APPROXIMATING SOLUTIONS OF *PARTIAL DIFFERENTIAL EQUATIONS* WITH PSUEDOSPECTRAL METHODS AND *SUPPORT VECTOR MACHINE*

A RESEARCH PAPER

Submitted as partial fulfillment of the requirements for
sarjana komputer degree

By
Ahmad Izzuddin
1908919

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS PENDIDIAKN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PENDIDIKAN INDONESIA
2024

ABSTRAK

JUDUL ABSTRAK

Oleh
Ahmad Izzuddin
NIM: 1908919

Abstrak merupakan penjelasan singkat dan padat tentang pekerjaan dan hasil penelitian TA, yang dituliskan secara teknis. Abstrak memiliki karakter tegas dan komprehensif, dan hanya dapat dituliskan setelah pekerjaan penelitian telah mencapai tahap tertentu, dan karenanya ada hasil penelitian yang dapat dilaporkan. Abstrak ditulis menjelang akhir penyelesaian penulisan buku TA.

Secara umum, abstrak memuat beberapa komponen penting, yaitu: konteks atau cakupan pekerjaan penelitian, tujuan penelitian, metodologi yang digunakan selama penelitian, hasil-hasil penting yang dapat ditambahkan dengan implikasinya, dan simpulan dari penelitian. Dengan demikian, suatu abstrak tidak dapat dituliskan apabila penelitian belum mencapai hasil tertentu, apalagi kalau penelitiannya pun belum dilakukan.

Panjang abstrak sebaiknya dicukupkan dalam satu halaman, termasuk kata kunci. Tiga kata kunci dipandang cukup, yang masing-masingnya memuat paduan kata utama, yang dapat merepresentasikan isi Abstrak. Halaman Abstrak tidak memuat informasi judul dan penulis, sehingga tidak secara langsung dapat digunakan sebagai lembaran Abstrak Sidang TA yang disediakan untuk hadirin, yang memerlukan tambahan (sekurangnya) dua informasi tersebut.

Kata kunci: Konsep Abstrak, Komponen Abstrak, Kata Kunci.

ABSTRACT

TITLE

by
Ahmad Izzuddin
NIM: 1908919

In general, Abstract is a translation of Abstrak. However, appropriate paraphrase may need some words or sentences whose meanings are close enough to those written in Abstrak.

Key words: Abstract Concepts, Abstract Components, Key Words.

PENGESAHAN

APPROXIMATING SOLUTIONS OF *PARTIAL DIFFERENTIAL EQUATIONS* WITH PSUEDOSPECTRAL METHODS AND *SUPPORT VECTOR MACHINE*

Oleh

Ahmad Izzuddin

NIM 1908919

Program studi Sarjana Fisika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Institut Teknologi Bandung

Menyetujui

Bandung, August 22, 2024

Dosen Pembimbing,

Prof. Dr. Lala Septem Riza, M.T.

NIP. 197809262008121001

Tim Penguji:

1. Nama Penguji 1
2. Nama Penguji 2

PEDOMAN PENGGUNAAN
BUKU TUGAS AKHIR

Tugas Akhir ini dipersembahkan untuk
Tuhan, Bangsa, dan Almamater

KATA PENGANTAR

Kata pengantar berperan sebagai gerbang masuk bagi pembaca dan mendapat sajian ringkas tentang hal-hal terkait paparan pada buku Tugas Akhir (TA). Sajian ini sejatinya merupakan pengenalan umum bagi pembaca tentang isi tulisan. Hal ini berbeda dengan abstrak yang mendeskripsikan pekerjaan dan hasil penelitian secara lebih teknis.

Kata pengantar merupakan wadah penulis untuk mengenalkan dan mempromosikan pekerjaan dan hasil penelitian dengan bahasa yang sederhana, sehingga pembaca tertarik untuk menelusuri lebih jauh dengan mencermati seluruh paparan pada buku TA. Ini salah satu tujuan kata pengantar. Contoh paragraf yang mengantar pembaca pada isi Buku TA: *Template* LaTeX diberikan berikut ini.

Menuliskan pekerjaan dan hasil penelitian TA dalam suatu laporan buku TA memerlukan panduan standar. Panduan ini dibuat dalam beberapa dokumen, yang salah satunya adalah Buku TA: *Template* LaTeX. Suatu template adalah cetakan yang siap dituang oleh curahan buah pikiran yang keluar dari pengalaman dalam melakukan pekerjaan penelitian dan hasil-hasilnya. Mencermati cetakan yang memberikan sejumlah contoh dapat memperlancar penulisan laporan tersebut menjadi suatu produk, yaitu buku TA.

Tujuan lain dari Kata Pengantar adalah memberi tempat untuk menyampaikan rasa syukur dan terima kasih kepada banyak pihak, misalnya keluarga, staf akademik, staf tenaga kependidikan, teman, individu atau komunitas pemberi dukungan dan inspirasi, dan institusi pendukung pendanaan seperti pemberi beasiswa atau dana penelitian, atau pendukung akses fasilitas.

Pengorbanan, kegigihan, dedikasi, dan penuh tanggung jawab dari para pahlawan pekerja medis dalam perawatan pasien terpapar Covid-19 telah memberi inspirasi melalui nilai-nilai kejuangan tanpa pamrih. Inspirasi inilah yang membangkitkan spirit pamungkas pada penyelesaian Buku TA: Template LaTeX *ini. Suatu inspirasi selalu bekerja dan mengena secara tidak langsung. Banyak berterima kasih atas inspirasi yang memantik spirit ini.*

Tidak ada sub bab/bagian pada Kata Pengantar, namun daftar rincian diperkenankan. Pada bagian indentitas akhir, seperti berikut ini, dituliskan nama mahasiswa dan NIM, bukan *penulis*, dan tidak perlu ditandatangani. Berikut adalah contoh penulisan rincian yang berisi ucapan terima kasih:

- Prof. Dr. Lala Septem Riza, M.T. dan Dr. Muhammad Nursalman, M.T. selaku dosen pembimbing tugas akhir.
- Jyesta, sebagai teman bimbingan yang selalu bersedia untuk diajak berdiskusi selama penelitian.
- Rekan-rekan Spectranova, ...

Bandung, August 22, 2024

Ahmad Izzuddin
1908919

DAFTAR ISI

DAFTAR NOTASI

Notasi	Arti
$F_{\mu\nu}$	Tensor Elektromagnetik
$R^{\mu}_{\ \alpha\nu\beta}$	Tensor Riemann
$\Gamma^{\rho}_{\mu\nu}$	Simbol Christoffel
$g_{\mu\nu}$	Tensor Metrik
A_{μ}	Medan Gauge
$R_{\mu\nu}$	Tensor Ricci
$\mathscr{L}$	Densitas Lagrangian
$\hbar$	Konstanta Planck Tereduksi
$\mathbb{R}$	Himpunan Bilangan Real

DAFTAR SINGKATAN

Notasi	**Arti**
FWHM	*Full width half maximum*
rms	*root mean square*
RFS	*Rotary forcespinning*
PVP	Polivinil pirolidon
SI	Satuan Internasional

DAFTAR GAMBAR

DAFTAR TABEL

CHAPTER I

INTRODUCTION

1.1 Background

Partial differential equations (PDE) are a common tool widely used in the modern scientific understanding and many engineering processes. This is because many systems can be described by the way they change and often this can be more intuitive. As an example, think of someone heating up a large frying pan on a gas stove. To describe how the pan heats up when the center is right above the burner, one can say that the center would heat up first followed by its surroundings. Eventually the pan would not heat up any further. Also notice that the edges of the pan would always be cooler than the center. However, once a more complex setup is introduced such as an uneven heat source or more complex materials with different heat rates of transferring heat it becomes much harder to describe how the pan heats up. A more useful way to describe the system is using the heat equation. For a temperature function $u(\mathbf{x},t)$ of spatial coordinates vector $\mathbf{x}$ and time t, the generalized heat equation is defined in equation (1.1).

$$\frac{\partial u}{\partial t} = \nabla \cdot (\alpha \nabla u) \tag{1.1}$$

The divergence operator $\nabla\cdot$ denotes sum of all first spatial derivatives. In three dimensions this is $\nabla\cdot = \frac{\partial\cdot}{\partial x_1} + \frac{\partial\cdot}{\partial x_2} + \frac{\partial\cdot}{\partial x_3}$. And the gradient operator ∇ is the vector of all first spatial derivatives which in three dimensions is $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} & \frac{\partial f}{\partial x_3} \end{bmatrix}^\intercal$. This equation says that the rate at which the temperature changes in time $\frac{\partial u}{\partial t}$ is proportional to how different the differences in temperature of a spot in the pan with its surroundings $\nabla^2 u$ multiplied with thermal diffusivity parameter α. One example of an insight this gives is that given a homogeneous material (α is constant) and the stove outputs heat at a constant rate the temperature will no longer change for a particular point once the temperature difference is constant. In other words once the function $u(\mathbf{x},t)$ at some time $t > 0$ approaches a linear function in space, the temperature at all spots will no longer change in time. Mathematically this is because the time derivative is zero if the second spatial derivatives are also zero. Intuitively this is because once the temperature distribution approaches linear, the current spot on the pan is outputting as much heat it is getting. This insight is an example of why PDEs are useful.

