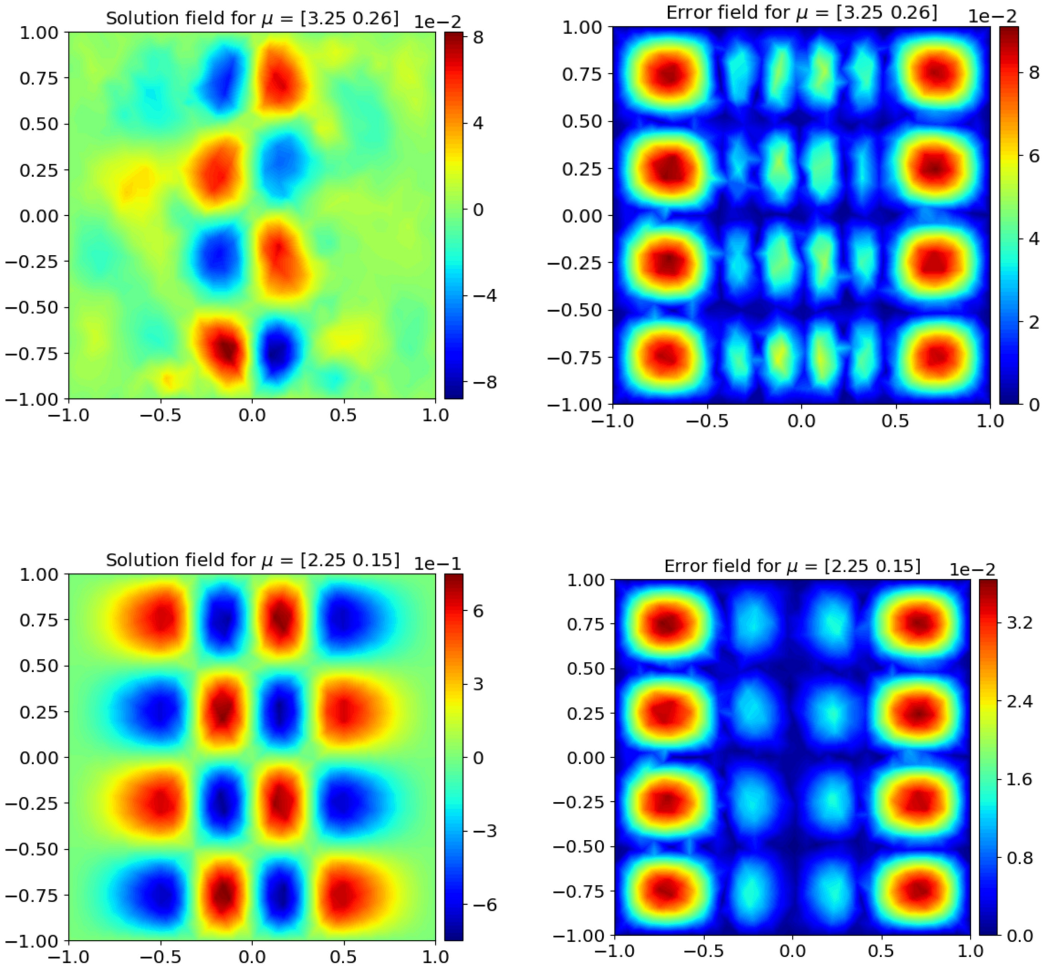


Figure 5.7: AVG-DG-GCA solution and error fields for the E^z component of the perfect electric conductor cavity with two material interfaces with $\varepsilon_1 = \{3.25,\, 2.25\}$

