UNIVERSIDADE ABERTA

Spectral Approach For Improving The Ill-Conditioning Of The Method Of Fundamental Solutions

Hernani Matamba Kujijila Calunga

Doutoramento em Matemática Aplicada e Modelação

2025

UNIVERSIDADE ABERTA

Spectral Approach For Improving The Ill-Conditioning Of The Method Of Fundamental Solutions

Hernani Matamba Kujijila Calunga

Doutoramento em Matemática Aplicada e Modelação

Tese orientada por

Professor Doutor Pedro Miguel Picado de Carvalho Serranho
Professor Doutor Pedro Ricardo Simão Antunes

July 2025

Copyright

Acknowledgements

The COVID-19 pandemic brought with it many challenges, but also unexpected opportunities. During that time, as I adjusted to working from home, I found myself with extra hours in the day. I felt a strong need to use that time meaningfully, so I turned to subjects I had always been passionate about, with the goal of deepening my understanding.

While discussing with close family about how I was spending my time, the idea of pursuing a doctoral degree came up. It was a long-standing personal ambition, and with his encouragement, I decided to take the opportunity. I enrolled in the PhD program in Applied Mathematics and Modelling at UAb.

Completing this academic journey means a great deal to me. It represents not only the fulfilment of a personal goal but also a way to inspire my children, to show them that learning is a lifelong pursuit and that it's never too late to chase your dreams.

As I approach the conclusion of this stage in my academic journey, I wish to express my deepest gratitude to those who have supported me along the way.

To **Prof. Pedro Miguel Picado de Carvalho Serranho, PhD**, and **Prof. Pedro Ricardo Simão Antunes, PhD** – your unwavering guidance throughout my research has been invaluable. The countless hours of discussions and exchanges alleviated my initial concerns and strengthened my confidence in this project. Your mentorship was instrumental in reaching this milestone.

To **Fernando Dala, PhD** – who first introduced me to UAb as a potential path for realizing this academic aspiration while allowing me to maintain balance in my existing responsibilities, thank you for your insight and encouragement.

To my uncle, **Mr Jonadab Jacob** – who constantly pushed me toward pursuing a PhD, uncle, here we are, with the degree almost in hand! Your belief in my potential has been a driving force throughout this process.

To the **academic community at UAb**, professors and colleagues – our interactions, though remote, were as enriching as if we had shared a physical classroom.

To my parents **Mr Armando Calunga and Mrs Azenate Calunga** – I owe a significant part of who I am. May God continue to bless you. I hold deep respect and appreciation for all you have done for me and my dear brother and sisters.

To my beloved wife, **Dinamene Calunga, and my childrens, Teresa, Isandro, Aléssia, and Lívia** – I know I stole time that should have been dedicated to you, yet you never complained. Your unwavering support and understanding mean everything to me. Thank you for standing by me, and I sincerely apologize for the moments we missed together due to my research commitments. Now, it's time to

celebrate!

Above all, to **God**, the source of wisdom – knowledge always begin with reverence for You.

Dinamene, what an incredible journey we are doing together;

Isandro, Aléssia and Lívia, hope this inspire you to dream big;

My Parents, so blessed to have been influenced by you.

UNIVERSIDADE
AbERTA
www.uab.pt

DECLARAÇÃO DE INTEGRIDADE

STATEMENT OF INTEGRITY

Declaro ter atuado com integridade na elaboração da presente dissertação/tese. Confirmo que em todo o trabalho conducente à sua elaboração não recorri à prática de plágio ou a qualquer outra forma de falsificação de resultados.

Mais declaro que tomei conhecimento integral do Regulamento Disciplinar da Universidade Aberta, publicado no Diário da República, 2.ª série, n.º 215, de 6 de novembro de 2013.

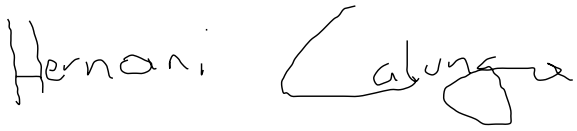
I hereby declare having conducted my thesis with integrity. I confirm that I have not used plagiarism or any form of falsification of results in the process of the thesis elaboration.

I further declare that I have fully acknowledged Disciplinary Regulations of the Universidade Aberta (regulation published in the official journal Diário da República, 2.ª série, N.º 215, de 6 de novembro de 2013).

Universidade Aberta, 21 de Março de 2025

Nome completo/Full name: Hernani Matamba Kujijila Calunga

Assinatura/Signature:



manuscrita ou digital / handwritten or digital

Spectral Approach For Improving The Ill-Conditioning Of The Method Of Fundamental Solutions

Abstract

The method of fundamental solutions with singular value decomposition (MFS-SVD) introduced in [12] is potentially a good alternative to the existing methods to improve the conditioning of the method of fundamental solutions (MFS). The MFS-SVD approach proposes a change of basis to a new basis that spans the same space of approximations in which the linear system becomes better conditioned. The original formulation considered polar coordinates and harmonic polynomials as basis functions and is restricted to the Laplace equation in 2D. In this thesis, we adapt the Laplace equation approach to the 2D Helmholtz equation in polar coordinates with Bessel functions as basis functions; additionally, we extend it to elliptic coordinates with Mathieu functions as basis functions to improve conditioning in other domains such as an eccentric ellipse. However, the implementation to both boundary value problems (BVPs) suggests that the singular value decomposition (SVD) should be applied to the ill-conditioned factor of the MFS linear system matrix decomposition, but this does not work for higher order problems. Therefore, this work brings a key contribution by proposing and illustrating a general MFS-SVD workflow, designed to be applicable regardless of the specific BVP under consideration. To illustrate this, we focus on the biharmonic BVP, a fourth-order partial differential equation, and explore two distinct ansatzes for its numerical solution. Each ansatz involves a different addition theorem and a corresponding decomposition, thereby reinforcing the robustness and flexibility of the MFS-SVD approach across varying numerical formulations. The effectiveness of the proposed workflow is demonstrated through numerical examples for both ansatzes.

Key-words: Method of fundamental solutions, Helmholtz BVP, biharmonic BVP, MFS-SVD, Ill-conditioning, Mathieu functions.

Método Espectral Para Melhorar o Mau Condicionamento Do Método Das Soluções Fundamentais

Resumo

O método das soluções fundamentais, em inglês, *method of fundamental solutions* (MFS), é um método para solução de problemas de valor de fronteira de equações diferenciais com derivadas parciais. O método pertence à classe dos métodos que não necessitam de discretização do domínio, evitando, dessa forma, os desafios que podem surgir com a discretização de domínios complexos, como ocorre nos métodos que requerem essa etapa.

O MFS busca a solução do problema de valor de fronteira na forma de uma combinação linear de soluções fundamentais da equação diferencial com derivadas parciais, com as singularidades da solução fundamental localizadas fora do domínio de definição do problema. Além de dispensar a discretização do domínio, outra vantagem que o torna um método atrativo para os pesquisadores é a facilidade de implementação numérica e sua excelente convergência à solução, que, em alguns casos, é de carácter exponencial [17, 31]. No entanto, a convergência do método é frequentemente comprometida devido ao mau condicionamento do sistema linear resultante da aplicação do método. Esse mau condicionamento decorre de vários fatores, sendo um deles a localização das singularidades, que, em princípio, podem ser colocadas em qualquer ponto do plano ou espaço, dependendo do domínio de definição do problema, excepto no seu interior e na fronteira do domínio.

Como o mau condicionamento pode gerar instabilidade numérica e degradar a precisao da solução, é recomendável o seu tratamento antes da resolução do sistema linear. Para o efeito existem várias abordagens para mitigar esse problema, cada uma com suas vantagens e limitações, o que faz desse tema um campo de pesquisa ainda ativo.

Esta tese trata do problema do mau condicionamento do método das soluções fundamentais aplicado a dois problemas concretos de valor de fronteira. Ela começa com uma apresentação das noções teóricas básicas sobre o MFS, sua formulação

numérica, bem como uma descrição das diferentes estratégias actualmente utilizadas para reduzir o mau condicionamento.

Como já mencionado, apesar da existência de diversas abordagens para mitigar o mau condicionamento, novas técnicas têm sido sugeridas como resultado de pesquisas recentes. Uma delas é uma técnica que combina o método das soluções fundamentais com a decomposição em valores singulares (SVD), conhecida como MFS-SVD, introduzida em [12], com o objectivo de melhorar o mau condicionamento do problema de valor de fronteira da equação de Laplace a duas dimensões.

Considerando que, na formulação do MFS, as funções de base formam uma base que gera o espaço funcional das aproximações do problema, e podendo esse mesmo espaço ser descrito por outras bases, o MFS-SVD explora exactamente esta ideia, propondo uma mudança de base para obter um novo conjunto de funções de base, mas que resulte em um sistema linear melhor condicionado.

A abordagem utilizada em [12] emprega coordenadas polares e polinómios harmónicos para definir a nova base, partindo de uma representação espectral da solução fundamental, e utiliza a decomposição em valores singulares para realizar a mudança de base. As ilustrações apresentadas em [12] mostram que o método é extremamente eficaz no caso da equação de Laplace a duas dimensões, apresentando números de condição de ordem $O(1)$. No entanto, a afirmação de que o MFS-SVD funcionaria para a generalidade dos problemas de valor de fronteira carece de verificação caso a caso. Nesse sentido, nesta tese estendemos a aplicação do MFS-SVD para dois problemas de valor de fronteira no plano: O problema da equação de Helmholtz e o problema biharmónico.

Para o problema da equação de Helmholtz, ilustramos neste trabalho que o MFS-SVD também é muito eficaz no tratamento do mau condicionamento quando o domínio considerado é o disco, utilizando, nesse caso, coordenadas polares e funções de Bessel como funções de base. Para qualquer outro domínio diferente do disco, mostramos que algumas melhorias são observadas, mas, quando aumentamos o número de funções de base, volta-se a observar um crescimento do número de condição para níveis semelhantes aos do caso clássico. Ou seja, a mudança de base, nestes casos, não garante a ortogonalidade das novas funções de base como ocorre no caso do disco, sugerindo a necessidade de se utilizar um sistema de coordenadas curvilíneo adaptado ao domínio em estudo para representar a expansão do teorema da adição. Para ilustrar/verificar essa hipótese, sugerimos nesta tese a resolução de problemas com domínio delimitado por uma elipse ou outros próximos à elipse, utilizando a expansão da solução fundamental em coordenadas elípticas, com funções de Mathieu como funções de base. Com essa abordagem, o condicionamento do sistema linear do MFS melhora consideravelmente para os casos com domínio elíptico e para domínios que apresentem um alongamento em uma determinada direcção, como por exemplo, domínios limitados do tipo amendoím.

Essa abordagem com funções de Mathieu foi desafiadora, pois não existe implementação oficial dessas funções no software Matlab. Por isso, recorremos a ferramentas de terceiros [32, 33], que se mostraram ineficientes, pois garantem apenas precisão simples. Como, para as nossas ilustrações, necessitávamos maior precisão, dada a natureza dos problemas que estamos resolvendo, foi feita uma adaptação ao código de Cojocaru utilizando ideias provenientes de [21], de forma a

garantir o cálculo das funções de Mathieu com maior precisão.

A alteração mostrou-se extremamente eficaz sendo, portanto, um outro aspecto relevante desta pesquisa. O código alterado poderá servir de base para a utilização das funções de Mathieu em outros contextos e problemas, algo que, actualmente provavelmente não ocorre, pois os códigos existentes apresentam baixa precisão devido aos erros subtractivos frequentemente associados à elas.

A equação biharmónica é abordada nesta tese por se tratar de uma equação diferencial de ordem superior, mais especificamente, de quarta ordem, e por ser um caso que apresenta diversas aproximações numéricas bem estabelecidas na literatura para a obtenção da solução via MFS. A busca pela solução numérica utilizando o MFS-SVD para este problema serve como ilustração da metodologia geral a ser seguida na resolução de problemas de valor de fronteira (BVPs) mais abrangentes.

Esta abordagem com o problema biharmónico é relevante pois para os casos de equações de segunda ordem, como as de Laplace e Helmholtz, a aplicação do MFS-SVD sugeria que a decomposição em valores singulares, SVD, deveria ser aplicada ao fator que contém o mau condicionamento, após a decomposição da matriz completa do sistema linear em fatores que dependem separadamente dos pontos de avaliação e dos pontos de observação. Ora, uma abordagem nesse sentido para um problema de quarta ordem, como o biharmónico, conduz a uma decomposição contendo matrizes singulares e, portanto, mal condicionadas, o que é ilustrado na tese. Portanto, ilustramos neste trabalho o procedimento a ser seguido para problemas de ordem superior, que reduz os casos Laplace e Helmholtz como casos particulares. Assim, este trabalho contribui para a formalização de um algoritmo aplicável à diferentes problemas de valor de fronteira, com as devidas adaptações.

Como mencionamos anteriormente, o facto de existirem diferentes representações numéricas para a aproximação da solução numérica do problema biharmónico é outra razão que nos levou a seleccioná-lo como um caso de implementação do MFS-SVD, pois pretendíamos ilustrar também que a performance do MFS-SVD é independente da representação usada para a obtenção da solução numérica.

Portanto esta tese ilustra que o MFS-SVD introduzido originalmente para o melhoramento do mau condicionamento da matriz do sistema linear do MFS da equação de Laplace, pode ser extendido aos problemas de valor de fronteira no plano, para a equação de Helmholtz e para a equação biharmónica. A relevância do que se mostra na tese reside no facto de que a abordagem tal como sugerida em [12] não se aplicar para a globalidade dos casos, sendo necessário as alterações sugeridas neste trabalho para se conseguir as melhorias no mau condicionamento do sistema linear do MFS, particularmente para problemas de ordem superior. Ilustramos também, que por vezes a melhoria do mau condicionamento poderá ser conseguida apenas utilizando um sistema de coordenadas curvilíneo adaptado ao domínio, e que a localização dos pontos de observação não é determinante para o mau condicionamento se a base for adequada.

Palavras-chave: Método das soluções fundamentais, Equação de Helmholtz, Equação biharmónica, MFS-SVD, Mau condicionamento, Funções de Mathieu.

Contents

List of Tables

List of Figures

Acronyms

BC boundary condition.

BVP boundary value problem.

FDM finite difference method.

FEM finite element method.

MFS method of fundamental solutions.

MFS-DIR direct MFS.

MFS-QR method of fundamental solutions with QR decomposition.

MFS-SVD method of fundamental solutions with singular value decomposition.

MFS-SVD$_E$ MFS-SVD in elliptical coordinates.

MFS-SVD$_P$ MFS-SVD in polar coordinates.

MMFS modified method of fundamental solutions.

MTM modified Trefftz method.

PDE partial differential equation.

RBF radial basis functions.

SVD singular value decomposition.

TSVD truncated singular value decomposition.

Chapter 1

Introduction

There is currently great interest in mathematical models as tools for decision makers in several areas of human activity. Physical, social, financial, and others phenomena are modelled using partial differential equations (PDEs). Some examples of cases modelled by PDEs include problems in areas such as electromagnetism [34], fluid mechanics [77], elasticity [36, 60, 77, 84], thermodynamics [60, 77], quantum mechanics [78], finance [49], neurosciences [24], optics and others [16, 34]. The deduction of the PDE that governs the problem under study is just an initial step in the modelling process. Once the PDE has been deducted and the boundary conditions (BCs) have been identified, it is necessary to solve the problem so that we can recover from it, the information we are looking for. However, finding the exact solution using analytical methods is not always trivial or possible for certain type of boundary value problems (BVPs); in a situation such as this, it is better to consider the use of numerical schemes to obtain an approximate numerical solution to the problem.

Traditionally, numerical methods based on domain discretization like finite difference method (FDM) or finite element method (FEM) are among the most used to find a numerical solution to a BVP. However, for domains with complicated shapes, these standard numerical methods may require a large number of nodes to better discretize the domain, which may become a highly time consuming task. Due to this, a new family of a meshfree methods emerged, relying the solution of the BVP only on nodes at the boundary. Some examples of meshfree methods are the radial basis functions (RBF) and the method of fundamental solutions (MFS), the one in which we are interested.

The advantages of the MFS over the domain discretization methods is well documented; beyond the fact of being meshless, it has an easy numerical implementation and presents exponential convergence in some cases, specially in analytic domains. However, it is known that the MFS can become highly ill-conditioned for different reasons. Among others, we have the increase of the number of its basis functions in the linear combination expressing the numerical approximation, or simply due to the location of the sources themselves, as some of the causes. The ill-conditioning represents a problem for the accuracy and stability of the solution. Therefore, one

needs to consider application of regularization techniques to minimize it. Currently, this topic remains an object of research, the main goal being the proposal of new techniques which are efficient without adding complexity or without changing the MFS solution. Recent works in this direction can be found in [9, 10, 12–14, 29]. Still regarding MFS, despite several years since its introduction, research is also being carried out in other directions, such as the search for new representations [26, 45] and the extension to inhomogeneous boundary value problems [8, 52].

In this thesis, we address the ill-conditioning of the MFS linear system for homogeneous Helmholtz and biharmonic BVP in planar domains. In fact, we propose to extend a relatively new technique, named method of fundamental solutions with singular value decomposition (MFS-SVD), introduced in [12] to reduce the ill-conditioning of Dirichlet BVP for the Laplace equation in 2D. This technique consists in changing the basis of the classical MFS numerical approximation to another one spanning the same space of approximation for the solution, leading to a better conditioned linear system. This is a spectral approach, as it proposes to approximate the solution of the PDE by a finite sum of orthogonal basis functions. To achieve this, the algorithm relies on the existence of an expansion for the fundamental solution, addition theorem, which in the best case is an expansion of orthogonal functions. This approach was found to be very effective for the Laplace equation, and we illustrate in this work that it is effective as well for Helmholtz and biharmornic problems with the necessary adjustments as they are non-trivial extensions.

1.1 Notation

In this work, we adopt the following notation conventions for clarity and consistency:

- **Scalars** are denoted by lowercase italic letters with or without subscript (e.g., a, or a_i).
- **Variables** are denoted by lowercase italic letters or Greek letters with or without subscript (e.g., x, x_i, ρ, θ). When the variables refers to the components of an element from $\mathbb{R}^2$ we denote by (x,y).
- **Vectors** are represented by bold lowercase letters (e.g., **v**, **w**). Their components are written as $(\mathsf{v}_1,\mathsf{v}_2)$.
- **Matrices** are denoted by bold uppercase letters (e.g., **A**, **B**). The entry in the i-th row and j-th column of matrix **A** is written as a_{ij}.
- **Sets** are denoted by uppercase letters (e.g., A) and standard sets such as the real numbers and complex are denoted by blackboard bold letters (e.g., $\mathbb{R}$, $\mathbb{C}$). The number of elements of a given finite set is represented by uppercase italic letters (e.g., M, N).
- **Operators** when is a generic one they are represented by uppercase calligraphic letters (e.g., $\mathfrak{L}$, $\mathfrak{B}$).

- **Functions** the image of an element $x \in$ X under f is denoted by $f(x)$.
- **Open domains** are represented by uppercase Greek letters (e.g., Ω).
- The **norm** of a vector **v** is written as $\|\mathbf{v}\|$.

Additional notation will be introduced as needed throughout the text for ease of reference.

1.2 Structure of the work

The thesis is subdivided into five chapters. In chapter 2, the foundations for the MFS are presented as well as the formulation of the approximation through MFS for Helmholtz and biharmonic problems; still in chapter 2, a description of the state of the art regarding strategies to treat ill-conditioning in MFS are presented. Chapter 3 is dedicated to the presentation of MFS-SVD, the approach being extended as a technique to reduce the ill-conditioning of the Helmholtz and biharmonic BVP. Chapter 4 is devoted to illustrations of the applicability of MFS-SVD for both problems. In chapter 5, some conclusions are drawn and some perspectives are shared regarding the evolution of the approach.

Two papers have been prepared with the results of this research, one published in mid-2024 [13] and another one at end 2025 [15].

Chapter 2

The MFS for linear partial differential equations

The MFS is a meshless numerical method introduced more than half a century ago by Kupradze and Aleksidze [59], which is quite often used to solve BVP for PDE, whose fundamental solution is known. The interest of researchers in this method is mainly due to its simple implementation and the level of accuracy that can be achieved, specially in analytic domains [17]. The method is becoming very popular as we can see from different applications in areas such as fluid mechanics [6, 54], elasticity [20, 67], acoustics [4, 17], electromagnetism [27, 86], stationary heat conduction [19] and option pricing [85].

In this chapter, we introduce the basics of the theory to solve a generic BVP by MFS, and we formulate the numerical schemes to approximate the solution of Helmholtz and biharmonic BVP in 2D domains using the classical MFS. This will set the starting point for our innovative contribution in the following chapters.

2.1 MFS general formulation

Let us consider the BVP

$$\begin{cases} \mathfrak{L}u(x) = 0, & x \in \Omega \subset \mathbb{R}^2, \\ \mathfrak{B}u(x) = g(x), & x \in \partial\Omega, \end{cases} \tag{2.1}$$

where $\mathfrak{L}$ is a linear partial differential operator with constant coefficients, Ω is an open bounded domain with boundary $\partial\Omega$, $g(x)$ is a known function and $\mathfrak{B}$ is the boundary condition operator. According to [22, 74] the problem (2.1) can be solved approximately provided there is a known set of solutions to $\mathfrak{L}u(x) = 0$ in Ω, dense in the space of all solutions. An approximation to the solution of problem (2.1) can be given by a boundary method using the following definition from [31, 74].

Definition 2.1 (Trefftz method)**.** *A numerical approximation to the solution of* (2.1) *is a function of the form*

$$u_N(x) = \sum_{j=1}^{N} c_j \varphi_j(x), \tag{2.2}$$

where $\{\varphi_j(x) | \ j = 1,2,...,N\}$ *is a linearly independent set of particular solutions of the equation* $\mathfrak{L}u(x) = 0$ *at every point of* Ω*. The coefficients* $C = \{c_j | \ j = 1,2,...,N\}$ *are determined in order to approximate the boundary data and therefore, the best approximation* $\bar{u}$ *is defined by*

$$||\mathfrak{B}\bar{u} - g|| = \min_{C} ||\mathfrak{B}u_N - g||,$$

where the norm $||\cdot||$ *is defined on* $\partial\Omega$*.*

Different algorithms can be derived depending on the choice of $\varphi(x)$. From [11] we have the following definition.

Definition 2.2 (Fundamental solution)**.** *Let* $\mathfrak{L}$ *be a linear partial differential operator.* $\Phi(x)$ *is said to be a fundamental solution of the equation* $\mathfrak{L}u = 0$ *if*

$$\mathfrak{L}\Phi = -\delta,$$

where δ *is the Dirac delta distribution.*

In [41] we have the proof of the following theorem.

Theorem 2.1 (Malgrange-Ehrenpreis)**.** *Every linear differential operator* $\mathfrak{L}$ *with constant coefficients has a fundamental solution.*

Let us now introduce the method of fundamental solutions (MFS).

Definition 2.3 (Method of Fundamental Solutions)**.** *A numerical approximation to the solution of* (2.1) *is a function of the form*

$$u_N(x) = \sum_{j=1}^{N} a_j \Phi(x - y_j), \tag{2.3}$$

where $\Phi(x)$ *is a fundamental solution of* $\mathfrak{L}$ *at every point of* Ω *with* $y_j \notin \bar{\Omega}, j = 1,2,...,N$*.*

The coefficients a_j are determined following the same ideas of definition 2.1. The y_j are normally referred to in the bibliography, as the sources points, the charge points or the singularities and the curve/surface from which they are selected is referred to as pseudo-boundary, artificial boundary, fictitious boundary and others. In

this work we adopt the term source points for y_j and pseudo-boundary if we want to refer to the curve described by the points y_j.

Note that definition 2.1 imposes that $\varphi_j(x)$ should be a linearly independent set of functions, which is important for the uniqueness of the numerical solution. However, in MFS, as we increase the number of basis functions to achieve higher accuracy, two or more elements $\Phi(x-y_j)$ may become very similar, i.e., almost linearly dependent. This generates issues on the accuracy of the MFS.

2.1.1 Applicability

The MFS is frequently described as fitting to solve only homogeneous elliptic partial differential equations. In fact, boundary methods such as the MFS are well established for those kind of PDEs but applications to parabolic and hyperbolic PDEs are also known, see for example [42]. MFS uses fundamental solution to build the approximation therefore, its implementation is heavily dependent on the existence of a fundamental solution for the differential equation, which exists in case of differential operator with constant coefficients according to theorem 2.1. In most theoretical developments, the applicability of the MFS is predicated on the assumption that the underlying domain is sufficiently regular, bounded, and, in many formulations, simply connected. However, applications to unbounded or non simply connected domains are also known.

Whenever the MFS is applicable it is easy to implement and in particular, for 2D domains close to a disk it shows higher convergence when compared to other methods such as FEM or FDM.

2.1.2 Error and stability

Although the MFS was introduced nearly 60 years ago, and a high amount of publication on its use and on computational results is available in different application contexts, according to Zi-Cai Li et al. published in 2023 [64] there exist only a few papers concerning its theoretical analysis in terms of stability, error estimates and convergence analysis. In fact we can quote them as follows: "*For the MFS, the theory is much behind the success of computation*".

In general, estimation of the theoretical error of the MFS approximation by means of sums (2.3) is normally produced case by case, it means it is dependent on the problem, the boundary conditions, the domain, the norm and on the selection of y_j. However, numerical experiments with different problems reveals that:

- If the solution u to a BVP possesses singularities of any kind, no sum (2.3) can assure reasonable approximation to u if the y_j are chosen far away from the boundary since the functions (2.3) are evidently all real analytic everywhere except at the singularities y_j [22];
- For well-behaved boundary conditions, error decreases exponentially as we increase the distance of the pseudo-boundary or as we increase the number of source points [31];

- The conditioning of the problem, which is a measure of numerical stability, deteriorates (grows exponentially) as the distance between the pseudo-boundary and the boundary of the domain increases or with increasing number of source points [82];
- Large traditional condition number may not represent issues for the accuracy of the solution, good results can still be obtained [31, 64];
- Careful choices of source nodes are an important issue in both convergence and stability [64].

2.2 The MFS formulation for the Helmholtz equation

2.2.1 The Helmholtz equation

The Helmholtz equation arises from different problems; we can mention, for example, the eigenvalue problem for the Laplace operator or solving the wave equation by separation of variables assuming time harmonic solutions. These are just a few examples of problems leading to the Helmholtz equation.

For illustration we derive the Helmholtz equation considering the propagation of acoustic waves of small amplitude in a homogeneous isotropic medium in $\mathbb{R}^2$ viewed as an inviscid fluid [34]. If we consider $\mathbf{v} = \mathbf{v}(x,t)$ the velocity field, $p = p(x,t)$ the pressure, $\rho = \rho(x,t)$ the density and $S = S(x,t)$ the specific entropy of the fluid, then, the motion of the acoustic wave is governed by the following laws

$$\begin{cases} \dfrac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v} + \dfrac{1}{\rho}\nabla p = 0, & \text{(Euler's equation)} \\ \dfrac{\partial \rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0, & \text{(the continuity equation)} \\ p = f(\rho, S), & \text{(the state equation)} \\ \dfrac{\partial S}{\partial t} + \mathbf{v}\cdot\nabla S = 0, & \text{(the adiabatic hypothesis)} \end{cases}$$

where the function f depends on the nature of the fluid. Assuming that $\mathbf{v}, p, \rho$ and S are small perturbations of the static state $\mathbf{v}_0 = 0$ and constants p_0, ρ_0, and S_0, by linearization, we obtain

$$\begin{cases} \dfrac{\partial \mathbf{v}}{\partial t} + \dfrac{1}{\rho_0}\nabla p = 0, & \text{(linearized Euler's equation)} \\ \dfrac{\partial \rho}{\partial t} + \rho_0\nabla\cdot\mathbf{v} = 0, & \text{(the linearized continuity equation)} \\ \dfrac{\partial p}{\partial t} = \dfrac{\partial f}{\partial \rho}(\rho_0, S_0)\dfrac{\partial \rho}{\partial t}, & \text{(the linearized state equation).} \end{cases}$$

By putting all together, from the system above, we obtain the wave equation

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = \Delta p,$$

where c is the speed of sound defined by

$$c^2 = \frac{\partial f}{\partial \rho}(\rho_0, S_0).$$

From the linearized Euler equation, we observe that there exists a velocity potential $U = U(x,t)$ such that

$$\mathbf{v} = \frac{1}{\rho_0}\nabla U, \quad p = -\frac{\partial U}{\partial t}.$$

The velocity potential U also satisfies the wave equation

$$\frac{1}{c^2}\frac{\partial^2 U}{\partial t^2} = \Delta U.$$

For time-harmonic acoustic waves of the form

$$U(x,t) = \mathsf{Re}\{u(x)e^{-\mathsf{i}\omega t}\}$$

with frequency $\omega > 0$ and $\mathsf{i} = \sqrt{-1}$, we deduce that the complex valued space dependent part u satisfies the Helmholtz equation

$$\Delta u + k^2 u = 0,$$

where $k = \omega/c$ is the wave number.

