Applied Machine Learning with Kernel Features for Variational Partial Differential Equations

by

Tan Phuong Dong Le

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2026

© Tan Phuong Dong Le 2026

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Cécile Piret Associate Professor, Department of Mathematical Sciences Michigan Technological University
Supervisor(s):	Hans De Sterck Professor, Department of Applied Mathematics University of Waterloo
	Giang Tran Associate Professor, Department of Applied Mathematics University of Waterloo
Internal Member(s):	Mohammad Kohandel Professor, Department of Applied Mathematics University of Waterloo
	Jun Liu Professor, Department of Applied Mathematics University of Waterloo
External Member:	Saeed Ghadimi Associate Professor, Department of Management Science and Engineering University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners. I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Tan Phuong Dong Le was the sole author of Chapters 1, 2, and 5, which were written under the supervision of Dr. Hans De Sterck and Dr. Giang Tran and were not intended for publication. Tan Phuong Dong Le's contributions to Chapters 3 and 4 are detailed as follows:

Chapter 3: The research results presented in this chapter are based on a collaboration between Tan Phuong Dong Le, Dr. Giang Tran, and Dr. Hans De Sterck. It has been submitted in [61]. Tan Phuong Dong Le worked on the derivations and performed the numerical experiments. Editing, verification, and formulation were provided by Dr. Hans De Sterck and Dr. Giang Tran. The manuscript was written by Tan Phuong Dong Le.

Chapter 4: The research project presented in this chapter is the result of a collaboration between Tan Phuong Dong Le, Dr. Giang Tran, and Dr. Hans De Sterck. The manuscript can be accessed at [60] and is currently being finalized for publication. Tan Phuong Dong Le worked on the theoretical formulation and conducted all numerical experiments. The manuscript was written by Tan Phuong Dong Le.

Abstract

This thesis develops mesh-free approximation methods for linear elliptic and free-boundary partial differential equations within variational formulations. It consists of two main contributions.

First, we construct radial basis function approximations for linear elliptic partial differential equations and free-boundary value problems. Classical problems, including the Poisson and Reaction–diffusion equations, are reformulated as minimization tasks, with boundary conditions imposed through penalty methods. For obstacle problems, we consider a variational formulation with a non-smooth ℓ^1-regularized objective. To address the ill-conditioning of the discrete linear systems, a truncated singular value decomposition is employed as a stabilization mechanism. Numerical experiments demonstrate that the proposed method achieves high accuracy and efficiency, and that it outperforms competing approaches, including finite-difference methods, Galerkin methods, and neural network-based solvers.

The second contribution extends this variational framework to high-dimensional problems by means of random feature approximations. The resulting formulation yields scalable convex optimization problems that are well suited for high-dimensional settings. Numerical results indicate that the proposed approach surpasses other mesh-free methodologies, particularly physics-informed neural networks and the Deep Ritz method, while preserving high accuracy.

In general, these contributions provide effective variational mesh-free algorithms for linear elliptic, obstacle-type, and high-dimensional partial differential equations. The findings highlight the potential of radial basis function networks and random feature models as accurate, stable, and scalable methods for the numerical approximation of variational partial differential equations.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Hans De Sterck and Associate Prof. Giang Tran. Their outstanding mentorship, thoughtful advice, and continued support have been the most important contributions to my entire PhD process. Prof. Hans De Sterck's broad experience in applied mathematics and his supportive challenge to learn new and more complex concepts have been an inspiration. Prof. Giang Tran's expertise in the field and her thorough critique of my work have significantly influenced the direction of my research.

I wish to extend my sincere appreciation to the members of my thesis committee, including Dr. Cécile Piret, Dr. Jun Liu, Dr. Mohammad Kohandel and Dr. Saeed Ghademi, for their generous dedication of time, specialized expertise, and constructive critiques. Their contributions have substantially enhanced the rigor, clarity, and overall quality of this research.

I would like to extend my sincere appreciation to my family for their sustained and unwavering support. I wish to acknowledge my father, Tan Anh Kiet Le, whose intellectual guidance and exemplary work ethic have consistently served as a source of motivation. His strong commitment to education and continuous encouragement have been instrumental to my academic achievements. I also express my gratitude to my mother, Ngoc Tram Huong Nguyen, whose enduring affection, patience, and care have provided essential support throughout this journey. Her dedication and selflessness have significantly contributed to the realization of my goals thus far. Furthermore, I acknowledge my brother, Tan Hai Dong Le, for his steadfast commitment and support. His sense of humor, empathy, and integrity have constituted a vital source of resilience during both the demanding and fulfilling stages of my research endeavors.

I would especially like to express my heartfelt gratitude to my fiancée, Thi Minh Thu Tran, for her unwavering love, patience, and encouragement throughout this journey. Her steadfast support, understanding, and faith in my abilities have been a constant source of strength during both the most challenging and the most rewarding moments of my doctoral studies. Her kindness, companionship, and ability to provide comfort and perspective during periods of stress have held more significance for me than words can fully express. I am profoundly grateful for her presence in my life and for the encouragement and happiness she has contributed to me throughout this pivotal stage of my academic and personal journey.

I am deeply grateful to my grandparents, Ngoc Quang Nguyen and Thi Thien Huong Phan, for their unconditional love, care, and support throughout my life. I also sincerely

thank my uncles and aunts, including Ngoc Hoai Nam Nguyen, Thi Nguyet Nga Huynh, Van Ngo Nguyen, Ngoc Que Huong Nguyen, Ngoc Nha Nam Nguyen, and Thanh Huyen Tran, for their kindness, encouragement, and support. Their presence has greatly shaped my personal growth and given me a strong sense of family throughout this journey.

I am deeply grateful to Dr. Ngoc Dong Quan Nguyen and Dr. Lizhen Lin for introducing me to mathematics and statistics, which shaped the foundation of my academic path. Their intellectual guidance and encouragement were crucial in inspiring my interest in mathematics. I am also thankful for their kindness and generosity during the COVID-19 pandemic, when they provided me with accommodation while I awaited the documents needed to come to Canada. Their support during that uncertain time meant a great deal to me, both personally and academically, and I will always remember it with deep appreciation.

I also appreciate the financial support from the Department of Applied Mathematics at the University of Waterloo. This assistance allowed me to devote myself fully to my research, for which I am extremely grateful.

I am grateful to Duy Nguyen, my colleague in the scientific computing lab during my first two PhD years. His advice on computational skills and guidance on academic and personal matters were invaluable in the early stages of my studies. I especially value that our collegial relationship has developed into a lasting friendship.

I would like to thank my office mate, friend, and colleague, Thanin (Juju) Quartz. Our discussions on coursework and mathematical problems, especially in AMATH 740 Numerical Analysis, AMATH 751 Applied Functional Analysis, and AMATH 741 Numerical Partial Differential Equations, gave me valuable insight and strengthened my understanding of key topics in applied mathematics.

I am sincerely grateful to Dr. Phillip Gabriel and Dr. Cuong Nguyen at the National Research Council of Canada in Ottawa, Ontario. Through our project on retrieving liquid water path using a microwave radiative transfer model and radar reflectivity measurements, they introduced me to Bayesian inverse problems in atmospheric science and helped sharpen my research and computational skills.

Last but not least, I would like to express my gratitude to everyone who has helped me directly or indirectly in preparing this thesis. Your impact, in terms of intellectual stimulation and moral support, has been highly valued.

This work is a tribute to the encouragement, guidance, and support of several individuals to whom I remain grateful for their contributions to my academic and personal development.

Dedication

This thesis is dedicated to my family, ***Tan Anh Kiet Le***, ***Ngoc Tram Huong Nguyen***, and ***Tan Hai Dong Le***, for their endless love, sacrifice, and steadfast support, which have made this journey possible. Their belief in me has been a constant source of strength and inspiration.

To my fiancée, ***Thi Minh Thu Tran***, for her love, patience, encouragement, and understanding throughout the many challenges and joys of this journey. Her presence has been a source of comfort, motivation, and happiness.

To all my teachers and friends, whose wisdom, mentorship, and commitment to education have profoundly influenced my intellectual development and inspired in me a lasting appreciation for learning and discovery.

Table of Contents

Examining Committee Membership ii

Author's Declaration iii

Statement of Contributions iv

Abstract v

Acknowledgements vi

Dedication viii

List of Figures xiii

List of Tables xvi

List of Illustrations xvii

List of Abbreviations xviii

List of Symbols xx

Statement of Positionality xxi

Quote xxii

1 Introduction **1**

1.1 Partial Differential Equations and Free-Boundary Value Problems 1

1.2 Radial Basis Function and Random Features Models 2

1.3 Variational Principles 6

1.4 Objectives and Contributions 8

1.5 Thesis Overview 8

2 Mathematical Preliminaries **11**

2.1 Linear Elliptic and Free-Boundary Value Problems 11

2.2 Variational Formulation for Elliptic Boundary Value Problems 13

2.3 Variational Formulation for Obstacle-Type Boundary Value Problem . . . 14

2.4 Radial Basis Functions, Random Features and Universal Approximation Properties 16

2.5 Numerical Linear Algebra Method: Truncated Singular Value Decomposition 21

2.6 Discrete Numerical Formulations 22

2.6.1 Discretization with radial basis functions 22

2.6.2 Discretization with random features models 25

2.7 Alternating Direction Method of Multipliers 29

3 Stable Mesh-Free Variational Radial Basis Function Approximation for Elliptic PDEs and Obstacle Problem **32**

3.1 Introduction 33

3.2 Problem Statement 33

3.2.1 Variational Functionals for Boundary Value Problem 33

3.2.2 Radial Basis Function Model 35

3.3 Variational Algorithm for Radial Basis Function Model 39

3.3.1 Algorithm for Linear Elliptic Boundary Value Problem 40

3.3.2 Algorithm for Obstacle-Type Boundary Value Problem 46

3.4 Numerical Experiment . . . 50
3.4.1 Linear Elliptic Boundary Value Problem . . . 53
3.4.2 Obstacle-Type Boundary Value Problem . . . 56
3.5 Robustness: Parameter Sensitivity . . . 68
3.5.1 Sensitivity Analysis of Domain Expansion Factor T . . . 72
3.5.2 Experiments on the Kernel Shape Multiplier Parameter $c(T, \tau)$ . . . 78
3.5.3 Experiments on Truncation Threshold Value τ . . . 79
3.5.4 Experiments on Boundary Penalty Weight β . . . 79
3.5.5 Comparison between the Kernel Functions ϕ . . . 80
3.6 Limitations . . . 90
3.6.1 Discontinuous Obstacle Problem . . . 90
3.6.2 Numerical Rank Efficiency and Computational Time . . . 92
3.7 Summary . . . 93

4 Random-Feature Solvers for High-Dimensional Variational Problems 94
4.1 Introduction . . . 95
4.2 Problem Statement and Motivation . . . 95
4.3 Algorithmic Framework . . . 98
4.3.1 Random-Feature Solver for High-Dimensional Linear Elliptic Boundary Value Problems . . . 98
4.3.2 Random-Feature Solver for High-Dimensional Free-Boundary Problems 101
4.4 Numerical Experiments: High-Dimensional PDEs . . . 106
4.4.1 Numerical experiments for high-dimensional elliptic boundary value problem . . . 108
4.4.2 Numerical experiment for obstacle problems . . . 116
4.5 Effect of the Variance Scale b_{RF}, Expansion Domain Factor T_{RF}, and Boundary Penalty Weight β on Random Feature Approximation . . . 119
4.5.1 Sensitivity to the expansion domain factor T_{RF} . . . 119
4.5.2 Sensitivity to the variance scale b_{RF} . . . 119

4.5.3 Sensitivity to the boundary penalty weight β 120
4.6 Comparison with Kernel Methods 120
4.7 Limitations and Remarks 126

5 Conclusion and Future Work **127**
5.1 Conclusion and Finding Remarks 127
5.2 Further Directions 129

References **131**

APPENDICES **143**

A Extendable Tables & Hyper-parameters **144**
A.1 Hyper-parameters for experiments in Chapter 3 144
A.2 Settings for experiments in Chapter 4 146

B Code Notes and Configuration Details **148**
B.1 Reproducibility in the proposed radial basis function method 148
B.2 Reproducibility in the random features method 149

Glossary **150**

List of Figures

3.1 Comparison of different methods for the one-dimensional Poisson equation $-\Delta u = f(x)$. 57

3.2 Comparison of different numerical methods for the one-dimensional reaction-diffusion equation $-\Delta u + u = f(x)$. 58

3.3 Radial basis function approximation of the one-dimensional obstacle problem for a single-bump ψ. 62

3.4 Radial basis function approximation of the one-dimensional obstacle problem for a two-bump ψ. 63

3.5 Comparison for the one-dimensional obstacle problems. 64

3.6 Setup for the two-dimensional dome obstacle problem. 65

3.7 Meshfree radial basis function approximation of a radially symmetric obstacle problem. 66

3.8 Comparison of different numerical methods for the two-dimensional dome obstacle problem. 67

3.9 Comparison of the expansion domain factor T in one-dimensional Poisson and reaction-diffusion equation. 69

3.10 Comparison of the expansion domain factor T for the one-dimensional obstacle problem under the scaling b. 70

3.11 Comparison of the expansion domain factor T for the radially symmetric obstacle problem. 71

3.12 Effect of the kernel shape multiplier $c(T, \tau)$ in the Poisson equation. 73

3.13 Effect of the kernel shape multiplier $c(T, \tau)$ in the reaction-diffusion equation. 74

3.14 Effect of the kernel shape multiplier $c(T, \tau)$ to the single-bump obstacle problem. 75
3.15 Effect of the kernel shape multiplier $c(T, \tau)$ to the two bump obstacle problem. 76
3.16 Effect of the kernel shape multiplier $c(T, \tau)$ for the radially symmetric obstacle problem. 77
3.17 Effect of the truncation threshold value τ on the Poisson boundary value problem. 81
3.18 Effect of the truncation threshold value τ on the one-dimensional reaction-diffusion problem. 82
3.19 Effect of the truncation threshold value τ on the one-dimensional single-bump obstacle problem. 83
3.20 Sensitivity to the truncation threshold value τ for the one-dimensional two-bump obstacle problem. 84
3.21 Sensitivity to the truncation threshold value τ for the dome obstacle problem. 85
3.22 Influence of the penalty parameter β in the boundary condition for the Poisson boundary value problem. 86
3.23 Influence of the boundary penalty weight β on the radially symmetric obstacle problem. 87
3.24 Comparison of the radial kernels for the one-dimensional Poisson and reaction-diffusion problems. 88
3.25 Comparison of the radial kernels for obstacle problems. 89
3.26 Radial basis function approximation for the two-dimensional obstacle problem with a discontinuous obstacle function using the variational method. . 91
3.27 Rank efficiency and computational cost in the one-dimensional Poisson problem. 92

4.1 Poisson problem in Eq. (4.25). 108
4.2 Result of Poisson's Equation in Experiment (4.25). 109
4.3 Random feature approximation for high-dimensional partial differential equation in Experiment (4.27). 112
4.4 Comparison of numerical solutions in the high-dimensional partial differential equation in Experiment (4.27) 113

4.5 Exact solution and sampling points in the Darcy's Flow problem in Experiment (4.29). 114
4.6 Random feature approximation in the Darcy's Flow Problem in Experiment (4.29). 115
4.7 Two-dimensional obstacle problem in Experiment (4.31). 117
4.8 Random feature approximation in the two-dimensional obstacle problem. . 118
4.9 Sensitivity of the expansion domain factor T_{RF}. 122
4.10 Sensitivity of the variance scale b_{RF}. 123
4.11 Sensitivity of the boundary penalty weight β. 124
4.12 Comparison between the kernel-features approximation methods for solving the variational partial differential equations. 125

List of Tables

2.1 Examples of widely used radial basis functions. 17

2.2 Examples of activation function employed in the random feature models. . 20

3.1 Summary of the parameters used in the numerical implementation of the variational radial basis function algorithm. 51

4.1 Summary of the parameters used in the numerical experiments for the variational random-features algorithm. 107

A.1 Summary of sensitivity experiments for the expansion domain factor T. . . 144

A.2 Summary of sensitivity experiments for the kernel shape multiplier $c(T, \tau)$. 145

A.3 Summary of sensitivity experiments for the truncation threshold τ. 145

A.4 Summary of sensitivity experiments for the boundary penalty parameter β. 146

A.5 Summary of sensitivity experiments for the basis/kernel function ϕ. 146

A.6 Hyperparameter settings in Chapter 4. 147

List of Illustrations

1.1 Architecture of d-dimensional radial basis function model illustrated in the form of a shallow neural network. 4

1.2 Architecture of random feature model illustrated in the form of a shallow neural network. 5

2.1 Schematic representation of a radial basis function network. 17

2.2 Graphical representation of a random feature model 19

List of Abbreviations

ADMM alternating direction method of multipliers 7, 8, 11, 28–31, 33, 46, 47, 49, 51, 52, 56, 59, 60, 66, 90, 91, 93, 95, 98, 102, 103, 105, 107, 117, 126, 127, 146, 147

BVP boundary value problem 2, 3, 6–9, 11, 12, 14, 20, 21, 26–28, 33–35, 39, 40, 43, 44, 46, 48, 50, 52–56, 90, 93, 95, 97, 98, 106, 108, 110, 114, 116, 120, 125–128

DRM deep Ritz method 106, 110, 111

FDM finite difference method 1, 2, 52, 59, 61

FEM finite element method 1, 2

GPU graphics processing unit 126

IMQ inverse multiquadric 3, 80

IQ inverse quadratic 3, 80

KF kernel feature 2, 120, 130

MQ multiquadric 3, 80

PDE partial differential equation 1–3, 6–9, 11, 14, 16, 21, 22, 33, 35, 39–42, 44–46, 48, 50, 52, 53, 55, 56, 61, 72, 78–80, 92, 93, 95, 97, 98, 100, 101, 103, 106, 108, 110, 114, 120, 125, 127–130

PGD proximal gradient descent 52, 59

PINN physics-informed neural network 52, 59, 106, 110, 111

RBF radial basis function 1–3, 7–9, 11, 13, 16, 18, 24, 33, 35–46, 49, 50, 52–56, 59–61, 68, 72, 79, 81, 90–93, 95, 101, 120, 121, 125–129, 148

RFM random feature model 3, 7–9, 11, 13, 16, 18, 19, 27, 28, 95–98, 100–102, 105, 106, 110, 111, 115–121, 125–129, 146

RKHS reproducing kernel Hilbert spaces 18, 19, 38

SVD singular value decomposition 11, 21, 22, 44, 45, 101

TSVD truncated singular value decomposition 8, 21, 22, 33, 38, 39, 44, 45, 47, 49, 54, 68, 79, 92, 93, 100, 101, 103, 107, 126, 127, 147

List of Symbols

$\vec{a}, \cdots, \vec{z}$	Column vectors
$A, \cdots, Z$	Matrices
$A^{-1}, \cdots, Z^{-1}$	Inverse of matrices
$A^{\dagger}, \cdots, Z^{\dagger}$	Pseudo-inverse of matrices
I_n	Identity matrix of size $n \times n$
$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers
$\|\cdot\|$	General norm
$\|\vec{x}\|_0$	ℓ_0 norm of $\vec{x}$ defined by the number of non-zero components
$\|\vec{x}\|_1$	ℓ_1 norm of $\vec{x}$ defined by $\sum_{i=1}^{d} \|x_i\|$ where $\vec{x} = [x_1, \cdots, x_d]^\top$
$\|\vec{x}\|_2$	ℓ_2 norm of $\vec{x}$ defined by $\left(\sum_{i=1}^{d} \|x_i\|^2\right)^{1/2}$ where $\vec{x} = [x_1, \cdots, x_d]^\top$
$\|\vec{x}\|_\infty$	ℓ_∞ norm of $\vec{x}$ defined by $\max_{1 \leq i \leq d} \{\|x_i\|\}$ where $\vec{x} = [x_1, \cdots, x_d]^\top$
$\text{card}(\cdot)$	Cardinality of the set i.e the number of elements in a set
Ω	Open bounded domain in $\mathbb{R}^d$
$\partial\Omega$	Boundary of the domain Ω
$\overline{\Omega}$	Closure of Ω, i.e., $\Omega \cup \partial\Omega$
$\overline{\Omega}_T$	Domain in $\mathbb{R}^d$ extended by factor $T \geq 1$
∇u	Gradient of the function u
Δu	Laplacian of u, i.e., $\nabla \cdot \nabla u$
$\mathcal{L}[u]$	Linear Differential operator acting on u
$\mathcal{B}[u]$	Boundary operator applied to u

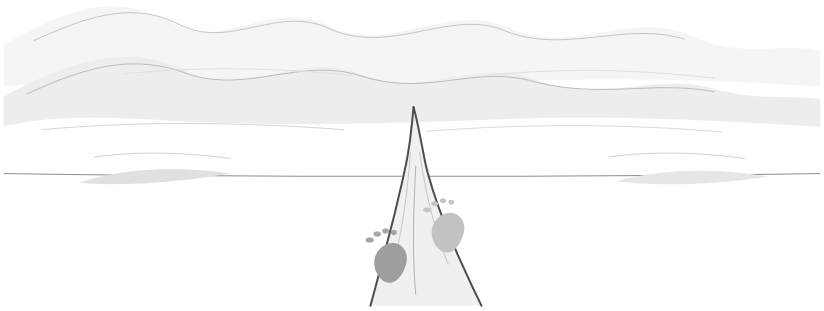
Statement of Positionality

My positionality is informed by formal training in applied mathematics. My educational background is summarized as follows:

- **September 2020 – April 2026:** Doctor of Philosophy in Applied Mathematics, *University of Waterloo*, Canada.
- **August 2016 – May 2020:** Bachelor of Science in Mathematics and Statistics, *Indiana University Bloomington*, United States of America.

This background informs the way I formulate research questions, design methodologies, and implement mathematical and computational methods. In particular, my training has influenced my perspective through a focus on machine learning, numerical partial differential equations, and scientific computing.

Quote



Vạn lý trường đồ, khởi ư nhất bộ.

A journey of ten thousand miles begins with a single step.

— Lão Tử (Lao Tzu) —

Chapter 1

Introduction

Partial differential equations (PDEs) play a significant role in the mathematical modeling of biological, physical, and engineering systems. PDEs arise in the study of diffusion processes, elasticity, heat, and phase transitions. They present significant challenges for numerical approximation and motivate the development of robust computational methods. In recent decades, radial basis function (RBF) methods have been developed as powerful tools in the area of numerical analysis, especially in the solution of PDEs in mesh-free settings [17, 35, 53, 76]. Their ability to approximate functions, coupled with their flexibility in high-dimensional situations, indicates that RBF-based solvers are in use across different types of PDEs in [53, 5, 61]. In this thesis, we develop and investigate mesh-free variational solvers in the form of RBF networks for linear elliptic PDEs and obstacle problems. We intend to formulate PDEs as optimization problems and solve them using numerical linear algebra methods for linear elliptic PDEs combined with proximal methods in obstacle-type PDEs.

1.1 Partial Differential Equations and Free-Boundary Value Problems

Linear elliptic PDEs [32, 81, 82] arise in many applications, including steady-state heat conduction, diffusion, and electrostatics. Elliptic PDEs can be formulated using variational forms and there are several classical methods to approximate. Those approaches include the finite difference method (FDM) and the finite element method (FEM), which

have been widely used for those problems and provide a strong foundation for computational analysis. Boundary value problems (BVPs) form an important problem class for PDEs in which the initial and boundary conditions are specified. Obstacle problems are a special kind of variational inequality that consist of the requirement that the solution lies above a prescribed obstacle function $\psi(x)$. We refer readers to [38, 107, 93] for a detailed understanding. Such problems arise in a broad variety of applications, including fluid mechanics, elasticity, financial mathematics, and phase transition modeling. The free-boundary aspect arises naturally since the location where the solution touches the obstacle is unknown beforehand and must be determined as part of the solution [38, 93]. According to [107], classical approaches to tackle such problems include FEM and FDM. These methods, including FDM and FEM, provide a solid theoretical foundation and well-established convergence results; however, they typically require domain meshing and, in many cases, adaptive mesh refinement near the contact region boundary. Furthermore, enforcing inequality constraints can add algorithmic complexity and computational cost.

Those limitations motivate the exploration of mesh-free approximation methods for free-boundary value problems. Kernel feature (KF) methods offer several advantages, including geometric flexibility, ease of extension to high-dimensional settings, and compatibility with variational formulations. We combine the KF method in the approximation space with appropriate optimization algorithm choices to incorporate the inequality constraints of obstacle-type BVPs.

In this thesis, we address linear elliptic PDEs and obstacle-type problems using optimization and variational frameworks built on the KF method. The variational inequalities are reformulated as minimization problems, allowing numerical optimization techniques to be employed. This approach provides a systematic method for the development of stable mesh-free methods that can handle a broad class of PDEs, such as Poisson and reaction-diffusion equations, and elastic obstacle problems.

1.2 Radial Basis Function and Random Features Models

RBF models provide an effective framework for mesh-free approximation and form a key building block of the methods developed in this thesis. An illustration of the RBF model is given in Figure 1.1. We refer the reader to [79, 96, 17] for a detailed understanding of the approach. RBF models offer an approach to construct approximate solutions that do not rely on domain meshing, making them suitable for BVPs with complex geometries and

inequality constraints. In the context of linear elliptic and obstacle-type BVPs, the solution is approximated as a linear combination of radial basis functions centered at selected points in the expanded domain. Let $\phi(r)$ denote a radial basis function, where $r = \|\vec{x} - \vec{c}_i\|_2$ is the Euclidean distance between a spatial point $\vec{x} \in \Omega$ and a center $\vec{c}_i$. A well-known choice of basis function is the Gaussian RBF, defined as follows.

$$\phi(r) = \exp(-(b\,r)^2),$$

where the shape parameter $b > 0$ controls the flatness of the basis. Other commonly used radial basis kernels include multiquadric (MQ), inverse multiquadric (IMQ), inverse quadratic (IQ), and compactly supported Wendland functions, as described in [34]. However, the matrices generated from these kernels suffer from severe ill-conditioning, particularly in the flat limit. This topic has been discussed in [5], and it becomes an issue in practical RBF computations, which motivates the stabilization strategies developed in this thesis.

RBF methods require a computational cost that scales with the dimension of the PDEs, and dense kernel matrices are expensive to form and manipulate. The random feature model (RFM), as described in [84, 67], provides an alternative that retains the advantages of kernel methods while reducing computational complexity. The RFM approximation is a linear combination of randomly generated features. This leads to low-rank approximations in which the computational cost is scalable with the number of features rather than the dimension, making RFM well-suited for high-dimensional PDEs. Illustrations of the architecture for RBF and RFM are introduced in Illustrations 1.1 and 1.2.

Overall, from a conceptual point of view, the RBF and RFM models occupy complementary roles within the framework of this thesis. The RBF network offers a high accuracy approximation most suitable for low and moderate dimensional settings, while RFM provides a scalable alternative for high-dimensional PDEs. Both approaches share a common variational principle and optimization strategy, enabling practical algorithms for linear elliptic and obstacle-type BVPs.

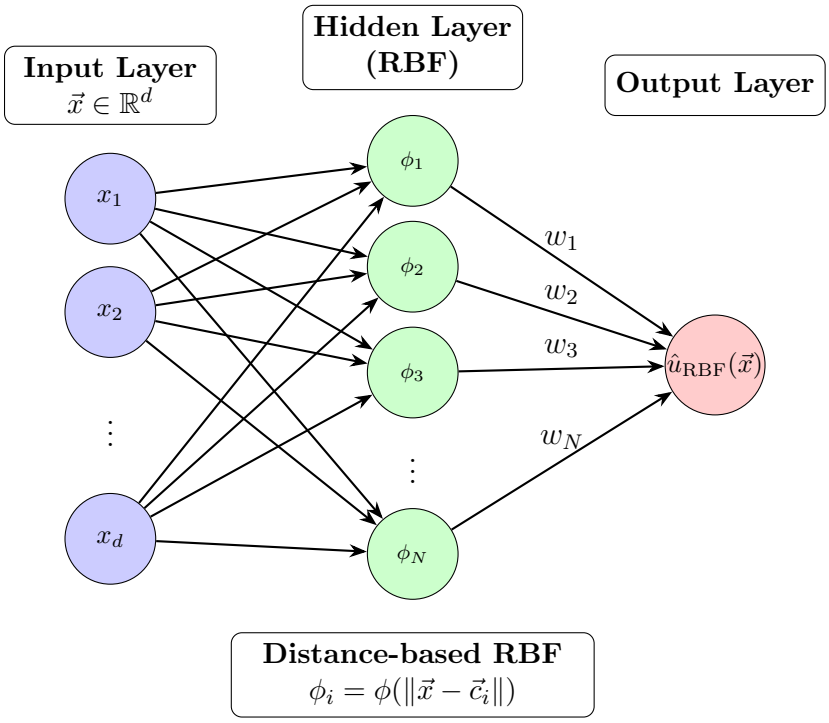


Illustration 1.1: **Architecture of d-dimensional RBF model illustrated in the form of a shallow neural network:** The input layer contains the variables $(x_1, x_2, \ldots, x_d)^\top$, which are mapped to N radial basis functions $\phi\left(\|\vec{x} - \vec{c}_i\|\right)$ for $1 \leq i \leq N$. The output layer then combines these responses linearly using the learnable coefficients w_i for $1 \leq i \leq N$ to produce the approximation $\hat{u}_{\mathrm{RBF}}(\vec{x}) = \sum_{i=1}^{N} w_i \, \phi(\|\vec{x} - \vec{c}_i\|)$

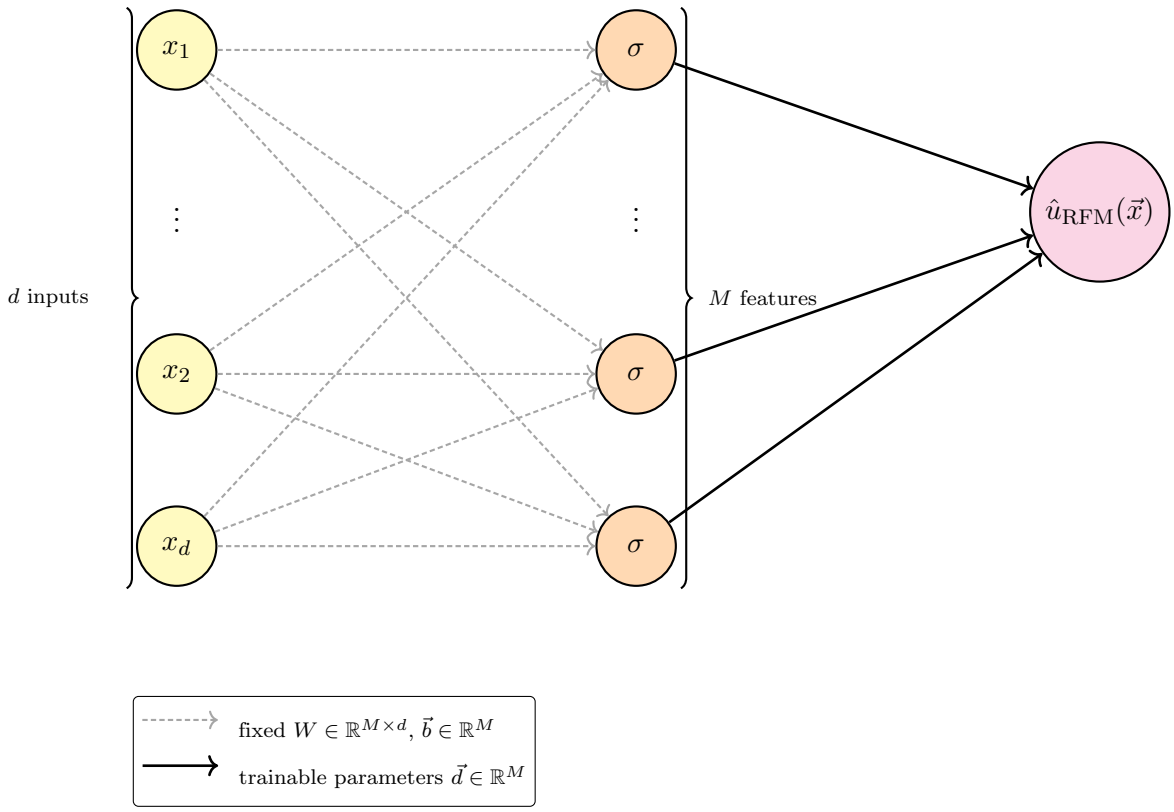


Illustration 1.2: **Architecture of RFM model illustrated in the form of a shallow neural network.** The input vector $\vec{x} \in \mathbb{R}^d$ is mapped through fixed random weights $(W, \vec{b})$ into M nonlinear random features $\{z_i\}_{i=1}^M$, each computed as $z_i = \sigma(\vec{\omega}_i^\top \vec{x} + b_i)$. These features form the hidden layer of the network. A linear readout with trainable coefficients $\vec{d} \in \mathbb{R}^M$ then produces the output $\hat{u}_{\mathrm{RFM}}(\vec{x}) = \vec{d}^\top \vec{z}$.

1.3 Variational Principles

Linear elliptic PDEs are ubiquitous in the mathematical modeling of steady-state phenomena in science and engineering systems, including diffusion processes, elasticity, and financial mathematics [32, 81, 82]. Some applications are subject not only to initial/boundary conditions but also to inequality constraints that reflect physical requirements. A prominent example is obstacle-type BVPs, in which the solution is constrained to remain above a given function [38, 107, 93]. Such problems involve free-boundary regions, which must be determined as part of the solution. The presence of inequality constraints and free boundaries complicates the analysis and numerical approximation of BVPs.

Complementary to the traditional strong formulations of PDE problems, variational formulations of elliptic and obstacle-type boundary value problems provide an alternative concept [38, 32, 122]. Rather than enforcing the governing equation in strong form, the variational framework characterizes the solution as a minimizer of an energy functional over an approximation space; for obstacle problems, this perspective incorporates inequality constraints directly leading to variational inequalities. Furthermore, variational formulations allow for flexibility in the choice of approximation spaces, which is beneficial in mesh-free domains and high-dimensional settings.

Consider the following Poisson equation:

$$\begin{cases} -\Delta u(\vec{x}) = f(\vec{x}), & \forall \vec{x} \in \Omega \subseteq \mathbb{R}^d \\ u(\vec{x}) = g(\vec{x}), & \vec{x} \in \partial\Omega. \end{cases} \tag{1.1}$$

According to the Dirichlet principle, the solution of Eq. (1.1) is a solution of the following optimization problem

$$\min_u J(u), \quad \text{where} \quad J(u) = \frac{1}{2}\int_\Omega |\nabla u|^2 \, d\vec{x} - \int_\Omega f u \, d\vec{x} \quad \text{such that } u(\vec{x}) = g(\vec{x}) \quad \text{on } \partial\Omega. \tag{1.2}$$

The functional $J(u)$ represents the total energy of the system in Eq. (1.1) for the Poisson boundary value problem. It balances a quadratic energy term associated with the gradient of the solution and a given forcing term induced by the source function f in Ω. The minimizer of this functional coincides with the weak solution of the corresponding elliptic PDE as described in [122].

For obstacle problems, the governing equations require additional inequality constraints that the solution must satisfy. Specifically, we impose the solution u within the closed and convex set.

$$\mathcal{C} = \{u \in H_0^1(\Omega) \, : \, u(\vec{x}) \geq \psi(\vec{x}) \quad \text{in } \Omega\},$$

where ψ is a given functional obstacle. The constrained minimization problem leads to a variational inequality that encodes both the governing PDEs and the obstacle constraint. The weak formulation can be expressed as a complementary problem [93]:

$$-\Delta u \geq f, \qquad u \geq \psi, \qquad (-\Delta u - f)(u - \psi) = 0 \quad \text{in} \quad \Omega; \quad u = g \quad \text{on } \partial\Omega. \tag{1.3}$$

By casting BVPs as an optimization problem, it becomes possible to leverage numerical optimization techniques. In particular, the convex structure of the energy functional and the constraint set allow for the use of efficient splitting-based algorithms, such as proximal methods and the alternating direction method of multipliers (ADMM) [15, 107, 6]. The ADMM is a classical operator-splitting method for constrained optimization problems. In this thesis, we do not introduce ADMM as a new algorithm; however, we use it as a standard numerical tool to solve the non-smooth constrained optimization problems arising from the obstacle-type BVPs.

In general, the variational framework both characterizes elliptic and obstacle-type BVPs and provides a foundation for designing stable numerical solvers [107, 123, 68]. The methods proposed in this thesis are constructed on this foundation to address challenges, including the condition number of matrix systems, non-smooth inequality constraints in the PDEs, and high-dimensional settings, while maintaining high accuracy and computational efficiency. A central reason for adopting the variational formulation is that it reduces the regularity requirements on the solution and allows the differential equation to be encoded through energy minimization principles. This is particularly useful for obstacle-type BVPs, where classical differentiability may fail near the contact region or free boundary. In contrast, strong-form residual minimization requires pointwise evaluation of differential operators and may be less robust when the solution has limited smoothness. The variational formulation also provides an approach to penalty methods for boundary conditions and convex optimization formulations for the finite-dimensional RBF and RFM approximations considered in this thesis.

A key idea in this thesis is to formulate the corresponding linear elliptic and obstacle-type BVPs as a minimization problem. We use penalty methods to incorporate Dirichlet boundary conditions and inequality constraints [15, 107, 6]. Linear elliptic PDEs lead to smooth quadratic minimization, while obstacle-types BVPs introduce non-smooth constraints, for which we use proximal and operator splitting methods, including ADMM [15, 107].

1.4 Objectives and Contributions

The primary objective of this thesis is to develop stable, accurate, and computationally efficient mesh-free numerical methods for solving linear elliptic and obstacle-type BVPs, which are reformulated into variational form. In particular, the thesis aims to address the challenges of ill-conditioning, non-smooth inequality constraints, and high-dimensional settings that arise in kernel-based discretizations. Consequently, the thesis seeks to provide systematic algorithmic insights about accuracy, stability, and efficiency for variational solvers. The main contributions of this thesis are summarized as follows:

- We develop a stabilized variational formulation based on the RBF approximation to solve linear elliptic and obstacle-type BVPs. The approach enables mesh-free discretization and flexible handling of boundary conditions and inequality constraints. The truncated singular value decomposition [2, 4, 5, 43] is used to solve linear algebra systems to reduce the ill-conditioning resulting from the kernel matrix generated by the RBF. The experiments study the trade-off between the choice of a truncation threshold value and computational error. Obstacle inequality constraints are treated using a ℓ_1 penalty according to [38, 107]. This leads to non-smooth optimization problems that are efficiently solved using the ADMM algorithm.
- We use the RFM model with the variational principle and combine optimization methods to solve high-dimensional PDEs. We validate the proposed RFM method using analytical solutions as references to compare the accuracy and convergence order with competing approaches. With low-rank RFM as in [84, 85, 86, 67], we construct scalable approximations in which the computational cost grows with the number of feature functions. Through various numerical experiments, we show that the proposed kernel-feature-based approaches retain an accuracy level on par with competing numerical methods, while providing better scalability and robustness in high-dimensional settings.

1.5 Thesis Overview

The outline of the remaining chapters is presented below.

Chapter 2. We recall mathematical preliminaries, which include some important definitions and concepts from obstacle-type BVPs, RBF and RFM models, truncated singular value decomposition (TSVD), and numerical optimization techniques. These are key

to understanding the content of the thesis. We then provide a brief definition of variational formulation, boundary conditions, and penalty enforcement under the constraints of obstacle-type problems. In addition, we discuss numerical discrete formulations in the variational framework. These are foundational to the presented studies of variational mesh-free methods and their application to high-dimensional PDE methods.

Chapter 3. We investigate variational RBF models to approximate linear elliptic PDEs and obstacle-type BVPs, emphasizing numerical accuracy, robustness, and efficiency. This chapter addresses the ill-conditioning of dense RBF discretizations by using truncated singular value decomposition. We examine the resulting stability and accuracy trade-offs. Numerical experiments are presented to evaluate performance and illustrate convergence behavior. The approximation and truncation errors are explored for practical choices of the number of basis functions N, truncation threshold τ, and domain expansion factor T. The empirical results demonstrate that the variational RBF method provides an accuracy that is competitive with existing methods. Proximal algorithms are employed to handle non-smooth ℓ_1 constraints, followed by an examination of the effects of parameters such as penalty weights, threshold values, the number of basis functions, and the oversampling ratio. The contents of this chapter have been adapted and modified from the following article:

Tan Phuong Dong Le, Giang Tran and Hans De Sterck. Stable Mesh-free Variational Radial Basis Function Approximation for Elliptic PDEs and Obstacle Problems. *arXiv preprint arXiv:2604.19882*, 2026.

Chapter 4. In this chapter, we present a framework for solving high-dimensional PDEs using RFM methods within a variational formulation. The proposed approach combines the advantages of random features with a principled variational formulation, leading to a convex quadratic optimization problem. The method is developed for high-dimensional settings, and we discuss in detail how the overall accuracy of the method depends on its main components. Numerical experiments are presented to demonstrate the performance of the method. We discuss a comparative study of the proposed variational solver. This chapter compares RBF and RFM models, as well as neural networks, in terms of accuracy and computational cost. A robustness analysis concludes the chapter by assessing sensitivity to parameter variations. The contents of this chapter have been modified and adjusted from the following article:

Tan Phuong Dong Le, Giang Tran and Hans De Sterck. Random-Feature Solvers for High-Dimensional Variational Problems: A Practical Framework. Preprint. 2026. Available as manuscript.

Chapter 5. We summarize the main findings of the thesis and highlight the remaining

challenges and future research directions that arise from this work.

Chapter 2

Mathematical Preliminaries

This chapter provides the foundations required for the development of RBF and RFM approximations within the variational framework for PDEs. In Section 2.1, we begin with a general introduction to BVPs and Sobolev spaces as functional settings for the solution of PDEs. Sections 2.2-2.3 formulate the BVPs into the variational formulations. In Section 2.4, we review the basic construction of RBF, RFM, and their universal approximation properties. In Section 2.5, we briefly review the numerical linear algebra tools, particularly singular value decomposition (SVD), and its truncated variant, which are presented as stabilization methods for solving ill-conditioned systems. Section 2.6 defines our BVPs and the obstacle problems to construct the variational form in the next chapter. Lastly, the details of ADMM are provided in Section 2.7.

2.1 Linear Elliptic and Free-Boundary Value Problems

We first define the linear elliptic PDEs and obstacle-type BVPs [32]:

Definition 1 (Boundary Value Problem [32]) *Let $\Omega \subset \mathbb{R}^d$ be a bounded open domain with its boundary $\partial\Omega$. The closure of the domain is denoted by $\overline{\Omega} = \Omega \cup \partial\Omega$. We seek an unknown function $u : \bar{\Omega} \to \mathbb{R}$ satisfying*

$$\begin{cases} \mathcal{L}[u](\vec{x}) = f(\vec{x}), & \forall \vec{x} \in \Omega, \\ \mathcal{B}[u](\vec{x}) = g(\vec{x}), & \vec{x} \in \partial\Omega. \end{cases} \tag{2.1}$$

where $\mathcal{L}$ is a differential operator (e.g., Laplacian Δ), $f \in L^2(\Omega)$ is a forcing function, $\mathcal{B}$ is a boundary operator encoding Dirichlet, Neumann or Robin-type conditions, and $g \in C^1(\partial\Omega)$ prescribes the boundary data.

Definition 2 (Linear Elliptic Boundary Value Problem) *We maintain the notation from Definition 1. The linear elliptic boundary value problem with Dirichlet boundary conditions is given by:*

$$\begin{cases} \mathcal{L}[u](\vec{x}) = f(\vec{x}), & \forall \vec{x} \in \Omega, \\ u(\vec{x}) = g(\vec{x}), & \vec{x} \in \partial\Omega, \end{cases} \tag{2.2}$$

where $\mathcal{L}$ is a linear elliptic differential operator that acts on the unknown function u with smooth coefficients.

For instance, one of the well-known linear elliptic BVPs is the Poisson equation

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases}$$

Definition 3 (Obstacle-Type Boundary Value Problem [93]) *Given an obstacle function $\psi : \Omega \to \mathbb{R}$, the Euler-Lagrange equations for this type of free-BVPs are given by:*

$$\begin{cases} -\Delta u \geq 0, & \forall \vec{x} \in \Omega, \\ u(\vec{x}) \geq \psi(\vec{x}), & \forall \vec{x} \in \Omega, \\ (-\Delta u)\,(u - \psi) = 0, & in \quad \Omega, \\ u(\vec{x}) = g(\vec{x}), & on \quad \partial\Omega, \end{cases} \tag{2.3}$$

When seeking weak solutions of the BVPs described in Definitions 2 and 3, one must require an appropriate function space that extends classical differentiability. This is provided by Sobolev spaces, as outlined below.

Definition 4 (Sobolev Space [32]) *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain and $1 \leq p < \infty$. The Sobolev space $W^{k,p}(\Omega)$ is defined as follows:*

$$W^{k,p}(\Omega) \coloneqq \left\{ u \in L^p(\Omega) \,|\quad D^\alpha u \in L^p(\Omega), \forall \alpha \in \mathbb{N}_0^d \text{ with } |\alpha| \leq k \right\}, \tag{2.4}$$

where $\alpha = (\alpha_1, \ldots, \alpha_d)$ *is a multi-index,*

$$|\alpha| = \sum_{m=1}^{d} \alpha_m, \quad \text{and} \quad D^\alpha u = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}$$

denotes the weak derivative of u*.*

A fundamental result in [32] states that $C^\infty(\Omega) \cap W^{k,p}(\Omega)$ is dense in $W^{k,p}(\Omega)$, ensuring that Sobolev functions can be approximated by smooth functions.

Theorem 1 (Global approximation by smooth functions [32]) *Let* $\Omega \subset \mathbb{R}^d$ *be a bounded domain and* $u \in W^{k,p}(\Omega)$ *for some* $1 \leq p < \infty$*. Then there exist functions* $u_m \in C^\infty(\Omega) \cap W^{k,p}(\Omega)$ *such that:*

$$u_m \to u \quad \text{in} \quad W^{k,p}(\Omega). \tag{2.5}$$

While Theorem 1 guarantees the existence of globally smooth approximations to weakly differentiable functions, practical solver methods require explicit approximations with stability and universal approximation properties. RBF and RFM methods provide parameterized models and offer universal approximation properties. The remainder of this section is devoted to introducing these representations and establishing fundamental approximation properties that are relevant to the variational framework considered in this thesis.

2.2 Variational Formulation for Elliptic Boundary Value Problems

In this section, we review the variational formulation for classical elliptic boundary value problems and introduce a penalty-based approach for enforcing Dirichlet boundary conditions. These formulations form the foundation for the mesh-free optimization framework developed in this work. We begin with the Poisson boundary value problem defined on a bounded domain $\Omega \subset \mathbb{R}^d$ with the boundary $\partial\Omega$ as stated below.

$$\begin{cases} -\Delta u = f(\vec{x}), & \forall \vec{x} \in \Omega, \\ u(\vec{x}) = g(\vec{x}), & \forall \vec{x} \in \partial\Omega, \end{cases} \tag{2.6}$$

where $f : \Omega \to \mathbb{R}$ is a given source term and $g : \partial\Omega \to \mathbb{R}$ prescribes Dirichlet boundary data. The classical variational formulation associated with Eq. (2.6) seeks a function u by minimizing the energy functional

$$\frac{1}{2}\int_{\Omega} \|\nabla u\|_2^2 \, d\vec{x} - \int_{\Omega} f\, u \, d\vec{x}, \quad \text{subject to } u(\vec{x}) = g(\vec{x}) \ \forall \vec{x} \in \partial\Omega. \tag{2.7}$$

While Eq. (2.7) is well-defined in standard Sobolev spaces with essential boundary conditions imposed strongly, it poses practical challenges in mesh-free approximation settings. In particular, constructing trial spaces that exactly satisfy $u = g$ on $\partial\Omega$ can be restrictive and computationally inconvenient. Following the approach of [88, 87, 120], we instead adopt a penalty method to enforce the Dirichlet boundary condition. Specifically, we consider the unconstrained minimization problem as defined in the following definition.

Definition 5 (Variational Formulation for Elliptic PDEs) *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with boundary $\partial\Omega$, and let $f : \Omega \to \mathbb{R}$ and $g : \partial\Omega \to \mathbb{R}$ be given functions. The variational formulation of the Poisson PDE with Dirichlet boundary conditions in Eq. (2.6) is defined as the unconstrained minimization problem*

$$\min_{u} \mathcal{J}_{\beta,\text{poisson}}(u), \tag{2.8}$$

where the penalized energy functional $\mathcal{J}_{\beta,\text{poisson}} : H^1(\Omega) \to \mathbb{R}$ is given by

$$\mathcal{J}_{\beta,\text{poisson}}(u) = \frac{1}{2}\int_{\Omega} \|\nabla u\|_2^2 \, d\vec{x} - \int_{\Omega} f\, u \, d\vec{x} + \frac{\beta}{2}\int_{\partial\Omega} (u-g)^2 \, d\vec{s}, \qquad \beta > 0. \tag{2.9}$$

The boundary penalty term enforces the Dirichlet condition $u = g \quad \text{on } \partial\Omega$, penalized by the parameter β. By an appropriate choice of β, the minimizers of $\mathcal{J}_{\beta,\text{poisson}}$ converge to the solution of the BVP [122, 32, 27, 29]. This formulation is particularly suitable for radial basis function and neural network approximations.

2.3 Variational Formulation for Obstacle-Type Boundary Value Problem

We now extend the variational framework for elliptic boundary value problems to obstacle-type problems, which arise naturally in variational inequalities and free-boundary phenomena. We let $\Omega \subset \mathbb{R}^d$ be a bounded domain with boundary $\partial\Omega$. Given an obstacle function

$\psi : \Omega \to \mathbb{R}$, and Dirichlet boundary function $g : \partial\Omega \to \mathbb{R}$, the classical obstacle problem seeks a function u satisfying

$$\begin{cases} -\Delta u(\vec{x}) \geq 0, & \forall \vec{x} \in \Omega, \\ u(\vec{x}) \geq \psi(\vec{x}), & \forall \vec{x} \in \Omega, \\ u(\vec{x}) = g(\vec{x}), & \forall \vec{x} \in \partial\Omega, \\ (-\Delta u)\,(u - \psi) = 0, & \forall \vec{x} \in \Omega. \end{cases} \tag{2.10}$$

Following the classical theory of variational inequalities [56], this problem is equivalent to minimizing the Dirichlet energy in Eq. (1.2) over the convex admissible set

$$\mathcal{K}_{\text{contact}} := \{u \in H^1(\Omega) : u \geq \psi \text{ in } \Omega,\ u = g \text{ on } \partial\Omega\}.$$

To avoid explicit constraint handling, we adopt an ℓ^1-penalty approach [107] that replaces the inequality constraint $u \geq \psi$ with a non-smooth penalty term, where we define $(\psi - u)_+ := \max\{\psi - u, 0\}$.