Many other fields in physics such as waves, quantum dynamics, fluid dynamics, elastics, and many more also define systems using PDEs. PDEs are also used outside the physical sciences. In Finance, the Black-Scholes-Merton or Black-Scholes equation models the dynamics of the financial market. In ecology, PDEs are used to model population growth which is useful for modelling location dependent carrying capacity. This is used to model species distribution which can help develop better conservation policies. In the social sciences, PDEs such as the cross-diffusion model is used to explore the evolution of the urban environment (Jin et al., 2023). Crowd dynamics is another field where researchers have found it useful to model using PDEs (Hughes, 2000; Mukherjee et al., 2015). These are used to model dynamics such as shock-waves, pathing, and crowd flow as a whole. There are a variety of applications, ranging from bird flocks, pedestrian crowds, to robotic swarms (Gong et al., 2023).

An increasingly critical use of PDEs is in Numerical weather prediction (NWP). The information weather forecasts provide is an integral part of modern life. Many sectors rely on timely and accurate weather forecasts. Individuals, day to day rely on forecasts for a range of different decisions ranging from activities they can do to personal safety from extreme weather. The weather also affects retail as consumer behavior change with whether it is rain or shine (Govind et al., 2020; Moon et al., 2018; J. Tian et al., 2018; X. Tian et al., 2021). Similarly, the types of transportation people use and operational decisions changes with the weather (Lepage & Morency, 2021; Nurmi et al., 2013). Another critical sector is agriculture, where extreme weather events and changes in climate threatens the food supply (Anwar et al., 2013; Cogato et al., 2019; Malhi et al., 2021). As the climate continues to warm and extreme weather events increase in frequency, the mitigation for these events become all the more important (*Climate Change 2022*, 2023). Because of this, forecasting is critical in making accurate decisions for policymakers and warnings of extreme weather events for civilians (Astitha & Nikolopoulos, 2023; Stott et al., 2016). A major challenge is the fact that observations of weather can be sparse in the sense that observations are localized and do not cover entire areas (Galkin et al., 2020; Monmonier, 1999; Wilby & Yu, 2013). This lack of spatially distributed meteorological information hampers decision-making on areas to be prioritized and what to prioritize based on location. For example, weather stations can only observe their immediate surroundings. And they are for the most part stationary. Even more mobile observation platforms like satellites only show the portion of the surface the satellite can see at any one time such as in figure 1.1. While accumulating observations through more satellite passes is possible, this, however, means short-lived features may not be observed. As impoverished regions often are the most data sparse, the effect compounds on the fact that for impoverished regions, scarce resources need to be effectively and efficiently applied. To this end, numerical models are employed to model a more representative view by using known physics such as the compressible Euler equations and statistical models (Kwasniok, 2012; Mengaldo et al., 2019). The next step is in forecasting future weather which in itself is a challenge. The information this could provide is invaluable because it means planning for future weather is possible.

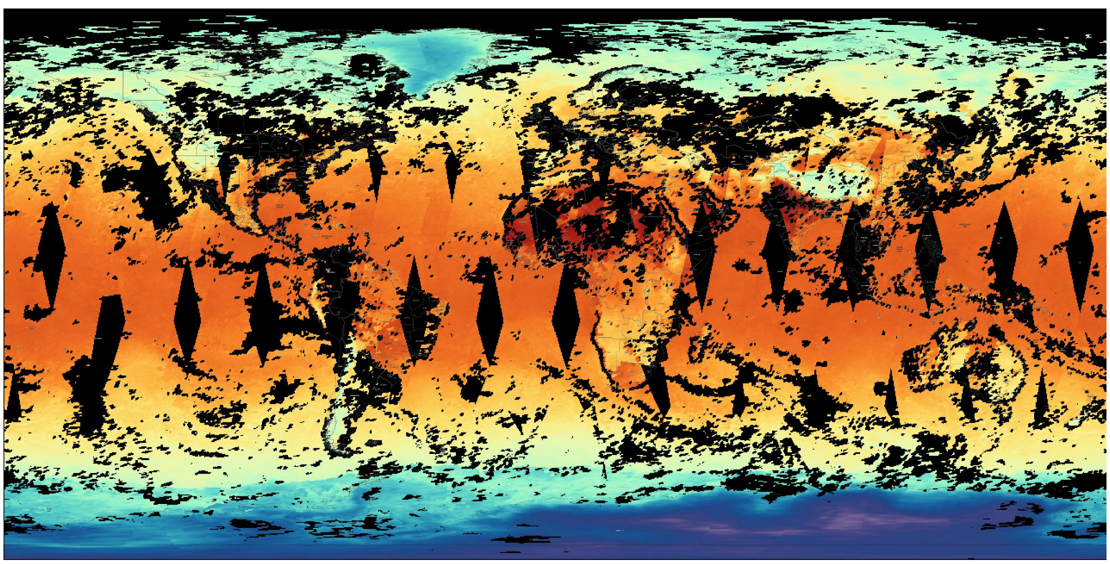


Figure 1.1: Daytime and nighttime surface air temperature from observations by the AIRS instrument onboard the NASA AQUA satellite. This shows observations recorded through a whole day on the 21st of May 2024. Notice that parts of the globe, especially near the equator have not been observed. This image was produced using NASA Worldview (https://go.nasa.gov/46SaYyJ).

Forecasting the weather or evolution of any other system is formulated as an initial value problem where past observations are used to predict an unknown future state of the system. A different formulation can be how heat spreads over time in different types of frying pans from known information of the pan's materials, their heat conduction properties, and that the stove gives off constant heat. In general the scenario is predicting the effect (future atmospheric temperature distribution, etc.) of some given causes (current and past weather, etc.). This scenario is termed the forward problem. The inverse problem on the other hand solves for an unknown parameter using other known parameters and observations of the partial solution such as temperature field at the object's surface. Traditional methods used to solve these problems utilized knowledge of the exact equations to numerically solve PDEs. As an example, the Finite Difference Method (FDM) approximates the solution by substituting the partial derivatives with finite differences and manipulating the equation such that the solution can be computed.

Traditional numerical solvers have enjoyed many decades of development due to their long history. There are many general approaches to solve PDEs and many more very specific approaches. Other than FDM, some widely used approaches include Finite Element Methods (FEM), Finite Volume Methods (FVM), Collocation Methods, and Spectral Methods. FEM and FDM are both mesh based approaches, meaning they rely on discretization of the computational domain <ADD DIAGRAM FOR THIS>. There are several challenges associated with this. First, irregular domains such as bio-inspired materials like bone, spider silk, or aggregate materials like gravel pose a challenge due to the complexity of the domain geometry (Buoni & Petzold, 2007; Gaul et al., 1991; Jia et al., 2024). Also, irregular domains which create large deformations or mesh entanglement cause these methods to become ineffective (J.-S. Chen et al., 2017). While there are strategies to mitigate this, by definition they are an additional layer of difficulty to the process. Second, multiscale applications where the micro and macro scales are both important require fine meshes such that the small scale structures are adequately simulated. This creates meshes with very large number of points that are very resource intensive (Buoni & Petzold, 2007). Third, a single evaluation of traditional mesh-based solvers may not be very costly, however multiple evaluations can add up. This is very apparent in inverse problems where the solver is queried multiple times to solve the forward problem in order to obtain parameter functions. Therefore, in problems where these issues important or resources are limited, mesh-free methods may be preferred. As an example, the spectral method solves PDEs by formulating the solution as a linear combination of global basis functions like the Fourier series. This can be likened to how music combines sound waves of different things to produce the overall sound. The complex formulation of the Fourier series is presented in equation (1.2).

$$s = \sum_k c_k e^{2\pi ikx} \tag{1.2}$$

The example differential equation we wish to solve is the simple derivative in equation (1.3). The aim is to find a function $u*$ from a known function $f*$. To do this we first formulate both into separate Fourier series approximations u and f in equations (1.4) and (1.5) respectively.