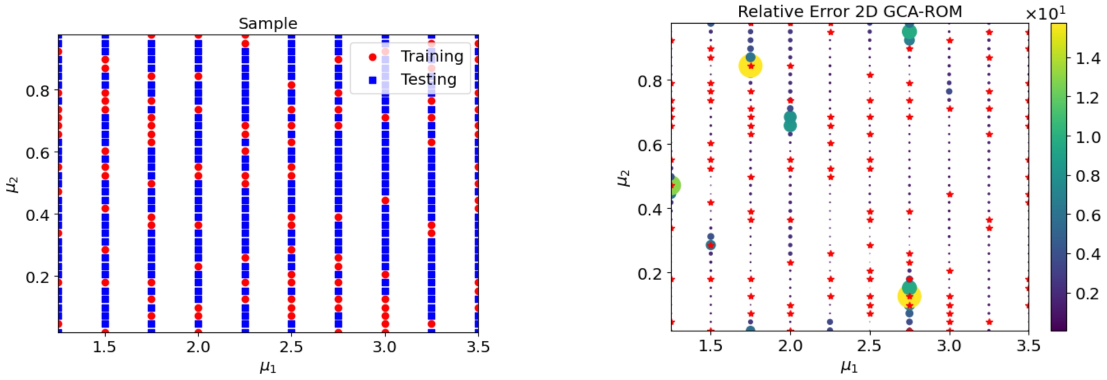


Figure 5.8: Dataset partitioning in training and testing set (left) and relative error (right).

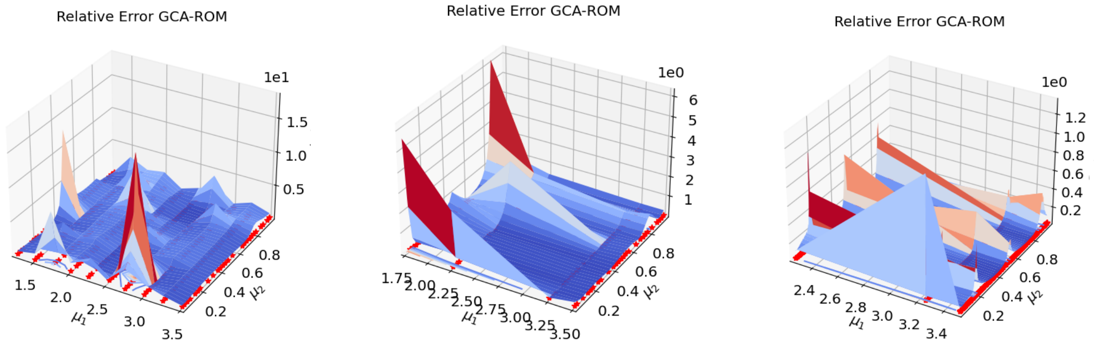


Figure 5.9: Relative error for the electric field over the parameter space $\mathcal{P} = [1.25,3.5] \times [0,1]$ (left), $\mathcal{P} = [1.75,3.5] \times [0,1]$ without critical values for ε_1(center) and $\mathcal{P} = [2.25,3.5] \times [0,1]$ (right).