In order to ensure the well posedness of the problem, boundary conditions should be defined depending on the problem being solved. In this work, we consider the interior Helmholtz equation with Dirichlet boundary conditions in the plane.

2.2.2 The Helmholtz equation: MFS formulation

Let Ω be a bounded and smooth domain in $\mathbb{R}^2$, i.e. of class C^1, whose boundary is denoted by $\partial\Omega$. We consider the following interior Dirichlet BVP for the Helmholtz equation

$$\begin{cases} \Delta u + k^2 u = 0 & \text{in} \quad \Omega, \\ u = g & \text{on} \quad \partial\Omega, \end{cases} \tag{2.4}$$

for the unknown field u, with a given function $g \in H^{1/2}(\partial\Omega)$ and wave number $k > 0$.

In order to solve numerically this problem by MFS, we recall that the fundamental solution associated to Helmholtz operator in $\mathbb{R}^2$ is given by

$$\Phi(x) = \frac{\mathsf{i}}{4}H_0^{(1)}(k|x|), \quad x \neq 0, \tag{2.5}$$

where $H_0^{(1)}$ is the Hankel function of the first kind of order zero.

Let us assume that we have $\{y_j \in \partial\Omega^{'} \,|\, j = 1,2,...,N\}$, a set of points over a closed curve $\partial\Omega^{'}$ which is the boundary of a fixed bounded open domain $\Omega^{'}$ containing $\bar{\Omega}$.

An approximation u_N to the solution of the problem (2.4) can be expressed as in (2.3) [22, 38, 74, 82, 83]. As mentioned previously we call $\partial\Omega^{'}$ as the pseudo-boundary.

As in [9] we refer to the classical approach as Direct MFS, and write an approximation to the solution of our problem as

$$u_N^D(x) = \sum_{j=1}^{N} a_j^D \Phi(x - y_j), \quad x \in \bar{\Omega}, \quad y_j \in \partial\Omega^{'}, \tag{2.6}$$

for some coefficients $\{a_j^D \in \mathbb{C} |\ j = 1, 2, ..., N\}$. Due to properties of the fundamental solution, the representation (2.6) for the field u_N^D satisfies the Helmholtz equation in Ω and the approximation of the boundary condition by the same representation can be justified by density results, since

$$\textit{span}\left\{ \Phi(\bullet - y)|_\Omega : y \in \partial\Omega^{'} \right\}$$

is dense in $\{v \in H^1(\Omega) : (\Delta + k^2)v = 0\}$, with a topology from the Sobolev space H^1, [4].

The coefficients $\{a_1^D, a_2^D, ..., a_N^D\}$ can be determined by collocation, by forcing the boundary condition to be satisfied at some points $\{x_i \in \partial\Omega |\ i = 1, 2, ..., M\}$. By doing so, the problem is then reduced to solving a linear system for the coefficients $\mathbf{a}^D$, which for $M > N$ is an overdetermined system that can be solved in the least squares sense, written as

$$\mathbf{A}^D \mathbf{a}^D = \mathbf{G}, \tag{2.7}$$

where $\mathbf{A}^D$, $\mathbf{a}^D$ and $\mathbf{G}$ are matrices of dimensions $M \times N$, $N \times 1$ and $M \times 1$ respectively, with entries

$$\mathbf{A}^D = \Big(\Phi(x_i - y_j)\Big)_{1 \le i \le M,\, 1 \le j \le N}, \quad \mathbf{a}^D = \big(a_j^D\big)^{\mathsf{T}}_{1 \le j \le N}, \quad \mathbf{G} = \Big(g(x_i)\Big)^{\mathsf{T}}_{1 \le i \le M}.$$

2.2.3 MFS for Helmholtz equation: Error and stability

A numerically stable representation of an approximate solution to the Helmholtz problem depends on how far into the complex plane a solution of (2.4) can be analytically continued [17].

Following [17, 61], the convergence rate for an interior Helmholtz problem can be estimated using the boundary norm. Let us assume that we have a L_2 norm defined on $\bar{\Omega}$ then, the interior error of the solution for a nonresonant k may be bounded by the boundary norm as follows

$$||u_N^D - u||_{L^2(\Omega)} \le \frac{C_\Omega}{d} ||u_N^D - g||_{L^2(\partial\Omega)},$$

where $d := \min_{\mathrm{j}} ||k^2 - E_j||/E_j$, with E_j being the domain's Dirichlet eigenvalues and C_Ω a domain dependent constant. In fact, in case of the unitary disk with the pseudo-boundary $\partial\Omega'$, being a circumference of radius R and also with $r_s > 1$ denoting the radius of the closest singularity of the analytic continuation of g we have the following result from [17]:

Theorem 2.2. *Let $R > 1$ and N be even. For analytic boundary data g with Fourier coefficients of g showing asymptotically exponential decay the minimum boundary error achievable with the MFS satisfies*

$$||u_N^D - g||_{L^2(\partial\Omega)} \leq \begin{cases} C_e r_s^{-N/2}, & r_s < R^2 \\ C_e \sqrt{N} R^{-N}, & r_s = R^2 \\ C_e R^{-N}, & r_s > R^2, \end{cases}$$

where each time C_e means a different constant which may depend on k, R and g but not N. Furthermore if g is analytically continuable to an entire function, the last of the three cases holds for any $R > 1$.

Proof. See [17]. □

The previous theorem holds for a unitary disk with a pseudo-boundary as a circumference of radius R. In case of an arbitrary analytic domain with pseudo-boundary being in a conformal map with fixed conformal radius $\rho_z = R$ with points being conformal equi-angular distributed, then the following conjecture is proposed by [17]

$$||u_N^D - g||_{L^2(\partial\Omega)} \leq \begin{cases} C_e r^{-N/2}, & r_s < R^2, \\ C_e R^{-N}, & r_s > R^2, \end{cases}$$

where C_e is a constant that may depend on Ω, k, R and g but not N and $r_s > 1$ is the conformal radius of the closest singularity of the analytic continuation of u.

In the case of a disk or any other analytic domain it is demonstrated in [17] that the achievable accuracy is limited by the size of the coefficient norm $||\mathbf{a}^D||$.

In summary the numerical stability and hence higher accuracy of the MFS relies on the choice of a pseudo-boundary which does not enclose any singularities of the analytic continuation of the solution u.

2.3 The MFS formulation for the biharmonic equation

2.3.1 The biharmonic equation

The biharmonic equation can arise also from different modelling problems, such as in elasticity [46, 68], fluid mechanics [77], and others. In this work we select a problem from fluid dynamics, the Stokes flow problem, to illustrate the derivation of a biharmonic equation.

Let us consider the motion of a highly viscous incompressible fluid with single viscosity and a negligible inertia effects in $\mathbb{R}^2$. Assuming that no body forces are acting on the fluid and considering in the steady state the velocity field, $\mathbf{v} = \mathbf{v}(\mathrm{v}_1(x), \mathrm{v}_2(x))$,

the pressure $p = p(x)$, the dynamic viscosity of the fluid μ, then the motion of the fluid is governed by the following laws [77]

$$\begin{cases} \nabla \cdot \mathbf{v} = 0, & \text{(the continuity equation)} \\ \nabla p = \mu \Delta \mathbf{v}, & \text{(Stokes equation).} \end{cases}$$

By introducing the stream function $u(x)$, which are lines drawn tangent to the velocity vector at every point, then, the fluid stream describes the flow where,

$$\frac{\partial u}{\partial \mathsf{y}} = \mathsf{v}_1, \quad \frac{\partial u}{\partial \mathsf{x}} = -\mathsf{v}_2,$$

which for a 2D incompressible flow can be rewritten as [73]

$$\nabla u = (-\mathsf{v}_2, \mathsf{v}_1). \tag{2.8}$$

By applying the *divergence* to both sides of (2.8) we get

$$\Delta u = \nabla \cdot (-\mathsf{v}_2, \mathsf{v}_1) = -\frac{\partial \mathsf{v}_2}{\partial \mathsf{x}} + \frac{\partial \mathsf{v}_1}{\partial \mathsf{y}} = -\nabla \times \mathbf{v},$$

i.e.

$$\nabla \times \mathbf{v} = -\Delta u. \tag{2.9}$$

If we apply the *curl* to the Stokes equation, $\nabla p = \mu \Delta \mathbf{v}$, we get

$$\nabla \times (\nabla p) = \mu \Delta (\nabla \times \mathbf{v}).$$

Since the *curl* of a gradient is always zero, i.e. $\nabla \times (\nabla p) = 0$ we obtain

$$\mu \Delta (\nabla \times \mathbf{v}) = 0. \tag{2.10}$$

By substituting (2.9) into (2.10) we get

$$\mu \Delta (-\Delta u) = 0,$$

which simplifies to the biharmonic equation for the stream function

$$\Delta^2 u = 0,$$

meaning that the stream function u satisfies the biharmonic equation in the case of 2D Stokes flow. Here also, to have the problem well posed boundary conditions should be defined.

2.3.2 The biharmonic equation: MFS formulation

Let Ω be a bounded domain in $\mathbb{R}^2$ whose boundary is denoted by $\partial\Omega$. We consider the following biharmonic BVP

$$\begin{cases} \Delta^2 u = 0 & \text{in} \quad \Omega, \\ u = g_1 & \text{on} \quad \partial\Omega, \\ \dfrac{\partial u}{\partial \mathbf{n}} = g_2 & \text{on} \quad \partial\Omega, \end{cases} \tag{2.11}$$

where $\mathbf{n}$ is the outward unit normal vector to the boundary $\partial\Omega$, $g_1 \in H^{3/2}(\partial\Omega)$, $g_2 \in H^{1/2}(\partial\Omega)$ are known. The discussion about existence and uniqueness of the solution for problem (2.11) can be found in several papers, as for example [57, 58].

Let us now address the numerical approximation of the solution of problem (2.11) by MFS. This is normally obtained by using a single layer potential representation [53], which leads to an approximation of the form

$$u^D_{N,1}(x) = \sum_{j=1}^{N} a^{(1)}_{j,1}\Phi_1(x-y_j) + \sum_{j=1}^{N} a^{(2)}_{j,1}\Phi_2(x-y_j), \quad x \in \bar{\Omega}, \quad y_j \in \partial\Omega', \tag{2.12}$$

with $\Phi_1(x) = 1/(2\pi)\log|x|$, $\Phi_2(x) = 1/(8\pi)|x|^2\log|x|$ being the fundamental solutions associated to the operators Δ and Δ^2, respectively, and $\partial\Omega'$ being a closed curve, boundary of a bounded domain containing $\bar{\Omega}$.

Recently, a novel approach with a formulation based on the extension of the auxiliary boundary biharmonic layer representation was proposed in [26]; it looks for numerical solution of the form,

$$u^D_{N,2}(x) = \sum_{j=1}^{N} a^{(1)}_{j,2}\Phi_2(x-y_j) + \sum_{j=1}^{N} a^{(2)}_{j,2}\frac{\partial\Phi_2(x-y_j)}{\partial\mathbf{n}(y_j)} \quad x \in \bar{\Omega}, \quad y_j \in \partial\Omega', \tag{2.13}$$

with $\mathbf{n}(y)$ being the outward unit normal vector to the pseudo-boundary $\partial\Omega'$. We will refer to the approximations (2.12) and (2.13) as the first and second approximations and denote them as approximations A_1 and A_2, respectively.

Due to its definition, both (2.12) and (2.13) satisfy the PDE in (2.11). We need to guarantee additionally that both fit also the boundary conditions, which will allow us to find the coefficients $a^{(1)}_{j,\bullet}$ and $a^{(2)}_{j,\bullet}$.

Let $\{x_i \in \partial\Omega \,|\, i = 1,2,...,M\}$ be a set of points from $\partial\Omega$ and $\{y_j \in \partial\Omega' \,|\, j = 1,2,...,N\}$ a set of points from $\partial\Omega'$ with $M > N$; by imposing the boundary conditions from (2.11) one gets, for each approximation

$$\begin{matrix} u^D_{N,p}(x_i) = g_1(x_i) \\ \dfrac{\partial u^D_{N,p}(x_i)}{\partial\mathbf{n}} = g_2(x_i) \end{matrix}, \quad i = 1,2,...,M,\ p = 1,2, \tag{2.14}$$

from which we obtain a $2M \times 2N$ collocation linear system

$$\begin{bmatrix} \mathbf{N} & \mathbf{P} \\ \dfrac{\partial}{\partial\mathbf{n}}\mathbf{N} & \dfrac{\partial}{\partial\mathbf{n}}\mathbf{P} \end{bmatrix} \mathbf{a} = \begin{bmatrix} \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix}, \tag{2.15}$$

where $\mathbf{N}, \mathbf{P}, \mathbf{a}, \mathbf{G}_1, \mathbf{G}_2$ are matrices with dimensions $M \times N$, $M \times N$, $2N \times 1$, $M \times 1$, $M \times 1$, respectively. In case of ansatz (2.12), the matrices are given by

$$\mathbf{N} = \Big(\Phi_1(x_i - y_j)\Big)_{1\le i\le M, 1\le j\le N}, \quad \mathbf{P} = \Big(\Phi_2(x_i - y_j)\Big)_{1\le i\le M, 1\le j\le N},$$
$$\mathbf{a} = \Big(\big(a_{1,j}^{(1)}\big)\ \big(a_{1,j}^{(2)}\big)\Big)^{\mathsf{T}}_{1\le j\le N},$$

while for ansatz (2.13) **N**, **P** and **a** are given by

$$\mathbf{N} = \Big(\Phi_2(x_i - y_j)\Big)_{1\le i\le M, 1\le j\le N}, \quad \mathbf{P} = \left(\frac{\partial \Phi_2(x_i - y_j)}{\partial \mathbf{n}(y_j)}\right)_{1\le i\le M, 1\le j\le N},$$
$$\mathbf{a} = \Big(\big(a_{2,j}^{(1)}\big)\ \big(a_{2,j}^{(2)}\big)\Big)^{\mathsf{T}}_{1\le j\le N},$$

with

$$\mathbf{G}_1 = \Big(g_1(x_i)\Big)^{\mathsf{T}}_{1\le i\le M}, \quad \mathbf{G}_2 = \Big(g_2(x_i)\Big)^{\mathsf{T}}_{1\le i\le M}.$$

2.3.3 MFS for biharmonic equation: Error and stability

As previously mentioned the theoretical study of the error and stability of numerical solution of a BVP is dependent on several factors. For the biharmonic problem one can find bounds estimation for the error and condition number for the problem using Almansi fundamental solution [62] and we can also find error bounds [35] for problems using the fundamental solution (2.12), the one being used in this work. Let us denote

$$< u, v >_H = \int_{\partial\Omega} uv + \frac{1}{N^2}\int_{\partial\Omega} \frac{\partial u}{\partial \mathbf{n}}\frac{\partial v}{\partial \mathbf{n}}$$

with the norm given by $||v|| = \sqrt{< v, v >_H}$. From [35] we have the following results:

Theorem 2.3. *Suppose that $u \in H^p(\Omega)$, $p \ge 5/2$ and N is chosen such that*

$$\left(\frac{R}{r_{max}}\right)^{2n-N}\left(\frac{r_{max}}{r_{min}}\right)^{n} \le \frac{1}{n^{p+1/2}}$$

then there exists the error bound

$$||u - u_{N,1}^D||_H = O\left(\frac{1}{N^{p-1/2}}\right)||u||_{p,\Omega}$$

where $r_{max} = max_\Omega r$ and r_{min} is the maximal radius of the disk $\Omega^+ \subseteq \Omega$, C is a constant not dependent of N and n is the degree of the harmonic polynomials being used to approximate the biharmonic solutions.

Proof. See [35]. □

Regarding the stability, there is from [35] the following result for a circular domain:

Theorem 2.4. *If $r < R$ and $N \gg 1$ is even, with $R \neq 1$ and $R \neq 1/e$, one has the asymptotic behaviour for the condition number*

$$\mathit{Cond} = O\left(N^3 \left(\frac{R}{r}\right)^{N/2}\right).$$

When N is odd we have Cond $= O(N^3(R/r)^{[N/2]})$, where $[N/2]$ is the maximal integer $\leq N/2$.

Normally, the MFS approach introduced for Helmholtz and biharmonic BVPs, is highly accurate for analytic domains, but the linear systems (2.7) and (2.15), solved in the least squares sense, are often severely ill-conditioned requiring strategies to deal with the ill-conditioning. Therefore, we will address the state of the art regarding this issue in the next section.

2.4 State of the art of strategies to address the ill-conditioning in MFS

As mentioned previously, the MFS can be highly accurate, but its accuracy is sometimes compromised as the linear system is often highly ill-conditioned. For this reason, regularization techniques are normally required to obtain a stable approximation to the solution. There exist several strategies to mitigate ill-conditioning. Among the most used today, we have the following.

2.4.1 Regularization schemes

Let us suppose that the linear operator $\mathbf{A} : X \to Y$ is injective. We would like to approximate the solution $\mathbf{x}$ to the ill-conditioned equation of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{2.16}$$

As we want to find a stable approximation, i.e. the approximation results must depend continuously on the actual data then, the idea is to find a family of bounded linear operators $\mathbf{R}_\alpha$, $\alpha > 0$ to approximate the unbounded linear operator $\mathbf{A}^{-1}$. Once the family is identified, a strategy is needed to choose the regularization parameter α, which must be selected by keeping the compromise between accuracy and stability. One of the most known and most used scheme in this class, is the *Tikhonov regularization* where the family of bounded operators $\mathbf{R}_\alpha$ is defined as [34]

$$\mathbf{R}_\alpha := (\alpha \mathbf{I} + \mathbf{A}^* \mathbf{A})^{-1} \mathbf{A}^*,$$

where $\mathbf{A}^*$ is the adjoint operator of $\mathbf{A}$. This means that instead of (2.16) we look for the unique solution of

$$\alpha \mathbf{x}_\alpha + \mathbf{A}^* \mathbf{A} \mathbf{x}_\alpha = \mathbf{A}^* \mathbf{b}, \tag{2.17}$$

which is equivalent to minimizing

$$||\mathbf{A}\mathbf{x}_\alpha - \mathbf{b}||^2 + \alpha ||\mathbf{x}_\alpha||^2$$

in terms of $||\mathbf{x}_\alpha||$. As we can see from (2.17) the problem being solved is not exactly the same as the original one in (2.16). In the new formulation the stabilization is achieved by the introduction of a penalty term, $\alpha||\mathbf{x}_\alpha||^2$. The challenge is to select an optimal α because if it is large it can modify significantly the original problem; on the other side if it is too small the norm of the solution is not penalised and the problem becomes closer to the original problem which we know yields unstable approximations that we want to avoid. So, although it is one of the most used methods, the main drawback is the selection of parameter α. Some examples of use of *Tikhonov regularization* to address ill-conditioned problems in MFS can be found in [48, 72, 89]. In [48] this strategy is applied for the multidimensional inverse heat conduction problem with parameter being selected using a cross-validation criteria. In [72] the ill-conditioning of MFS for the biharmonic equation is addressed. The selection of the parameter is based on the L-curve method. In [89] the method is applied to the biharmonic equation also; the optimal regularization parameter that ensures the minimum condition number is proposed as $\alpha = \sigma_{min}\sigma_{max}$ with σ_{min} and σ_{max} being the minimal and maximal singular values of matrix $\mathbf{A}$, respectively. For other examples of using *Tikhonov regularization* for MFS problems see also [25, 70, 71], and for existing strategies to select α, see [25, 55].

In this class of regularization approaches, there is also the *spectral cut-off*. Assuming that $\mathbf{A} : X \to Y$ is a compact linear operator and X is an Hilbert space then, the solution of (2.16), is given by

$$\mathbf{x} = \sum_{i=1}^{\infty} \frac{1}{\sigma_i} (\mathbf{b}, \mathbf{v}_i) \mathbf{u}_i,$$

where $(\sigma_i, \mathbf{u}_i, \mathbf{v}_i)$ is a singular system for the operator $\mathbf{A}$, i.e. $\mathbf{A}\mathbf{u}_i = \sigma_i \mathbf{v}_i$, $\mathbf{A}^*\mathbf{v}_i = \sigma_i \mathbf{u}_i$ with σ_i being the singular values of $\mathbf{A}$. The idea of the scheme is to try to regularize the equation by damping the influence of the factor $1/\sigma_i$ in the solution of $\mathbf{x}$. This approach is normally described as truncated singular value decomposition (TSVD) and was applied in [40, 47, 50, 69, 76, 89] for different BVPs. As for *Tikhonov regularization*, the fact that we discard some singular values means that we are approximating $\mathbf{A}$ in (2.16), by a new one with a finite rank which does not describe entirely the original space. We are therefore, introducing an error whose magnitude depends on the singular values discarded.

2.4.2 Optimal placement of source points

The choice of source points in the MFS is identified by some authors as one of the causes of poor conditioning. Therefore, some authors explore this to mitigate it, like

in [17] where strategies are introduced to select the best location of the source points for the interior Helmholtz problem. In [29], it is proposed to use the effective condition number as a strategy to select the best location of the source points. On the same direction see also [3, 28, 51, 88]. These strategies may work well for a limited amount of source points. As we increase the number of source points these schemes may find a best location which is too close to the boundary of the domain and this can impact negatively the accuracy as MFS has singularities at the boundary.

2.4.3 Change of basis

2.4.3.1 Some of the existing approaches

In this approach the idea is to apply a suitable change of basis to a new one that spans the same functional space of approximations. Some of the existing algorithms are: The modified method of fundamental solutions (MMFS), the method of fundamental solutions with QR decomposition (MFS-QR) and method of fundamental solutions with singular value decomposition (MFS-SVD). The MMFS, for example, is applied in [66] for the Laplace equation in the plane. There, to alleviate the ill-conditioning an existing relation between the modified Trefftz method (MTM) and the MFS is explored to propose a new system of linear equations. In [9, 10, 14] a new basis for Laplace 2D, 3D and 2D Helmholtz problems, are generated using the MFS-QR approach.

2.4.3.2 The MFS-SVD

This approach was first proposed in 2022 by Antunes in [12]. There, the method is applied to the Laplace equation with Dirichlet BC in the plane. The new basis functions in polar coordinates, uses harmonic polynomials and is constructed combining SVD with an addition theorem for the fundamental solution of the Laplace operator. Since then, to the best of our knowledge, no additional work was published on the possibility to extend this approach to other BVP till 2024. In that year, in [13] an extension of the method to Helmholtz BVP in the plane was proposed. Two extensions were presented: One in polar coordinates using Bessel functions and another one in elliptic coordinates using Mathieu functions. In both cases, appropriate addition theorems for the fundamental solution of the Helmholtz operator are used in combination with SVD.

In 2025 two new papers were published on this topic. In [14] the extension to the Laplace BVP in 3D with Dirichlet BC; later in the same year, in [15] a first time extension to higher order BVP, the biharmonic equation in the plane, and also a first time comparison of two different MFS-SVD formulations was presented.

This thesis is a detailed presentation of the full research work from which publications [13, 15] originated.

2.4.4 Others

Other strategies are proposed within the literature. One of them is the *domain decomposition* [5, 18, 56]. In [56] for example, a combination between MFS and domain decomposition techniques is explored to control the ill-conditioning for the Laplace BVP. The idea is to write the Dirichlet BVP of Laplace equation as a problem in 2,3 or 4 subdomains of the original domain; another strategy proposed by some authors is *the regularization using radial and nearly radial basis functions* like in [43]. This approach uses nearly fundamental solutions and special fundamental solutions concentrated on lines instead of points; there is also *the multi-level technique* in [44] consisting in categorising the source nodes into groups to decrease the density of the spatial distribution of the source nodes when far from the boundary. Another approach for solving Laplace and biharmonic 2D problems was explored in [39]. It is designated as *localized method of fundamental solutions* and consists in representing each solution at each interior node as a linear combination of solutions at nearby nodes. In [65, 79, 81] techniques identified as *adaptive greedy techniques* are used. These techniques are data-dependent which adaptively selects the suitable source nodes based on a specific adaptive procedure.

Chapter 3

The MFS-SVD as a spectral approach to improve the conditioning of MFS

As mentioned in previous chapter, among the alternatives to tackle ill-conditioning there is an emerging new technique which consists in the change of the basis functions to a more adequate and closer to orthogonal. The MFS-SVD algorithm was introduced for the first time in [12] for the planar Dirichlet BVP of Laplace equation in which the new set of basis functions is described using harmonic polynomials.

This chapter addresses the innovative contribution we are proposing, to address the ill-conditioning of the Helmholtz and the biharmonic BVPs, by generalizing the MFS-SVD approach to these problems for 2D domains. The results have been published in [13] for the Helmholtz case and in [15] for the biharmonic. In the Helmholtz case the MFS-SVD is first presented in polar coordinates and after extended to domains defined in elliptic coordinates. We explore Mathieu functions to describe the new basis functions for problems involving elliptic coordinates, instead of the expansion in terms of Bessel functions used for polar coordinates. In the biharmonic case, the extension is made by using harmonic polynomials in polar coordinates, as in the Laplace case [12], and by considering two different representations for the numerical approximation.

3.1 The MFS-SVD formulation

In this section, we address the MFS-SVD generalization to problems (2.4) and (2.11) following the ideas introduced in [12] for the Laplace equation in 2D.

We consider the numerical approximation (2.6), (2.12), (2.13) and assume that the fundamental solution can be expressed in a given curvilinear system of coordinates through an appropriate addition theorem. For Helmholtz and biharmonic BVP in the plane, we have the following addition theorems.

Let (ρ, α) be the polar coordinates for a given $(\mathsf{x}, \mathsf{y}) \in \mathbb{R}^2$ where,

$$\mathsf{x} = \rho\cos(\alpha), \quad \mathsf{y} = \rho(\sin\alpha). \tag{3.1}$$

Theorem 3.1 (Graf addition theorem in polar coordinates for $H_0^{(1)}$)**.** *Let $x = (r_P, \theta)$, $y = (\rho_P, \alpha)$ be the polar coordinates of x, y. For $\rho_P > r_P$, there exists the expansion*

$$H_0^{(1)}(k|x-y|) = H_0^{(1)}(k\rho_P)J_0(kr_P) + 2\sum_{n=1}^{\infty} H_n^{(1)}(k\rho_P)J_n(kr_P)\big(\cos(n\theta)\cos(n\alpha) + \sin(n\theta)\sin(n\alpha)\big). \tag{3.2}$$

Proof. See [9, 34]. □

Suppose now, that instead of polar coordinates our domain is better described by an elliptic coordinates system which can be defined as in [63] by

$$\mathsf{x} = \sigma\cosh\eta\cos\nu, \quad \mathsf{y} = \sigma\sinh\eta\sin\nu, \tag{3.3}$$

where $\sigma > 0$, $0 \le \eta < \infty$ and $0 \le \nu \le 2\pi$. If $\eta = \eta_0$ from (3.3) we obtain the ellipse defined by

$$\frac{\mathsf{x}^2}{\sigma^2\cosh^2\eta_0} + \frac{\mathsf{y}^2}{\sigma^2\sinh^2\eta_0} = 1,$$

with semi-axes $a = \sigma\cosh\eta_0$ and $b = \sigma\sinh\eta_0$. When the two semi-axes of the ellipse are known then

$$\eta_0 = \tanh^{-1}\frac{b}{a}, \quad \sigma = \sqrt{a^2 - b^2}.$$

To transform the cartesian coordinates (x, y) to elliptic coordinates (η, ν) the following explicit transformations are used

$$\eta = \sinh^{-1}F(\mathsf{x}, \mathsf{y}, \sigma), \quad \nu = \cos^{-1}\left(\frac{\mathsf{x}}{\sigma\cosh\eta}\right), \tag{3.4}$$

with

$$F(\mathsf{x}, \mathsf{y}, \sigma) = \frac{1}{\sqrt{2}\sigma}\sqrt{(\mathsf{x}^2 + \mathsf{y}^2 - \sigma^2) + \sqrt{(\mathsf{x}^2 + \mathsf{y}^2 - \sigma^2)^2 + 4\sigma^2\mathsf{y}^2}}.$$

When $\mathsf{y} \ge 0$ from (3.4) we get $0 \le \nu \le \pi$ while if $\mathsf{y} < 0$ we get $\pi < \nu < 2\pi$, [87].