$$\frac{\mathrm{d}u(x)}{\mathrm{d}x} = f(x) \tag{1.3}$$

$$u(x) = \sum_k \hat{u}_k e^{2\pi ikx} \tag{1.4}$$

$$f(x) = \sum_k \hat{f}_k e^{2\pi ikx} \tag{1.5}$$

Next we take the derivative of equation (1.4) which result in equation (1.6). Using this we can substitute the terms in equation (1.3) with equations (1.5) and (1.6) giving equation (1.7). Finally, after some algebraic manipulation we obtain equation (1.8). One also needs to choose $\hat{u}_0$, which is the integration constant. Practically this would be done with a finite number of modes and the coefficients $\hat{f}_k$ are obtained using a Discrete Fourier Transform (DFT) algorithm like the Fast Fourier Transform (FFT).

$$\frac{\mathrm{d}u(x)}{\mathrm{d}x} = \sum_k \hat{u}_k \times (2\pi ik)\, e^{2\pi ikx} \tag{1.6}$$

$$\sum_k \hat{u}_k \times (2\pi ik)\, e^{2\pi ikx} = \sum_k \hat{f}_k e^{2\pi ikx} \tag{1.7}$$

$$\hat{u}_k = \hat{f}_k / (2\pi ik) \tag{1.8}$$

The fourth challenge with traditional methods is that they require prior knowledge of analytic forms of PDEs. This is because the solvers use the equations to formulate solutions. This makes traditional numerical approaches unsuited to data dominant problems. These challenges altogether are some of what weather forecasting faces; NWP has the immense task of modeling multiple scales of the Earth's atmosphere while accounting for geographical features of physical systems that are still not fully understood. Many other fields also face these challenges which has motivated research into alternative methods that do not completely rely on prior knowledge, are mesh free, and fast enough when solving forward problems.

Machine Learning for PDEs

With the increasing prevalence of machine learning methods and their use in more and more fields, research into their use for scientific computing has taken off in recent years. While statistical modeling has already been widely used in areas such as physical constants, stellar population studies, risk assessments of events such as earthquakes and coronal mass ejections (Anselmo & Pardini, 2005; Berliner, 2003; Bernardi et al., 2022; Reinhardt et al., 2016; Spanos, 2006; Uzan, 2003), the dominant approach for forward modeling or inverse modeling has remained physics based numerical models. Machine learning has provided an alternative approach to model the solutions of PDEs. In their work, Aarts and Van Der Veer (2001) utilized neural networks to approximate each term of PDEs describing damped and undamped free vibrations and substituting them into the PDEs and associated initial conditions. The network parameters were then optimized to reduce the PDE residual and boundary condition loss using evolutionary algorithms. This approach was taken in order to make machine learning models more transparent which at the time was being pursued because of the high cost of optimizing uncertainties in water management numerical simulators. However, since this method approximates the mapping between coordinates and the values of a function and their derivatives, retraining would be necessary for changes to the function itself. This could become very costly as retraining costs accumulate. A more recent approach that also utilizes soft constrains from PDE residuals is termed physics informed neural network (PINN) (Raissi et al., 2019). The authors propose a framework that leverages advancements in computing, namely automatic differentiation (AD) techniques made readily available by modern machine learning libraries. In general, PINNs use AD to compute each term of the PDE from the output of the model and this is then substituted into the PDE to in order to compute the residuals and loss from boundary conditions. The network residual and boundary loss are then weighted and summed with the data loss. For a neural network $\hat{u}(\mathbf{x},t)$ approximating the real solution $u(\mathbf{x},t)$, the residual loss for the heat equation in equation (1.1) is equation (1.9). There are several advantages of incorporating physics knowledge into the model including regularization of the model outputs to be more consistent with physics, faster convergence, and less to no data required depending on whether the network is trained in a manner that is supervised, self-supervised, or a combination of both. One issue with using AD to compute the residual is that the network input needs to be the independent variable (i.e. coordinates, time, etc.). This once again means that if one wants to compute a different solution, the network needs to be retrained.

$$\mathscr{L}_{PDE} = \left(\frac{\partial \hat{u}(\mathbf{x},t)}{\partial t} - \nabla \cdot (\alpha \nabla \hat{u}(\mathbf{x},t)) \right)^2 \tag{1.9}$$

Learning PDEs with CNNs

Other works utilize convolutional neural networks (CNN) to compute the solution from input functions such as forcing terms or initial conditions. This approach generally means discretizing the functions on a grid and using these as training data. One study by Wang et al. (2020) predicts turbulent flow using spatial and temporal decomposition and a specialized U-Net, an architecture based on CNNs, to predict the velocity field from the decomposition of the previous velocity field. Part of the loss function is a regularization term for zero divergence in the velocity field to enforce incompressible fluid flow. This term was calculated using finite differences since auto differentiation is not applicable in this situation. Finite differences was also utilized in another CNN based fluid flow upscaling model by Gao et al. (2021b) to compute the residual terms of the steady incompressible Navier-Stokes equation. This model also inferred unknown physical parameters such as boundary conditions. However, this approach would mean the model would need to scale as a quadratic in 2D, cubic in 3D, and much steeper in higher dimensions. Outside fluid dynamics, the combination of specialized CNNs and finite differences or another numerical differentiation method have been used for many other PDEs including Poisson's equation for temperature fields (Gao et al., 2021a; Zhao et al., 2023), velocity models from seismic data (Muller et al., 2023), and seismic response of structures (Ni et al., 2022; Zhang et al., 2020). While the use of CNNs mean that discretization is implied, solutions of different initial conditions or parameter functions can be computed by inference and no retraining is required. This property is especially useful for many-query problems such as computing gradients for inverse problems.

Operator Learning

The mapping between discretized functions done by CNNs are related to an alternative approach that starts by viewing PDEs as operators, which are generalized mappings between spaces. One group of familiar operators are functions which maps between spaces of scalar values or vector values. PDEs on the other hand are operators that map between function spaces. A simple example is the derivative. The derivative takes in a function and returns the derivative of said function. In other words it is an operator that maps between the space of all functions to the space of derivatives of those functions. Another way to view operators starts by viewing functions as infinite dimensional vectors. Where elements in the vector are the function's value evaluated at every point in space. The operator can be seen as a vector function mapping between these infinite dimensional vector spaces. Operators are important because a field of research has sprung up around this mathematical concept. Operator learning is the use of machine learning to learn operators using data driven approaches. As an analogy, function regression traditionally has been used to approximate the mapping between input values such as coordinates and output values of functions evaluated at said coordinates. In the case of operator learning, the mapping between function spaces are approximated. The aforementioned approaches using CNNs does this directly using the values of functions at discrete points. There are other approaches like DeepONet that does not require the uniform grid like CNNs (Lu et al., 2021). This architecture instead uses both input functions and coordinates as inputs. The output is the output function evaluated at the coordinates provided. This architecture is based on an extension for deep learning of the universal operator approximation theory for neural networks first proposed almost three decades ago at the time of writing by T. Chen and Chen (1995). The proposed architecture is composed of two subnetworks, where one termed the trunk $\hat{\mathbf{T}}(\mathbf{x},t)$learns the latent mapping for coordinates and the other network termed the branch $\hat{\mathbf{B}}(\mathbf{f})$ learns the latent mapping for the input function f. The two latent mappings are combined through a dot product to obtain the approximated output function value $u(\mathbf{x},t)$. This is formulated in equation (1.10).

$$u(\mathbf{x},t) \approx \hat{G}(\mathbf{f})(\mathbf{x},t) = \hat{\mathbf{B}}(\mathbf{f}) \cdot \hat{\mathbf{T}}(\mathbf{x},t) \tag{1.10}$$

Fourier Neural Operators (FNO) is an alternative avenue for learning operators by utilizing the fact that functions can be decomposed into linear combinations of basis functions, namely trigonometric basis in this particular case (Li et al., 2021). With this method the input function value is mapped to its corresponding output function value. This is done by first lifting the input function value to a higher dimension using a neural network and this is then passed through blocks composed of a Fourier transform, then a linear transform and filtering of higher modes, and finally the inverse Fourier transform. These blocks are stacked to a desired depth and finally another network projects the outputs to the target dimension. The reason a linear can be used is that differentiation is multiplication in the Fourier domain. One drawback with FNO is the requirement that output functions are not parameterized by coordinates and therefore is implicitly relative to the input function coordinates. To avoid this issue, Fanaskov and Oseledets (2023) reframes the problem by directly utilizing the coefficients of Fourier or Chebyshev basis. The model, termed Spectral Neural Operator (SNO) is trained on features of input function coefficients which are computed using Fourier or Chebyshev transforms and labels of output function coefficients using the same transforms. The authors point out one motivation for this approach which is that training neural networks on discretized data may not be ideal because unexpected outputs such as non-smooth interpolation may happen when the network is trained on one grid size and evaluated other grid sizes. With SNO, the interpolation of the function is smooth due to interpolation being done by Fourier basis functions which are sines and cosines for example. In a similar study, Du et al. (2024) extends the concept of mapping coefficients by proposing residuals in the spectral domain and leveraging Parseval's Identity to compute the spectral analog to the loss term in PINNs. This allows for self supervised learning in the spectral domain. The same benefits incorporating physics into PINNs also apply here without the pain points introduced by discretized model inputs and outputs. As a whole, operator learning creates an alternative approach that addresses the issue of retraining or recomputing the solution model. In addition, due to its data based approach, even systems with partially or fully unknown governing equations may be simulated.