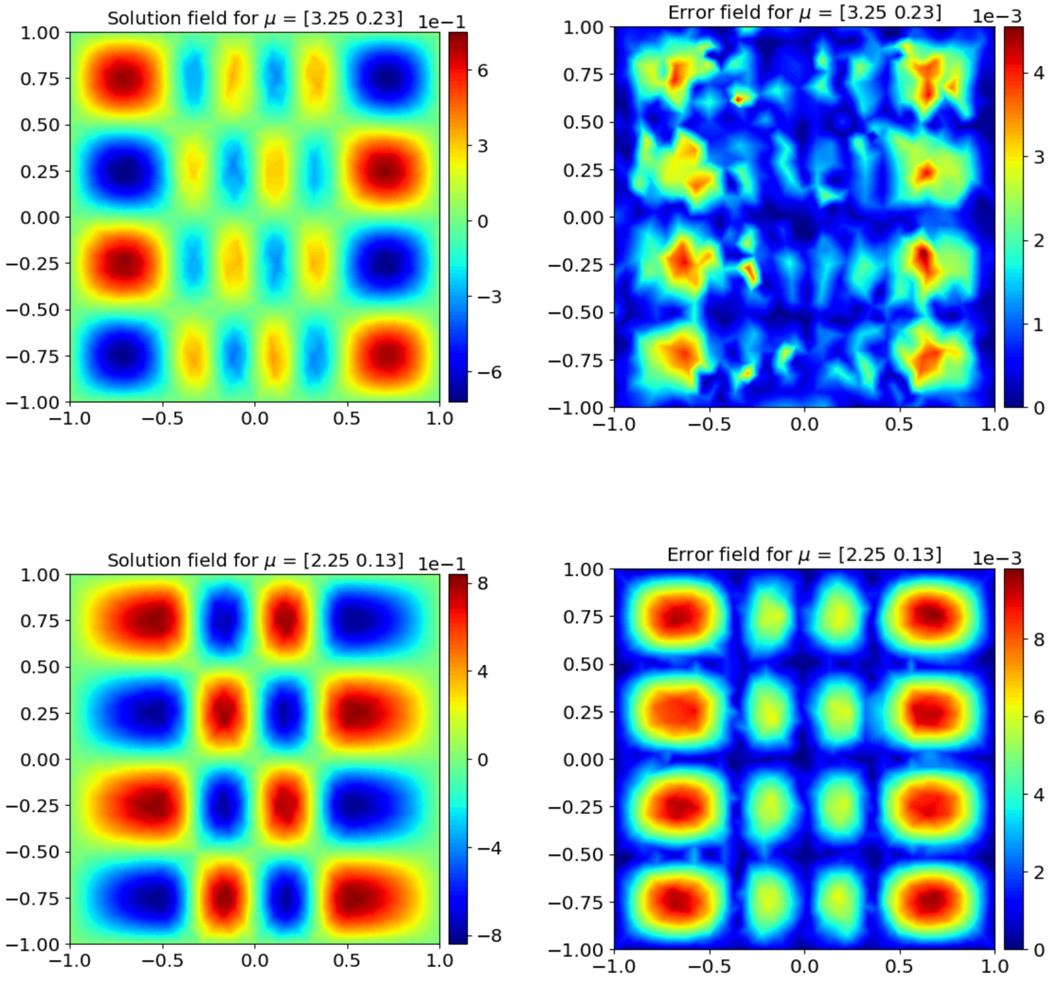


Figure 5.10: AVG-DG-GCA solution and error fields for the E_z component of the perfect electric conductor cavity with two material interfaces with $\varepsilon_1 = \{3.25,2.25\}$.