This leads to the following unconstrained variational formulation.

Definition 6 (Variational Formulation for Obstacle-Type Problems) *The variational formulation of the obstacle problem in Eq. (2.10) is defined as a minimization problem*

$$\min_u \mathcal{J}_{\mu,\beta,\,\text{obstacle}}(u), \tag{2.11}$$

where the penalized energy functional $\mathcal{J}_{\mu,\beta,\text{obstacle}} : H^1(\Omega) \to \mathbb{R}$ is given by

$$\mathcal{J}_{\mu,\beta,\,\text{obstacle}}(u) = \frac{1}{2}\int_\Omega \|\nabla u\|_2^2\, d\vec{x} + \mu \int_\Omega (\psi - u)_+\, d\vec{x} + \frac{\beta}{2}\int_{\partial\Omega} (u-g)^2\, d\vec{s}, \qquad \mu, \beta > 0. \tag{2.12}$$

For an appropriate choice of parameter μ and an appropriate choice of β, the minimizers of $\mathcal{J}_{\mu,\beta,\text{obstacle}}$ converge to the solution of the classical obstacle problem in Eq. (2.10) according to [91, 18].

This penalty-based formulation transforms the variational inequality into an unconstrained, non-smooth optimization problem, making it well suited for mesh-free approximation using radial basis function networks. The penalty parameters μ and β control the enforcement of the obstacle constraint and the Dirichlet boundary condition, respectively. In the following sections, we describe the discretization of Eqs. (2.9)–(2.12) and the optimization algorithms used to compute their minimizers.

2.4 Radial Basis Functions, Random Features and Universal Approximation Properties

The RBF and RFM models define mesh-free approximation spaces that are suitable for variational formulations of PDEs. In this section, we discuss approximations using radial basis and random feature functions. We give a brief overview of these kernel methods and their advantages. For a detailed understanding, we refer the reader to [61, 60]. First, we define the RBF model as below.

Definition 7 (Radial Basis Function Approximation Model) *Given $u : \overline{\Omega} \to \mathbb{R}$, where $\Omega \subseteq \mathbb{R}^d$. Let $\phi : [0, \infty) \to \mathbb{R}$ be a radial basis function and N be the number of radial basis functions. Assume that the radial basis centers $\vec{c}_1, \ldots, \vec{c}_N$ are sampled independently from an expanded domain $\overline{\Omega}_T$, where $T \geq 1$ is the expansion domain factor, which is discussed in more detail in Chapter 3. We then define the radial basis function approximation $\hat{u}_{\mathrm{RBF}} : \overline{\Omega} \to \mathbb{R}$ to the function u by*

$$\hat{u}_{\mathrm{RBF}}(\vec{x}) := \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i \, \phi(\|\vec{x} - \vec{c}_i\|_2), \tag{2.13}$$

where the weight vector $\vec{w} = (w_1, \ldots, w_N)^\top \in \mathbb{R}^N$ is to be learned. We note that the scaling factor $\frac{1}{\sqrt{N}}$ is to normalize the representation with respect to the number of centers. The graphical schematic representation is shown in Illustration 2.1.

Definition 8 (Radial Basis Functions, see [34]) *A functional $\Phi : \mathbb{R}^d \to \mathbb{R}$ is called radial kernel centered at $\vec{c} \in \mathbb{R}^d$, if there exists a uni-variate mapping $\phi : [0, \infty) \to \mathbb{R}$ such that*

$$\Phi(\vec{x}) = \phi(r), \quad \text{where} \quad r = \|\vec{x} - \vec{c}\| \tag{2.14}$$

and $\|\cdot\|$ denotes a norm on $\mathbb{R}^d$ - normally we use the Euclidean norm.

The specific form of ϕ determines the smoothness, locality, and conditioning properties of the resulting RBF approximation and affects the empirical performance of the variational method. More details will be discussed in Chapter 3. Some well-known examples of kernel functions are defined in Table 2.1. With regard to the choice of the kernel in Table 2.1, RBF networks possess the universal approximation theorem, ensuring the ability to approximate arbitrary functions in compact domains.

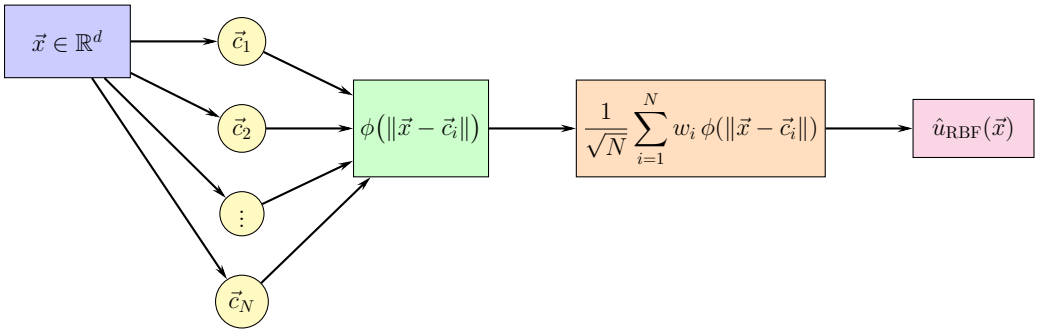


Illustration 2.1: **Schematic representation of a RBF-network in the Definition 7.** A single input vector $\vec{x} \in \mathbb{R}^d$ is evaluated with a set of radial basis centers $\{\vec{c}_1, \ldots, \vec{c}_N\}$ via the kernel functional $\phi(\|\vec{x} - \vec{c}_i\|_2)$ for $1 \leq i \leq N$. The resulting approximation consists of a linear combination of functional values ϕ with the weighted learnable coefficients w_i. Source: [61].

Name	**Formula** $\phi(r)$	**Smoothness**	**Locality**
Gaussian	$e^{-(b_{\mathrm{RBF}}\, r)^2}$	C^∞	Global
Multiquadric	$\sqrt{1 + (b_{\mathrm{RBF}}\, r)^2}$	C^∞	Global
Inverse Quadric	$\frac{1}{1+(b_{\mathrm{RBF}}\, r)^2}$	C^∞	Global
Inverse MQ	$\frac{1}{\sqrt{1+(b_{\mathrm{RBF}}\, r)^2}}$	C^∞	Global
Wendland C^2	$\chi_{[0,1]}(b_{\mathrm{RBF}}\, r)(1 - b_{\mathrm{RBF}}\, r)^4(4 b_{\mathrm{RBF}}\, r + 1)$	C^2	Compact
Wendland C^4	$\chi_{[0,1]}(b_{\mathrm{RBF}}\, r)(1 - b_{\mathrm{RBF}}\, r)^6(35(b_{\mathrm{RBF}}\, r)^2 + 18\, b_{\mathrm{RBF}}\, r + 3)$	C^4	Compact

Table 2.1: Examples of widely used radial basis functions. The parameter $b_{\mathrm{RBF}} > 0$ controls the shape of the kernel. The term $\chi_{[0,1]}(b_{\mathrm{RBF}} r)$ denotes the indicator function, which takes the value 1 if $b_{\mathrm{RBF}}\, r \in [0, 1]$ and 0 otherwise.

Theorem 2 (Universal Approximation Theorem for RBF [79]) *Let $\Omega \subseteq \mathbb{R}^d$ be an open bounded set. We suppose that $\phi : [0, \infty) \to \mathbb{R}$ is continuous, then for every $h \in L^p(\Omega)$ with $p \in [1, \infty)$ and every $\epsilon > 0$, there exists an integer $N \in \mathbb{N}$, radial centers $\vec{c}_1, \ldots, \vec{c}_N$ where each $\vec{c}_i \in \mathbb{R}^d$, weighted coefficients $w_1, \ldots, w_N$, and a scale parameter $b > 0$ such that:*

$$\|\hat{u}_{\text{RBF}} - h\|_{L^p(\Omega)} < \epsilon. \tag{2.15}$$

The RBF approximation typically requires the construction and inversion of dense, ill-conditioned kernel matrices, especially in a high-dimensional setting, which will be discussed later. This limitation motivates the consideration of alternative representations that retain the advantages of kernel methods. One such approach is the RFM methodology, which uses randomized feature maps derived from the spectral representation of positive-definite kernels. In particular, Bochner's theorem in [14] enables the kernel to be expressed over trigonometric functions, leading to finite-dimensional approximations based on random sine and cosine features. We refer the reader to [67] for a detailed exposition. To construct the RFM approximation, we recall Mercer's condition and reproducing kernel Hilbert spaces (RKHS) as a foundation.

Theorem 3 (Mercer's theorem) *Suppose that $K : \Omega \times \Omega \to \mathbb{R}$ is a symmetric function. Then K is a positive semi-definite kernel if and only if*

$$\sum_{i=1}^{m} \sum_{j=1}^{m} a_i \, a_j \, K(\vec{x}^{(i)}, \vec{x}^{(j)}) \geq 0,$$

for all $\vec{x}^{(i)}, \vec{x}^{(j)} \in \Omega \subseteq \mathbb{R}^d$ and real numbers $a_1, \ldots, a_m$.

Definition 9 (Reproducing Kernel Hilbert Space) *Let $K : \Omega \times \Omega \to \mathbb{R}$ be a symmetric and positive semi-definite kernel. Then there exists a Hilbert space $\mathcal{H}$ and a mapping γ from Ω to $\mathcal{H}$ such that*

$$K(\vec{x}, \vec{y}) = \langle \gamma(\vec{x}), \gamma(\vec{y}) \rangle_{\mathcal{H}}, \quad \forall \vec{x}, \vec{y} \in \Omega.$$

Furthermore, $\mathcal{H}$ has the reproducing property given by

$$\forall h \in H, \forall \vec{x} \in \Omega, h(\vec{x}) = \langle h, K(\vec{x}, \cdot) \rangle_{\mathcal{H}}.$$

Then $\mathcal{H}$ is called a reproducing kernel Hilbert space associated with K. The definition implies that the evaluation of h at $\vec{x}$ can be written as,

$$h(\vec{x}) = \sum_{j=1}^{m} a_j \, K(\vec{x}^{(j)}, \vec{x}), \tag{2.16}$$

for some $a_1, \ldots, a_m \in \mathbb{R}$.

The RKHS $\mathcal{H}$ associated with the functional kernel K in Definition 9 is viewed as a feature space induced by the mapping σ. In general, this feature space is not unique. Although kernel methods allow inner products in high-dimensional feature spaces to be computed efficiently, identifying the kernels that yield data separation can still be challenging. To address these limitations, RFM was introduced in [84]. For a detailed understanding of the thesis, we review the random feature framework and related recent developments below.

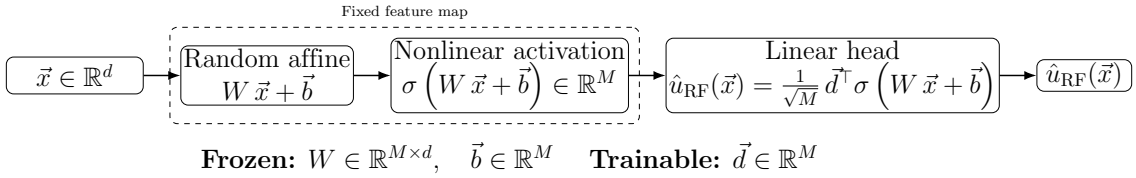


Illustration 2.2: **Graphical representation of a random features model**: fixed feature map $\sigma(W\,\vec{x} + \vec{b})$ and trainable linear head.

Definition 10 (Random Feature Approximation [84, 85, 86])
We let $u : \mathbb{R}^d \to \mathbb{R}$ be a given unknown function. The RFM approximation takes the form

$$\hat{u}_{\mathrm{RF}}(\vec{x}) := \frac{1}{\sqrt{M}}\,\vec{d}^{\top}\,\sigma\Big(W\,\vec{x} + \vec{b}\Big) = \frac{1}{\sqrt{M}}\sum_{i=1}^{M} d_i\,\sigma\Big(\vec{\omega}_i^{\top}\,\vec{x} + b_i\Big), \tag{2.17}$$

where $W \in \mathbb{R}^{M\times d}$ and $\vec{b} \in \mathbb{R}^M$ are chosen at random and $\vec{d} \in \mathbb{R}^M$ is trainable. The graphical schematic representation for RFM is shown in Illustration 2.2.

In Definition 10, the parameters of the matrix $W = [\omega_{j,k}]$ are sampled independently and identically from a probability density function $\rho(\vec{\omega})$, i.e., $\omega_{j,k} \sim \rho(\vec{\omega})$ for all $1 \le j \le M$ and $1 \le k \le d$. Similarly, the components of the bias vector $\vec{b}$ are also sampled from a probability distribution. More precisely, we assume that the components of the random matrix and the bias vector are sampled as follows:

$$\vec{\omega}_j \sim \mathcal{N}(0, b_{\mathrm{RF}}^2\,I_d), \qquad b_j \sim \mathcal{U}(\overline{\Omega}_{T,d}), \quad j = 1, \ldots, M. \tag{2.18}$$

where $\overline{\Omega}_{T,d}$ is an expanded domain that depends on the parameters T_{RF} and d, which will be discussed later in Chapter 4. The scale b_{RF}^2 represents the variance of the normal distribution for the random matrix W in the RFM model. The activation function $\sigma : \mathbb{R}^M \to \mathbb{R}^M$ is a fixed nonlinear function applied element-wise. For the remainder of the thesis, we assume that the activation function is the hyperbolic tangent function $\sigma(z) =$

$\tanh(z)$ for $z \in \mathbb{R}$, unless otherwise stated. There is a variety of activation functions that have been considered in the literature. We refer readers to [27, 64, 65, 66] for a detailed understanding of different random features approximation models. The choice of activation function also influences both the smoothness of the approximation and the locality of the kernel features. For instance, if we use sine and cosine features, the bias vector $\vec{b}$ is initialized using a different probability density function. In that case, we initialize the bias values as shown below.

$$b_j \sim \mathcal{U}[0, 2\pi], \quad j = 1, \ldots, M.$$

The following Table 2.2 summarizes the various activations utilized in the literature.

Name	**Formula**	**Approximation type**
Hyperbolic tangent (tanh)	$\sigma(z) = \tanh(z)$	Global smooth
Logistic sigmoid	$\sigma(z) = \dfrac{1}{1+e^{-z}}$	Global smooth
Softplus	$\sigma(z) = \log(1+e^{z})$	Global smooth
Gaussian	$\sigma(z) = \exp(-z^2)$	Global smooth
Cosine feature	$\sigma(z) = \cos(z)$	Global smooth (periodic)
Sine feature	$\sigma(z) = \sin(z)$	Global smooth (periodic)
ReLU	$\sigma(z) = \max(0, z)$	Piecewise; non-smooth at zero
Leaky ReLU	$\sigma(z) = \max(\alpha z, z)$, $\alpha \in (0,1)$	Piecewise; non-smooth at zero

Table 2.2: Examples of activation function $\sigma(\cdot)$ employed in random feature models.

In the random feature model, a feature refers to a scalar nonlinear function of the input form

$$\sigma\left(\vec{\omega}_j \, \vec{x} + b_j\right),$$

where $\vec{\omega}_j$ is a sampled weight vector and b_j is a sampled bias. The activation function $\sigma(\cdot)$ determines the nonlinear feature used to approximate the unknown solution.

In the next section, we present a concise overview of the numerical linear algebra techniques used to solve least-squares problems. In particular, Section 2.5 provides the theoretical and algorithmic foundations for the methods developed later in the thesis to solve BVPs.

2.5 Numerical Linear Algebra Method: Truncated Singular Value Decomposition

A common problem in numerical linear algebra is to solve the system $A\vec{w} = \vec{y}$ where the data $\vec{y} \in \mathbb{R}^m$ and an unknown vector $\vec{w} \in \mathbb{R}^N$ are connected via a measurement matrix $A \in \mathbb{R}^{m\times N}$. This system can be encountered in many problems, such as solving PDEs according to the results of [38, 40, 107]. Many of the discretization techniques used to solve BVPs give rise to large linear systems. The SVD [43, 47, 108] provides a fundamental tool for analyzing such systems, offering insight on rank, conditioning, and numerical stability. The truncated version of SVD, the TSVD [2, 4], provides a regularization method by discarding small singular values that amplify errors. First, we start by recalling the SVD method, and then we give a brief introduction to the TSVD as follows:

Definition 11 (Singular Value Decomposition, see [43, 108]) *Let A be an $m \times N$ matrix. Let r denote the number of nonzero singular values of A, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$, or equivalently the rank of A. Then a singular value decomposition of A is a factorization*

$$A = U\Sigma V^\top \tag{2.19}$$

where:

- *U is an $m \times m$ orthogonal matrix.*
- *V is an $N \times N$ orthogonal matrix.*
- *Σ is an $m \times N$ matrix whose i^{th} diagonal entry equals the i^{th} singular value σ_i for $i = 1, \ldots, r$. All other entries of Σ are zero.*

Definition 12 (Truncated Singular Value Decomposition, see [2, 4, 5]) *Suppose that the matrix $A \in \mathbb{R}^{m\times N}$ has rank $r \leq \min(m, N)$ and is possibly ill-conditioned. Given a singular value decomposition of A*

$$A = U\Sigma V^\top, \quad U \in \mathbb{R}^{m\times m}, \quad V \in \mathbb{R}^{N\times N}.$$

For any $\tau > 0$, the truncated approximation of A

$$A_\tau := \sum_{\sigma_i > \sigma_1 \tau} \sigma_i \vec{u}_i \, \vec{v}_i^\top = U_p \Sigma_p \, V_p^\top, \tag{2.20}$$

where $\Sigma_p = diag(\sigma_1, \ldots, \sigma_p)$ with $\sigma_p > \sigma_1 \tau \geq \sigma_{p+1}$ and $U_p \in \mathbb{R}^{m\times p}, V_p \in \mathbb{R}^{N\times p}$ contain the first p singular vectors of U and V, respectively.

We introduce the pseudo-inverse of the TSVD as follows:

Definition 13 (TSVD Pseudo-inverse [5]) *Let $A \in \mathbb{R}^{m\times N}$ and $\tau > 0$. Suppose that $A = U\,\Sigma\,V^\top$ is a SVD of A. Then*

$$A_\tau^\dagger := \sum_{\sigma_i \geq \sigma_1\,\tau} \frac{\vec{v}_i\,\vec{u}_i^\top}{\sigma_i} = V_p \Sigma_p^{-1} U_p^\top, \tag{2.21}$$

where $\Sigma_p^{-1} = diag(1/\sigma_1, \ldots, 1/\sigma_p)$ with $\sigma_p > \sigma_1\,\tau \geq \sigma_{p+1}$. The truncated SVD solution of $A\,\vec{w} = \vec{y}$ with the truncation parameter τ is

$$\vec{w}_\tau := A_\tau^\dagger\,\vec{y} = V_p \Sigma_p^{-1} U_p^\top\,\vec{y}. \tag{2.22}$$

The TSVD in Definition 12 offers a means of stabilizing rank-deficient and ill-conditioned systems and provides an approximate solution to the linear system. Our PDE formulation involves additional constraints, such as inequality conditions in obstacle problems. To address the inequality constraints, we require optimization strategies that go beyond just linear solvers. For this reason, in Section 2.7, we consider a flexible algorithmic framework for solving constrained optimization problems.

2.6 Discrete Numerical Formulations

We now derive the discrete optimization problems corresponding to the penalized variational formulations for elliptic and obstacle-type boundary value problems. Let $m = m_\Omega + m_{\partial\Omega}$ denote the total number of collocation points in $\overline{\Omega}$ for approximating the integrals in the variational formulation, where m_Ω and $m_{\partial\Omega}$ are the numbers of interior and boundary points, respectively. These points are used to approximate the interior variational integral and the boundary penalty term, respectively.

2.6.1 Discretization with radial basis functions

We select N uniformly distributed centers $\{\vec{c}_i\}_{i=1}^N \subset \overline{\Omega}_T$, where $\overline{\Omega}_T$ is the domain extended by a factor $T \geq 1$ and a radial basis kernel $\phi(\cdot)$. Using the approximation in Definition 7, the gradient of the approximate solution is given by

$$\nabla \hat{u}(\vec{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^N w_i\,\nabla\phi(\|\vec{x} - \vec{c}_i\|_2) \in \mathbb{R}^d. \tag{2.23}$$

We aggregate the centers into the matrix

$$C := \begin{bmatrix} \vec{c}_1 & \vec{c}_2 & \cdots & \vec{c}_N \end{bmatrix} \in \mathbb{R}^{d\times N},$$

We define the feature vector $\Phi(\vec{x}, C)$ and the gradient matrix $\nabla\Phi(\vec{x}, C)$ as follows:

$$\Phi(\vec{x}, C) = \begin{bmatrix} \phi(\|\vec{x} - \vec{c}_1\|_2) \\ \vdots \\ \phi(\|\vec{x} - \vec{c}_N\|_2) \end{bmatrix} \in \mathbb{R}^N \quad \text{and} \quad \nabla\Phi(\vec{x}, C) = \begin{bmatrix} \nabla\phi^\top(\|\vec{x} - \vec{c}_1\|_2) \\ \vdots \\ \nabla\phi^\top(\|\vec{x} - \vec{c}_N\|_2) \end{bmatrix} \in \mathbb{R}^{N\times d}. \tag{2.24}$$

The pointwise Dirichlet energy admits the quadratic form

$$\|\nabla\hat{u}(\vec{x})\|_2^2 = \frac{1}{N}\vec{w}^\top\nabla\Phi(\vec{x}, C)\nabla\Phi(\vec{x}, C)^\top\vec{w}. \tag{2.25}$$

Approximating the integral over Ω via the interior sample points $\{\vec{x}_j^{(\Omega)}\}_{j=1}^{m_\Omega}$ yields

$$\int_\Omega \frac{1}{2}\|\nabla\hat{u}\|_2^2\, d\vec{x} \approx \frac{1}{2}\vec{w}^\top A_1\vec{w}, \tag{2.26}$$

where

$$A_1 := \frac{|\Omega|}{Nm_\Omega}\sum_{j=1}^{m_\Omega}\nabla\Phi(\vec{x}_j^{(\Omega)}, C)\nabla\Phi(\vec{x}_j^{(\Omega)}, C)^\top \in \mathbb{R}^{N\times N}. \tag{2.27}$$

The load matrix is discretized as

$$\int_\Omega f(\vec{x})\,\hat{u}(\vec{x})\, d\vec{x} \approx \vec{f}^\top A_2\vec{w}, \tag{2.28}$$

with

$$A_2 := \frac{|\Omega|}{\sqrt{N}\, m_\Omega}\begin{bmatrix} \Phi(\vec{x}_1^{(\Omega)}, C)^\top \\ \vdots \\ \Phi(\vec{x}_{m_\Omega}^{(\Omega)}, C)^\top \end{bmatrix} \in \mathbb{R}^{m_\Omega\times N} \quad \text{and} \quad \vec{f} := \begin{bmatrix} f\left(\vec{x}_1^{(\Omega)}\right) \\ \vdots \\ f\left(\vec{x}_{m_\Omega}^{(\Omega)}\right) \end{bmatrix} \in \mathbb{R}^{m_\Omega}. \tag{2.29}$$

The boundary penalty term is approximated using boundary sample points $\{\vec{x}_\ell^{(\partial\Omega)}\}_{\ell=1}^{m_{\partial\Omega}}$ as

$$\frac{\beta}{2}\int_{\partial\Omega}(\hat{u} - g)^2\, d\vec{s} \approx \frac{\beta}{2}\|A_3\vec{w} - \vec{g}\|_2^2, \tag{2.30}$$

where

$$A_3 := \sqrt{\frac{|\partial\Omega|}{N m_{\partial\Omega}}} \begin{bmatrix} \Phi(\vec{x}_1^{(\partial\Omega)}, C)^\top \\ \vdots \\ \Phi(\vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}, C)^\top \end{bmatrix} \in \mathbb{R}^{m_{\partial\Omega} \times N} \quad \text{and} \quad \vec{g} := \sqrt{\frac{|\partial\Omega|}{m_{\partial\Omega}}} \begin{bmatrix} g(\vec{x}_1^{(\partial\Omega)}) \\ \vdots \\ g(\vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}) \end{bmatrix} \in \mathbb{R}^{m_{\partial\Omega}}. \tag{2.31}$$

Collecting the above terms leads to the following discrete variational problem:

Definition 14 (Discrete Variational Formulation for Elliptic PDEs using RBF) *The penalized variational formulation of the Poisson problem in Eq. (3.2) admits the discrete form*

$$\min_{\vec{w} \in \mathbb{R}^N} \left\{ \frac{1}{2} \vec{w}^\top A_1 \vec{w} - \vec{f}^\top A_2 \vec{w} + \frac{\beta}{2} \|A_3 \vec{w} - \vec{g}\|_2^2 \right\}, \tag{2.32}$$

where A_1, A_2, and A_3 are defined in Eqs. (2.27)–(2.31).

For reaction–diffusion equations, the stiffness matrix A_1 is modified by the addition of a mass term,

$$A_{1,\,\text{reaction-diffusion}} = \frac{|\Omega|}{N m_\Omega} \sum_{j=1}^{m_\Omega} \left(\nabla\Phi(\vec{x}_j^{(\Omega)}, C) \nabla\Phi(\vec{x}_j^{(\Omega)}, C)^\top + \Phi(\vec{x}_j^{(\Omega)}, C) \Phi(\vec{x}_j^{(\Omega)}, C)^\top \right). \tag{2.33}$$

We now consider the obstacle-type boundary value problem. The non-smooth penalty term is discretized as

$$\mu \int_\Omega (\psi - \hat{u})_+ \, d\vec{x} \approx \mu \|(\vec{\psi} - A_2 \vec{w})_+\|_1, \tag{2.34}$$

where

$$\vec{\psi} := \frac{|\Omega|}{m_\Omega} \begin{bmatrix} \psi(\vec{x}_1^{(\Omega)}) \\ \vdots \\ \psi(\vec{x}_{m_\Omega}^{(\Omega)}) \end{bmatrix}. \tag{2.35}$$

This yields the following discrete obstacle formulation.

Definition 15 (Discrete Variational Formulation for Free-BVP using RBF) *The penalized variational formulation of the obstacle problem in Eq. (3.6) is given by*

$$\min_{\vec{w} \in \mathbb{R}^N} \left\{ \frac{1}{2} \vec{w}^\top A_1 \vec{w} + \frac{\beta}{2} \|A_3 \vec{w} - \vec{g}\|_2^2 + \mu \|(\vec{\psi} - A_2 \vec{w})_+\|_1 \right\}. \tag{2.36}$$

Example 1 (One-dimensional discretization) *For $d = 1$, the feature and gradient matrices evaluated in the m_Ω interior sample points, according to Eq. (2.24), can be written as*

$$\Phi(X_{m_\Omega}, C) = \begin{pmatrix} \phi(\|x_1^{(\Omega)} - c_1\|_2) & \cdots & \phi(\|x_1^{(\Omega)} - c_N\|_2) \\ \vdots & & \vdots \\ \phi(\|x_{m_\Omega}^{(\Omega)} - c_1\|_2) & \cdots & \phi(\|x_{m_\Omega}^{(\Omega)} - c_N\|_2) \end{pmatrix},$$

$$\nabla\Phi(X_{m_\Omega}, C) = \begin{pmatrix} \phi'(\|x_1^{(\Omega)} - c_1\|_2) & \cdots & \phi'(\|x_1^{(\Omega)} - c_N\|_2) \\ \vdots & & \vdots \\ \phi'(\|x_{m_\Omega}^{(\Omega)} - c_1\|_2) & \cdots & \phi'(\|x_{m_\Omega}^{(\Omega)} - c_N\|_2) \end{pmatrix}.$$

where $X_{m_\Omega} = \left(x_1^{(\Omega)}, \ldots, x_{m_\Omega}^{(\Omega)}\right)^\top$. Consequently,

$$A_1 = \frac{|\Omega|}{N m_\Omega} \big(\nabla\Phi(X_{m_\Omega}, C)\big)^\top \nabla\Phi(X_{m_\Omega}, C) \quad \textit{and} \quad A_2 = \frac{|\Omega|}{\sqrt{N} m_\Omega} \Phi(X_{m_\Omega}, C),$$

and, similarly,

$$A_3 = \sqrt{\frac{|\partial\Omega|}{N m_{\partial\Omega}}} \Phi(X_{m_{\partial\Omega}}, C).$$

2.6.2 Discretization with random features models

Following from Sections 2.2 and 2.3 and the approximation model in Eq. (2.17), we derive the discrete numerical formulation corresponding to the variational equations for elliptic and obstacle-type boundary value problems. We let $m = m_\Omega + m_{\partial\Omega}$ denote the total number of collocation points in $\overline{\Omega} = \Omega \cup \partial\Omega$. We select M features such that the weight vectors and biases are sampled independently from a probability density function as follows.

$$\vec{\omega}_i \sim \mathcal{N}(0, b_{RF}^2\, I_d), \quad b_i \sim \mathcal{U}\left(\overline{\Omega}_{T_{\mathrm{RF}},d}\right), \quad i = 1, \ldots, M,$$

where $b_{RF} > 0$ is a standard deviation scaling parameter. We recall the random features model with hyperbolic tangent activation in Eq. (2.17):

$$\hat{u}_{RF}(\vec{x}) = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} d_i\, \sigma\Big(\vec{\omega}_i^\top\, \vec{x} + b_i\Big), \quad \sigma(z) = \tanh(z),$$

where $\vec{d} \in \mathbb{R}^M$ are the unknown coefficients of the model to be learned, and $\{\vec{\omega}_i, b_i\}_{i=1}^M$ are fixed parameters with the random initialization, as described above. Similar to Section 2.6.1, we specify the variational problem in Definition 14 for elliptic BVPs as follows.

$$\min_u \mathcal{J}_{\beta,\,\text{poisson}}(u), \quad \text{where} \quad \mathcal{J}_{\beta,\text{poisson}}(u) = \frac{1}{2}\int_\Omega \|\nabla u\|_2^2 d\,\vec{x} - \int_\Omega f\,u\,d\,\vec{x} + \frac{\beta}{2}\int_{\partial\Omega}(u-g)^2 d\,\vec{s}.$$

The corresponding discrete numerical formulation is given by

$$\min_{\vec{d}\in\mathbb{R}^M} \underbrace{\frac{|\Omega|}{2m_\Omega}\sum_{j=1}^{m_\Omega}\Big(\|\nabla\hat{u}(\vec{x}_j^{(\Omega)})\|_2^2 - f(\vec{x}_j^{(\Omega)})\,\hat{u}(\vec{x}_j^{(\Omega)})\Big) + \frac{\beta|\partial\Omega|}{2m_{\partial\Omega}}\sum_{\ell=1}^{m_{\partial\Omega}}\Big(\hat{u}(\vec{x}_\ell^{(\partial\Omega)}) - g(\vec{x}_\ell^{(\partial\Omega)})\Big)^2}_{\widehat{\mathcal{J}}_{\beta,\text{poisson}}(\hat{u}_{\text{RF}})}.$$

Then we deduce that

$$\nabla\hat{u}_{\text{RF}}(\vec{x}) = \frac{1}{\sqrt{M}}\sum_{i=1}^{M} d_i\sigma'\Big(\vec{\omega_i}^\top\vec{x} + b_i\Big)\,\vec{\omega_i} = \frac{1}{\sqrt{M}}\,J(\vec{x})\,\vec{d}, \tag{2.37}$$

where the (fixed) Jacobian matrix $J(\vec{x}) \in \mathbb{R}^{d\times M}$ is

$$J(\vec{x}) \coloneqq \Big[\sigma'(\vec{\omega_1}^\top\vec{x} + b_1)\vec{\omega_1}\ \cdots\ \sigma'(\vec{\omega_M}^\top\vec{x} + b_M)\vec{\omega_M}\Big].$$

Thus, for $j = 1, \ldots, m_\Omega$ and the interior sample points $\{\vec{x}_j^{(\Omega)}\}_{j=1}^{m_\Omega}$ we get that:

$$\|\nabla\hat{u}_{RF}(\vec{x}_j^{(\Omega)})\|_2^2 = \|J(\vec{x}_j^{(\Omega)})\,\vec{d}\|_2^2 = \vec{d}^\top\,J\left(\vec{x}_j^{(\Omega)}\right)^\top\,J\left(\vec{x}_j^{(\Omega)}\right)\,\vec{d}. \tag{2.38}$$

From Eq. (2.38), this yields

$$\int_\Omega \|\nabla u\|_2^2 d\,\vec{x} \approx \frac{|\Omega|}{m_\Omega}\,\vec{d}^\top\sum_{j=1}^{m_\Omega} J\left(\vec{x}_j^{(\Omega)}\right)^\top\,J\left(\vec{x}_j^{(\Omega)}\right)\,\vec{d}. \tag{2.39}$$

Hence, we define

$$B_1 \coloneqq \frac{|\Omega|}{M\,m_\Omega}\sum_{j=1}^{m_\Omega} J\left(\vec{x}_j^{(\Omega)}\right)^\top\,J\left(\vec{x}_j^{(\Omega)}\right) \in \mathbb{R}^{M\times M} \tag{2.40}$$

Similarly, in the forcing term in Ω, since $\hat{u}_{RF}(\vec{x}) = \frac{1}{\sqrt{M}}\,\vec{d}^\top\,\sigma\left(W\,\vec{x} + \vec{b}\right)$, we have

$$\frac{|\Omega|}{m_\Omega}\sum_{j=1}^{m_\Omega} f(\vec{x}_j^{(\Omega)})\,\hat{u}_{RF}(\vec{x}_j^{(\Omega)}) = \frac{|\Omega|}{m_\Omega\sqrt{M}}\sum_{j=1}^{m_\Omega} f(\vec{x}_j^{(\Omega)})\,\sigma\left(W\,\vec{x}_j^{(\Omega)} + \vec{b}\right)^\top\,\vec{d}$$

Then, the forcing term becomes expressed as follows:

$$\int_{\Omega} f(\vec{x})u(\vec{x})d\,\vec{x} \approx \vec{f}^{\top}\, B_2\vec{d},$$

with

$$B_2 \coloneqq \frac{|\Omega|}{\sqrt{M}\, m_{\Omega}} \begin{bmatrix} \sigma\left(W\,\vec{x}_1^{(\Omega)} + \vec{b}\right)^{\top} \\ \vdots \\ \sigma\left(W\,\vec{x}_{m_{\Omega}}^{(\Omega)} + \vec{b}\right)^{\top} \end{bmatrix} \in \mathbb{R}^{m_{\Omega}\times M} \quad \text{and} \quad \vec{f} = \begin{bmatrix} f\left(\vec{x}_1^{(\Omega)}\right) \\ \vdots \\ f\left(\vec{x}_{m_{\Omega}}^{(\Omega)}\right) \end{bmatrix} \tag{2.41}$$

With the boundary sample points $\left\{\vec{x}_{\ell}^{(\partial\Omega)}\right\}_{\ell=1}^{m_{\partial\Omega}}$, we let

$$B_3 \coloneqq \sqrt{\frac{|\partial\Omega|}{M\, m_{\partial\Omega}}} \begin{bmatrix} \sigma\left(W\,\vec{x}_1^{(\partial\Omega)} + \vec{b}\right)^{\top} \\ \vdots \\ \sigma\left(W\,\vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)} + \vec{b}\right)^{\top} \end{bmatrix} \in \mathbb{R}^{m_{\partial\Omega}\times M}. \tag{2.42}$$

Then, we deduce that:

$$\frac{\beta}{2}\int_{\partial\Omega}(u-g)^2\, d\,\vec{s} \approx \frac{\beta}{2}\,\frac{|\partial\Omega|}{m_{\partial\Omega}}\sum_{\ell}\left(\hat{u}_{RF}(\vec{x}_{\ell}^{(\partial\Omega)}) - g(\vec{x}_{\ell}^{(\partial\Omega)})\right)^2 = \frac{\beta}{2}\|B_3\vec{d} - \vec{g}\|_2^2, \tag{2.43}$$

where the Dirichlet boundary condition term is $\vec{g} \coloneqq \sqrt{\frac{|\partial\Omega|}{m_{\partial\Omega}}}\left(g(\vec{x}_1^{(\partial\Omega)}), \ldots, g(\vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)})\right)^{\top}$. We combine the terms defined from Eqs. (2.40)-(2.42), which yields the following discrete variational problem for Poisson BVPs using RFM:

Definition 16 (Discrete Variational Formulation for Poisson BVP using RFM) *The variational formulation of linear elliptic PDEs with Dirichlet boundary conditions admits the discrete optimization form*

$$\min_{\vec{d}\in\mathbb{R}^M}\left\{\frac{1}{2}\vec{d}^{\top}\, B_1\,\vec{d} - \vec{f}^{\top}\, B_2\,\vec{d} + \frac{\beta}{2}\|B_3\,\vec{d} - \vec{g}\|_2^2\right\} \tag{2.44}$$

where B_1, B_2 and B_3 are defined in Eqs. (2.40), (2.41), and (2.42), respectively.

Similarly, according to Section 2.3, the variational equation for obstacle problems is as follows:

$$\min_{u} \left\{ \frac{1}{2} \int_{\Omega} \|\nabla u(\vec{x})\|_2^2 \, d\,\vec{x} + \mu \int_{\Omega} \Big(\psi - u\Big)_+ \, d\,\vec{x} + \frac{\beta}{2} \int_{\partial\Omega} \Big(u - g)^2 \, d\,\vec{s} \right\}.$$

The difference in Definition 16 is the non-smooth penalty term. Maintaining the same notation for RFM, this can be discretized as follows:

$$\mu \int_{\Omega} (\psi - u)_+ \, d\,\vec{x} \approx \mu \|(\vec{\psi} - B_2\,\vec{d})_+\|_1, \tag{2.45}$$

where the obstacle term $\vec{\psi}$ is defined in Eq. (3.20). This results in the following discrete obstacle-type variational formulation using RFM:

Definition 17 (Discrete Variational Formulation for Free-BVP using RFM) *The discrete variational formulation of the obstacle problem with RFM yields*

$$\min_{\vec{d}\in\mathbb{R}^M} \left\{ \frac{1}{2} \vec{d}^{\top} B_1\,\vec{d} + \mu \|(\vec{\psi} - B_2\,\vec{d})_+\|_1 + \frac{\beta}{2} \|B_3\,\vec{d} - \vec{g}\|_2^2 \right\} \tag{2.46}$$

where the matrices B_1, B_2 and B_3 are defined according to Eqs. (2.40), (2.41), and (2.42) respectively.

Since B_1 is symmetric positive semi-definite and the boundary penalty contributes the positive semi-definite term $\beta B_3^{\top} B_3$, the quadratic component of both formulations is convex. The addition of the ℓ^1-penalty preserves convexity, ensuring that the obstacle formulation remains a convex optimization problem. The random feature model in Definition 10 implicitly encodes the active set in the coefficient vector $\vec{d}$, while in classical finite-element discretizations, the active set is identified in the nodal space. Both optimization problems defined in Definitions 16-17 shape the methodologies to apply the random feature model in Definition 10 to solve boundary value problems and establish a unified framework. The Poisson BVPs yield a smooth quadratic optimization problem, whereas the obstacle problem leads to a convex but non-smooth optimization problem. This enables the systematic application of modern optimization algorithms when solving high-dimensional boundary value problems. The quadratic Poisson formulation in Definition 16 can be solved using linear solvers. In contrast, the obstacle formulation requires non-smooth convex optimization techniques, such as proximal gradient descent methods, proximal point methods, or ADMM. In this thesis, we use ADMM to solve the corresponding non-smooth convex optimization problem.

2.7 Alternating Direction Method of Multipliers

Optimization problems arising from variational formulations of partial differential equations frequently involve non-smooth constraints, such as inequality conditions in obstacle-type boundary value problems [107]. These constraints naturally lead to constrained minimization problems, whose numerical treatment requires careful consideration of both primal feasibility and dual optimality. In this section, we review the primal-dual structure underlying such problems and use the ADMM as a robust algorithmic framework for their solution. For details, we refer the reader to [15]. We begin with the primal constrained optimization problem, which encodes the variational energy along with its associated constraints.

Definition 18 (Primal Optimization Problem) *Let $F : \mathbb{R}^N \to \mathbb{R}$ be a convex and continuously differentiable function, and let $G : \mathbb{R}^m \to \mathbb{R}$ be a convex functional that may be non-differentiable. The primal problem is defined as*

$$\begin{aligned} \min_{\vec{w},\vec{v}} \quad & F(\vec{w}) + G(\vec{v}) \\ \textit{subject to} \quad & Q_1\vec{w} + Q_2\vec{v} = \vec{q}_3, \end{aligned} \tag{2.47}$$

where $\vec{w} \in \mathbb{R}^N$ and $\vec{v} \in \mathbb{R}^m$ are primal decision variables, $Q_1 \in \mathbb{R}^{p\times N}$ and $Q_2 \in \mathbb{R}^{p\times m}$ are given matrices, and $\vec{q}_3 \in \mathbb{R}^p$ is a prescribed constraint vector.

This formulation is sufficiently general to encompass a wide class of variational problems arising from linear elliptic- and obstacle-type PDEs. In particular, the function F typically corresponds to a smooth quadratic energy derived from the weak form of a differential operator, while G encodes non-smooth penalties or constraint enforcement terms. To analyze optimality and derive iterative solvers, one introduces Lagrange multipliers associated with the equality constraint and constructs the dual problem as follows.

Definition 19 (Dual Optimization Problem) *Let $\vec{\lambda} \in \mathbb{R}^p$ denote the Lagrange multiplier associated with the constraint in Definition 18. The dual problem is defined as*

$$\max_{\vec{\lambda}\in\mathbb{R}^p} \; d(\vec{\lambda}), \tag{2.48}$$

where the dual function $d : \mathbb{R}^p \to \mathbb{R}$ is given by

$$d(\vec{\lambda}) = \inf_{\vec{w},\vec{v}} \Big[F(\vec{w}) + G(\vec{v}) + \vec{\lambda}^\top \big(Q_1\vec{w} + Q_2\vec{v} - \vec{q}_3\big) \Big]. \tag{2.49}$$

Under standard assumptions of convexity and appropriate constraint qualifications, strong duality holds, implying that the optimal objective values of the primal and dual problems coincide. The primal-dual gap, therefore, provides a natural measure of optimality. However, directly solving the dual problem via gradient ascent or dual decomposition can be inefficient when the constraint sets are large or when the system matrices are ill-conditioned, which is often the case in mesh-free discretizations [15]. To improve numerical stability and convergence, augmented Lagrangian methods modify the classical Lagrangian by adding a quadratic penalty on constraint violations. This leads to a formulation that blends primal and dual information.

Definition 20 (Primal-Dual Augmented Lagrangian) *Let $\rho > 0$ be a penalty parameter, and let $\vec{z} \in \mathbb{R}^p$ be the dual variable. The scaled augmented Lagrangian associated with Definition 18 is defined as*

$$L_\rho(\vec{w}, \vec{v}, \vec{z}) = F(\vec{w}) + G(\vec{v}) + \frac{\rho}{2} \|Q_1\vec{w} + Q_2\vec{v} - \vec{q}_3 + \vec{z}\|_2^2 - \frac{\rho}{2}\|\vec{z}\|_2^2. \tag{2.50}$$

The quadratic penalty term enforces the constraint progressively during iteration, while the dual variable $\vec{z}$ accumulates information about the violations of the constraint. This formulation mitigates the slow convergence typically associated with pure dual ascent methods. ADMM exploits the separable structure of the augmented Lagrangian by alternating minimization with respect to the primal variables $\vec{w}$ and $\vec{v}$, followed by an update of the dual variable. This splitting strategy is particularly effective when F and G admit efficient minimization subproblems.

Definition 21 (ADMM Iterative Updates) *Given an initial iterate $(\vec{w}^{(0)}, \vec{v}^{(0)}, \vec{z}^{(0)})$, ADMM iterations are defined for $i \geq 0$ by*

$$\vec{w}^{(i+1)} := \arg\min_{\vec{w}} \left\{ F(\vec{w}) + \frac{\rho}{2} \left\|Q_1\vec{w} + Q_2\vec{v}^{(i)} - \vec{q}_3 + \vec{z}^{(i)}\right\|_2^2 \right\}, \tag{2.51}$$

$$\vec{v}^{(i+1)} := \arg\min_{\vec{v}} \left\{ G(\vec{v}) + \frac{\rho}{2} \left\|Q_1\vec{w}^{(i+1)} + Q_2\vec{v} - \vec{q}_3 + \vec{z}^{(i)}\right\|_2^2 \right\}, \tag{2.52}$$

$$\vec{z}^{(i+1)} := \vec{z}^{(i)} + Q_1\vec{w}^{(i+1)} + Q_2\vec{v}^{(i+1)} - \vec{q}_3. \tag{2.53}$$

The convergence of ADMM is assessed using residuals that measure the feasibility and stability of the iteration as follows:

Definition 22 (Primal and Dual Residuals) *In iteration $(i+1)$, the primal and dual residuals are defined as*

$$\vec{r}^{(i+1)} := Q_1\vec{w}^{(i+1)} + Q_2\vec{v}^{(i+1)} - \vec{q}_3, \tag{2.54}$$

$$\vec{s}^{(i+1)} := \rho\, Q_1^\top Q_2 \left(\vec{v}^{(i+1)} - \vec{v}^{(i)}\right). \tag{2.55}$$

The primal residual measures the violation of the equality constraint, whereas the dual residual measures the change in the dual variables induced by updates of the primal variables.

Stopping criteria. Iterations are terminated once both residuals satisfy the prescribed relative tolerance criteria.

Definition 23 (ADMM Stopping Criteria) *Given tolerances $\epsilon_{\text{primal}}, \epsilon_{\text{dual}} > 0$, the ADMM iterations are terminated when*

$$\frac{\|\vec{r}^{(i+1)}\|_2}{\max(\|Q_1\vec{w}^{(i+1)}\|_2, \|Q_2\vec{v}^{(i+1)}\|_2, \|\vec{q}_3\|_2)} \le \epsilon_{\text{primal}}, \tag{2.56}$$

$$\frac{\|\vec{s}^{(i+1)}\|_2}{\|Q_1^\top \vec{z}^{(i+1)}\|_2} \le \epsilon_{\text{dual}}. \tag{2.57}$$

Typical tolerance choices are $\epsilon_{\text{primal}} = \epsilon_{\text{dual}} = 10^{-6}$ or 10^{-4}, depending on the desired numerical accuracy.

The ADMM framework described above provides a flexible and robust foundation for solving the non-smooth discrete variational formulations introduced in Section 2.6. In the next chapters, this general algorithm is specialized to the obstacle-type boundary value problems considered in this thesis.

Chapter 3

Stable Mesh-Free Variational Radial Basis Function Approximation for Elliptic PDEs and Obstacle Problem

[1]This chapter is based on the following paper: Tan Phuong Dong Le, Giang Tran, and Hans De Sterck, *Stable mesh-free variational radial basis function approximation for elliptic PDEs and obstacle problems*, arXiv:2604.19882, 2026.

3.1 Introduction

In this chapter, we develop a stable mesh-free variational framework using RBF approximations for elliptic boundary value problems and classical obstacle-type variational inequalities. PDEs are reformulated as optimization problems, as mentioned in Section 2.6.1, and are projected onto finite-dimensional spaces. Boundary conditions are imposed through quadratic penalty terms, while obstacle inequalities are incorporated via an ℓ^1 penalty term, resulting in a convex and non-smooth optimization problem.

The difficulty in global RBF discretizations lies in the severe ill-conditioning associated with the shape parameter b_{RBF} and the number of basis functions N. To ensure numerical stability, TSVD is applied to regularize the least-squares systems derived from a variational framework. For obstacle problems, the non-smooth formulation is solved using ADMM and TSVD within each convex quadratic minimization problem.

The following chapters provide rigorous derivations of the discrete variational forms for Poisson and reaction-diffusion equations, as well as for obstacle-type PDEs. Section 3.2 introduces the boundary value problems, variational formulations, and algorithmic frameworks that we address. In Section 3.3, we formulate the algorithms into pipelines to solve the associated boundary value problems. Numerical experiments in Section 3.4 demonstrate the monotone convergence for linear elliptic PDEs and obstacle-type BVPs. With suitable choices of parameters and scaling, the proposed variational RBF method achieves high accuracy, indicating fast convergence.

These results substantiate the effectiveness of the stable variational framework RBF for elliptic and free-boundary value problems. This analysis is a central component of the thesis, supporting the empirical results of the theoretical developments presented in the preceding chapters.

3.2 Problem Statement

3.2.1 Variational Functionals for Boundary Value Problem

The central objective of this chapter is to construct a stable, variational, mesh-free framework for solving elliptic BVPs and obstacle-type PDEs. Before introducing the approximation methodology, including algorithmic frameworks, it is essential to clearly define the class of differential equations and inequality constraints that we study.

We state the mathematical problem, including both strong and weak formulations. For details of the derivation and understanding, we refer readers to Sections 2.1, 2.4, and 2.6.1. Let $\Omega \subseteq \mathbb{R}^d$ with $d \geq 1$ be a bounded domain with a sufficiently smooth boundary $\partial\Omega$. We consider the linear elliptic BVPs as defined in Definition 2 as follows:

$$\mathcal{L}u = f \quad \text{in } \Omega, \qquad \mathcal{B}u = g \quad \text{on } \partial\Omega, \tag{3.1}$$

where $\mathcal{L}$ is a second-order elliptic differential operator,

$$\mathcal{L}u \coloneqq -\nabla \cdot (a(\vec{x})\, \nabla u) + c(\vec{x})u,$$

with $a(\vec{x}) > 0$ and $c(\vec{x}) \geq 0$ in Ω, $\mathcal{B}$ denotes a boundary operator for Dirichlet, Neumann, or mixed boundary conditions. According to the results in [61], we consider Poisson, Reaction-Diffusion, and Free-boundary value problems. The Poisson equation is given by:

$$\begin{cases} -\Delta u(\vec{x}) = f(\vec{x}), & \vec{x} \in \Omega \subseteq \mathbb{R}^d \\ u(\vec{x}) = g(\vec{x}), & \text{on } \partial\Omega. \end{cases} \tag{3.2}$$

From Section 2.2, we recall that the variational problem for Eq. (3.2) is given by:

$$\frac{1}{2}\int_\Omega \|\nabla u\|_2^2 \, d\,\vec{x} - \int_\Omega f\, u \, d\,\vec{x}, \quad \text{such that} \quad u(\vec{x}) = g(\vec{x}) \quad \forall \vec{x} \in \partial\Omega. \tag{3.3}$$

Using the penalty method as in [120], we impose the boundary condition and minimize a modified functional from Definition 5.

$$\min_u \mathcal{J}_{\beta,\text{poisson}}(u),$$

where

$$\mathcal{J}_{\beta,\text{poisson}}(u) = \frac{1}{2}\int_\Omega \|\nabla u\|_2^2 \, d\,\vec{x} - \int_\Omega f\, u \, d\,\vec{x} + \frac{\beta}{2}\int_{\partial\Omega} (u-g)^2 \, d\,\vec{s}, \quad (\beta > 0).$$

Similarly, we consider the reaction-diffusion equation, which is defined as follows:

$$\begin{cases} -\Delta u + u = f(\vec{x}), & \forall \vec{x} \in \Omega \\ u(\vec{s}) = g(\vec{s}), & \forall \vec{s} \in \partial\Omega. \end{cases} \tag{3.4}$$

The corresponding modified variational formulation of the reaction–diffusion boundary value problem in Eq. (3.4) is expressed as

$$\min_u \mathcal{J}_{\beta,\text{reaction-diffusion}}(u), \tag{3.5}$$

where the functional $\mathcal{J}_{\beta,\text{reaction-diffusion}}$ is defined by

$$\mathcal{J}_{\beta,\text{reaction-diffusion}}(u) = \frac{1}{2}\int_\Omega \left(\|\nabla u\|_2^2 + u^2\right)\, d\vec{x} - \int_\Omega f\, u\, d\vec{x} + \frac{\beta}{2}\int_{\partial\Omega}(u-g)^2\, d\vec{s}.$$

Lastly, we consider the obstacle-type BVPs which is defined as follows:

$$\begin{cases} -\Delta u(\vec{x}) \geq 0, & \forall \vec{x} \in \Omega, \\ u(\vec{s}) = g(\vec{s}), & \forall \vec{s} \in \partial\Omega, \\ u(\vec{x}) \geq \psi(\vec{x}), & \text{in } \Omega, \\ (-\Delta u)\,(u-\psi) = 0, & \text{in } \Omega, \end{cases} \tag{3.6}$$

where ψ denotes a prescribed obstacle function.