Let both $x \in \partial\Omega$ and $y \in \partial\Omega'$ be represented in the same elliptic coordinate system. We consider the elliptic coordinates for x and y given by $x = (r_E, \vartheta)$ and $y = (\rho_E, \omega)$. From [37, 75] we have the following result.

Theorem 3.2 (Addition theorem in elliptic coordinates for $H_0^{(1)}$)**.** *For $\rho_E > r_E$ there exists the expansion*

$$H_0^{(1)}(k|x-y|) = 4\sum_{n=0}^{\infty}\left(\frac{1}{Ne_n}Re_n^{(1)}(q, r_E)Re_n^{(3)}(q, \rho_E)Se_n(q, \vartheta)Se_n(q, \omega) + \frac{1}{No_n}Ro_n^{(1)}(q, r_E)Ro_n^{(3)}(q, \rho_E)So_n(q, \vartheta)So_n(q, \omega)\right), \tag{3.5}$$

where $q = k^2\sigma^2/4$, Se_n *and* So_n *are the even and odd angular Mathieu functions,* $\mathsf{Re}_n^{(1)}$, $\mathsf{Re}_n^{(3)}$ *the even radial Mathieu functions of first and third kind,* $\mathsf{Ro}_n^{(1)}$ *and* $\mathsf{Ro}_n^{(3)}$ *are the odd radial Mathieu functions of first and third kind, and* Ne_n *and* No_n *are the even and odd normalization coefficients, respectively, defined by [21, 33, 37]*

$$Nee_n = 2\pi\left(A_{ee}^{(0)}(q,n)\right)^2 + \pi\sum_{j=1}^{\infty}\left(A_{ee}^{(2j)}(q,n)\right)^2, \quad Neo_n = \pi\sum_{j=0}^{\infty}\left(A_{eo}^{(2j+1)}(q,n)\right)^2,$$

$$Noo_n = \pi\sum_{j=0}^{\infty}\left(A_{oo}^{(2j+1)}(q,n)\right)^2, \quad Noe_n = \pi\sum_{j=1}^{\infty}\left(A_{oe}^{(2j)}(q,n)\right)^2,$$

$$\mathsf{See}_n(q,u) = \sum_{j=0}^{\infty}A_{ee}^{(2j)}(q,n)\cos(2ju), \quad \mathsf{Seo}_n(q,u) = \sum_{j=0}^{\infty}A_{eo}^{(2j+1)}(q,n)\cos((2j+1)u),$$

$$\mathsf{Soe}_n(q,u) = \sum_{j=1}^{\infty}A_{oe}^{(2j)}(q,n)\sin(2ju), \quad \mathsf{Soo}_n(q,u) = \sum_{j=0}^{\infty}A_{oo}^{(2j+1)}(q,n)\sin((2j+1)u),$$

$$\mathsf{Ree}_n^{(1,2)}(q,u) = \sqrt{\frac{\pi}{2}}\frac{(-1)^r}{A_{ee}^{(2s)}(q,n)}Mc_{ee}^{(1,2)}(q,u), \quad r = t/2, \quad t = 0,2,4,...,$$

$$\mathsf{Reo}_n^{(1,2)}(q,u) = \sqrt{\frac{\pi}{2}}\frac{(-1)^r}{A_{eo}^{(2s+1)}(q,n)}Mc_{eo}^{(1,2)}(q,u), \quad r = (t-1)/2, \quad t = 1,3,5,...,$$

$$\mathsf{Roe}_n^{(1,2)}(q,u) = \sqrt{\frac{\pi}{2}}\frac{(-1)^r}{A_{oe}^{(2s)}(q,n)}Ms_{oe}^{(1,2)}(q,u), \quad r = t/2, \quad t = 2,4,6,...,$$

$$\mathsf{Roo}_n^{(1,2)}(q,u) = \sqrt{\frac{\pi}{2}}\frac{(-1)^r}{A_{oo}^{(2s+1)}(q,n)}Ms_{oo}^{(1,2)}(q,u) \quad r = (t-1)/2, \quad t = 1,3,5,...,$$

$$\mathsf{Re}_n^{(3)}(q,u) = \mathsf{Re}_n^{(1)}(q,u) + i\mathsf{Re}_n^{(2)}(q,u),$$

$$\mathsf{Ro}_n^{(3)}(q,u) = \mathsf{Re}_n^{(1)}(q,u) + i\mathsf{Ro}_n^{(2)}(q,u),$$

with,

$$Mc_{ee}^{(p)}(q,v_1,v_2) = \sum_{j=0}^{\infty}(-1)^j A_{ee}^{(2j)}(q,n)\left(J_{j-s}(v_1)Z_{j+s}^p(v_2) + J_{j+s}(v_1)Z_{j-s}^p(v_2)\right),$$

$$Mc_{eo}^{(p)}(q,v_1,v_2) = \sum_{j=0}^{\infty}(-1)^j A_{eo}^{(2j+1)}(q,n)\left(J_{j-s}(v_1)Z_{j+s+1}^p(v_2) + J_{j+s+1}(v_1)Z_{j-s}^p(v_2)\right),$$

$$Ms_{oe}^{(p)}(q,v_1,v_2) = \sum_{j=1}^{\infty}(-1)^j A_{oe}^{(2j)}(q,n)\left(J_{j-s}(v_1)Z_{j+s}^p(v_2) - J_{j+s}(v_1)Z_{j-s}^p(v_2)\right),$$

$$Ms_{oo}^{(p)}(q,v_1,v_2) = \sum_{j=0}^{\infty}(-1)^j A_{oo}^{(2j+1)}(q,n)\left(J_{j-s}(v_1)Z_{j+s+1}^p(v_2) - J_{j+s+1}(v_1)Z_{j-s}^p(v_2)\right)$$

where, $v_1 = \sqrt{q}\,e^{-u}$, $v_2 = \sqrt{q}\,e^{u}$, $Z^1 = J$, $Z^2 = Y$, *are the Bessel functions of first and*

second kind respectively, s is an arbitrary integer parameter, n denote the order in the succession of all orders and t is the true value or order n.

The series expansion (3.5) is to be considered for both cases of the even and the odd Mathieu functions; that is, we will have four terms for each term in the sum. Note that among the several notations used to denote Mathieu functions, in this work we use the Stratton convention [37].

Those are the two addition theorems to be used for the Helmholtz case.

Let's now enunciate the ones to be used for the biharmonic case. We denote, $x = \big(r\cos(\theta), r\sin(\theta)\big)$ and $y = \big(\rho\cos(\alpha), \rho\sin(\alpha)\big)$. From [9, 30] we have the proof of the theorem.

Theorem 3.3 (Addition theorem in polar coordinates for $\log|x-y|$)**.** *For $\rho > r$ there exists the expansion*

$$\log|x-y| = \log(\rho) - \sum_{m=1}^{\infty} \frac{r^m}{m\rho^m}\big(\cos(m\theta)\cos(m\alpha) + \sin(m\theta)\sin(m\alpha)\big). \tag{3.6}$$

Theorem 3.4 (Addition theorem in polar coordinates for $|x-y|^2\log|x-y|$)**.** *For $\rho > r$ there exists the expansion*

$$\begin{aligned}
&|x-y|^2\log|x-y| = \\
&r^2\log(\rho) + \rho_j^2\log(\rho) + r^2 - \left(2r\rho\log(\rho) + \frac{r^3}{2\rho} + r\rho\right)\Big(\cos(\theta)\cos(\alpha) + \sin(\theta)\sin(\alpha)\Big) \\
&+ \sum_{m=2}^{\infty}\left(\frac{r^m}{m(m-1)\rho^{m-2}} - \frac{r^{m+2}}{m(m+1)\rho^m}\right)\Big(\cos(m\theta)\cos(m\alpha) + \sin(m\theta)\sin(m\alpha)\Big).
\end{aligned} \tag{3.7}$$

Proof. From (3.6), we can deduce $|x-y|^2\log|x-y|$ in polar coordinates,

$$\begin{aligned}
|x-y|^2\log|x-y| &= \Big((r\cos(\theta) - \rho\cos(\alpha))^2 + \big(r\sin(\theta) - \rho\sin(\alpha)\big)^2\Big)\log|x-y| \\
&= \Big(r^2 + \rho^2 - 2r\rho(\cos(\theta)\cos(\alpha) + \sin(\theta)\sin(\alpha))\Big)\log|x-y| \\
&= \Big(r^2 + \rho^2 - 2r\rho(\cos(\theta)\cos(\alpha) + \sin(\theta)\sin(\alpha))\Big)\Big(\log(\rho) \\
&\quad - \sum_{m=1}^{\infty}\frac{r^m}{m\rho^m}\big(\cos(m\theta)\cos(m\alpha) + \sin(m\theta)\sin(m\alpha)\big)\Big)
\end{aligned}$$

$$
\begin{aligned}
&= r^2\log(\rho)+\rho^2\log(\rho)-2r\rho\log(\rho)\Big(\cos(\theta)\cos(\alpha)+\sin(\theta)\sin(\alpha)\Big)\\
&+\sum_{m=1}^{\infty}\Bigg[-\left(\frac{r^{m+2}}{m\rho^m}+\frac{r^m}{m\rho^{m-2}}\right)\Big(\cos(m\theta)\cos(m\alpha)+\sin(m\theta)\sin(m\alpha)\Big)\\
&+2\frac{r^{m+1}}{m\rho^{m-1}}\Big(\cos(m\theta)\cos(m\alpha)+\sin(m\theta)\sin(m\alpha)\Big)\Big(\cos(\theta)\cos(\alpha)+\sin(\theta)\sin(\alpha)\Big)\Bigg]\\
&= r^2\log(\rho)+\rho^2\log(\rho)-2r\rho\log(\rho)\Big(\cos(\theta)\cos(\alpha)+\sin(\theta)\sin(\alpha)\Big)\\
&-\sum_{m=1}^{\infty}\left(\frac{r^{m+2}}{m\rho^m}+\frac{r^m}{m\rho^{m-2}}\right)\Big(\cos(m\theta)\cos(m\alpha)+\sin(m\theta)\sin(m\alpha)\Big)\\
&+\sum_{m=1}^{\infty}\frac{r^{m+1}}{m\rho^{m-1}}\Big(\cos\big((m-1)\theta\big)\cos\big((m-1)\alpha\big)+\sin\big((m-1)\theta\big)\sin\big((m-1)\alpha\big)\\
&+\cos\big((m+1)\theta\big)\cos\big((m+1)\alpha\big)+\sin\big((m+1)\theta\big)\sin\big((m+1)\alpha\big)\Big),
\end{aligned}
$$

from which we get finally,

$$
\begin{aligned}
&|x-y|^2\log|x-y| =\\
&r^2\log(\rho)+\rho_j^2\log(\rho)+r^2-\left(2r\rho\log(\rho)+\frac{r^3}{2\rho}+r\rho\right)\Big(\cos(\theta)\cos(\alpha)+\sin(\theta)\sin(\alpha)\Big)\\
&+\sum_{m=2}^{\infty}\left(\frac{r^m}{m(m-1)\rho^{m-2}}-\frac{r^{m+2}}{m(m+1)\rho^m}\right)\Big(\cos(m\theta)\cos(m\alpha)+\sin(m\theta)\sin(m\alpha)\Big).
\end{aligned}
$$

□

Now to build the new basis functions one need to consider, (2.6), (2.12), (2.13) and the addition theorems (3.2), (3.5), (3.6), (3.7), from which we define an operator for the array of the basis functions for MFS.

Let us define $\Psi(\chi,\xi):=H_0^{(1)}(k|x-y|)$ for the Helmholtz case, or for the biharmonic case $\Psi(\chi,\xi):=\log|x-y|$, $\Psi(\chi,\xi):=|x-y|^2\log|x-y|$, with (χ,ξ) being either the polar coordinates or elliptic coordinates, when applicable. For $x\in\partial\Omega$, $y\in\partial\Omega'$, $|y|>|x|$ we define the operator as

$$
\Theta_\infty(\chi,\boldsymbol{\xi})=\begin{bmatrix}\Psi(\chi,\xi_1)\\ \Psi(\chi,\xi_2)\\ \vdots\\ \Psi(\chi,\xi_N)\end{bmatrix}. \tag{3.8}
$$

Note that for the Helmholtz case it follows from (2.6) that $\Theta_\infty(\chi,\boldsymbol{\xi})$ contains N rows, and for the biharmonic it follows from (2.12), (2.13) that we have $2N$ rows.

Replacing $\Psi(\chi,\xi)$ by the addition theorem, we proceed by writing $\Theta_\infty(\chi,\boldsymbol{\xi})$ as a product of factors depending separately on the source points and the observation point. This means that we should write

$$
\Theta_\infty(\chi,\boldsymbol{\xi})=\mathbf{B}_\infty(\boldsymbol{\xi})\mathbf{F}_\infty(\chi).
$$

We truncate the expansions (2.6), (2.12), (2.13) by some $n = T \in \mathbb{N}$ in such a way that we keep double precision in the results and we ensure that matrices **B** and **F** (the truncated matrices from $\mathbf{B}_\infty$ and $\mathbf{F}_\infty$, respectively), have at least as many columns or many rows, respectively, as the number of rows of $\Theta_\infty(\chi, \boldsymbol{\xi})$. Then, we can simply write

$$\Theta(\chi, \boldsymbol{\xi}) = \mathbf{B}(\boldsymbol{\xi})\mathbf{F}(\chi). \tag{3.9}$$

We compute the singular value decomposition of **B**,

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^*,$$

with **U** and **V** being unitary and **S** diagonal with non negative entries. Multiplying from the left by $\mathbf{U}^*$ we get

$$\mathbf{U}^*\mathbf{B} = \mathbf{U}^*\mathbf{U}\mathbf{S}\mathbf{V}^* = \mathbf{I}\mathbf{S}\mathbf{V}^* = \mathbf{S}\mathbf{V}^*. \tag{3.10}$$

Since we have truncated the expansion by T such that the number of columns of **B** is higher or equal to the number of rows of $\Theta_\infty(\chi, \boldsymbol{\xi})$, then we can write

$$\mathbf{S} = \left[\mathbf{S}_1 | \mathbf{0}\right],$$

where $\mathbf{S}_1$ is a square matrix and **0** denotes a block matrix with all entries equal to zero. It follows that

$$\mathbf{S}\mathbf{V}^* = \left[\mathbf{S}_1 | \mathbf{0}\right] \begin{bmatrix} \mathbf{V}_1^* \\ \mathbf{V}_2^* \end{bmatrix} = \mathbf{S}_1\mathbf{V}_1^*,$$

where $\mathbf{V}_1^*$ is the matrix composed of the first N rows of $\mathbf{V}^*$, with N being the number of rows of $\Theta_\infty(\chi, \boldsymbol{\xi})$. Now, from (3.10) by multiplying on the left by the matrix $\mathbf{S}_1^{-1}$ we obtain

$$\mathbf{S}_1^{-1}\mathbf{U}^*\mathbf{B} = \mathbf{S}_1^{-1}\mathbf{S}\mathbf{V}^* = \mathbf{S}_1^{-1}\mathbf{S}_1\mathbf{V}_1^* = \mathbf{V}_1^*.$$

This shows that we obtain a new set of basis functions $\phi(\chi)$ by formally multiplying by $\mathbf{S}_1^{-1}\mathbf{U}^*$ on the left, which is

$$\phi(\chi) = \left(\mathbf{S}_1^{-1}\mathbf{U}^*\right)\Theta(\chi, \boldsymbol{\xi}) = \mathbf{V}_1^*\mathbf{F}(\chi). \tag{3.11}$$

To take advantage of better numerical conditioning, one should now use the new basis functions computed using the last matrix product in (3.11).

The approach MFS-SVD consists in the approximation obtained by a linear combination of the new basis function $\{\varphi_j(\chi) | \, j = 1, 2, ..., N\}$,

$$u_N^{SVD}(\chi) = \sum_{j=1}^{N} a_j^{SVD}\varphi_j(\chi) \tag{3.12}$$

where φ_j are the rows of the column matrix $\phi(\chi)$.

The determination of the coefficients $\{a_j^{SVD} | \, j = 1, 2, ..., N\}$ is made by ensuring that (3.12) fits the BCs of the problem. This means that by collocation at the M boundary points $\{\chi_i | \, i = 1, 2, ..., M\}$ we obtain the linear system that allows us to determine them.

Remark 3.1. *In general matrix/vector* $\boldsymbol{F}(\chi)$ *is badly scaled and this is the source of the ill-conditioning. Therefore, before applying SVD on* $\boldsymbol{B}(\boldsymbol{\xi})$ *a proper scaling of* $\boldsymbol{F}(\chi)$ *should be done which will transfer the source of ill-conditioning to matrix* $\boldsymbol{B}(\boldsymbol{\xi})$.

3.2 The MFS-SVD formulation for Helmholtz equation in polar coordinates

We follow the generic steps described in section 3.1 to extend MFS-SVD to Helmholtz equation in polar coordinates, which we denote by MFS-SVD$_P$.

Therefore, dropping the constant $\mathrm{i}/4$ from the fundamental solution (2.5), that can be assumed to be incorporated in the coefficients of MFS, the operator (3.8) is given by

$$\begin{aligned}\Theta_\infty(r_P,\theta,\boldsymbol{\rho}_P,\boldsymbol{\alpha}) &= \begin{bmatrix} H_0^{(1)}(k|x-y_1|) \\ H_0^{(1)}(k|x-y_2|) \\ \vdots \\ H_0^{(1)}(k|x-y_N|) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{B}_{p,0}^1 & \mathbf{B}_{p,1}^1 & \mathbf{B}_{p,2}^1 & \mathbf{B}_{p,3}^1\cdots \\ \mathbf{B}_{p,0}^2 & \mathbf{B}_{p,1}^2 & \mathbf{B}_{p,2}^2 & \mathbf{B}_{p,3}^2\cdots \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{B}_{p,0}^N & \mathbf{B}_{p,1}^N & \mathbf{B}_{p,2}^N & \mathbf{B}_{p,3}^N\cdots \end{bmatrix} \begin{bmatrix} J_0(kr_P)/J_0^* \\ J_1(kr_P)/J_1^*\cos(\theta) \\ J_1(kr_P)/J_1^*\sin(\theta) \\ J_2(kr_P)/J_2^*\cos(2\theta) \\ J_2(kr_P)/J_2^*\sin(2\theta) \\ J_3(kr_P)/J_3^*\cos(3\theta) \\ J_3(kr_P)/J_3^*\sin(3\theta) \\ \vdots \end{bmatrix} \\ &= \mathbf{B}_{P,\infty}(\boldsymbol{\rho}_P,\boldsymbol{\alpha})\mathbf{F}_{P,\infty}(r_P,\theta), \end{aligned} \tag{3.13}$$

where

$$\mathbf{B}_{p,0}^j = \left[H_0^{(1)}(k\rho_{P,j})J_0^*\right], \mathbf{B}_{p,n}^j = \left[2H_n^{(1)}(k\rho_{P,j})J_n^*\cos(n\alpha_j) \quad 2H_n^{(1)}(k\rho_{P,j})J_n^*\sin(n\alpha_j)\right],$$

$$r_\Omega := \min_{\mathsf{x}\in\partial\Omega}||\mathsf{x}||, \quad \rho_\Omega := \max_{\mathsf{x}\in\partial\Omega}||\mathsf{x}||, \quad J_n^* = \max_{r\in[r_\Omega,\rho_\Omega]}|J_n(kr)|, \tag{3.14}$$

for $n = 1,2,...$ and $j = 1,2,...,N$.

We truncate the expansion (2.6) by some $T_P \in \mathbb{N}$ such that $2T_P+1 \geq N$, to ensure that we have at least as many columns as the number of rows of $\Theta_\infty(r_P,\theta,\boldsymbol{\rho}_P,\boldsymbol{\alpha})$. Then (3.13) can be written as

$$\Theta(r_P,\theta,\boldsymbol{\rho}_P,\boldsymbol{\alpha}) = \mathbf{B}_P(\boldsymbol{\rho}_P,\boldsymbol{\alpha})\mathbf{F}_P(r_P,\theta).$$

We proceed as described in section 3.1 by performing a change of basis through SVD of $\mathbf{B}_P$, with $\mathbf{U},\mathbf{V}$ being unitary and $\mathbf{S}$ diagonal with non negative entries such that

$$\mathbf{B}_P = \mathbf{U}\mathbf{S}\mathbf{V}^*$$

by taking $\mathbf{V}_1^*$ as the matrix composed of the first N rows of $\mathbf{V}^*$ we obtain a new set of basis functions defined by (3.11) i.e. $\phi(r_P,\theta)$ given by

$$\phi(r_P,\theta) = \mathbf{V}_1^*\mathbf{F}_P(r_P,\theta).$$

The MFS-SVD approximation is the linear combination defined by

$$u_N^{SVD}(r_P,\theta) = \sum_{j=1}^{N} a_j^{SVD}\varphi_j(r_P,\theta),$$

where the coefficients can be determined by solving

$$\mathbf{A}^P\mathbf{a}^{SVD} = \mathbf{G},$$

where

$$\mathbf{A}^P = \Big(\varphi_j(r_{P,i},\theta_i)\Big)_{1\le i\le M,1\le j\le N}, \quad \mathbf{a}^{SVD} = \Big(a_j^{SVD}\Big)_{1\le j\le N}^{\mathsf{T}}, \quad \mathbf{G} = \Big(g(r_{P,i},\theta_i)\Big)_{1\le i\le M}^{\mathsf{T}}.$$

and

Remark 3.2. *Note that, as we assumed $N \le 2T_P+1$, an increasing number of source points N leads to an increasing truncation number T_P. This may raise underflow problems, due to vanishing asymptotic of the Bessel function J_n with increasing n, when computing the ratios J_n/J_n^* from a certain $n \ge N^*$. To avoid this, for N greater than a fixed N^*, the ratio J_n/J_n^* is calculated using the asymptotic behavior [1] given by*

$$J_n(z) \sim \frac{1}{\sqrt{2\pi n}}\left(\frac{ez}{2n}\right)^n, \quad n \longrightarrow \infty.$$

Remark 3.3. *In the context of the MFS to apply* (3.2) *one needs to ensure that $min_{\Omega'}\rho_P > max_{\Omega} r_P$.*

3.3 The MFS-SVD formulation for Helmholtz equation in elliptic coordinates

As we will see in section 4.1, the MFS-SVD$_P$ approach takes care of the ill-conditioning for disks, but it has a mild effect for domains that are elongated in one direction.

We consider now, the problem (2.4) with Ω being an elliptical domain. It is clear that the geometry of the domain is better described using elliptic coordinates, and Mathieu functions will be a natural choice for the basis functions, instead of the Bessel functions used for the disk.

In this section, we illustrate how to formulate the MFS-SVD approach when considering domains in the elliptic coordinates system. We will denote this by MFS-SVD$_E$.

We aim to solve (2.4) by transforming the problem from cartesian to elliptic coordinates.

Now by following the same rationale as in section 3.2, we can write

$$\begin{aligned}\Xi_\infty(r_E,\vartheta,\boldsymbol{\rho}_E,\boldsymbol{\omega}) &= \begin{bmatrix} H_0^{(1)}(k|x-y_1|)\\ H_0^{(1)}(k|x-y_2|)\\ \vdots\\ H_0^{(1)}(k|x-y_N|)\end{bmatrix} = \begin{bmatrix} \mathbf{B}^1_{E,0} & \mathbf{B}^1_{E,1} & \mathbf{B}^1_{E,2} & \mathbf{B}^1_{E,3}\cdots\\ \mathbf{B}^2_{E,0} & \mathbf{B}^2_{E,1} & \mathbf{B}^2_{E,2} & \mathbf{B}^2_{E,3}\cdots\\ \vdots & \vdots & \vdots & \vdots\\ \mathbf{B}^N_{E,0} & \mathbf{B}^N_{E,1} & \mathbf{B}^N_{E,2} & \mathbf{B}^N_{E,3}\cdots\end{bmatrix}\begin{bmatrix}\mathbf{F}_{E,0}\\ \mathbf{F}_{E,1}\\ \mathbf{F}_{E,2}\\ \mathbf{F}_{E,3}\\ \vdots\end{bmatrix}\\ &= \mathbf{B}_{E,\infty}(\boldsymbol{\rho}_E,\boldsymbol{\omega})\mathbf{F}_{E,\infty}(r_E,\vartheta),\end{aligned} \tag{3.15}$$

where for $i,n = 0,1,...$ and $j = 1,2,...,N$.

$$\begin{aligned}&\mathbf{B}^j_{E,i} = \begin{bmatrix} Cee_i\mathsf{See}_i(q,\omega_j) & Ceo_i\mathsf{Seo}_i(q,\omega_j) & Coe_i\mathsf{Soe}_i(q,\omega_j) & Coo_i\mathsf{Soo}_i(q,\omega_j)\end{bmatrix},\\ &\mathbf{B}_{E,i} = \begin{bmatrix}\mathsf{Ree}^{(1)}{}_i(q,r_E)\mathsf{See}_i(q,\vartheta)/Lee_i\\ \mathsf{Reo}^{(1)}{}_i(q,r_E)\mathsf{Seo}_i(q,\vartheta)/Leo_i\\ \mathsf{Roe}^{(1)}{}_i(q,r_E)\mathsf{Soe}_i(q,\vartheta)/Loe_i\\ \mathsf{Roo}^{(1)}{}_i(q,r_E)\mathsf{Soo}_i(q,\vartheta)/Loo_i\end{bmatrix}, \quad Lst_n = \max_{r\in[r_\Omega,\rho_\Omega]}|\mathsf{Rst}^{(1)}_n(q,r_E)|, \quad s,t=[e,o],\end{aligned} \tag{3.16}$$

r_Ω and ρ_Ω are defined in (3.14), and

$$Cee_n = \frac{4\mathsf{Ree}^{(3)}_n(q,\rho_E)}{Nee_n}Lee_n, \quad Ceo_n = \frac{4\mathsf{Reo}^{(3)}_n(q,\rho_E)}{Neo_n}Leo_n,$$
$$Coe_n = \frac{4\mathsf{Roe}^{(3)}_n(q,\rho_E)}{Noe_n}Loe_n, \quad Coo_n = \frac{4\mathsf{Roo}^{(3)}_n(q,\rho_E)}{Noo_n}Loo_n.$$

When expansion (3.5) is truncated for some $T_E \in \mathbb{N}$ satisfying $4T_E \geq N$, we get

$$\Xi(r_E,\vartheta,\boldsymbol{\rho}_E,\boldsymbol{\omega}) = \mathbf{B}_E(\boldsymbol{\rho}_E,\boldsymbol{\omega})\mathbf{F}_E(r_E,\vartheta).$$

From here, we proceed as described in section 3.1 to obtain the new basis functions.

3.4 The MFS-SVD formulation for biharmonic equation

In this section, we introduce the generalization of MFS-SVD to the biharmonic problem (2.11). Two different approximations will be considered as described in section 2.3.2.

We define as well,

$$\rho_\Omega := \max_{x\in\partial\Omega}||x||. \tag{3.17}$$

For simplicity and without loss of generality, let us drop the constants $1/(2\pi)$ and $1/(8\pi)$ from the definition of Φ_1 and Φ_2 respectively, in (2.12).