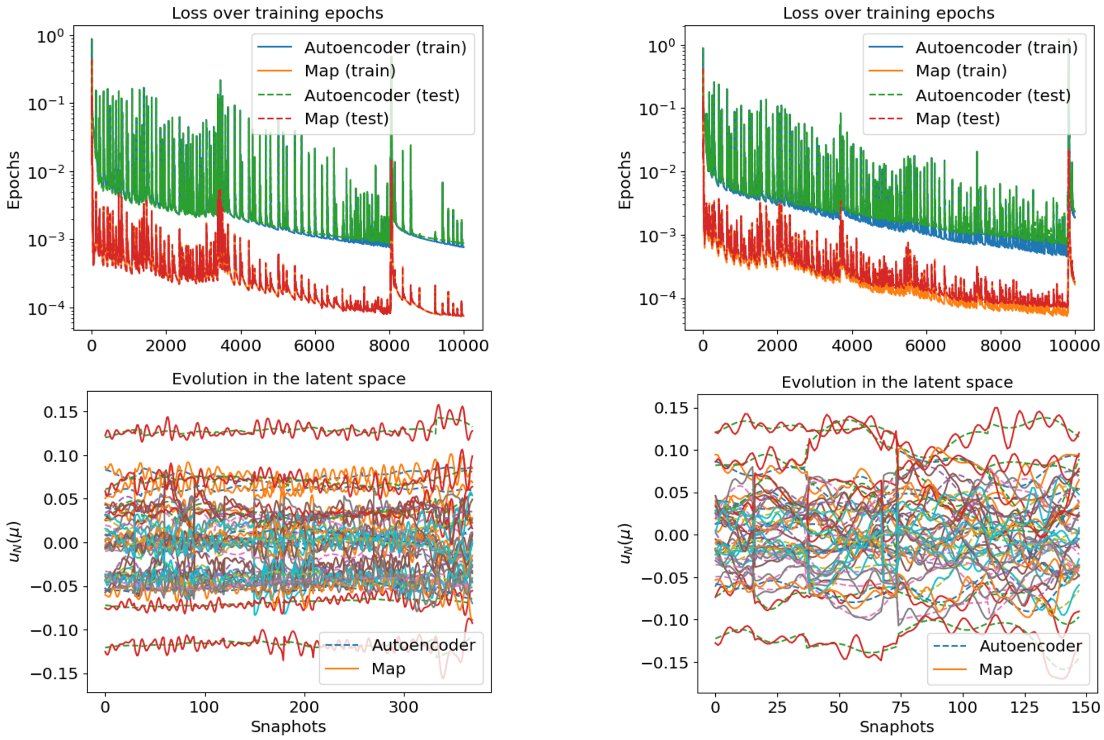


Figure 5.11: Loss and latent space for the training of the AVG-DG-GCA method over all the parameter space $\mathcal{P} = [1.25{,}3.5] \times [0{,}1]$ (left) and $\mathcal{P} = [1.75{,}3.5] \times [0{,}1]$ without critical values for ε_1(right).

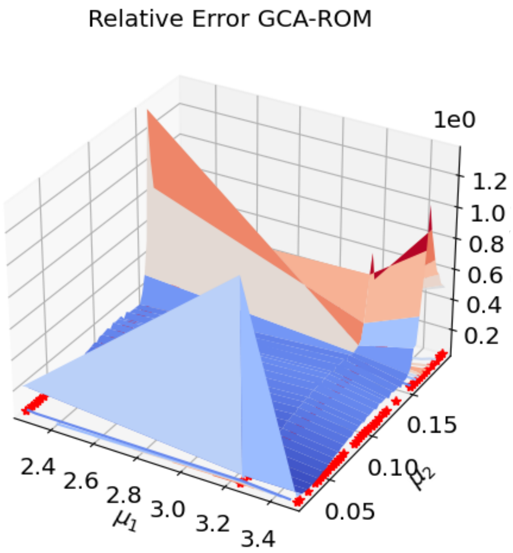


Figure 5.12: Relative error for the electric field over the parameter space $\mathcal{P} = [1.75,3.5] \times [0,1]$ without critical values and over half a period.

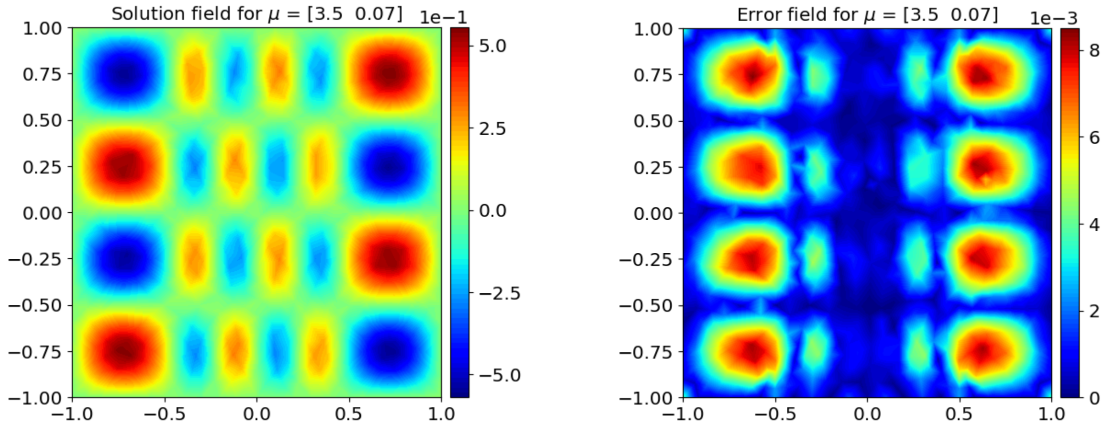


Figure 5.13: AVG-DG-GCA solution and error fields for the E^z component of the perfect electric conductor cavity with two material interfaces with $\varepsilon_1 = 3.5$ (training over half a period).