A persistent challenge with all these approaches is the issue of optimization. While neural networks are modular and expressive which is proven by the universal approximation theorem (Cybenko, 1989; Hornik et al., 1989), their loss function present many local minima meaning it is non-convex. This can be mitigated by using advanced optimization techniques that can find a local minima close enough to the global minima such as Adam (Shrestha & Mahmood, 2019; Soydaner, 2020). However, the addition of PDE residuals into the loss function have worsened the highly non-convex loss landscape issue (Basir & Senocak, 2022; Krishnapriyan et al., 2021; Rathore et al., 2024). These problems range from the disparity in size of boundary and residual loss gradients to the fact that incorporation of residuals and boundary conditions themselves create a much more complex loss landscape. As a result, it is desirable to utilize a different machine learning algorithm that possesses a convex loss landscape. One family of such algorithms are Support Vector Machines (SVM) (Vapnik, 2000). The appeal of SVMs are the fact that the model is formulated as a quadratic programming problem. This means there are strong guarantees for convergence, generalization, and complexity. Another formulation called Least Squares Support Vector Machines (LSSVM) reformulates the problem as a linear system (Suykens, 2005). This leads to an easier problem that can be computed faster by well established algorithms like the many implementations of least squares solvers. Another advantage of the linear formulation is that this can be easily parallelized to exploit hardware like graphics processing units more widely known as GPUs in contrast to the commonly used Sequential Minimal Optimization (SMO) used for SVMs with quadratic objective functions.

The advantageous properties of SVM based methods have attracted research into their use for solving PDEs. An early work using SVMs to solve PDEs by Youxi Wu et al. (2005) introduced a method for solving the forward problem of Electro-Impedance Tomography. This work solved for the mathematical model of EIT which is given by Maxwell's equations by modeling the trial function as using an ε-SVR model. Another approach much more similar to PINNs was presented by Mehrkanoon and Suykens (2015). The residual and initial/boundary conditions are imposed as equality constraints on an LS-SVM objective function. A different study by Leake et al. (2019), the incorporation of physics into the model is done slightly differently by utilizing the theory of functional connections to directly embed constraints into the solution. This means that the proposed method would satisfy the boundary condition exactly. However, the authors point out that for PDEs in higher dimensions deriving and implementing this method can become cumbersome. These approaches, however, do not learn the PDE operator itself. Meaning they are also not practical for many-query problems.

Operator Learning for Weather Forecasting

In terms of weather forecasting, operator learning has been applied in terms of initial value problems. This problem formulation is reminiscent of time series prediction problems widely found in machine learning research. Researchers Kurth et al. (2023) developed FourCastNet which utilized Adaptive Fourier Neural Operator (AFNO), a transformer based model containing the previously mentioned FNO computational blocks by Li et al. (2021). This model was then able to be trained in a massively parallel manner. In a comparison with a traditional model called the Integrated Forecasting System from the European Center for Medium Range Weather Forecasts (ECMWF), FourCastNet is faster and much more efficient in terms of inference time, resulting in about 80,000 times speed up for a 100-member ensemble forecast. This is while performing much better than a previous deep learning approach. In another study, Bonev et al. (2023) proposed a variation on neural operators called Spherical Fourier Neural Operator (SFNO) which exploited the spherical nature of global forecasting by using Spherical Harmonic Transform (SHT) in place of Fourier Transform. This model when compared to AFNO and FNO, produced no visible artifacts in autoregressive rollouts for long range forecasting. In terms of forecasting, the model shows outcomes that matches the IFS which is a big leap forward in parity for traditional and machine learning based methods.

1.2 Problem Statement

The problems this work sets out to solve based on section 1.1 are:

1. What is the formulation a computational model for operator regression and therefore solving PDEs in the spectral domain using support vector machines?
2. How does the number of basis functions and model parameters impact the model performance?
3. How can one interpret the learned model?
4. Can the model learn to forecast the weather effectively?

1.3 Aims

Based on the stated problems in section 1.2, this study aims to acomplish the following:

1. The design and implementation of a computational model that maps coefficients in the spectral domain using Least Squares Support Vector Regression (LSSVR).
2. Hyperparameter optimization of the model.

3. Interpretation the model results and why some predictions turn out the way they do.
4. Comparison of SpectralSVR with SNO, DeepONet, FNO, and FDM on 4 problems

1.4 Contribution

In achieving the aims of this study the following contributions are made:

1. A novel use LSSVR which has a convex objective to learn solution operators of partial differential equations and operators in general.
2. Comparison of non-uniform Fourier transform for arbitrary spatial sampling with interpolated grid point values.
3. Interpretation of trained model.

1.5 Limitations

This work is limited to the following:

1. Functions the model works with are only continuous functions.
2. Hardware used in this project is limited to the standard offering on kaggle.com as of August 22, 2024.
3. The basis functions are limited to Fourier basis.

1.6 Sistematika Penulisan

CHAPTER II

KAJIAN PUSTAKA

Bab ini mengulas secara rinci konsep-konsep dasar yang berkaitan dengn pekerjaan penelitian TA dan deskripsi studi pustaka yang dilakukan. Judul bab tidak harus seperti yang dituliskan, melainkan dapat lebih fleksibel yang mencerminkan isi paparan pada bab ini. Demikian halnya dengan judul sub bab.

2.1 Least Squares Support Vector Machine

An arguably fundamental model widely used whether in pedagogical settings or otherwise is the support vector machine. It dates back to works by Vapnik & Lerner in 1963 (Recognition of Patterns with help of Generalized Portraits) and V. N. Vapnik & A. Ya. Chervonenkis in 1964 (A note on one class of perceptrons/On a perceptron class). As the development on SVM continued, what originally was a model for classification of separable data generalized to regression tasks as well Vapnik (2000 The Nature of Statistical Learning Theory).

2.1.1 Disadvantages

However, the main disadvantage of LSSVMs are the fact that they do not have the sparse property of SVMs which can leave performance on the table. A simple mitigation can be done by filtering training samples with small absolute values of lagrangian multipliers (Haifeng Wang & Dejin Hu, 2005).

To derive the least squares support vector regression (J.A. Suykens LSSVM Book) model we start with the linear expression:

$$y = W^{\mathsf{T}}\mathbf{x} + b \tag{2.1}$$

$$\min_{W,e} J(W,e) = \frac{1}{2}W^{\intercal}W + C\frac{1}{2}\sum_{k=1}^{n} e_k \tag{2.2}$$

Such that

$$y_k = W^{\intercal}\mathbf{x}_k + b + e_k \qquad k = 1,\ldots,n \tag{2.3}$$

$$L(W,b,e;\alpha) = \frac{1}{2}W^{\intercal}W + C\frac{1}{2}\sum_{k=1}^{n} e_k + \sum_{k=1}^{n} \alpha_k \left(W^{\intercal}\mathbf{x}_k + b + e_k - y_k\right) \tag{2.4}$$

Derive the KKT system

$$\begin{aligned}
\frac{\partial L}{\partial W} &= 0 \rightarrow W = \sum_{k=1}^{n} \alpha_k x_k \\
\frac{\partial L}{\partial b} &= 0 \rightarrow \sum_{k=1}^{n} \alpha_k = 0 \\
\frac{\partial L}{\partial e_k} &= 0 \rightarrow \alpha_k = Ce_k \qquad & k = 1,\ldots,n \\
\frac{\partial L}{\partial \alpha_k} &= 0 \rightarrow W^{\intercal}\mathbf{x}_k + b + e_k - y_k = 0 \qquad & k = 1,\ldots,n
\end{aligned} \tag{2.5}$$

After eliminating W and e, with $\mathbf{1}_n = \langle 1,\ldots,1\rangle$, $\mathbf{y} = [y_1,\ldots,y_n]$, and $\alpha = [\alpha_1,\ldots,\alpha_n]$ the solution is as follows in block matrix notation

$$\begin{bmatrix} 0 & \mathbf{1}_n^{\intercal} \\ \mathbf{1}_n & \Omega + \frac{I}{C} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \tag{2.6}$$

The solution in eq. (2.6)