The following ℓ_1-penalty formulation provides a robust and flexible alternative to impose the obstacle constraint in Eq. (3.6) in the presence of inhomogeneous Dirichlet boundary conditions as in Definition 6, namely:

$$\min_u \mathcal{J}_{\mu,\beta,\,\text{obstacle}}(u) \tag{3.7}$$

where the functional $\mathcal{J}_{\mu,\beta,\,\text{obstacle}}$ is given by

$$\mathcal{J}_{\mu,\beta,\,\text{obstacle}}(u) = \frac{1}{2}\int_\Omega \|\nabla u\|_2^2\, d\,\vec{x} + \mu\int_\Omega (\psi - u)_+\, d\,\vec{x} + \frac{\beta}{2}\int_{\partial\Omega}(u-g)^2\, d\,\vec{s},$$

where $\mu > 0$ is the penalty coefficient for the free-boundary term, and $(\,\cdot\,)_+ = \max\{\cdot, 0\}$. In the next section 3.2.2, we recall the model RBF, which is summarized in Definition 7, and useful settings to constrain the necessary coefficients and parameters in the approximate representation.

3.2.2 Radial Basis Function Model

In this section, we introduce the RBF approximation model outlined in Definition 7, which serves as the finite-dimensional space for the variational formulations developed in the previous Section 3.2.1. The main purpose is to represent the weak solution u using a mesh-free expansion in terms of globally supported basis functions defined at center points in the computational domain. This approximation framework allows the continuous minimization problems for linear elliptic BVPs and obstacle-type PDEs. This reduces to finite-dimensional optimization problems for the RBF coefficients. The definition of the

approximate model, including the selection of centers and kernel parameters, is a fundamental step in constructing the stable variational framework RBF. We refer the reader to the previous Section 2.4 for a detailed understanding of the approximation model.

We let $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a well-defined basis function. Given a set of N center vectors $\{\vec{c}_i\}_{i=1}^N \subseteq \mathbb{R}^d$, a RBF approximation $\hat{u}_{\text{RBF}} : \overline{\Omega} \to \mathbb{R}$ is given by:

$$\hat{u}_{\text{RBF}}(\vec{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i \, \phi(\|\vec{x} - \vec{c}_i\|_2),$$

where $\vec{x} \in \overline{\Omega}$ and the weight coefficient vector is $\vec{w} = (w_1, \ldots, w_N)^\top \in \mathbb{R}^N$.

The representation defines a finite-dimensional approximation based on a radial kernel centered at points $\{\vec{c}_i\}_{i=1}^N$. The universal approximation theorem in [79] states that a single hidden layer with fixed radial basis units can uniformly approximate any integrable function in a compact subset of $\mathbb{R}^d$. Mesh-free approximation frameworks utilize the smoothness and locality properties of RBF to approximate the solution [114, 34]. For this approximation to be admissible in the variational framework introduced in Section 3.2.1, we impose standard settings on the basis function, center locations, and coefficient vector. These ensure that the approximation is well-defined in $\overline{\Omega} \subseteq \mathbb{R}^d$, yielding a consistent finite-dimensional approximation space. In practice, we place centers in an expanded domain $\overline{\Omega}_T$ where $T \geq 1$; see below.

Setting 1 (Expanded domain for center points, see [5]) *Let $\overline{\Omega} = [b_1, b_2] \subseteq \mathbb{R}$ be a bounded rectangular domain with midpoint $\vec{q}_{\text{mid}} = \frac{b_1+b_2}{2}$ and half-length $L_{\text{half}} = \frac{b_2-b_1}{2}$, so that*

$$\overline{\Omega}_T \coloneqq [q_{\text{mid}} - T\, L_{\text{half}},\, q_{\text{mid}} + T\, L_{\text{half}}] \subseteq \mathbb{R}, \tag{3.8}$$

where $T \geq 1$ is a prescribed expansion domain factor. We assume that the RBF center points $\{\vec{c}_i\}_{i=1}^N$ are sampled from the expanded domain Ω_T, while the approximation and error evaluation are performed on the original domain $\overline{\Omega}$.

The setting 2 states that we utilize Gaussian kernels that satisfy the universal approximation property of RBFs, as expressed by:

Setting 2 *In numerical experiments, we normally use the Gaussian kernel defined in Table 2.1*

$$\phi(r) = e^{-(b_{\text{RBF}}\, r)^2}, \tag{3.9}$$

In some numerical tests, we state the use of different kernels.

The Settings 3 and 4 guide the construction of the center points. The fill distance h_Ω measures the largest distance from any point in the domain Ω to the nearest center, while the separation distance q_Ω measures the minimal spacing between centers.

Setting 3 (Fill distance, see [34, 76]) *The center points are sampled from a distribution such that $\{\vec{c}_1, \ldots, \vec{c}_N\} \subseteq \overline{\Omega}_T$. Then, the fill distance property state that*

$$h_\Omega := \sup_{\vec{x}\in\Omega} \min_{1\le i\le N} |\vec{x} - \vec{c}_i| \to 0, \quad \text{as} \quad N \to \infty.$$

Setting 4 (Separation distance, see [34, 76]) *For a set of center points $\{\vec{c}_1, \ldots, \vec{c}_N\} \subseteq \overline{\Omega}_T$, the separation distance q_Ω satisfies*

$$q_\Omega := \frac{1}{2} \min_{\substack{i,j=1,\ldots,N \\ i\neq j}} \|\vec{c}_i - \vec{c}_j\|_2 \to 0, \quad \text{as} \quad N \to \infty.$$

Settings 3 and 4 ensure that the center points become increasingly dense in the computational domain. In [114, 76], these lead to the quasi-uniform condition, which is given by the following definition:

Definition 24 (Quasi-uniform distribution of centers, see [114]) *Let $C = \{\vec{c}_i\}_{i=1}^N \subset \overline{\Omega}_T$ be a set of RBF centers. The fill distance relative to the approximation domain Ω is defined by*

$$h_\Omega = \sup_{\vec{x}\in\Omega} \min_{1\le i\le N} \|\vec{x} - \vec{c}_i\|_2,$$

and the separation distance of the center set is defined by

$$q_\Omega = \frac{1}{2} \min_{i\neq j} \|\vec{c}_i - \vec{c}_j\|_2.$$

The center set C is called quasi-uniform if there exists a constant $A_{\text{qu}} > 0$, independent of N, such that

$$q_\Omega \le h_\Omega \le A_{\text{qu}}\, q_\Omega.$$

Equivalently, the ratio

$$\gamma_C := \frac{h_\Omega}{q_\Omega}$$

satisfies

$$1 \le \gamma_C \le A_{\text{qu}},$$

and hence remains uniformly bounded as $N \to \infty$.

With Settings 1-4 and Definition 24, one can derive the formulation of the shape multiplier factor, which is $c(T,\tau)$ as in [5]. For the domain expansion factor $T \geq 1$ and the truncation threshold $\tau \in (0,1)$ in TSVD, the shape-factor multiplier is chosen as follows:

$$c(T,\tau) = \min\left\{1,\ \frac{\pi}{T\sqrt{2\log(1+\tau^{-2})}}\right\}. \tag{3.10}$$

In practice, we assume that the resulting shape parameter lies in the stable scaling regime, namely

$$b_{\mathrm{RBF}} = c(T,\tau)\,N \qquad \text{in 1D},$$

and

$$b_{\mathrm{RBF}} = c(T,\tau)\,\sqrt{N} \qquad \text{in 2D}.$$

In Eq. (3.10), the multiplier $c(T,\tau)$ determines not only the proportionality between b_{RBF} and the number of basis functions N, but also the effective width of the Gaussian RBFs. As a result, it governs the trade-off between approximation accuracy and numerical stability. A larger c yields more localized basis functions, which generally improve conditioning but may reduce approximation accuracy. In contrast, a smaller c produces flatter and more overlapping basis functions, which can enhance approximation accuracy for smooth solutions but may also lead to ill-conditioning and greater sensitivity to truncation errors. Therefore, the choice $c = c(T,\tau)$ serves as a practical compromise between expressive approximation and stable numerical computation. Lastly, Remark 1 states that using the radial kernel from Setting 2 results in a frame sequence for the closed span $\{\phi(\|\vec{x}-\vec{c}_i\|_2)\}_{i=1}^N$, ensuring uniform equivalence between the coefficient and function norms. Remark 1 is given by:

Remark 1 (Riesz-sequence Frame for RBFs, see [3]) *Let H be a Hilbert space, for example $L^2(\Omega)$, $H^s(\Omega)$, or the RKHS associated with a positive-definite kernel. A family of RBFs*

$$\Phi = \{\phi_i\}_{i=1}^N, \qquad \phi_i(\vec{x}) = \phi(\|\vec{x}-\vec{c}_i\|_2),$$

is called a frame for H if there exist constants $0 < B_1 \leq B_2 < \infty$ such that

$$B_1\|\vec{\omega}\|_H^2 \leq \sum_{i=1}^N |\langle \omega_i, \phi_i\rangle_H|^2 \leq B_2\|\vec{\omega}\|_H^2, \qquad \forall \vec{\omega} \in H.$$

Remark 1 is generalized in [3] from Gaussian RBF to arbitrary compactly supported functions $\phi \in L^1(\mathbb{R}^d) \cap C(\mathbb{R}^d)$. This implies that the kernels in Table 2.1, including both globally and compactly supported functions, satisfy the frame property in Remark 1. In finite-dimensional setting, the corresponding stiffness matrices may still be ill-conditioned. This frame property is crucial for applying the TSVD methods developed in [2, 4, 5].

In general, Section 3.2 has established the mathematical and approximation framework that underpins the methodology in Section 3.3 of this thesis. The variational formulations for the linear elliptic BVPs and obstacle-type PDEs models provide a continuous optimization setting where the unknown solution is the minimizer of an appropriate functional, as described in Section 3.2.1. This perspective is suitable for mesh-free approximation, as it enables the governing equations and constraints to be incorporated directly into an energy-based formulation without relying on a mesh-dependent discretization.

Within this framework, the RBF model provides a finite-dimensional approximation space to represent the unknown solution. By expressing the approximation as a linear combination of radial basis functions centered at prescribed points, the original infinite-dimensional problem is reduced to determining a finite set of coefficients. Settings 1-4 on the basis function, center distribution, and expanded computational domain ensure that this approximation space is mathematically admissible and flexible for the considered problems. Using sampled center points in an expanded domain provides the construction of the kernel space and supports improved approximation behavior near the boundary of the physical domain $\overline{\Omega}$.

These results provide the foundation for the methodology in [61], discussed later in Section 3.3. In this way, the variational functions of BVPs in Section 3.2.1 are translated into a numerical method for the stable mesh-free approximation of the BVPs studied in this thesis.

3.3 Variational Algorithm for Radial Basis Function Model

Section 3.3 focuses on the numerical implementation of the variational approximation framework RBF developed in the preceding section. The continuous formulations in Section 3.2 define each model problem's solution as the minimizer of a well-defined variational functional. By replacing the associated infinite-dimensional function space with a finite-dimensional RBF trial space, the problem reduces to finding a finite number of expansion

coefficients. This section derives the corresponding discrete optimization problem and describes the computational methodology used in this thesis. To discretize the variational functional, we introduce two finite sets of collocation points: m_Ω interior sampling points in Ω and $m_{\partial\Omega}$ boundary sampling points in $\partial\Omega$. To discretize the variational functional, we introduce two finite sets: m_Ω interior sample points in Ω and $m_{\partial\Omega}$ boundary sample points in $\partial\Omega$.

The interior discretization approximates the domain contributions in the variational functional, while the boundary discretization imposes the prescribed boundary conditions and evaluates any arising boundary terms. Consequently, the continuous objective functional is approximated by a discrete counterpart expressed entirely in terms of the RBF coefficients and the values of the approximation at the selected sample points. This results in a finite-dimensional, mesh-free optimization problem that can be solved numerically.

The methodology introduced above is now adapted to the two problem classes studied in this thesis. Although both rely on the same variational RBF approximation framework, their discrete formulations differ due to the analytical structure of the underlying problem. The next two subsections describe these constructions for linear elliptic BVPs and obstacle-type PDEs, respectively.

3.3.1 Algorithm for Linear Elliptic Boundary Value Problem

This subsection is devoted to the numerical methodology of linear elliptic boundary value problems using the variational RBF methodology developed in Section 3.2.1. For a detailed understanding, we refer readers to Section 2.6.1.

Starting from the continuous variational formulation, we approximate the unknown solution by a RBF expansion and replace the continuous functional with a discrete objective defined on suitable interior and boundary point sets. The numerical solution is obtained by solving the resulting finite-dimensional optimization problem for the RBF coefficient vector.

The methodology developed here provides a direct computational realization of the variational principle for elliptic problems. By evaluating the governing equation in the interior of the domain and enforcing the boundary conditions through boundary discretization, the method yields a fully mesh-free approximation procedure that is suitable for the class of model problems considered in this thesis. To start approximating the integrals in the variational PDEs, we select m collocation points from the domain $\overline{\Omega} = \Omega \cup \partial\Omega$, including m_Ω interior points $\vec{x}_1^{(\Omega)}, \ldots, \vec{x}_{m_\Omega}^{(\Omega)}$ and $m_{\partial\Omega}$ boundary points $\vec{x}_1^{(\partial\Omega)}, \ldots, \vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}$

From Section 2.6.1, we recall that the numerical discrete formulation for solving the corresponding modified minimization problem in Eq. (3.3) is given by:

$$\widehat{\mathcal{J}}_{\beta,\text{poisson}}(\vec{x}) = \frac{|\Omega|}{m_\Omega}\sum_{j=1}^{m_\Omega}\left(\frac{1}{2}\|\nabla\hat{u}(\vec{x}_j^{(\Omega)})\|_2^2 - f(\vec{x}_j^{(\Omega)})\hat{u}(\vec{x}_j^{(\Omega)})\right) + \frac{\beta\,|\partial\Omega|}{2\,m_{\partial\Omega}}\sum_{n=1}^{m_{\partial\Omega}}\left(\hat{u}(\vec{x}_n^{(\partial\Omega)}) - g(\vec{x}_n^{(\partial\Omega)})\right)^2. \tag{3.11}$$

Similarly, the discrete form of Eq. (3.4) for Reaction-Diffusion PDEs is defined as follows:

$$\begin{aligned}\widehat{\mathcal{J}}_{\mu,\beta,\text{reaction-diffusion}}(\vec{x}) = \frac{|\Omega|}{m_\Omega}\sum_{j=1}^{m_\Omega}\left[\frac{1}{2}\left\|\nabla\hat{u}\left(\vec{x}_j^{(\Omega)}\right)\right\|_2^2 + \hat{u}\left(\vec{x}_j^{(\Omega)}\right)^2 - f\left(\vec{x}_j^{(\Omega)}\right)\hat{u}\left(\vec{x}_j^{(\Omega)}\right)\right] \\ + \frac{\beta\,|\partial\Omega|}{2\,m_{\partial\Omega}}\sum_{n=1}^{m_{\partial\Omega}}\left(\hat{u}\left(\vec{x}_n^{(\partial\Omega)}\right) - g\left(\vec{x}_n^{(\partial\Omega)}\right)\right)^2.\end{aligned} \tag{3.12}$$

Here, $|\Omega|$ and $|\partial\Omega|$ denote the Lebesgue measure of the computational domains $\Omega, \partial\Omega$, respectively.

The discrete functionals in Eqs.(3.11) and (3.12) are obtained by replacing the continuous integrals in the corresponding variational formulations by finite averages over the selected interior and boundary sample points. The factors $|\Omega|/m_\Omega$ and $|\partial\Omega|/m_{\partial\Omega}$ provide the appropriate normalization so that the discrete objectives remain consistent with the underlying continuous measures. Consequently, the first summation represents the interior contribution associated with the differential operator and source term, while the second summation introduces a penalty term for the enforcement of the Dirichlet boundary condition on $\partial\Omega$.

For the Poisson problem, the discrete objective functional consists of the standard energy contribution involving the gradient of the approximation, together with the forcing term f, supplemented by a boundary penalty weighted by the parameter $\beta > 0$. For the reaction–diffusion PDEs, the discrete formulation additionally includes the reaction contribution in the interior term, which modifies the energy landscape of the optimization problem. In both cases, the penalty parameter β controls the weight in which the boundary conditions are imposed.

By substituting the RBF approximation $\hat{u}_{\text{RBF}}$ into the discrete functionals in Eqs. (3.11)-(3.12), both problems are reduced to finite-dimensional minimization problems concerning the coefficient vector $\vec{w} \in \mathbb{R}^N$.

The numerical approximation is therefore obtained by solving

$$\vec{w}^* \in \arg\min_{\vec{w}\in\mathbb{R}^N} \widehat{\mathcal{J}}_\beta(\vec{w}),$$

where $\widehat{\mathcal{J}}_\beta$ denotes the corresponding discrete variational function in Eqs. (3.11)-(3.12) for the problem under consideration. Once the optimal coefficient vector $\vec{w}^*$ has been computed, the approximate solution is recovered through the associated RBF expansion. We now reformulate the discrete objective functional in vectorized matrix form so that the variational RBF approximation problem can be expressed as a finite-dimensional minimization problem over the coefficient vector $\vec{w}$. For a detailed understanding of the derivation, we refer readers to Section 2.6.1.

By assembling the evaluations of the basis functions and their derivatives at the interior and boundary discretization points into suitable matrices, the discrete functional admits a compact algebraic representation in terms of matrix–vector products. For the Poisson PDEs, we approximate the Dirichlet energy integral, which is given by:

$$\|\nabla \hat{u}_{\mathrm{RBF}}(\vec{x})\|_2^2 = \frac{1}{N}\, \vec{w}^\top\, \nabla\Phi(\vec{x}, C)\, \nabla\Phi(\vec{x}, C)^\top\, \vec{w},$$

where the center points are collected into the matrix

$$C = \begin{bmatrix} \vec{c}_1 & \vec{c}_2 & \cdots & \vec{c}_N \end{bmatrix} \in \mathbb{R}^{d\times N}.$$

We deduce the approximation of the integral over Ω, which is outlined in Eq. (2.27) as follows:

$$\frac{1}{2}\int_\Omega \|\nabla \hat{u}\|_2^2\, d\,\vec{x} \approx \frac{1}{2}\vec{w}^\top\, A_1\, \vec{w},$$

where

$$A_{1,\mathrm{poisson}} = \frac{|\Omega|}{N\, m_\Omega} \sum_{j=1}^{m_\Omega} \nabla\Phi(\vec{x}_j^{(\Omega)}, C)\, \nabla\Phi(\vec{x}_j^{(\Omega)}, C) \in \mathbb{R}^{N\times N}.$$

In the same manner as Reaction-diffusion PDE, the stiffness matrix A_1 incorporates a reaction term in Eq.(2.33). We recall that,

$$A_{1,\,\text{reaction-diffusion}} = \frac{|\Omega|}{N m_\Omega} \sum_{j=1}^{m_\Omega} \left(\nabla\Phi(\vec{x}_j^{(\Omega)}, C) \nabla\Phi(\vec{x}_j^{(\Omega)}, C)^\top + \Phi(\vec{x}_j^{(\Omega)}, C) \Phi(\vec{x}_j^{(\Omega)}, C)^\top \right) \in \mathbb{R}^{N\times N}.$$

We also note that the load matrix term is discretized as follows:

$$\int_\Omega f\, \hat{u}\, d\,\vec{x} \approx \vec{f}^\top\, A_2\, \vec{w},$$

where the matrices for the terms are defined in Eq. (2.29), which is provided by:

$$A_2 = \frac{|\Omega|}{\sqrt{N}\, m_\Omega} \begin{bmatrix} \Phi\left(\vec{x}_1^{(\Omega)}, C\right)^\top \\ \vdots \\ \Phi\left(\vec{x}_{m_\Omega}^{(\Omega)}, C\right)^\top \end{bmatrix} \in \mathbb{R}^{m_\Omega \times N}, \qquad \vec{f} = \begin{bmatrix} f\left(\vec{x}_1^{(\Omega)}\right) \\ \vdots \\ f\left(\vec{x}_{m_\Omega}^{(\Omega)}\right) \end{bmatrix} \in \mathbb{R}^{m_\Omega}.$$

Similarly, the matrices for the integral over the boundary domain $\partial\Omega$ are given by:

$$\frac{\beta}{2} \int_{\partial\Omega} \left(u - g\right)^2 d\,\vec{s} \approx \frac{\beta}{2} \left\| A_3 \vec{w} - \vec{g} \right\|_2^2,$$

with the following matrices in Eq. (2.31)

$$A_3 = \sqrt{\frac{|\partial\Omega|}{N\, m_{\partial\Omega}}} \begin{bmatrix} \Phi\left(\vec{x}_1^{(\partial\Omega)}, C\right)^\top \\ \vdots \\ \Phi\left(\vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}, C\right)^\top \end{bmatrix} \in \mathbb{R}^{m_{\partial\Omega} \times N}, \qquad \vec{g} = \sqrt{\frac{|\partial\Omega|}{m_{\partial\Omega}}} \begin{bmatrix} g\left(\vec{x}_1^{(\partial\Omega)}\right) \\ \vdots \\ g\left(\vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}\right) \end{bmatrix} \in \mathbb{R}^{m_{\partial\Omega}}$$

Therefore, the discretized interior and boundary functionals together constitute a finite-dimensional unconstrained minimization problem in the coefficient vector $\vec{w}$, as stated in Definition 14. In this way, the continuous variational formulations in Eqs.(3.3)-(3.5) are converted into a discrete quadratic optimization problem, whose solution determines the RBF model for the linear elliptic BVPs. We recall the unconstrained minimization problem, as stated in Definition 14:

$$\min_{\vec{w} \in \mathbb{R}^N} \left\{ \frac{1}{2} \vec{w}^\top A_1\, \vec{w} - \vec{f}^\top A_2\, \vec{w} + \frac{\beta}{2} \left\| A_3\, \vec{w} - \vec{g} \right\|_2^2 \right\},$$

where the stiffness matrix is defined as either in Eq. (2.27) or (2.33). The matrices A_2 and A_3 are defined as stated in Eqs. (2.29)-(2.31), respectively.

The minimizer $\vec{w}^*$ satisfies the following equation:

$$A_{\text{elliptic}}\, \vec{w}^* = \vec{b}_{\text{elliptic}}, \tag{3.13}$$

where

$$A_{\text{elliptic}} \coloneqq A_1 + \beta\, A_3^\top\, A_3 \in \mathbb{R}^{N \times N}, \quad \vec{b}_{\text{elliptic}} = A_2^\top\, \vec{f} + \beta\, A_3^\top\, \vec{g}. \tag{3.14}$$

However, the choice of the shape parameter b_{RBF} in the kernel function ϕ affects the condition number of the stiffness matrix in Eq. (2.27). To resolve this issue, we compute

the TSVD solution using the pseudo-inverse, as stated in Definition 13. This is provided by:

$$\vec{w}_\tau = A^{\dagger}_{\text{elliptic},\tau}\, \vec{b}_{\text{elliptic}} = V_p \Sigma_p^{-1} U_p^{\top}\, \vec{b}_{\text{elliptic}}, \tag{3.15}$$

where

$$\Sigma_p^{-1} = \operatorname{diag}\Big(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_p}\Big), \quad \text{with } \sigma_p > \sigma_1\, \tau \geq \sigma_{p+1}. \tag{3.16}$$

Furthermore, we require $A_{\text{elliptic}} = U_p \Sigma_p V_p^{\top}$, which is the SVD of the corresponding A_{elliptic}, as stated in Definition 11. According to [5], with a suitable choice of parameters for RBF, the TSVD solution yields an accurate approximation given by:

$$\hat{u}_{\text{RBF},\tau}(\vec{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_{(\tau),i}\, \phi\left(\|\vec{x} - \vec{c}_i\|_2\right). \tag{3.17}$$

Hence, the Algorithm 1 summarizes the schematic solver for linear elliptic BVPs and also establishes the foundation for the subsequent development of obstacle-type PDEs. The algorithm is given by:

Algorithm 1: Stable RBF weak-form solver for variational elliptic PDEs using TSVD.

Input : $\phi(\cdot)$ (kernel function); $\{\vec{c}_i\}_{i=1}^{N} \subseteq \overline{\Omega}_T$ (sampled center nodes from the expansion domain); $\{\vec{x}_j^{(\Omega)}\}_{j=1}^{m_\Omega}$ (interior nodes); $\{\vec{x}_n^{(\partial\Omega)}\}_{n=1}^{m_{\partial\Omega}}$ (boundary nodes); $f(\cdot)$, $g(\cdot)$ (source/boundary data); $\beta > 0$ (boundary penalty parameter); $\tau > 0$ (truncation tolerance)

Output : Weight vector $\vec{w}_\tau \in \mathbb{R}^N$; RBF approximation $\hat{u}_{\mathrm{RBF},\tau}(\vec{x}) \in \mathbb{R}$.

Assemble discrete matrices;

Compute $A_1 \in \mathbb{R}^{N\times N}$ using Eqs. (2.27) or (2.33);

Compute $A_2 \in \mathbb{R}^{m_\Omega \times N}$ and $A_3 \in \mathbb{R}^{m_{\partial\Omega}\times N}$ using Eqs. (2.29)–(2.31);

Form vectors $\vec{f}$, $\vec{g}$;

Build the normal-equation system according to Eq. (3.13);

$A_{\mathrm{elliptic}} \leftarrow A_1 + \beta A_3^\top A_3$;

$\vec{b}_{\mathrm{elliptic}} \leftarrow A_2^\top \vec{f} + \beta A_3^\top \vec{g}$;

Stabilize via TSVD;

Compute the SVD: $A_{\mathrm{elliptic}} = U\Sigma V^\top$;

Identify the index set $\mathcal{T}_\tau = \{i : \sigma_i > \tau\sigma_1\}$;

Form the pseudo-inverse of the TSVD: Σ_p^{-1}, where $(\Sigma_p^{-1})_{i,i} = 1/\sigma_i$ if $i \in \mathcal{T}_\tau$ and 0 otherwise as stated in Eq. (3.16);

Form the pseudo-inverse matrix $A^\dagger_{\mathrm{elliptic},\tau} \leftarrow V_p\,\Sigma_p^{-1}\,U_p^\top$;

Compute the weight coefficients in RBF;

$\vec{w}_\tau \leftarrow A^\dagger_{\mathrm{elliptic},\tau}\,\vec{b}_{\mathrm{elliptic}}$ using Eq. (3.15);

return;

$\vec{w}_\tau$ and $\hat{u}_{\mathrm{RBF},\tau}(\vec{x}) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_{(\tau),i}\,\phi(\|\vec{x} - \vec{c}_i\|_2)$ as in Eq. (3.17);

3.3.2 Algorithm for Obstacle-Type Boundary Value Problem

Algorithm 1 provides a stable weak-form RBF solver for linear elliptic BVPs. We now generalize this framework to obstacle-type PDEs, in which both the state variable and the a priori unknown moving interface must be determined simultaneously. This class of problems introduces an additional level of difficulty due to the coupling between the governing differential operator and the geometry of the free-boundary region. To address this difficulty, we extend the previously introduced variational mesh-free approximations and adapt them to a formulation that can simultaneously approximate the solution and the free boundary. As before, we select m collocation points from the closure of the domain $\overline{\Omega}$, consisting of m_Ω interior points $\vec{x}_1^{(\Omega)}, \ldots, \vec{x}_{m_\Omega}^{(\Omega)}$ and $m_{\partial\Omega}$ boundary points $\vec{x}_1^{(\partial\Omega)}, \ldots, \vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}$. The resulting discrete form of Eq. (3.7) can then be expressed as:

$$\widehat{\mathcal{J}}_{\mu,\beta,\text{obstacle}}(\hat{u}) = \frac{|\Omega|}{m_\Omega} \sum_{j=1}^{m_\Omega} \left[\frac{1}{2} \|\nabla \hat{u}(\vec{x}_j^{(\Omega)})\|_2^2 + \mu \max\left(\psi(\vec{x}_j^{(\Omega)}) - \hat{u}(\vec{x}_j^{(\Omega)}),\, 0\right) \right] + \frac{\beta\,|\partial\Omega|}{2\,m_{\partial\Omega}} \sum_{n=1}^{m_{\partial\Omega}} \left(\hat{u}(\vec{x}_n^{(\partial\Omega)}) - g(\vec{x}_n^{(\partial\Omega)})\right)^2. \tag{3.18}$$

Using the RBF representation for $\hat{u}$, as stated in Definition 7 and by Definition 15, the discrete form of the obstacle problem in Eq. (3.18) becomes:

$$\widehat{\mathcal{J}}_{\mu,\beta,\text{obstacle}}(\vec{w}) = \frac{1}{2}\,\vec{w}^\top A_1\,\vec{w} + \mu \|(\vec{\psi} - A_2\,\vec{w})_+\|_1 + \frac{\beta}{2}\|A_3\,\vec{w} - \vec{g}\|_2^2, \tag{3.19}$$

where the matrices A_1, A_2 and A_3 are defined in the same way as in Eqs. (2.27), (2.29), and (2.31), respectively. The vector of the obstacle function is defined as follows:

$$\vec{\psi} = \frac{|\Omega|}{m_\Omega} \begin{bmatrix} \psi\left(\vec{x}_1^{(\Omega)}\right) \\ \vdots \\ \psi\left(\vec{x}_{m_\Omega}^{(\Omega)}\right) \end{bmatrix} \in \mathbb{R}^{m_\Omega}. \tag{3.20}$$

The unconstrained optimization problem involving the ℓ^1-norm is convex but non-differentiable. We address this problem using the ADMM, as outlined in Section 2.7. Specifically, we introduce the slack variable $\vec{v} = \vec{\psi} - A_2\,\vec{w}$, which reformulates the original problem as an equivalent constrained optimization problem:

$$\begin{cases} \min\limits_{\vec{w},\vec{v}} \widehat{\mathcal{J}}_{\mu,\beta,\text{obstacle}}(\vec{w},\vec{v}) \\ \widehat{\mathcal{J}}_{\mu,\beta,\text{obstacle},}(\vec{w},\vec{v}) = \frac{1}{2}\vec{w}^\top A_1 \vec{w} + \mu\|\vec{v}_+\|_1 + \frac{\beta}{2}\|A_3\vec{w} - \vec{g}\|_2^2 \\ \text{s.t.} \quad \vec{v} - \vec{\psi} + A_2\vec{w} = 0. \end{cases} \tag{3.21}$$

According to Definition 20, the associated augmented Lagrangian functional, parameterized by a penalty coefficient $\rho > 0$ and a scaled dual variable $\vec{z}$, is defined as:

$$\widehat{L}_\rho(\vec{w}, \vec{v}, \vec{z}) = \frac{1}{2}\,\vec{w}^\top A_1 \vec{w} + \frac{\beta}{2}\|A_3\vec{w} - \vec{g}\|_2^2 + \mu\|\vec{v}_+\|_1 + \frac{\rho}{2}\big\|\vec{v} - \vec{\psi} + A_2\vec{w} + \vec{z}\big\|_2^2 - \frac{\rho}{2}\|\vec{z}\|_2^2. \quad (3.22)$$

Following Definition 21, the ADMM algorithm iteratively updates the primal variables $\vec{w}, \vec{v}$ and the dual variable $\vec{z}$ at each iteration accordingly:

$$\vec{w}^{(i+1)} = \arg\min_{\vec{w}} \left\{ \frac{1}{2}\vec{w}^\top A_1 \vec{w} + \frac{\beta}{2}\|A_3\vec{w} - \vec{g}\|_2^2 + \frac{\rho}{2}\big\|\vec{v}^{(i)} - \vec{\psi} + A_2\vec{w} + \vec{z}^{(i)}\big\|_2^2 \right\}, \quad (3.23)$$

$$\vec{v}^{(i+1)} = \arg\min_{\vec{v}} \left\{ \mu\|\vec{v}_+\|_1 + \frac{\rho}{2}\big\|\vec{v} - \vec{\psi} + A_2\vec{w}^{(i+1)} + \vec{z}^{(i)}\big\|_2^2 \right\}, \quad (3.24)$$

$$\vec{z}^{(i+1)} = \vec{z}^{(i)} + \big(\vec{v}^{(i+1)} - \vec{\psi} + A_2\vec{w}^{(i+1)}\big). \quad (3.25)$$

The iterate $\vec{w}^{(i+1)}$, which solves the $\vec{w}$-subproblem, is characterized by the optimality condition associated with the above quadratic minimization problem.

$$\Big(A_1 + \beta A_3^\top A_3 + \rho A_2^\top A_2\Big)\,\vec{w}^{(i+1)} = \beta A_3^\top \vec{g} + \rho A_2^\top \Big(\vec{\psi} - \vec{v}^{(i)} - \vec{z}^{(i)}\Big).$$

Since the total stiffness matrix is ill-conditioned in the $\vec{w}$-update, we stabilize the linear system by computing the TSVD solution with the parameter τ:

$$\vec{w}_\tau^{(i+1)} = A^\dagger_{\text{obstacle},\tau}\,\vec{b}^{(i)}_{\text{obstacle}}, \quad (3.26)$$

where $A_{\text{obstacle}} \coloneqq A_1 + \beta A_3^\top A_3 + \rho A_2^\top A_2 \in \mathbb{R}^{N\times N}$ and $\vec{b}^{(i)}_{\text{obstacle}} \coloneqq \beta A_3^\top \vec{g} + \rho A_2^\top(\vec{\psi} - \vec{v}^{(i)} - \vec{z}^{(i)}) \in \mathbb{R}^N$. The matrix $A^\dagger_{\text{obstacle},\tau}$ is computed in the same manner as the matrix $A^\dagger_{\text{elliptic},\tau}$ in Eq. (3.15).

Let $\vec{t}^{(i)} \coloneqq \vec{\psi} - A_2\vec{w}^{(i+1)} - \vec{z}^{(i)} \in \mathbb{R}^{m_\Omega}$. Then, for each component index $j = 1, \ldots, m_\Omega$, the update of the variable $\vec{v}$ is subsequently given by

$$v_j^{(i+1)} = \begin{cases} t_j^{(i)}, & t_j^{(i)} < 0, \\ 0, & 0 \le t_j^{(i)} \le \frac{\mu}{\rho}, \\ t_j^{(i)} - \frac{\mu}{\rho}, & t_j^{(i)} > \frac{\mu}{\rho}. \end{cases} \quad (3.27)$$

The primal and dual residuals at iteration $(i+1)$ are defined by

$$\vec{r}^{(i+1)} \coloneqq \vec{v}^{(i+1)} - \vec{\psi} + A_2\vec{w}^{(i+1)}, \quad \vec{s}^{(i+1)} \coloneqq \rho A_2^\top\big(\vec{v}^{(i+1)} - \vec{v}^{(i)}\big),$$

where $\vec{r}^{(i+1)}$ denotes the primal residual and $\vec{s}^{(i+1)}$ denotes the dual residual associated with the $(i+1)$-th iteration.

In accordance with Definitions 22 and 23, the algorithm is terminated once both the primal and dual residuals, $\vec{r}^{(i+1)}$ and $\vec{s}^{(i+1)}$, respectively, satisfy the prescribed relative tolerance criteria $\varepsilon_{\text{primal}}, \varepsilon_{\text{dual}} > 0$.

$$\frac{\|\vec{r}^{(i+1)}\|_2}{\max\left(\|A_2\vec{w}^{(i+1)}\|_2, \|\vec{v}^{(i+1)}\|_2, \|\vec{\psi}\|_2\right)} \leq \epsilon_{\text{primal}}, \tag{3.28}$$

$$\frac{\|\vec{s}^{(i+1)}\|_2}{\|\rho A_2^\top \vec{z}^{(i+1)}\|_2} \leq \epsilon_{\text{dual}}. \tag{3.29}$$

The proposed algorithm for the obstacle problem is given in Algorithm 2. In Section 3.4, we employ numerical experiments for linear elliptic BVPs and obstacle-type PDEs to test the accuracy of Algorithms 1 and 2.

Algorithm 2: Stable ADMM-RBF solver for the classical obstacle problem

Input : $\phi(\cdot)$ (kernel function); $\{\vec{c}_\ell\}_{\ell=1}^{N} \subseteq \overline{\Omega}_T$ (sampled center nodes from the expanded domain); $\left\{\vec{x}_j^{(\Omega)}\right\}_{j=1}^{m_\Omega}$ (interior nodes); $\left\{\vec{x}_n^{(\partial\Omega)}\right\}_{n=1}^{m_{\partial\Omega}}$ (boundary nodes); $f(\cdot)$, $g(\cdot)$ (source/boundary data); $\psi(\cdot)$ (obstacle data); $\beta > 0$ (boundary penalty); $\mu > 0$ (obstacle weight); $\rho > 0$ (ADMM penalty); $\tau > 0$ (truncation tolerance); maxIter (maximum number of iterations); $\varepsilon_{\text{pri}}, \varepsilon_{\text{dual}}$ (tolerances for primal and dual residuals)

Output : Weight vector $\vec{w}_\tau \in \mathbb{R}^N$; RBF approximation $\hat{u}_{\text{RBF},\tau}(\vec{x}) \in \mathbb{R}$

Assemble operators and vectors;

Compute A_1, A_2, A_3 as stated in Eqs. (2.27), (2.29) and (2.31), respectively ;

Form $A_{\text{obstacle}} \leftarrow A_1 + \beta A_3^\top A_3 + \rho A_2^\top A_2$;

Form $\vec{f}$, $\vec{g}$, $\vec{\psi}$;

Initialize $\vec{w}^{(0)}$, $\vec{v}^{(0)}$, $\vec{z}^{(0)}$;

for $i = 0, \ldots,$ maxIter **do**

`// 1.` $\vec{w}$`-update (TSVD solve)`

$\vec{b}_{\text{obstacle}}^{(i)} \leftarrow \beta A_3^\top \vec{g} + \rho A_2^\top (\vec{\psi} - \vec{v}^{(i)} - \vec{z}^{(i)})$;

$\vec{w}_\tau^{(i+1)} \leftarrow A_{\text{obstacle},\tau}^{\dagger}\, \vec{b}_{\text{obstacle}}^{(i)}$ using Eq. (3.26);

`// 2.` $\vec{v}$`-update (positive hard-shrinkage)`

$\vec{t}^{(i)} \leftarrow \vec{\psi} - A_2 \vec{w}^{(i+1)} - \vec{z}^{(i)}$;

foreach $j = 1, \ldots, m_\Omega$ **do**

if $t_j^{(i)} < 0$ **then**

$v_j^{(i+1)} \leftarrow t_j^{(i)}$

else if $0 \le t_j^{(i)} \le \mu/\rho$ **then**

$v_j^{(i+1)} \leftarrow 0$

else

$v_j^{(i+1)} \leftarrow t_j^{(i)} - \mu/\rho$

`// 3. Dual update`

$\vec{z}^{(i+1)} \leftarrow \vec{z}^{(i)} + \vec{v}^{(i+1)} - \vec{\psi} + A_2 \vec{w}^{(i+1)}$;

`// 4. Primal and dual residuals; stopping criteria`

$\vec{r}^{(i+1)} \leftarrow \vec{v}^{(i+1)} - \vec{\psi} + A_2 \vec{w}^{(i+1)}$;

$\vec{s}^{(i+1)} \leftarrow \rho A_2^\top (\vec{v}^{(i+1)} - \vec{v}^{(i)})$;

if $\frac{\|\vec{r}^{(i+1)}\|_2}{\max\{\|A_2\vec{w}^{(i+1)}\|_2\, \|\vec{v}^{(i+1)}\|_2 \|\vec{\psi}\|_2\}} \le \varepsilon_{\text{pri}} \;\wedge\; \frac{\|\vec{s}^{(i+1)}\|_2}{\|\rho A_2^\top \vec{z}^{(i+1)}\|_2} \le \varepsilon_{\text{dual}}$ **then**

break

return $\vec{w}_\tau^{(i+1)} \in \mathbb{R}^N$, $\hat{u}_\tau(\vec{x}) = \dfrac{1}{\sqrt{N}} \displaystyle\sum_{l=1}^{N} w_{(\tau),l}^{(i+1)}\, \phi(\|\vec{x} - \vec{c}_l\|_2)$;

3.4 Numerical Experiment

In the following experiments, we evaluate the accuracy, stability, and computational efficiency of the RBF-based variational method, as stated in Definition 7. The numerical experiments encompass linear elliptic PDEs, specifically Poisson and reaction–diffusion BVPs, as well as the obstacle-type PDEs. All tests are conducted under inhomogeneous Dirichlet boundary conditions. The Gaussian kernel specified in Table 3.1 is utilized with a shape parameter $b_{\mathrm{RBF}} > 0$ throughout all experiments, except in cases where the objective is to compare the performance of different kernel functions. We employ a dimension-dependent scaling for the kernel shape parameter. Specifically, in the one-dimensional setting, we set $b_{\mathrm{RBF}} = c\,N$, whereas in the two-dimensional setting, we choose $b_{\mathrm{RBF}} = c\sqrt{N}$. Here, $c = c(T,\tau)$ is selected according to Eq. (3.10); see [5]. In particular, we recall the formula for the shape-factor multiplier $c(T,\tau)$, which is given by:

$$c(T,\tau) = \min\left\{1,\ \frac{\pi}{T\sqrt{2\,\log\left(1+\tau^{-2}\right)}}\right\},$$

As discussed in Section 3.2.2, the multiplier $c(T,\tau)$ governs the trade-off between approximation accuracy and numerical stability through the shape parameter b_{RBF}. In particular, larger values of c improve conditioning but may reduce approximation quality, whereas smaller values enhance accuracy at the expense of stability. We adopt Eq. (3.10) from [5] as a reference to assess its performance in the numerical experiments of Section 3.4, while recognizing that it does not represent the optimal choice for solving the variational PDEs.

In [5], the centers are not strictly constrained to be within $\overline{\Omega}$; instead, they are sampled uniformly from an enlarged interval $[q_{\mathrm{mid}} - T\,L_{\mathrm{half}},\ q_{\mathrm{mid}} + T\,L_{\mathrm{half}}]$, as specified in Setting 1. Sampling in an extended domain beyond the physical domain reduces boundary effects on the approximation and enhances the numerical stability of the resulting collocation system.

For each problem, we sample $m = \zeta\,N$ collocation points, which include m_Ω interior points and $m_{\partial\Omega}$ boundary points, where ζ is an oversampling scale and N is the number of RBF centers. In particular, the total number of collocation points is expressed as $m = m_\Omega + m_{\partial\Omega} = \zeta\,N$. In the one-dimensional setting, the boundary points consist of two endpoints b_1, b_2. Hence, the number of interior points is $m_\Omega = m - 2$. However, in the two-dimensional setting, we uniformly sample $\sqrt{\zeta\,N}$ points on each side of the boundary domain $\partial\Omega$, which yields $m_{\partial\Omega} = 4\sqrt{\zeta\,N}$. For interior collocation points, we uniformly sample $m_\Omega = \zeta\,N - 4\sqrt{\zeta N}$ points of the domain Ω.

To compare accuracy and facilitate performance benchmarking, we report the following

Parameter	**Symbol**	**Value/Formula**
Number of basis functions	N	`problem-dependent`
Oversampling factor	ζ	$2\,(1D)$; $4\,(2D)$
Total number of all collocation points	m	$\zeta \times N$
Half-length of interval $[b_1, b_2]$	L_{half}	$L_{\text{half}} = \dfrac{b_2 - b_1}{2}$
Midpoint of the interval $[b_1, b_2]$	q_{mid}	$q_{\text{mid}} = \dfrac{b_1 + b_2}{2}$
Domain expansion factor	T	centers are sampled on $\left[q_{\text{mid}} - TL_{\text{half}},\, q_{\text{mid}} + TL_{\text{half}}\right]$ in 1D
Truncation threshold value	τ	10^{-15}
Shape-factor multiplier	c	$\min\left\{1,\ \dfrac{\pi}{T\sqrt{2\log(1+\tau^{-2})}}\right\}$ as in [5]
Kernel shape parameter	b_{RBF}	$c\,N$ in 1D; $c\,\sqrt{N}$ in 2D
Maximum number of iterations in the ADMM	`maxIter`	50,000
Obstacle penalty weight	μ	`problem-dependent`
Boundary-condition penalty term	β	`problem-dependent`
ADMM penalty parameter	ρ	`problem-dependent`

Table 3.1: Summary of the parameters used in the numerical implementation of the variational RBF algorithm. The table reports the approximation, stabilization, and penalty parameters employed throughout the computational experiments.

error metric. The relative ℓ^2 error, which measures the fidelity of the approximation, is defined as

$$E(N) := \frac{\|\hat{u}_N - u_{\text{exact}}\|_2}{\|u_{\text{exact}}\|_2}, \tag{3.30}$$

where $\hat{u}_N$ denotes the numerical approximation of the optimization problem; see Eq. (3.17).

Furthermore, we compute the local estimate of the convergence order p, defined as

$$p := \log\left(\frac{E\,(N/2)}{E(N)}\right), \tag{3.31}$$

where $E(N)$ denotes the relative error associated with the number of RBF centers N, as mentioned in Eq. (3.30).

We compare the proposed RBF-based variational solver with finite-difference ADMM, neural network, and Galerkin methods in the linear elliptic BVPs. For the obstacle-type PDEs, we compare the proposed RBF-based variational method, FDM solvers, with ADMM, proximal gradient descent (PGD), and neural networks. For the finite-difference benchmark, we use the method of [107] to solve the variational form of the Poisson problem under the same Dirichlet boundary conditions. In the two-dimensional setting, we compare with finite-difference and solve the optimization problems with PGD as in [6]. The packages for finite-difference methods with ADMM[1] and PGD[2] can be accessed online, respectively. For the neural-network benchmark, we use the physics-informed neural networks (PINNs)[3] from [88, 89, 87], in which the network parameters are trained by minimizing the corresponding variational loss from Section 2.6. In one-dimensional experiments, we set the fully-connected neural network to consist of one hidden layer with the hyperbolic tangent activation function using N as the number of neurons, followed by a linear output layer. In two-dimensional experiments, we consider two architectures: the same one-hidden-layer architecture as in the one-dimensional case and a deeper network with three connected layers that have N neurons each, again using `tanh` activations. In the Galerkin[4] method, we solve the weak form of the boundary value problems using finite-element discretization; a common open-source implementation is `FEniCSx` in [25]. We use N as the number of mesh resolution in Galerkin method. For reproducibility, we report

[1] FDM with ADMM implementation is obtained from https://github.com/jwcalder/MinimalSurfaces

[2] FDM with PGD implementation is obtained from https://github.com/lumeilevel/MGProx

[3] PINNs implementation is obtained from https://github.com/maziarraissi/PINNs

[4] Galerkin package is implemented in https://github.com/METHODS-Group/ProximalGalerkin

the main implementation parameters for each method. In the neural network, we use the same parameter settings as in [88] with a learning rate of 10^{-2}, the Adam optimizer, and a hyperbolic tangent activation function. For the finite-difference and Galerkin methods, the stopping criterion is chosen so that the relative error is below a prescribed tolerance. These numerical experiments provide a quantitative assessment of convergence for linear elliptic PDEs and obstacle-type BVPs. We note that the method of [5] is closely related to the RBF approximation considered in this chapter. However, the method in [5] is formulated for the strong-form solution of PDEs, whereas the present chapter is based on the variational formulations. This distinction is important for obstacle problems, since the obstacle problem is naturally posed as a variational inequality and its solution theory is most naturally expressed in a weak form. The variational formulation also allows the obstacle constraint to be incorporated directly into an energy-minimization framework. For this reason, a direct numerical comparison with [5] is not considered in this chapter. The implementation details, parameter choices, and discretization setting of the strong-form method in [5] are not identical to the variational RBF framework.

3.4.1 Linear Elliptic Boundary Value Problem

In this subsection, we present numerical experiments for linear elliptic PDEs to assess the performance of the variational RBF method introduced in Algorithm 1. The primary objective is to investigate the accuracy, stability, and robustness of the proposed mesh-free approximation framework when applied to problems characterized by different differential operators and distinct qualitative solution behaviors.

The test problems considered comprise the Poisson equation and the reaction-diffusion equation, which are presented in further detail later. Those BVPs serve as representative benchmark problems for assessing the performance of the proposed method. For each problem, we compare the computed numerical solution with the corresponding analytical solution and quantify the discrepancy using the error measure defined in Eq. (3.30). This procedure demonstrates the capability of the variational RBF framework to produce highly accurate approximations for linear elliptic boundary value problems.

Poisson equation

In Section 3.4.1, we investigate a class of elliptic PDEs posed on a bounded, closed domain $\overline{\Omega} \subset \mathbb{R}$. Specifically, we consider the following Poisson boundary value problem

with inhomogeneous Dirichlet boundary conditions:

$$\begin{cases} -\Delta u = f(x), \quad x \in \Omega = (0,1), \\ u(0) = a_1 \\ u(1) = a_2, \end{cases} \tag{3.32}$$

For a detailed understanding, we refer readers to the corresponding experiment presented in [61]. The analytical solution, which corresponds to the Poisson BVPs in Eq. (3.32), is given by:

$$\begin{cases} u(x) = \sin(\pi\, x) - (a_1 - a_2)\, x + a_1, \qquad x \in \Omega = (0,1), \\ u(0) = a_1,\ u(1) = a_2. \end{cases} \tag{3.33}$$

where the external forcing function is set at $f(x) = \pi^2 \sin(\pi\, x)$ and the boundary values are set at $a_1 = -1.0$, $a_2 = -\frac{3}{2}$, respectively.