Conclusions

In this Master's thesis, we investigated the application of the GCA architecture for simulating Maxwell's equations using a DG method. Two simple 2D test cases were used: a vacuum-filled perfect electric conductor cavity and a cavity with the introduction of a dielectric slab. These cases were designed to introduce the DG scheme and evaluate the performance of the GCA architecture.

The AVG-DG-GCA method was able to produce good results, effectively capturing the behavior of the electromagnetic field. However, the DG-GCA method outperformed AVG-DG-GCA, providing more uniform accuracy across the field, albeit with longer training times. This method, particularly when using higher-order elements (e.g., P_2), led to a steeper loss curve compared to P_1 elements. Although considering all degrees of freedom increased the dataset size and extended the training time for DG-GCA, the model achieved better precision, as reflected in the error fields.

When introducing the permittivity parameter, solving the problem using the GCA method proved to be more challenging than expected. The low accuracy in capturing solutions for critical values of the permittivity was linked to the nonlinear behavior of electromagnetic waves within the cavity. Adjusting the parameter space and hyperparameters did not sufficiently resolve these issues.

The model's accuracy was highly sensitive to the structure of the dataset, particularly with respect to time, mesh refinement, and physical parameters like permittivity. Proper scaling and careful dataset construction were crucial for capturing accurate physical behavior in numerical simulations, as demonstrated by the improvements observed when scaling and hyperparameters were optimized.

Training a single architecture to capture the behavior of all components of the electromagnetic field would be prohibitively expensive due to the requirement for a very large dataset. However, there are potential improvements to the methodology. For instance, leveraging the properties of the DG scheme could offer a solution. By focusing on the values of the solution fields at the barycenter—derived from the degrees of freedom and basis functions—it is possible to construct a more compact dataset. This approach could reduce the dataset size without compromising accuracy, as it avoids the loss of information that might occur with averaging or omitting certain degrees of freedom.

Appendix A

Dichotomy method

The dichotomy method, also known as the bisection method, is a numerical technique used to find the roots of a continuous function. A root of a function $f(x)$ is a value x such that $f(x) = 0$. The method is particularly useful when you have a continuous function and you want to find the value of x that makes the function equal to zero within a certain interval. In the present work, we implemented the code reported here to evaluate the solution of 2.11 in Chapter 2. The obtained results are presented in Tab A.1, with a chosen tolerance $\eta = 1e^{-14}$.

```
function omega = dichotomy_solver(a, b, epsilon1, eta)
    % Function to solve
    f = @(omega) sqrt(epsilon2 * omega^2 - 4 * pi^2) * tan(sqrt(omega^2 - 4 * pi^2)/2) + sqrt(omega^2 - 4 * pi^2) * tan(sqrt(epsilon2 * omega^2 - 4 * pi^2)/2);

    % While the interval size is greater than eta
    while abs(b - a) > eta
        c = (a + b) / 2;
        fa = f(a);
        fb = f(b);
        fc = f(c);

        if fa * fc < 0
            b = c;
        elseif fb * fc < 0
            a = c;
        else
            omega = c;
            return;
        end
    end

    % solution is the final interval
    omega = (a + b) / 2;
end
```

ε_1	ω	L_2 error
1.25	8.451365660882516	0.0596
1.50	8.150576639168921	0.1527
1.75	9.657458497408417	0.2252
2.00	9.325449970024170	0.1550
2.25	9.077161758851740	0.0795
2.50	7.163933478590142	0.5860
2.75	6.880812768081277	0.6708
3.00	8.590706586245833	0.1757
3.25	8.459595371133265	0.3309
3.50	8.328310981355269	0.5238

Table A.1: Values of ω obtained with the bisection method, according to the variation of the permittivity ε_1

.