Algorithm 1 LSSVR Training

Input: $\mathbf{X}, \mathbf{y}, \gamma$
Output: α, b
$\Omega \leftarrow [[]]$ ▷ Construct matrix of inner products in high dimensional space
for $k = 0 \rightarrow n$ **do**
  for $l = 0 \rightarrow n$ **do**
    $\Omega_{k,l} \leftarrow K(\mathbf{X}_k, \mathbf{X}_l)$
  end for
end for
$\mathbf{H} \leftarrow \Omega + \frac{\mathbf{I}}{\gamma}$
$\mathbf{A} \leftarrow [[]]$ ▷ Construct left hand side matrix
$\mathbf{A}_{0,0} \leftarrow 0$
for $k = 0 \rightarrow n$ **do**
  $\mathbf{A}_{k+1,0} \leftarrow 1$
  $\mathbf{A}_{0,k+1} \leftarrow 1$
end for
for $k = 0 \rightarrow n$ **do**
  for $l = 0 \rightarrow n$ **do**
    $\mathbf{A}_{k+1,l+1} \leftarrow \mathbf{H}_{k,l}$
  end for
end for
$\mathbf{B} \leftarrow []$ ▷ Construct left hand side of the equation
$\mathbf{B}_0 \leftarrow 0$
for $k = 0 \rightarrow n$ **do**
  $\mathbf{B}_{k+1} \leftarrow \mathbf{y}_k$
end for
$\mathbf{A}^{\dagger} \leftarrow pseudoInverse(\mathbf{A})$ ▷ Compute solution using pseudo inverse
$\mathbf{S} \leftarrow \mathbf{A}^{\dagger}\mathbf{B}$
$b \leftarrow \mathbf{S}_0$
for $k = 0 \rightarrow n$ **do**
  $\alpha_k \leftarrow \mathbf{S}_{k+1}$
end for

The basic psuedocode from the LSSVM Equation for function regression is defined in LSSVR Training for a training set of length n, features $\mathbf{X}$, and labels $\mathbf{y}$. Training the LSSVM means computing the values of langrange multipliers α and bias b. K is the kernel function used to compute the inner products in high dimensional space, here we asume the RBF kernel. $\mathbf{A}$ is a matrix of size $n+1$ by $n+1$. $\mathbf{H}$ is a matrix of size n by n. $\mathbf{I}$ is the identity. $\mathbf{B}$ is a vector of size $n+1$. $\mathbf{S}$ is a vector of size $n+1$.

After training the model can be used for prediction of unseen points. The psuedocode for prediction is shown in LSSVR Prediction for prediction features $\mathbf{U}$. The trained model uses the learned multipliers of training points α, the training points themselves $\mathbf{X}$, and the bias b.

Algorithm 2 LSSVR Prediction

Input: $\mathbf{U}, \alpha, \mathbf{X}, b$
Output: $\mathbf{v}$

```
Ω ← [[]]                    ▷ Construct matrix of inner products in high dimensional space
for k = 0 → m do
    for l = 0 → n do
        Ω_{k,l} ← K(U_k, X_l)
    end for
end for
v ← Ωα + 1_m b
```

In this exmple we will be using the function $2x^2+4$. The values of this function can be seen in Table 2.1.

No	x	y
1	0.0	4.0
2	0.33...	4.22...
3	0.66...	4.88...
4	1.0	6.0

Table 2.1: Example data of function $2x^2+4$

For $\Omega_{i,j} = K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

For example, with $\sigma = 1$, $x_i = 0.0$, & $x_j = 0.33\ldots$

$$\begin{aligned} K(0.0, 0.33) &= exp\left(-\frac{\|0.0 - 0.33\|^2}{2(1)^2}\right) \\ &= exp\left(-\frac{0.33^2}{2}\right) \\ &= 0.9460 \end{aligned} \tag{2.7}$$

$$\Omega \leftarrow \begin{bmatrix} 1.0000 & 0.9460 & 0.8007 & 0.6065 \\ 0.9460 & 1.0000 & 0.9460 & 0.8007 \\ 0.8007 & 0.9460 & 1.0000 & 0.9460 \\ 0.6065 & 0.8007 & 0.9460 & 1.0000 \end{bmatrix} \tag{2.8}$$

$$\mathbf{I}\frac{1}{\gamma} \rightarrow \mathbf{I}\frac{1}{5} \rightarrow \begin{bmatrix} 0.2000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.2000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.2000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.2000 \end{bmatrix} \tag{2.9}$$

$$\Omega + \mathbf{I}\frac{1}{5} \rightarrow H \rightarrow \begin{bmatrix} 1.2000 & 0.9460 & 0.8007 & 0.6065 \\ 0.9460 & 1.2000 & 0.9460 & 0.8007 \\ 0.8007 & 0.9460 & 1.2000 & 0.9460 \\ 0.6065 & 0.8007 & 0.9460 & 1.2000 \end{bmatrix} \tag{2.10}$$

$$A \rightarrow \begin{bmatrix} 0.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \\ 1.0000 & 1.2000 & 0.9460 & 0.8007 & 0.6065 \\ 1.0000 & 0.9460 & 1.2000 & 0.9460 & 0.8007 \\ 1.0000 & 0.8007 & 0.9460 & 1.2000 & 0.9460 \\ 1.0000 & 0.6065 & 0.8007 & 0.9460 & 1.2000 \end{bmatrix} \tag{2.11}$$

$$B \rightarrow \begin{bmatrix} 0.0000 \\ 4.0000 \\ 4.2222 \\ 4.8889 \\ 6.0000 \end{bmatrix} \tag{2.12}$$

$$A^{\dagger} \rightarrow \begin{bmatrix} -0.8994 & 0.4348 & 0.0652 & 0.0652 & 0.4348 \\ 0.4348 & 2.0686 & -1.6490 & -0.5292 & 0.1096 \\ 0.0652 & -1.6490 & 3.3774 & -1.1992 & -0.5292 \\ 0.0652 & -0.5292 & -1.1992 & 3.3774 & -1.6490 \\ 0.4348 & 0.1096 & -0.5292 & -1.6490 & 2.0686 \end{bmatrix} \tag{2.13}$$

$$A^{\dagger}B \rightarrow S \rightarrow \begin{bmatrix} 4.9421 \\ -0.6177 \\ -1.3737 \\ -0.5625 \\ 2.5538 \end{bmatrix} \tag{2.14}$$

$$b \rightarrow 4.9421 \tag{2.15}$$

$$\alpha \rightarrow \begin{bmatrix} -0.6177 \\ -1.3737 \\ -0.5625 \\ 2.5538 \end{bmatrix} \tag{2.16}$$

Prediction

$$U \rightarrow \begin{bmatrix} 0.3 \\ 0.2 \\ 0.5 \end{bmatrix} \tag{2.17}$$

$$\Omega \rightarrow \begin{bmatrix} 0.9560 & 0.9994 & 0.9350 & 0.7827 \\ 0.9802 & 0.9912 & 0.8968 & 0.7261 \\ 0.8825 & 0.9862 & 0.9862 & 0.8825 \end{bmatrix} \tag{2.18}$$

$$\Omega\alpha \rightarrow \begin{bmatrix} -0.4904 \\ -0.6170 \\ -0.2008 \end{bmatrix} \tag{2.19}$$

$$\Omega\alpha + b\mathbf{1}_m \rightarrow v \rightarrow \begin{bmatrix} 4.4516 \\ 4.3251 \\ 4.7413 \end{bmatrix} \tag{2.20}$$

Where $\mathbf{1}_m$ is a vector of 1s with the length of U.

2.2 Sub Bab B

Suatu penelitian tidak dapat lepas dari capaian pengetahuan dan pemahaman yang sudah dipublikasikan. Deskripsi tentang capaian ini menjadi penting karena selain menunjukkan tingkat pemahaman mahasiswa, juga mengetahui tempat pekerjaan penelitian TA dalam konstelasi capaian tersebut. Studi pustaka dan paparan hasilnya dapat memperkaya wawasan tentang topik yang diangkat pada penelitian TA.

2.3 Membuat Persamaan

Secara prinsip, suatu persamaan menyatu dalam kalimat. Letak persamaan dapat berada di awal, tengah, atau akhir kalimat. Dengan demikian, pada akhir persamaan harus diberikan tanda baca, misalnya koma, titik koma, atau titik, yang menekankan kehadiran persamaan dalam kalimat. Tidak semua persamaan harus diberi nomor. Persamaan yang dirujuk pada naskah TA saja yang harus diberi nomor. Kode awal penomoran ini adalah nomor urut bab, termasuk untuk persamaan pada Lampiran, dengan urutan alfabet kapital.

Setiap notasi harus unik atau tunggal, sehingga arti setiap notasi adalah unik atau tunggal juga. Arti satu notasi harus dituliskan segera ketika notasi tersebut muncul, dan tidak diulang lagi setelahnya.

2.3.1 Contoh Persamaan Sederhana

Persamaan (2.21) mendeskripsikan dinamika fungsi gelombang $\psi(\vec{r},t)$ di bawah pengaruh potensial $V(\vec{r})$ dan dituliskan sebagai berikut:

$$i\hbar\frac{\partial\psi}{\partial t} = -\frac{\hbar^2}{2m}\vec{\nabla}^2\psi + V(\vec{r})\psi, \tag{2.21}$$

dengan m adalah massa partikel dan $\hbar$ merupakan konstanta Planck tereduksi.