3.4.1 The case with approximation A_1

Let us now define the operator (3.8) for the biharmonic case using representations (2.12). In this case the operator is defined by

$$\Theta_\infty(r,\theta,\boldsymbol{\rho},\boldsymbol{\alpha}) = \begin{bmatrix} \Phi_1(x-y_1) \\ \vdots \\ \Phi_1(x-y_N) \\ \hline \Phi_2(x-y_1) \\ \vdots \\ \Phi_2(x-y_N) \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \mathbf{F}_\infty(r,\theta) = \mathbf{B}_\infty(\boldsymbol{\rho},\boldsymbol{\alpha})\mathbf{F}_\infty(r,\theta). \tag{3.18}$$

3.4.2 The case with approximation A_2

In case of approximation (2.13) the operator is given by

$$\Theta_\infty(r,\theta,\boldsymbol{\rho},\boldsymbol{\alpha}) = \begin{bmatrix} \Phi_2(x-y_1) \\ \vdots \\ \Phi_2(x-y_N) \\ \hline \dfrac{\partial\Phi_2(x-y_1)}{\partial\mathbf{n}(y_1)} \\ \vdots \\ \dfrac{\partial\Phi_2(x-y_N)}{\partial\mathbf{n}(y_N)} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{B}_3 \end{bmatrix} \mathbf{F}_\infty(r,\theta) = \mathbf{B}_\infty(\boldsymbol{\rho},\boldsymbol{\alpha})\mathbf{F}_\infty(r,\theta). \tag{3.19}$$

In both cases, (3.18) and (3.19) the blocks $\mathbf{B}_l$ and $\mathbf{F}_\infty(r,\theta)$ are given by

$$\mathbf{B}_l = \begin{bmatrix} \mathbf{B}_l^0(\alpha_1) & \mathbf{B}_l^1(\alpha_1) & \ldots & \mathbf{B}_l^m(\alpha_1) & \ldots \\ \vdots & \vdots & \vdots & \ldots & \\ \mathbf{B}_l^0(\alpha_N) & \mathbf{B}_l^1(\alpha_N) & \ldots & \mathbf{B}_l^m(\alpha_N) & \ldots \end{bmatrix}, \quad l = 1,2,3, \tag{3.20}$$

$$\mathbf{F}_\infty(r,\theta) = \begin{bmatrix} 1 \\ r^2/\rho_\Omega^2 \\ r/\rho_\Omega \cos(\theta) \\ r/\rho_\Omega \sin(\theta) \\ r^3/\rho_\Omega^3 \cos(\theta) \\ r^3/\rho_\Omega^3 \sin(\theta) \\ r^m/\rho_\Omega^m \cos(m\theta) \\ r^m/\rho_\Omega^m \sin(m\theta) \\ r^{m+2}/\rho_\Omega^{m+2} \cos(m\theta) \\ r^{m+2}/\rho_\Omega^{m+2} \sin(m\theta) \\ \vdots \end{bmatrix}, \tag{3.21}$$

with

$$\begin{aligned}
\mathbf{B}_1^0(\alpha_j) &= \begin{bmatrix} \log\rho_j & 0 \end{bmatrix}, \\
\mathbf{B}_1^m(\alpha_j) &= \begin{bmatrix} \frac{\rho_\Omega^m}{m\rho_j^m}\cos(m\alpha_j) & \frac{\rho_\Omega^m}{m\rho_j^m}\sin(m\alpha_j) & 0 & 0 \end{bmatrix}, \quad m \geq 1, \\
\mathbf{B}_2^0(\alpha_j) &= \begin{bmatrix} \rho_j^2\log\rho_j & (1+\log\rho_j)\rho_\Omega^2 \end{bmatrix}, \\
\mathbf{B}_2^1(\alpha_j) &= \begin{bmatrix} c_1\rho_\Omega\cos(\alpha_j) & c_1\rho_\Omega\sin(\alpha_j) & -\frac{\rho_\Omega^3}{2\rho_j}\cos(\alpha_j) & -\frac{\rho_\Omega^3}{2\rho_j}\sin(\alpha_j) \end{bmatrix}, \\
c_1 &= -(2\rho_j\log\rho_j + \rho_j), \\
\mathbf{B}_2^m(\alpha_j) &= \begin{bmatrix} c_2\rho_\Omega^m\cos(m\alpha_j) & c_2\rho_\Omega^m\sin(m\alpha_j) & c_3\rho_\Omega^{m+2}\cos(m\alpha_j) & c_3\rho_\Omega^{m+2}\sin(m\alpha_j) \end{bmatrix}, \\
c_2 &= \frac{1}{m(m-1)\rho_j^{m-2}}, \quad c_3 = -\frac{1}{m(m+1)\rho_j^m}, \quad m \geq 2,
\end{aligned}$$

$$\mathbf{B}_3^0(\alpha_j)^\mathsf{T} = \begin{bmatrix} d(\alpha_j)(2\rho_j\log\rho_j + \rho_j) \\ d(\alpha_j)\frac{1}{\rho_j}\rho_\Omega^2 \end{bmatrix},$$

$$\mathbf{B}_3^1(\alpha_j)^\mathsf{T} = \begin{bmatrix} \big(-d(\alpha_j)(2\log\rho_j+3)\cos(\alpha_j) + b(\alpha_j)(2\rho_j\log\rho_j+\rho_j)\sin(\alpha_j)\big)\rho_\Omega \\ \big(-d(\alpha_j)(2\log\rho_j+3)\sin(\alpha_j) - b(\alpha_j)(2\rho_j\log\rho_j+\rho_j)\cos(\alpha_j)\big)\rho_\Omega \\ \big(d(\alpha_j)\frac{1}{2\rho_j^2}\cos(\alpha_j) + b(\alpha_j)\frac{1}{2\rho_j}\sin(\alpha_j)\big)\rho_\Omega^3 \\ \big(d(\alpha_j)\frac{1}{2\rho_j^2}\sin(\alpha_j) - b(\alpha_j)\frac{1}{2\rho_j}\cos(\alpha_j)\big)\rho_\Omega^3 \end{bmatrix},$$

$$\mathbf{B}_3^m(\alpha_j)^\mathsf{T} = \begin{bmatrix} \big(-d(\alpha_j)\frac{(m-2)}{m(m-1)\rho_j^{m-1}}\cos(m\alpha_j) - b(\alpha_j)\frac{1}{(m-1)\rho_j^{m-2}}\sin(m\alpha_j)\big)\rho_\Omega^m \\ \big(-d(\alpha_j)\frac{(m-2)}{m(m-1)\rho_j^{m-1}}\sin(m\alpha_j) + b(\alpha_j)\frac{1}{(m-1)\rho_j^{m-2}}\cos(m\alpha_j)\big)\rho_\Omega^m \\ \big(d(\alpha_j)\frac{1}{(m+1)\rho_j^{m+1}}\cos(m\alpha_j) + b(\alpha_j)\frac{1}{(m+1)\rho_j^m}\sin(m\alpha_j)\big)\rho_\Omega^{m+2} \\ \big(d(\alpha_j)\frac{1}{(m+1)\rho_j^{m+1}}\sin(m\alpha_j) - b(\alpha_j)\frac{1}{(m+1)\rho_j^m}\cos(m\alpha_j)\big)\rho_\Omega^{m+2} \end{bmatrix}$$

with

$$d(\alpha_j) = \mathbf{n}(y_j) \cdot \big(\cos(\alpha_j), \sin(\alpha_j)\big), \quad b(\alpha_j, \rho_j) = \mathbf{n}(y_j) \cdot \left(-\frac{\sin(\alpha_j)}{\rho_j}, \frac{\cos(\alpha_j)}{\rho_j} \right).$$

If we truncate the expansions (3.6) and (3.7) for some $T \in \mathbb{N}$ such that $4T+2 \geq 2N$ in such a way that we obtain a matrix $\mathbf{B}$, the truncated version of $\mathbf{B}_\infty$, with at least as many columns as the number of rows of $\Theta_\infty(r, \theta, \boldsymbol{\rho}, \boldsymbol{\alpha})$ then, (3.18) can be written as

$$\Theta = \mathbf{B}(\boldsymbol{\rho}, \boldsymbol{\alpha})\mathbf{F}(r, \theta). \tag{3.22}$$

Note that as pointed out in [12], the ill-conditioning of the linear system is aggravated by the increasing powers of r in $\mathbf{F}$; the normalization of the elements of $\mathbf{F}$ by dividing by ρ_Ω guarantees $\mathbf{F}$ to be properly scaled for domains not far from a ball, transferring the sources for ill-conditioning to matrix $\mathbf{B}$ to be addressed afterwards by the SVD.

From (3.22), we proceed as in section (3.1) by performing a change of basis through SVD of $\mathbf{B}$, with $\mathbf{U}, \mathbf{V}$ being unitary and $\mathbf{S}$ diagonal with non negative entries such that

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^*$$

by taking $\mathbf{V}_1^*$ as the matrix composed of the first $2N$ rows of $\mathbf{V}^*$ we obtain a new set of basis functions $\phi(r, \theta)$ given by

$$\phi(r, \theta) = \mathbf{V}_1^* \mathbf{F}(r, \theta).$$

In this sense, the ansatz for the MFS-SVD approach is obtained by a linear combination of the basis function $\phi(r, \theta)$. This means we can write

$$u_{N,1}^{SVD}(r, \theta) = \sum_{j=1}^{2N} a_j^{SVD} \varphi_n(r, \theta)$$

with $\varphi_j(r, \theta), j = 1, 2, ..., 2N$ being the rows of $\phi(r, \theta)$.

The linear system (2.14) is obtained by collocation at M boundary points (r_i, θ_i)

$$\begin{aligned} u_{N,1}^{SVD}(r_i, \theta_i) &= g_1(r_i, \theta_i) \\ \frac{\partial u_{N,1}^{SVD}(r_i, \theta_i)}{\partial \mathbf{n}} &= g_2(r_i, \theta_i) \end{aligned}, \quad i = 1, 2, ..., M.$$

Remark 3.4. *The outward normal vector to a C^1 plane curve Γ given by a counter-clockwise parametrization $(x = x(t), y = y(t))$ can be expressed as $\boldsymbol{n} = \left(\frac{dy}{dt}, -\frac{dx}{dt} \right)$. In this case the normal derivative for $u(r, \theta)$ can be determined by [67]*

$$\frac{\partial u(r, \theta)}{\partial \boldsymbol{n}} = \eta(\theta) \left(\frac{\partial u(r, \theta)}{\partial r} - \frac{r'}{r^2} \frac{\partial u(r, \theta)}{\partial \theta} \right), \quad \eta(\theta) = \frac{r(\theta)}{\sqrt{r(\theta)^2 + r'(\theta)^2}},$$

with $r = r(\theta)$ being the radial parametrization of the curve Γ and r' denoting the derivative with respect to θ.

Remark 3.5. *If the pseudo-boundary is a circumference, following Remark 3.4 the unit normal is given by* $\boldsymbol{n}(y) = \big(\cos(\alpha), \sin(\alpha)\big)$. *Therefore,* $d(\alpha) = 1$ *and* $b(\alpha, \rho) = 0$.

Remark 3.6. *We would like to point out that, opposite to the Laplace case [12] and the Helmholtz case [13], the operator* Θ_∞ *does not generate the complete MFS linear system matrix* (2.15). *In fact, if we considered the operator* (3.8) *with the components* $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{n}}$, $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{n}}$, *for approximation* A_2 *and pseudo-boundary as circumference we would have*

$$\Theta_\infty = \left[\begin{array}{c|c} \Phi_2(x-y_1) & \dfrac{\partial \Phi_2(x-y_1)}{\partial \boldsymbol{n}(x)} \\ \vdots & \vdots \\ \Phi_2(x-y_N) & \dfrac{\partial \Phi_2(x-y_N)}{\partial \boldsymbol{n}(x)} \\ \hline \dfrac{\partial \Phi_2(x-y_1)}{\partial \boldsymbol{n}(y_1)} & \dfrac{\partial}{\partial \boldsymbol{n}(x)}\left(\dfrac{\partial \Phi_2(x-y_1)}{\partial \boldsymbol{n}(y_1)}\right) \\ \vdots & \vdots \\ \dfrac{\partial \Phi_2(x-y_N)}{\partial \boldsymbol{n}(y_N)} & \dfrac{\partial}{\partial \boldsymbol{n}(x)}\left(\dfrac{\partial \Phi_2(x-y_N)}{\partial \boldsymbol{n}(y_N)}\right) \end{array}\right] = \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{B}_3 \\ \boldsymbol{B}_2 & \boldsymbol{B}_4 \end{bmatrix} \begin{bmatrix} \boldsymbol{F}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}_2 \end{bmatrix}$$

$$= \boldsymbol{B}_\infty(\boldsymbol{\rho}, \boldsymbol{\alpha}) \boldsymbol{F}_\infty(r, \theta)$$

where

$$\boldsymbol{B}_l = \begin{bmatrix} \boldsymbol{B}_l^0(\alpha_1) & \boldsymbol{B}_l^1(\alpha_1) & \boldsymbol{B}_l^m(\alpha_1) & \dots \\ \vdots & \vdots & \vdots & \dots \\ \boldsymbol{B}_l^0(\alpha_N) & \boldsymbol{B}_l^1(\alpha_N) & \boldsymbol{B}_l^m(\alpha_N) & \dots \end{bmatrix}, \quad l = 1, 2, 3, 4$$

with

$$\boldsymbol{B}_1^0(\alpha_j) = \left[\rho_j^2 log\rho_j \quad 1+log\rho_j\right],$$

$$\boldsymbol{B}_1^1(\alpha_j) = \left[c_1\cos(\alpha_j) \quad c_1\sin(\alpha_j) \quad -\frac{1}{2\rho_j}\cos(\alpha_j) \quad -\frac{1}{2\rho_j}\sin(\alpha_j)\right],$$

$$\boldsymbol{B}_1^m(\alpha_j) = \left[c_2\cos(m\alpha_j) \quad c_2\sin(m\alpha_j) \quad c_3\cos(m\alpha_j) \quad c_3\sin(m\alpha_j)\right], \quad m \geq 2,$$

$$\boldsymbol{B}_2^0(\alpha_j) = \left[(2\rho log\rho_j + \rho_j) \quad \frac{1}{\rho_j}\right],$$

$$\boldsymbol{B}_2^1(\alpha_j) = \left[c_4\cos(\alpha_j) \quad c_4\sin(\alpha_j) \quad \frac{1}{2\rho_j^2}\cos(\alpha_j) \quad \frac{1}{2\rho_j^2}\sin(\alpha_j)\right],$$

$$c_4 = -(2log\rho_j + 3),$$

$$\boldsymbol{B}_2^m(\alpha_j) = \left[c_5\cos(m\alpha_j) \quad c_5\sin(m\alpha_j) \quad c_6\cos(m\alpha_j) \quad c_6\sin(m\alpha_j)\right],$$

$$c_5 = -\frac{(m-2)}{m(m-1)\rho_j^{m-1}}, \quad c_6 = \frac{1}{(m+1)\rho_j^{m+1}},$$

$$\boldsymbol{B}_3^0(\alpha_j) = \left[2log\rho_j + 2\right],$$

$$\boldsymbol{B}_3^1(\alpha_j) = \left[c_1\cos(\alpha_j) \quad c_1\sin(\alpha_j) \quad -\frac{3}{2\rho_j}\cos(\alpha_j) \quad -\frac{3}{2\rho_j}\sin(\alpha_j)\right],$$

$$\boldsymbol{B}_3^m(\alpha_j) = \left[c_7\cos(m\alpha_j) \quad c_7\sin(m\alpha_j) \quad c_8\cos(m\alpha_j) \quad c_8\sin(m\alpha_j)\right],$$

$$c_7 = \frac{m}{m(m-1)\rho_j^{m-2}}, \quad c_8 = -\frac{m+2}{m(m+1)\rho_j^m},$$

$$\boldsymbol{B}_4^0(\alpha_j) = \left[\frac{2}{\rho_j}\right],$$

$$\boldsymbol{B}_4^1(\alpha_j) = \left[c_4\cos(\alpha_j) \quad c_4\sin(\alpha_j) \quad \frac{3}{2\rho^2}\cos(\alpha_j) \quad \frac{3}{2\rho^2}\sin(\alpha_j)\right],$$

$$\boldsymbol{B}_4^m(\alpha_j) = \left[c_9\cos(m\alpha_j) \quad c_9\sin(m\alpha_j) \quad c_{10}\cos(m\alpha_j) \quad c_{10}\sin(m\alpha_j)\right],$$

$$c_9 = -\frac{(m-2)}{(m-1)\rho_j^{m-1}}, \quad c_{10} = \frac{m+2}{(m+1)\rho_j^{m+1}},$$

$$\boldsymbol{F}_1 = \begin{bmatrix} 1 \\ r^2 \\ r\cos(\theta) \\ r\sin(\theta) \\ r^3\cos(\theta) \\ r^3\sin(\theta) \\ r^m\cos(m\theta) \\ r^m\sin(m\theta) \\ r^{m+2}\cos(m\theta) \\ r^{m+2}\sin(m\theta) \\ \vdots \end{bmatrix}, \quad \boldsymbol{F}_2 = \begin{bmatrix} r \\ \cos(\theta) \\ \sin(\theta) \\ r^2\cos(\theta) \\ r^2\sin(\theta) \\ r^{m-1}\cos(m\theta) \\ r^{m-1}\sin(m\theta) \\ r^{m+1}\cos(m\theta) \\ r^{m+1}\sin(m\theta) \\ \vdots \end{bmatrix}$$

which illustrate that the factor $\boldsymbol{F}$ *would become singular after discretization. In fact for instance, for the unit disk (*$r = 1$*), the first and second lines of* $\boldsymbol{F}_1$ *are linearly dependent, as well as the* 3^{rd} *and* 5^{th}*,* 4^{th} *and* 6^{th}*, and so on. Similarly, the* 2^{nd} *and* 4^{th} *lines of* $\boldsymbol{F}_2$ *are linearly dependent, as well as the* 3^{rd} *and* 5^{th}*,* 4^{th} *and* 6^{th}*, etc... This illustrates that this factorization leads to an ill-conditioned system, showing that the approach to be considered should be the one described in section 3.4 which is to actually change the basis functions first and consider its normal derivatives after, instead of applying the SVD procedure to the decomposition of the MFS linear system. In Laplace [12] and Helmholtz [13] cases, both approaches lead to the same result, but for higher order BVP with higher order BC, this is not the case; this points out the innovative aspect of this approach for the biharmonic problem and similarly to other higher order operators.*

3.5 Numerical algorithm for MFS-SVD

The computational implementation of MFS-SVD remains similar to the classical MFS with few complexity added. There is an additional SVD decomposition and a matrix multiplication as in (3.11) to consider. There is also the need to take into account the truncation of the addition theorem, that defines the precision of the approximation of the fundamental solution and the dimensions of matrices **B** and **F** in

(3.9). Although the implementation complexity does not significantly increase, the inclusion of additional steps such as the SVD, leads to a higher computational cost for MFS-SVD compared to the classical MFS, as we will see in section 4.1. The implementation algorithm for this technique, originally presented in [12], has been adapated in this work to address the specific cases under discussion.

Algorithm 1 MFS-SVD algorithm

1: Define the number of basis function, $N \in \mathbb{N}$.

2: Place $M > N$ collocation points on $\partial\Omega$.

3: Define the pseudo-boundary location and place N source points satisfying possible geometry restrictions imposed to ensure the convergence of the addition theorem.

4: Build the matrices **B, F**.

5: Calculate the SVD factorization of **B** to obtain $\mathbf{V}_1^*$.

6: Calculate the system matrix of the MFS-SVD by using (3.11). In fact, for each collocation point the corresponding line in the MFS-SVD system matrix is obtained by transposing the matrix $\mathbf{V}_1^*\mathbf{F}(\chi)$.

7: Solve the linear system to calculate the MFS-SVD coefficients.

8: Evaluate the solution at an arbitrary point in the convenient system of coordinates.

Chapter 4

Numerical application of MFS-SVD to Helmholtz and biharmonic BVP

In this chapter, we present some numerical results obtained by applying the MFS-SVD method to Helmholtz and biharmonic BVP. The results are always presented in comparison to the classical MFS to illustrate the improvements with the new approach. Normally, we present a comparison of the error and the evolution of the condition number between both approaches. In some cases we present also a comparison of the computation time, which is expected to be much higher for MFS-SVD. This is normally due to the SVD decomposition needed in the proposed approach, which is aggravated in the MFS-SVD$_E$ case due to larger matrices and also due to the evaluation of Mathieu functions. All results have been computed with Matlab on a desktop computer with an Intel(R) Core(TM) i5-4460T CPU @1.9GHz processor and 4.00 Gb of RAM and the figures are exported using a Matlab toolbox [2], which is supposed to improve the quality of the images.

4.1 Numerical examples for Helmholtz problems

To illustrate the MFS-SVD feasibility for Helmholtz BVP, we consider the following domains: Disk, ellipse, square and peanut. We evaluate the performance by measuring the error between u and its approximation $\bar{u}$ on a given set of test points Q, using the discrete L^2-norm

$$\varepsilon = \left(\frac{1}{\#Q} \sum_{z_i \in \mathrm{Q}} |u(z_i) - \bar{u}(z_i)|^2 \right)^{1/2}.$$

The points are selected in the interior of Ω, if the exact solution is known, or on its boundary otherwise. The source and the collocation points are distributed almost in all cases and we always consider $M = 2N$, with M being the number of collocation points and N the number of source points.

4.1.1 Examples for MFS-SVD in polar coordinates

The normalization parameters J_n^* defined in (3.14) are obtained as in [9] simply by evaluating $|J_n^*|$ at 10000 equally spaced points $r_i \in [r_{\partial\Omega}, \rho_{\partial\Omega}]$ and taking their maximum. In the context of Remark 3.2, to overcome underflow problems while computing the ratio J_n/J_n^*, we use a threshold of $N^* = 140$ since it is the first n that the Matlab calculation returns $J_n(1) = 0$.

Example 4.1. *Consider problem* (2.4) *in the unit disk with exact solution given by*

$$u(x) = -\frac{1}{4} Y_0(k|x - (x_0, 0)|), \quad x \in \bar{\Omega}, \quad x_0 > 1.$$

As in [9], we take the wavenumber $k = 8$ *and* $g = u$ *at the boundary. We look for a numerical approximation for* u *considering two sources locations:* $x_0 = 3$*, and* $x_0 = 1.1$*.*

Unless stated differently, we will always consider the pseudo-boundary as a circle to benefit from the fact that we are working in the polar coordinates system. This circle should be placed far enough to ensure that the geometric restriction mentioned in remark 3.3 is satisfied.

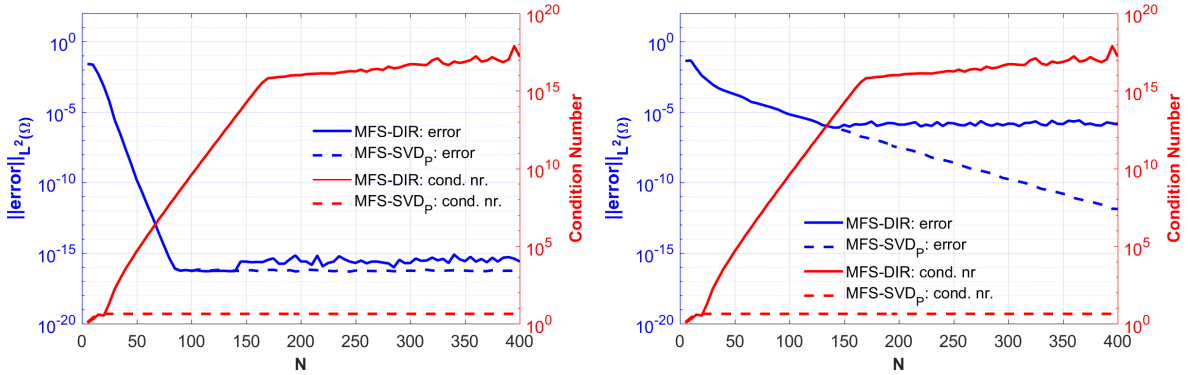
Let us select the pseudo-boundary as a circle with radius $\rho = 1.5$ centered at the origin. The results for both approaches, namely MFS-DIR and the proposed MFS-SVD$_P$ when $x_0 = 3$ and $x_0 = 1.1$ are shown in figure 4.1. In case $x_0 = 3$, figure 4.1a, both approaches have similar behavior, regardless of the number of source points considered, and in both cases around $N = 70$ source points are sufficient to reach machine precision. However, the similarities in the results are not observed in the computation time where MFS-SVD$_P$ is much more expensive than MFS-DIR, table 4.1.

The case $x_0 = 1.1$ is more challenging, since the solution cannot be analytically extended up to the pseudo-boundary $\partial\Omega'$. We consider approximations for different pseudo-boundaries, circles with $\rho = 1.5$, figure 4.1b, and $\rho = 1.05$, figure 4.1c, since it is known that MFS-DIR performs better when the pseudo-boundary is close to the domain. Figure 4.1c illustrates this behavior with an error of order 10^{-13} for $\rho = 1.05$ against 10^{-6} for $\rho = 1.5$ at 400 source points. On the other hand, MFS-SVD$_P$ is not affected by the location of the pseudo-boundary, showing results in both cases, similar to the best MFS-DIR case, all this due to the condition number, which remains bounded for MFS-SVD$_P$ while it explodes for MFS-DIR, namely in the case for $\rho = 1.5$.

This first example already illustrates that for large N, MFS-SVD$_P$ performs better than MFS-DIR. In order to understand the advantages of the change of basis, which is the rationale behind MFS-SVD, let us plot the restrictions of the basis functions at the boundary of the domain for both cases, MFS-DIR and MFS-SVD$_P$, that is, in the unitary disk plotting the function

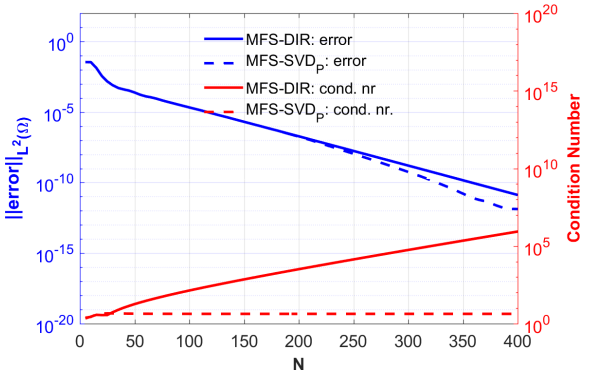
$$f(\theta) = \Phi\big((\cos(\theta), \sin(\theta)) - y_n\big), \quad 0 \leq \theta < 2\pi, \quad n = 1, 2, ..., N.$$

Note that in both approaches, either MFS-DIR or MFS-SVD, the coefficients are determined by collocation. In fact, the determination of the numerical solution can



(a) $\rho = 1.5$, $x_0 = 3$

(b) $\rho = 1.5$, $x_0 = 1.1$



(c) $\rho = 1.05$, $x_0 = 1.1$

Figure 4.1: Plot of error and condition number for MFS-SVD$_P$ vs. MFS-DIR for Helmholtz equation in the unit disk, with exact solution given by $u(x) = (-1/4)Y_0(k|x - (x_0, 0)|)$, for $k = 8$ and source points on circles with $\rho = 1.5$ and $\rho = 1.05$.

be viewed as an interpolation problem where the basis functions are the restrictions at boundary of the fundamental solution as described above.

In figure 4.2 (left) we show the first 8 of 150 MFS basis functions that span the functional space of the solutions of the case presented in figure 4.1a. We can observe that they are almost identical before the change of basis. This means that these functions are almost linearly dependent, and this has consequences on the conditioning of the linear system. On the other hand, after the change of basis, figure 4.2 (right), the first 8 functions of the new basis clearly differs one from another.

In addition to the geometric illustration provided in figure 4.2, we have calculated the Gram matrix for both basis functions, MFS and MFS-SVD, by considering the $L^2([0, 2\pi])$ inner product. In figure 4.3 we present an image of the Gram matrices for both, MFS on left and MFS-SVD on right. The matrices are constructed using the full set of 150 normalized functions of both basis. We can observe a dense matrix for the Gram matrix of MFS with a thicker yellow zone at the diagonal, figure 4.3 (left), which is related to the fact that we have some functions of the MFS basis almost identical, see figure 4.2 (left). With figure 4.2 (right) we clearly illustrate that we obtain an independent and orthogonal set of functions for the MFS-SVD basis.

Table 4.1: Computation time, in seconds, as a function of the number of source points for the case considered in figure 4.1b.