Bibliography

[1] Jérémy Grebot, Rémi Helleboid, Gabriel Mugny, Isobel Nicholson, Louis-Henri Mouron, Stéphane Lanteri, and Denis Rideau. «Bayesian Optimization of Light Grating for High Performance Single-Photon Avalanche Diodes». In: Sept. 2023, pp. 365–368. DOI: `10.23919/SISPAD57422.2023.10319552` (cit. on p. 1).

[2] Jonathan Viquerat. «Simulation of electromagnetic waves propagation in nano-optics with a high-order discontinuous Galerkin time-domain method». Theses. Université Nice Sophia Antipolis, Dec. 2015. URL: `https://hal.science/tel-01272010` (cit. on pp. 1, 17).

[3] Emmanuel Agullo, Luc Giraud, Alexis Gobe, Matthieu Kuhn, Stéphane Lanteri, and Ludovic Moya. «High order HDG method and domain decomposition solvers for frequency-domain electromagnetics». In: *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields* 33 (Oct. 2019). DOI: `10.1002/jnm.2678` (cit. on p. 1).

[4] Yanlai Chen, Jan Hesthaven, and Yvon Maday. «A Seamless Reduced Basis Element Method for 2D Maxwell's Problem: An Introduction». In: vol. 76. Oct. 2010, pp. 141–152. ISBN: 978-3-642-15336-5. DOI: `10.1007/978-3-642-15337-2_11` (cit. on p. 1).

[5] Liang Li, Kun Li, Ting-Zhu Huang, and Stéphane Lanteri. «Non-intrusive reduced-order modeling of parameterized electromagnetic scattering problems using cubic spline interpolation». In: Dec. 2020, pp. 1–3. DOI: `10.1109/NEMO49486.2020.9343422` (cit. on pp. 1, 6, 24).

[6] Federico Pichi, Beatriz Moya, and Jan Hesthaven. «A graph convolutional autoencoder approach to model order reduction for parametrized PDEs». In: (May 2023) (cit. on pp. 2, 7, 22, 25, 26, 29, 30).

[7] William H. Reed and Thomas R. Hill. *Triangular mesh method for the neutron transport equation.* Tech. rep. Los Alamos National Laboratory, 1973 (cit. on p. 4).

[8] Mohamed Remaki and Lahcen Fezoui. *Une méthode de Galerkin discontinu pour la résolution des équa tions de Maxwell en milieu hétérogene.* Tech. rep. Inria Sophia Antipolis, 1998 (cit. on p. 4).

[9] Jan Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations.* Jan. 2016. ISBN: 978-3-319-22470-1. DOI: 10.1007/978-3-319-22470-1 (cit. on pp. 5, 24).

[10] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction.* 1st. La Matematica per Il 3+2, 92. Cham: Springer International Publishing, 2016 (cit. on p. 5).

[11] Maxime Barrault, Yvon Maday, N. C. Nguyen, and Anthony T. Patera. «An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations». In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672. DOI: 10.1016/j.crma.2004.08.006 (cit. on p. 6).

[12] Kuangdai Lee and Kevin T. Carlberg. «Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders». In: *Journal of Computational Physics* 404 (2020), p. 108973 (cit. on pp. 6, 7, 24, 25).

[13] Stefania Fresca, Luca Dede, and Andrea Manzoni. «A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs». In: *Journal of Scientific Computing* 87.2 (2021), pp. 1–36 (cit. on pp. 6, 7, 24, 25).

[14] Ricardo Vinuesa and Steven L. Brunton. «Enhancing computational fluid dynamics with machine learning». In: *Nature Computational Science* 2.6 (2022), pp. 358–366 (cit. on p. 6).

[15] Jan Hesthaven and Stefano Ubbiali. «Non-intrusive reduced order modeling of nonlinear problems using neural networks». In: *Journal of Computational Physics* 363 (Feb. 2018). DOI: 10.1016/j.jcp.2018.02.037 (cit. on pp. 6, 24).