Di akhir persamaan (2.21) diberi koma karena berada ditengah kalimat. Untuk merujuk ke persamaan yang telah ditulis, gunakan perintah `\ref{}`.

Untuk menuliskan beberapa set persamaan yang masih terhubung, gunakan `\subequations{}`. Misal kita punya persamaan diferensial terkopel, kita bisa tuliskan

$$\frac{dy}{dt} = -x, \tag{2.22a}$$

$$\frac{dx}{dt} = -y. \tag{2.22b}$$

Kalau perlu matriks, kita bisa tulis seperti berikut.

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.23}$$

2.4 Referensi dan Citation

Sitasi dapat dimasukkan ke dalam Tugas Akhir seperti ini **Fujita1996**. Untuk sitasi dengan beberapa sumber, dapat dituliskan juga **hohen1964**; **Kim2006**. Atau untuk tiga sumber berarti **kongkanand2006**; **kresse1999**; **Leibb1993**.

CHAPTER III

METODOLOGI PENELITIAN

Secara umum, metode penelitian yang digunakan pada pekerjaan penelitian disampaikan pada bab ini. Judul bab tidak harus seperti yang dituliskan. Dalam kata lain bisa diubah sesuai kebutuhan.

3.1 Computational Model

The proposed model derives inspiration from DeepONet, Neural Spectral methods, and Spectral Neural Operators. The method uses LSSVR to map between the input coefficients/functions and the solution coefficients. This mapping is the surrogate of the operator in the spectral domain. To exploit the nature of the problem, the kernel chosen has particular characteristics that makes it useful for the spectral/function domain. These are kernels such as the maximum distance between two functions or something that lies in the spectral domain favoring low/high frequencies depending on the particular problem.

$$\min_{W_b, W_t, \xi} \quad \frac{1}{2}\|W_b\|_F^2 + \frac{1}{2}\|W_t\|_F^2 + \frac{c}{2}\sum_{i=1}^{N} \xi_i^2 \tag{3.1a}$$

$$\text{s.t.} \quad y_i = (W_b f_i)^\top (W_t v_i) + \xi_i, \quad i = 1, \ldots, N \tag{3.1b}$$

Converting the optimization problem above to the lagrange formulation

$$\begin{aligned} L(W_b, W_t, \xi; \alpha) = &\frac{1}{2}\|W_b\|_F^2 + \frac{1}{2}\|W_t\|_F^2 + \frac{c}{2}\sum_{i=1}^{N} \xi_i^2 \\ &+ \sum_{i=1}^{N} \alpha_i (y_i - \xi_i - (W_b f_i)^\top (W_t v_i)) \end{aligned} \tag{3.2}$$

Taking partial derivatives with respect to

$$\frac{\partial L(W_b, b_b, W_t, b_t, \xi; \alpha)}{\partial W_b} = 0$$

$$\frac{2}{2}W_b + 0 + 0 - \sum_{i=1}^{l} \alpha_i((W_t v_i + b_t) f_i^{\top}) = 0$$

$$W_b - \sum_{i=1}^{l} \alpha_i((W_t v_i + b_t) f_i^{\top}) = 0$$

$$W_b = \sum_{i=1}^{l} \alpha_i(W_t v_i f_i^{\top} + b_t f_i^{\top}) \tag{3.3}$$

$$W_b = \sum_{i=1}^{l} (\alpha_i W_t v_i f_i^{\top}) + \sum_{i=1}^{l} (\alpha_i b_t f_i^{\top})$$

$$W_b = W_t \sum_{i=1}^{l} (\alpha_i v_i f_i^{\top}) + b_t \sum_{i=1}^{l} (\alpha_i f_i^{\top})$$

$$W_b = W_t \sum_{i=1}^{l} (\alpha_i v_i f_i^{\top}) + b_t \sum_{i=1}^{l} (\alpha_i f_i^{\top})$$

$$
\begin{aligned}
\frac{\partial L(W_b, b_b, W_t, b_t, \xi; \alpha)}{\partial W_t} &= 0 \\
0 + \frac{2}{2} W_t + 0 - \sum_{i=1}^{l} \alpha_i ((W_b f_i + b_b) v_i^\top) &= 0 \\
W_t - \sum_{i=1}^{l} \alpha_i ((W_b f_i + b_b) v_i^\top) &= 0 \\
W_t &= \sum_{i=1}^{l} \alpha_i (W_b f_i v_i^\top + b_b v_i^\top) \\
W_t &= \sum_{i=1}^{l} \alpha_i ((\sum_{j=1}^{l} \alpha_j (W_t v_j f_j^\top + b_t f_j^\top)) f_i v_i^\top + b_b v_i^\top) \\
W_t &= \sum_{i=1}^{l} (\sum_{j=1}^{l} (\alpha_i \alpha_j W_t v_j f_j^\top f_i v_i^\top + \alpha_i \alpha_j b_t f_j^\top f_i v_i^\top) + \alpha_i b_b v_i^\top) \\
W_t &= \sum_{i=1}^{l} (\sum_{j=1}^{l} (\alpha_i \alpha_j W_t v_j f_j^\top f_i v_i^\top) + \sum_{j=1}^{l} (\alpha_i \alpha_j b_t f_j^\top f_i v_i^\top) + \alpha_i b_b v_i^\top) \\
W_t - \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j W_t v_j f_j^\top f_i v_i^\top &= \sum_{i=1}^{l} (\sum_{j=1}^{l} (\alpha_i \alpha_j b_t f_j^\top f_i v_i^\top) + \alpha_i b_b v_i^\top) \\
W_t (I - \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j v_j f_j^\top f_i v_i^\top) &= \sum_{i=1}^{l} (\sum_{j=1}^{l} (\alpha_i \alpha_j b_t f_j^\top f_i v_i^\top) + \alpha_i b_b v_i^\top) \\
W_t &= \sum_{i=1}^{l} (\sum_{j=1}^{l} (\alpha_i \alpha_j b_t f_j^\top f_i v_i^\top) + \alpha_i b_b v_i^\top)(I - \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j v_j f_j^\top f_i v_i^\top)^{-1} \\
W_t &= \sum_{i=1}^{l} (\alpha_i (\sum_{j=1}^{l} (\alpha_j b_t f_j^\top f_i) + b_b) v_i^\top)(I - \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j v_j f_j^\top f_i v_i^\top)^{-1}
\end{aligned}
\tag{3.4}
$$

$$
\begin{aligned}
\frac{\partial L(W_b, b_b, W_t, b_t, \xi; \alpha)}{\partial b_b} &= 0 \\
0 + 0 + 0 - \sum_{i=1}^{l} \alpha_i (W_t v_i + b_t) &= 0 \\
\sum_{i=1}^{l} \alpha_i (W_t v_i + b_t) &= 0
\end{aligned}
\tag{3.5}
$$

$$
\begin{aligned}
\frac{\partial L(W_b, b_b, W_t, b_t, \xi; \alpha)}{\partial b_t} &= 0 \\
0 + 0 + 0 - \sum_{i=1}^{l} \alpha_i (W_b f_i + b_b) &= 0 \\
\sum_{i=1}^{l} \alpha_i (W_b f_i + b_b) &= 0
\end{aligned}
\tag{3.6}
$$

$$
\begin{aligned}
\frac{\partial L(W_b, b_b, W_t, b_t, \xi; \alpha)}{\partial \xi} &= 0 \\
0 + 0 + \frac{2c}{2} \sum_{i=1}^{l} \xi_i - \sum_{i=1}^{l} \alpha_i &= 0 \\
\sum_{i=1}^{l} c\xi_i - \alpha_i &= 0 \\
c\xi_i &= \alpha_i
\end{aligned}
\tag{3.7}
$$

$$
\begin{aligned}
\frac{\partial L(W_b, b_b, W_t, b_t, \xi; \alpha)}{\partial \alpha} &= 0 \\
0 + 0 + 0 + \sum_{i=1}^{l} y_i - \xi_i - (W_b f_i + b_b)^\top (W_t v_i + b_t) &= 0 \\
\sum_{i=1}^{l} y_i - \xi_i - (W_b f_i + b_b)^\top (W_t v_i + b_t) &= 0
\end{aligned}
\tag{3.8}
$$

3.2 Sub Bab α

Misal kita mau masukin tabel, kita bisa juga.

Table 3.1: Tabel Sederhana Pertama

G	dim G	dim F
$SU(N)$	N^2-1	N
$SO(N)$	$\frac{1}{2}N(N-1)$	N
$Sp(N)$	$N(2N+1)$	$2N$
E_6	78	27
E_7	133	56
E_8	248	248
F_4	52	6
G_2	14	7

3.3 Sub Bab β

Atau bisa juga seperti berikut.