Our method employs the RBF representation introduced in Definition 7. The resulting linear system is regularized by TSVD with a truncation threshold of $\tau = 10^{-15}$, and the boundary contribution is weighted by a penalty parameter $\beta = 3 \times 10^5$. The pointwise approximation error remains uniformly small, as evidenced by Figure 3.1(a). In particular, the error $|u_{\text{exact}}(x) - \hat{u}_\tau(x)|$ exhibits oscillatory behavior with values ranging between 10^{-9} and 10^{-12}. The magnitudes at the interior collocation points decay to approximately 10^{-14} in the vicinity of the boundary regions.

Figure 3.1(b) provides a comparison of the numerical schemes used to solve the Poisson equation, including the proposed RBF-based approach, a neural network-based method, a Galerkin formulation, and a finite-difference discretization. For the benchmark results reported in Figures 3.1(b,c), all methods are run with the following hyperparameter settings, as detailed in Section 3.4. The proposed RBF scheme employs a Gaussian kernel with $\zeta = 2, T = 8, \tau = 10^{-15}$ and $\beta = 3 \times 10^5$. The neural-network baseline consists of a single hidden layer with a hyperbolic tangent activation `tanh()` and a linear output layer. It uses N neurons, chosen such that the number of trainable parameters is comparable to that of the proposed RBF method. The Galerkin approach relies on the standard weak formulation with an equivalent nominal number of basis functions. The finite-difference scheme is implemented on a uniform grid with a similar number of grid points as in the RBF.

In Figure 3.1(b), it is evident that the RBF method yields a rapid decrease in the relative ℓ^2 error as the number of basis functions N increases. The error baseline of the RBF exhibits fast convergence. The Galerkin and finite-difference methods exhibit the expected second-order accuracy, while the neural-network converges slowly and stagnates

at higher resolutions. Figure 3.1(c) confirms the trends by showing the local estimated convergence order p. The RBF approach achieves an approximate convergence order of 2.7, exceeding the second-order convergence rates of the Galerkin and finite-difference schemes.

Reaction-diffusion equation

In this section 3.4.1, we apply the Algorithm 1 to solve the Reaction-Diffusion BVPs. The PDEs to the Reaction-Diffusion equation is given by:

$$\begin{cases} -\Delta u(x) + u(x) = f(x), & \text{in } \Omega = (0, L), \\ u(0) = u(L) = 0, & \text{on } \partial\Omega, \end{cases} \tag{3.34}$$

where $L = 5$ represents the length of soliton waves.

In this experiment, we consider a weak solution to the reaction-diffusion equation in one-dimensional space $[0, L]$, as stated in [61]. The forcing function is given by:

$$f(x) = 50 \sin(\pi\, x), \quad \forall x \in (0, L). \tag{3.35}$$

The corresponding weak solution is presented as follows:

$$u_{\text{exact}}(x) = \frac{50}{1 + \pi^2} \sin(\pi\, x), \quad \forall x \in [0, L]. \tag{3.36}$$

We maintain a similar setup to evaluate the accuracy, stability, and convergence behavior of numerical methods. Because the reaction-diffusion PDEs admits the weak solution in Eq. 3.36, it serves as a benchmark test for validating Algorithm 1 and for comparing it with approximation methods, including finite-difference, Galerkin, and neural-networks.

According to Figure 3.2(a), we verify Algorithm 1 using the RBF approximation with resolution parameters $(N, m) = (4096, 8192)$. The relative error $E(N) = 7.697 \times 10^{-9}$ indicates that the RBF method accurately approximates the weak solution of the reaction-diffusion equation. The error distribution is shown in Figure 3.36(a), indicating accuracy, with errors remaining uniformly around 10^{-10} across the domain.

In Figure 3.36(b)(c), we examine the convergence behavior of the RBF method. For hyperparameter settings, we choose $T = 2.0, \zeta = 2, \tau = 10^{-15}, \beta = 10^6$ in this experiment. We discuss the details of choosing those parameters in Section 3.5. Particularly, it consistently achieves a low relative error compared to the Galerkin, finite-difference, and

neural network methods. Since the reaction-diffusion equation in Eq. (3.34) possesses a weak solution, the Galerkin method and the proposed RBF-variational approach yield a more accurate approximation to the solution in Eq. 3.36 than other methods. Hence, the estimated convergence order stabilizes around 2.4 for the RBF method, similar to the rate of convergence of the Galerkin method, while other methods exhibit larger fluctuations. In Section 3.4.2, we consider Algorithm 2 to solve free-BVPs, which also validates the convergence behavior of the proposed RBF method.

3.4.2 Obstacle-Type Boundary Value Problem

In this section, we validate the efficiency of Algorithm 2 to solve the obstacle-type BVP and test the benchmark performance against other numerical methods. We start with the setting of one-dimensional PDEs in Section 3.4.2. In Section 3.4.2, we examine the accuracy and scalability of the algorithm concerning the radially symmetric obstacle problem. The hyperparameters in the variational form in Section 2.3 are set according to Table 3.1. The parameters (μ, ρ) in Eq. (3.18) are tuned according to [107].

We vary the number N of basis functions to assess accuracy, tracking relative ℓ^2 error, and monitor ADMM convergence through the relative primal and dual residuals in Eq. (3.28), respectively. Generally, we set the tolerances as $\epsilon_{tol} = \epsilon_{\text{primal}} = \epsilon_{\text{dual}}$, depending on the desired accuracy for the experiment.

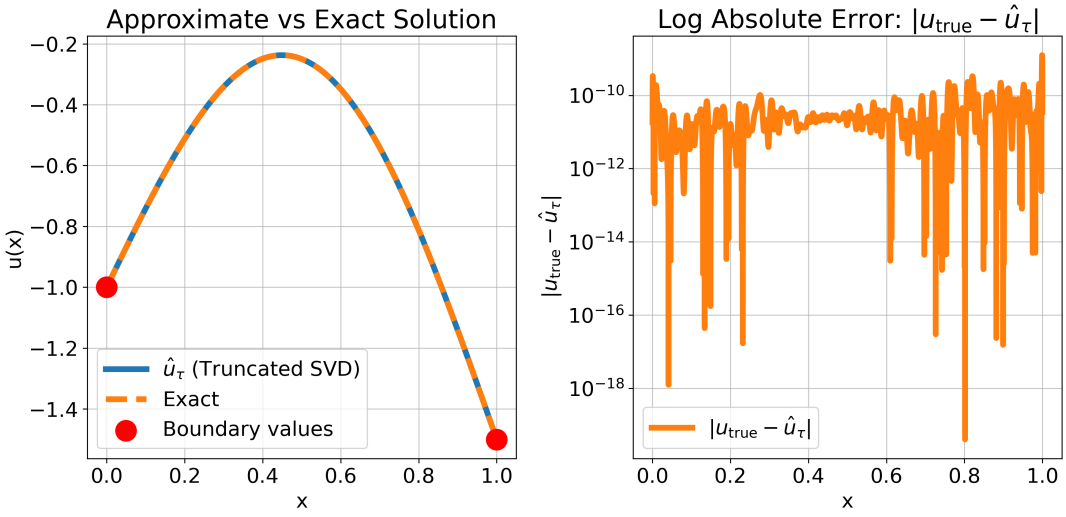
One-dimensional obstacle problem

We address the one-dimensional obstacle problem using the ADMM-based penalized formulation described in Section 3.2, employing the following piecewise-defined obstacles ψ_1 (single bump) and ψ_2 (two bump), respectively:

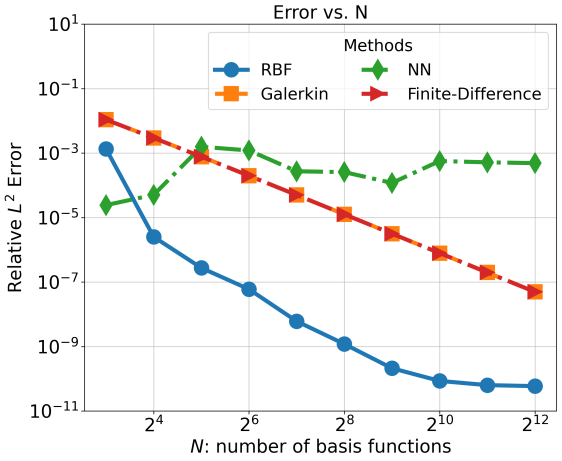
$$\psi_1(x) = \begin{cases} 100\,x^2, & x \in [0, 0.25], \\ 100\,x(1-x) - 12.5, & x \in (0.25, 0.5], \\ \psi_1(1-x), & x \in (0.5, 1], \end{cases} \tag{3.37}$$

and

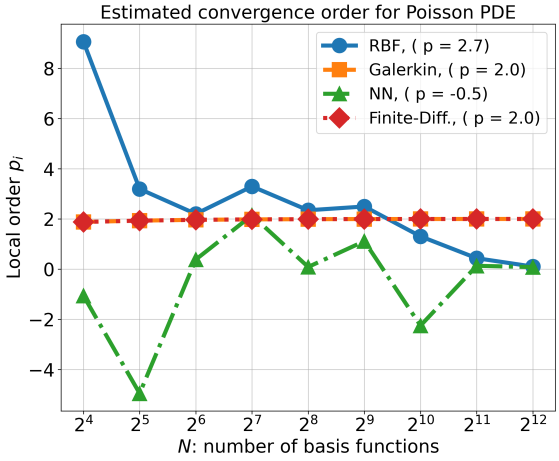
$$\psi_2(x) = \begin{cases} 10\sin(2\pi x), & x \in [0, 0.25], \\ 5\cos\big(\pi(4x-1)\big) + 5, & x \in (0.25, 0.5], \\ \psi_2(1-x), & x \in (0.5, 1]. \end{cases} \tag{3.38}$$



(a) **Numerical approximation for the Poisson equation. The relative error is** $E(N) = 7.419 \times 10^{-11}$. The resolution parameters are $(N, m) = (4096, 8192)$, with $\zeta = 2$, $T = 8$, $\tau = 10^{-15}$, and $\beta = 3 \times 10^5$.

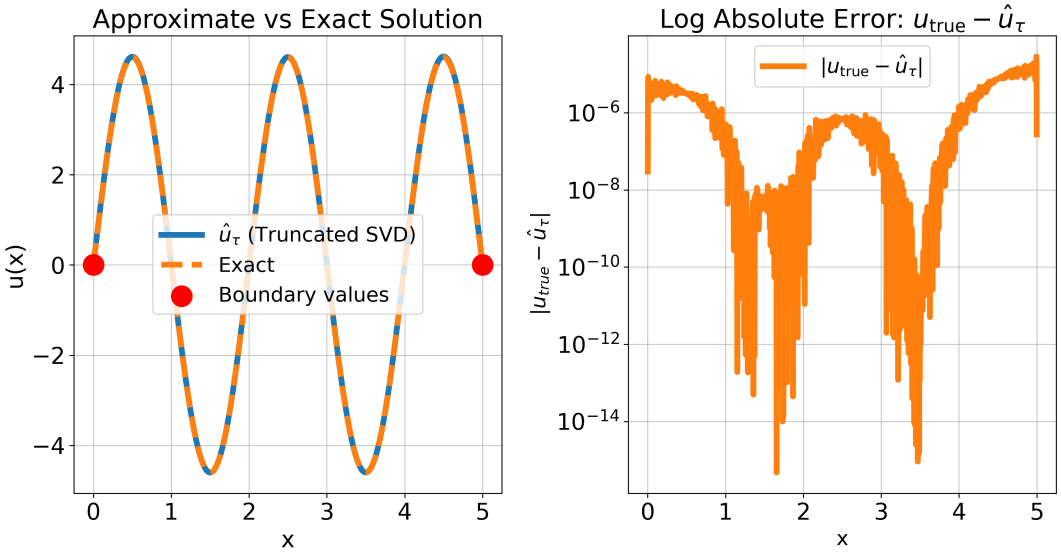


(b) Relative ℓ^2 error versus N.

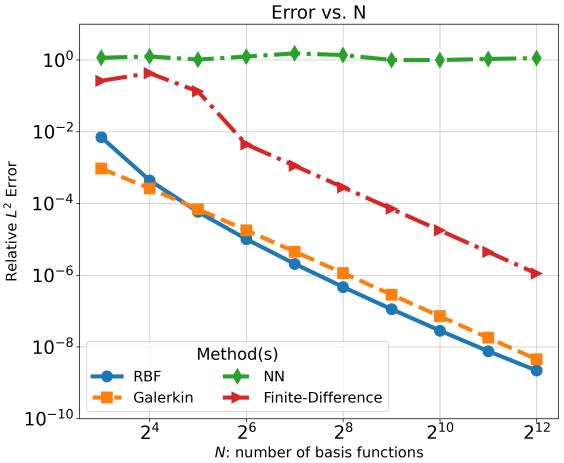


(c) Local estimated convergence order p.

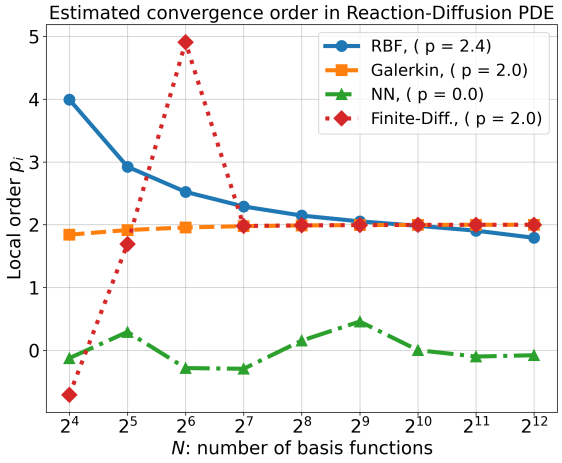
Figure 3.1: **Comparison of different methods for the one-dimensional Poisson equation** $-\Delta u = f(x)$. The left panel shows the relative ℓ^2 error versus the number of basis functions N, and the right panel shows the local estimated convergence order p. The RBF method gives the smallest error and the fastest convergence, while the Galerkin and finite-difference methods exhibit approximately second-order behavior.



(a) **Numerical approximation of the Reaction-diffusion equation**. The relative error is $E(N) = 7.697 \times 10^{-9}$. The resolution parameters are $(N, m) = (4096, 8192)$, with $\zeta = 2$, $T = 2$, $\tau = 10^{-15}$, and $\beta = 10^6$.



(b) Relative ℓ^2 error versus N.



(c) Local estimated convergence order p.

Figure 3.2: **Comparison of different numerical methods for the one-dimensional reaction-diffusion equation** $-\Delta u + u = f(x)$. The left panel shows the relative ℓ^2 error versus the number of basis functions N, while the right panel presents the local estimated convergence order p_i.

For each obstacle problem, the exact solutions are given as follows:

$$u_{1,\text{exact}}(x) = \begin{cases} (100 - 50\sqrt{2})\, x, & x \in [0, \frac{1}{2\sqrt{2}}], \\ 100\, x(1-x) - 12.5, & x \in (\frac{1}{2\sqrt{2}}, 0.5], \\ u_{1,\text{exact}}(1-x), & x \in (0.5, 1], \end{cases} \tag{3.39}$$

$$u_{2,\text{exact}}(x) = \begin{cases} 10\sin(2\pi x), & x \in [0, 0.25], \\ 10, & x \in (0.25, 0.5], \\ u_{2,\text{exact}}(1-x), & x \in (0.5, 1]. \end{cases} \tag{3.40}$$

We test our Algorithm 2 on the unit domain $[0, 1]$ for Eqs. (3.37) and (3.38). These admit the exact solutions $u_{1,\text{exact}}$ and $u_{2,\text{exact}}$, which are provided by Eqs. (3.39) and (3.40), respectively. We specify the choice of hyperparameters as follows. The boundary penalty weight β is set at 10^6 for the single bump problem, while it is set at 10^8 for the two bump problem. The expansion factor T is chosen to be 2.0 for the single bump; for the two bump problem, it is set at $T = 3.0$. For truncation threshold values, we let $\tau = 10^{-15}$. We refer readers to Section 3.5 for a detailed understanding and discussion of the robustness and sensitivity of the parameters. As mentioned in the beginning, the parameters for the variational form and ADMM are set at $(\mu, \rho) = (300, 45)$ for the single bump problem, while for the two bump problem, they are set at $(\mu, \rho) = (2.5 \times 10^4, 250)$ according to [107]. For the tolerance in ADMM, we let $\varepsilon_{tol.} = \varepsilon_{\text{primal}} = \varepsilon_{\text{dual}} = 10^{-6}$. Significantly, we use the kernel shape $b = c\, N$, where c is determined in Eq. (3.10).

In Figures 3.3-3.4, we approximate the solutions of the obstacle problem using resolutions $(N, m) = (1024, 2048)$. In the ADMM, convergence of the single bump is achieved after 7234 iterations, while for the two bump problem, it takes $17,796$ iterations. The relative errors for the single and two bump problems are $E(N) = 1.904 \times 10^{-12}$ and 6.654×10^{-10}, respectively, indicating that the RBF approximates the solutions accurately. The absolute pointwise errors in Figures 3.3-3.4 demonstrate that the approximations are well-aligned with the exact solutions.

Similarly to Section 3.4.1, we compare the performance of the RBF method with others for different values of basis functions N. From Figure 3.5(a)(b), it is indicated that the proposed RBF method outperforms others, achieving a lower machine error than the FDM, ADMM, PGD, and PINNs with depths and widths in the architecture. As shown in Figure 3.5(c)(d), the proposed RBF method is able to converge faster than the other comparison methods. Finite-difference (ADMM, PGD) methods exhibit slower convergence rates and require finer grid resolutions. The one-dimensional obstacle experiments set our baseline for validating Algorithm 2. In the next step, we determine the performance and efficiency of our proposed RBF method in two-dimensional settings.

Two-dimensional obstacle problem

In this experiment, we test the performance of our proposed method in Algorithm 2 using the radially symmetric dome obstacle problem from [107, 125]. The interior domain is $\Omega = (0,1)^2$. First, we consider an obstacle given by a single dome on the unit square, centered at $(\frac{1}{2}, \frac{1}{2})$, defined by

$$\psi(x,y) = \begin{cases} 1 - \dfrac{r(x,y)^2}{r_c^2}, & \text{if } r(x,y) \le r_c, \\ 0, & \text{if } r(x,y) > r_c, \end{cases} \tag{3.41}$$

where the inner radius is $r_c \coloneqq \frac{1}{2}$, and the distance from the center is given by $r(x,y) \coloneqq \sqrt{(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2}$.

The linear obstacle problem admits the following radially symmetric analytical solution:

$$u_{\text{exact}}(x,y) = \begin{cases} \left(1 - \frac{r(x,y)^2}{r_c^2}\right), & r(x,y) \le r_* \\ 1 - \frac{r_*^2}{r_c^2}\left(1 + 2\log\left(\frac{r(x,y)}{r_*}\right)\right), & r_* < r(x,y), \end{cases} \tag{3.42}$$

where the contact radius $r_* \approx 0.260197$ is determined by solving the following equation:

$$\frac{r_*^2}{r_c^2}\left(1 + 2\log\left(\frac{1}{r_*}\right)\right) = 1, \quad \text{with } 0 < r_* < r_c \tag{3.43}$$

For the boundary conditions, we impose $g(x,y) = u_{\text{exact}}(x,y)$ for every $(x,y) \in \partial\Omega$. We then randomly sample m grid points (x,y) and let N denote the total number of basis functions. As illustrated in Figure 3.6, we take $N = 400$ basis functions overall, which, for $\zeta = 4$, yield $m = 1{,}600$ collocation points over the domain $\overline{\Omega}$. A key aspect of our approach is the use of linear scaling for the kernel parameter, given by $b = c\sqrt{N}$, where $c = c(T,\tau)$ is listed in Table 3.1. This scaling ensures that the effective kernel length scale remains proportional to the fill distance h_Ω as the number of basis function N increases. Furthermore, we let $T = 3, \tau = 10^{-15}, \beta = 10^6$. The ADMM parameters are set at $(\mu, \rho) = (10/h^2, 20.3)$, where $h = \frac{1}{N}$ is the mesh size of the system.

The dome obstacle function $\psi(x,y)$ is shown on the left in Figure 3.6. As illustrated in Figure 3.7(a)(b), the proposed RBF method reproduces the solution with high accuracy after 4 016 iterations. The relative ℓ^2 error is $E(N) = 4.681 \times 10^{-11}$, confirming that the exact solution $u_{exact}(x,y)$ in Eq. (3.42) is recovered. Both the relative primal and dual residuals fall below the tolerance $\epsilon_{\text{tol.}} = 10^{-6}$, indicating convergence of the ADMM

algorithm in Figure 3.7(c). The absolute pointwise error shows that it remains under 10^{-6} in the interior and is mainly concentrated near the boundaries where the solution decays, as shown in Figure 3.7(d).

According to Figure 3.8(a), the variational RBF method distinctly surpasses both the FDM and neural-network methods, achieving faster convergence and the smallest computational errors. More specifically, as discussed in Section 3.4.2, the proposed RBF approach converges faster than the alternative techniques, displaying monotone convergence in the error, as illustrated in Figure 3.8(b). In general, we find that the high accuracy approximation for the obstacle-type PDEs, especially the RBF solver, depends on the following choices of parameters: the expansion domain factor $T = 3.0$, the truncated threshold value $\tau = 10^{-15}$, and the boundary penalty weight $\beta = 10^{6}$. In Section 3.5, we examine the different types and influences of the aforementioned parameters. Our objective is to verify their impact and finalize our parameter choices using the relative error metric.

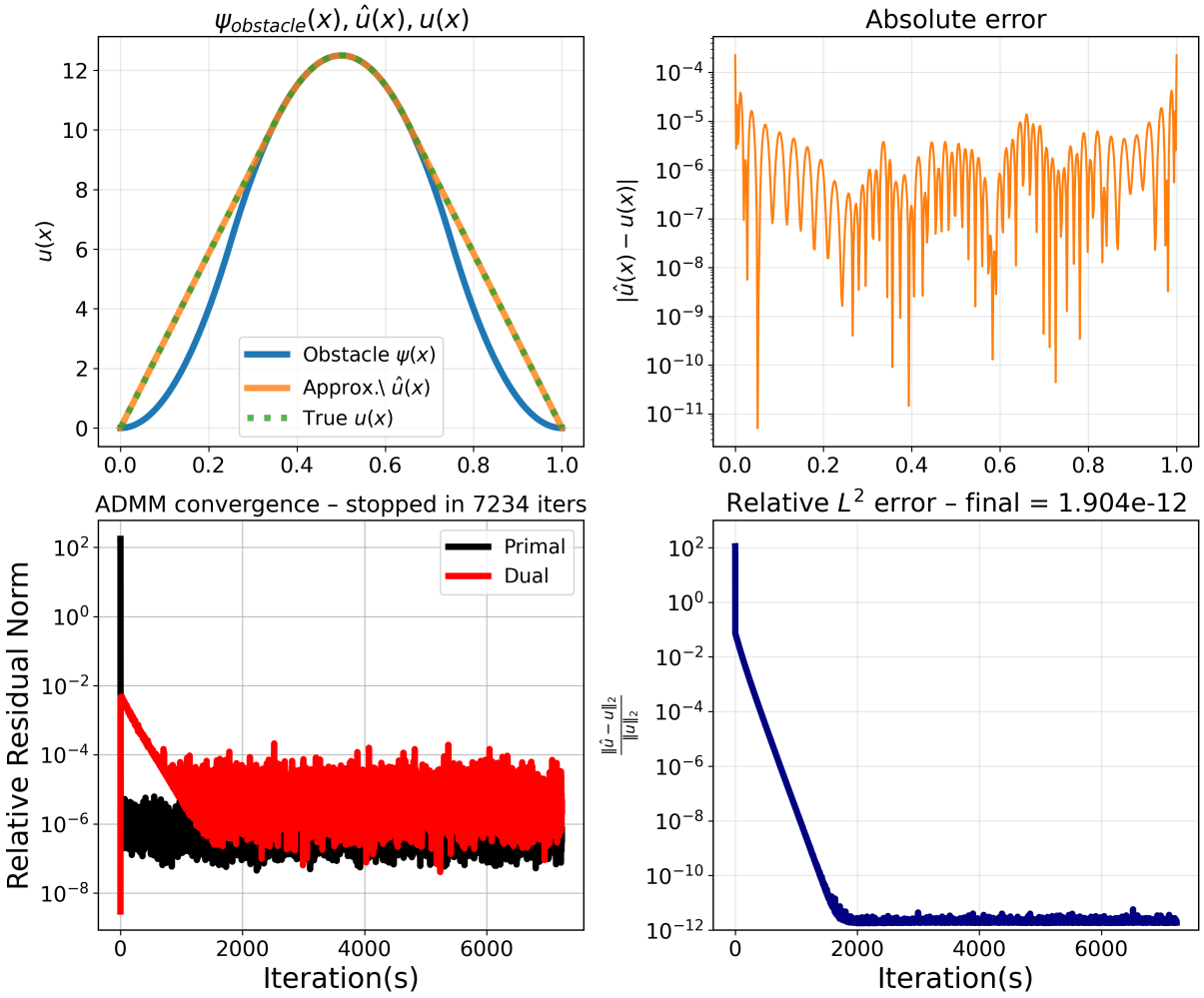


Figure 3.3: **RBF approximation of the one-dimensional obstacle problem with a smooth single-bump obstacle function $\psi(x)$ on the domain $\Omega = [0, 1]$, using $(N, m) = (2048, 4096)$ and the ADMM parameters $(\mu, \rho) = (300, 45)$.** Top left: Exact solution $u(x)$, numerical approximation $\hat{u}(x)$, and obstacle function $\psi(x)$. Top right: Absolute error $|\hat{u} - u|$ shown on a logarithmic scale. Bottom left: convergence history in the ADMM, showing rapid decay of the relative primal and dual residuals toward the stopping tolerance $\varepsilon_{\text{tol}} = 10^{-6}$; convergence is achieved in 7,234 iterations. Bottom right: Relative ℓ_2-error history, reaching 1.904×10^{-12}.

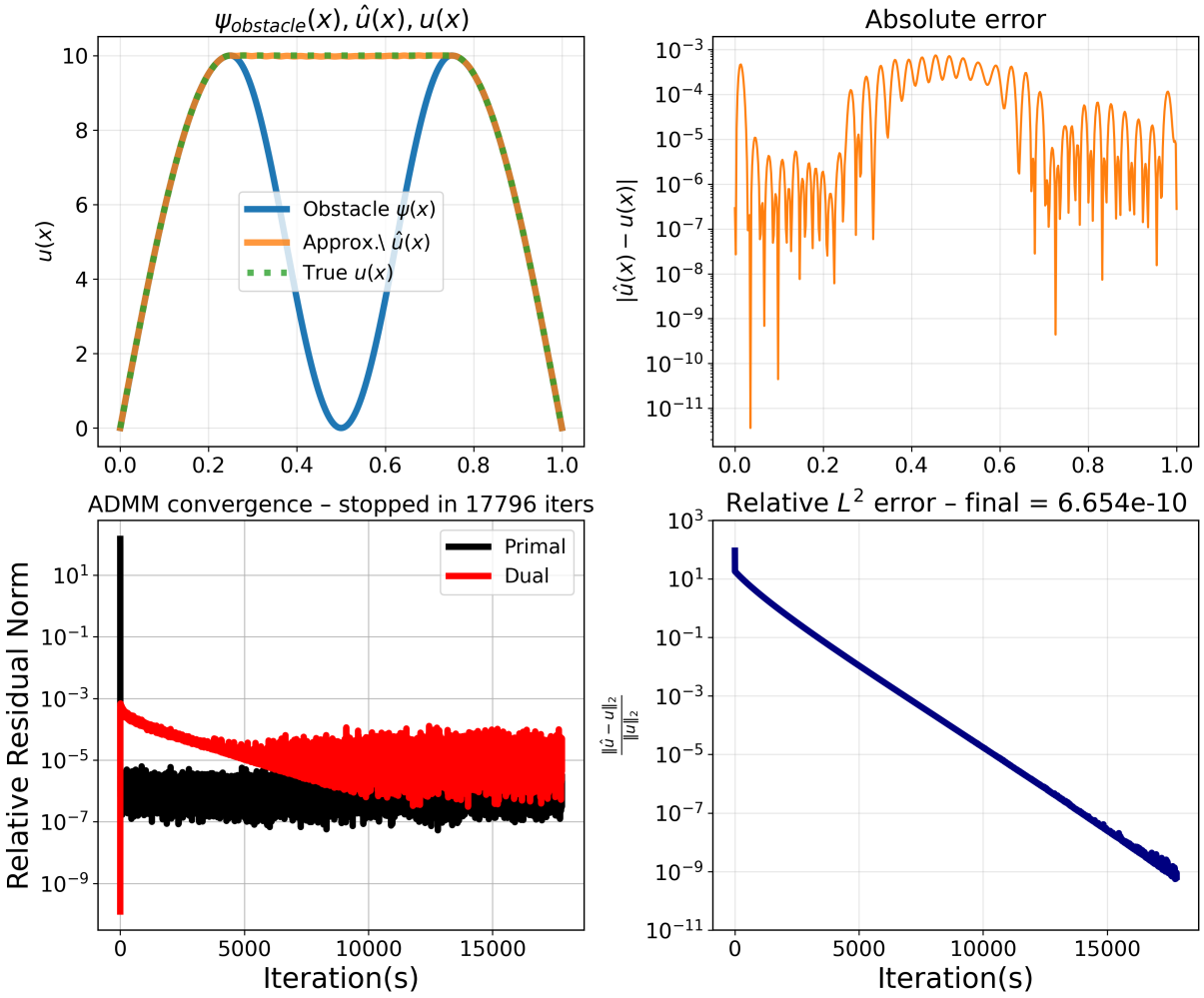
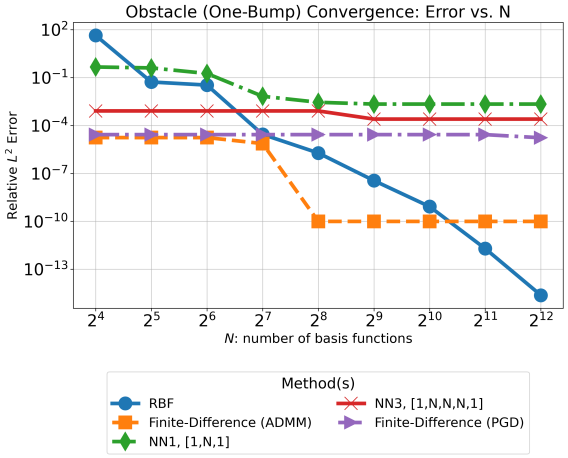
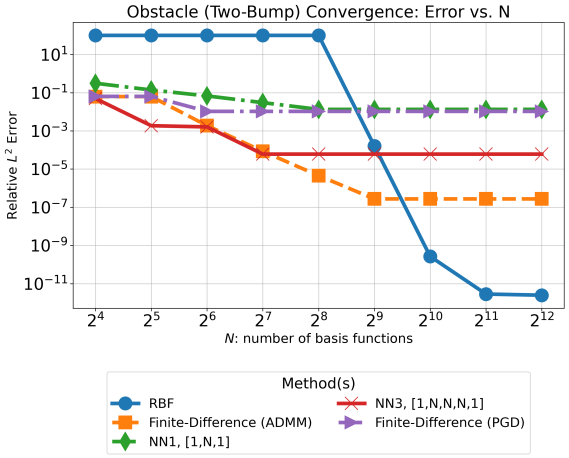


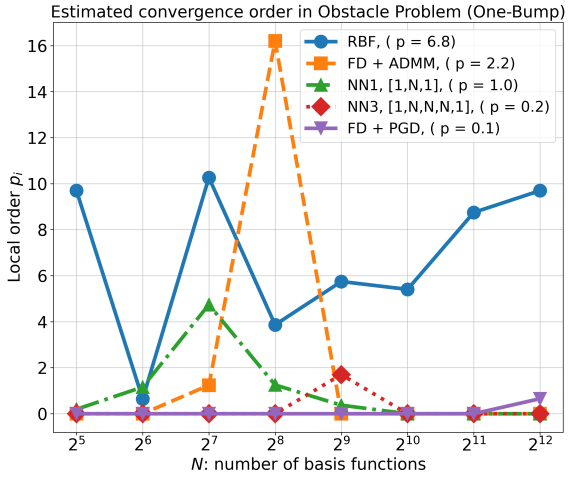
Figure 3.4: **RBF approximation of the one-dimensional obstacle problem with a two bump obstacle function** $\psi(x)$ **on** $\Omega = [0, 1]$. The computation is performed with $(N, m) = (2048, 4096)$ and hyper-parameters $(\mu, \rho) = (2.5 \times 10^4, 250)$. Top left: comparison of the exact solution $u(x)$, the RBF approximation $\hat{u}(x)$, and the obstacle profile $\psi(x)$. Top right: absolute error $|\hat{u} - u|$ plotted on the logarithmic scale. Bottom left: convergence in ADMM, illustrating the decay of the relative primal and dual residuals to the tolerance $\epsilon_{\text{tol}} = 10^{-6}$ after 17,796 iterations. Bottom right: evolution of the relative ℓ_2 error, which attains 6.654×10^{-10}.



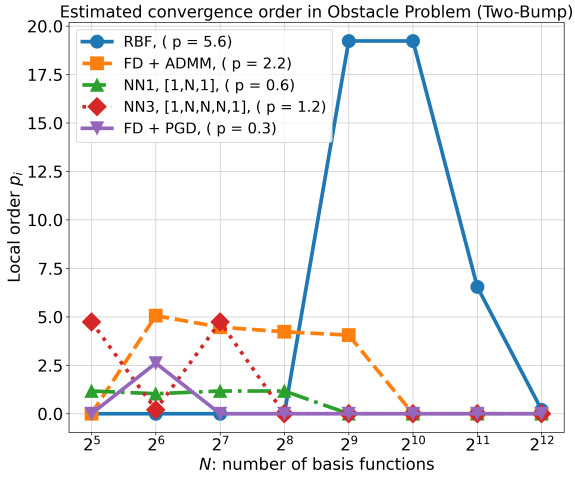
(a) Single-bump problem: relative ℓ^2 error vs. N with $T = 2.0$, $\zeta = 2$, $\tau = 10^{-15}$, and $\beta = 10^6$.



(b) Two bump problem: relative ℓ^2 error vs. N with $T = 3.0$, $\zeta = 2$, $\tau = 10^{-15}$, and $\beta = 10^8$.

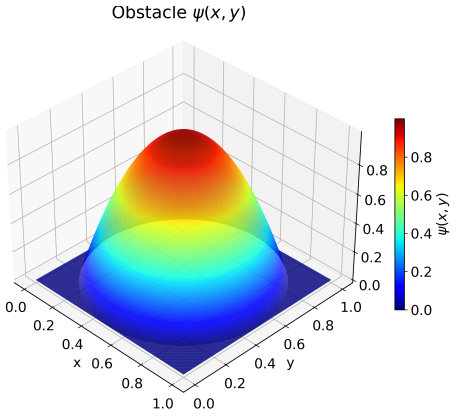


(c) Single-bump problem: local estimated convergence order p and average order with $T = 2.0$, $\zeta = 2$, and $\tau = 10^{-15}$.

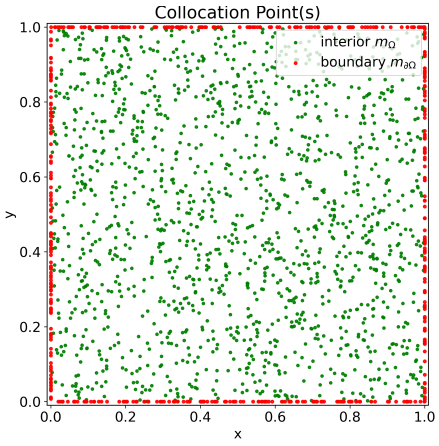


(d) Two bump problem: local estimated convergence order p and average order with $T = 3.0$, $\zeta = 2$, and $\tau = 10^{-15}$.

Figure 3.5: **Comparison for the one-dimensional obstacle problems**. The top row shows the relative ℓ^2 error versus the number of basis functions N for single-bump problem (top left) and two bump problem (top right). The bottom row shows the corresponding local estimated convergence order p and average orders $\hat{p}$ for single-bump problem (bottom left) and two bump problem (bottom right).

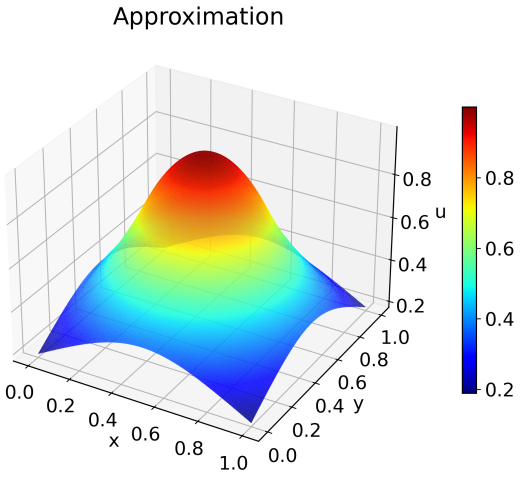


(a) Obstacle function $\psi(x, y)$ on $\overline{\Omega} = [0, 1]^2$.

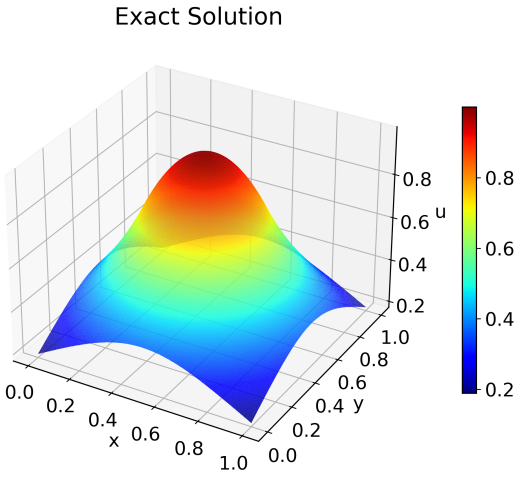


(b) Interior and boundary collocation points used by the variational solver.

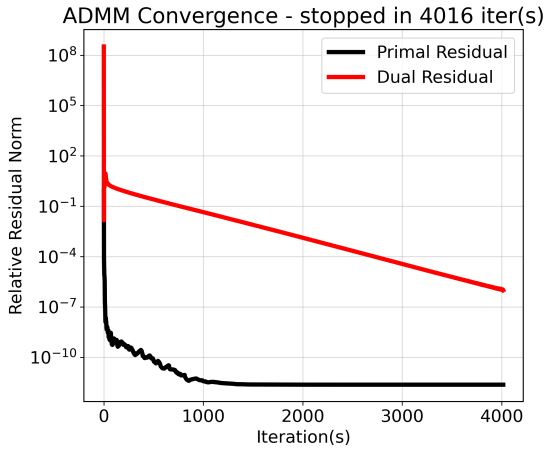
Figure 3.6: **Setup for the two-dimensional dome obstacle problem on** $\overline{\Omega} = [0, 1]^2$. The left panel shows the obstacle function $\psi(x, y)$, and the right panel shows the collocation points, where green dots denote interior points and red dots denote boundary points.



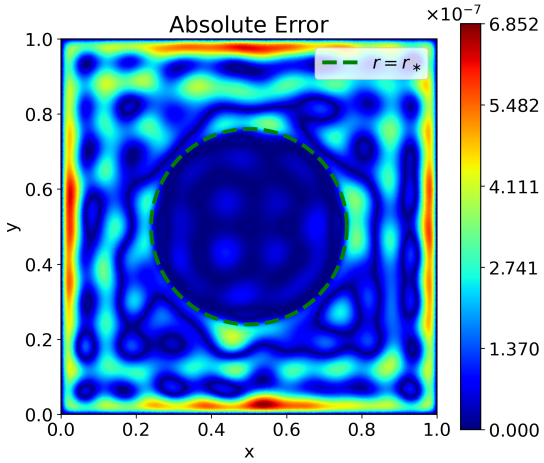
(a) Approximation $\hat{u}(x, y)$ with ADMM optimization using hyper-parameters $\mu = 10/h^2$, $\rho = 20.3$, and $\beta = 10^6$.



(b) Exact solution $u_{\text{exact}}(x, y)$.

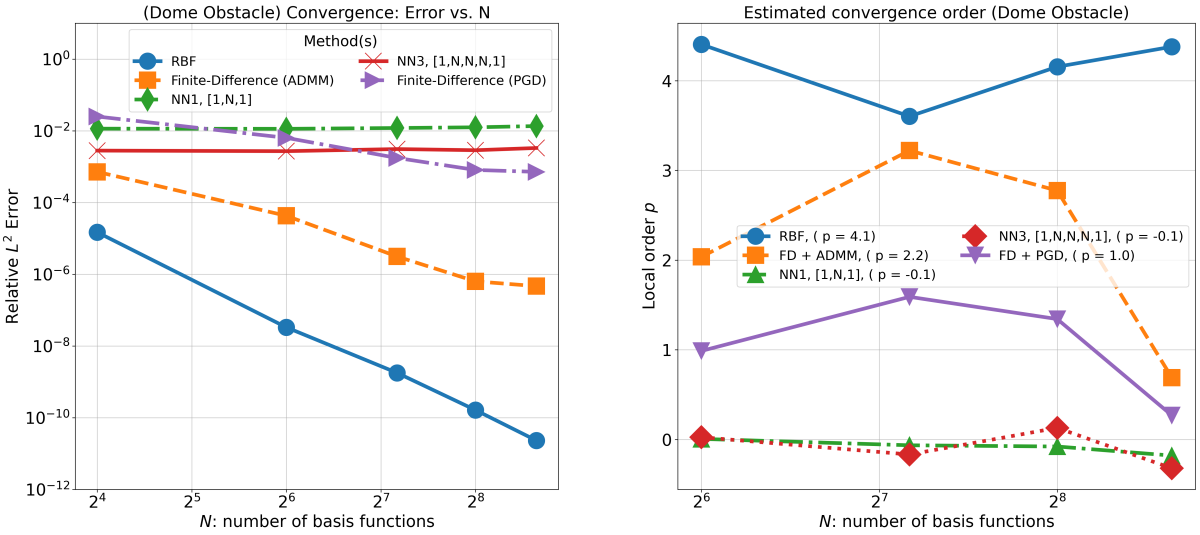


(c) Relative primal and dual residuals showing convergence to the tolerance $\epsilon_{\text{tol}} = 10^{-6}$ in 4,016 iterations.



(d) Pointwise error $|\hat{u} - u|$; the relative ℓ^2 error between the approximation and analytical solution is $E(N) = 4.681 \times 10^{-11}$.

Figure 3.7: **Meshfree RBF approximation of a radially symmetric obstacle problem on $\overline{\Omega} = [0, 1]^2$ using ADMM, with $N = 400$ and 1,600 collocation points.** The top-left panel shows the numerical approximation $\hat{u}(x, y)$, the top-right panel shows the exact solution $u_{\text{exact}}(x, y)$, the bottom-left panel shows the convergence history of the primal and dual residuals, and the bottom-right panel shows the pointwise error $|\hat{u} - u|$. The method reaches the tolerance $\epsilon_{\text{tol}} = 10^{-6}$ in $4,016$ iterations and attains a relative ℓ^2 error of $E(N) = 4.681 \times 10^{-11}$.



(a) Relative ℓ^2 error versus the number of basis functions N.

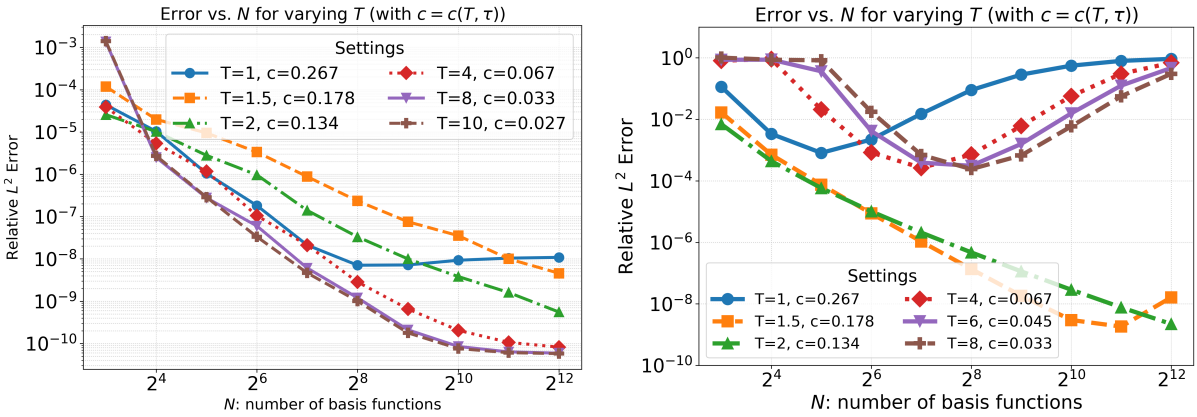
(b) Local estimated convergence order p for the corresponding methods.

Figure 3.8: **Comparison of different numerical methods for the two-dimensional dome obstacle problem**. The left panel shows the relative ℓ^2 error versus the number of basis functions N, while the right panel shows the corresponding local estimated convergence order p.

3.5 Robustness: Parameter Sensitivity

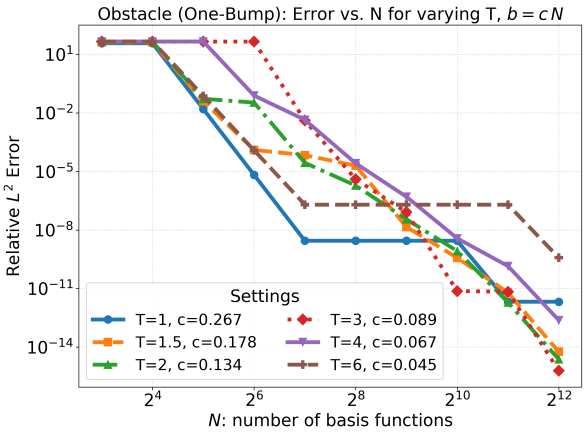
For Section 3.5, we examine the sensitivity of the proposed RBF approach to the following parameters: the expansion domain factor T, the kernel shape multiplier $c = c(T, \tau)$, the truncated threshold value τ, and the boundary penalty weight β. Respectively, we refer readers to Sections 3.5.1-3.5.4. In Section 3.5.5, we examine and validate the choices for the radial basis function ϕ in Setting 2, and we use the RBF kernels listed in Table 2.1.

The mentioned parameters directly impact the numerical stability and approximation accuracy of the method. The expansion factor T controls how far the RBF centers are placed beyond the physical domain, thereby governing the balance between reducing boundary effects and maintaining local resolution in the region. The truncated threshold value τ affects the number of singular values that are retained in the TSVD, which is the core method we use to solve the linear system. Furthermore, we examine the penalty weight β, which needs to be tuned to a sufficiently large value to impose the Dirichlet boundary condition $u = g$ on $\partial\Omega$. We also examine the relationship of the kernel shape parameters $b = c\,N$ and $b = c\,\sqrt{N}$. Although the choice of the kernel shape parameters is made in Eq. (3.10), it is necessary to investigate that, in the one-dimensional setting, we use $b = c\,N$, while in the two-dimensional setting, we consider $b = c\,\sqrt{N}$. Unless otherwise stated in Section 3.5.4, for the 1D Poisson and reaction-diffusion equations, we set the boundary-condition penalty weight to $\beta = 3 \times 10^5$ and 10^6, respectively. For the 1D obstacle problems, we choose $\beta = 10^6$ and $\beta = 10^8$, respectively. Finally, for the 2D dome obstacle problem, we take $\beta = 10^6$. Except for the experiments in Section 3.5.3, we let the truncated threshold value be $\tau = 10^{-15}$. Lastly, the kernel ϕ specifies the smoothness, support, and approximation characteristics of the basis. Consequently, a systematic comparison of the parameters and kernels is necessary to determine robust configurations for the numerical experiments conducted in this work.

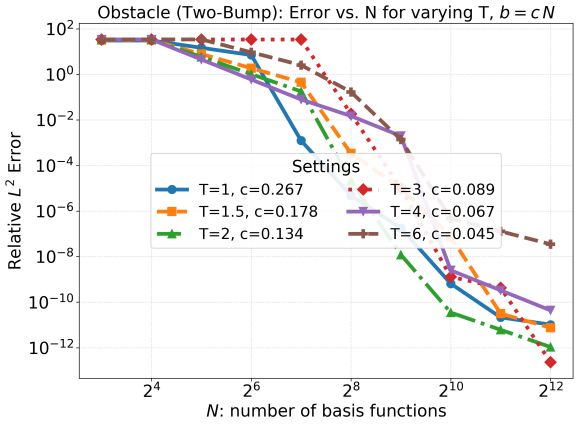


(a) 1D Poisson problem, $\beta = 3 \times 10^5$. (b) 1D reaction-diffusion problem, $\beta = 10^6$.

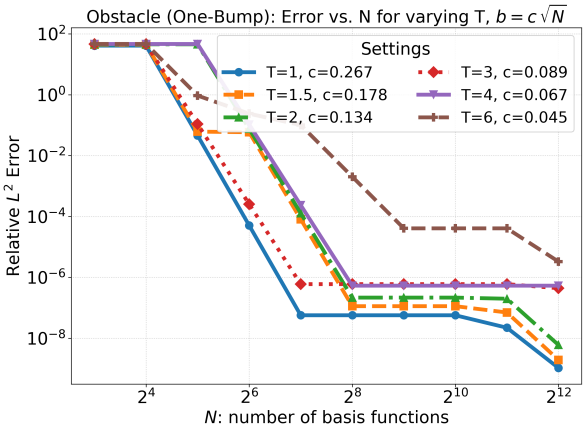
Figure 3.9: **Comparison of the expansion domain factor T in one-dimensional Poisson and reaction-diffusion equation.** Relative ℓ^2 error versus N for several choices of the expansion factor T across different test problems. In all experiments, we calculate $c = c(T, \tau)$ using the formula provided in Table 3.1. In (a), the 1D Poisson problem is shown with $\zeta = 2$, $\tau = 10^{-15}, \beta = 3 \times 10^5$. In (b), the 1D reaction-diffusion problem is shown with $\zeta = 2$, $\tau = 10^{-15}$, $\beta = 10^6$. We use the relationship of the kernel shape parameter $b = c\,N$.