N	**MFS-DIR**	**MFS-SVD**$_P$
5	0.097	450.494
100	0.893	717.109
200	2.063	1001.109

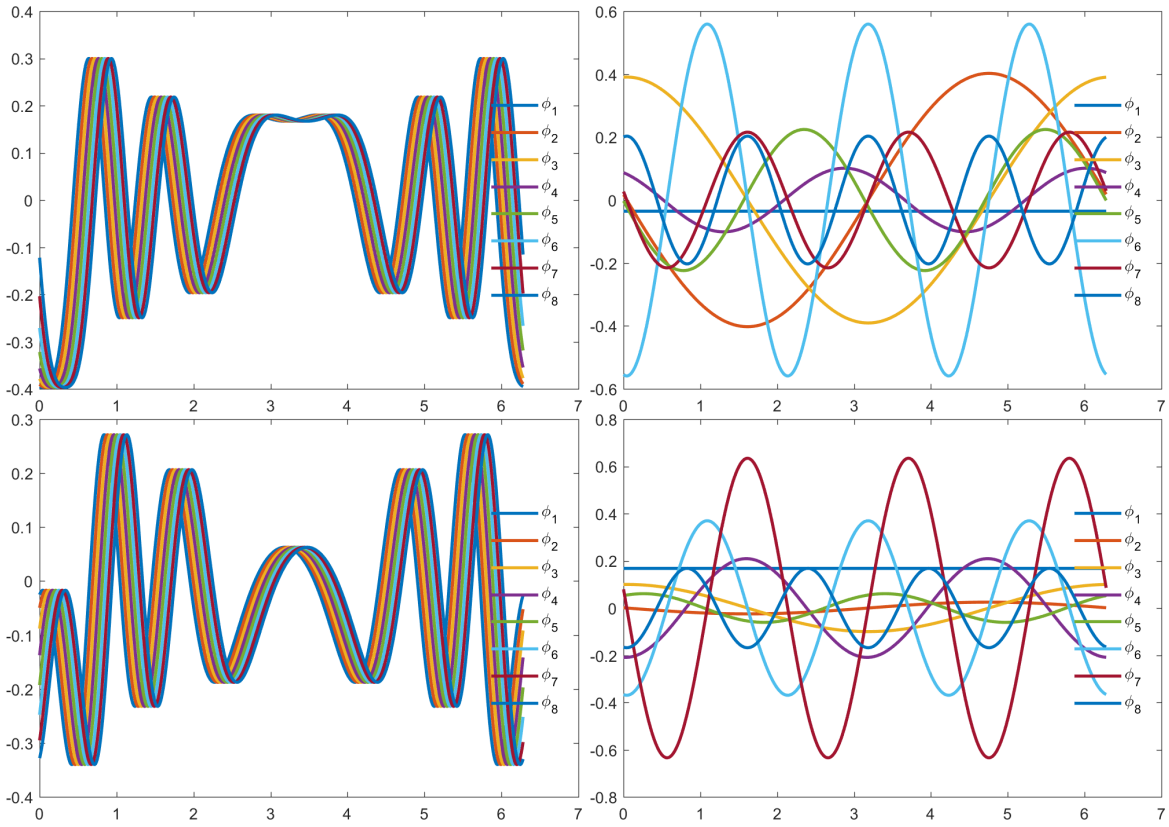


Figure 4.2: First 8 of 150 functions of the basis functions restricted at the boundary for the Helmholtz BVP with unitary circle as boundary, $k = 8$, pseudo-boundary as a circle with $\rho = 1.5$. On the (left) MFS basis and on the (right) MFS-SVD basis. At the (top) real part and (bottom) imaginary part.

To further test the robustness of MFS-SVD, we consider another example from [9].

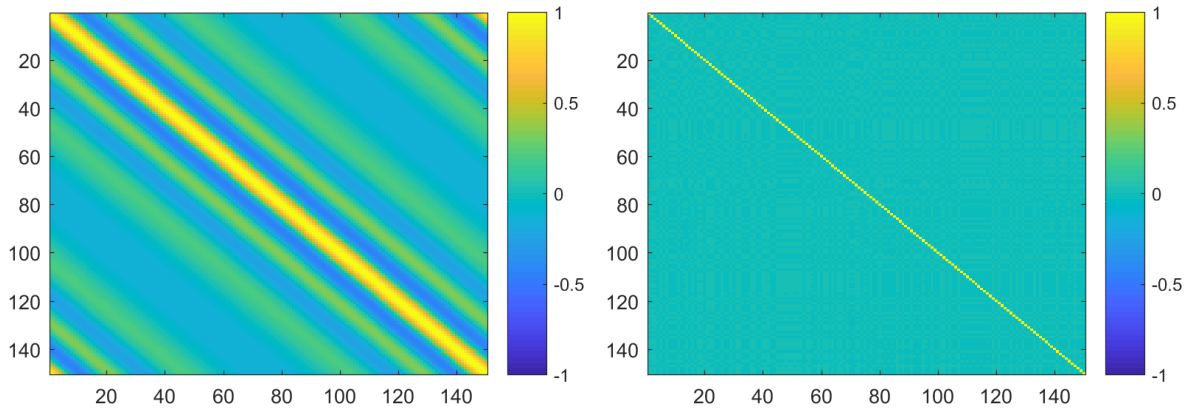


Figure 4.3: Images of Gram matrices for MFS (left) and MFS-SVD$_P$ (right) of the basis functions restricted at the boundary for the Helmholtz BVP in the unitary circle as boundary, $k = 8$, pseudo-boundary as a circle with $\rho = 1.5$.

Example 4.2. *Consider problem* (2.4) *in the unit disk with exact solution given by*

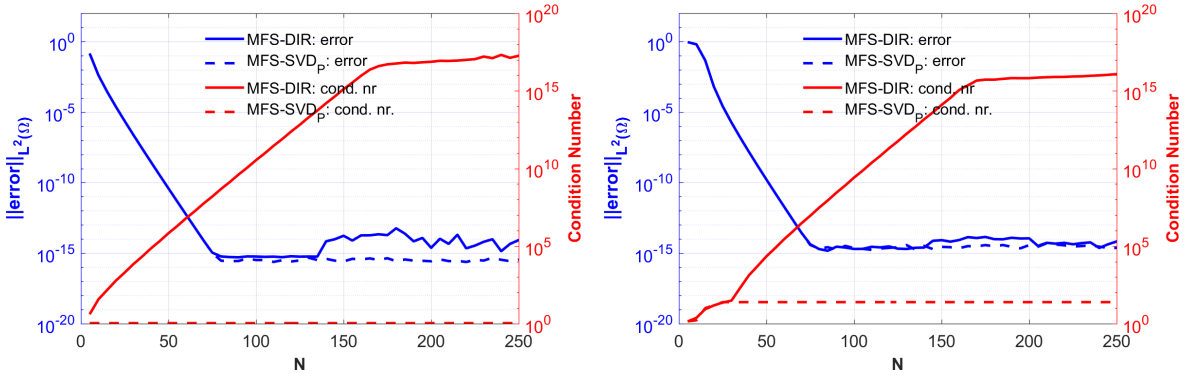
$$u(r,\theta) = \frac{e^{im\theta} J_m(kr)}{J_m(k)}, \quad m \in \mathbb{N}.$$

Fix $m = 2$ *and first look for approximations to the cases with wavenumber* $k = 1, 10, 100$ *and second fix* $k = 1$ *and explore the cases with* $m = 10, 30, 50$.

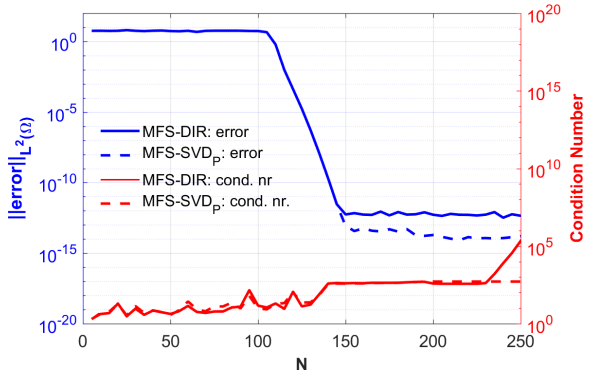
The scenario in which the wavenumber k varies is presented in figure 4.4. We can observe a similar behavior of the error for both approaches with minor improvements for MFS-SVD$_P$, with much better conditioning. In the second scenario, where we explore the approximations for $m = 10, 30, 50$ at fixed $k = 1$, we clearly see in figure 4.5 that MFS-SVD$_P$ performs better than the classical approach; in all cases MFS-SVD$_P$ reaches machine precision, at the cost of increasing the source points as m increases, whilst MFS-DIR shows results that deteriorate from the machine precision order in $m = 10$ to an error around 10^{-4} in case $m = 50$. In both scenarios, we can observe the boundedness of the condition number of MFS-SVD$_P$.

Remark 4.1. *The MFS-SVD$_P$ is based on the expansion* (3.13) *which is then truncated for* $T_P \in \mathbb{N}$, *with* $2T_P + 1 \geq N$. *The selection of* T_P *should be made carefully. In fact, increasing the value of* T_P *does not always translate into better approximations, due to the numerical computations of the truncated version of* (3.2). *In each case, the choice of source points and the guarantee of machine precision in* (3.13) *should define the value of* T_P.

Next, we analyze the MFS-SVD performance for other domains.



(a) $m = 2$, $k = 1$ (b) $m = 2$, $k = 10$



(c) $m = 2$, $k = 100$

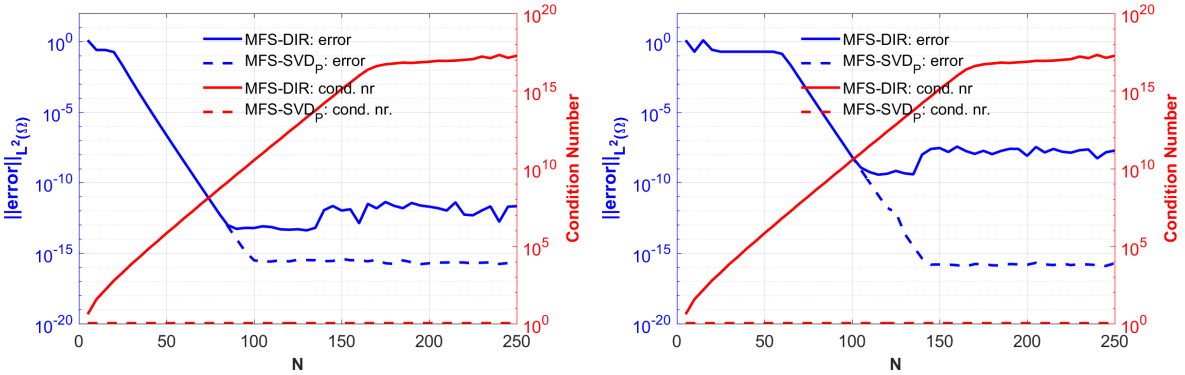
Figure 4.4: Plot of error and condition number for MFS-SVD$_P$ vs. MFS-DIR for Helmholtz equation with $u(r,\theta) = e^{im\theta} J_m(kr)/J_m(k)$, fixed m in the unit disk domain and the circle with $\rho = 1.5$ as pseudo-boundary.

Example 4.3. *We consider now, the problem* (2.4) *with an elliptic domain having as a boundary the ellipse defined by* $x^2/6^2 + y^2 = 1$. *As in [7] and [9] let us take*

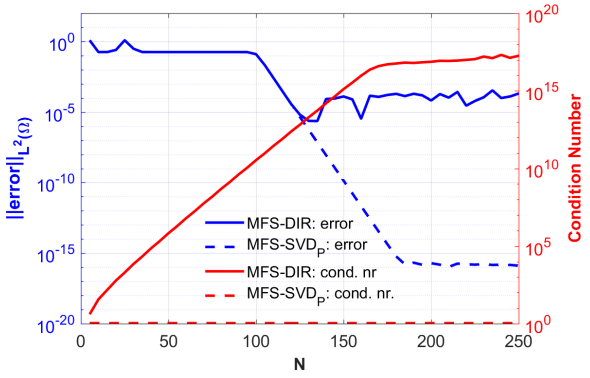
$$u(x,y) = g(x,y) = e^{i(x^2+y^2)} \sin(x+y) \quad \text{on} \quad \partial\Omega.$$

We look for the numerical approximation of u *when* $k = 1$.

We consider a scenario in which the classical MFS is expected to perform poorly. Specifically, we place the source points far from the domain, on a circle of radius $\rho = 50$, centered at origin, and on an ellipse whose semi-axes are magnified by factor of 10. It is important to ensure that these source points locations comply with the constraints, imposed by the convergence conditions of the addition theorem. The case with an ellipse as pseudo-boundary is to illustrate that MFS-SVD is not being benefited by the selection of pseudo-boundary geometry. The results in figure 4.6 show that regardless of the location of the source points, the classical approach always shows an error of order 1; in contrast, the MFS-SVD$_P$ presents slightly better results with an error of order 10^{-4}, as the evolution of the condition number is slightly



(a) $k = 1, m = 10$ (b) $k = 1, m = 30$



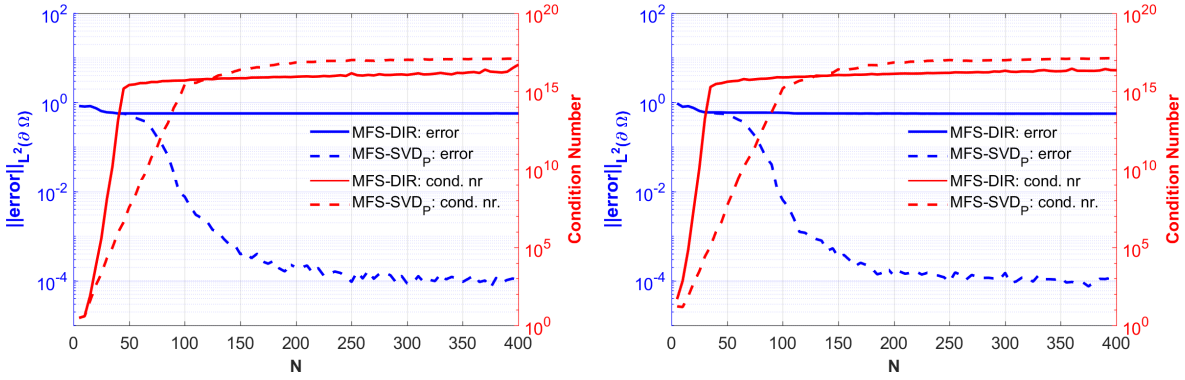
(c) $k = 1, m = 50$

Figure 4.5: Plot of error and condition number for MFS-SVD$_P$ vs. MFS-DIR for Helmholtz equation with $u(r,\theta) = e^{im\theta}J_m(kr)/J_m(k)$, fixed k in the unit disk domain and the circle with $\rho = 1.5$ as pseudo-boundary.

different compared to the classical approach. However, in both cases, we do not manage to control the condition number as in the disk case nor achieve machine precision accuracy.

Let us have a look as well to the restrictions of the basis function at the elliptic boundary. In figure 4.7 we have the first 8 functions of 150 for the basis restricted at the boundary. Again, before the change of basis, figure 4.7 (left), some functions from the basis are very similar. The MFS-SVD case, figure 4.7 (right), seems to show little improvements; this can be better illustrated by computing the Gram matrix for both basis functions. From figure 4.8 we can see that we are far from obtaining an orthogonal basis after the change of basis, figure 4.8 (right).

To finalize the examples in the polar coordinates section we consider a non-smooth domain, the square.



(a) Pseudo-boundary: Circle with radius $\rho = 50$

(b) Pseudo-boundary: Ellipse magnified by factor of 10

Figure 4.6: Plot of error at the boundary and condition number for MFS-SVD$_P$ vs. MFS-DIR for Helmholtz equation on an elliptic domain with elliptic boundary $x^2/6^2 + y^2 = 1$ and values at boundary given by $g(x,y) = e^{i(x^2+y^2)}\sin(x+y)$.

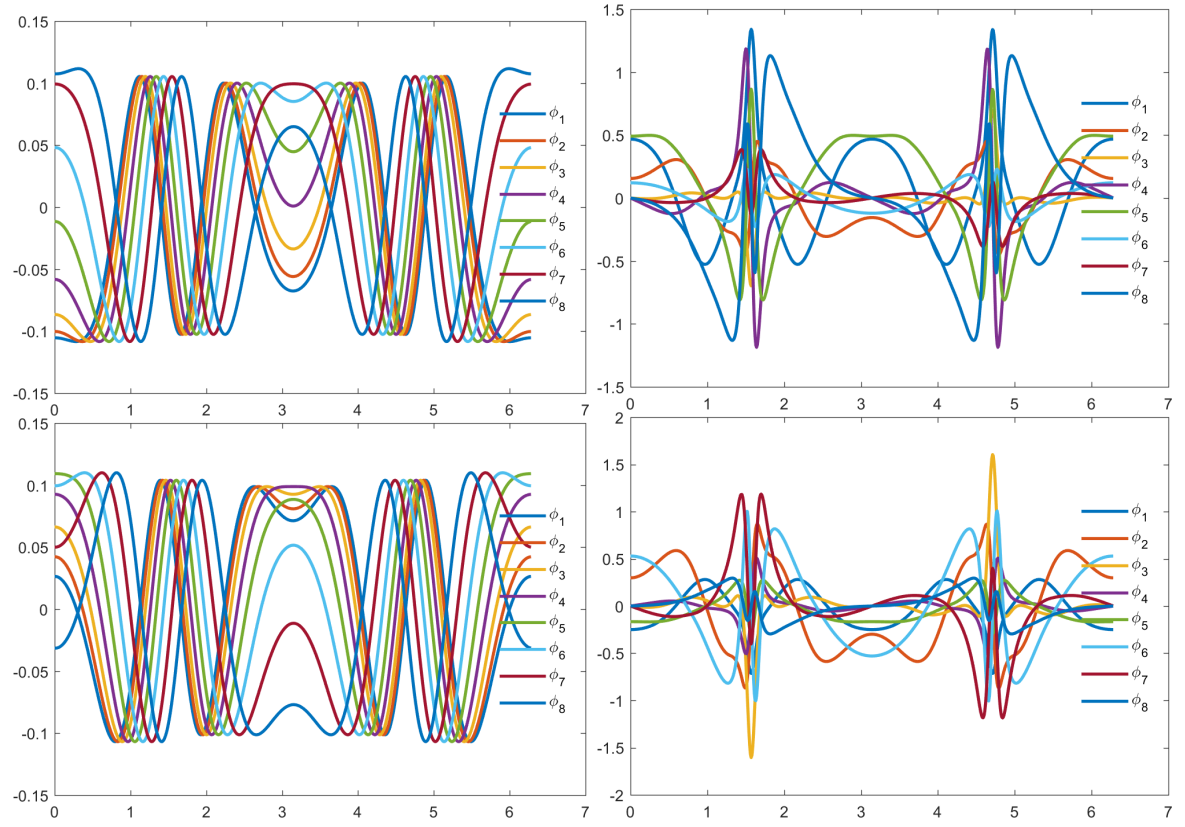


Figure 4.7: First 8 of 150 functions for the basis functions restricted at the boundary of Helmholtz BVP for case with elliptic boundary, $k = 8$, pseudo-boundary as an ellipse magnified by 10. On the (left) MFS basis and on the (right) MFS-SVD basis. At the (top) real part and (bottom) imaginary part.

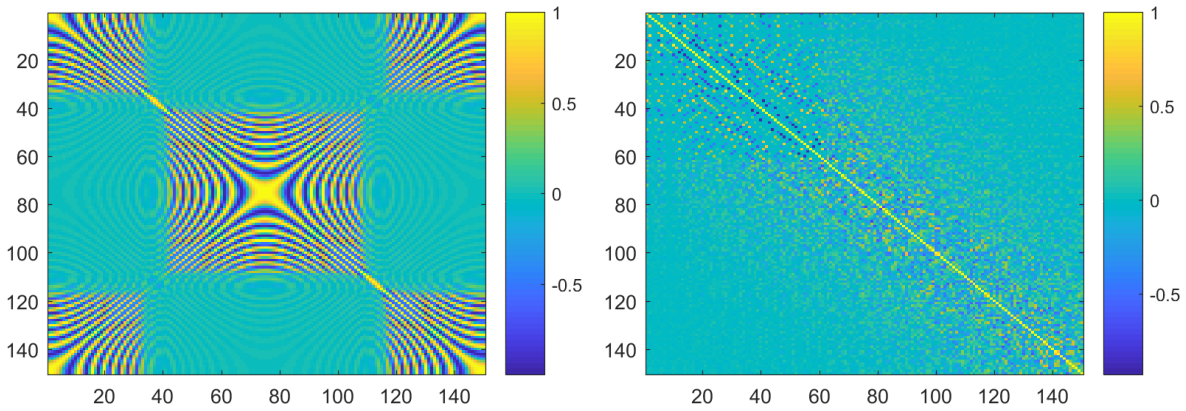
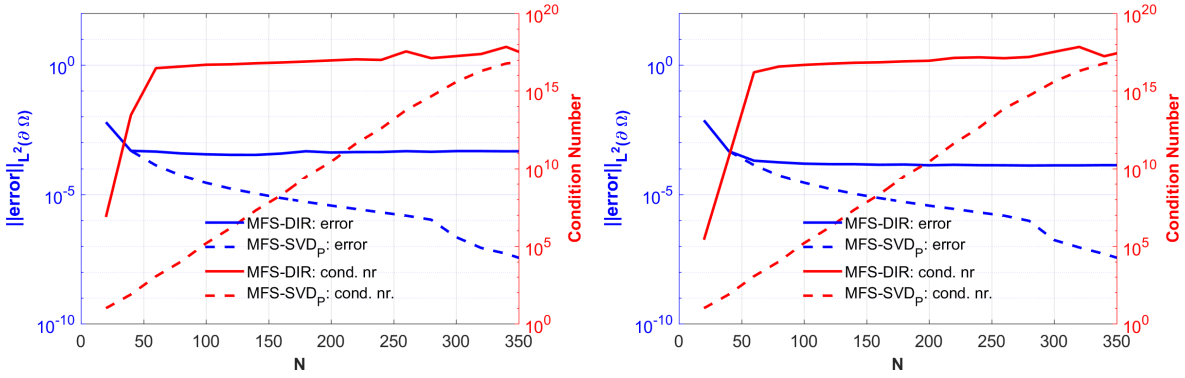


Figure 4.8: Images of Gram matrices for MFS (left) and MFS-SVD$_P$ (right) of the basis functions restricted at the boundary for the Helmholtz BVP in the elliptic boundary, pseudo-boundary as a circle with $\rho = 10$.

Example 4.4. *Let us consider the problem* (2.4) *with the domain being the square* $\Omega =]-1,1[\times]-1,1[$. *We take the same boundary conditions as in example 4.3. We look for numerical approximations to* u *when* $k = 1$.

We select as pseudo-boundaries two cases, a circle with radius $\rho = 5$ and the boundary of the square $]-3,3[\times]-3,3[$. The results are presented in figure 4.9 and regardless of the pseudo-boundary considered, the MFS-SVD$_P$ still performs better than the MFS-DIR. The improvement is not as pronounced as in the disk case, but still the MFS-SVD$_P$ has the condition number increasing at a slower pace than in the MFS-DIR which leads to improvements in accuracy. An important remark is that we have observed that to obtain good results with MFS-SVD$_P$ the set of source points should always include the corners of the square.

In figure 4.10, we can find some functions from the basis functions restricted at the square boundary.



(a) Pseudo-boundary: Circle with radius $\rho = 5$

(b) Pseudo-boundary: Boundary of a square with side length 6

Figure 4.9: Plot of error at the boundary of a square domain with side length equal to 2 and condition number for MFS-SVD$_P$ vs. MFS-DIR for Helmholtz equation and at boundary given by $g(\mathrm{x},\mathrm{y}) = e^{i(\mathrm{x}^2+\mathrm{y}^2)}\sin(\mathrm{x}+\mathrm{y})$.

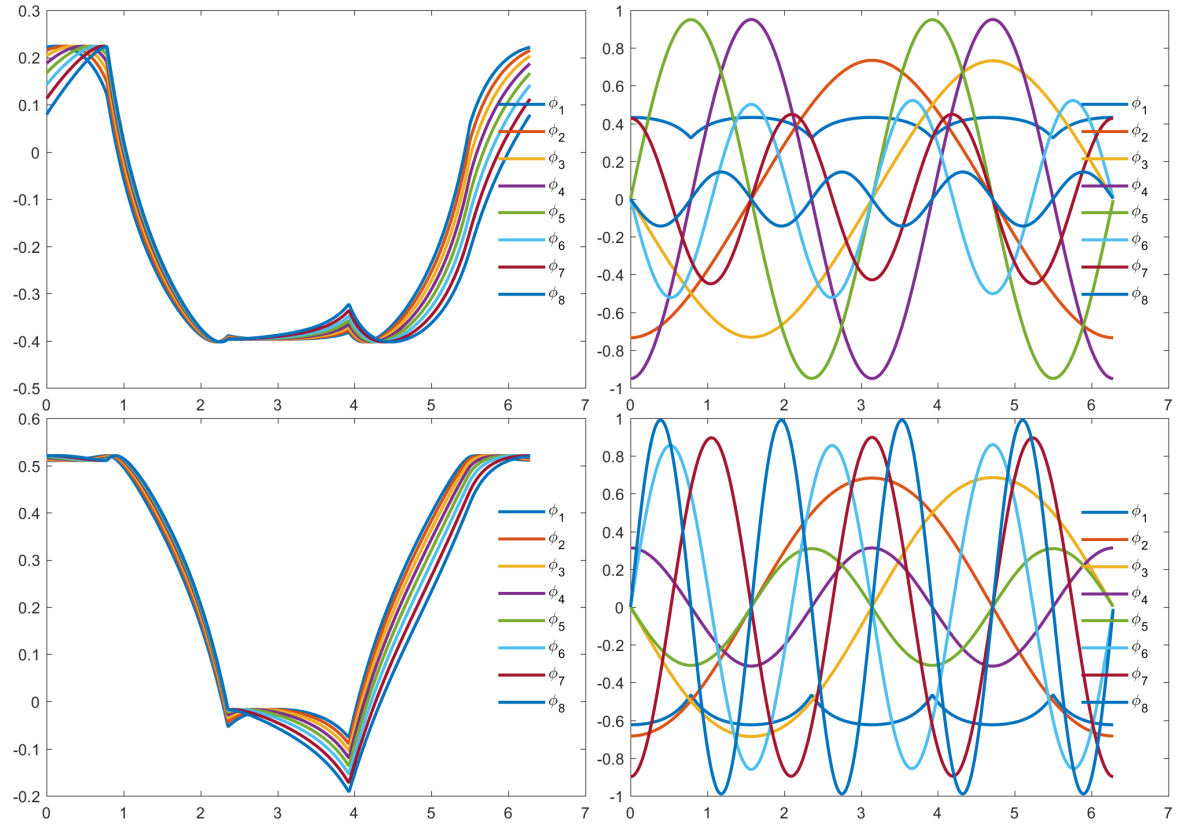


Figure 4.10: First 8 of 150 functions for the basis functions restricted at the square with side length 2, of the Helmholtz BVP with $k = 1$, pseudo-boundary as a square with side length 6. On the (left) MFS basis and on the (right) MFS-SVD basis. At the (top) real part and (bottom) imaginary part.

4.1.2 MFS-SVD in elliptic coordinates

As shown in example 4.3, elliptic domain case, the MFS-SVD$_P$ produces results that are superior to those of MFS-DIR. However, it still fails to significantly enhance the conditioning of the problem.

Let us now consider the elliptic coordinates formulation of the MFS-SVD described in section 3.3 to illustrate how to improve the MFS-SVD for elliptic domains. First, note that in the elliptic coordinates case the approximations for Lst_n, defined in (3.16), were obtained, as in the polar case, by simply evaluating $|\mathrm{Rst}_n^{(1)}(q,r)|$ at 10000 equally spaced points $r_i \in [r_{\partial\Omega}, \rho_{\partial\Omega}]$ and taking their maximum. In the following example, as we consider the case of two confocal ellipses, one representing the domain under study and the other the closed curve containing the source points, we simply evaluate $|\mathrm{Rst}_n^{(1)}(q,r)|$ at $r = r_E$ since in the elliptic coordinate system $r_{\partial\Omega} = \rho_{\partial\Omega}$.