[16] Stefania Fresca and Andrea Manzoni. «POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition». In: *Computer Methods in Applied Mechanics and Engineering* 388 (Jan. 2022), p. 114181. DOI: 10.1016/j.cma.2021.114181 (cit. on pp. 6, 7, 24).

[17] Francesco Romor, Giovanni Stabile, and Gianluigi Rozza. «Non-linear Manifold Reduced-Order Models with Convolutional Autoencoders and Reduced Over-Collocation Method». In: *Journal of Scientific Computing* 94 (Feb. 2023). DOI: 10.1007/s10915-023-02128-2 (cit. on pp. 7, 24).

[18] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. «Geometric deep learning: grids, groups, graphs, geodesics, and gauges». In: *arXiv preprint arXiv:2104.13478* (2021) (cit. on p. 7).

[19] P. W. Battaglia et al. «Relational inductive biases, deep learning, and graph networks». In: *arXiv preprint arXiv:1806.01261* (2018) (cit. on p. 7).

[20] Zonghan Wu, Shirui Pan, Feng Chen, Guodong Long, Cheng Zhang, and Shengyao Philip. «A comprehensive survey on graph neural networks». In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2020), pp. 4–24 (cit. on p. 7).

[21] Jure Zhou, Guohao Cui, Sheng Hu, Zhen Zhang, Chao Yang, Zhen Liu, Liang Wang, Cheng Li, and Ming Sun. «Graph neural networks: A review of methods and applications». In: *AI Open* 1 (2020), pp. 57–81 (cit. on p. 7).

[22] William L. Hamilton. «Graph representation learning». In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14.3 (2020), pp. 1–159 (cit. on p. 7).

[23] Hassan Fahs. «High-order discontinuous Galerkin methods for solving the time-domain Maxwell equations on non-conforming simplicial meshes». Theses. Université Nice Sophia Antipolis, Dec. 2008. URL: `https://theses.hal.science/tel-00359874` (cit. on p. 13).

[24] Shan Zhao and Guo-Wei Wei. «High-order FDTD methods via derivative matching for Maxwell's equations with material interfaces». In: *Journal of Computational Physics* 200 (Oct. 2004), pp. 60–103. DOI: `10.1016/j.jcp.2004.03.008` (cit. on pp. 13, 14).

[25] Jan Hesthaven and Tim Warburton. «Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications». In: vol. 54. Jan. 2007 (cit. on pp. 15, 17–19).

[26] Christopher A. Kennedy, Mark H. Carpenter, and R.Michael Lewis. «Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations». In: *Applied Numerical Mathematics* 35.3 (2000), pp. 177–219. ISSN: 0168-9274. DOI: `https://doi.org/10.1016/S0168-9274(99)00141-5`. URL: `https://www.sciencedirect.com/science/article/pii/S0168927499001415` (cit. on p. 18).

[27] Tim Warburton. *nodal-dg*. `https://github.com/tcew/nodal-dg` (cit. on p. 19).

[28] Peter Benner, Stefano Grivet Talocia, Alfio Quarteroni, Gianluigi Rozza, Wilhelm Schilders, and Luís Miguel Silveira. *Model Order Reduction.* 1-3 vols. De Gruyter, 2020 (cit. on p. 22).

[29] Peter Benner, Albert Cohen, Mario Ohlberger, and Karen Willcox. *Model Reduction and Approximation: Theory and Algorithms*. Computational Science and Engineering Series. SIAM, Society for Industrial and Applied Mathematics, 2017 (cit. on p. 22).

[30] Trinh Bui-Thanh, M. Damodaran, and Karen Willcox. «Proper Orthogonal Decomposition Extensions for Parametric Applications in Compressible Aerodynamics». In: *21st AIAA Applied Aerodynamics Conference, Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics, 2003 (cit. on p. 24).