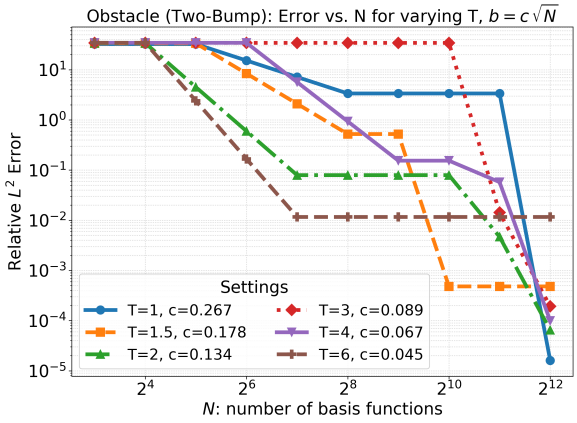
(a) Single-bump obstacle problem with linear scaling $b = cN$, using $\beta = 10^6$.



(b) Two bump obstacle problem with linear scaling $b = cN$, using $\beta = 10^8$.

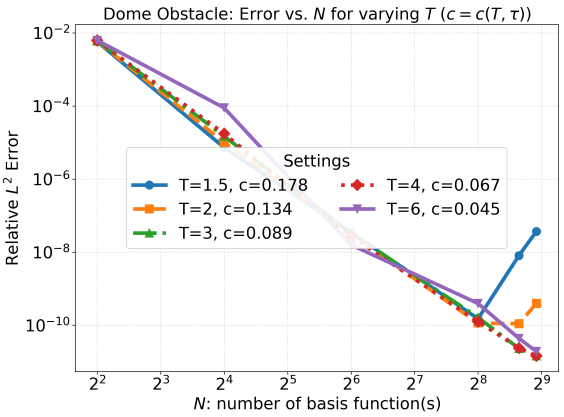


(c) Single-bump obstacle problem with the scaling $b = c\sqrt{N}$, using $\beta = 10^6$.

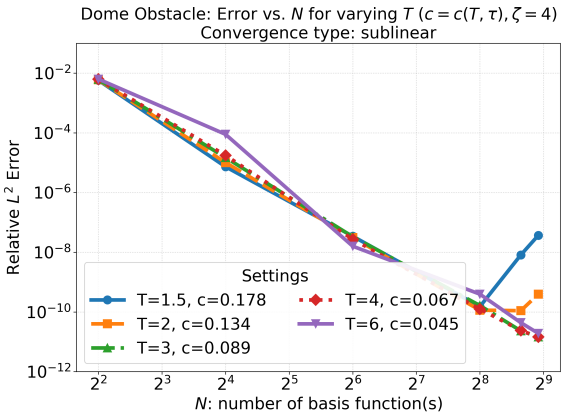


(d) Two bump obstacle problem with the scaling $b = c\sqrt{N}$, using $\beta = 10^8$.

Figure 3.10: **Comparison of the expansion domain factor T for the one-dimensional obstacle problem under the scaling b.** Relative ℓ^2 error versus N for several choices of the expansion factor T in two test problems. The top row corresponds to the scaling $b = cN$, while the bottom row corresponds to the scaling $b = c\sqrt{N}$. In all experiments, the parameter $c = c(T, \tau)$ is computed using the formula given in Table 3.1. The left column shows the single-bump obstacle problem, and the right column shows the two bump obstacle problem.



(a) Radially symmetric obstacle problem with linear scaling $b = cN$, using $\beta = 10^6, \zeta = 4$.



(b) Radially symmetric problem with scaling $b = c\sqrt{N}$, using $\beta = 10^6, \zeta = 4$.

Figure 3.11: **Comparison of the expansion domain factor T for the radially symmetric obstacle problem.** The plots show the relative ℓ^2 error versus the number of basis functions N for several choices of the expansion factor T, where the coefficient $c = c(T, \tau)$ is determined accordingly. The left panel corresponds to the linear scaling $b = cN$, while the right panel corresponds to the scaling $b = c\sqrt{N}$. In both cases, the error decreases steadily as N increases, demonstrating convergence of the RBF approximation. The results also show that the choice of T influences the accuracy, particularly for moderate values of N, while both scaling strategies achieve very small relative ℓ^2 errors for sufficiently large N. Here, the penalty parameter is set to $\beta = 10^6$, the oversampling factor is $\zeta = 4$, and the truncated threshold value is $\tau = 10^{-15}$.

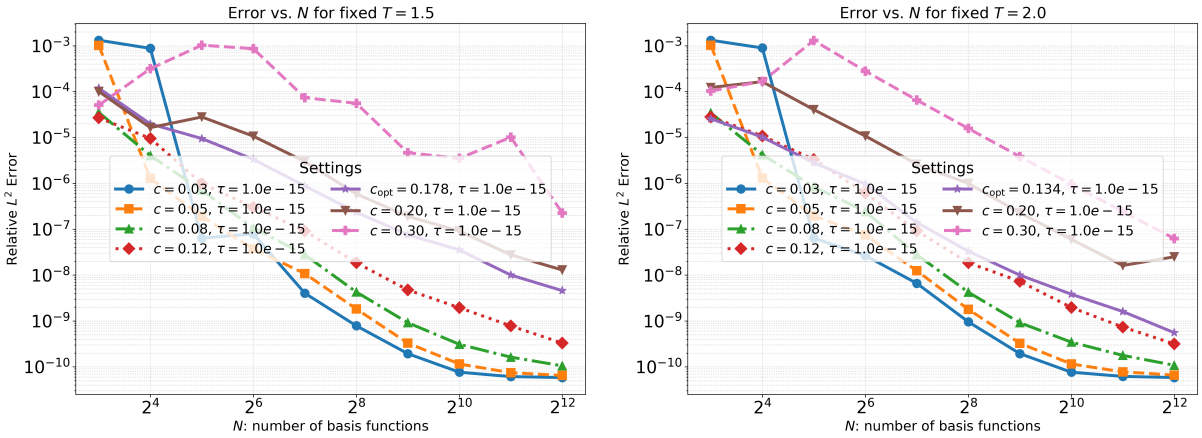
3.5.1 Sensitivity Analysis of Domain Expansion Factor T

In this section 3.5.1, we investigate the sensitivity of the parameter T, which represents the expansion domain factor in Setting 1. The role of the expansion domain factor T is crucial because it expands the physical domain $\overline{\Omega}$ from PDEs to the expanded domain $\overline{\Omega}_T$. It is also one of the parameters that determines the kernel shape multiplier $c = c(T, \tau)$. Furthermore, we investigate the effect of the relationship between the kernel shape parameters, including linear scaling $b = c\,N$ and the scaling $b = c\,\sqrt{N}$. First, we start with the examination of T in Algorithm 1 to solve linear elliptic PDEs. Figures 3.9(a)(b) indicate that the best choice of T is problem-dependent. For the 1D Poisson equation in Figure 3.9(a), we select $T = 8$ because it yields a more consistent error decay than smaller expansion factors and remains among the top-performing values across the examined range of N. In contrast, for the 1D reaction–diffusion problem, we take $T = 2$, since for other values of T the relative error plateaus and then increases after a certain number of RBF functions, as illustrated in Figure 3.9(b).

As shown in Figure 3.10(a)(b), we set $T = 2$ for the 1D one bum obstacle problem and $T = 3$ for the two bump case. These values are sufficient to reveal the primary advantages of the domain expansion factor T, while still preserving an accurate representation of the localized obstacle structures. Increasing T further does not yield systematically improved performance in either example. Moreover, from Figure 3.10(c)(d), it is evident that in this setting the kernel shape parameter should not follow the scaling $b = c\sqrt{N}$. Under this choice, the relative error levels off, becomes less accurate, and converges more slowly than when using the linear scaling $b = cN$.

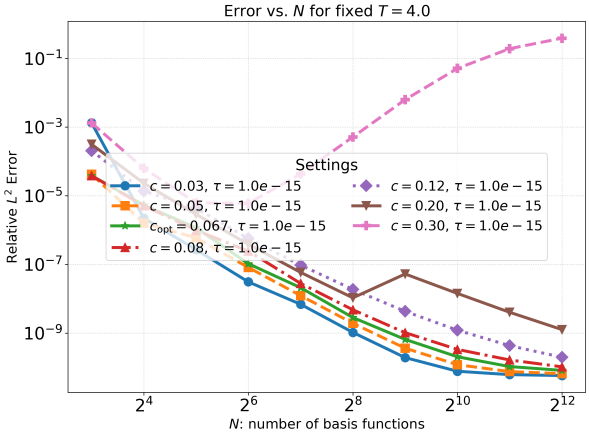
As shown in Figure 3.11(a)(b), for the 2D dome obstacle problem we choose $T = 3$, which yields one of the most stable error profiles among the tested values and strikes a good balance between mitigating boundary effects and maintaining adequate resolution near the physical domain. Recall that the shape parameter for the 2D dome obstacle problem is set to $b = c\,\sqrt{N}$. Consistent with the one-dimensional case, as illustrated in Figure 3.11(a), the relative error clearly converges more slowly when the linearly scaled parameter $b = c\,N$ is used. Thus, the scaling $b = c\,\sqrt{N}$ is indeed preferred to be selected in the two-dimensional setting.

Therefore, the chosen expansion factors are selected empirically according to the sensitivity results for each problem. In the next section, we examine the effect of the kernel shape multiplier $c = c(T, \tau)$ based on Eq. (3.10).

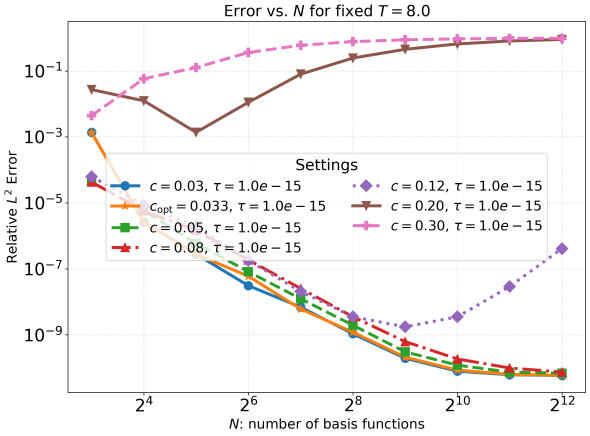


(a) $T = 1.5$

(b) $T = 2.0$

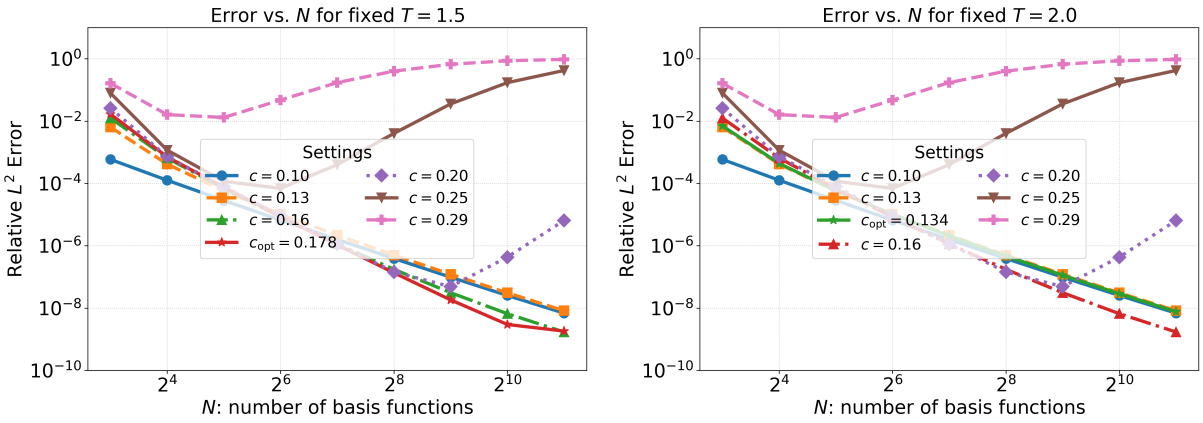


(c) $T = 4.0$



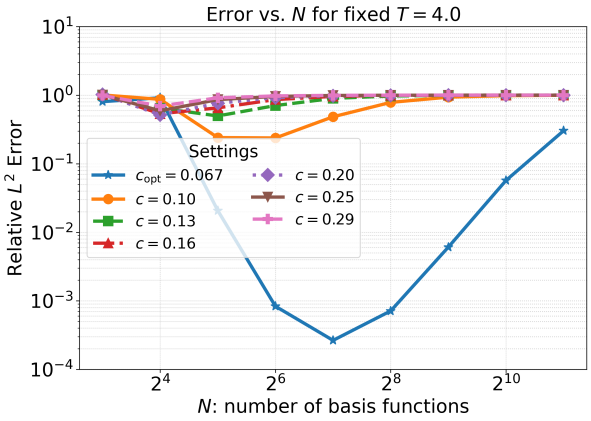
(d) $T = 8.0$

Figure 3.12: **Effect of the kernel shape multiplier $c(T, \tau)$ in the Poisson equation**. Relative ℓ^2 error versus the number of basis functions N for fixed expansion factor T, with oversampling ratio $\zeta = 2$ and truncation threshold $\tau = 10^{-15}$. Each panel compares several values of the proportionality constant c in the linear shape-parameter scaling, including the selected value $c_{\mathrm{opt}}(T, \tau)$. For moderate values of T, the error decreases steadily with N for a range of stable c-values, while overly large c leads to a loss of stability at large N.

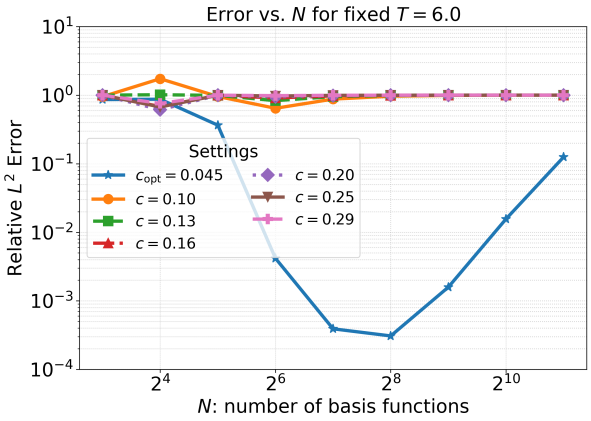


(a) $T = 1.5$

(b) $T = 2.0$

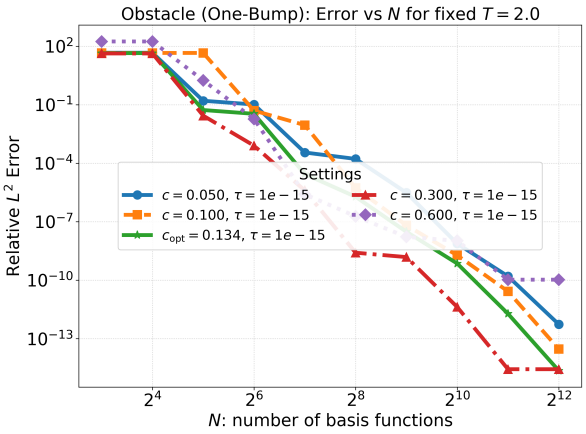


(c) $T = 4.0$

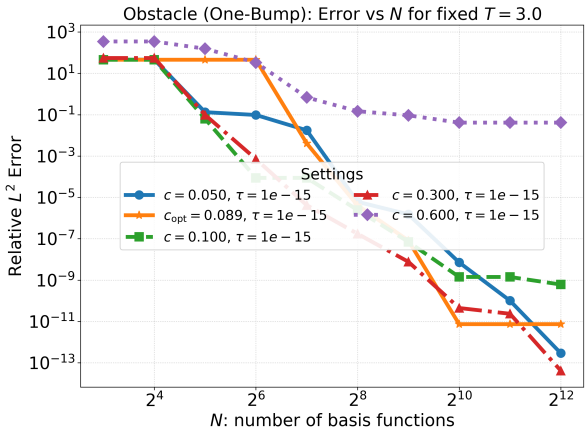


(d) $T = 6.0$

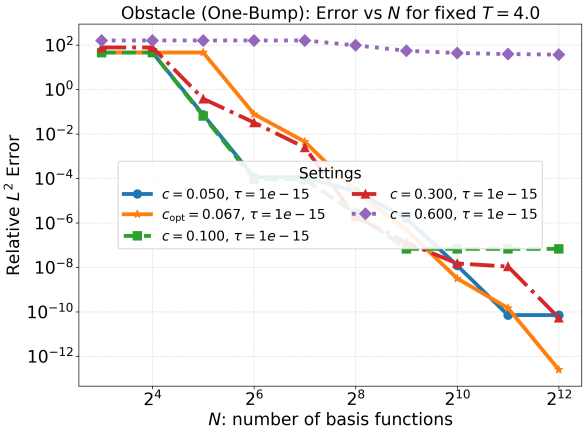
Figure 3.13: **Effect of the kernel shape multiplier $c(T, \tau)$ in the reaction-diffusion PDE**. Relative ℓ^2 error versus N for fixed expansion factors T and varying parameters c. Each panel shows the sensitivity of the RBF approximation to the choice of c for a fixed value of T. The value $c_{\text{opt}} = c(T, \tau)$, selected from the formula in Eq. (3.10), consistently gives the best or most stable performance. For smaller values $T = 1.5$ and $T = 2.0$, the error decreases steadily as N increases, whereas for larger values $T = 4.0$ and $T = 6.0$, non-optimal choices of c lead to noticeably larger errors. These results support the use of $c = c(T, \tau)$ for balancing accuracy and stability. Here, $\tau = 10^{-15}$ and $\zeta = 2$.



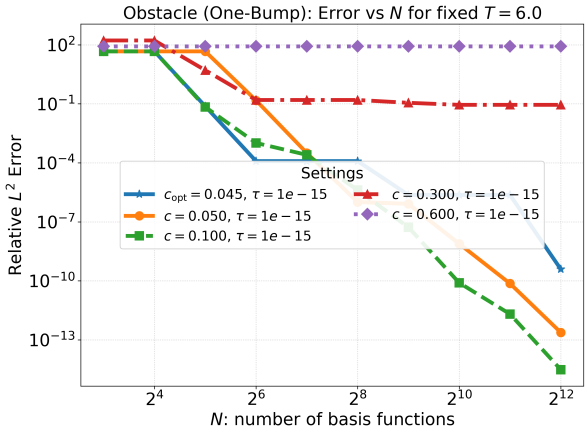
(a) Single-bump problem, $T = 2.0$



(b) Single-bump problem, $T = 3.0$

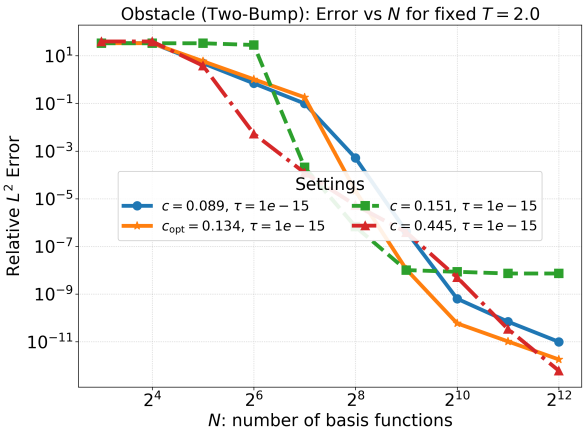


(c) Single-bump problem, $T = 4.0$

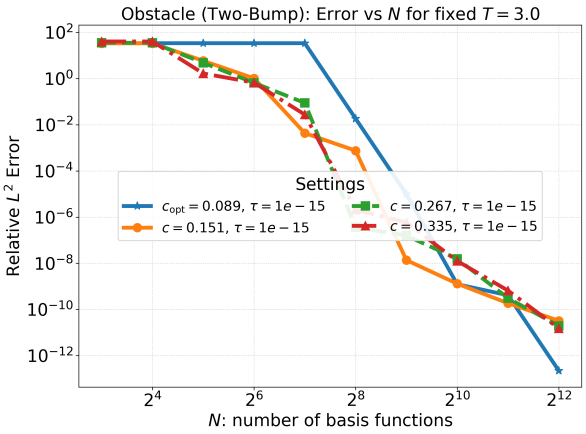


(d) Single-bump problem, $T = 6.0$

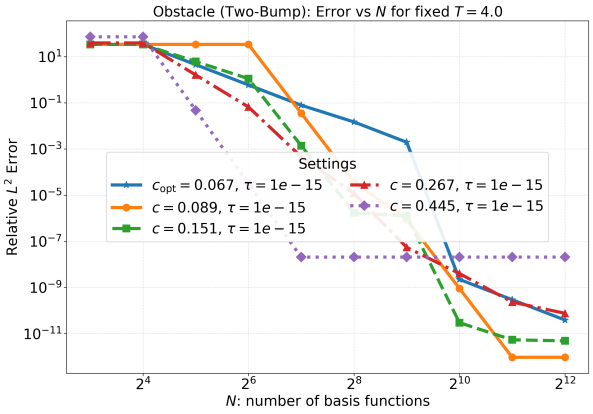
Figure 3.14: **Panels of the kernel shape multiplier $c(T, \tau)$ in obstacle-type BVP (single-bump problem)**. Relative ℓ^2 error versus N for fixed expansion factors T and varying parameters c. For each fixed T, the figure compares several choices of c in the kernel shape parameter. The theoretically selected value $c_{\text{opt}} = c(T, \tau)$ gives the best or most stable accuracy across all cases. The approximation is less sensitive to c for smaller values of T, while for larger T the error deteriorates significantly for non-optimal choices. Here, $\tau = 10^{-15}$, $\zeta = 2$, and $\beta = 10^6$.



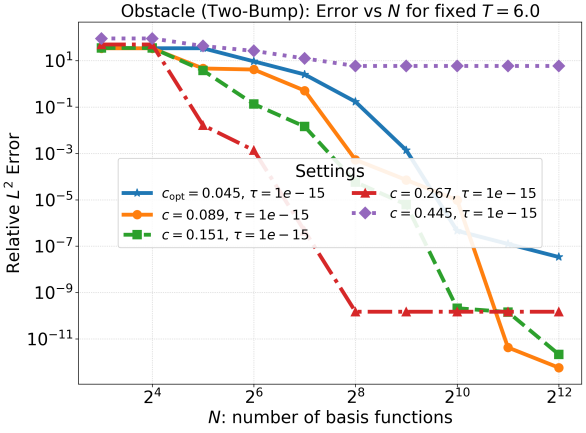
(a) Two bump problem, $T = 2.0$



(b) Two bump problem, $T = 3.0$

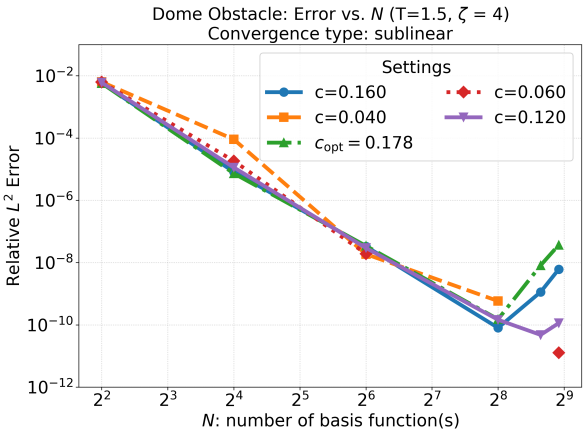


(c) Two bump problem, $T = 4.0$

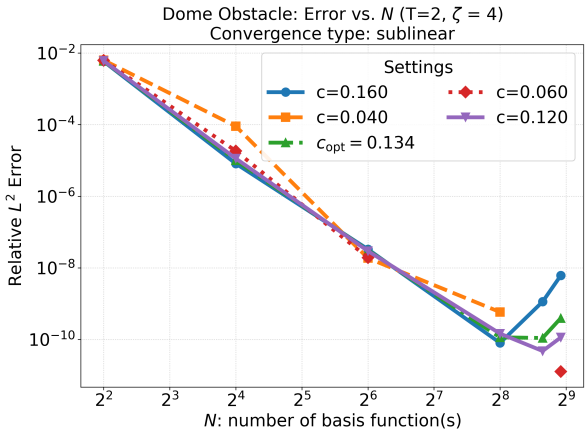


(d) Two bump problem, $T = 6.0$

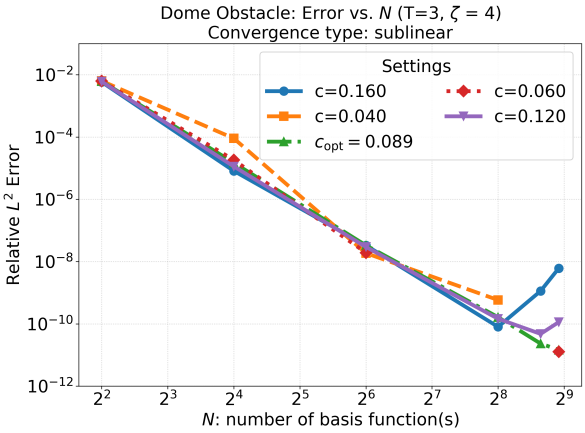
Figure 3.15: **Effect of the kernel shape multiplier $c(T, \tau)$ to the two bump obstacle problem.** Relative ℓ^2 error versus N for fixed expansion factors T and varying values of c. For each fixed T, the theoretically selected value $c_{\text{opt}} = c(T, \tau)$ provides the best or most stable accuracy, while non-optimal choices become increasingly less reliable for larger T. Here, $\tau = 10^{-15}$, $\zeta = 2$, and $\beta = 10^8$.



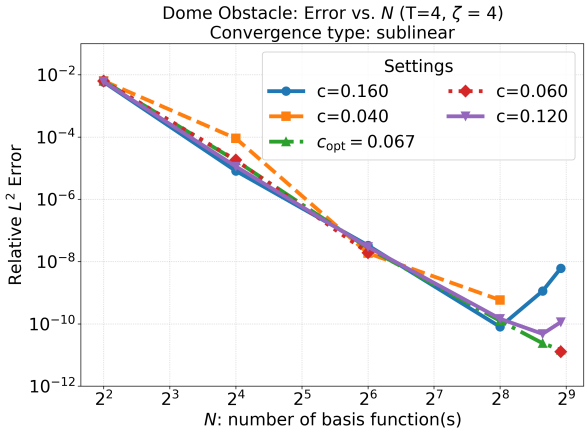
(a) Dome obstacle problem, $T = 2.0$



(b) Dome obstacle problem, $T = 3.0$



(c) Dome obstacle problem, $T = 4.0$



(d) Dome obstacle problem, $T = 6.0$

Figure 3.16: **Effect of the kernel shape multiplier $c(T, \tau)$ for the radially symmetric obstacle problem.** Relative ℓ^2 error versus N is plotted for fixed values of the expansion factor T and different choices of the kernel shape multiplier parameter c. The theoretically motivated value $c_{\text{opt}} = c(T, \tau)$ consistently gives the best performance, confirming its effectiveness in balancing approximation accuracy and numerical stability. Here, $\tau = 10^{-15}$, $\zeta = 4$, $\beta = 10^6$, and $(\mu, \rho) = (10/h^2, 20.3)$, where $h = \frac{1}{N}$.

3.5.2 Experiments on the Kernel Shape Multiplier Parameter $c(T, \tau)$

In this subsection, we investigate the effect of the kernel shape multiplier parameter $c(T, \tau)$, which is calculated based on Eq. 3.10 by [5]. We acknowledge that the formula for $c = c(T, \tau)$ is used for functional approximation in [5]; however, in the experiment, it works effectively close to the optimal value for the kernel shape multiplier. Given the analysis in Section 3.5.1, we consider fixing the following parameters in this experiment. For linear elliptic PDEs, we set $T = 8.0$ for the Poisson problem and $T = 2.0$ for the reaction-diffusion equation, followed by the boundary penalty weight $\beta = 3 \times 10^5$, 10^6, respectively. In the one-dimensional obstacle problem, we choose $T = 2.0, \beta = 10^6$ for the single-bump obstacle and $T = 3.0, \beta = 10^8$ for the two bump problem. In the radially symmetric obstacle problem, we keep $T = 3.0$, $\beta = 10^6$. In all cases, the truncation threshold $\tau = 10^{-15}$ is selected.

As shown in Figure 3.12, the c_{opt} does not always work effectively across all choices in the Poisson problem, particularly in the case of $T = 1.5$, 2.0.. However, when the expansion domain factor T is selected to be sufficiently large, the kernel shape multiplier $c_{\text{opt}} = c(T, \tau)$ influences the convergence of the Poisson problem, as illustrated in Figure 3.12(c)(d). It is evident that for $T = 4$, 8, c_{opt} causes the relative error to increase and decreases steadily with the N number of basis functions.

Similarly, Figure 3.13(a)(b) shows that when T is chosen sufficiently large enough, as $T = 1.5$, 2.0, c_{opt} causes the relative error to decay as N increases. On the other hand, for larger values of $T = 4.0$, 6.0 non-optimal choices of c lead to a larger relative error.

For the one-dimensional setting in Figures 3.14-3.15, a similar observation is noted in the Poisson and reaction-diffusion problems. It is necessary to choose T sufficiently large to ensure that the relative error decays steadily as N increases due to the kernel shape multiplier $c_{\text{opt}} = c(T, \tau)$. As illustrated in Figure 3.14(a)(b) and Figure 3.15(a)(b), we need to select $T = 2.0$ for the single-bump obstacle problem, while in the two bump problem, we choose $T = 3.0$. Because when $T = 2.0$ there are not enough selective center points in the expansion domain to ensure the convergence of the problem. In the two-dimensional settings, Figure 3.16 shows a similar observation, as it is not recommended in panels (a) and (b) to choose $T = 1.5$, 2.0. For $T = 3.0, 4.0$ and $c_{\text{opt}} = 0.089, 0.067$, respectively, the formula $c(T, \tau)$ consistently causes the relative error to decay steadily as N increases, confirming its effectiveness in balancing approximation accuracy and stability.

Thus, the experiments in Section 3.5.2 confirm our selective choices for the kernel shape multiplier $c_{\text{opt}} = c(T, \tau)$. The formula in Eq. (3.10) serves as a guideline for deciding the

choices of the kernel shape multiplier c in the relationship $b = c\,N$ for one-dimensional problems or $b = c\sqrt{N}$ for the two-dimensional setting. In most cases, we observe that the relative error decays and converges as N increases for the appropriate choices of c_{opt} from the formula $c(T, \tau)$. It is also noted that the expansion domain factor T needs to be chosen sufficiently large, as illustrated in Figures 3.12-3.16, to ensure that c_{opt} becomes effective in the RBF model. Next, we examine the influence of the truncation threshold value τ in the TSVD.

3.5.3 Experiments on Truncation Threshold Value τ

In this subsection, we investigate the influence of the truncation threshold value τ in Algorithms 1-2. We maintain the settings for each corresponding PDE in the following. For the one-dimensional Poisson and Reaction-diffusion equations, we let $\beta = 3 \times 10^5$, 10^6, respectively. In the single- and two bump obstacle problems, we choose $\beta = 10^6$, 10^8, respectively. Lastly, in the dome obstacle problem, we let $\beta = 10^6$. The oversampling ratio ζ is selected as 2 for the one-dimensional setting and $\zeta = 4$ for the two-dimensional setting. As shown in Section 3.5.2, we assume the kernel shape multiplier to be $c = c(T, \tau)$ in [5]. According to Figures 3.17-3.21, it is noticeable that smaller truncation thresholds cause a more stable relative error decay and improve the approximation accuracy in the TSVD. For instance, all panels in Figure 3.17 show that although the sensitivity of the truncation threshold value τ depends on the choice of T, choices for $\tau = 10^{-12}, 10^{-15}$ generally influence robust error decay, particularly for larger N. Similarly, in the two-dimensional setting, as shown in Figure 3.21, all panels indicate that smaller values of τ tend to yield lower error levels, especially for $\tau = 10^{-12}, 10^{-15}$. Therefore, selecting the truncation threshold $\tau = 10^{-15}$ for the numerical experiments in Section 3.4 is regarded as well justified. We then examine the parameter weight β, which is used to penalize the Dirichlet boundary condition.

3.5.4 Experiments on Boundary Penalty Weight β

We now examine the effect of the boundary penalty weight β on the Dirichlet condition. We maintain the hyperparameter settings as in the previous experiments. In particular, we select $T = 8$ in the Poisson equation and $T = 2$ for the reaction-diffusion problem, respectively. In the one-dimensional obstacle problems, we set $T = 2$, 3 for the single and two bump problems, respectively. Lastly, we keep $T = 3$ for the two-dimensional dome obstacle problem. In all cases, we choose $\tau = 10^{-15}$ as the truncation threshold value in the TSVD.

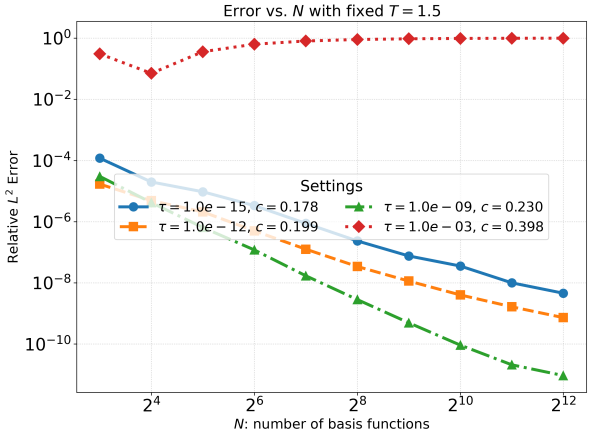
Figures 3.22 and 3.23 indicate that the boundary penalty parameter β must be chosen sufficiently large to enforce the Dirichlet condition. For the one-dimensional problems, small values of β result in large relative ℓ^2 errors, whereas increasing β quickly enhances the accuracy until a stable, low-error regime is attained, as shown in Figure 3.22. Thus, increasing β enforces the Dirichlet constraint more strongly, so that if β is chosen too small, the solution fails to match the prescribed boundary data and the approximation error becomes large. The appropriate range depends on the specific problem: $\beta = 3 \times 10^5$ already suffices for the Poisson problem; $\beta = 10^6$ yields robust and accurate solutions for the reaction-diffusion and single-bump obstacle problems; and the two bump obstacle problem necessitates a stronger choice, $\beta = 10^8$. For the two-dimensional dome obstacle test case, the relative error becomes insensitive to β once the penalty parameter is sufficiently large, suggesting that the approach remains stable across a wide spectrum of boundary weights. Taken together, these findings demonstrate that the chosen β values achieve a suitable compromise between enforcing the boundary conditions and maintaining numerical stability.

Overall, these experiments demonstrate that the boundary penalty weight needs to be selected sufficiently large to reliably impose the Dirichlet condition, yet not so large that it introduces excessive numerical stiffness. Therefore, in the Poisson and reaction-diffusion problems, we select $\beta = 3 \times 10^5$, 10^6, respectively. In the single- and two bump obstacle problems, we set $\beta = 10^6$, 10^8 for each corresponding PDEs. Finally, we choose $\beta = 10^6$ in the dome obstacle problem. In the last set of experiments, we compare the performance of the radial kernels presented in Table 2.1 and justify Setting 2.

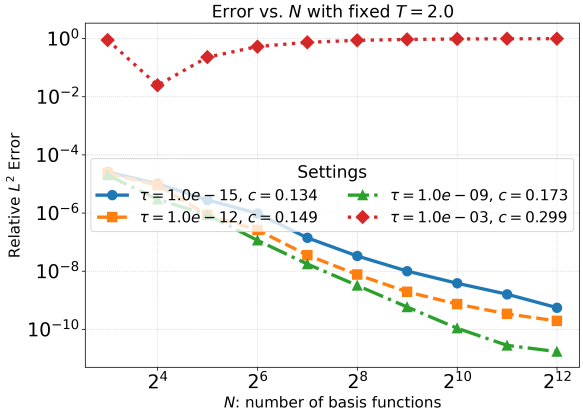
3.5.5 Comparison between the Kernel Functions ϕ

The influence of the kernel function ϕ on the approximation accuracy is examined in Figures 3.24 and 3.25, where we compare several globally supported and compactly supported radial basis functions, including the Gaussian, MQ, IMQ, IQ, and Wendland C^2 and C^4 kernels. For a detailed understanding of smoothness, we refer readers to Table 2.1. For the one-dimensional Poisson and reaction-diffusion problems, the Gaussian-type globally supported kernels consistently provide the most favorable convergence behavior as the number of basis functions N increases. In particular, the Gaussian, MQ, IMQ, and IQ kernels all produce rapidly decaying relative errors, whereas the compactly supported Wendland kernels exhibit noticeably weaker performance, especially for larger values of N, where the error either stagnates or increases. The same qualitative behavior is also observed in the obstacle problems, including the two-dimensional dome obstacle problem, where the Gaussian kernel remains the most stable and accurate choice, as illustrated in Figure 3.25. This

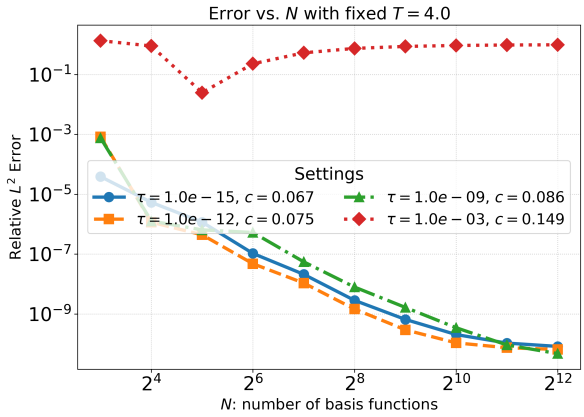
indicates that globally supported kernels are more effective in capturing the smooth structure of the exact solution within the present variational RBF framework. In Section 3.6, we discuss the drawbacks and challenges inherent to the variational RBF approach.



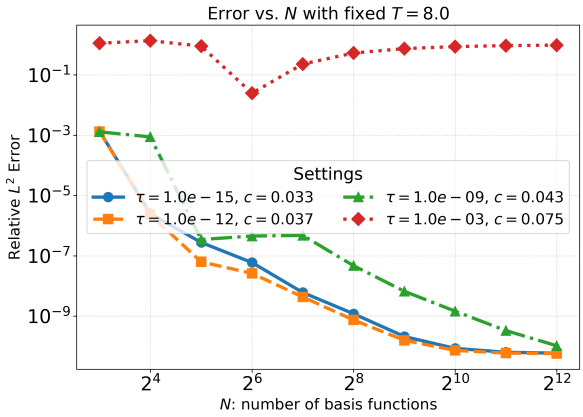
(a) Poisson problem (1D), $T = 1.5$



(b) Poisson problem (1D), $T = 2.0$

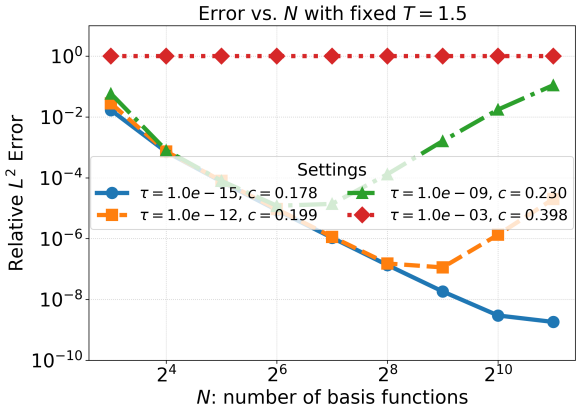


(c) Poisson problem (1D), $T = 4.0$

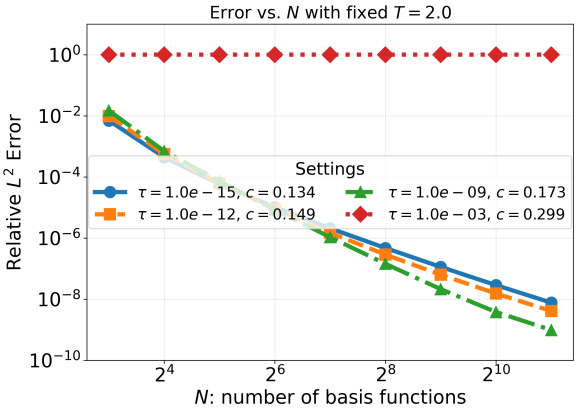


(d) Poisson problem (1D), $T = 8.0$

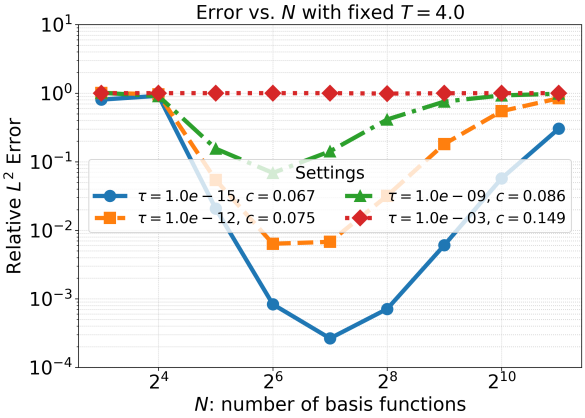
Figure 3.17: **Effect of the truncation threshold value τ on the Poisson BVP**. Relative ℓ^2 error versus the number of basis functions N for the 1D Poisson problem with fixed expansion factor T and oversampling ratio $\zeta = 2$. In each panel, the truncation threshold $\tau \in \{10^{-3}, 10^{-9}, 10^{-12}, 10^{-15}\}$ is varied, with $c = c(T, \tau)$. Smaller truncation thresholds yield more stable error decay and lower error floors, while $\tau = 10^{-12}$ and $\tau = 10^{-15}$ give the most accurate results overall.



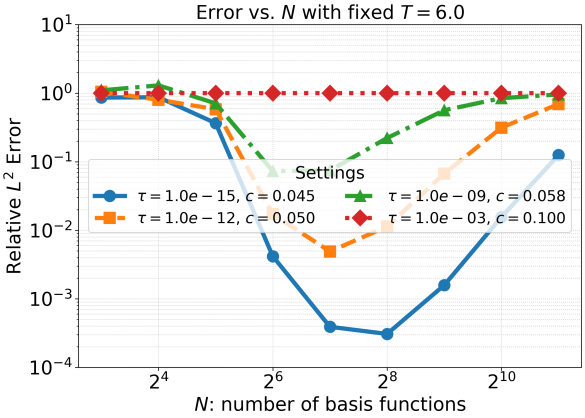
(a) Reaction-diffusion problem (1D), $T = 1.5$



(b) Reaction-diffusion problem (1D), $T = 2.0$

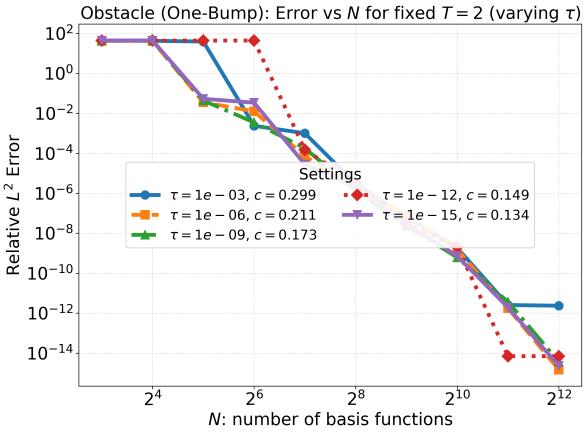


(c) Reaction-diffusion problem (1D), $T = 4.0$

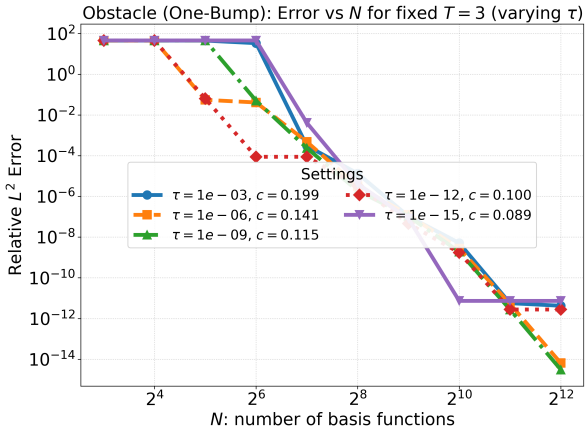


(d) Reaction-diffusion problem (1D), $T = 6.0$

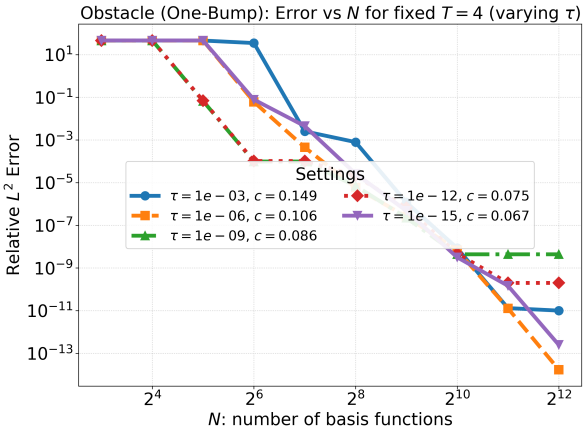
Figure 3.18: **Effect of the truncation threshold value τ on the reaction-diffusion problem in 1D.** Relative ℓ^2 error versus the number of basis functions N for fixed expansion factors T, with oversampling ratio $\zeta = 2$. In each panel, $\tau \in \{10^{-3}, 10^{-9}, 10^{-12}, 10^{-15}\}$ is varied, and the parameter c is selected according to $c = c(T, \tau)$. The results indicate that smaller truncation thresholds lead to more stable error decay and improved accuracy, while $\tau = 10^{-12}$ and $\tau = 10^{-15}$ provide the best overall performance.



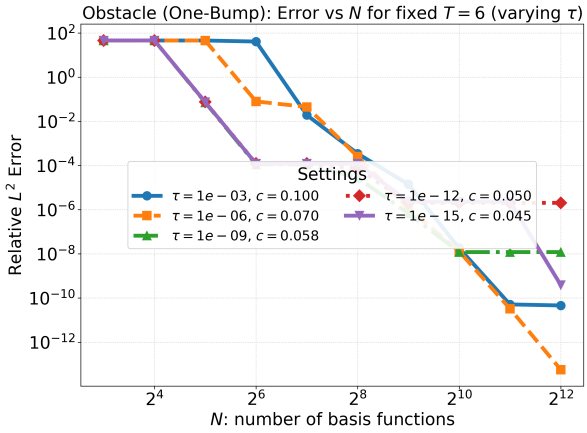
(a) Single-bump obstacle problem, $T = 2.0$



(b) Single-bump obstacle problem, $T = 3.0$

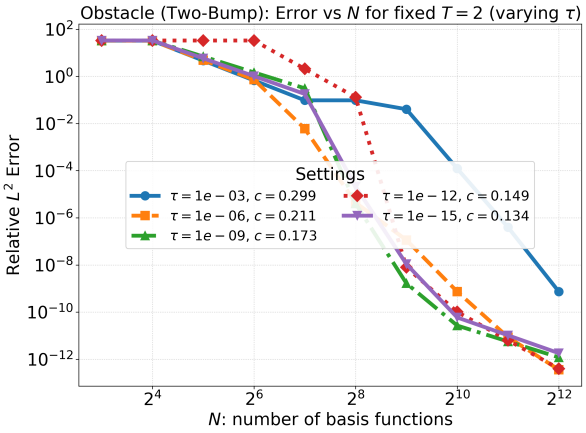


(c) Single-bump obstacle problem, $T = 4.0$

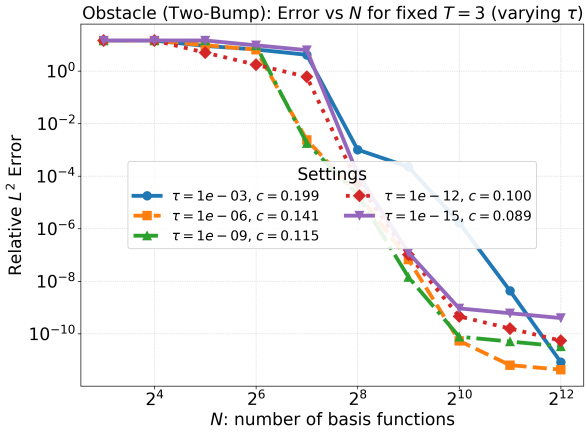


(d) Single-bump obstacle problem, $T = 6.0$

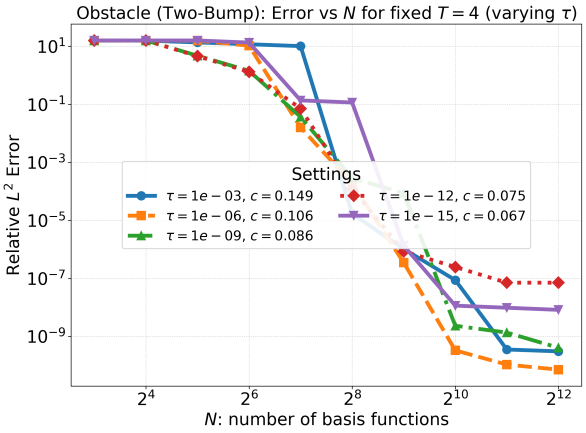
Figure 3.19: **Effect of the truncation threshold value τ on the single-bump obstacle problem in 1D.** Relative ℓ^2 error versus the number of basis functions N for fixed expansion factors T, with oversampling ratio $\zeta = 2$. In each panel, $\tau \in \{10^{-3}, 10^{-6}, 10^{-9}, 10^{-12}, 10^{-15}\}$ is varied, and the parameter c is chosen according to $c = c(T, \tau)$. The results show that smaller truncation thresholds generally lead to improved accuracy and more stable error decay, although the sensitivity to τ depends on the choice of T.