Remark 4.2. *The numerical implementation was carried out using Matlab which does not have Mathieu functions implemented; to evaluate these functions we used the toolbox "Matlab Mathieu Functions Toolbox v.1.0 ", [32, 33], with a few modifications to increase the accuracy. The main modification performed in that code is on the definition of radial Mathieu functions. These functions have in their definitions an offset integer parameter s for the order of the Bessel functions. This parameter is set by default to 0 for the even-even, even-odd and odd-odd radial functions and set to 1 for the odd-even in the above toolbox. This, somehow, traditional selection of values for the parameter s is known to provide acceptable results, namely single precision accuracy, when n is not large [21]; for high values of n one can find the evaluation errors increasing without bound. Since for the proposed MFS-SVD approach the number of collocation points depends on the number of terms used to truncate the series* (3.5)*, single precision accuracy is not satisfactory, and we have therefore modified the code in order to allow a better parameter choice and therefore improve the accuracy to double precision.*

It has been observed that the use of larger offset values tends to improve accuracy [23]. We have therefore performed a modification of Cojocaru's code by introducing the possibility to change the offset parameter s and we select this parameter by considering the ideas of the "tuning algorithm" described in [21]. In figure 4.11 we illustrate the improvement we can obtain after this modification by considering the following example; in (3.5) *we take* $x = (6,0)$, $y = (16.45904993363967, 0)$ *and*

$q = 8.75$. *The elliptic coordinates for* x, y *are respectively* $(0.1682361183106064, 0)$ *and* $(1.682361183106065, 0)$. *We evaluate the* L^2*-error between the fundamental solution computed by Matlab built-in functions and the evaluation using* (3.5). *As we can see by using the original toolbox we would be limited to truncating the series around* $n = 10$ *as we get the best accuracy at this point. However, truncating at this level would give us only a maximum of* $N = 40$ *source points which can be very low to achieve good approximations. After modification of the code, we can reach machine precision accuracy and stability with increasing* N *as shown in figure 4.11.*

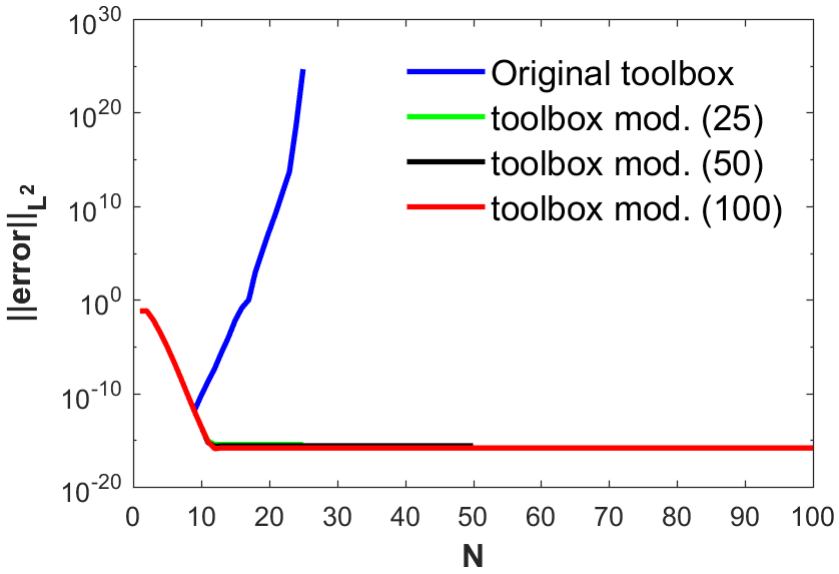


Figure 4.11: L^2-error of Cojocaru's code using $x = (6, 0)$, $y = (16.45904993363967, 0)$ and elliptic coordinates $(0.1682361183106064, 0)$, $(1.682361183106065, 0)$ in (3.5), where $q = 8.75$. The original code is the one that considers only a maximum of 25 terms in Mathieu functions. The modified code was tested with 25, 50 and 100 terms.

Example 4.5. *We naturally, consider the same case as in example 4.3, an elliptic domain case, since elliptic coordinates will be used.*

Let us first illustrate that the accuracy of MFS-DIR does not improve significantly with changes in the pseudo-boundary location. For that we have performed some tests by placing the pseudo-boundary at locations obtained by magnifying the original ellipse by the factors 1.05, 2.1, 3.15, 4.2, 5.25. From figure 4.12, we observe that in this particular case the location of the pseudo-boundary does not seem to improve that much the accuracy of MFS-DIR. The best we get is an accuracy of order 10^{-5}

at 400 source points for the case with a pseudo-boundary as an ellipse magnified by 2.1.

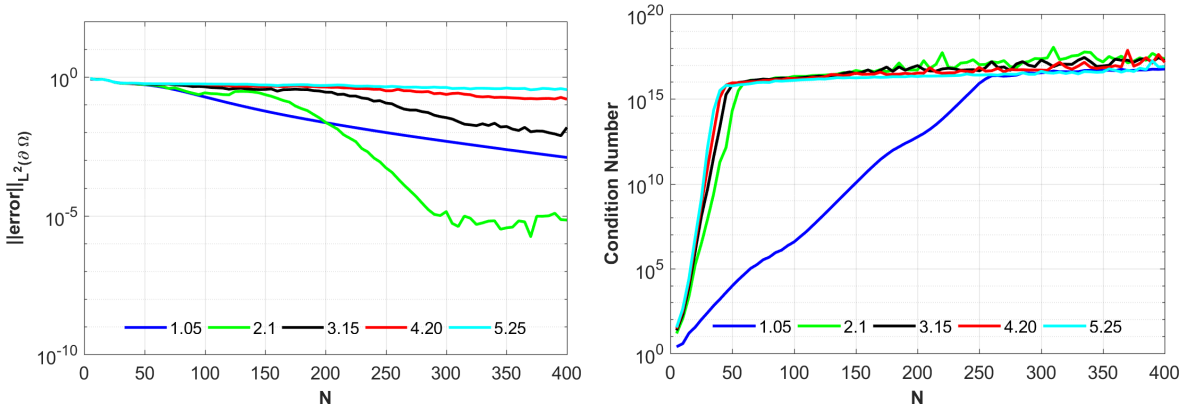


Figure 4.12: Plot of error in the boundary and condition number of MFS-DIR for Helmholtz equation on an elliptic domain given by $x^2/6^2+y^2=1$ with source points on an ellipse with semi-axes multiplied by different magnified factor, boundary values given by $g(x,y)=e^{i(x^2+y^2)}\sin(x+y)$.

Now, we consider the pseudo-boundary as an ellipse magnified by 10, similar assumption made in example 4.3. Figure 4.13 shows the comparison between the MFS-DIR versus the MFS-SVD$_P$ and MFS-SVD$_E$. Clearly we can see that the elliptic coordinates formulation is better when considering elliptic domains. We have the error of MFS-SVD$_E$ reaching order 10^{-14} while MFS-SVD$_P$ stabilizes at order 10^{-4} and MFS-DIR at order 1.

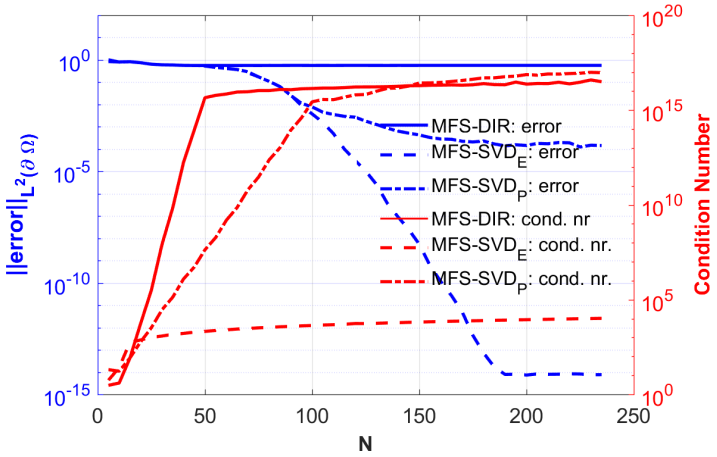


Figure 4.13: Plot of error at the boundary and condition number of MFS-DIR vs. MFS-SVD$_P$ and MFS-SVD$_E$, for Helmholtz equation for an elliptic domain given by $x^2/6^2+y^2=1$ with source points on an ellipse with semi-axes multiplied by 10, boundary values given by $g(x,y)=e^{i(x^2+y^2)}\sin(x+y)$.

Here also, the computational performance is clearly unfavourable to MFS-SVD$_E$ which is even worst when compared to MFS-SVD$_P$, for the reasons mentioned previously. The comparison of computation time for this case, are presented in table 4.2.

Table 4.2: Computation time, in seconds, as a function of the number of source points, for the case considered in figure 4.13.

N	**MFS-DIR**	**MFS-SVD$_P$**	**MFS-SVD$_E$**
5	0.065	387.648	3958.186
100	0.331	505.707	5052.500
200	0.712	634.791	39118.740

In figure 4.14, we can find the restrictions at the elliptic boundary of the basis functions. The Gram matrix is also presented, in figure 4.15 and we clearly see that MFS-SVD$_E$ is an orthogonal basis.

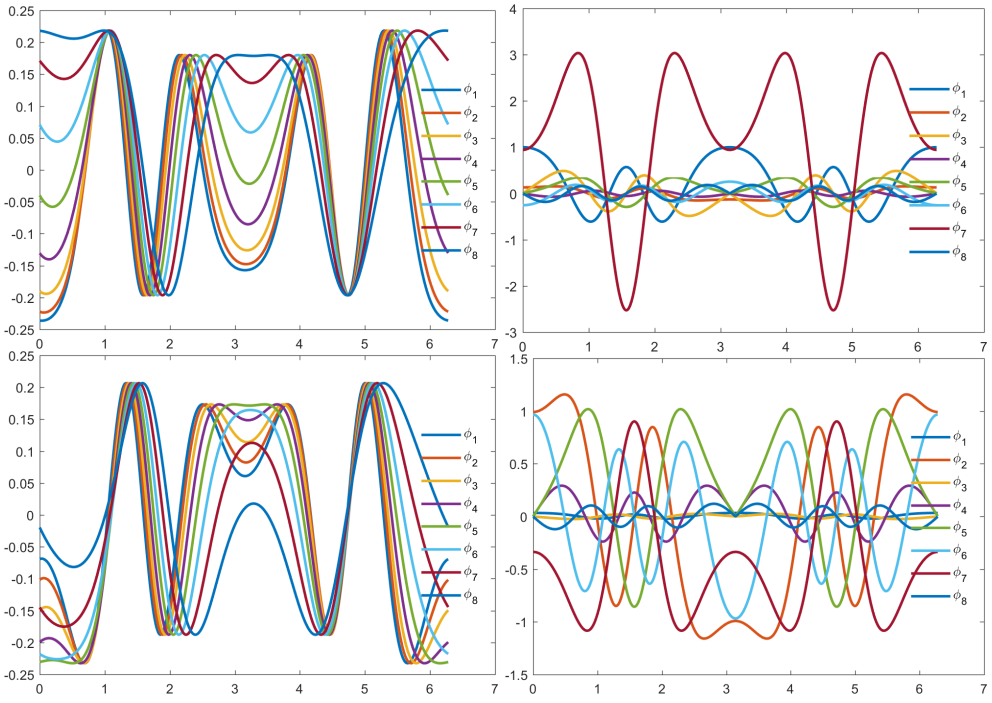


Figure 4.14: First 8 of 50 functions for the basis functions restrictions of Helmholtz BVP for case with boundary as an ellipse, $k = 1$, pseudo-boundary as an ellipse with semi-axes multiplied by 10. On the (left) MFS basis and on the (right) MFS-SVD$_E$ basis. At the (top) real part and (bottom) imaginary part.

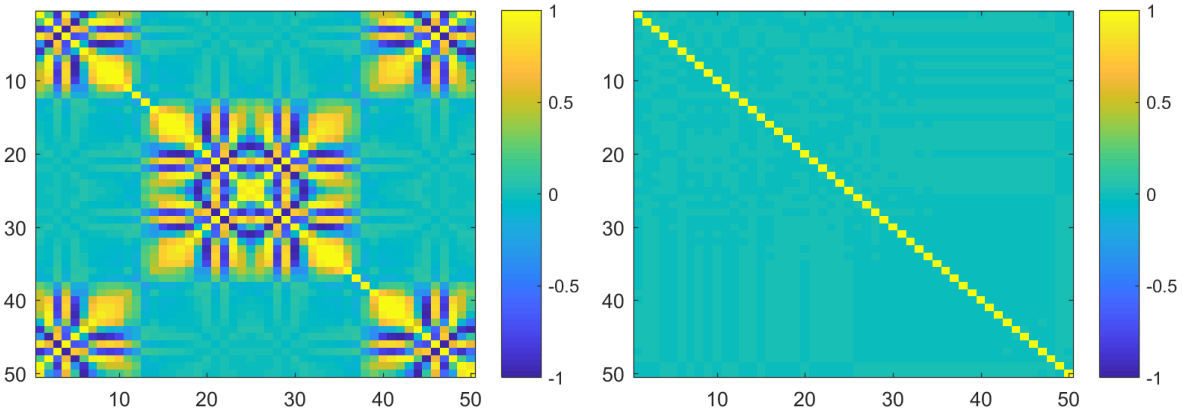


Figure 4.15: Images of Gram matrices for MFS (left) and MFS-SVD$_E$ (right) of the basis functions restricted at the boundary for the Helmholtz BVP in the elliptic boundary and pseudo-boundary as a ellipse magnified by 10.

Example 4.6. *Still considering the same problem, example 4.3, now we take the domain as a peanut [80] given by the parametrization*

$$\partial D = \left\{ 3\sqrt{\cos^2(t) + 0.25\sin^2(t)}\{\cos(t), \sin(t)\}, \quad 0 \leq t < 2\pi \right\}.$$

To continue the evaluation of the MFS-SVD$_E$ performance we consider three additional cases:

1. The case with boundary values given by $g(\mathrm{x},\mathrm{y}) = e^{\mathrm{i}(\mathrm{x}^2+\mathrm{y}^2)}\sin(\mathrm{x}+\mathrm{y})$;
2. the case with exact solution $u(\mathrm{x},\mathrm{y}) = -\frac{1}{4}Y_0(k|(\mathrm{x},\mathrm{y}) - \gamma|)$ with $\gamma = (5,0)$ and $\gamma = (400,0)$ at inside and outside the domain defined by the pseudo-boundary, respectively;
3. the plane wave $u(\mathrm{x},\mathrm{y}) = e^{\mathrm{i}k(\mathrm{x}+\mathrm{y})/\sqrt{2}}$.

In all cases the pseudo-boundary is an ellipse, the points will be distributed as in the previous cases, and we consider the wavenumber $k = 1$.

When dealing with elliptic coordinates it is essential to define appropriately the parameter σ, (3.1). In the case with two confocal ellipses the parameter σ is given by the ellipse that defines the domain. In this case, peanut domain, σ is defined from the ellipse with semi-axes a and b given by the distance between the origin and the intersection of the peanut-shaped curve with the x-axes and y-axes respectively, i.e. $a = 3$ and $b = 1.5$.

The numerical results for case 1 are presented in figure 4.16, for case 2 in figure 4.17 and for case 3 in figure 4.18. In all cases, we clearly see an improvement in conditioning when using the MFS-SVD$_P$ approach and even more when considering the

MFS-SVD$_E$, as expected. We can observe that although MFS-SVD$_P$ shows better conditioning at the beginning compared to MFS-DIR, the condition number still grows relatively quickly to reach an order of 10^{15} or higher in all cases; as illustrated in figure 4.16 this limits the improvement of the accuracy of MFS-SVD$_P$ and even deteriorates it as the condition number surpasses 10^{15}. On the other hand, MFS-SVD$_E$ shows a slowly growing condition number, which allows for an improvement in accuracy.

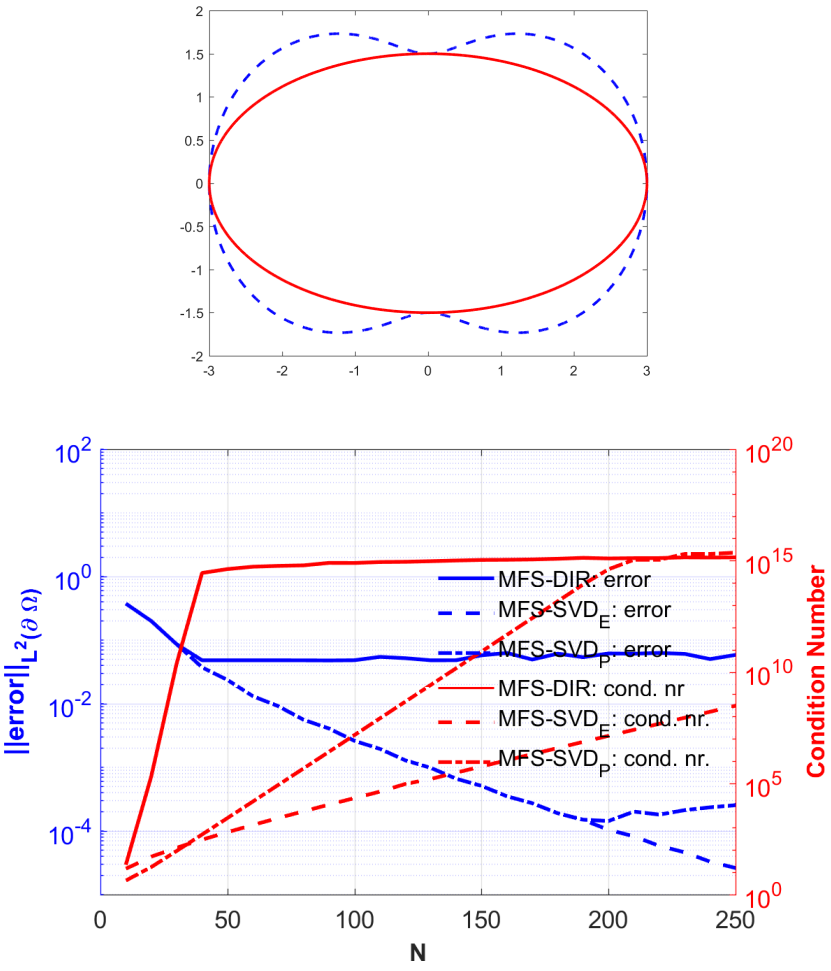
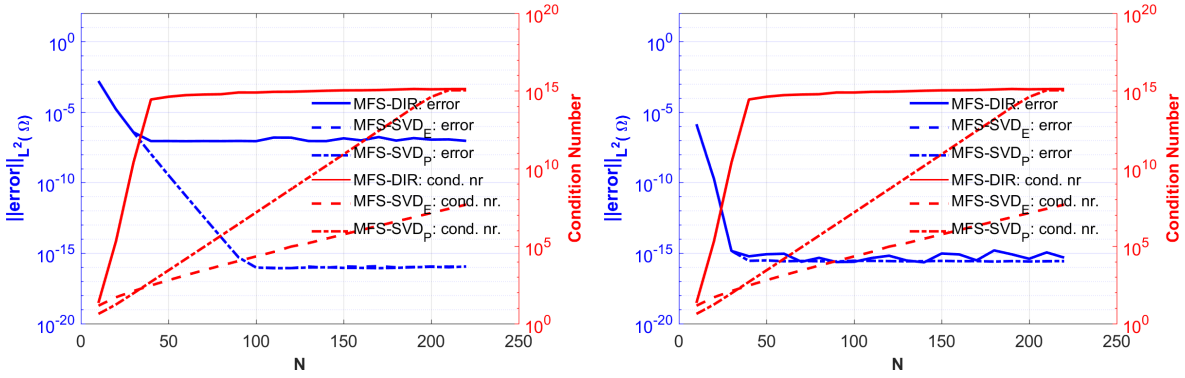


Figure 4.16: Peanut domain and the ellipse used to define the elliptical coordinates (top). Plot of error at the boundary and condition number of MFS-DIR vs. MFS-SVD$_P$ and MFS-SVD$_E$ (bottom), for Helmholtz equation on a peanut domain with source points on an ellipse, magnified from the ellipse defining the elliptic coordinates system, semi-axes multiplied by 10 and boundary values given by $g(x,y) = e^{i(x^2+y^2)}\sin(x+y)$.



(a) $\gamma = (5,0)$ (b) $\gamma = (400,0)$

Figure 4.17: Plot of error and condition number of MFS-DIR vs. MFS-SVD$_P$ and MFS-SVD$_E$, for Helmholtz equation on a peanut domain with source points on an ellipse, magnified from the ellipse defining the elliptic coordinates system, semi-axes multiplied by 10 and exact solution $u(x) = -\frac{1}{4}Y_0(k|x-\gamma|)$ with $k = 1$.

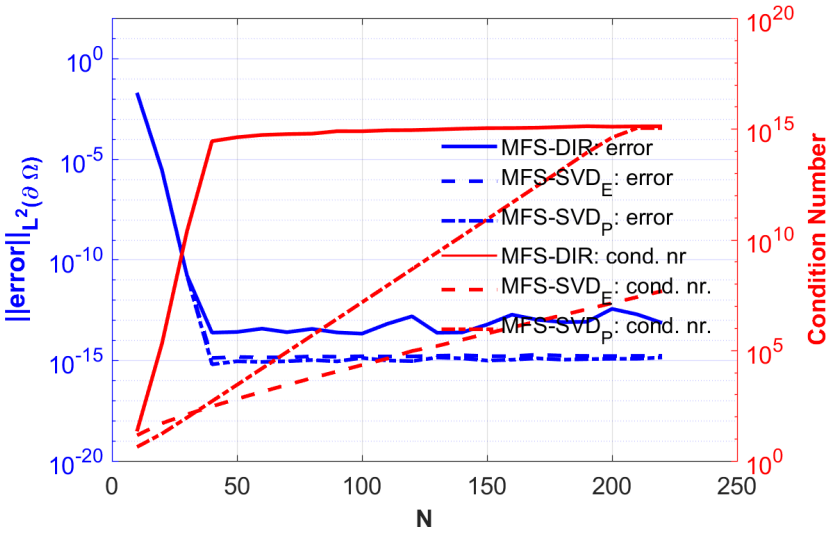


Figure 4.18: Plot of error at the boundary and condition number of MFS-DIR vs. MFS-SVD$_E$ and MFS-SVD$_P$, for Helmholtz equation on a peanut domain with source points on an ellipse, magnified by 10 from the ellipse defining the elliptic coordinates system and boundary values given by $u(\mathrm{x},\mathrm{y}) = e^{\mathrm{i}k(\mathrm{x}+\mathrm{y})/\sqrt{2}}$ and $k = 1$.

4.2 Numerical examples for biharmonic problems

We continue the illustration of MFS-SVD's performance by considering biharmonic BVP for polar coordinates formulation only. We take different planar domains, and since we have observed that for Helmholtz case the MFS-SVD is not too sensitive to the geometry of the pseudo-boundary, we take for simplicity a circle as pseudo-boundary for the majority of the cases. In some cases, for benchmark purposes, we consider different shapes for the pseudo-boundary. The performance is measured by selecting a set Q of points, which may be located at the boundary or inside the domain depending on the example. For this, we compute the error between u and the approximation $\bar{u}$ using the maximum absolute norm

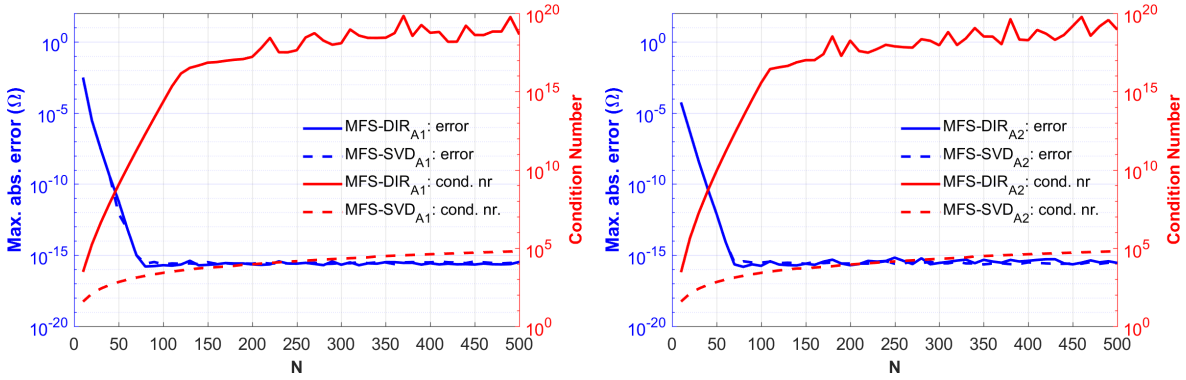
$$\varepsilon = \max_{z_i \in \mathsf{Q}} |u(z_i) - \bar{u}(z_i)|.$$

We still consider $M = 2N$, and to obtain ρ_Ω, we select 1000 points from the boundary $\partial\Omega$ and proceed as defined in (3.17); M and N will be distributed according to $\theta_m = 2\pi m/M$, $m = 1,2,\ldots,M$, $\alpha_n = 2\pi n/N$, $n = 1,2,\ldots,N$, respectively and in all cases the expansions (3.6) and (3.7) have been truncated at $T = 250$.

Example 4.7. *Let us consider the biharmonic problem,* (2.11) *with an exact solution given by* $u(x,s) = -1/(8\pi)|x-s|^2 \log|x-s|$, $s \notin \bar{\Omega}$. *The boundary conditions will be deduced from* u. *We look for a numerical approximation in the unit disk centered at the origin.*

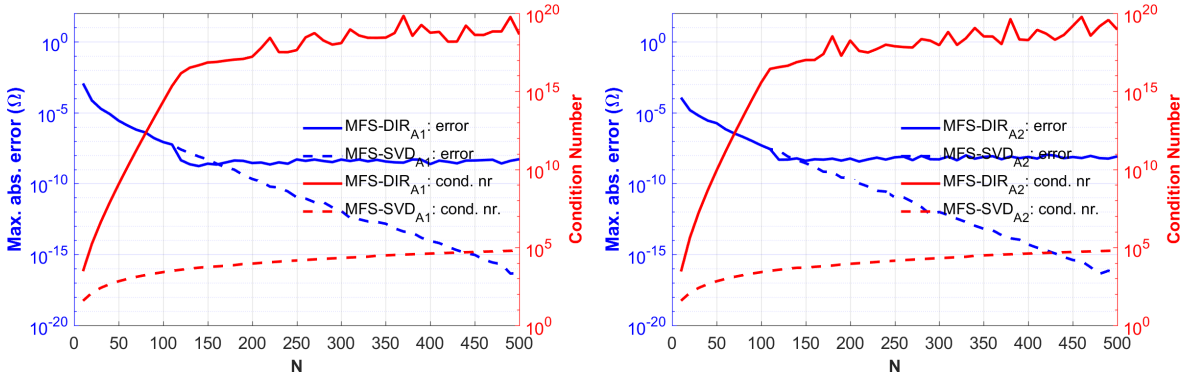
In this case, we consider the pseudo-boundary as a circle centered at origin with radius $\rho = 1.5$. First we take $s = (3,0)$ and present the results in figure 4.19, for both approaches, the classical MFS and the MFS-SVD and for both representations A_1 and A_2. We observe that each approach (the MFS-DIR or the MFS-SVD) have similar behavior for both representations A_1 and A_2, either for the accuracy or the condition number. However, we also observe an improvement of the condition number on both representations for the MFS-SVD, which does not translate into an improvement in the accuracy, as we reach machine precision accuracy with a low number of source points in MFS-DIR and in MFS-SVD. This improvement in the condition number will be converted into an improvement in the accuracy in those cases where the MFS-DIR needs more source points to reach approximations of machine precision magnitude. To illustrate this, let us consider $s = (1.1,0)$, inside the pseudo-boundary. In figure 4.20, we can observe that both representations of MFS-DIR do not improve accuracy beyond order 10^{-7}, whilst both approaches for the MFS-SVD reach machine precision order at 500 source points.

Let us have a look at the plot of the restrictions of the basis function to the boundary, as we did in the Helmholtz case. Figure 4.21 and the Gram matrix, figure 4.22, help us to understand the reasons for the ill-conditioning. Just by looking to the first 8 functions from the basis, we observe indistinguishable functions Φ_1 and Φ_2 before the change of basis, which is an indication of an almost linear dependency between



(a) Representation A_1 (b) Representation A_2

Figure 4.19: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the unit disk for $u(x,s) = -1/(8\pi)|x-s|^2\log|x-s|$ where $s=(3,0)$ and pseudo-boundary as a circle with $\rho = 1.5$, for representations A_1 and A_2.



(a) Representation A_1 (b) Representation A_2

Figure 4.20: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the unit disk for $u(x,s) = -1/(8\pi)|x-s|^2\log|x-s|$ where $s=(1.1,0)$ and pseudo-boundary as a circle with $\rho = 1.5$, for representations A_1 and A_2.

elements of the basis. The MFS Gram matrix is dense while the one for the MFS-SVD is sparse which illustrate that the functions from the MFS-SVD basis are almost orthogonal.