[31] Niccolò Demo, Marco Tezzele, and Gianluigi Rozza. «A non-intrusive approach for the reconstruction of POD modal coefficients through active subspaces». In: *Comptes Rendus Mécanique* 347.11 (2019), pp. 873–881 (cit. on p. 24).

[32] Christian Greif and Klaus Urban. «Decay of the Kolmogorov N-width for wave problems». In: *Applied Mathematics Letters* 96 (2019), pp. 216–222 (cit. on p. 24).

[33] A. Muixí, S. Zlotnik, and A. García-González. «Nonlinear dimensionality reduction for parametric problems: A kernel proper orthogonal decomposition». In: *International Journal for Numerical Methods in Engineering* 122.24 (2021), pp. 7306–7327 (cit. on p. 24).

[34] Jens Reiss, Peter Schulze, Jörg Sesterhenn, and Volker Mehrmann. «The Shifted Proper Orthogonal Decomposition: A Mode Decomposition for Multiple Transport Phenomena». In: *SIAM Journal on Scientific Computing* 40.3 (2018), A1322–A1344 (cit. on p. 24).

[35] Lorenzo Cicci, Simone Fresca, and Andrea Manzoni. «Deep-HyROMnet: A Deep Learning-Based Operator Approximation for Hyper-Reduction of Nonlinear Parametrized PDEs». In: *Journal of Scientific Computing* 93.2 (2022), p. 57 (cit. on p. 24).

[36] Quoc Hernandez, Alba Badias, David Gonzalez, Francisco Chinesta, and Elias Cueto. «Deep learning of thermodynamics-aware reduced-order models from data». In: *Computer Methods in Applied Mechanics and Engineering* 379 (2021), p. 113763 (cit. on p. 24).

[37] B. Moya, A. Badias, D. Gonzalez, F. Chinesta, and E. Cueto. «Physics perception in sloshing scenes with guaranteed thermodynamic consistency». In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022) (cit. on p. 24).

[38] M. Guo and J. S. Hesthaven. «Reduced order modeling for nonlinear structural analysis using Gaussian process regression». In: *Computer Methods in Applied Mechanics and Engineering* 341 (2018), pp. 807–826 (cit. on p. 24).

[39] B. Peherstorfer and K. Willcox. «Data-driven operator inference for nonintrusive projection-based model reduction». In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 196–215 (cit. on p. 24).

[40] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. «Neural operator: Learning maps between function spaces». In: *arXiv preprint arXiv:2108.08481* (2021) (cit. on p. 24).

[41] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. «Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators». In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229 (cit. on p. 24).

[42] Niccolò Demo, Marco Tezzele, and Gianluigi Rozza. «A DeepONet Multi-Fidelity Approach for Residual Learning in Reduced Order Modeling». In: *arXiv preprint arXiv:2302.12682* (2023) (cit. on p. 24).

[43] Michele Milano and Petros Koumoutsakos. «Neural Network Modeling for Near Wall Turbulent Flow». In: *Journal of Computational Physics* 182.1 (2002), pp. 1–26 (cit. on p. 25).

[44] Raktim Maulik, Brandon Lusch, and Prabhat Balaprakash. «Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders». In: *Physics of Fluids* 33.3 (2021), p. 037106 (cit. on p. 25).

[45] Peng Wu, Shuang Gong, Kai Pan, Feng Qiu, Wei Feng, and Christopher Pain. «Reduced order model using convolutional auto-encoder with self-attention». In: *Physics of Fluids* 33.7 (2021), p. 077107 (cit. on p. 25).

[46] Franco Scarselli, Marco Gori, Ah Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. «The Graph Neural Network Model». In: *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council* 20 (Jan. 2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605 (cit. on p. 26).

[47] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. «Geometric deep learning on graphs and manifolds using mixture model CNNs». In: (Nov. 2016) (cit. on pp. 27, 28).

[48] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. «Pointnet++: Deep hierarchical feature learning on point sets in a metric space.» In: *Advances in neural information processing systems,30,* (2017) (cit. on p. 27).