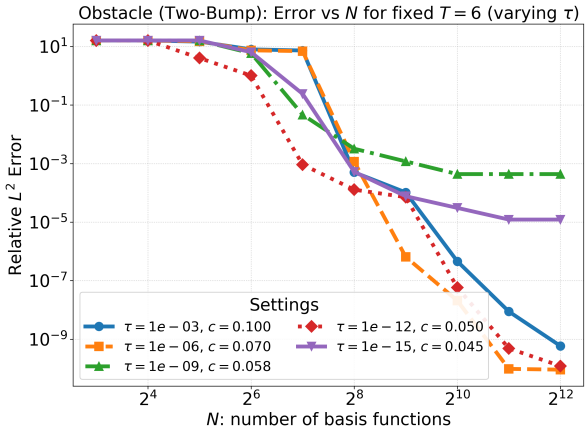
(a) two bump obstacle problem, $T = 2.0$



(b) Two bump obstacle problem, $T = 3.0$

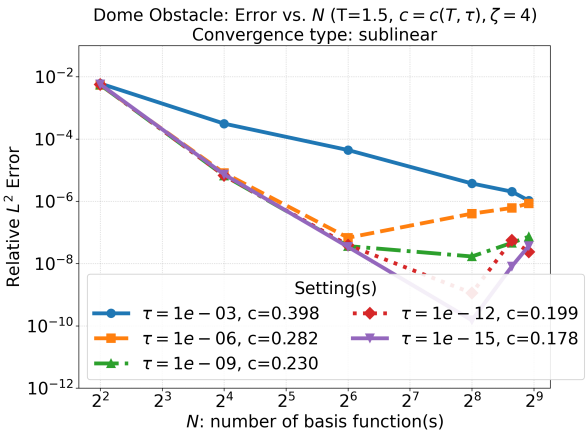


(c) Two bump obstacle problem, $T = 4.0$

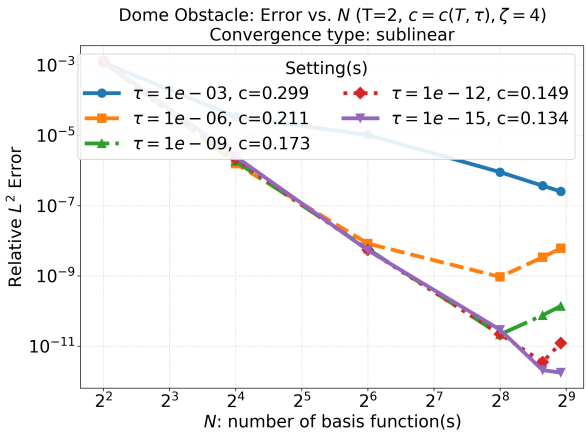


(d) Two bump obstacle problem, $T = 6.0$

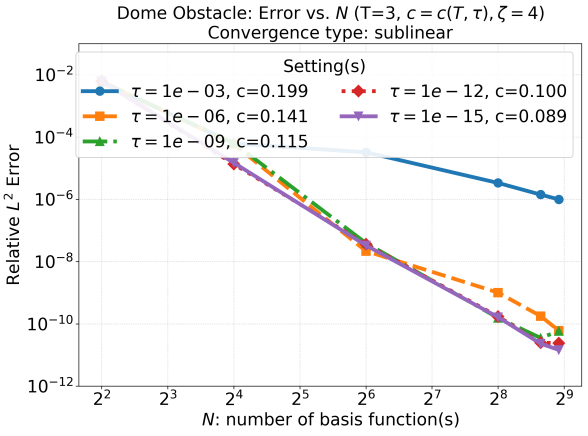
Figure 3.20: **Sensitivity to the truncation threshold τ for the 1D two bump obstacle problem.** Relative ℓ^2 error as a function of the number of basis functions N for several fixed values of the expansion factor T, using oversampling ratio $\zeta = 2$. For each threshold τ, the proportionality constant is chosen as $c = c(T, \tau)$. The figure indicates that smaller truncation thresholds generally improve stability and accuracy, although the optimal behavior remains dependent on T.



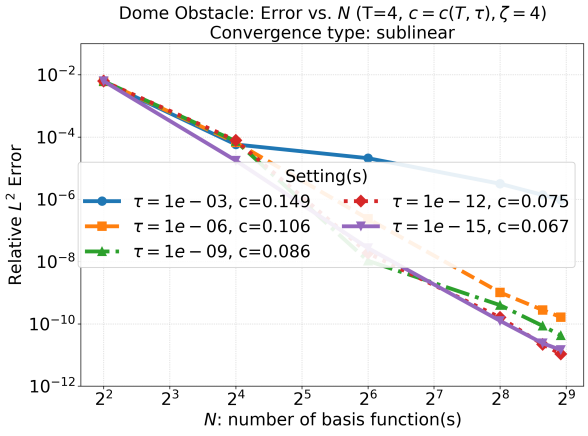
(a) Dome obstacle problem (2D), $T = 1.5$



(b) Dome obstacle problem (2D), $T = 2.0$

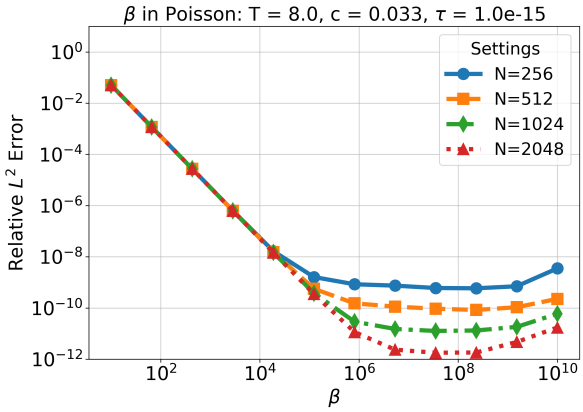


(c) Dome obstacle problem (2D), $T = 3.0$

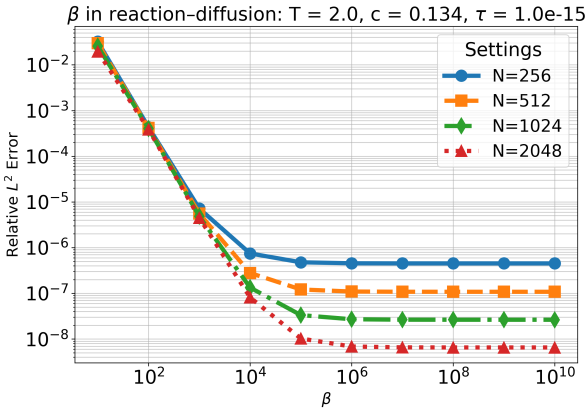


(d) Dome obstacle problem (2D), $T = 4.0$

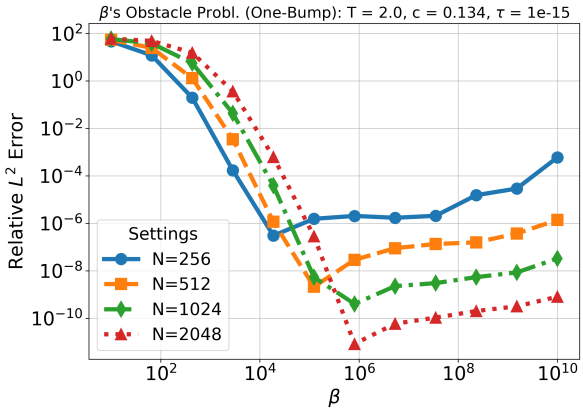
Figure 3.21: **Sensitivity of the truncation threshold τ on the approximation error for the 2D dome obstacle problem.** Relative ℓ^2 error versus N for fixed expansion factor T and oversampling ratio $\zeta = 4$. Each panel compares $\tau \in \{10^{-3}, 10^{-6}, 10^{-9}, 10^{-12}, 10^{-15}\}$, with $c = c(T, \tau)$. Smaller values of τ tend to yield lower error levels and more robust decay, particularly for larger N. Here, we use the kernel shape relationship $b = c(T, \tau)\sqrt{N}$.



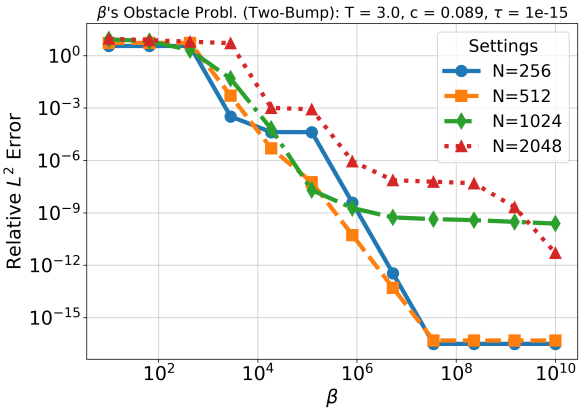
(a) Poisson problem (1D), $T = 8.0$



(b) Reaction-diffusion problem (1D), $T = 2.0$

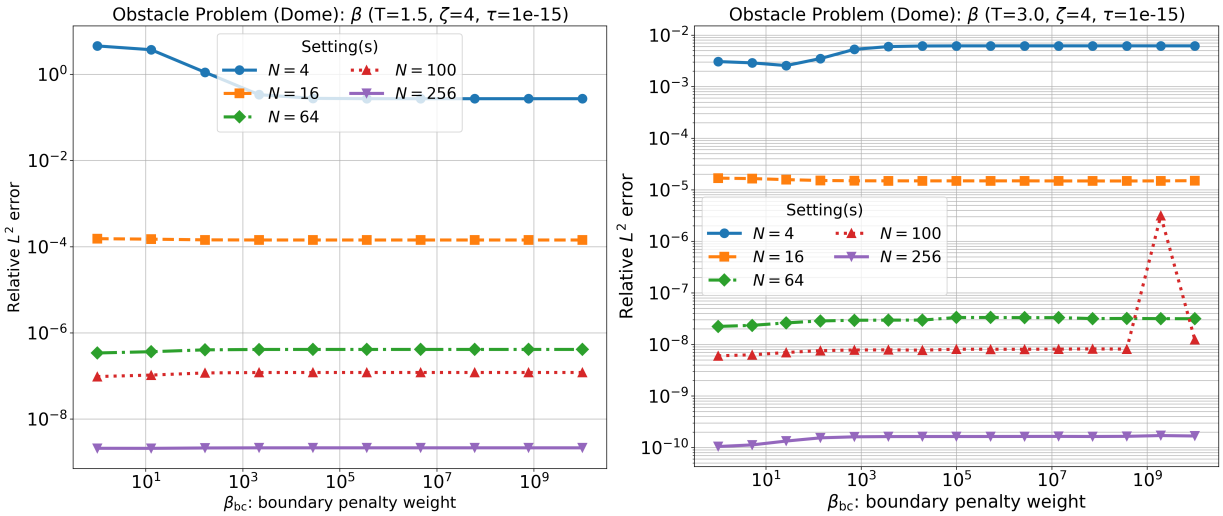


(c) Single bump obstacle problem (1D), $T = 2.0$



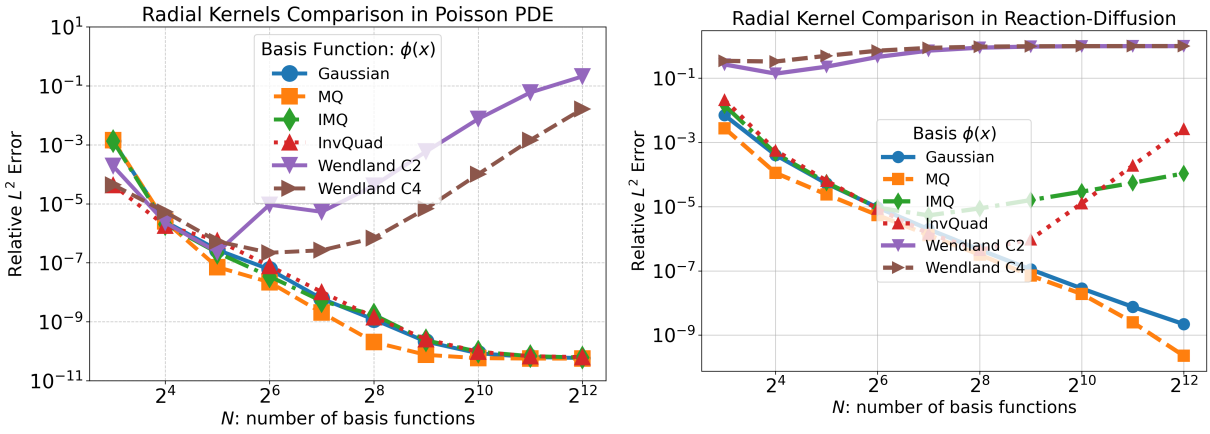
(d) Two bump problem (1D), $T = 3.0$

Figure 3.22: **Influence of the penalty parameter β in the boundary condition.** Relative ℓ^2 error versus β for the 1D Poisson, reaction-diffusion, single-bump obstacle, and two bump obstacle problems, with fixed T, τ, and $c = c(T, \tau)$. Each panel compares several values of N. The results indicate that each problem admits a suitable β-range balancing boundary enforcement and numerical stability; the selected values are $\beta = 3 \times 10^5$ for Poisson equation, $\beta = 10^6$ for reaction-diffusion, $\beta = 10^6$ for the single bump obstacle problem, and $\beta = 10^8$ for the two bump obstacle problem.



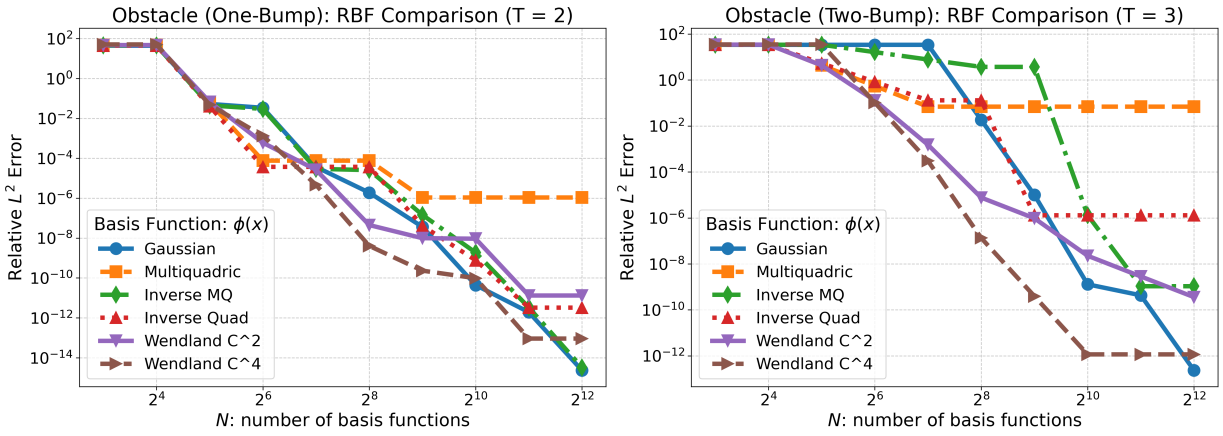
(a) Dome obstacle problem (2D), $T = 1.5$ (b) Dome obstacle problem (2D), $T = 3.0$

Figure 3.23: **Influence of the boundary penalty weight β on the radially symmetric obstacle problem.** Relative ℓ^2 error versus β for the 2D dome obstacle problem with fixed expansion factor T, oversampling ratio $\zeta = 4$, and truncation threshold $\tau = 10^{-15}$. Each panel compares several values of N. The results indicate that the relative error is stable across a wide range of β, with only limited variation in the final error. Here, we use the scaling for the kernel shape $b = c\sqrt{N}$.

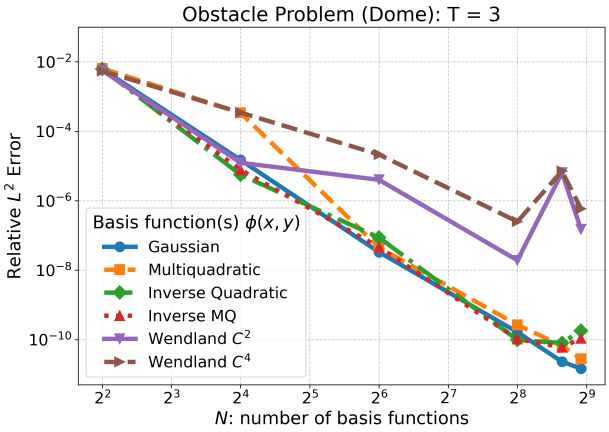


(a) $T = 8.0, \beta = 3 \times 10^5$ (b) $T = 2.0, \beta = 10^6$

Figure 3.24: **Comparison of the radial kernels for the Poisson and reaction-diffusion problems in 1D.** Relative ℓ^2 error versus the number of basis functions N for several radial kernels, including Gaussian, multiquadric (MQ), inverse multiquadric (IMQ), inverse quadratic (InvQuad), Wendland C^2, and Wendland C^4. Panel (a) shows the Poisson problem with $T = 8.0$ and $\beta = 3 \times 10^5$, while panel (b) shows the reaction-diffusion problem with $T = 2.0$ and $\beta = 10^6$. In both cases, the Gaussian, MQ, IMQ, and inverse quadratic kernels yield fast convergence as N increases, whereas the compactly supported Wendland kernels exhibit lower accuracy, especially for large N. Overall, the Gaussian-type globally supported kernels provide the most accurate and fast convergence behavior for these two model problems.



(a) $T = 2.0, \beta = 10^6$ (b) $T = 2.0, \beta = 10^8$



(c) $T = 3.0, \beta = 10^6$

Figure 3.25: **Comparison of the radial kernels for obstacle problems.** Relative ℓ^2 error versus N for several radial kernels in three obstacle-type test problems: the 1D single-bump problem, the 1D two bump problem, and the 2D dome problem. In all three cases, the Gaussian-type globally supported kernels show stronger convergence behavior and attain smaller final errors than the compactly supported Wendland kernels, supporting their use in the subsequent numerical experiments. Here, for 1D obstacle problem we use the linear scaling for kernel shape parameter $b = c\,N$ while in the two-dimensional setting, we let $b = c\sqrt{N}$.

3.6 Limitations

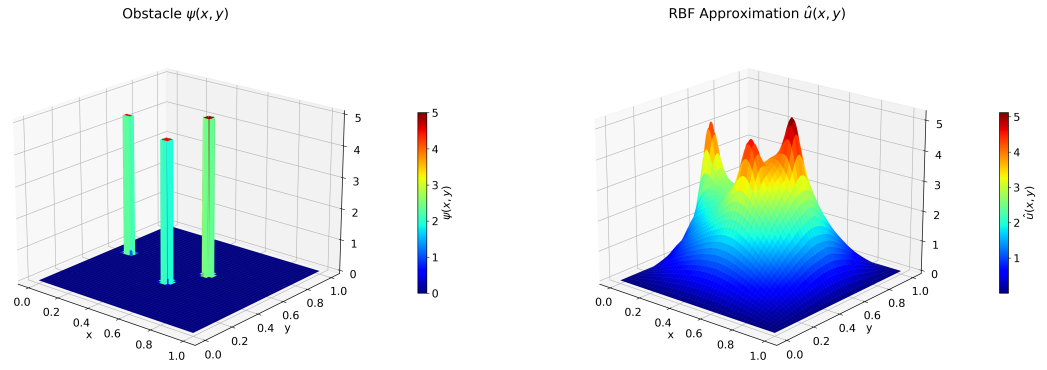
Although the proposed variational RBF method performs well for the benchmark problems considered in this chapter, several limitations remain. In particular, difficulties arise when the obstacle function is discontinuous because a globally smooth basis saturates sharp features and diminishes local accuracy. Additionally, the method is based on dense linear systems, which means that rank efficiency and computation time become important practical concerns as the number of basis functions N increases.

3.6.1 Discontinuous Obstacle Problem

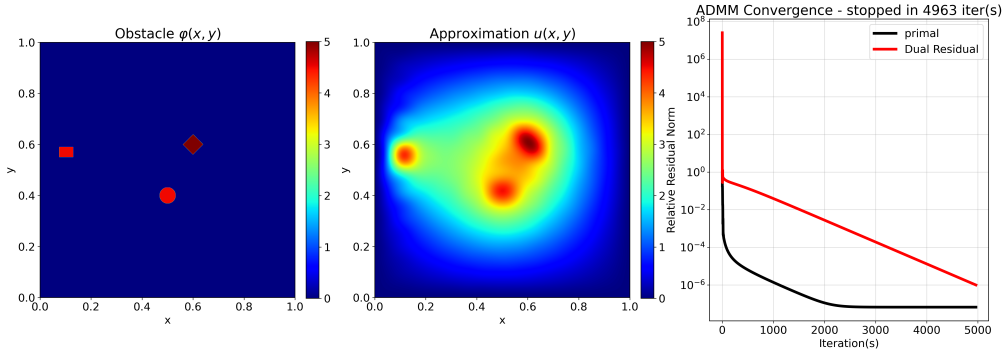
In this subsection, we consider the following discontinuous obstacle function from [107, 125]. The free-BVP is solved using Algorithm 2. We let $\overline{\Omega} = [0,1]^2$, and the obstacle function is defined as follows:

$$\psi(x,y) = \begin{cases} 5.0, & |x-0.6| + |y-0.6| < 0.04, \\ 4.5, & (x-0.6)^2 + (y-0.25)^2 > 0.001, \\ 4.5, & \text{for} \quad y = 0.57 \quad \text{and} \quad 0.075 < x < 0.13, \\ 0, & \text{otherwise.} \end{cases} \tag{3.44}$$

The obstacle problem with the function ψ in Eq. 3.44 consists of different disjoint shapes within the domain $[0,1] \times [0,1]$, as shown in the left panels of Figure 3.26(a) and (b). However, our RBF method does not provide an accurate solution in this case. As shown in the right and middle panels of Figure 3.26(a)(b), it is clear that the approximation behaves smoothly away from the obstacle; however, it fails to accurately represent the region where the obstacle is active on its support level set, despite the convergence in ADMM after 4963 iterations. We maintain the same hyperparameters as in the numerical experiment with the domain obstacle problem in Section 3.4.2. For the ADMM parameters, we choose $(\mu, \rho) = (6.5 \times 10^5, 1.3 \times 10^4)$ in [107]. The discontinuous obstacle problem in Figure 3.26 exemplifies the limitations of the variational RBF framework when the contact set is associated with disjoint shapes and sharp transitions. Because the Gaussian kernel is globally smooth, it tends to smooth away the discontinuous features of the obstacle, reducing the accuracy of the contact region when the ADMM has converged. Adding more basis functions would improve the solution accuracy. This limitation motivates future work on localized basis functions and adaptive refinement.



(a) Surface plots of the discontinuous obstacle function $\psi(x, y)$ and the corresponding RBF approximation $\hat{u}(x, y)$.



(b) Contour plots of the obstacle function $\psi(x, y)$ and the RBF approximation $\hat{u}(x, y)$, and the convergence history of the primal and dual residuals in the ADMM iteration.

Figure 3.26: **RBF approximation of the two-dimensional obstacle problem with a discontinuous obstacle function using the variational method**. The figure compares the obstacle profile and the computed RBF approximation in both surface and contour representations, together with the convergence history of the ADMM iterations. The number of basis functions N is set at 2048, oversampling ratio $\zeta = 4$, and $\beta = 10^6$. For ADMM and TSVD parameters, we let the truncation threshold value $\tau = 10^{-15}$, and $(\mu, \rho) = (6.5 \times 10^5, 1.3 \times 10^4)$ as in [107].

3.6.2 Numerical Rank Efficiency and Computational Time

This subsection examines the numerical rank efficiency and computational cost of the linear systems in the variational RBF framework. To begin, we consider the one-dimensional Poisson problem in the experiment of Section 3.4.1 as the baseline. We quantify the usefulness of the basis representation with the rank efficiency of $A \in \mathbb{R}^{m \times N}$, which is given by:

$$\eta_{\text{rank}} \coloneqq \frac{r_{\text{eff}}}{N}, \tag{3.45}$$

where $r_{\text{eff}} = \text{card}(\mathcal{T}_\tau)$ and we recall $\mathcal{T}_\tau = \{i : \sigma_i > \tau\, \sigma_1\}$ as mentioned in Definition 12.

In Eq. 3.45, r_{eff} is the effective numerical rank after TSVD truncation. The ratio η_{rank} represents the fraction of basis functions N that remain numerically informative. Figure 3.27 shows that the rank efficiency decreases as N increases, indicating growing redundancy and ill-conditioning in the system. The right panel in Figure 3.27 shows that the computational time rises rapidly, approximately following the expected $\mathcal{O}(N^3)$ complexity for large N. In general, the results show that increasing the number of basis functions N leads to substantially higher computational costs without a proportional increase in effective approximation capacity. This demonstrates that the current variational RBF formulation has limited scalability for high-dimensional PDEs and large-scale problems. This limitation motivates the development of scalable strategies, such as adaptive basis selection.

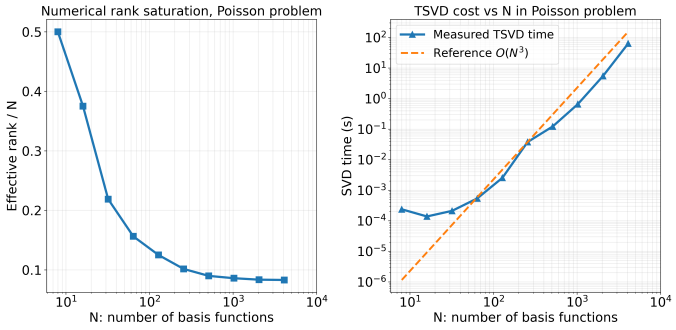


Figure 3.27: **Rank efficiency and computational cost in the one-dimensional Poisson problem**. The left panel plots the effective numerical rank normalized by N, showing that the proportion of useful modes decreases as the number of basis functions increases. The right panel shows the measured TSVD runtime together with a reference $O(N^3)$, confirming the growth in computational cost for large N.

3.7 Summary

In this chapter, we have presented a mesh-free variational framework to solve the linear elliptic BVPs and obstacle-type PDEs using the RBF approximation according to Definition 7. To address the ill-conditioning inherent in global RBF discretizations, we incorporated TSVD as a stabilization mechanism in the least-squares solver. We extend the methodology to solve classical obstacle problems by introducing a variational formulation with ℓ^1-penalty on the contact constraint in Section 3.3. The resulting non-smooth optimization problem is efficiently solved using the ADMM, as described in Section 3.3.2. In our methodology of Sections 3.3.1-3.3.2, we enlarge the effective basis support through the expansion factor T. The kernel parameter is chosen using linear scaling $b = c(T, \tau)\, N$ in one dimension and $b = c(T, \tau)\, \sqrt{N}$ in two dimensions, which leads to stable convergence and high accuracy. Extensive numerical experiments in Section 3.4 confirm that our RBF solution method achieves fast convergence for linear elliptic and obstacle-type BVPs. The results offer pragmatic guidance for selecting the (N, T, τ) parameters to balance accuracy, condition number, and computational stability. There are future research directions that include:

- Generalizing the framework to high-dimensional PDEs, including adaptive center distributions.
- Investigating alternative penalty formulations (e.g., Huber or augmented Lagrangian methods) in order to improve conditioning.
- Developing a systematic theoretical characterization of the trade-offs between hyperparameters (N, T, τ) in terms of accuracy and conditioning by deriving a relation, such as in Eq. (3.10), specifically tailored to our variational RBF method in multiple dimensions.
- Extending the framework to other nonlinear PDEs and time-dependent variational problems.

Chapter 4

Random-Feature Solvers for High-Dimensional Variational Problems

[2]This chapter is based on the following paper: Tan Phuong Dong Le, Giang Tran, and Hans De Sterck. *Random-feature solvers for high-dimensional variational problems: A practical framework.* Preprint, 2026.

4.1 Introduction

RFM methods constitute a framework for the numerical approximation of high-dimensional PDEs, as they have advantageous approximation capabilities relative to the number of features. In this chapter, we propose a mesh-free RFM framework for high-dimensional variational elliptic PDEs and free-BVPs. The method approximates the solution using a random-feature expansion and determines the coefficients through discrete variational principles. This results in convex finite-dimensional optimization problems: quadratic minimization for linear elliptic PDEs and a nonsmooth constrained minimization for free-BVPs, which are solved by the ADMM.

The proposed framework is particularly efficient in high dimensions, where mesh-based discretizations are infeasible and fully trained neural-network methods may be computationally demanding. Boundary conditions are imposed through penalty terms, which preserve a mesh-free formulation. We further investigate the effects of the variance scale and domain expansion parameters on the accuracy and robustness of the approximation, and we compare the method with the radial basis approximation from Chapter 3 and neural-network approaches.

Numerical results for various high-dimensional elliptic and free-BVPs demonstrate that the proposed RFM solvers are accurate, efficient, and scalable. In particular, the approximation error consistently decreases as the number of random features increases. The method compares favorably with radial basis approximation schemes at a lower memory cost and achieves prescribed tolerances faster than gradient-trained neural-network solvers. These results indicate that our variational random-feature methods constitute a practical computational framework for high-dimensional PDEs.

4.2 Problem Statement and Motivation

Chapter 3 introduced the variational formulation of elliptic PDEs and free-BVPs in the context of RBF models. In Section 4.2, we consider the same class of variational problems but focus on the high-dimensional regime and replace the RBF approximation space with the RFM approach. The main objective is not to reformulate the underlying PDE model but rather to develop a computationally efficient mesh-free solver that remains practical when the spatial dimension d becomes moderately large.

The motivation for this shift is the increasing computational difficulty associated with high-dimensional PDEs. Classical grid-based discretizations suffer from the curse of dimensionality, while local kernel methods may become expensive in terms of memory and

computation as the approximation space expands. RFM methods provide an alternative by fixing the nonlinear features. This results in an explicit low-complexity approximation framework that is sufficient compared to fully trained neural networks and potentially more scalable than local kernel-based methods. Accordingly, this chapter investigates variational RFM solvers for high-dimensional elliptic and free-boundary problems, with particular emphasis on computational scalability, approximation accuracy, and parameter sensitivity. To recall RFM approximations, we refer readers to Definition 10 in Section 2.4. We let $\vec{x} \in \overline{\Omega} \subseteq \mathbb{R}^d$ and consider the function $\hat{u}_{\mathrm{RF}} : \overline{\Omega} \to \mathbb{R}$, which is given by:

$$\hat{u}_{\mathrm{RF}}(\vec{x}) = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} d_i \, \sigma\big(\vec{\omega}_i^{\top} \vec{x} + b_i\big) = \frac{1}{\sqrt{M}} \, \vec{d}^{\top} \sigma\Big(W\vec{x} + \vec{b}\Big) , \tag{4.1}$$

where $W = [\omega_{i,j}] \in \mathbb{R}^{M \times d}$ denotes a random matrix whose entries are drawn according to a probability density function $\rho(\vec{\omega})$, and $\vec{b} \in \mathbb{R}^M$ is a corresponding bias vector. The coefficient vector $\vec{d} \in \mathbb{R}^M$ is trainable. Here, the factor $1/\sqrt{M}$ is introduced to normalize the random-feature map with respect to the number of features M, thereby preventing the magnitude of the approximation from growing with M.

To fully define the random feature approximation, we now introduce the sampling procedure in Assumption 1 for the random weight and bias parameters, as well as Assumption 2 on the kernel function employed in the RFM in Eq. (4.1).

Assumption 1 (Sampling distribution of random features [84, 85, 86]) *Let $\Omega = [q_1, q_2]^d$ and M be the number of random features. For each $i = 1, \ldots, M$, let*

$$\vec{\omega}_i \in \mathbb{R}^d, \qquad b_i \in \mathbb{R},$$

be the random parameters associated with the i-th feature. We assume that the random vectors $\vec{\omega}_1, \ldots, \vec{\omega}_M$ and the bias terms $b_1, \ldots, b_M$ are independently and identically distributed. Specifically,

$$\vec{\omega}_1, \ldots, \vec{\omega}_M \overset{\text{i.i.d.}}{\sim} \mathcal{N}\Big(\vec{0}, \, b_{\mathrm{RF}}^2 I_d\Big) , \qquad b_1, \ldots, b_M \overset{\text{i.i.d.}}{\sim} \mathcal{U}\big(\overline{\Omega}_{T_{\mathrm{RF}},d}\big) ,$$

where

$$\overline{\Omega}_{T_{\mathrm{RF}},d} = \Big[q_{\mathrm{mid}} - T_{\mathrm{RF}} \, L_{\mathrm{half}} \sqrt{d}, \; q_{\mathrm{mid}} + T_{\mathrm{RF}} \, L_{\mathrm{half}} \sqrt{d}\Big] ,$$

with

$$q_{\mathrm{mid}} := \frac{q_1 + q_2}{2}, \qquad L_{\mathrm{half}} := \frac{q_2 - q_1}{2}.$$

Here, $b_{\mathrm{RF}} > 0$ controls the variance scale of the random-feature map, while the expansion factor $T_{\mathrm{RF}} > 0$ determines the sampling range of the bias parameters. Accordingly, the random matrix $W \in \mathbb{R}^{M\times d}$ and the bias vector $\vec{b} \in \mathbb{R}^M$ are defined by

$$W = \begin{bmatrix} \vec{\omega}_1^\top \\ \vdots \\ \vec{\omega}_M^\top \end{bmatrix} \quad \text{and} \quad \vec{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix}.$$

Assumption 2 (Hyperbolic tangent activation function [60]) *The activation function $\sigma : \mathbb{R}^M \to \mathbb{R}^M$ in Eq. (4.1) is assumed to be the hyperbolic tangent function applied element-wise. More precisely, for any $\vec{z} = (z_1, \ldots, z_M)^\top \in \mathbb{R}^M$, we define*

$$\sigma(\vec{z}) = (\tanh(z_1), \tanh(z_2), \ldots, \tanh(z_M))^\top,$$

This choice yields a bounded nonlinear feature map and provides a stable scaling for the variational random feature approximation considered in this chapter.

Assumption 1 details the construction of the random feature functions via sampling the weight vectors and bias parameters. Drawing the weights from a Gaussian distribution yields isotropic random directions in $\mathbb{R}^d$, whereas using a uniform distribution over the expanded domain $\overline{\Omega}_{T,d}$ broadens the sampling range of the bias terms beyond the physical domain. The parameters b_{RF} and T_{RF} are crucial in regulating the expressivity, numerical stability, and spatial coverage of the resulting random feature approximation. When alternative activation functions are employed, as shown in Table 2.2, the sampling distribution for the bias vector must be modified. For instance, if we use cosine and sine features instead, we change the domain of the probability density function as follows:

$$b_i \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0, 2\pi), \qquad i = 1, \ldots, M.$$

Since the random weights and biases are fixed after sampling, and only the output coefficients are optimized, the resulting variational formulation in RFM is a convex quadratic problem for linear elliptic PDEs. Assumption 2 specifies the particular nonlinear feature map in this study. The use of the hyperbolic tangent function yields a smooth, bounded activation, which is compatible with the variational formulation and promotes stable numerical computation. The additional scaling factor $1/\sqrt{M}$ serves to normalize the RFM, as introduced in Definition 10. In the next section, we use the RFM in an optimization framework by employing a variational formulation to solve linear elliptic and obstacle-type BVPs.

4.3 Algorithmic Framework

We develop the numerical algorithms associated with the variational framework for the high-dimensional BVPs examined in this chapter. Similar to Chapter 3, we consider the high-dimensional elliptic and obstacle-type BVPs, utilizing the RFM in Eq. (4.1) in the variational equations. This yields an optimization problem to minimize the unknown variables $\vec{d} \in \mathbb{R}^M$, as presented in Eq. (4.1). The precise form of the discrete optimization problem depends on the underlying variational framework. For linear elliptic BVPs, the resulting formulation is a convex quadratic minimization problem. For obstacle-type BVPs, however, the variational formulation contains additional nonsmooth terms, which lead to a constrained optimization problem that is solved using the ADMM. We next derive these two cases separately and present the corresponding solvers.

4.3.1 Random-Feature Solver for High-Dimensional Linear Elliptic Boundary Value Problems

In this section, we consider high-dimensional linear elliptic PDEs, which were also discussed in Section 3.3.1 in Chapter 3. Specifically, we examine the high-dimensional Poisson and Darcy flow BVPs, which are studied and formulated as discrete optimization problems. We let $\Omega \subseteq \mathbb{R}^d$ be a bounded domain. We consider the following linear elliptic BVP:

$$\begin{cases} -\nabla \cdot \big(a(\vec{x}) \nabla u(\vec{x})\big) = f(\vec{x}), & \vec{x} \in \Omega, \\ u(\vec{x}) = g(\vec{x}), & \vec{x} \in \partial\Omega, \end{cases} \tag{4.2}$$

where $a : \Omega \to \mathbb{R}$ denotes the diffusion coefficient, $f : \Omega \to \mathbb{R}$ is the source term, and $g : \partial\Omega \to \mathbb{R}$ is the prescribed Dirichlet boundary function.

Similar to the derivation in Section 2.2, the variational formulation corresponding to Eq. (4.2) is given by:

$$\min_u \mathcal{J}_{\text{darcy},\beta}(u), \tag{4.3}$$

where the penalized functional $\mathcal{J}_{\text{darcy},\beta} : H^1(\Omega) \to \mathbb{R}$ is defined as:

$$\mathcal{J}_{\text{darcy},\beta}(u) = \frac{1}{2} \int_\Omega a(\vec{x}) \, \|\nabla u(\vec{x})\|_2^2 \, d\,\vec{x} - \int_\Omega f(\vec{x}) \, u(\vec{x}) \, d\,\vec{x} + \frac{\beta}{2} \int_{\partial\Omega} \big(u(\vec{s}) - g(\vec{s})\big)^2 \, d\vec{s}. \tag{4.4}$$

In the same manner as in Section 3.3.1, we select m_Ω interior sampling points in Ω, $\vec{x}_1^{(\Omega)}, \ldots, \vec{x}_{m_\Omega}^{(\Omega)}$, and $m_{\partial\Omega}$ boundary sampling points in $\partial\Omega$, $\vec{x}_1^{(\partial\Omega)}, \ldots, \vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}$. Then, the dis-

crete variational equation is given by:

$$\min_{\vec{d}\in\mathbb{R}^M} \hat{\mathcal{J}}_{\beta,\text{darcy}}(\hat{u}_{\text{RF}}), \tag{4.5}$$

where

$$\begin{aligned}\hat{\mathcal{J}}_{\beta,\text{darcy}}(\hat{u}) = \frac{|\Omega|}{m_\Omega}\sum_{j=1}^{m_\Omega}\left(\frac{1}{2}\,a\left(\vec{x}_j^{(\Omega)}\right)\left\|\nabla\hat{u}\left(\vec{x}_j^{(\Omega)}\right)\right\|_2^2 - f\left(\vec{x}_j^{(\Omega)}\right)\,\hat{u}\left(\vec{x}_j^{(\Omega)}\right)\right)\\ + \frac{\beta\,|\partial\Omega|}{2\,m_{\partial\Omega}}\sum_{\ell=1}^{m_{\partial\Omega}}\left(\hat{u}\left(\vec{x}_\ell^{(\partial\Omega)}\right) - g\left(\vec{x}_\ell^{(\partial\Omega)}\right)\right)^2.\end{aligned} \tag{4.6}$$

Using the Jacobian matrix $J(\vec{x}) \in \mathbb{R}^{d\times M}$, which is defined in Section 2.6.2 as

$$J(\vec{x}) = \Big[\sigma'(\vec{\omega}_1^\top\,\vec{x}+b_1)\,\vec{\omega}_1\;\cdots\;\sigma'(\vec{\omega}_M^\top\,\vec{x}+b_M)\,\vec{\omega}_M\Big],$$

We obtain for $j = 1,\ldots,m_\Omega$

$$a\left(\vec{x}_j^{(\Omega)}\right)\left\|\nabla\hat{u}_{\text{RF}}\left(\vec{x}_j^{(\Omega)}\right)\right\|_2^2 = a\left(\vec{x}_j^{(\Omega)}\right)\left\|J\left(\vec{x}_j^{(\Omega)}\right)\,\vec{d}\right\|_2^2.$$

Hence, we deduce that

$$\widehat{J}_{\beta,\text{darcy}}(\hat{u}) = \frac{1}{2}\,\vec{d}^\top\,B_{1,\text{darcy}}\,\vec{d} - \vec{f}^\top\,B_2\,\vec{d} + \frac{\beta}{2}\left\|B_3\,\vec{d}-\vec{g}\right\|_2^2, \tag{4.7}$$

where

$$B_{1,\text{darcy}} = \frac{|\Omega|}{M\,m_\Omega}\sum_{j=1}^{m_\Omega} a\big(\vec{x}_j^{(\Omega)}\big)\,J\big(\vec{x}_j^{(\Omega)}\big)^\top J\big(\vec{x}_j^{(\Omega)}\big) \in \mathbb{R}^{M\times M}, \tag{4.8}$$

$$B_2 = \frac{|\Omega|}{\sqrt{M}\,m_\Omega}\begin{bmatrix}\sigma\left(W\,\vec{x}_1^{(\Omega)}+\vec{b}\right)^\top\\ \vdots\\ \sigma\left(W\,\vec{x}_{m_\Omega}^{(\Omega)}+\vec{b}\right)^\top\end{bmatrix} \in \mathbb{R}^{m_\Omega\times M},$$

and

$$B_3 = \sqrt{\frac{|\partial\Omega|}{M\,m_{\partial\Omega}}}\begin{bmatrix}\sigma\left(W\,\vec{x}_1^{(\partial\Omega)}+\vec{b}\right)^\top\\ \vdots\\ \sigma\left(W\,\vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}+\vec{b}\right)^\top\end{bmatrix} \in \mathbb{R}^{m_{\partial\Omega}\times M}.$$

The discrete variational formulation for Poisson's equation is

$$\min_{\vec{d}\in\mathbb{R}^M} \frac{1}{2}\,\vec{d}^{\top} B_1\,\vec{d} - \vec{f}^{\top}\,B_2\,\vec{d} + \frac{\beta}{2}\,\|B_3\,\vec{d} - \vec{g}\|_2^2, \tag{4.9}$$

where

$$B_{1,\text{poisson}} = \frac{|\Omega|}{M\,m_\Omega}\sum_{j=1}^{m_\Omega} J\big(\vec{x}_j^{(\Omega)}\big)^{\top} J\big(\vec{x}_j^{(\Omega)}\big) \in \mathbb{R}^{M\times M}.$$

The load and boundary data matrices B_2 and B_3 are defined as above, see also Eqs. (2.41)-(2.42). For the remaining of the section, we use B_1 to mean either $B_{1,\text{poisson}}$ or $B_{1,\text{darcy}}$.

We apply the TSVD and its pseudo-inverse in the same manner as in Section 3.2.1. The minimizer $\vec{d}^*$ satisfies the following system:

$$B_{\text{elliptic}}\,\vec{d}^* = \vec{y}_{\text{elliptic}}, \tag{4.10}$$

where

$$B_{\text{elliptic}} \coloneqq B_1 + \beta\,B_3^{\top}\,B_3 \in \mathbb{R}^{N\times N}, \qquad \vec{y}_{\text{elliptic}} \coloneqq B_2^{\top}\,\vec{f} + \beta\,B_3^{\top}\,\vec{g}. \tag{4.11}$$

We compute the pseudo-inverse of the TSVD, as stated in Definition 13, which gives

$$\vec{d}_\tau = B_{\text{elliptic},\tau}^{\dagger}\,\vec{y}_{\text{elliptic}} = V_p\,\Sigma_p^{-1}\,U_p^{\top}\,\vec{y}_{\text{elliptic}}, \tag{4.12}$$

where

$$\Sigma_p^{-1} = \operatorname{diag}\Big(\frac{1}{\sigma_1},\ldots,\frac{1}{\sigma_p}\Big), \qquad \text{with} \quad \sigma_p > \sigma_1\,\tau \geq \sigma_{p+1}. \tag{4.13}$$

By [5], the TSVD solution yields an accurate approximation $\vec{d}_\tau$, which we use in

$$\hat{u}_{\text{RF},\tau}(\vec{x}) = \frac{1}{\sqrt{M}}\sum_{i=1}^{M} d_{(\tau),i}\,\sigma\big(\vec{\omega}_i^{\top}\,\vec{x} + b_i\big). \tag{4.14}$$

Accordingly, Algorithm 3 presents the variational RFM solver for linear elliptic PDEs.

Algorithm 3: Stable RFM solver for high-dimensional elliptic PDEs using TSVD

Input: $\sigma(\cdot)$ (activation function); M (number of random features); $b_{\mathrm{RF}} > 0$ (variance scale); $T_{\mathrm{RF}} > 0$ (domain expansion factor); $\left\{\vec{x}_j^{(\Omega)}\right\}_{j=1}^{m_\Omega}$ (interior nodes); $\left\{\vec{x}_n^{(\partial\Omega)}\right\}_{n=1}^{m_{\partial\Omega}}$ (boundary nodes); $f(\cdot)$, $g(\cdot)$ (source/boundary data); $\beta > 0$ (boundary penalty parameter); $\tau > 0$ (truncation tolerance)

Output: coefficient vector $\vec{d}_\tau \in \mathbb{R}^M$; RFM approximation $\hat{u}_{\mathrm{RF},\tau} : \Omega \to \mathbb{R}$

Generate random features:

Sample $W \in \mathbb{R}^{M\times d}$ row-wise by $\vec{\omega}_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\vec{0}, b_{\mathrm{RF}}^2 I_d)$

Sample $\vec{b} \in \mathbb{R}^M$ componentwise by $b_i \overset{\text{i.i.d.}}{\sim} \mathcal{U}(\overline{\Omega}_{T_{\mathrm{RF}},d})$

Assemble discrete matrices:

Compute $B_1 \in \mathbb{R}^{M\times M}$ as either in Eq. (2.40) or Eq. (4.8)

Compute $B_2 \in \mathbb{R}^{m_\Omega \times M}$ as in Eq. (2.41) and $B_3 \in \mathbb{R}^{m_{\partial\Omega}\times M}$ as in Eq. (2.42)

Form vectors $\vec{f}$, $\vec{g}$

Build the normal-equation system according to Eq. (4.11)

$$B_{\mathrm{elliptic}} \leftarrow B_1 + \beta\, B_3^\top B_3, \qquad \vec{y}_{\mathrm{elliptic}} \leftarrow B_2^\top \vec{f} + \beta\, B_3^\top \vec{g}.$$

Stabilize via TSVD:

Compute the SVD: $B_{\mathrm{elliptic}} = U\Sigma V^\top$

Identify the index set $\mathcal{T}_\tau = \{i : \sigma_i > \tau\sigma_1\}$

Form Σ_p^{-1}, where $(\Sigma_p^{-1})_{i,i} = 1/\sigma_i$ if $i \in \mathcal{T}_\tau$ and 0 otherwise

Form the pseudo-inverse matrix $B^\dagger_{\mathrm{elliptic},\tau} \leftarrow V_p \Sigma_p^{-1} U_p^\top$

Compute the coefficient vector in RFM using Eq. (4.10)**:**

$$\vec{d}_\tau \leftarrow B^\dagger_{\mathrm{elliptic},\tau}\, \vec{y}_{\mathrm{elliptic}}.$$

return: $\vec{d}_\tau \in \mathbb{R}^M$ and $\hat{u}_{\mathrm{RF},\tau}(\vec{x}) = \frac{1}{\sqrt{M}} \sum_{i=1}^M d_{(\tau),i}\, \sigma\big(\vec{\omega}_i^\top \vec{x} + b_i\big)$.

4.3.2 Random-Feature Solver for High-Dimensional Free-Boundary Problems

In this section, we continue the development of the variational random feature method for obstacle-type PDEs, based on Algorithm 3. Similar to Section 3.3.2 in Chapter 3, we extend the previously introduced variational RBF framework to a mesh-free RFM and adapt it to a formulation that can approximate the solution of obstacle-type PDEs. In the same manner, we start by selecting m collocation points from the domain $\overline{\Omega}$, which consists of m_Ω interior points $\vec{x}_1^{(\Omega)}, \ldots, \vec{x}_{m_\Omega}^{(\Omega)}$ and $m_{\partial\Omega}$ boundary points $\vec{x}_1^{(\partial\Omega)}, \ldots, \vec{x}_{m_{\partial\Omega}}^{(\partial\Omega)}$. From Eq. (3.18)

and utilizing the RFM in Eq. (4.1), the resulting discrete form of the obstacle problem is given by:

$$\hat{\mathcal{J}}_{\mu,\beta,\text{obstacle}}\left(\vec{d}\right) = \frac{1}{2}\,\vec{d}^{\top}\,B_1\,\vec{d} + \mu\,\left\|\left(\vec{\psi} - B_2\,\vec{d}\right)_+\right\|_1 + \frac{\beta}{2}\,\|B_3\,\vec{d} - \vec{g}\|_2^2, \tag{4.15}$$

where the matrices B_1, B_2, B_3 are defined as in Eq. (2.40), (2.41), and (2.42), respectively. The vector for the obstacle function is given by:

$$\vec{\psi} = \frac{|\Omega|}{m_\Omega}\begin{bmatrix} \psi\left(\vec{x}_1^{(\Omega)}\right) \\ \vdots \\ \psi\left(\vec{x}_{m_\Omega}^{(\Omega)}\right) \end{bmatrix} \in \mathbb{R}^{m_\Omega}. \tag{4.16}$$

The optimization problem $\min\limits_{\vec{d}\in\mathbb{R}^M} \hat{\mathcal{J}}_{\mu,\beta,\text{obstacle}}(\vec{d})$ is an unconstrained problem that includes an ℓ^1-norm term, and is convex but not differentiable. We consider applying the ADMM, which is outlined in Section 2.7.

We let $\vec{v} = \vec{\psi} - B_2\,\vec{d}$ be the slack variable and reformulate the optimization problem into an equivalent constrained problem, which is given by:

$$\begin{cases} \min\limits_{\vec{d},\,\vec{v}} \hat{\mathcal{J}}_{\mu,\beta,\text{obstacle}}(\vec{d},\vec{v}), \\ \hat{\mathcal{J}}_{\mu,\beta,\text{obstacle}}(\vec{d}) = \frac{1}{2}\,\vec{d}^{\top}\,B_1\,\vec{d} + \mu\,\|\,(\vec{v})_+\,\|_1 + \frac{\beta}{2}\,\|B_3\,\vec{d} - \vec{g}\|_2^2, \\ \text{such that} \quad \vec{v} - \vec{\psi} + B_2\,\vec{d} = 0. \end{cases} \tag{4.17}$$

We form the associated augmented Lagrangian functional, penalized by the parameter $\rho > 0$ and with a scaled dual variable $\vec{z}$

$$\widehat{L}_\rho(\vec{d},\vec{v},\vec{z}) = \frac{1}{2}\,\vec{d}^{\top}\,B_1\,\vec{d} + \frac{\beta}{2}\,\|B_3\,\vec{d} - \vec{g}\|_2^2 + \mu\,\|\,(\vec{v})_+\,\|_1 + \frac{\rho}{2}\|\vec{v} - \vec{\psi} + B_2\,\vec{d} + \vec{z}\|_2^2 - \frac{\rho}{2}\|\vec{z}\|_2^2. \tag{4.18}$$

By Definition 21, the algorithm updates the primal and dual variables at each iteration $i = 0, \ldots, \texttt{maxIter}$, as follows:

$$\vec{d}^{(i+1)} = \arg\min_{\vec{d}} \left\{ \frac{1}{2}\,\vec{d}^{\top} B_1 \vec{d} + \frac{\beta}{2}\|B_3\vec{d} - \vec{g}\|_2^2 + \frac{\rho}{2}\left\|\vec{v}^{(i)} - \vec{\psi} + B_2\vec{d} + \vec{z}^{(i)}\right\|_2^2 \right\}, \tag{4.19}$$

$$\vec{v}^{(i+1)} = \arg\min_{\vec{v}} \left\{ \mu\|\vec{v}_+\|_1 + \frac{\rho}{2}\left\|\vec{v} - \vec{\psi} + B_2\vec{d}^{(i+1)} + \vec{z}^{(i)}\right\|_2^2 \right\}, \tag{4.20}$$

$$\vec{z}^{(i+1)} = \vec{z}^{(i)} + \left(\vec{v}^{(i+1)} - \vec{\psi} + B_2\vec{d}^{(i+1)}\right). \tag{4.21}$$

The $\vec{d}$-subproblem is a quadratic minimization problem, and its minimizer $\vec{d}^{(i+1)}$ is characterized by the corresponding first-order optimality condition:

$$\left(B_1 + \beta B_3^{\top} B_3 + \rho B_2^{\top} B_2\right)\vec{d}^{(i+1)} = \beta B_3^{\top}\vec{g} + \rho B_2^{\top}\left(\vec{\psi} - \vec{v}^{(i)} - \vec{z}^{(i)}\right).$$

Since the coefficient matrix may be ill-conditioned, we compute the update using the TSVD pseudo-inverse, as presented in Definition 13. Thus,

$$\vec{d}_{\tau}^{(i+1)} = B_{\text{obstacle},\tau}^{\dagger}\,\vec{y}_{\text{obstacle}}^{(i)},$$

where

$$B_{\text{obstacle}} := B_1 + \beta B_3^{\top} B_3 + \rho B_2^{\top} B_2 \in \mathbb{R}^{M\times M}, \qquad \vec{y}_{\text{obstacle}}^{(i)} := \beta\, B_3^{\top}\vec{g} + \rho B_2^{\top}\left(\vec{\psi} - \vec{v}^{(i)} - \vec{z}^{(i)}\right) \in \mathbb{R}^{M}.$$

The TSVD pseudo-inverse $B_{\text{obstacle},\tau}^{\dagger}$ is constructed in the same manner for the linear elliptic PDEs, which was presented in Section 4.3.1.