Table 3.2: Tabel Sederhana Kedua

G	dim G	dim F
$SU(N)$	N^2-1	N
$SO(N)$	$\frac{1}{2}N(N-1)$	N
$Sp(N)$	$N(2N+1)$	$2N$
E_6	78	27
E_7	133	56
E_8	248	248
F_4	52	6
G_2	14	7

CHAPTER IV

HASIL DAN ANALISIS

4.1 Sub Bab Hasil

Bab ini memaparkan pekerjaan penelitian dan, terutama, hasil-hasilnya, untuk dianalisis. Secara komprehensif bab ini merepre-sentasikan curahan pemikiran dan kemampuan mahasiswa dalam menjalani pekerjaan penelitian, yang hasil-hasilnya dapat dipertanggungjawabkan. Banyak pendukung yang diperlukan dalam penulisan bab ini, seperti skema penting pengolahan data, penurunan model matematika, asumsi khusus, tabulasi hasil dan analisis, dan gambar atau grafik yang membantu dalam paparan analisis. Judul bab dan sub bab disesuaikan dengan isi paparan.

4.2 Memasukkan Gambar

Setiap gambar harus dirujuk pada naskah TA, termasuk gambar pada Lam- piran, menggunakan huruf pertama kapital (G) dan nomor gambar, tidak berdasarkan posisi relatifnya (misalnya di bawah ini atau sebelum ini). Format gambar yang umum adalah jpg, png, dan postscript (ps atau eps). Ukuran huruf pada nama sumbu dan label figures/grafik harus cukup besar dan jelas, demikian halnya dengan angka pada sumbu. Gambar dan grafik dapat berwarna dengan pilihan warna yang tegas dan jelas.

4.2.1 Contoh Gambar Sederhana

Contoh menginput gambar pada buku TA pada Apabila ada dua gambar, kita juga

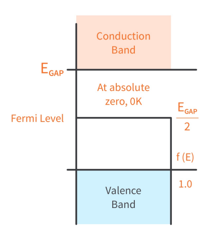
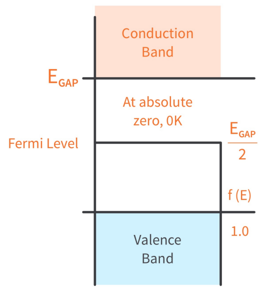
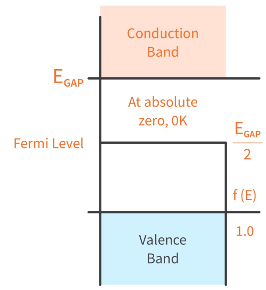


Figure 4.1: Tingkatan Fermi pada Bahan Semikonduktor

bisa menaruh keduanya berdampingan.



(a)



(b)

Figure 4.2: Dengan menempatkan gambar (a) dan (b), pembaca akan lebih mudah membandingkan keduanya.

CHAPTER V

SIMPULAN DAN SARAN

5.1 Simpulan

Bab ini merupakan pamungkas berupa rincian rangkuman yang merupakan simpulan dari analisis yang telah dilakukan. Simpulan ini menyajikan sejumlah hal penting yang disampaikan secara ringkas, padat, dan utuh, yang menjawab tujuan penelitian yang dituliskan pada Bab Pendahuluan. Sangat mungkin ada beberapa konsekuensi dan implikasi yang ditimbulkan dari simpulan yang dihasilkan, yang sepatutnya menjadi perhatian pada penelitian berikutnya. Judul bab dapat disesuaikan, namun umumnya ada *Simpulan* yang memang mendominasi isi bab ini.

5.2 Saran

Sejumlah ide yang muncul ketika melaksanakan penelitian TA dapat menjadi bahan atau topik untuk pekerjaan selanjutnya. Hal ini dapat berupa perbaikan atau ragam lain dari apa yang telah dilakukan sepanjang penelitian. Sub bab ini menjadi sumber informasi penting bagi, utamanya mahasiswa, yang akan melakukan penelitian lanjutan.

Bibliography

Aarts, L. P., & Van Der Veer, P. (2001). Neural Network Method for Solving Partial Differential Equations. *Neural Processing Letters*, *14*(3), 261–271. https://doi.org/10.1023/A:1012784129883

Anselmo, L., & Pardini, C. (2005). Computational methods for reentry trajectories and risk assessment. *Advances in Space Research*, *35*(7), 1343–1352. https://doi.org/10.1016/j.asr.2005.04.089

Anwar, M. R., Liu, D. L., Macadam, I., & Kelly, G. (2013). Adapting agriculture to climate change: A review. *Theoretical and Applied Climatology*, *113*(1-2), 225–245. https://doi.org/10.1007/s00704-012-0780-1

Astitha, M., & Nikolopoulos, E. (Eds.). (2023). *Extreme weather forecasting : State of the science, uncertainty and impacts*. Elsevier.
OCLC: 1347429101.

Basir, S., & Senocak, I. (2022). Critical Investigation of Failure Modes in Physics-informed Neural Networks. *AIAA SCITECH 2022 Forum*. https://doi.org/10.2514/6.2022-2353

Berliner, L. M. (2003). Physical-statistical modeling in geophysics. *Journal of Geophysical Research: Atmospheres*, *108*(D24), 2002JD002865. https://doi.org/10.1029/2002JD002865

Bernardi, M., Sánchez, H. D., Sheth, R. K., Brownstein, J. R., & Lane, R. R. (2022). Stellar population analysis of MaNGA early-type galaxies: IMF dependence and systematic effects. *Monthly Notices of the Royal Astronomical Society*, *518*(3), 4713–4733. https://doi.org/10.1093/mnras/stac3287

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., & Anandkumar, A. (2023, July 23–29). Spherical Fourier neural operators: Learning stable dynamics on the sphere. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th international conference on machine learning* (pp. 2806–2823, Vol. 202). PMLR. https://proceedings.mlr.press/v202/bonev23a.html

Buoni, M., & Petzold, L. (2007). An efficient, scalable numerical algorithm for the simulation of electrochemical systems on irregular domains. *Journal of Computational Physics*, *225*(2), 2320–2332. https://doi.org/10.1016/j.jcp.2007.03.025

Chen, J.-S., Hillman, M., & Chi, S.-W. (2017). Meshfree Methods: Progress Made after 20 Years. *Journal of Engineering Mechanics*, *143*(4), 04017001. https://doi.org/10.1061/(ASCE)EM.1943-7889.0001176

Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, *6*(4), 911–917. https://doi.org/10.1109/72.392253

Climate change 2022: Impacts, adaptation and vulnerability : Working Group II contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. (2023). Cambridge University Press.
OCLC: 1391150208.

Cogato, A., Meggio, F., De Antoni Migliorati, M., & Marinello, F. (2019). Extreme Weather Events in Agriculture: A Systematic Review. *Sustainability*, *11*(9), 2547. https://doi.org/10.3390/su11092547

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314. https://doi.org/10.1007/BF02551274

Du, Y., Chalapathi, N., & Krishnapriyan, A. S. (2024). Neural spectral methods: Self-supervised learning in the spectral domain. *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=2DbVeuoa6a

Fanaskov, V. S., & Oseledets, I. V. (2023). Spectral Neural Operators. *Doklady Mathematics*, *108*(S2), S226–S232. https://doi.org/10.1134/S1064562423701107

Galkin, I. A., Reinisch, B. W., Vesnin, A. M., Bilitza, D., Fridman, S., Habarulema, J. B., & Veliz, O. (2020). Assimilation of sparse continuous near-earth weather measurements by NECTAR model morphing. *Space weather : the international journal of research & applications*, *18*(11), e2020SW002463. https://doi.org/10.1029/2020SW002463

Gao, H., Sun, L., & Wang, J.-X. (2021a). PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *Journal of Computational Physics*, *428*, 110079. https://doi.org/10.1016/j.jcp.2020.110079

Gao, H., Sun, L., & Wang, J.-X. (2021b). Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, *33*(7), 073603. https://doi.org/10.1063/5.0054312

Gaul, L., Klein, P., & Plenge, M. (1991). Simulation of wave propagation in irregular soil domains by BEM and associated small scale experiments. *Engineering Analysis with Boundary Elements*, *8*(4), 200–205. https://doi.org/10.1016/0955-7997(91)90014-K

Gong, X., Herty, M., Piccoli, B., & Visconti, G. (2023). Crowd Dynamics: Modeling and Control of Multiagent Systems. *Annual Review of Control, Robotics, and Autonomous Systems*, *6*(1), 261–282. https://doi.org/10.1146/annurev-control-060822-123629

Govind, R., Garg, N., & Mittal, V. (2020). Weather, Affect, and Preference for Hedonic Products: The Moderating Role of Gender. *Journal of Marketing Research*, *57*(4), 717–738. https://doi.org/10.1177/0022243720925764

Haifeng Wang & Dejin Hu. (2005). Comparison of SVM and LS-SVM for Regression. *2005 International Conference on Neural Networks and Brain*, *1*, 279–283. https://doi.org/10.1109/ICNNB.2005.1614615

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hughes, R. (2000). The flow of large crowds of pedestrians. *Mathematics and Computers in Simulation*, *53*(4-6), 367–370. https://doi.org/10.1016/S0378-4754(00)00228-7

Jia, Y., Liu, K., & Zhang, X. S. (2024). Modulate stress distribution with bio-inspired irregular architected materials towards optimal tissue support. *Nature Communications*, *15*(1), 4072. https://doi.org/10.1038/s41467-024-47831-2

Jin, M., Wang, L., Ge, F., & Yan, J. (2023). Detecting the interaction between urban elements evolution with population dynamics model. *Scientific Reports*, *13*(1), 12367. https://doi.org/10.1038/s41598-023-38979-w

Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., & Mahoney, M. W. (2021). Characterizing possible failure modes in physics-informed neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems* (pp. 26548–26560, Vol. 34).

Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2021/file/df438e5206f31600e6ae4af72f2725f1-Paper.pdf

Kurth, T., Subramanian, S., Harrington, P., Pathak, J., Mardani, M., Hall, D., Miele, A., Kashinath, K., & Anandkumar, A. (2023). FourCastNet: Accelerating Global High-Resolution Weather Forecasting Using Adaptive Fourier Neural Operators. *Proceedings of the Platform for Advanced Scientific Computing Conference*, 1–11. https://doi.org/10.1145/3592979.3593412

Kwasniok, F. (2012). Data-based stochastic subgrid-scale parametrization: An approach using cluster-weighted modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *370*(1962), 1061–1086. https://doi.org/10.1098/rsta.2011.0384

Leake, C., Johnston, H., Smith, L., & Mortari, D. (2019). Analytically Embedding Differential Equation Constraints into Least Squares Support Vector Machines Using the Theory of Functional Connections. *Machine Learning and Knowledge Extraction*, *1*(4), 1058–1083. https://doi.org/10.3390/make1040060

Lepage, S., & Morency, C. (2021). Impact of Weather, Activities, and Service Disruptions on Transportation Demand. *Transportation Research Record: Journal of the Transportation Research Board*, *2675*(1), 294–304. https://doi.org/10.1177/0361198120966326

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*. https://openreview.net/forum?id=c8P9NQVtmnO

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, *3*(3), 218–229. https://doi.org/10.1038/s42256-021-00302-5

Malhi, G. S., Kaur, M., & Kaushik, P. (2021). Impact of Climate Change on Agriculture and Its Mitigation Strategies: A Review. *Sustainability*, *13*(3), 1318. https://doi.org/10.3390/su13031318

Mehrkanoon, S., & Suykens, J. A. (2015). Learning solutions to partial differential equations using LS-SVM. *Neurocomputing*, *159*, 105–116. https://doi.org/10.1016/j.neucom.2015.02.013

Mengaldo, G., Wyszogrodzki, A., Diamantakis, M., Lock, S.-J., Giraldo, F. X., & Wedi, N. P. (2019). Current and Emerging Time-Integration Strategies in Global Numerical Weather and Climate Prediction. *Archives of Computational Methods in Engineering*, *26*(3), 663–684. https://doi.org/10.1007/s11831-018-9261-8

Monmonier, M. S. (1999). *Air apparent: How meteorologists learned to map, predict, and dramatize weather*. Univ. of Chicago Press.

Moon, S., Kang, M. Y., Bae, Y. H., & Bodkin, C. D. (2018). Weather sensitivity analysis on grocery shopping. *International Journal of Market Research*, *60*(4), 380–393. https://doi.org/10.1177/1470785317751614

Mukherjee, S., Goswami, D., & Chatterjee, S. (2015). A Lagrangian Approach to Modeling and Analysis of a Crowd Dynamics. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(6), 865–876. https://doi.org/10.1109/TSMC.2015.2389763

Muller, A. P. O., Costa, J. C., Bom, C. R., Klatt, M., Faria, E. L., de Albuquerque, M. P., & de Albuquerque, M. P. (2023). Deep pre-trained FWI: Where supervised learning meets the physics-informed neural networks. *Geophysical Journal International*, *235*(1), 119–134. https://doi.org/10.1093/gji/ggad215

Ni, P., Sun, L., Yang, J., & Li, Y. (2022). Multi-End Physics-Informed Deep Learning for Seismic Response Estimation. *Sensors*, *22*(10), 3697. https://doi.org/10.3390/s22103697

Nurmi, P., Perrels, A., & Nurmi, V. (2013). Expected impacts and value of improvements in weather forecasting on the road transport sector. *Meteorological Applications*, *20*(2), 217–223. https://doi.org/10.1002/met.1399

Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Rathore, P., Lei, W., Frangella, Z., Lu, L., & Udell, M. (2024). *Challenges in Training PINNs: A Loss Landscape Perspective* (2). https://doi.org/10.48550/ARXIV.2402.01868

Reinhardt, J. C., Chen, X., Liu, W., Manchev, P., & Paté-Cornell, M. E. (2016). Asteroid Risk Assessment: A Probabilistic Approach. *Risk Analysis*, *36*(2), 244–261. https://doi.org/10.1111/risa.12453

Shrestha, A., & Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. *IEEE Access*, *7*, 53040–53065. https://doi.org/10.1109/ACCESS.2019.2912200

Soydaner, D. (2020). A Comparison of Optimization Algorithms for Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence*, *34*(13), 2052013. https://doi.org/10.1142/S0218001420520138

Spanos, A. (2006). Where do statistical models come from? Revisiting the problem of specification. In *Institute of Mathematical Statistics Lecture Notes - Monograph Series* (pp. 98–119). Institute of Mathematical Statistics. https://doi.org/10.1214/074921706000000419

Stott, P. A., Christidis, N., Otto, F. E. L., Sun, Y., Vanderlinden, J.-P., van Oldenborgh, G. J., Vautard, R., von Storch, H., Walton, P., Yiou, P., & Zwiers, F. W. (2016). Attribution of extreme weather and climate-related events. *WIREs Climate Change*, *7*(1), 23–41. https://doi.org/10.1002/wcc.380

Suykens, J. A. K. (Ed.). (2005). *Least squares support vector machines* (Repr). World Scientific.

Tian, J., Zhang, Y., & Zhang, C. (2018). Predicting consumer variety-seeking through weather data analytics. *Electronic Commerce Research and Applications*, *28*, 194–207. https://doi.org/10.1016/j.elerap.2018.02.001

Tian, X., Cao, S., & Song, Y. (2021). The impact of weather on consumer behavior and retail performance: Evidence from a convenience store chain in China. *Journal of Retailing and Consumer Services*, *62*, 102583. https://doi.org/10.1016/j.jretconser.2021.102583

Uzan, J.-P. (2003). The fundamental constants and their variation: Observational and theoretical status. *Reviews of Modern Physics*, *75*(2), 403–455. https://doi.org/10.1103/RevModPhys.75.403

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory* (2nd ed). Springer New York. https://doi.org/10.1007/978-1-4757-3264-1

Wang, R., Kashinath, K., Mustafa, M., Albert, A., & Yu, R. (2020). Towards Physics-informed Deep Learning for Turbulent Flow Prediction. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1457–1466. https://doi.org/10.1145/3394486.3403198

Wilby, R. L., & Yu, D. (2013). Rainfall and temperature estimation for a data sparse region. *Hydrology and Earth System Sciences*, *17*(10), 3937–3955. https://doi.org/10.5194/hess-17-3937-2013

Youxi Wu, Ying Li, Lei Guo, Weili Yan, Xueqin Shen, & Kun Fu. (2005). SVM for Solving Forward Problems of EIT. *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 1559–1562. https://doi.org/10.1109/IEMBS.2005.1616732

Zhang, R., Liu, Y., & Sun, H. (2020). Physics-guided convolutional neural network (PhyCNN) for data-driven seismic response modeling. *Engineering Structures*, *215*, 110704. https://doi.org/10.1016/j.engstruct.2020.110704

Zhao, X., Gong, Z., Zhang, Y., Yao, W., & Chen, X. (2023). Physics-informed convolutional neural networks for temperature field prediction of heat source layout without labeled data. *Engineering Applications of Artificial Intelligence*, *117*, 105516. https://doi.org/10.1016/j.engappai.2022.105516

LAMPIRAN A
KODE PROGRAM

1.1 PROGRAM SATU

1.2 PROGRAM DUA

LAMPIRAN B
GAMBAR-GAMBAR