Still in the context of case with $s=(1.1,0)$, as the accuracy of the MFS-DIR is known to be dependent on the geometry of the pseudo-boundary, and on its location regarding the original boundary, we have performed sensitivities on the location of the pseudo-boundary to see how both approaches behave. We selected $\rho = 1.3, 1.6, 1.9, 5$ and the results for approximation A_2 only, (the case A_1 shows similar

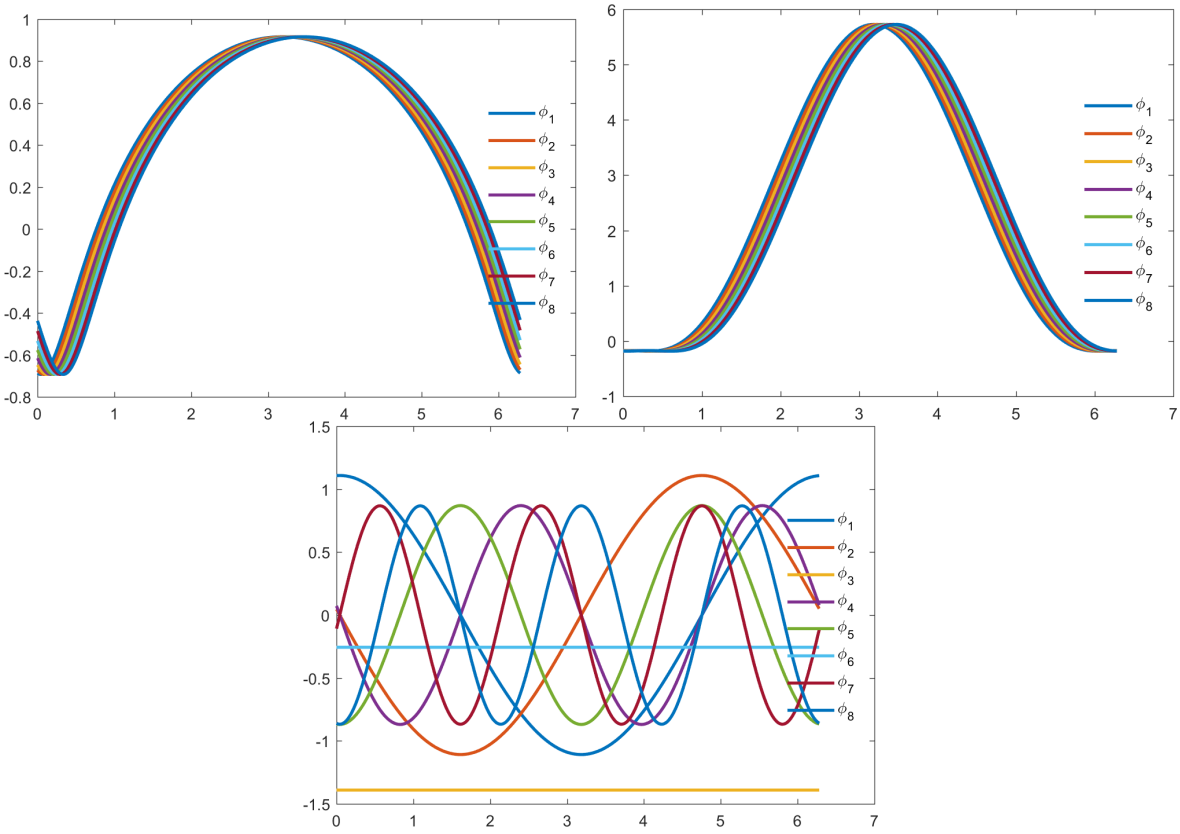


Figure 4.21: First 8 functions of 150 for the basis functions restrictions of the biharmonic BVP at the boundary for representation A_1. On the (left) MFS basis Φ_1 and on the (middle) MFS basis Φ_2 and on (right) MFS-SVD basis.

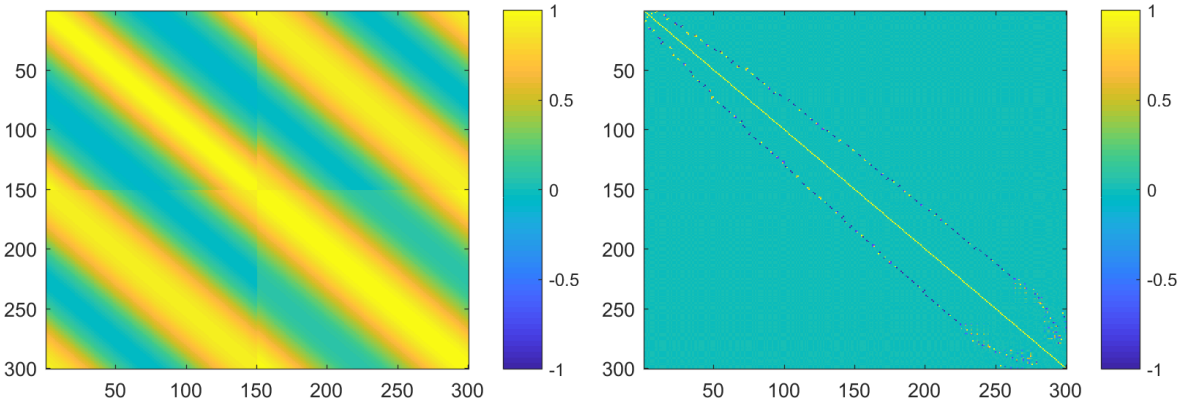
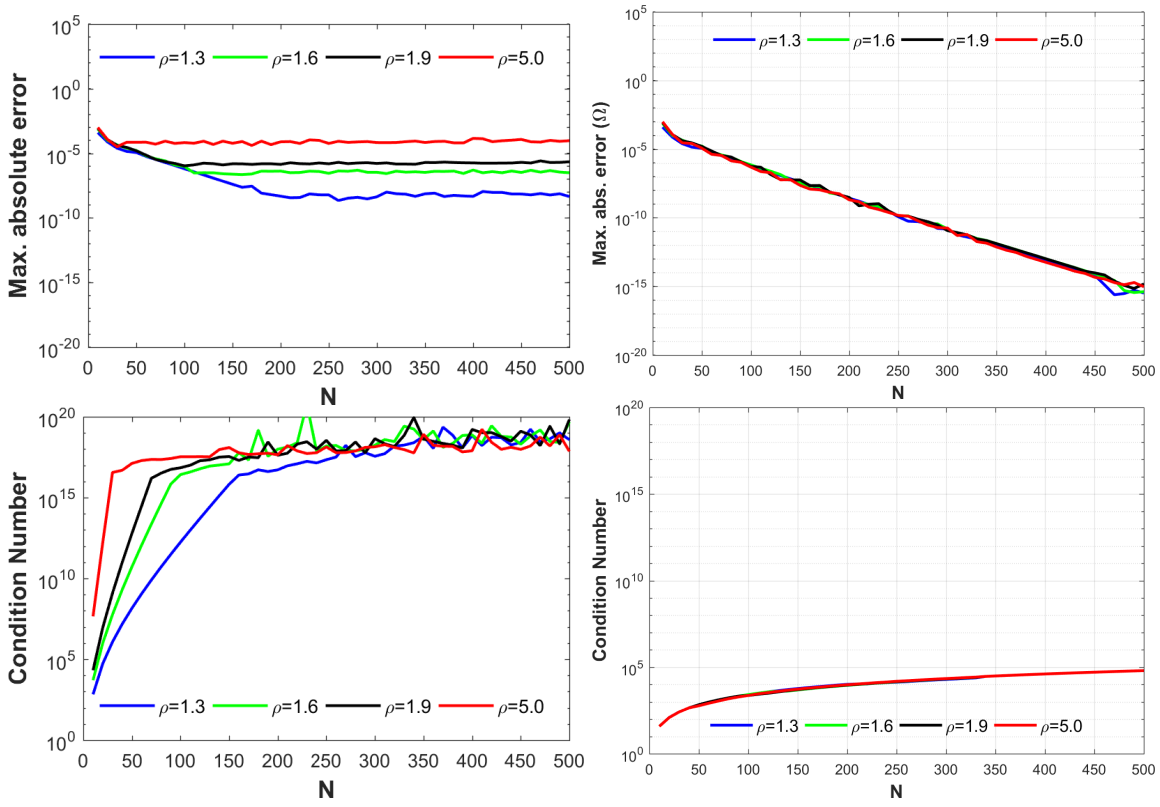


Figure 4.22: Images of Gram matrices for MFS (left) and MFS-SVD (right) of the basis functions restricted at the circle for the biharmonic BVP with pseudo-boundary as a circle with $\rho = 1.5$.

behavior) are presented in figure 4.23. As expected, the accuracy of the MFS-DIR$_{A_2}$, deteriorates as we move far from the boundary. The case with $\rho = 1.3$ is slightly better than the case we took as reference for the illustrations in the previous results, $\rho = 1.5$, which shows that there is room for improvements on the accuracy of the MFS-DIR by looking for the best radius for ρ, similarly to [29]. However, the MFS-SVD shows consistently the same accuracy regardless of the location of the pseudo-boundary. In fact, as observed already for the Helmholtz case, the MFS-SVD seems to illustrate that if an adequate set of basis functions is selected to span the space of approximations, the location of the pseudo-boundary may not impact much the accuracy.



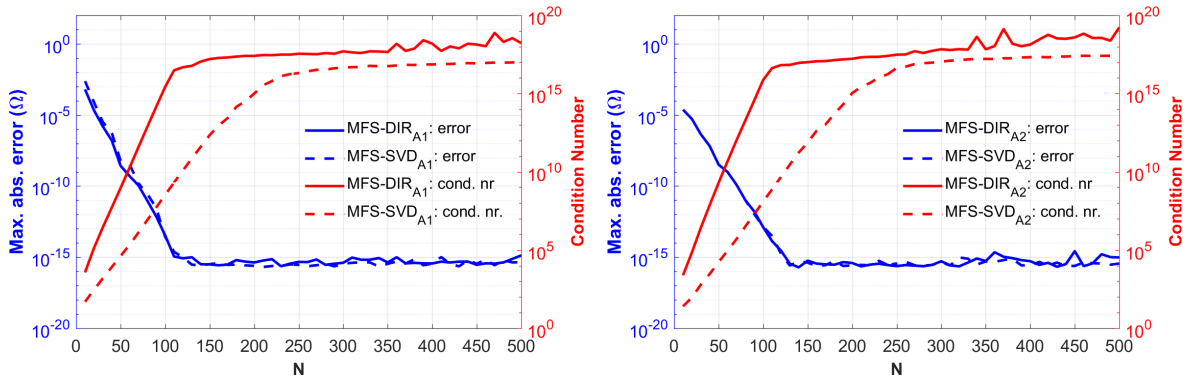
(a) MFS-DIR$_{A_2}$ (b) MFS-SVD$_{A_2}$

Figure 4.23: Maximum absolute error and condition number for different pseudo-boundary locations for both MFS-DIR$_{A2}$ (left) and MFS-SVD$_{A2}$ (right) for biharmonic problem on the unit disk with for $u(x,s) = -1/(8\pi)|x-s|^2 \log|x-s|$ where $s = (1.1, 0)$ and pseudo-boundary at $\rho = 1.3, 1.6, 1.9, 5$, for representation A_2.

Example 4.8. *We consider the same problem as in example 4.7 but the domain is now the 3-petal rose with boundary parametrized by*

$$(x,y) = r(\theta)\big(\cos(\theta), \sin(\theta)\big), \quad r(\theta) = 1 + \cos^2(3\theta/2), \quad 0 \leq \theta < 2\pi.$$

We consider two pseudo-boundaries, a circle and a 3-petal rose. If we take $s = (3,0)$, the results in figure 4.24 show similar accuracies between both approaches with slight improvements on the condition number of the MFS-SVD cases. Let us now take the source point $s = (-2,0)$, $s \in \Omega' \setminus \bar{\Omega}$. The results are presented in figure 4.25. We can observe the gain in condition number being translated into gain in accuracy, with the MFS-SVD cases reaching 10^{-15} against order 10^{-10} for the MFS-DIR cases and this is independent of the geometry of the pseudo-boundary, as we can see in figure 4.26, where the pseudo-boundary is obtained by a dilation of the original boundary by factor of 2.5.



(a) Representation A_1

(b) Representation A_2

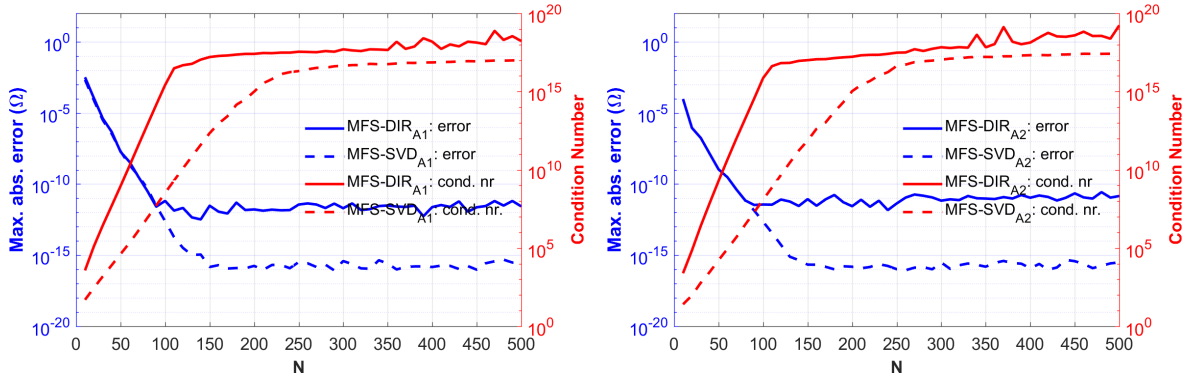
Figure 4.24: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the 3-petal rose domain for $u(x,s) = -1/(8\pi)|x-s|^2 \log|x-s|$ where $s = (3,0)$ and pseudo-boundary as circle with $\rho = 2.5$, for representations A_1 and A_2.

Example 4.9. *Still considering the example* (4.7)*, we take as domain the square* $]-1,1[\times]-1,1[$.

Let us consider for this problem as a pseudo-boundary the circle with radius $\rho = 2.0$. We analyse only the problem with the singularity placed in $\Omega' \setminus \bar{\Omega}$. The collocation points are determined as in Helmholtz case with the particularity that the selection must include the corner points. The results in figure 4.27 show, as in previous cases, better conditioning for MFS-SVD cases, which then leads to a better accuracy for both representations.

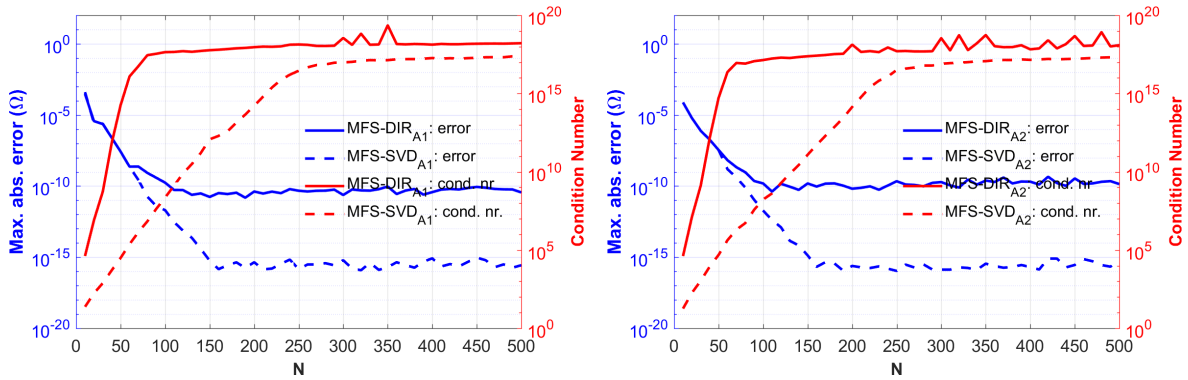
Example 4.10. *We consider now, the biharmonic problem* (2.11) *from [29] with domains considered in the previous cases, the disk, the 3-petal rose and the square.*

$$u(x,y) = x^2y^3, \quad \frac{\partial u(x,y)}{\partial \boldsymbol{n}} = (\nabla(x^2y^3)) \cdot \boldsymbol{n}, \quad (x,y) \in \partial\Omega. \tag{4.1}$$



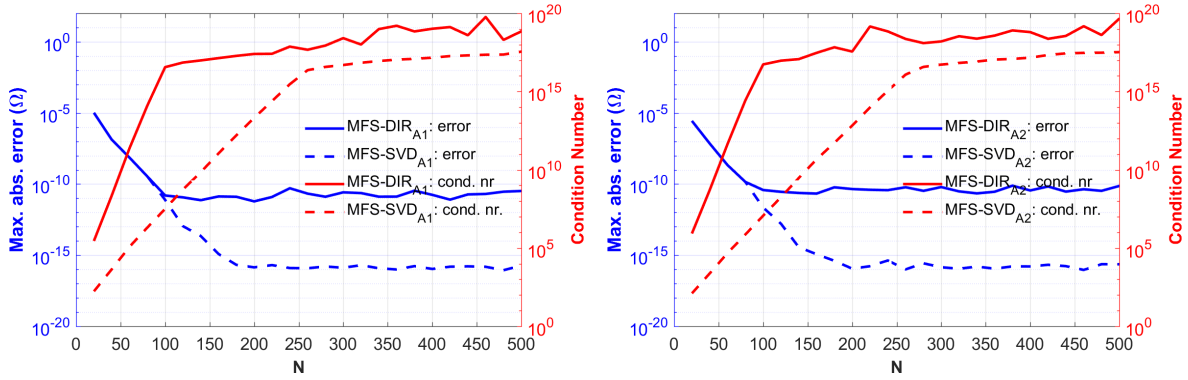
(a) Representation A_1 (b) Representation A_2

Figure 4.25: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the 3-petal rose domain for $u(x,s) = -1/(8\pi)|x-s|^2 \log|x-s|$ where $s = (-2,0)$ and pseudo-boundary as circle of $\rho = 2.5$, for representations A_1 and A_2.



(a) Representation A_1 (b) Representation A_2

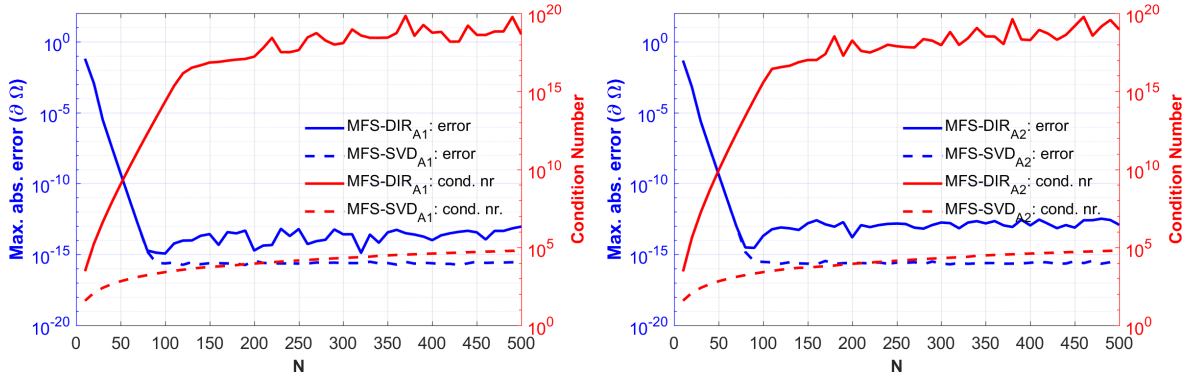
Figure 4.26: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the 3-petal rose domain for $u(x,s) = -1/(8\pi)|x-s|^2 \log|x-s|$ where $s = (-2,0)$ and pseudo-boundary as 3-petal rose domain dilated by factor of $\rho = 2.5$, for representations A_1 and A_2.



(a) Representation A_1 (b) Representation A_2

Figure 4.27: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the square for $u(x,s) = -1/(8\pi)|x-s|^2 \log|x-s|$ where $s = (1.5, 0)$ and pseudo-boundary as a circle with $\rho = 2.0$, for representations A_1 and A_2.

We address the numerical solution of this problem by evaluating the error at the boundary, since the exact solution is not available. Taking the unitary disk as a domain, the results are presented in figure 4.28 and show good accuracy for both approaches with a small improvement and much more stable solution for MFS-SVD cases, as the condition number is better controlled.

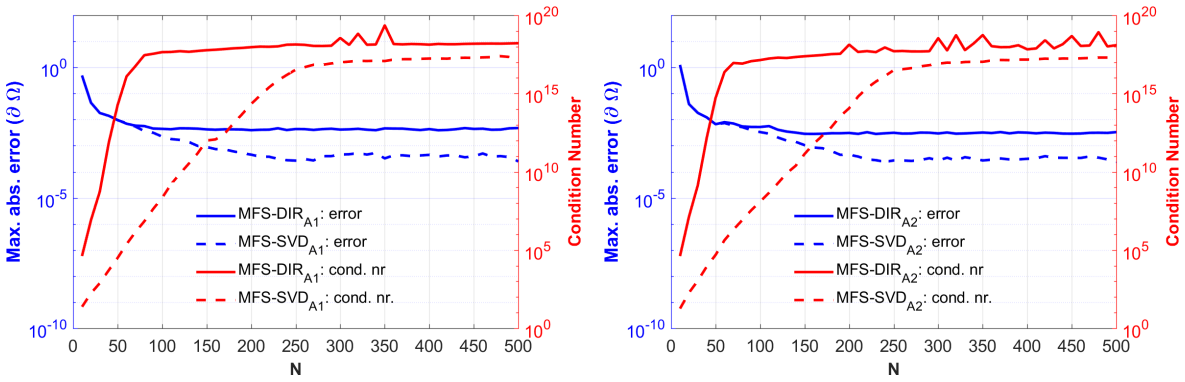


(a) Representation A_1 (b) Representation A_2

Figure 4.28: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the unitary disk domain for $u(x,y) = x^2y^3$ and pseudo-boundary as a circle with $\rho = 1.5$, for representations A_1 and A_2.

In figure 4.29, we show the results when considering the 3-petal rose domain with the pseudo-boundary taken as a dilated 3-petal rose, the case with the circle as

pseudo-boundary behaves quite similarly as this one for the MFS-SVD.



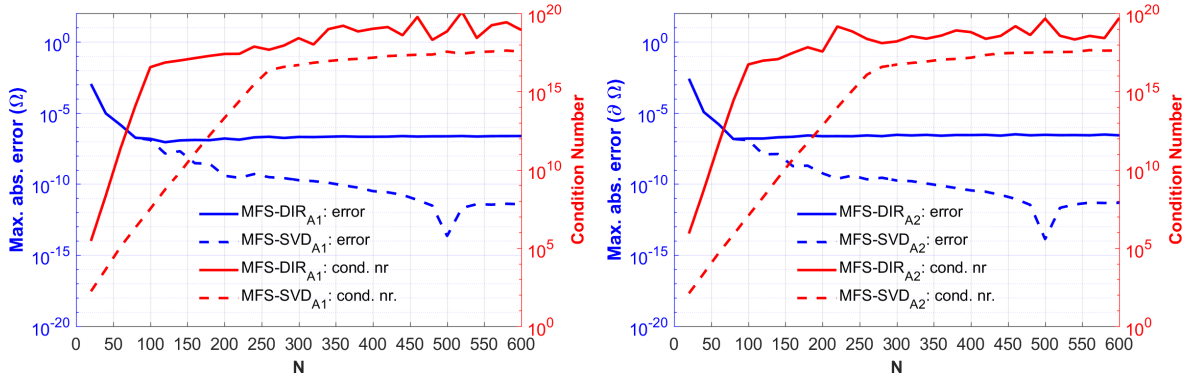
(a) Representation A_1 (b) Representation A_2

Figure 4.29: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the 3-petal rose domain for $u(x,y) = x^2y^3$ and pseudo-boundary as a 3-petal rose domain dilated by a factor of $\rho = 2.5$, for representations A_1 and A_2.

The behavior of the error is not that good in this case. In fact this is not a surprise as we have mentioned already in Helmholtz case that the MFS-SVD approach performs better if we use the appropriate curvilinear system of coordinates to describe the domain with the adequate addition theorem. The addition theorem we are using is adapted for the disk. In all other domains far from the disk, we may observe improvements but not as marked as in the disk case.

The example 4.10 was considered in [29] with a pseudo-boundary taken as a 3-petal rose domain at a constant distance h, from the original domain; in that work they have results reaching order 10^{-11} for their called optimal approach at 400 source points by placing the pseudo-boundary at distance $h = 0.134$. We remind that due to restrictions associated with the convergence of the addition theorem we cannot place a pseudo-boundary so close to the domain. Additionally, all the illustrations we have performed indicate that this would not improve the MFS-SVD, which means no influence in the results is expected.

Finally, we present in figure 4.30 the results for the square domain. We observe as always improvements in accuracy for MFS-SVD. In this particular case although the condition number is not so well controlled we manage to reach accuracy of 10^{-12} for MFS-SVD against 10^{-7} for classical MFS.



(a) Representation A_1 (b) Representation A_2

Figure 4.30: Plot of maximum absolute error in the domain and condition number of MFS-DIR vs. MFS-SVD for biharmonic problem on the square domain for $u(x,y) = x^2y^3$ and pseudo-boundary as a circle with radius $\rho = 2$, for representations A_1 and A_2.

Chapter 5

Conclusions

In this work generalizations of MFS-SVD, originally introduced in [12] for planar Laplace equation, are presented for two BVP: The Helmholtz and the biharmonic in $\mathbb{R}^2$. We have developed the MFS-SVD approach from the associated addition theorems and illustrated that when applied to both problems, it performs also very well as observed in Laplace BVP. In the Helmholtz case we have extended the MFS-SVD even further by proposing a formulation in elliptic coordinates on top of the formulation in polar coordinates, originally proposed; this has shown to be an advantage to improve the conditioning of the MFS in case of BVP with elliptic boundaries or other domains with geometry close to an ellipse. This formulation requires the introduction of Mathieu functions, as basis functions to represent the solutions in those domains, instead of Bessel functions which better fit the problems with disks domains. This work shows that by following this approach we can effectively control the condition number and therefore improve the accuracy, being able to use a higher number of source points without compromising the conditioning of the linear system.

Another contribution of this work is on the discussion about the numerical evaluation of Mathieu functions. These functions sometimes considered by some authors as intractable, and many times reported as showing important cancellation errors, are recognized as the ideal functions to address problems involving elliptic domains. We can use Mathieu functions with high accuracy, due to the adaptation of a code from a Matlab toolbox, allowing for a better choice of the parameter s for each case.

The biharmonic BVP is used to illustrate the feasibility and the changes in the MFS-SVD workflow needed to be performed for high order problems. In fact in this work we emphasize that the approach applied to Helmholtz case, which is similar to what was done in Laplace case, relies on a factorization of an operator that generates the MFS collocation linear system (2.15) and the SVD decomposition of its ill-conditioned factor; for higher order problems such as the biharmonic BVP, proceeding by the same steps, with the factorization of the operator that generates the MFS collocation linear system (2.15), leads to a singular factors. In this way, one should consider an operator that maps the basis functions and properly alter these basis functions, imposing the boundary conditions using these new basis functions, rather than factorize the MFS collocation linear system. In the case of Dirichlet boundary

conditions for Laplace and Helmholtz, both approaches lead to the same result, so one of the main innovation of this work is to establish a general approach for higher order PDEs and general BCs.

Another contribution is to illustrate that MFS-SVD is independent of the ansatz. In fact, we considered two different ansatz with two different addition theorems with results showing similar accuracy and similar conditioning by the MFS-SVD approach. We also illustrate numerically that MFS-SVD approaches are independent of the pseudo-boundary, while it affects the results of MFS. This broadens the feasibility and the application of MFS-SVD to a larger family of BVP, in particular, with higher order.

The proposed extension of [12] reinforces the idea that the MFS-SVD is suited for several boundary value problems as long as an addition theorem is known. Therefore future works will continue focus on illustrating this case, namely by assessing the MFS-SVD performance for the biharmonic equation in $\mathbb{R}^3$, the polyharmonic equation in $\mathbb{R}^2$ and $\mathbb{R}^3$, the propagation of elastic waves in $\mathbb{R}^2$ and $\mathbb{R}^3$ and others. Among those research topics the propagation of elastic waves either in $\mathbb{R}^2$ or in $\mathbb{R}^3$ is possibly the one in which I have already some encouraging results. This problem is different from Helmholtz or biharmonic as the formulation uses tensors at the basis function instead of scalars. This fact brings additional complexity but preliminary results shows as well some improvements in the condition number.