Next, we define

$$\vec{t}^{(i)} := \vec{\psi} - B_2\,\vec{d}^{(i+1)} - \vec{z}^{(i)} \in \mathbb{R}^{m_\Omega}.$$

Then the $\vec{v}$-update is obtained componentwise by the proximal mapping of the ℓ^1-penalty on the positive part, namely,

$$v_j^{(i+1)} = \begin{cases} t_j^{(i)}, & t_j^{(i)} < 0, \\ 0, & 0 \le t_j^{(i)} \le \dfrac{\mu}{\rho}, \\ t_j^{(i)} - \dfrac{\mu}{\rho}, & t_j^{(i)} > \dfrac{\mu}{\rho}, \end{cases} \qquad j = 1, \ldots, m_\Omega.$$

The stopping criterion of the ADMM is monitored through the primal and dual residuals

$$\vec{r}^{(i+1)} := \vec{v}^{(i+1)} - \vec{\psi} + B_2\vec{w}^{(i+1)}, \qquad \vec{s}^{(i+1)} := \rho B_2^{\top}\left(\vec{v}^{(i+1)} - \vec{v}^{(i)}\right).$$

That is, the algorithm terminates when both residuals satisfy the prescribed relative tolerance criteria defined by:

$$\frac{\|\vec{r}^{(i+1)}\|_2}{\max\left(\|B_2\vec{w}^{(i+1)}\|_2, \|\vec{v}^{(i+1)}\|_2, \|\vec{\psi}\|_2\right)} \leq \epsilon_{\text{primal}}, \tag{4.22}$$

$$\frac{\|\vec{s}^{(i+1)}\|_2}{\|\rho B_2^\top \vec{z}^{(i+1)}\|_2} \leq \epsilon_{\text{dual}}. \tag{4.23}$$

This is implemented in Algorithm 4.

Algorithm 4: Stable ADMM-RFM solver for obstacle-type problems

Input: $\sigma(\cdot)$ (activation function); M (number of random features); $b_{\mathrm{RF}} > 0$ (variance scale); $T_{\mathrm{RF}} > 0$ (domain expansion factor); $\left\{\vec{x}_j^{(\Omega)}\right\}_{j=1}^{m_\Omega}$ (interior nodes); $\left\{\vec{x}_n^{(\partial\Omega)}\right\}_{n=1}^{m_{\partial\Omega}}$ (boundary nodes); $f(\cdot)$, $g(\cdot)$ (source/boundary data); $\psi(\cdot)$ (obstacle data); $\beta > 0$ (boundary penalty); $\mu > 0$ (obstacle weight); $\rho > 0$ (ADMM penalty); $\tau > 0$ (truncation tolerance); `maxIter` (maximum number of iterations); $\epsilon_{\mathrm{primal}}, \epsilon_{\mathrm{dual}}$ (relative stopping tolerances)

Output: $\vec{d}_\tau \in \mathbb{R}^M$, RFM approximation $\hat{u}_{\mathrm{RF},\tau} : \Omega \to \mathbb{R}$

Assemble operators and vectors:

Compute B_1, B_2, B_3 from the discrete variational formulation, as in Eqs. (2.40), (2.41), and (2.42)

Set $B_{\mathrm{obstacle}} \leftarrow B_1 + \beta B_3^\top B_3 + \rho B_2^\top B_2$

Form $\vec{f}, \vec{g}, \vec{\psi}$

Initialize $\vec{d}^{(0)}, \vec{v}^{(0)}, \vec{z}^{(0)}$

for $i = 0, \ldots,$ *maxIter* **do**

`// 1.` $\vec{d}$`-update (TSVD solve)`

$\vec{y}_{\mathrm{obstacle}}^{(i)} \leftarrow \beta B_3^\top \vec{g} + \rho B_2^\top (\vec{\psi} - \vec{v}^{(i)} - \vec{z}^{(i)})$

$\vec{d}_\tau^{(i+1)} \leftarrow B_{\mathrm{obstacle},\tau}^{\dagger} \vec{y}_{\mathrm{obstacle}}^{(i)}$

`// 2.` $\vec{v}$`-update`

$\vec{t}^{(i)} \leftarrow \vec{\psi} - B_2\, \vec{d}^{(i+1)} - \vec{z}^{(i)}$

for $j = 1, \ldots, m_\Omega$ **do**

if $t_j^{(i)} < 0$ **then** $v_j^{(i+1)} \leftarrow t_j^{(i)}$

else if $0 \le t_j^{(i)} \le \mu/\rho$ **then** $v_j^{(i+1)} \leftarrow 0$

else $v_j^{(i+1)} \leftarrow t_j^{(i)} - \mu/\rho$

end

`// 3. Dual update`

$\vec{z}^{(i+1)} \leftarrow \vec{z}^{(i)} + \vec{v}^{(i+1)} - \vec{\psi} + B_2 \vec{d}^{(i+1)}$

`// 4. Residuals and stopping`

$\vec{r}^{(i+1)} \leftarrow \vec{v}^{(i+1)} - \vec{\psi} + B_2 \vec{d}^{(i+1)}$

$\vec{s}^{(i+1)} \leftarrow \rho B_2^\top (\vec{v}^{(i+1)} - \vec{v}^{(i)})$

if $\dfrac{\|\vec{r}^{(i+1)}\|_2}{\max\{\|B_2\vec{d}^{(i+1)}\|_2, \|\vec{v}^{(i+1)}\|_2, \|\vec{\psi}\|_2\}} \le \epsilon_{\mathrm{primal}} \ \wedge \ \dfrac{\|\vec{s}^{(i+1)}\|_2}{\|\rho B_2^\top \vec{z}^{(i+1)}\|_2} \le \epsilon_{\mathrm{dual}}$ **then**

break

end

end

return $\vec{d}_\tau \in \mathbb{R}^M$, $\hat{u}_{\mathrm{RF},\tau}(\vec{x}) = \frac{1}{\sqrt{M}} \sum_{k=1}^{M} d_{(\tau),k}\, \sigma\left(\vec{\omega}_k^\top \vec{x} + b_k\right)$

4.4 Numerical Experiments: High-Dimensional PDEs

In this section, we consider high-dimensional linear PDEs and free-BVPs to validate Algorithms 3-4, as presented in Section 4.3. We use the PDEs from [121, 116, 113, 19] to test our proposed variational methods. In each experiment, we uniformly sample $m_\Omega = \zeta N$ interior points at random from Ω, where ζ is the oversampling ratio. For the boundary sampling, we select k points uniformly at random from each component of $\partial\Omega$, comprising $m_{\partial\Omega} = \zeta\, d\, k$ boundary points. The total number of collocation points is thus $m = m_\Omega + m_{\partial\Omega} = \zeta\,(M + d\,k)$. In addition, we generate an independent testing set consisting of m^{test} points to evaluate the accuracy of the PDEs approximation. To quantify the difference between the exact and predicted solutions on the testing set, we use the following error metric:

$$E(M) \coloneqq \frac{\|\hat{u}_{\text{RF}}(\vec{x}_j) - u(\vec{x}_j)\|_2}{\|u(\vec{x}_j)\|_2}, \quad j = 1, \ldots, m^{\text{test}} \tag{4.24}$$

All parameters utilized in the numerical experiments are compiled and presented in Table 4.1. Unless otherwise stated, the same parameter configuration is used in all experimental cases. We specifically utilize the hyperbolic tangent activation function for the RFM, except in the comparison studies specified in Section 4.6. To compare with mesh-free approximation techniques, we employ machine learning approaches, specifically PINNs [88, 89, 87, 90] and the deep Ritz method (DRM) [29, 87]. In the PINNs formulation of [88], the objective function is imposed in strong form by minimizing a loss function constructed from the mean squared residuals of the PDEs and the boundary conditions, both evaluated at interior and boundary sampling points. The DRM approach in [87] is developed for PDEs that admits a variational formulation, with boundary conditions enforced through penalty contributions. In DRM, boundary constraints are incorporated by adding a penalty term to the energy functional. PINNs uses a fully-connected neural network with a hyperbolic tangent activation, while DRM adopts a ResNet architecture with $\sigma_{\text{activation}}(\vec{z}) = \max\{(\vec{z})^3, 0\}$. Boundary penalty weight β is set to 1 for PINNs and 10^4 for DRM. To ensure reproducibility and fair comparison as in [121], all networks consist of three hidden layers with 40 neurons each hidden layer. We set the learning rate of 0.001 for the Adam optimizer as suggested in PINNs[1] and DRM[2].

[1]Package of PINNs is implemented in https://github.com/maziarraissi/PINNs
[2]Package of DRM is obtained from https://github.com/junbinhuang/DeepRitz

Table 4.1: Hyper-parameter settings used in the numerical experiments.

Parameter	**Description**	**Value**
$\sigma(\cdot)$	random feature activation functional	$\tanh(\cdot)$
M	total number of random features	`problem-dependent`
b_{RF}	variance scale parameter in random feature matrix	$b_{\mathrm{RF}} = 1.75$
T	expansion factor for domain in bias vector of random features model	`problem-dependent`
ζ	oversampling ratio	$\zeta = 2$
k	number of sampled points per boundary component	`problem-dependent`
d	dimensionality of boundary value problem	`problem-dependent`
Ω	interior domain	`problem-dependent`
$\partial\Omega$	boundary domain	`problem-dependent`
m	total number of collocation points	$m = m_\Omega + m_{\partial\Omega}$
m_Ω	interior collocation points	$m_\Omega = \zeta\, M$
$m_{\partial\Omega}$	boundary collocation points	$m_{\partial\Omega} = \zeta\, d\, k$
m^{test}	total number of testing points	$m^{\mathrm{test}} = 2000$
β	boundary penalty weight	`problem-dependent`
`maxIter`	maximum number of iterations in ADMM	$50,0000$
μ	ℓ^1 penalty parameter weight	$\mu = 10\, M^2$
ρ	penalty parameter for the augmented Lagrangian in ADMM	$\rho = 10.3$
ϵ_{stop}	relative stopping tolerance	$\epsilon_{stop} = 10^{-4}$
τ	truncated threshold tolerance in TSVD	$\tau = 10^{-15}$

4.4.1 Numerical experiments for high-dimensional elliptic boundary value problem

Weak form versus strong form for Poisson boundary value problem

In the first numerical experiment of Section 4.4.1, we show that Algorithm 3, based on the variational formulation of PDEs, can be advantageous for solving Poisson BVPs, which does not have a strong solution. In particular, we consider the following Poisson equation with the Dirichlet boundary condition as in [121], which is given by:

$$\begin{cases} -\Delta u(\vec{x}) = f(\vec{x}), & \forall \vec{x} = (x_1, x_2)^\top \in \Omega, \\ u = g, & \text{on } \partial\Omega, \end{cases} \tag{4.25}$$

where $\Omega = (0,1)^2 \subseteq \mathbb{R}^2$ and the source function $f \equiv -2$ in Ω. The boundary condition is followed by $g(x_1, 0) = g(x_1, 1) = x_1^2$ for $0 \leq x_1 \leq \frac{1}{2}$, $g(x_1, 0) = g(x_1, 1) = (x_1 - 1)^2$ for $\frac{1}{2} \leq x_1 \leq 1$, and $g(0, x_2) = g(1, x_2) = 0$ for $0 \leq x_2 \leq 1$ on $\partial\Omega$. This problem, however, does not admit a strong (classical) solution, but has only a unique solution, which is given by:

$$u_{\text{exact}}(x_1, x_2) = \begin{cases} x_1^2, & 0 \leq x_1 \leq \frac{1}{2}, \\ (x_1 - 1)^2, & \frac{1}{2} \leq x_1 \leq 1. \end{cases} \tag{4.26}$$

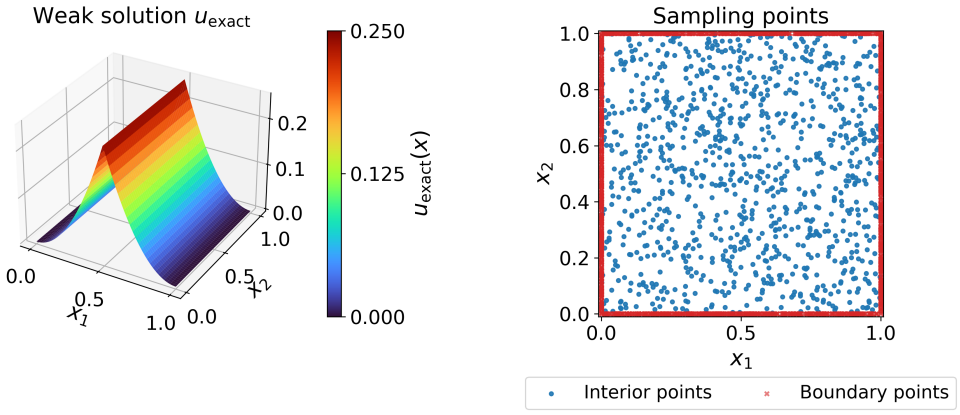
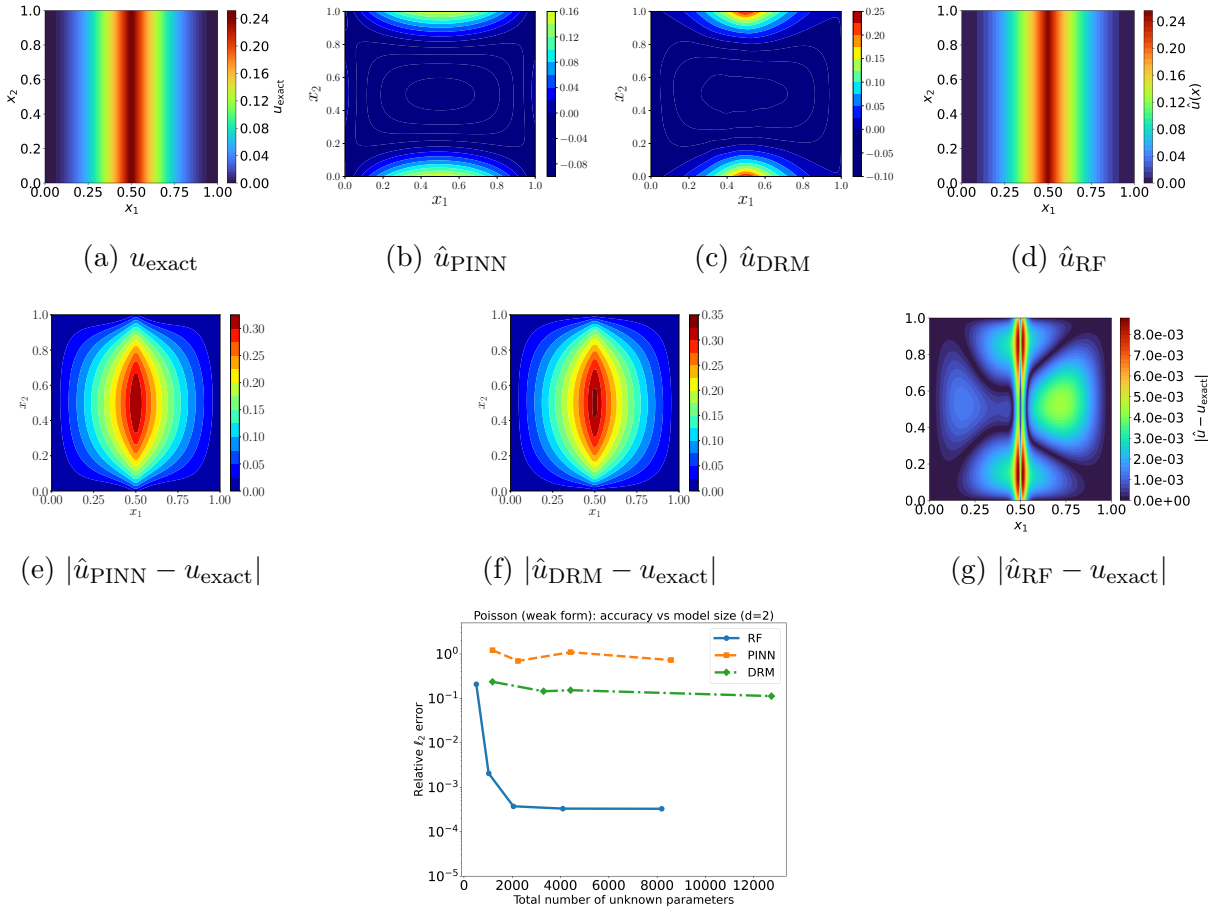


Figure 4.1: **Poisson problem in Eq. (4.25):** Weak solution u_{exact} in Eq.(4.26) on the domain $\overline{\Omega} = [0,1]^2$ (left) and the corresponding sampling points used in the numerical experiments (right). Interior collocation points are shown in blue, while boundary points are indicated in red.



(a) u_{exact} (b) $\hat{u}_{\text{PINN}}$ (c) $\hat{u}_{\text{DRM}}$ (d) $\hat{u}_{\text{RF}}$

(e) $|\hat{u}_{\text{PINN}} - u_{\text{exact}}|$ (f) $|\hat{u}_{\text{DRM}} - u_{\text{exact}}|$ (g) $|\hat{u}_{\text{RF}} - u_{\text{exact}}|$

(h) Relative error vs model size

Figure 4.2: **Result of Poisson's Equation in Experiment** (4.25). (a) The weak solution u_{exact}; (b) $\hat{u}_{\text{PINN}}$ obtained by PINN based on the strong form; (c) $\hat{u}_{\text{DRM}}$ obtained by DRM; (d) $\hat{u}_{\text{RF}}$ obtained by the RF approximation based on the weak form using $N = 2048$ total number of features; (f)–(h) pointwise absolute errors between approximation and exact solution $|\hat{u} - u_{\text{exact}}|$; (e) relative ℓ_2 error versus number of unknown parameters of RF, PINN, and DRM methods.

In this experiment, we use the following resolution parameters $M = 2048, \zeta = 2, k = 100$ to approximate the solution in Eq. (4.26) in Figure 4.2(d)(h). For other hyperparameters, we keep the expansion domain factor $T_{\text{RF}} = 2.2$, the boundary penalty weight $\beta = 3 \times 10^4$, the truncation threshold $\tau = 10^{-15}$, and the variance scale in the RFM model, which is $b_{\text{RF}} = 1.75$. The choices of parameters are discussed in detail later in Section 4.5. As shown in Figure 4.2, the RFM approximation $\hat{u}_{\text{RF}}$ closely aligns with the exact solution u_{exact} when visualized on the (x_1, x_2)-slice for $d = 5$. As shown in Figure 4.2(b)(c), contours of the PINNs and DRM approximate solutions do not converge to the exact solution, respectively. In Figure 4.2(g), the absolute difference between the RFM approximation and the exact solution is relatively small and demonstrates the accuracy of $E(M) = 3.458 \times 10^{-4}$. While the absolute difference for the PINNs and DRM approximate solutions in Figure 4.2(e)(f) show that the errors are distributed large and do not converge to the exact solution. In Figure 4.2(e), we compare RFM, PINNs, and DRM methods on the same plot. For PINNs and DRM we vary layers from 2 to 3, using 32 or 64 neurons in each. This demonstrates that the RFM outperforms other mesh-free methods while using significantly fewer trainable parameters. Due to the ResNet architecture, the trainable parameters of the DRM method are increasing greatly, but its relative error plateaus at 10^{-1} level. A similar behavior is observed with the PINNs method; it does not converge as the relative error stays around the 10^{-1} level as well.

High-dimensional linear elliptic PDEs - Poisson equation

In this numerical experiment, we demonstrate that Algorithm 3, which is grounded in variational principles, can offer advantages for high-dimensional linear elliptic PDEs that admit an exact solution. The following problem is taken from [121]. We consider the Poisson BVPs with the following boundary conditions, with $d \geq 5$ given by:

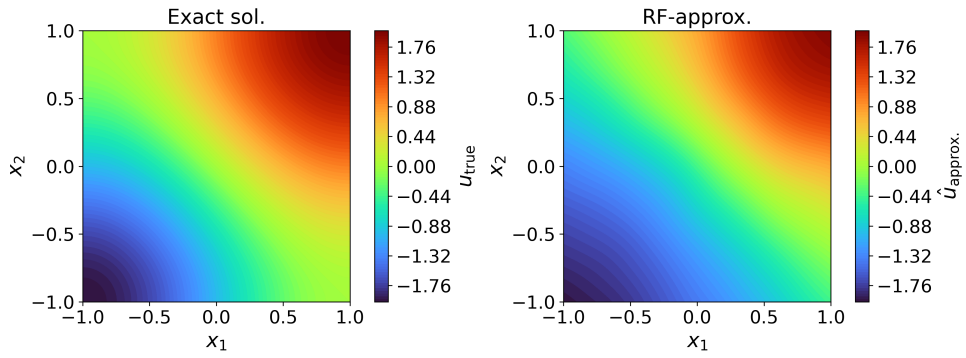
$$\begin{cases} -\Delta u = \frac{\pi^2}{4} \sum_{i=1}^{d} \sin(\frac{\pi}{2} x_i), & \text{in } \Omega = (-1, 1)^d, \\ u = \sum_{i=1}^{d} \sin(\frac{\pi}{2} x_i), & \text{on } \partial\Omega. \end{cases} \tag{4.27}$$

The BVP in Eq.(4.27) has an exact solution, which is given by:

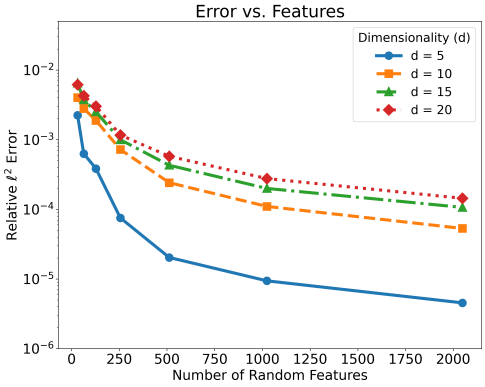
$$u_{\text{exact}}(\vec{x}) = \sum_{i=1}^{d} \sin\left(\frac{\pi}{2} x_i\right), \quad \text{in} \quad \overline{\Omega} = [-1, 1]^d. \tag{4.28}$$

Figure 4.3 shows the contours of the exact solution in Eq. (4.28) and the RFM approximation in the first two dimensions for $d = 5$ and $M = 2048, \zeta = 2, k = 500, T_{\text{RF}} =$

$2.2, b_{\mathrm{RF}} = 1.75$. For details of the choices of hyperparameters, we discuss them later in Section 4.5. The RFM approximate solution aligns well with the exact solution, with the error $E(M) = 2.628 \times 10^{-6}$. As shown in Figure 4.3(b), it demonstrates the convergence behavior of the RFM method as the dimensions increase. The result illustrates that the RFM method converges with a relative error level between 10^{-6} and 10^{-3} for dimensions up to $d = 20$. It further demonstrates that the RFM approach is influenced by a trade-off between accuracy and the number of features. In particular, for $d = 20$, one needs $M = 20,000$ features to attain a similar level of accuracy.. However, we require more memory storage to train the large learnable coefficients. Figure 4.4 shows the comparison of the competing methods with the RFM approaches. According to Figure 4.4(a)(b)(c), the absolute error between the RFM method and the exact solution remains relatively small when compared with the PINNs and DRM approaches. Figure 4.4, panels (d) and (e), show the minimization of residual loss and variational loss functions, respectively, for the PINNs and DRM methods over $20,000$ iterations. Figure 4.4(f) demonstrates that the RFM method converges to an error level of around 10^{-6} as the number of trainable parameters increases. While the PINNs method remains at an error level of around 10^{-3}, the DRM approaches attain a level of 10^{-2} despite the increasing number of trainable parameters in the ResNet architecture.

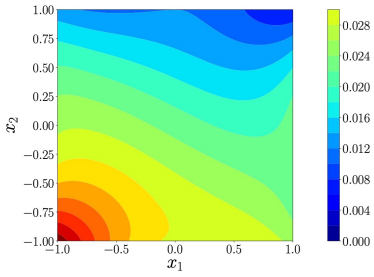


(a) Exact solution in Eq. (4.30) and random feature approximation.

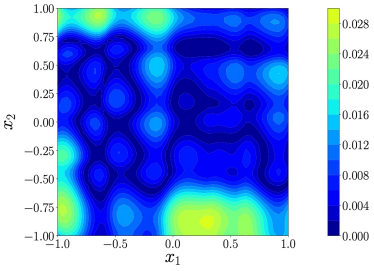


(b) Convergence behavior of random feature methods across dimensions.

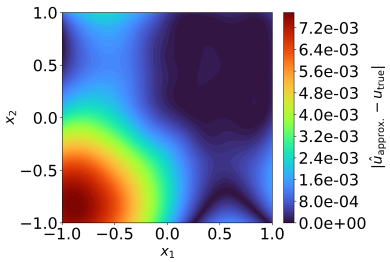
Figure 4.3: **High-dimensional PDE in Experiment** (4.27). Top: comparison between the approximation and the exact solution with the resolution parameters $M = 4096, \zeta = 2, k = 500$. We plot the contour of the first and second dimensions of $[-1, 1]^5$. Bottom: convergence behavior for increasing spatial dimensions. Results are shown for $d = 5, 10, 15, 20$.



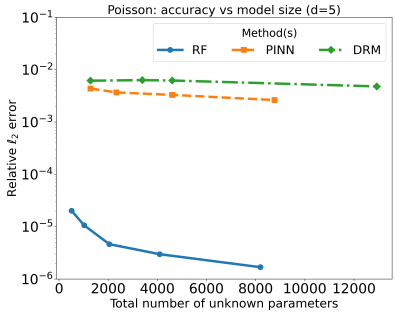
(a) $|\hat{u}_{PINN} - u_{exact}|$



(b) $|\hat{u}_{DRM} - u_{exact}|$



(c) $|\hat{u}_{RF} - u_{exact}|$



(d) Relative error vs model size

Figure 4.4: **High-dimensional PDE in Experiment** (4.27): Comparison of numerical solutions obtained using PINN, DRM, and the proposed RF method over the domain $[-1, 1]^5$. For RF method, we use the following parameters $M = 4096, \zeta = 2, k = 500, T_{\text{RF}} = 2.2, \beta = 10^6$. (a)–(c) Pointwise absolute error $|\hat{u} - u_{\text{exact}}|$ for PINN (*left*), DRM (*middle*), and RF (*right*). (d) Relative ℓ^2 error versus the size of model, illustrating the convergence behavior of the three methods. We use $M = \{512, 1024, 2048, 4096, 8192\}$ for RF model, for PINN and DRM we use the architecture with depth $= \{2, 3\}$, width $= \{32, 64\}$.

High-dimensional linear elliptic PDEs - Darcy's Flow problem

We now consider a linear elliptic PDE with Dirichlet boundary conditions on an irregular, nonconvex domain with dimensionality $d = 2, 5, 10$, as described in [121], which is given by.

$$\begin{cases} -\nabla \cdot (a(\vec{x})\, \nabla u) = f(\vec{x}), & \text{in} \quad \Omega = (-1,1)^d \setminus [0,1)^d \\ u(\vec{s}) = g(\vec{s}), & \forall \vec{s} \in \partial\Omega, \end{cases} \tag{4.29}$$

where $a(\vec{x}) = 1 + \|\vec{x}\|_2^2$, and $f(\vec{x}) = \frac{\pi^2}{2}\left(1 + \|\vec{x}\|^2\right) \sin(\tilde{x}_1) \cos(\tilde{x}_2) + \pi x_2 \sin(\tilde{x}_1) \sin(\tilde{x}_2) - \pi x_1 \cos(\tilde{x}_1) \cos(\tilde{x}_2)$ in Ω and $g\left(\vec{x}\right) = \sin(\tilde{x}_1) \cos(\tilde{x}_2)$ on $\partial\Omega$ with $\tilde{x}_i := \frac{\pi}{2}\, x_i$ for $i = 1, 2$. The BVP in Eq. (4.29) admits an exact solution in Ω, which is as follows:

$$u_{\text{exact}}\left(\vec{x}\right) = \sin(\tilde{x}_1)\ \cos(\tilde{x}_2). \tag{4.30}$$

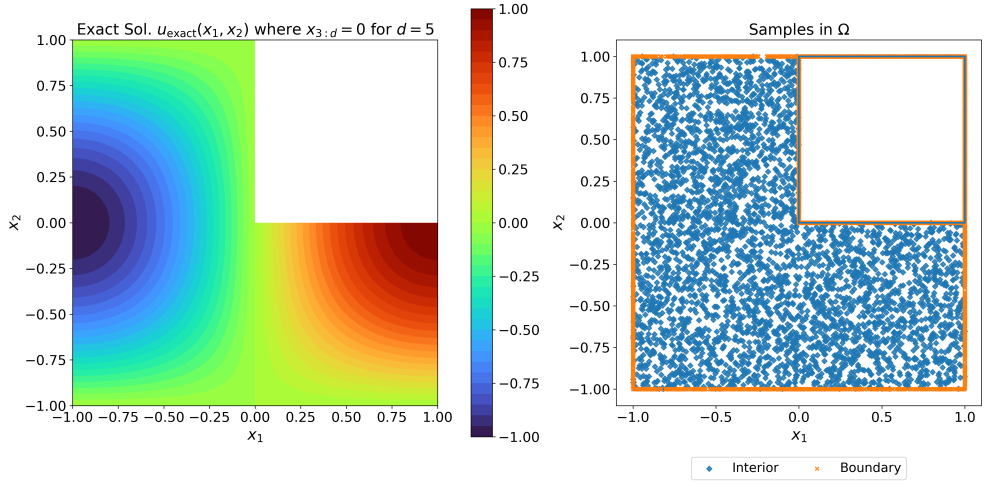
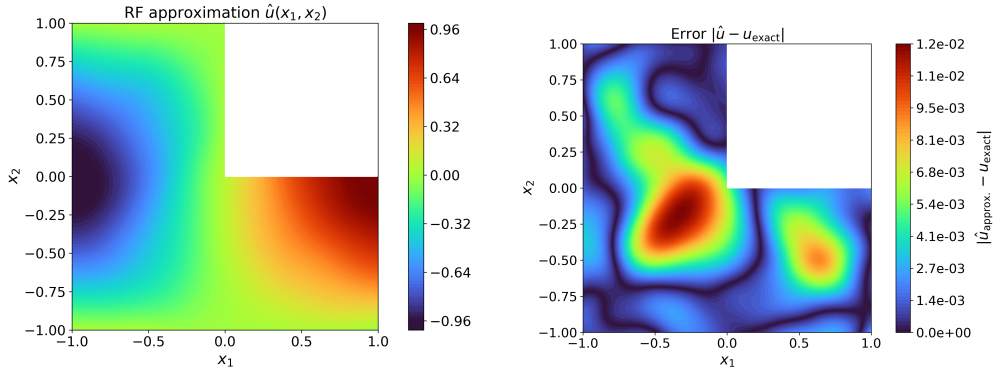
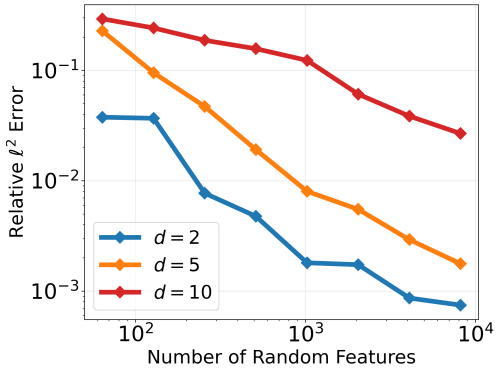


Figure 4.5: **Darcy's Flow Problem in Experiment** (4.29)**:** *Left:* Exact solution in the first two dimensions for $d = 5$ in the Eq. (4.29). *Right:* Sampled interior and boundary collocation points in the domain Ω with a square exclusion.



(a) RFM approximation (b) Error distribution



(c) Convergence of RFM methods across dimensions

Figure 4.6: **Darcy's Flow Problem in Experiment** (4.29)**:** *Top left:* RF approximation $\hat{u}_{\text{RF}}$ in the first two dimensions for $d = 5$ using parameters $M = 4096, \zeta = 2, k = 500$ random features. *Top right:* Pointwise absolute error $|\hat{u} - u_{\text{exact}}|$ on the same slice. *Bottom:* Relative ℓ^2 error versus the number of random features M for $d = 2, 5, 10$, showing that larger M is required as the dimension increases to achieve better accuracy.

Figure 4.5 shows the contour of the exact solution for Eq. (4.29) and the sampled interior and boundary points from the irregular nonconvex domain. We select $M = 4096, \zeta = 2, k = 500, T_{\text{RF}} = 1.0, \beta = 10^6, \tau = 10^{-15}$ and discuss this choice in Section 4.5. Figure 4.6 shows the RFM approximation, its absolute error distribution, and the convergence of the proposed method for different dimensions. The RFM approximation aligns well with the exact solution when $d = 5$, as shown in Figure 4.6(a) for the first two dimensions. The absolute error in Figure 4.6(b) is relatively small, and the relative error $E(M) = 2.581 \times 10^{-3}$ for $d = 5$. Figure 4.6(c) shows the convergence behavior of the RFM methods

as the dimension d increases up to 10. It demonstrates that larger M is required as the dimension increases to achieve the desired accuracy. For example, when $d = 2$ the RFM method needs $M = 10^3$ to achieve a relative error level of around 10^{-3}, but when $d = 5$ it is necessary to choose M between 5×10^3 and 10^4 to attain the similar error level.

4.4.2 Numerical experiment for obstacle problems

In this experiment, we consider a two-dimensional obstacle problem defined on the domain $\overline{\Omega} = [0, 1]^2$ such that

$$\begin{cases} u(x_1, x_2) \geq \psi_1(x_1, x_2), & \text{in } \Omega, \\ -\dfrac{\partial^2 u}{\partial x_1^2} - \dfrac{\partial^2 u}{\partial x_2^2} \geq 0, & \text{in } \Omega, \\ \big(u(x_1, x_2) - \psi_1(x_1, x_2)\big) \left(-\dfrac{\partial^2 u}{\partial x_1^2} - \dfrac{\partial^2 u}{\partial x_2^2}\right) = 0, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega. \end{cases} \tag{4.31}$$

We let the distance to center $(\frac{1}{2}, \frac{1}{2})$ be such that $r(x_1, x_2) = \sqrt{\left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2}$ for any $(x_1, x_2) \in \overline{\Omega}$. The radially symmetric obstacle function is given by:

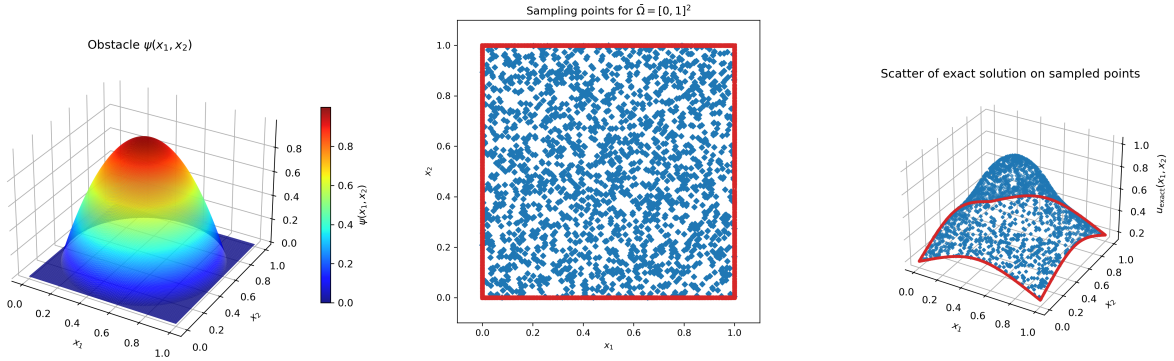
$$\psi_1(x_1, x_2) = \begin{cases} 1 - \frac{r(x_1, x_2)^2}{r_c^2}, & \text{for } x_1^2 + x_2^2 \leq r_c^2, \\ 0, & \text{otherwise}, \end{cases} \tag{4.32}$$

where the inner radius is $r_c := \frac{1}{2}$. This obstacle-related equation has been widely used in [107, 68, 31] to demonstrate its accuracy. We also assume that the BVP satisfies the Dirichlet boundary condition, where $g(x_1, x_2) = u_{\text{exact}}(x_1, x_2)$ on $\partial\Omega$. The obstacle problem corresponding to the governing equation in Eq. (4.31), subject to the radially symmetric function in Eq. (4.32), admits an exact solution, which is given by:

$$u_{\text{exact}}(x_1, x_2) = \begin{cases} \left(1 - \frac{r(x_1, x_2)^2}{r_c^2}\right), & r(x_1, x_2) \leq r_* \\ 1 - \frac{r_*^2}{r_c^2}\left(1 + 2\log\left(\frac{r(x_1, x_2)}{r_*}\right)\right), & r_* < r(x_1, x_2) \end{cases} \tag{4.33}$$

where the contact radius $r_* \approx 0.260197$ is determined by solving the following equation

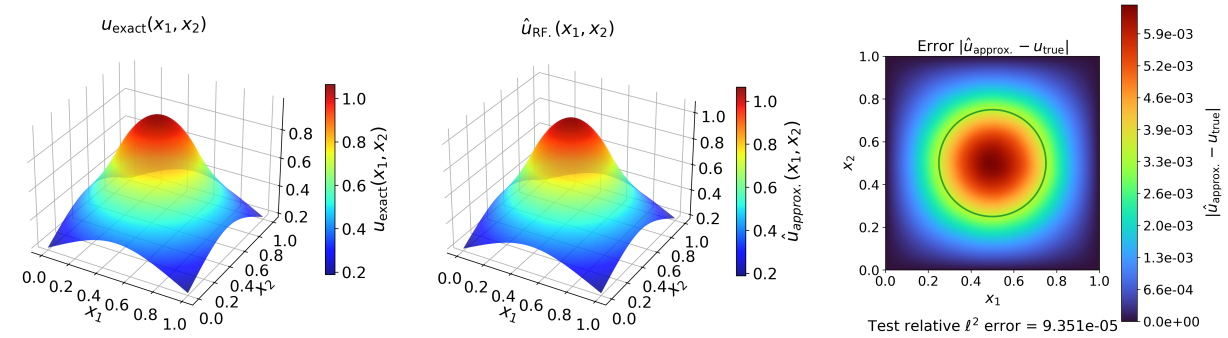
$$\frac{r_*^2}{r_c^2}\left(1 + 2\log\left(\frac{1}{r_*}\right)\right) = 1, \quad \text{where } 0 < r_* < r_c. \tag{4.34}$$



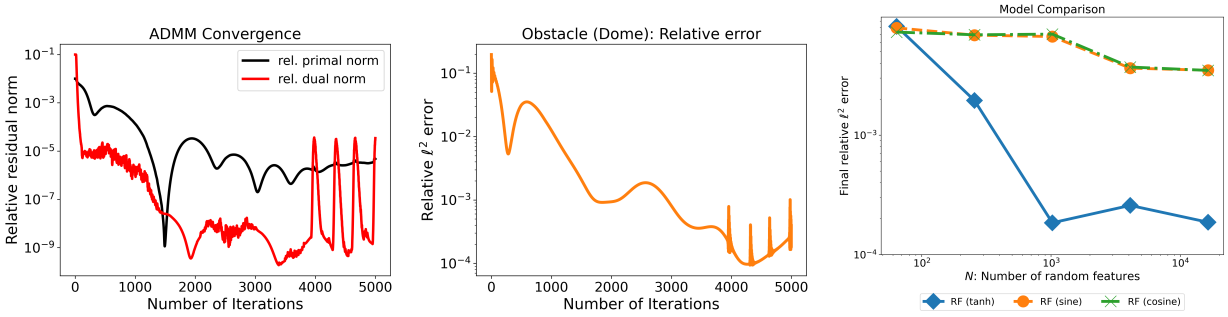
(a) The obstacle function (b) Sampled collocation points (c) Scattered points of u_{exact}

Figure 4.7: **Two-dimensional obstacle problem in Experiment** (4.31): Obstacle function $\psi(x_1, x_2)$ defined in Eq. (4.32). Interior and boundary collocation points sampled in the domain $\overline{\Omega} = [0, 1]^2$. Scattered values of the exact solution $u_{\text{exact}}(x_1, x_2)$ in Eq. (4.33) evaluated at the sampled points.

Figure 4.7 shows the obstacle function defined in Eq. (4.32), the sampling points in the closed convex domain $[0, 1]^2$, and the exact solution in Eq. (4.33). We select $T_{\text{RF}} = 0.5, \beta = 10^4, \tau = 10^{-15}, M = 2048, k = 100, \zeta = 2$, and we discuss the choices later in Section 4.5. Figure 4.8 shows the random feature approximation, absolute error distribution, ADMM convergence, and relative ℓ^2 error, along with a comparison to sine and cosine features. As shown in Figure 4.8(a)(b)(c), the RFM model approximates the exact solution well, and the error distribution remains relatively small in panel (c). The error metric value $E(M) = 9.351 \times 10^{-5}$ indicates the accuracy of the random feature approximation. Figure 4.8(d) shows that the relative norm of primal and dual residuals meets the ADMM tolerance value $\epsilon_{tol.} = 10^{-4}$. The relative error, as shown in Figure 4.8(e), indicates that the RFM attains an error level of around 10^{-4} after 5000 iterations. Lastly, the RFM with hyperbolic tangent activation achieves significantly better accuracy than the sine and cosine features, as shown in Figure 4.8(f).



(a) Exact solution (b) RFM approximation (c) Absolute error



(d) Relative primal and dual residuals (e) Relative error versus iteration (f) Relative error versus number of features

Figure 4.8: **Two-dimensional obstacle problem in Experiment (4.31).** *(a)* Exact solution $u_{\text{exact}}(x_1, x_2)$. *(b)* Random feature approximation $\hat{u}_{\text{RF}}(x_1, x_2)$ with $M = 2048, \zeta = 2, k = 100, T_{\text{RF}} = 0.5, \beta = 10^4$. *(c)* Absolute error $|\hat{u}_{\text{RF}} - u_{\text{exact}}|$ on $\overline{\Omega}$, with test relative ℓ^2 error $E(M) = 9.351 \times 10^{-5}$. *(d)* Relative primal and dual residuals during the ADMM iterations. *(e)* Relative ℓ^2 error versus iteration. *(f)* Relative ℓ^2 error versus number of random features, comparison with other $\sigma(\cdot)$ function, sine and cosine.

4.5 Effect of the Variance Scale b_{RF}, Expansion Domain Factor T_{RF}, and Boundary Penalty Weight β on Random Feature Approximation

In this section, we examine the selection of hyperparameters in the random feature approach. Specifically, we provide justification for the chosen values of the expansion domain factor T_{RF}, the variance scale b_{RF}, and the boundary penalty weight β in the numerical experiments 4.4.1-4.4.2. The following Sections 4.5.2-4.5.3 examine each parameter, respectively.

4.5.1 Sensitivity to the expansion domain factor T_{RF}

In this subsection, we consider the choices of the expansion domain factor T_{RF} for each of the numerical experiments. Figure 4.9 shows the effect of the expansion domain factor T_{RF}. In particular, Figure 4.9(a) indicates that T_{RF} needs to be set at $T_{\mathrm{RF}} = 0.1$ in the experiment 4.4.1. Meanwhile, in high-dimensional Poisson equations, we set $T_{\mathrm{RF}} = 2.2$, as illustrated in Figure 4.9(b). The physical domain $\overline{\Omega}$ differs in each experiment, and the purpose of the expansion domain $\overline{\Omega}_{T,d}$ is to select the random features for the bias term outside of the physical domain, along with the dimensionality-scaling factor $T_{\mathrm{RF}}\sqrt{d}$. Similarly, Figure 4.9(c)(d) shows that the choices of T_{RF} are $1.0, 0.5$, respectively, for Darcy flow and obstacle problems. Other choices of T_{RF} in Figure 4.9 still show an effect on the accuracy; however, they do not show monotone convergence as the number of features increases.

4.5.2 Sensitivity to the variance scale b_{RF}

We examine the choices of the variance scale b_{RF}, which is crucial to the random matrix W in the RFM model. We select the other parameters for each corresponding experiment in Section 4.4.1. Figure 4.10 shows the effect of the variance scaling b_{RF}. Generally, all panels in Figure 4.10 indicate that the variance scale b_{RF} needs to be selected in the domain $[1, 2]$ as long as the number of features is sufficiently large enough. For instance, when $M = 512, 1024$ is not large enough, the relative error does not converge and causes instability regardless of the choices of the variance scale, as shown in Figure 4.10. Hence, we choose $b_{\mathrm{RF}} = 1.75$ that can work in all experiments as well as when the number of features M is sufficiently large.

4.5.3 Sensitivity to the boundary penalty weight β

Lastly, we study the choices of the boundary penalty weight β, which is crucial for the integration between the random feature approximation and the Dirichlet boundary conditions. Figure 4.11 shows the sensitivity of the random feature method with respect to the choice of boundary penalty parameter β. In all panels of Figure 4.11, there is an indication to choose β at a sufficiently large value in order to obtain monotone convergence. Figure 4.11 demonstrates that when M is large enough, β needs to be chosen above around 10^2, and then the relative error plateaus at a level of 10^{-4}. However, this is not true for $M = 512, 1024$ because the number of features is not enough for the random feature approximation to converge. A similar observation in Figure 4.11(b) indicates that β needs to be set above 10^3 to obtain the desired accuracy at error level 10^{-5}. For the Darcy flow problem, panel (c) in Figure 4.11 shows that as long as M is large enough, then the choice of β above 10^2 makes the RFM converge monotonically. In the obstacle problem, Figure 4.11(d) indicates that the suitable range for β is $[10^2, 10^4]$; otherwise, the relative error level plateaus at 10^{-2}. Therefore, we select $\beta = 10^6$ for Poisson and Darcy flow problems, respectively, while β is set at 10^4 in the obstacle problem.

4.6 Comparison with Kernel Methods

This section provides a direct comparison between the variational RFM developed in this chapter and the RBF-based variational approximation introduced in Chapter 3. Both approaches can be interpreted as the KF methods, since the approximate solution is represented as a finite linear combination of nonlinear basis functions. However, both methods differ substantially in the way the features are constructed. In the RBF approach, the basis functions are sampled at center locations in the expansion domain, and the approximation is built using a structured kernel expansion. In contrast, the RFM uses randomly sampled and fixed nonlinear features, and only the output-layer coefficients are optimized.

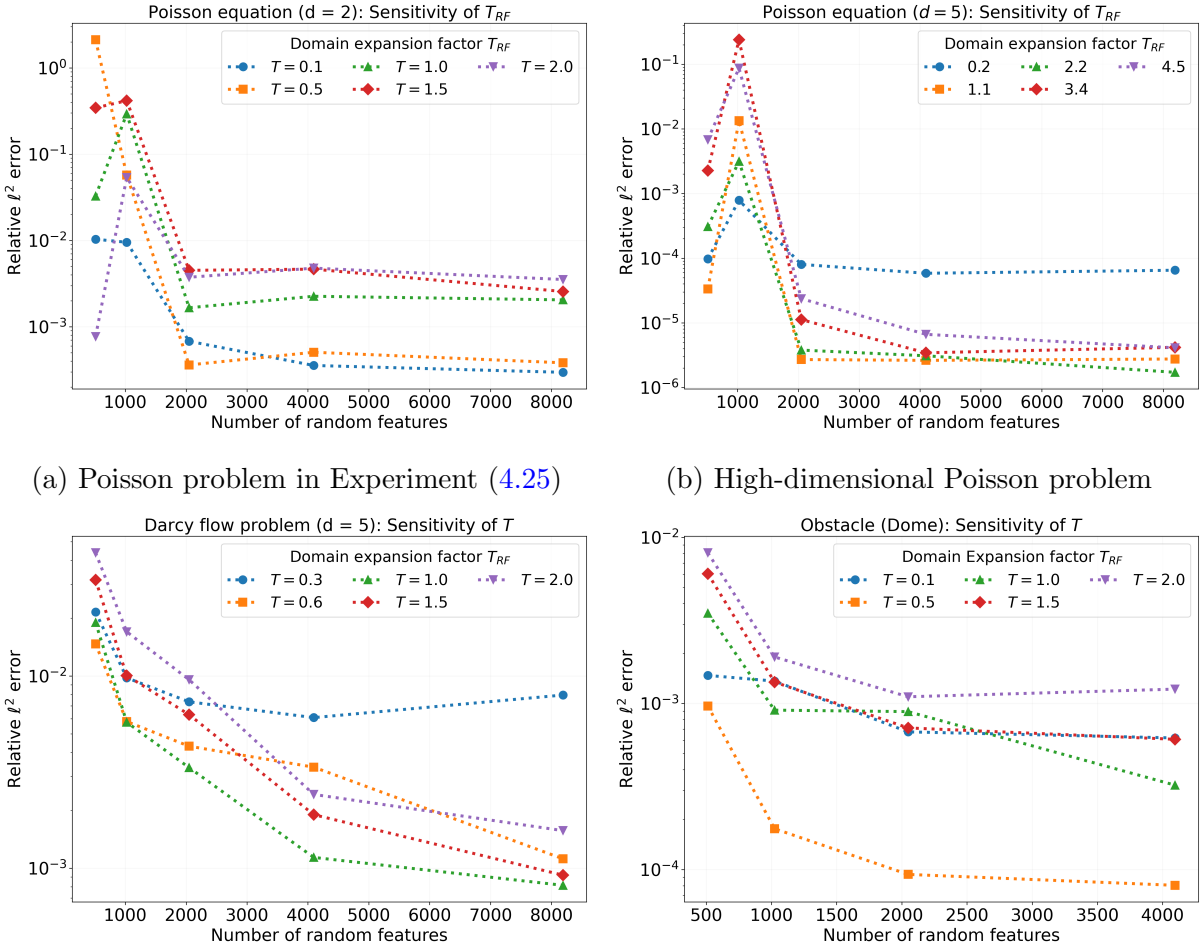
Figure 4.12 compares the relative ℓ^2 error of the RBF and RFM approximations as a function of the total number of parameters. The comparison is shown for four BVPs considered in this thesis: the one-dimensional Poisson equation in Eq. (3.32), the one-dimensional reaction-diffusion equation in Eq. (3.4), the two-dimensional Darcy flow problem in Eq. (4.29), and the two-dimensional dome obstacle problem presented in Eq. (2.10). These examples include both linear elliptic PDEs and a free-BVP, and therefore provide a comparison for evaluating the accuracy of both kernel-feature models. For the one-dimensional Poisson problem, both methods converge as the number of parameters

increases in Figure 4.12(a). The RBF approximation reaches very high accuracy, with errors close to machine precision, while the RFM also improves but shows less monotonic convergence and eventually stagnates at a higher error level. This difference reflects the structured nature of the RBF basis, whose centers are directly associated with the spatial domain, compared to the fixed randomized features used in the RFM.

A similar behavior is observed for the reaction-diffusion problem in Figure 4.12(b). The RBF method produces a monotonic convergence in the relative ℓ^2 error and achieves a lower error than the RFM. Although the RFM converges over the tested range, its error curve contains visible oscillations, indicating that randomly sampled features may require more parameters to match the efficiency of a well-tuned structured kernel basis for smooth low-dimensional problems.

For the two-dimensional Darcy flow problem, the difference becomes more pronounced in Figure 4.12(c). The RBF approximation exhibits a sharp error reduction after a sufficient number of basis functions is used and continues to improve as the parameter count increases. The RFM also converges, but more slowly and with a higher plateau. This suggests that the structured kernel basis remains effective for resolving spatial variation in two dimensions, while the RFM provides a simpler and more scalable alternative. The significant difference is observed in the dome obstacle problem in Figure 4.12(d). The RBF approximation achieves substantially smaller errors because the obstacle problem contains a free boundary and a contact set, whose localized geometry is accurately resolved. The RBF centers provide a more accurate representation of these features, whereas the RFM relies on fixed random nonlinear features; therefore, it is necessary to require more features to accurately approximate the free-boundary region.

In summary, Figure 4.12 shows that the RBF method is more accurate for one- and two-dimensional problems, especially for the obstacle problem. However, the RFM still exhibits convergence as the number of random features increases and continues to serve as an alternative approach for high-dimensional problems., where dense RBF systems become costly to assemble, store, and solve.



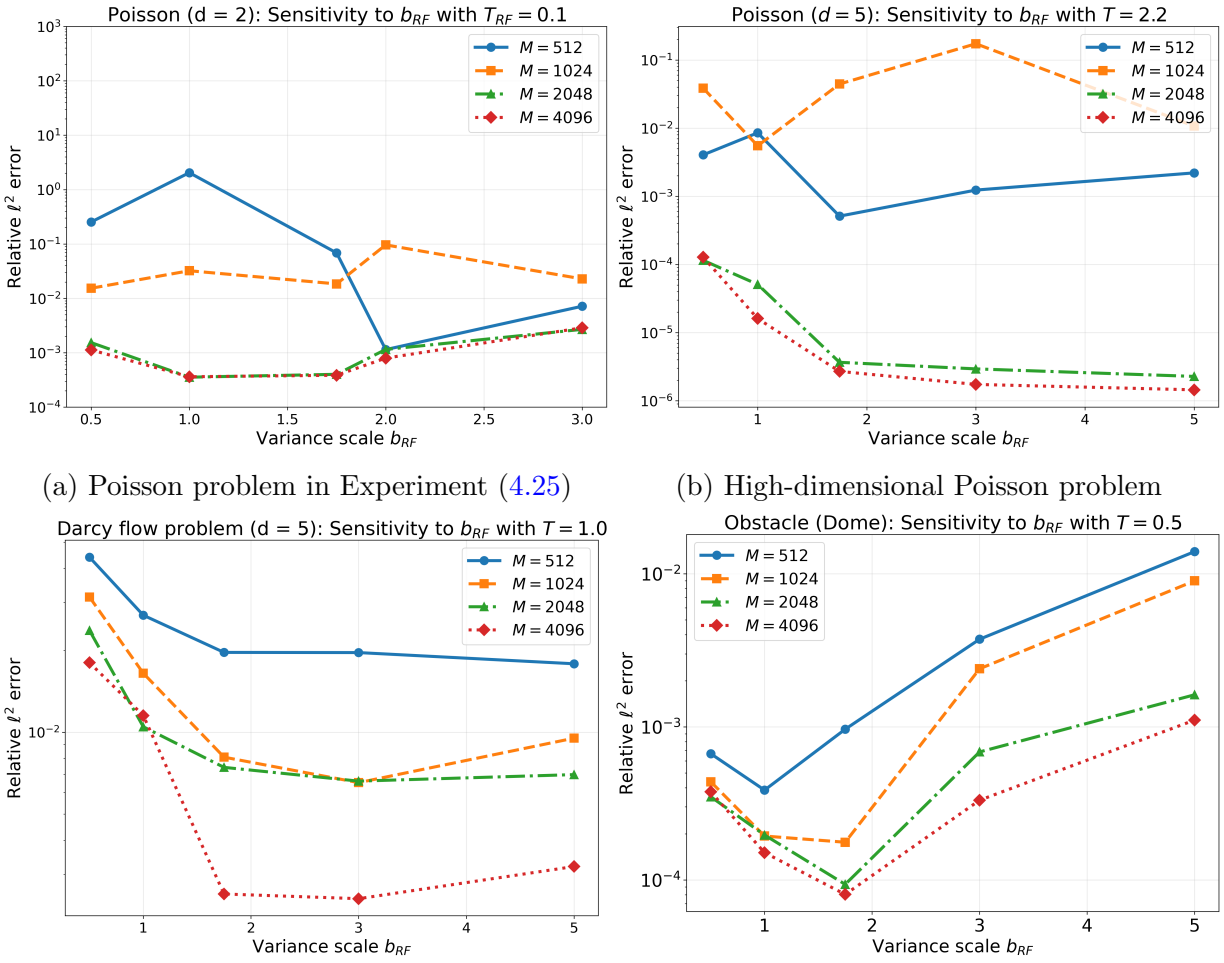
(a) Poisson problem in Experiment (4.25)

(b) High-dimensional Poisson problem

(c) Darcy Flow in Experiment (4.29)

(d) Obstacle problem in Experiment (4.31)

Figure 4.9: **Sensitivity of the expansion domain factor** T_{RF}: Relative ℓ^2 error versus the number of random features for (a) the two-dimensional Poisson problem, (b) the high-dimensional Poisson problem, (c) the Darcy flow problem, and (d) the dome obstacle problem. The results indicate that the expansion factor T_{RF} significantly affects the accuracy of the variational random feature approximation.



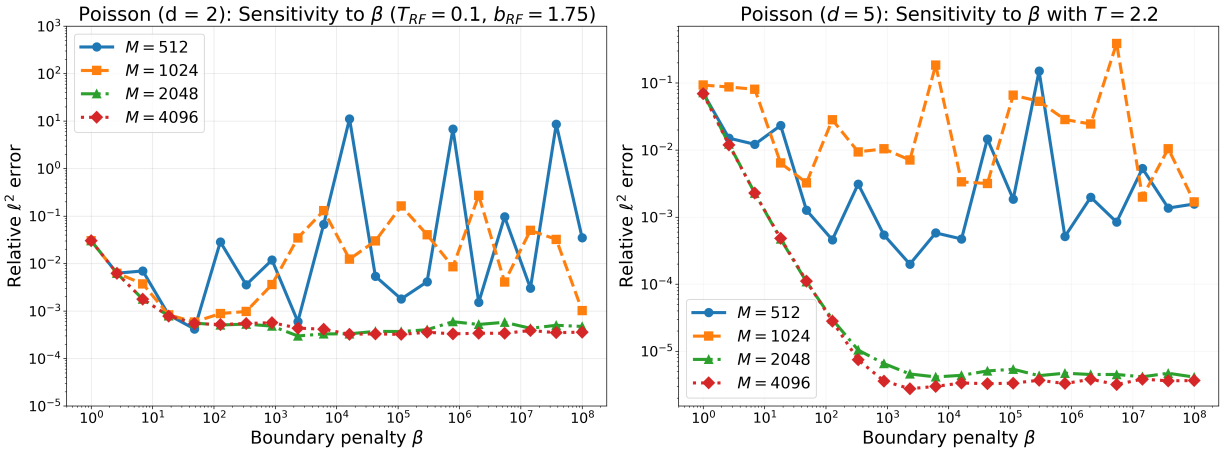
(a) Poisson problem in Experiment (4.25)

(b) High-dimensional Poisson problem

(c) Darcy Flow in Experiment (4.29)

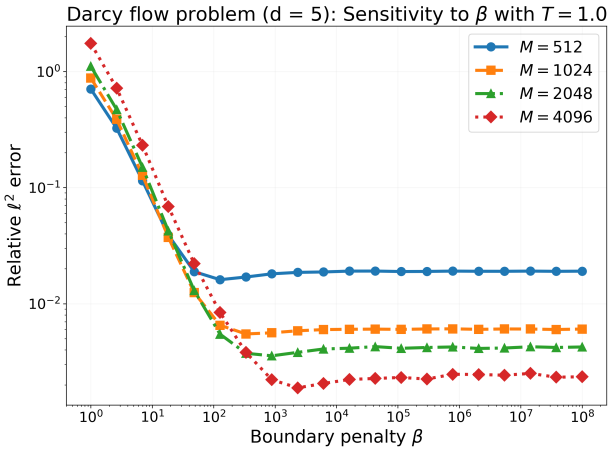
(d) Obstacle problem in Experiment (4.31)

Figure 4.10: **Sensitivity of the variance scale** b_{RF}: Relative ℓ^2 error versus the number of random features for (a) the two-dimensional Poisson problem, (b) the high-dimensional Poisson problem, (c) the Darcy flow problem, and (d) the dome obstacle problem.

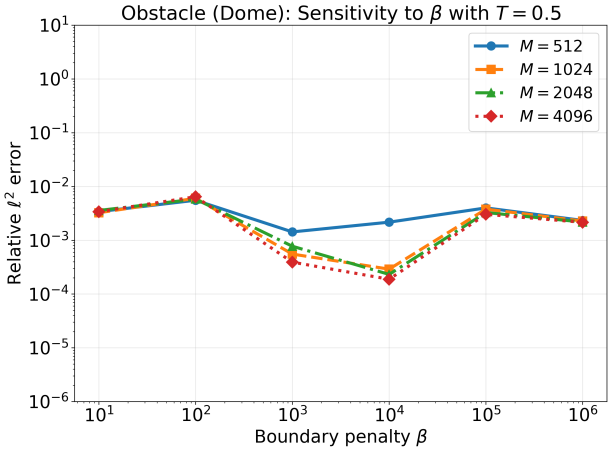


(a) Poisson problem in Experiment (4.25)

(b) High-dimensional Poisson problem

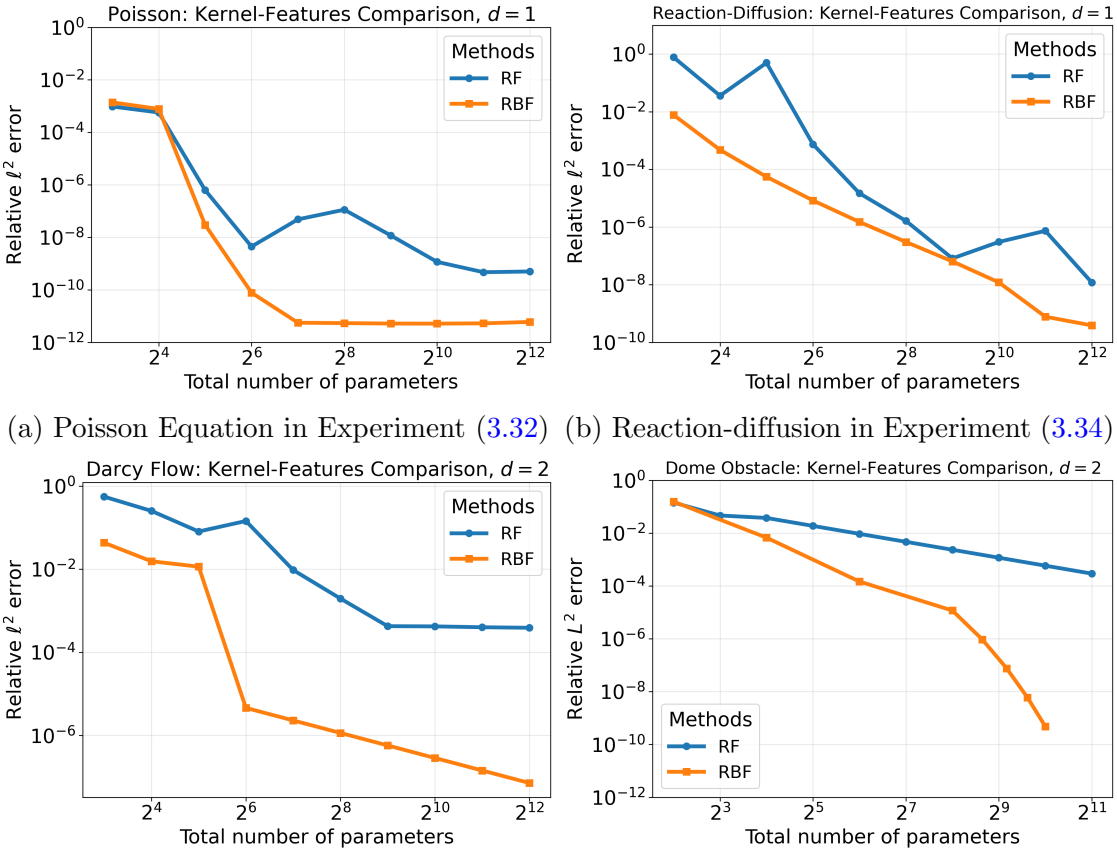


(c) Darcy Flow in Experiment (4.29)



(d) Obstacle problem in Experiment (4.31)

Figure 4.11: **Sensitivity of the boundary penalty weight** β: Relative ℓ^2 error versus the number of random features for (a) the two-dimensional Poisson problem, (b) the high-dimensional Poisson problem, (c) the Darcy flow problem, and (d) the dome obstacle problem.



(a) Poisson Equation in Experiment (3.32) (b) Reaction-diffusion in Experiment (3.34)

(c) Darcy Flow in Experiment (4.29) (d) Obstacle Problem in Experiment (4.31)

Figure 4.12: **Relative ℓ^2 error versus the total number of parameters for the kernel-feature approximations.** The comparison is shown using linear elliptic PDEs and free-BVP: (a) the one-dimensional Poisson equation, (b) the one-dimensional reaction-diffusion equation, (c) the two-dimensional Darcy flow problem, and (d) the two-dimensional dome obstacle problem. The RBF approximation consistently achieves lower errors as the total number of basis functions increases, while the RFM exhibits convergence as the number of randomized features increases.

4.7 Limitations and Remarks

Limitations. Despite the numerical performance of the proposed variational random feature framework, several limitations should be acknowledged. First, the approximation model is not fully adaptive because the hidden parameters are sampled at random from the probability density function and then kept fixed throughout the computation. As a result, the accuracy depends on the number of random features M and on the selected sampling hyperparameters, such as the variance scale b_{RF} and the expansion factor T_{RF}. Another practical limitation of high-dimensional variational solvers is memory consumption. Although the RFM formulation avoids the dense global RBF kernel matrices used in Chapter 3, the construction and storage of feature matrices, derivative matrices, and boundary penalty matrices still become expensive when both the number of random features and the number of collocation points are large. In high dimensions, memory cost depends on both the number of features M and the number of interior and boundary samples needed for Monte Carlo approximations of the variational integrals. As seen in Figures 4.3(b)-4.6(c), higher dimensions $d \geq 2$ require more features for accurate approximation, creating a trade-off between accuracy, sampling resolution, and memory. Future work may leverage sparse feature representations, matrix-free optimization, and improved graphics processing unit (GPU) memory management. Furthermore, the discrete systems stemming from the variational formulation can become ill-conditioned when the number of features is large or when the sampled features exhibit strong correlations, which makes stabilization via TSVD necessary. Although this procedure is effective in practice, as demonstrated in Chapter 3, it introduces an additional tuning parameter, namely the truncation threshold τ. For obstacle-type problems, the ADMM formulation also involves penalty parameters such as ρ, stopping tolerances, and iteration limits, which further increase the tuning effort and computational complexity for more demanding free-BVPs.

Remarks. Overall, the numerical experiments demonstrate that the variational random feature method offers an effective balance between accuracy and efficiency for high-dimensional elliptic and obstacle-type BVPs. The approach is mesh-free, requires only sampled interior and boundary points, and yields optimization problems that can be solved efficiently using TSVD and ADMM. At the same time, the sensitivity analyses show that parameter selection is crucial to ensure monotone convergence. Consequently, the proposed approach should be regarded as a practical and flexible numerical framework that integrates variational principles with the random feature approximation, even though further improvements in adaptivity, conditioning, and theoretical analysis of the variance scale b_{RF} are still required.

Chapter 5

Conclusion and Future Work

This thesis developed mesh-free variational approximation frameworks for elliptic PDEs and obstacle-type problems using kernel-feature representations. The principal objective was to construct numerical methods that are stable, accurate, and computationally efficient, while avoiding the need for mesh generation and remaining compatible with variational formulations. Within this setting, two complementary approximation paradigms were investigated: RBF models for low- and moderate-dimensional problems, and RFMs for high-dimensional problems.

5.1 Conclusion and Finding Remarks

The primary contribution of this thesis is the development of a stabilized variational RBF framework for the numerical solution of linear elliptic and free-BVPs. Within this framework, the governing partial differential equations are reformulated as variational minimization problems, in which Dirichlet boundary conditions are imposed through penalty terms, while obstacle constraints are incorporated through an ℓ^1-penalty formulation. Since the dense matrices generated by RBF discretizations are ill-conditioned, TSVD is employed as a stabilization technique. The choice of the domain expansion factor T leads to high accuracy in the variational RBF framework. For the resulting non-smooth minimization problems associated with obstacle constraints, the ADMM is adopted as an efficient optimization algorithm. The numerical experiments presented in Chapter 3 indicate that the proposed approach attains high accuracy for Poisson, reaction–diffusion, and obstacle-type problems and exhibits robustness with respect to the algorithmic hyperparameters, includ-

ing the expansion factor T, the truncation threshold τ, the variance scale b_{RBF}, the kernel shape multiplier $c(T, \tau)$, and the boundary penalty weight β.

The second contribution is the extension of this variational framework to high-dimensional problems via the RFMs. In contrast to the classical RBF method, which computational cost becomes increasingly prohibitive in higher dimensions due to the formation and solution of dense linear systems, the RFMs offer a scalable alternative by using a randomized feature mapping. This construction preserves the variational structure of the original problem and leads to convex optimization formulations that remain practical in high-dimensional PDEs. The results in Chapter 4 show that the proposed RFM method is an effective and flexible methodology for high-dimensional elliptic and obstacle-type problems. Furthermore, the numerical experiments indicate that the RFM performs competitively with other mesh-free approximations, while the accuracy and sensitivities depend in a structured way on the variance scale b_{RF}, expansion factor T_{RF}, and boundary penalty parameter β.

In summary, the results presented in this thesis establish that kernel-feature-based variational methods provide an effective framework for the numerical solution of linear elliptic and obstacle-type problems. The RBF approach offers more competitive accuracy and robustness in low-dimensional settings. The random feature approach complements this by offering a more scalable framework for high-dimensional problems. Thus, one of the main conceptual outcomes of the thesis is that both RBF and random feature frameworks can be treated within the variational principles, thereby connecting kernel-feature methods, numerical linear algebra, and optimization techniques in a unified manner.

Concurrently, the present work also reveals several limitations and drawbacks. Within the RBF framework, ill-conditioning emerges as a major challenge as the number of basis functions increases, while the computational cost associated with solving the resulting dense linear systems becomes prohibitive for high-dimensional PDEs. In the context of obstacle problems, reduced regularity and potential discontinuities in the obstacle function further deteriorate approximation accuracy and complicate the analysis of convergence behavior. Regarding the random feature method, although it exhibits favorable scalability, its performance remains sensitive to the choice of feature distributions and hyper-parameters. Collectively, these findings indicate that further advances in adaptive approximation strategies and principled feature selection may be required to improve the efficiency, robustness, and generality of kernel-feature variational methods.

In conclusion, the thesis provides computational contributions to the study of mesh-free variational frameworks for PDEs. We combine RBF networks, RFMs, variational formulations, and optimization techniques to establish a framework for solving linear elliptic, obstacle-type, and high-dimensional BVPs. Accordingly, it also identifies several

directions in which the methodology may be further strengthened and extended.

5.2 Further Directions

Several directions for future research arise from the developments presented in this thesis.

First and foremost, it is necessary to extend the current framework to broader classes of PDEs, including nonlinear elliptic equations, time-dependent problems, and systems of coupled equations. Since the methods are developed based on variational principles, such extensions appear feasible, provided that optimization techniques and stability analyses are incorporated.

Second, additional research is required to address more general boundary conditions and complex geometries. While the present work primarily focuses on Dirichlet boundary conditions enforced through a penalty formulation, it is also important to examine Neumann, Robin, and mixed boundary conditions in the context of the mesh-free variational framework. Likewise, problems posed on irregular domains, non-convex geometries, or domains with limited boundary regularity provide a stronger test of the flexibility of both RBF and RFM approximations.

Third, computational efficiency remains an important research direction. For RBF models, this includes developing more efficient low-rank approximations, localized bases, and preconditioners for large kernel systems. For random feature methods, further work examines how the sampling distributions for random weight matrices and biases, as well as adaptive feature selection strategies, affect approximation accuracy, conditioning, and scalability.

A promising direction is adaptive sampling. In this thesis, interior and boundary collocation points were sampled independently, yielding a reproducible framework. However, for PDEs with singularities or sharp transitions, adaptive refinement near high-error regions is needed to improve both accuracy and efficiency.

A theoretical analysis of kernel-feature variational methods remains an important long-term objective. As shown in Chapter 3, this thesis acknowledges the formula for the kernel shape multiplier $c = c(T, \tau)$ from [5] as a guideline for the empirical validation of the variational RBF framework, but many analytical questions remain. These include deriving $c(T, \tau)$ within the variational principles, establishing convergence rates, and proving stability results that quantify sensitivities to truncation and the expansion domain factor. Such developments strengthen the foundations of these methods and clarify their range of applicability.

This thesis concludes that the KF method with a variational framework forms a class of mesh-free numerical methods for PDEs. Their accuracy and compatibility with optimization techniques provide a useful basis for future work in scientific machine learning, high-dimensional numerical PDEs, and constrained variational problems.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: a system for Large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[2] Ben Adcock and Daan Huybrechs. Frames and numerical approximation. *SIAM Review*, 61(3):443–473, 2019.

[3] Ben Adcock and Daan Huybrechs. Frames and numerical approximation. *SIAM Review*, 61(3):443–473, 2019.

[4] Ben Adcock and Daan Huybrechs. Frames and numerical approximation ii: Generalized sampling. *Journal of Fourier Analysis and Applications*, 26(6):87, 2020.

[5] Ben Adcock, Daan Huybrechs, and Cécile Piret. Stable and accurate least squares radial basis function approximations on bounded domains. *SIAM Journal on Numerical Analysis*, 62(6):2698–2718, 2024.

[6] Andersen Ang, Hans De Sterck, and Stephen Vavasis. MGProx: A nonsmooth multigrid proximal gradient method with adaptive restriction for strongly convex optimization. *SIAM Journal on Optimization*, 34(3):2788–2820, 2024.

[7] Cosmin Anitescu, Elena Atroshchenko, Naif Alajlan, and Timon Rabczuk. Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua*, 59(1):345–359, 2019.

[8] Claudio Baiocchi. Variational and quasivariational inequalities. *Applications to free-boundary problems*, 1984.

[9] Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 685–694. PMLR, 13–18 Jul 2020.

[10] Pau Batlle, Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. Error analysis of kernel/GP methods for nonlinear and parametric pdes. *Journal of Computational Physics*, 520:113488, 2025.

[11] Christian Beck, Weinan E, and Arnulf Jentzen. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science*, pages 1–57, 2017.

[12] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 541–549. PMLR, 10–15 Jul 2018.

[13] Jen Berg and Kaj Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, 317:28–41, 2018.

[14] Salomon Bochner. *Harmonic analysis and the theory of probability*. Courier Corporation, 2005.

[15] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine learning*, 3(1):1–122, 2011.

[16] Susanne C. Brenner and Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3rd edition, 2008.

[17] Martin Dietrich Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, 2003.

[18] Luis A Caffarelli and Antoine Mellet. Random homogenization of an obstacle problem. In *Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, volume 26, pages 375–395. Elsevier, 2009.

[19] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M Stuart. Solving and learning nonlinear PDEs with Gaussian processes. *Journal of Computational Physics*, 447:110668, 2021.

[20] Zhijun Chen and Hayden Schaeffer. Conditioning of random Fourier feature matrices: double descent and generalization error. *Information and Inference: A Journal of the IMA*, 13(2):iaad054, 04 2024.

[21] Philippe G. Ciarlet. *The Finite Element Method for Elliptic Problems*. SIAM, 2002.

[22] John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Advances in Computational Mathematics*, 6(1):207–226, 1996.

[23] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[24] MWM Gamini Dissanayake and Nhan Phan-Thien. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10(3):195–201, 1994.

[25] Jørgen S Dokken, Patrick E Farrell, Brendan Keith, Ioannis PA Papadopoulos, and Thomas M Surowiec. The latent variable proximal point algorithm for variational problems with constraints. *Computer Methods in Applied Mechanics and Engineering*, 445:118181, 2025.

[26] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[27] Weinan E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

[28] Weinan E, Chao Ma, Lei Wu, and Stephan Wojtowytsch. Towards a mathematical understanding of neural network-based machine learning: What we know and what we don't. *CSIAM Transactions on Applied Mathematics*, 1(4):561–615, 2020.

[29] Weinan E and Bing Yu. The Deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[30] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[31] Hamid El Bahja, Jan C Hauffen, Peter Jung, Bubacarr Bah, and Issa Karambal. A physics-informed neural network framework for modeling obstacle-related equations. *Nonlinear Dynamics*, pages 1–12, 2025.

[32] Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society, 2nd edition, 2010.

[33] Shikai Fang, Madison Cooley, Da Long, Shibo Li, Mike Kirby, and Shandian Zhe. Solving high frequency and multi-scale PDEs with Gaussian processes. In *The Twelfth International Conference on Learning Representations*, 2024.

[34] Gregory Eric Fasshauer. *Meshfree Approximation Methods with MATLAB*. World Scientific, 2007.

[35] Bengt Fornberg. *A Practical Guide to Radial Basis Functions*. SIAM, Philadelphia, PA, 2015.

[36] Simon Foucart. *Mathematical Pictures at a Data Science Exhibition*. Cambridge University Press, 2022.

[37] Simon Foucart, Chunyang Liao, Shahin Shahrampour, and Yinsong Wang. Learning from non-random data in Hilbert spaces: an optimal recovery perspective. *Sampling Theory, Signal Processing, and Data Analysis*, 20(5), 2022.

[38] Avner Friedman. *Variational Principles and Free-Boundary Problems*. Wiley, 1982.

[39] Masaaki Fujii, Akihiko Takahashi, and Masayuki Takahashi. Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs. *Asia-Pacific Financial Markets*, pages 1–18, 2017.

[40] Roland Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Springer-Verlag, 1984.

[41] Roland Glowinski, Jacques-Louis Lions, and Raymond Trémolières. *Numerical Analysis of Variational Inequalities*. North-Holland, 1981.

[42] D. Gobovic and Mona Zaghloul. Analog cellular neural network with application to partial differential equations with variable mesh-size. In *Proceedings of IEEE International Symposium on Circuits and Systems*, volume 6, pages 359–362. IEEE, 1994.

[43] Gene Howard Golub and Charles Francis Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 4 edition, 2013.

[44] Jiequn Han, Arnulf Jentzen, and Weinan E. Overcoming the curse of dimensionality: Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences of the United States of America*, 115(34):8505–8510, 2018.

[45] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.

[46] Per Christian Hansen. *Discrete inverse problems: insight and algorithms*. SIAM, 2010.

[47] Roger Alan Horn and Charles Royal Johnson. *Matrix analysis*. Cambridge university press, 2012.

[48] Daniel Hsu, Clayton H Sanford, Rocco Servedio, and Emmanouil Vasileios Vlatakis-Gkaragkounis. On the approximation power of two-layer networks of random ReLUs. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 2423–2461. PMLR, 15–19 Aug 2021.

[49] Guang-Bin Huang, Lei Chen, and Chee-Kheong Siew. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, 17(4):879–892, 2006.

[50] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1):489–501, 2006. Neural Networks.

[51] Thomas Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation, 2012.

[52] Boris Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 6(6):1320–1329, 1995.

[53] Edward Kansa. A numerical method for solving ill-conditioned equation systems arising from radial basis functions. *American Journal of Computational Mathematics*, 13(2):356–370, 2023.

[54] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*, 2019.

[55] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Solving for high-dimensional committor functions using artificial neural networks. *Research in the Mathematical Sciences*, 6(1):1, 2019.

[56] David Kinderlehrer and Guido Stampacchia. *An Introduction to Variational Inequalities and Their Applications*. SIAM, 2000.

[57] Nikolay Krylov. Essentials of real analysis and Morrey-Sobolev spaces for second-order elliptic and parabolic PDEs with singular first-order coefficients. *arXiv preprint arXiv:2505.14863*, 2025.

[58] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[59] Samuel Lanthaler and Nicholas H. Nelsen. Error bounds for learning with vector-valued random features. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[60] Tan Phuong Dong Le, Giang Tran, and Hans De Sterck. Random-feature solvers for high-dimensional variational problems: A practical framework. *Preprint*, 2026. Available as manuscript.

[61] Tan Phuong Dong Le, Giang Tran, and Hans De Sterck. Stable mesh-free variational radial basis function approximation for elliptic PDEs and obstacle problems. *arXiv preprint arXiv:2604.19882*, 2026.

[62] Hyuk Lee and In Seok Kang. Neural algorithm for solving differential equations. *Journal of Computational Physics*, 91(1):110–131, 1990.

[63] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel "Ridgeless" regression can generalize. *The Annals of Statistics*, 48(3):1329 – 1347, 2020.

[64] Chunyang Liao. Solving partial differential equations with random feature models. *arXiv preprint arXiv:2501.00288*, 2024.

[65] Chunyang Liao, Deanna Needell, and Hayden Schaeffer. Cauchy random features for operator learning in Sobolev space. *arXiv preprint arXiv:2503.00300*, 2025.

[66] Chunyang Liao, Deanna Needell, Hayden Schaeffer, and Alexander Xue. Differentially private random feature model. *arXiv preprint arXiv:2412.04785*, 2024.

[67] Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens. Random features for kernel approximation: A survey on algorithms, theory, and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7128–7148, 2021.

[68] Hao Liu and Dong Wang. Fast operator splitting methods for obstacle problems. *Journal of Computational Physics*, 477:111941, 2023.

[69] Martin Magill, Faisal Z Qureshi, and Hendrick W de Haan. Neural networks trained to solve differential equations learn general representations. In *Advances in Neural Information Processing Systems*, pages 4071–4081, 2018.

[70] A.K. Malik, Ruobin Gao, M.A. Ganaie, M. Tanveer, and Ponnuthurai Nagaratnam Suganthan. Random vector functional link network: Recent developments, applications, and future directions. *Applied Soft Computing*, 143:110377, 2023.

[71] Andrew J Meade Jr and Alvaro A Fernandez. The numerical solution of linear ordinary differential equations by feedforward neural networks. *Mathematical and Computer Modelling*, 19(12):1–25, 1994.

[72] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration. *Applied and Computational Harmonic Analysis*, 59:3–84, 2022. Special Issue on Harmonic Analysis and Machine Learning.

[73] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75, 2019.

[74] Boris S. Mityagin. The zero set of a real analytic function. *Mathematical Notes*, 107(3):529–530, 2020.

[75] Mohammad Amin Nabian and Hadi Meidani. A deep neural network surrogate for high-dimensional random partial differential equations. *arXiv preprint arXiv:1806.02957*, 2018.

[76] Francis J Narcowich, Joseph D Ward, and Holger Wendland. Sobolev error estimates and a Bernstein inequality for scattered data interpolation via radial basis functions. *Constructive Approximation*, 24(2):175–186, 2006.

[77] Deanna Needell, Aaron A. Nelson, Rayan Saab, Palina Salanevich, and Olov Schavemaker. Random vector functional link networks for function approximation on manifolds. *Frontiers in Applied Mathematics and Statistics-Optimization*, 10, 2024.

[78] Houman Owhadi and Clint Scovel. *Operator-Adapted Wavelets, Fast Solvers, and Numerical Homogenization: From a Game Theoretic Approach to Numerical Approximation and Algorithm Design*. Cambridge University Press, 2019.

[79] Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[80] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2006.

[81] Alfio Quarteroni and Alberto Valli. *Numerical approximation of partial differential equations*, volume 23. Springer, 2008.

[82] Alfio Quarteroni and Alberto Valli. *Numerical Models for Differential Problems*. Springer, 2nd edition, 2009.

[83] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310. PMLR, 09–15 Jun 2019.

[84] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Neural Information Processing Systems*, 2007.

[85] Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561, 2008.

[86] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

[87] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1–24, 2018.

[88] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[89] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.

[90] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. 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, 2019.

[91] Jose Francisco Rodrigues. *Obstacle problems in mathematical physics*, volume 134. Elsevier, 1987.

[92] Basri Ronen, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[93] Xavier Ros-Oton. Obstacle problems and free boundaries: an overview. *SeMA Journal*, 75(3):399–419, 2018.

[94] Keith Rudd and Silvia Ferrari. A constrained integration (cint) approach to solving partial differential equations using artificial neural networks. *Neurocomputing*, 155:277–285, 2015.

[95] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3218–3228, Red Hook, NY, USA, 2017. Curran Associates Inc.

[96] Robert Schaback. Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3):251–264, 1995.

[97] Bernhard Scholköpf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[98] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[99] Sirignano, Justin and Spiliopoulos, Konstantinos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[100] Yitong Sun, Anna Gilbert, and Ambuj Tewari. But how does it work in theory? linear SVM with random features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[101] Yoshiro Suzuki. Neural network-based discretization of nonlinear differential equations. *Neural Computing and Applications*, pages 1–16, 2017.

[102] Gábor Szegő. *Orthogonal Polynomials*, volume 23 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI, 4 edition, 1975.

[103] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

[104] Luc Tartar. *An introduction to Sobolev spaces and interpolation spaces*. Springer, 2007.

[105] James William Thomas. *Numerical Partial Differential Equations: Finite Difference Methods*, volume 22. Springer, 2013.

[106] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating Eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3424–3433, 2017.

[107] Giang Tran, Hayden Schaeffer, William M Feldman, and Stanley J Osher. An L^1 penalty method for general obstacle problems. *SIAM Journal on Applied Mathematics*, 75(4):1424–1444, 2015.

[108] Lloyd Nick Trefethen and David Bau. *Numerical linear algebra*. SIAM, 2022.

[109] Raymond Trémolieres, Jacques-Louis Lions, and Roland Glowinski. *Numerical analysis of variational inequalities*, volume 8. Elsevier, 2011.

[110] Jian Wang, Siyuan Lu, Shui-Hua Wang, and Yu-Dong Zhang. A review on extreme learning machine. *Multimedia Tools and Applications*, 81(29), 2022.

[111] Lixin Wang and Jerry Mendel. Structured trainable networks for matrix algebra. In *Proceedings of the International Joint Conference on Neural Networks*, pages 125–132. IEEE, 1990.

[112] Sifan Wang, Hanwen Wang, and Paris Perdikaris. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384:113938, 2021.

[113] Yiran Wang and Suchuan Dong. An extreme learning machine-based method for computational PDEs in higher dimensions. *Computer Methods in Applied Mechanics and Engineering*, 418:116578, 2024.

[114] Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, 2005.

[115] Kailai Xu, Bella Shi, and Shuyi Yin. Deep learning for partial differential equations, 2018.

[116] Zhitong Xu, Da Long, Yiming Xu, Guang Yang, Shandian Zhe, and Houman Owhadi. Toward efficient kernel-based solvers for nonlinear PDEs. *arXiv:2410.11165*, 2024.

[117] Zhitong Xu, Da Long, Yiming Xu, Guang Yang, Shandian Zhe, and Houman Owhadi. Toward Efficient Kernel-Based Solvers for Nonlinear PDEs. *arXiv preprint arXiv:2410.11165*, 2024.

[118] Yibo Yang and Paris Perdikaris. Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, 394:136–152, 2019.

[119] R. Yentis and Mona Zaghloul. VLSI implementation of locally connected neural network for solving partial differential equations. *IEEE Transactions on Circuits and Systems I*, 43(8):687–690, 1996.

[120] Bing Yu and Weinan E. The Deep Ritz Method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[121] Yuncheng Zang, Gang Bao, Xiaoming Ye, and Haomin Zhou. Weak adversarial networks for high-dimensional partial differential equations. *Journal of Computational Physics*, 411:109409, 2020.

[122] Eberhard Zeidler. *Applied functional analysis: applications to mathematical physics*, volume 108. Springer Science & Business Media, 2012.

[123] Shougui Zhang, Xiyong Cui, Guihua Xiong, and Ruisheng Ran. An optimal ADMM for unilateral obstacle problems. *Mathematics*, 12(12):1901, 2024.

[124] Xiaoqun Zhang, Martin Burger, and Stanley Osher. A unified primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing*, 46:20–46, 2011.

[125] Dominique Zosso, Braxton Osting, Mandy Xia, and Stanley J Osher. An efficient primal-dual method for the obstacle problem. *Journal of Scientific Computing*, 73(1):416–437, 2017.

APPENDICES

Appendix A

Extendable Tables & Hyper-parameters

A.1 Hyper-parameters for experiments in Chapter 3

Tables A.1-A.5 summarize the experimental settings used in the sensitivity studies for the expansion domain factor T, the kernel shape multiplier $c(T, \tau)$, the truncation threshold τ, the boundary penalty parameter β, and the basis function ϕ.

Table A.1: Summary of sensitivity experiments for the expansion domain factor T.

Problem	**Tested T**	**Selected T**	**Scaling of b**	**Oversampling ζ**
1D Poisson	{1.5, 2, 4, 8}	8	$b = cN$	2
1D Reaction–diffusion	{1.5, 2, 4, 6}	2	$b = cN$	2
1D Single-bump obstacle	{2, 3, 4, 6}	2	$b = cN$	2
1D Two-bump obstacle	{2, 3, 4, 6}	3	$b = cN$	2
2D Dome obstacle	{2, 3, 4, 6}	3	$b = c\sqrt{N}$	4

Table A.2: Summary of sensitivity experiments for the kernel shape multiplier $c(T, \tau)$.

Problem	**Fixed** T	**Fixed** τ	**Fixed** β	**Scaling of** b	**Shape multiplier** c**)**
1D Poisson	8	10^{-15}	3×10^5	$b = cN$	$c = c(T, \tau)$
1D Reaction–diffusion	2	10^{-15}	10^6	$b = cN$	$c = c(T, \tau)$
1D Single-bump obstacle	2	10^{-15}	10^6	$b = cN$	$c = c(T, \tau)$
1D Two-bump obstacle	3	10^{-15}	10^8	$b = cN$	$c = c(T, \tau)$
2D Dome obstacle	3	10^{-15}	10^6	$b = c\sqrt{N}$	$c = c(T, \tau)$

Table A.3: Summary of sensitivity experiments for the truncation threshold τ.

Problem	**Fixed** T	**Fixed** β	**Fixed** ζ	**Shape parameter**	**Tested** τ
1D Poisson	8	3×10^5	2	$c = c(T, \tau)$	multiple values, including 10^{-12} and 10^{-15}
1D Reaction–diffusion	2	10^6	2	$c = c(T, \tau)$	multiple values, including 10^{-12} and 10^{-15}
1D Single-bump obstacle	2	10^6	2	$c = c(T, \tau)$	multiple values, including 10^{-12} and 10^{-15}
1D Two-bump obstacle	3	10^8	2	$c = c(T, \tau)$	multiple values, including 10^{-12} and 10^{-15}
2D Dome obstacle	3	10^6	4	$c = c(T, \tau)$	multiple values, including 10^{-12} and 10^{-15}

Table A.4: Summary of sensitivity experiments for the boundary penalty parameter β.

Problem	**Fixed T**	**Fixed τ**	**Scaling of b**	**Reference β**
1D Poisson	8	10^{-15}	$b = cN$	3×10^5
1D Reaction–diffusion	2	10^{-15}	$b = cN$	10^6
1D Single-bump obstacle	2	10^{-15}	$b = cN$	10^6
1D Two-bump obstacle	3	10^{-15}	$b = cN$	10^8
2D Dome obstacle	3	10^{-15}	$b = c\sqrt{N}$	10^6

Table A.5: Summary of sensitivity experiments for the basis/kernel function ϕ.

Problem	**Fixed T**	**Fixed τ**	**Fixed β**	**Scaling of b**	**Tested basis functions ϕ**
1D Poisson	8	10^{-15}	3×10^5	$b = cN$	Gaussian, MQ, IMQ, IQ, Wendland
1D Reaction–diffusion	2	10^{-15}	10^6	$b = cN$	Gaussian, MQ, IMQ, IQ, Wendland
1D Single-bump obstacle	2	10^{-15}	10^6	$b = cN$	Gaussian, MQ, IMQ, IQ, Wendland
1D Two-bump obstacle	3	10^{-15}	10^8	$b = cN$	Gaussian, MQ, IMQ, IQ, Wendland
2D Dome obstacle	3	10^{-15}	10^6	$b = c\sqrt{N}$	Gaussian, MQ, IMQ, IQ, Wendland

A.2 Settings for experiments in Chapter 4

In the numerical experiments presented in Chapter 4, a baseline configuration is used for the variational RFMs. Unless specified otherwise, we take the activation function to be $\tanh(\cdot)$, fix the variance scale at $b_{\mathrm{RF}} = 1.75$, set the oversampling ratio to $\zeta = 2$, choose a test set size of $m^{\mathrm{test}} = 2000$, and use a truncated singular value threshold of $\tau = 10^{-15}$. The numbers of interior and boundary collocation points are given by $m_\Omega = \zeta M$ and $m_{\partial\Omega} = 2dk$, respectively. For the obstacle problem, the ADMM solver additionally employs $\rho = 10.3$, $\mu = 10N^2$, and a stopping tolerance of $\epsilon_{\mathrm{stop}} = 10^{-4}$.

The experiment-specific parameter choices are as follows: for the variational form of the Poisson problem, $(M, k, T_{\mathrm{RF}}, \beta) = (2048, 100, 0.1, 3\times10^4)$; for the high-dimensional Poisson problem, the main approximation experiment uses $(M, k, T_{\mathrm{RF}}) = (4096, 500, 2.2)$, whereas the comparison experiment uses $(M, k, T_{\mathrm{RF}}, \beta) = (4096, 500, 2.2, 10^6)$; for the Darcy flow problem, we set $(M, k, T_{\mathrm{RF}}, \beta) = (4096, 500, 1.0, 10^6)$; and for the two-dimensional obstacle problem, we choose $(M, k, T_{\mathrm{RF}}, \beta) = (2048, 100, 0.5, 10^4)$.

Table A.6: Hyperparameter settings in Chapter 4.

Parameter	**Description**	**Value**
$\sigma(\cdot)$	Random-feature activation function	$\tanh(\cdot)$
b_{RF}	Variance scale of the random-feature matrix	1.75
ζ	Oversampling ratio	2
m_Ω	Number of interior collocation points	$m_\Omega = \zeta M$
$m_{\partial\Omega}$	Number of boundary collocation points	$m_{\partial\Omega} = 2\,d\,k$
m	Total number of collocation points	$m = m_\Omega + m_{\partial\Omega}$
m^{test}	Number of testing points	2000
τ	Truncation tolerance for TSVD	10^{-15}
ρ	ADMM penalty parameter	10.3
μ	ℓ^1-penalty parameter in obstacle problem	$10M^2$
ϵ_{stop}	Relative stopping tolerance	10^{-4}
`maxIter`	Maximum ADMM iterations	500000

Appendix B

Code Notes and Configuration Details

This appendix provides supplementary implementation notes and configuration specifications for the numerical methods developed in this thesis. Its purpose is to document the practical aspects of the computational framework that underpin the reproducibility, transparency, and systematic organization of the numerical experiments presented in the main chapters. Whereas the principal text emphasizes mathematical formulation, algorithmic design, and numerical analysis, the present appendix B consolidates technical information related to code architecture, parameter configuration, and implementation decisions.

B.1 Reproducibility in the proposed radial basis function method

In particular, this appendix section summarizes the computational settings employed in the experiments, including solver parameters, discretization schemes, initialization strategies, stopping criteria, and other implementation-related conventions. It also records selected notes on the structure of the codebase and the numerical workflow adopted in the experimental studies of the RBF presented in Chapter 3. These details are provided to serve as a precise reference for the practical realization of the proposed methods and to facilitate subsequent verification, replication, reuse, and extension of the computational framework. We present the configuration details and code notes for the numerical experiment for the Poisson's equation, as mentioned in Section 3.3.1. For details on the

configuration and implementation of other experiments, we refer to the following GitHub link at https://github.com/phuongdongle/Variational-Formulation-RBFNets.git.

B.2 Reproducibility in the random features method

This section summarizes the computational settings employed in the experiments, including solver parameters, sampling strategies for interior and boundary points, initialization of random weights and bias terms, stopping criteria, and other implementation-related conventions. It also records selected notes on the structure of the codebase and the numerical workflow adopted in the experimental studies of the random feature method presented in Chapter 4. These details are provided to serve as a precise reference for the practical realization of the proposed methods and to facilitate subsequent verification, replication, reuse, and extension of the computational framework. We present the configuration details and code notes for the numerical experiments on high-dimensional elliptic and obstacle-type problems. For details on the implementation and configuration of other experiments, we refer to the accompanying code repository, which can be accessed at the following GitHub link at https://github.com/phuongdongle/RF-Variational-Formulation-PDE-EOP.

Glossary

activation function A nonlinear function applied to affine feature maps in neural networks or random feature models.

active set The set of points or degrees of freedom where an inequality constraint is active.

alternating direction method of multipliers An optimization algorithm that solves constrained or non-smooth convex optimization problems by splitting them into subminimization problems.

augmented Lagrangian A Lagrangian formulation augmented by a quadratic penalty term for constraint violations.

boundary penalty A penalty term added to the variational functional to enforce Dirichlet boundary conditions.

boundary value problem A partial differential equation problem in which the solution is required to satisfy prescribed conditions on the boundary of the computational domain.

collocation point A sampled point in the interior or on the boundary of the domain used to discretize the variational formulation.

condition number A numerical measure of the sensitivity of a linear system to perturbations in the data.

contact region The subset of the domain where the solution coincides with the obstacle function.

convex optimization An optimization problem in which the objective function and feasible set are convex, so that local minimizers are global minimizers.

Darcy flow problem A linear elliptic boundary value problem involving a spatially varying diffusion coefficient, commonly used to model flow through porous media.

Deep Ritz method A neural-network-based method for solving variational problems by minimizing an energy functional.

Dirichlet boundary condition A boundary condition that prescribes the value of the solution on the boundary of the computational domain.

domain expansion factor A parameter that determines how far the center points of the basis functions are sampled outside the physical domain.

dual residual A residual measuring the change in dual variables or dual feasibility in an optimization algorithm.

energy functional A scalar functional whose minimizer corresponds to the solution of a variational boundary value problem.

free-boundary problem A boundary value problem in which part of the boundary or interface is unknown in advance and must be determined as part of the solution.

hyperbolic tangent activation The activation function $\sigma(z) = \tanh(z)$, used in this thesis to define smooth random features.

ill-conditioning A numerical situation in which small perturbations in the data or round-off errors can cause large changes in the computed solution.

kernel feature method A mesh-free approximation method based on kernel functions or feature representations. Radial basis function and random feature models are examples of kernel-feature-based approximation methods.

kernel shape multiplier A proportionality factor used to determine the radial basis function shape parameter from the domain expansion factor and the truncation threshold.

linear elliptic partial differential equation A partial differential equation whose highest-order terms define an elliptic differential operator. In this thesis, Poisson, reaction-diffusion, and Darcy flow equations are treated as representative elliptic boundary value problems.

mesh-free method A numerical method that constructs an approximation without requiring a mesh or triangulation of the computational domain.

non-smooth optimization An optimization problem involving objective terms that are not differentiable everywhere, such as ℓ^1 penalties or positive-part functions.

obstacle problem A variational inequality problem in which the solution is constrained to lie above a prescribed obstacle function. The unknown contact region between the solution and the obstacle gives rise to a free boundary.

oversampling ratio A parameter controlling the number of collocation points relative to the number of basis functions or random features.

penalty method A method for enforcing constraints by adding weighted penalty terms to an objective functional.

physics-informed neural network A neural-network-based method that trains a model by minimizing residuals of the governing differential equation and boundary conditions.

Poisson equation A linear elliptic partial differential equation of the form $-\Delta u = f$, together with prescribed boundary conditions.

primal residual A residual measuring violation of the primal constraint in an optimization algorithm.

proximal gradient descent An optimization method for problems involving a non-smooth term, using sub-gradient steps combined with proximal updates.

quadratic minimization An optimization problem whose objective function is quadratic in the unknown coefficients.

radial basis function A function whose value depends only on the distance between an input point and a center point.

random feature model A finite-dimensional approximation model constructed from randomly sampled feature functions and trainable linear coefficients.

rank efficiency The ratio between the effective numerical rank after truncation and the total number of basis functions or features.

reaction–diffusion equation A partial differential equation combining diffusion terms with a reaction term. In this thesis, it is used as a benchmark elliptic boundary value problem.

relative ℓ^2 error An error metric defined as the ratio between the discrete ℓ^2 norm of the approximation error and the discrete ℓ^2 norm of the exact solution.

shape parameter A parameter that controls the width or flatness of a radial basis function and affects both approximation accuracy and numerical conditioning.

strong form A formulation of a differential equation requiring sufficient differentiability of the solution.

truncated singular value decomposition A regularization technique in which small singular values are discarded when solving ill-conditioned linear systems.

truncation threshold A tolerance value used in the truncated singular value decomposition to determine which singular values are retained.

variance scale A parameter controlling the variance of the random weights in the random feature model.

variational formulation A formulation of a differential equation in which the solution is characterized as the minimizer of an energy functional or as the solution of a weak integral equation.

variational inequality A variational problem with inequality constraints, commonly arising in obstacle and free-boundary problems.

weak form An integral formulation of a differential equation obtained by testing against suitable functions, often used as the basis for variational and Galerkin methods.