Conclusões

Neste trabalho generaliza-se a aplicação do MFS-SVD, desenvolvido em [12] para a equação de Laplace no plano, a dois problemas de valor de fronteira nomeadamente, o problema de Helmholtz e o problema biharmónico ambos no plano. Desenvolvemos na tese que o MFS-SVD tem um desempenho tão bom quanto o observado para a equação de Laplace em determinados domínios. Para o caso do problema de Helmholtz uma novidade do trabalho é a formulação do MFS-SVD em coordenadas elípticas; isto representa uma vantagem para os casos em que pretendemos melhorar o condicionamento do MFS em domínios elípticos ou com geometrias próximas à elipse. Para o efeito foi necessário introduzir funções de Mathieu como funções de base para representar as soluções neste domínio em vez de funções de Bessel que se ajustam melhor aos problemas em discos. O trabalho mostra que seguindo esta abordagem podemos efetivamente controlar melhor o número de condição em tais casos e portanto melhorar a precisão, pois nos permite usar um número maior de pontos fonte sem comprometer o condicionamento do sistema linear.

Outra contribuição deste trabalho está na discussão sobre a avaliação numérica das funções de Mathieu. Essas funções, algumas vezes consideradas por alguns autores como intratáveis, e muitas vezes relatadas como apresentando erros importantes de cancelamento, são reconhecidas como as funções ideais para abordar problemas envolvendo domínios elípticos. No entanto, a sua utilização é refreada, por vezes, devido a inexistência de implementação no Matlab. Para este trabalho as funções de Mathieu foram calculadas, com recurso à um código de uma caixa de ferramentas do Matlab desenvolvida por terceiros, mas que a partida não é muito eficiente, pois devolve apenas precisão simples e para um número pequeno de pontos fonte. A novidade do nosso trabalho consistiu em adaptar o código de maneiras a garantir maior precisão. Basicamente os ajustes ao código que sugerimos permitem uma escolha melhor do parâmetro s usado na definição, e devolvendo em consequência uma precisão maior.

Relativamente ao problema biharmónico realçar que este caso nos permite ilustrar a viabilidade e as alterações necessárias a serem realizadas na metodologia do MFS-SVD para problemas de ordem superior. Portanto, neste trabalho enfatizamos que a abordagem aplicada ao caso de Helmholtz, que é semelhante ao que foi feito no caso de Laplace, depende da fatorização de um operador que gera o sistema linear de colocação do MFS e da decomposição em valores singulares de seu fator mal condicionado; para problemas de ordem superior, como o biharmónico, se prosseguirmos com a fatorização do operador que gera o sistema linear de colo-

cação do MFS, obteremos factores singulares o que não ajudaria na melhoria do mau condicionamento. Logo a melhor abordagem, ilustrada neste trabalho, é considerar um operador que mapeia as funções de base e altera adequadamente essas funções de base, impondo as condições de fronteira usando essas novas funções de base, em vez de fatorizar o sistema linear de colocação do MFS. No caso dos problemas de valor de fronteira de Dirichlet, tanto para a equação de Laplace como para a equação de Helmholtz, ambas as abordagens levam ao mesmo resultado; então uma das principais inovações deste trabalho é estabelecer uma abordagem geral para equações diferenciais parciais de ordem superior e condições de fronteira gerais. Ainda usando o problema biharmónico, ilustramos que MFS-SVD é independente do tipo de aproximação numérica usada. De fato, nós consideramos duas aproximações numéricas diferentes com dois teoremas de adição diferentes, que dão origem a resultados em termos de precisão e condicionamento semelhantes usando o método MFS-SVD. Também ilustramos numericamente que as abordagens MFS-SVD são independentes da fronteira artificial selecionada, o que não é o caso quando usamos o método clássico. Isso amplia a viabilidade e a aplicação do MFS-SVD para uma família maior de problemas de fronteira, em particular, com ordem superior.

A extensão proposta neste trabalho do que foi feito em [12], reforça a ideia de que o MFS-SVD é adequado para vários problemas de valor de fronteira desde que se conheça a expansão/teorema de adição para a representação da solução fundamental. Portanto, trabalhos futuros continuarão focando em ilustrar a viabilidade deste método, MFS-SVD, para problemas como o da equação biharmónica no espaço, o problema da equação polyharmonica, a propagação de ondas elásticas em $\mathbb{R}^2$ e $\mathbb{R}^3$ e outros. Destes problemas ora mencionados, o da propagação de ondas elásticas em $\mathbb{R}^2$ e $\mathbb{R}^3$ é o que destacamos pois existem já alguns resultados encorajadores. Este caso apresenta alguma complexidade adicional pois as funções de base são tensores ao contrário do caso Helmholtz ou biharmonico onde temos funções escalares.

Bibliography

[1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. National Bureau of Standards, 1964.

[2] Y. Altman. export_fig (https://github.com/altmany/export_fig/releases /tag/v3.47). *GitHub. Retrieved December 9,*, 2024.

[3] C. J. S. Alves. On the choice of source points in the method of fundamental solutions. *Engineering Analysis with Boundary Elements*, 33(12):1348 – 1361, 2009.

[4] C. J. S. Alves and P. R. S. Antunes. The method of fundamental solutions applied to some inverse eigenproblems. *SIAM Journal on Scientific Computing*, 35:A1689–A1708, 2013.

[5] C. J. S. Alves and V. M. A. Leitão. Crack analysis using an enriched MFS domain decomposition technique. *Engineering Analysis with Boundary Elements*, 30 (3):160–166, 2006.

[6] C. J. S. Alves and A. L. Silvestre. Density results using Stokeslets and a method of fundamental solutions for the Stokes equations. *Engineering Analysis with Boundary Elements*, 28:1245–1252, 2004.

[7] C. J. S. Alves and S. S. Valtchev. Numerical comparison of two meshfree methods for acoustic waves scattering. *Engineering Analysis with Boundary Elements*, 29(4):371–382, 2005.

[8] C. J. S. Alves, N. F. M. Martins, and S. S. Valtchev. Extending the method of fundamental solutions to non-homogeneous elastic wave problems. *Applied Numerical Mathematics*, 115:299–313, 2017.

[9] P. R. S. Antunes. A numerical algorithm to reduce ill-conditioning in meshless methods for the Helmholtz equation. *Numerical Algorithms*, 79(3):879–897, 2018.

[10] P. R. S. Antunes. Reducing the ill conditioning in the method of fundamental solutions. *Advances in Computational Mathematics*, 44:351–365, 2018.

[11] P. R. S. Antunes. *A short introduction to numerical methods for elliptic partial differential equations, Lecture notes*. Universidade Aberta, 2021.

[12] P. R. S. Antunes. A well-conditioned method of fundamental solutions for Laplace equation. *Numerical Algorithms*, 91:1381–1405, 2022.

[13] P. R. S. Antunes, H. Calunga, and P. Serranho. Improving the conditioning of the method of fundamental solutions for the Helmholtz equation on domains in polar or elliptic coordinates. *Applied Mathematics and Computation*, 482: 128969, 2024.

[14] P. R. S. Antunes, V. Santos, and P. Serranho. The MFS-SVD method for the Laplace equation in three dimensions. *SIAM Journal on Scientific Computing*, 47(1):A454–A471, 2025.

[15] P. R. S. Antunes, H. Calunga, and P. Serranho. On improving the conditioning of the method of fundamental solutions for biharmonic BVPs in 2D domains. *Mathematics and Computers in Simulation*, 243:237–250, 2026.

[16] S. Barbeiro and P. Serranho. *The Method of Fundamental Solutions for the Direct Elastography Problem in the Human Retina*, pages 87–101. Springer International Publishing, Cham, 2020.

[17] A. H. Barnett and T. Betcke. Stability and convergence of the method of fundamental solutions for Helmholtz problems on analytic domains. *Journal of Computational Physics*, 227(14):7003–7026, 2008.

[18] J. R. Berger and A. Karageorghis. The method of fundamental solutions for heat conduction in layered materials. *International Journal for Numerical Methods in Engineering*, 45(11):1681 – 1694, 1999.

[19] J. R. Berger and A. Karageorghis. The method of fundamental solutions for heat conduction in layered materials. *Numerical Methods in Engineering*, 45: 1681–1694, 1999.

[20] J. R. Berger and A. Karageorghis. The method of fundamental solutions for layered elastic materials. *Engineering Analysis with Boundary Elements*, 25: 877–86, 2001.

[21] M. M. Bibby and A. F. Peterson. *Accurate Computation of Mathieu Functions*. Morgan and Claypool, 2013.

[22] A. Bogomolny. Fundamental solutions method for elliptic boundary value problems. *SIAM Journal on Numerical Analysis*, 22(4):644–669, 1985.

[23] A. L. V. Buren and J. E. Boisvert. Accurate calculation of modified Mathieu functions of integer order. *Quarterly of Applied Mathematics*, 65:1–23, 2007.

[24] J. Carrillo de la Plata and P. Roux. Nonlinear partial differential equations in neuroscience: from modelling to mathematical theory. *Mathematical Models and Methods in Applied Sciences*, 35(2):403–584, 2025.

[25] J. Chamorro-Serven, R. Dubois, and Y. Coudière. Exploring possible choices of the Tikhonov regularization parameter for the method of fundamental solutions in electrocardiography. *Computing in Cardiology*, 44, 2017.

[26] C. S. Chen and A. Karageorghis. Novel method of fundamental solutions formulation for polyharmonic BVPs. *Mathematics and Computers in Simulation*, 227:85–102, 2025.

[27] C. S. Chen, S. Y. Reutskiy, and V. Y. Rozov. The method of fundamental solutions and its modifications for electromagnetic field problems. *Computer Assisted Methods in Engineering and Science*, 16:21–33, 2009.

[28] C. S. Chen, A. Karageorghis, and Y. Li. On choosing the location of the sources in the MFS. *Numerical Algorithms*, 72(1):107 – 130, 2016.

[29] C. S. Chen, A. Noorizadegan, D. L. Young, and C.-S. Chen. On the determination of locating the source points of the MFS using effective condition number. *Journal of Computational and Applied Mathematics*, 423, 2023.

[30] J. T. Chen, C. S. Wu, Y. T. Lee, and K. H. Chen. On the equivalence of the Trefftz method and method of fundamental solutions for Laplace and biharmonic problems. *Computers & Mathematics with Applications*, 53:851–879, 2007.

[31] A. H. D. Cheng and Y. Hong. An overview of the method of fundamental solutions - solvability, uniqueness, convergence, and stability. *Engineering Analysis with Boundary Elements*, (120):118–152, 2020.

[32] E. Cojocaru. Mathieu functions toolbox v.1.0. `https://www.mathworks.com/matlabcentral/fileexchange/22081-mathieu-functions-toolbox-v-1-0`, `MATLABCentralFileExchange.RetrievedOctober9,2023`, 2008.

[33] E. Cojocaru. Mathieu functions computational toolbox implemented in Matlab. `arXiv:0811.1970v2`, 2008.

[34] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer, 2019.

[35] F. Dou, Z.-C. Li, C. S. Chen, and Z. Tian. Analysis on the method of fundamental solutions for biharmonic equations. *Applied Mathematics and Computation*, (339):346–366, 2018.

[36] M. M. Doyley. Model-based elastography: A survey of approaches to the inverse elasticity problem. *Physics in Medicine and Biology*, 57:R35–R73, 2012.

[37] D. Erricolo. Acceleration of the convergence of series containing Mathieu functions using shanks transformation. *IEEE Antennas and Wireless Propagation Letters*, 2:58–61, 2003.

[38] G. Fairweather and A. Karageorghis. The method of fundamental solutions for elliptic boundary value problems. *Advances in Computational Mathematics*, 9(1):69–95, 1998.

[39] C. M. Fan, Y. K. Huang, C. S. Chen, and S. R. Kuo. Localized method of fundamental solutions for solving two-dimensional Laplace and biharmonic equations. *Engineering Analysis with Boundary Elements*, 101:188 – 197, 2019.

[40] G. Feng, M. Li, and C. S. Chen. On the ill-conditioning of the MFS for irregular boundary data with sufficient regularity. *Engineering Analysis with boundary elements*, 18:98–102, 2014.

[41] G. B. Folland. *Introduction to Partial Differential Equations*. Princeton University Press, 1995.

[42] M. A. Golberg, C. S. Chen, and A. S. Muleshkov. The method of fundamental solutions for time-dependent problems. *Transactions on Modelling and Simulation*, 22, 1999.

[43] C. Gáspár. Some variants of the method of fundamental solutions: regularization using radial and nearly radial basis functions. *Central European Journal of Mathematics*, 11(8):1429–1440, 2013.

[44] C. Gáspár. A multi-level technique for the method of fundamental solutions without regularization and desingularization. *Engineering Analysis with Boundary Elements*, 103:145 – 159, 2019.

[45] C. Gáspár and A. Karageorghis. Method of fundamental solutions formulations for biharmonic problems. *Engineering Analysis with Boundary Elements*, 175 (106180), 2025.

[46] H. Hassanein. Elastic problems leading to the biharmonic equation in regions of sector type. Master thesis, Apr. 1973. URL `https://summit.sfu.ca/_flysystem/fedora/sfu_migrate/3352/b13768475.pdf`.

[47] M. R. Hematiyan, M. Arezou, N. K. Dezfouli, and M. Khoshroo. Some remarks on the method of fundamental solutions for two dimensional elasticity. *Computer Modeling in Engineering & Sciences*, 121(2):661–686, 2019.

[48] Y. C. Hon and T. Wei. The method of fundamental solutions for solving multidimensional inverse heat conduction problems. *Computer Modeling in Engineering & Sciences*, 7(2):119 – 132, 2005.

[49] K. Hout. *Numerical Partial Differential Equations in Finance Explained*. Springer Nature, 2017.

[50] B. Jin and Y. Zheng. A meshless method for some inverse problems associated with the Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 195(19-22):2270 – 2288, 2006.

[51] A. Karageorghis. A practical algorithm for determining the optimal pseudo-boundary in the method of fundamental solutions. *Advances in Applied Mathematics and Mechanics*, 1(4):510 – 528, 2009.

[52] A. Karageorghis. Efficient MFS algorithms for inhomogeneous polyharmonic problems. *Journal of Scientific Computing*, 46:519–541, 2011.

[53] A. Karageorghis and G. Fairweather. The method of fundamental solutions for the numerical solution of the biharmonic equation. *Journal of Computational Physics*, 69:434–459, 1987.

[54] A. Karageorghis and D. Lesnic. The method of fundamental solutions for the Oseen steady-state viscous flow past obstacles of known or unknown shapes. *Numerical Methods for Partial Differential Equations*, 35:2103–19, 2009.

[55] A. Karoui, L. Bear, P. Migerditichan, and N. Zemzemi. Evaluation of fifteen algorithms for the resolution of the electrocardiography imaging inverse problem using ex-vivo and in-silico data. *Frontiers in Physiology*, 9, 2018.

[56] Y. Kharbaoui, O. Askour, B. Braikat, A. Tri, H. Zahrouni, and M. Potier-Ferry. Coupling of the method of fundamental solutions and the domain decomposition technique for solving Laplace equation. *Lecture Notes in Mechanical Engineering*, page 37 – 45, 2024.

[57] B. D. Koshanov. About solvability of boundary value problems for nonhomogeneous polyharmonic equation in a ball. *Advances in Pure and Applied Mathematics*, 4:351–373, 2013.

[58] V. V. Krachik, M. A. Sadybekov, and B. T. Torebek. Uninequeness of solutions to boundary-value problems for the biharmonic equation in a ball. *Electronic Journal of Differential Equations*, 15:1–9, 2015.

[59] V. D. Kupradze and M. A. Aleksidze. The method of fundamental equations for the approximate solution of certain boundary value problems. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 4:82–126, 1964.

[60] V. D. Kupradze, T. G. Gegelia, M. O. Basheleishvili, and T. V. Burchuladze. *Three-dimensional Problems of The Mathematical Theory Of Elasticity And Thermoelasticity*. North-Holland Publishing Company, Netherlands, 1971.

[61] J. R. Kuttler and V. G. Sigillito. Bounding eigenvalues of elliptic operators. *SIAM Journal on Mathematical Analysis*, 9:768–773, 1978.

[62] Z.-C. Li, M.-G. Lee, and J. Y. C. abd Ya Ping Liu. The trefftz method using fundamental solutions for biharmonic equations. *Journal of Computational and Applied Mathematics*, (235):4350–4367, 2011.

[63] Z.-C. Li, L.-P. Zhang, Y. Wei, M.-G. Lee, and J. Y. Chiang. Boundary methods for Dirichlet problems of Laplace´s equation in elliptic domains with elliptic holes. *Engineering Analysis with Boundary Elements*, 61:91–103, 2015.

[64] Z.-C. Li, H.-T. Huang, Y. Wei, and L. Zhang. *The Method of Fundamental Solutions: Theory and Applications*. Sciense Press, EDP Sciences, 2023.

[65] J. Lin, A. R. Lamichhane, C. S. Chen, and J. Lu. The adaptive algorithm for the selection of sources of the method of fundamental solutions. *Engineering Analysis with Boundary Elements*, 95:154–159, 2018. ISSN 0955-7997.

[66] C.-S. Liu. Improving the ill-conditioning of the method of fundamental solutions for 2D Laplace equation. *Computer Modeling in Engineering & Sciences*, 28 (2):77 – 93, 2008.

[67] Q. G. Liu and B. Sarles. A non-singular method of fundamental solutions for the two-dimensional steady-state isotropic thermoelasticity problems. *Engineering Analysis with Boundary Elements*, 75:89–102, 2017.

[68] S. A. Lurie. *Biharmonic Problem in the Theory of Elasticity*. Taylor & Francis Group, 1995.

[69] L. Marin. The method of fundamental solutions for inverse problems associated with the steady-state heat conduction in the presence of sources. *Computer Modeling in Engineering & Sciences*, 30(2):99 – 122, 2008.

[70] L. Marin. An alternating iterative MFS algorithm for the Cauchy problem for the modified Helmholtz equation. *Computational Mechanics*, 45:665–677, 2010.

[71] L. Marin and A. Karageorghis. Regularized MFS-based boundary identification in two-dimensional Helmholtz-type equations. *Computers, Materials and Continua*, 10:259–293, 2016.

[72] L. Marin and D. Lesnic. The method of fundamental solutions for inverse boundary value problems associated with the two-dimensional biharmonic equation. *Mathematical and Computer Modelling*, 42(3-4):261 – 278, 2005.

[73] T. Martinod, A. Fierro, C. Posada, and J. J. Sanchez. On the biharmonic equation. https://repository.eafit.edu.co/bitstreams/b178a4eb-a816-47b2-8bae-699340ebfa7f/download, 2024.

[74] R. Mathon and R. L. Johnston. The approximate solution of elliptic boundary-value problems by fundamental solutions. *SIAM Journal on Numerical Analysis*, 14(4):638–650, 1977.

[75] P. M. Morse and H. Feshbach. *Methods of theoretical physics*. McGraw-Hill Company, Inc., 1953.

[76] P. A. Ramachandran. Method of fundamental solutions: singular value decomposition analysis. *Numerical Methods in Biomedical Engineering*, 18:789–801, 2002.

[77] S. S. Rao. *The Finite element method in engineering*. Elsevier, Burlington, 2011.

[78] J. J. Sakurai and J. Napolitano. *Modern Quantum Mechanics*. Addison-Wesley, San Francisco, 2011.

[79] R. Schaback. Adaptive numerical solution of MFS systems. *The Method of Fundamental Solutions - A Meshless Method*, page 1 – 27, 2008.

[80] P. Serranho. A hybrid method for inverse scattering for shape and impedance. *Inverse Problems*, 22:663–680, 2006.

[81] T. Shigeta, D. L. Young, and C.-S. Liu. Adaptive multilayer method of fundamental solutions using a weighted greedy QR decomposition for the Laplace equation. *Journal of Computational Physics*, 231(21):7118 – 7132, 2012.

[82] Y.-S. Smyrlis. Applicability and applications of the method of fundamental solutions. *Mathematics of Computation*, 78:1399–1434, 2009.

[83] Y.-S. Smyrlis and A. Karageorghis. Efficient implementation of the MFS: the three scenarios. *Journal of Computational and Applied Mathematics*, 227(1): 83–92, 2009.

[84] S. Timoshenko and J. N. Goodier. *Theory of elasticity*. McGraw-Hill book Company, New York, 1951.

[85] C. C. Tsai, D. L. Young, J. H. Chiang, and D. C. Lo. The method of fundamental solutions for solving options pricing models. *Applied Mathematics and Computation*, 45:390–401, 2006.

[86] D. L. Young and J. W. Ruan. Method of fundamental solutions for scattering problems of electromagnetic waves. *Computer Modeling in Engineering & Sciences*, 7(2):223–32, 2005.

[87] L.-P. Zhang, Z.-C. Li, H.-T. Huang, and M.-G. Lee. Comparisons of method of fundamental solutions, method of particular solutions and the MFS-QR: stability analysis. *Engineering Analysis with Boundary Elements*, 123:182–199, 2021.

[88] L.-P. Zhang, Z.-C. Li, H.-T. Huang, and M.-G. Lee. New locations of source nodes for method of fundamental solutions solving Laplace's equation; pseudo radial-lines. *Engineering Analysis with Boundary Elements*, 136:93 – 115, 2022.

[89] L.-P. Zhang, Z.-C. Li, Y. Wei, and H.-T. Huang. New regularization techniques for ill-conditioning problems and their applications: Choices of regularization parameters. *Engineering Analysis with Boundary Elements*, 152:347–361, 2023.

Appendix I

MFS-SVD Matlab code for Helmholtz case in polar coordinates, Example 4.1

```
V = 5:5:500;
Ptos_colloc = zeros(1, numel(V));
Ptos_fonte = zeros(1, numel(V));
Err_max = zeros(1, numel(V));
Err_relativo = zeros(1, numel(V));
Err_L2 = zeros(1, numel(V));
NrCond = zeros(1, numel(V));
NrPontos = zeros(1, numel(V));
Err_maxsvd = zeros(1, numel(V));
Err_relativosvd = zeros(1, numel(V));
Err_L2svd = zeros(1, numel(V));
NrCondsvd = zeros(1, numel(V));
NrCondbeffe = zeros(1, numel(V));
NrCondeff = zeros(1, numel(V));

count = 1;
for N = V
  M = 2*N;
alpha = 1.5;
r = 1.0; % radius of the circumference
k = 8.0;
T = 200; % Truncature of the aadithiom theorem
th=(1:M)'/M*2*pi;
Rb = r*ones(size(th));
xb = Rb.*cos(th);
yb = Rb.*sin(th);
BP = [xb(:), yb(:)];
```

```
th_sp =(1:N)'/N*2*pi;
Rs = r*alpha*ones(size(th_sp));
xs = Rs.*cos(th_sp);
ys = Rs.*sin(th_sp);
SP = [xs(:), ys(:)];
BPsh =[Rb(:), th(:)];
SPsh =[Rs(:), th_sp(:)];
% Building the system
DM= pdist2(BP,SP);
A = matrizfundamental(BP,SP,DM,k);
Pf =[1.1, 0]; % Pto rho =(x0,0)
Pfs = [sqrt(Pf(1,1).^2 + Pf(1,2).^2 ) atan(Pf(1,2)/Pf(1,1))];
DMS= pdist2(BP,Pf);
rhsf = rhs(BP,Pf,DMS,k);
a=A\rhsf;
% Finding the numerical solution
qtpts=33; %numero de divisoes
dr = linspace(0,r,qtpts);
dth = (1:qtpts)'/qtpts*2*pi;
[Rr, Tt] = meshgrid(dr,dth);
pto_aval_sp =[Rr(:), Tt(:)];
xxx = Rr.*cos(Tt);
yyy = Rr.*sin(Tt);
pto_aval = [xxx(:) yyy(:)];
% Computation of error for MFS
DMSS= pdist2(pto_aval,Pf);
SolExa =  rhs(pto_aval,Pf,DMSS,k);
ExaSol = SolExa;
% Numerical Solution
D_eval= pdist2(pto_aval,SP);
[AD] = matrizfundamental(pto_aval,SP,D_eval,k);
```

```
59 NumSolad = solnumerica(AD,a);
60 [D, L, Err_rel] = todoserros(NumSolad,ExaSol);
61 Nrcon_bef = cond(full(A));
62 s = svd(full(A));
63 Nrcon_Eff = norm(rhsf)/(norm(a)*min(s));
64 % Output MFS
65 fprintf("MFS solution \n")
66 fprintf("Nr Colocation points: %d \n",size(BP,1))
67 fprintf("Nr Source points: %d \n",size(SP,1))
68 fprintf("L2 error: %1.5E \n",max(sqrt(L./size(NumSolad,1))))
69 fprintf("Cond. No: %1.5E \n",Nrcon_bef)
70 fprintf("No of arbitrary points: %d \n",size(pto_aval,1))
71 fprintf("Effective cond nr.: %1.5E, \n",Nrcon_Eff)
72 Ptos_colloc(1:1, count:count) = size(BP,1);
73 Ptos_fonte(1:1, count:count) = size(SP,1);
74 Err_max(1:1, count:count) = max(D);
75 Err_relativo(1:1, count:count) =  max(Err_rel);
76 Err_L2(1:1, count:count) = max(sqrt(L./size(NumSolad,1)));
77 NrCond(1:1, count:count) = Nrcon_bef;
78 NrPontos(1:1, count:count) = size(pto_aval,1);
79 NrCondbeffe(1:1, count:count) = Nrcon_Eff;
80 % Computation of error by MFS-SVD
81 MPF = MatrizPF(SPsh,BPsh,T,k);
82 V1 = Regularizacao(MPF,size(SP,1));
83 MMreg = (1i/4)*matrizfundreg(V1,BPsh,SPsh,T,k);
84 aa = MMreg\rhsf;
85 [ADreg] = (1i/4)*matrizfundreg(V1,pto_aval_sp,SPsh,T,k);
86 NumSolreg = solnumerica(ADreg,aa);
87 [Dreg, Lreg, Err_relreg] = todoserros(NumSolreg,ExaSol);
88 Nrcon_aft = cond(full(MMreg));
89 NrconPf = cond(full(MPF));
```

```
ss = svd(full(MMreg));
NrconPc = norm(rhsf)/(norm(aa)*min(ss));
% Output  MFS-SVD
fprintf("Solution After regularization \n")
fprintf("L2 error: %1.5E \n",max(sqrt(Lreg./size(NumSolreg,1)
   )))
fprintf("Cond. No: %1.5E \n",Nrcon_aft)
fprintf("Effective cond. No: %1.5E \n",NrconPc)
Err_maxsvd(1:1, count:count) = max(Dreg);
Err_relativosvd(1:1, count:count) = max(Err_relreg);
Err_L2svd(1:1, count:count) = max(sqrt(Lreg./size(NumSolreg
   ,1)));
NrCondsvd(1:1, count:count) = Nrcon_aft;
NrCondPf(1:1, count:count) = NrconPf;
NrCondeff(1:1, count:count) = NrconPc;
count = count + 1;
Aresults = [Ptos_colloc; Ptos_fonte; Err_max; Err_relativo;
   Err_L2; NrCond; NrPontos; NrCondbeffe];
Aresultssvd = [Err_maxsvd;Err_relativosvd; Err_L2svd;
   NrCondsvd; NrCondeff ];
dlmwrite('myResultsfile_x0_1_1.csv',Aresults);
dlmwrite('myResultsfile_x0_1_1.csv',Aresultssvd,'-append','
   roffset',1);
end
```

```
  function [A] = matrizfundamental(BP,SP,DM,k)
%BP, SP: boundary and source points matrix & DM of distances
M = size(BP,1);
N = size(SP,1);
A = zeros(M,N);
for p=1:M
```

```
   for j=1:N
      A(p,j)= (1i./4).*besselh(0,k*DM(p,j));
  end
 end
end
```

```
function [RHS] = rhs(BP,SP,DM,k)
 MM = size(BP,1);
 NN = size(SP,1);
RHS = zeros(MM,NN);
for p=1:MM
   for j=1:NN
      RHS(p,j)= (-1/4).*bessely(0,k*DM(p,j));
  end
end
end
```

```
 function [NumSol] = solnumerica(AA,a)
% AA fundamental matrix at evaluation pts. & a: weights
 u = AA*a;
NumSol =  u;
end
```

```
function [D, L, Err_rel] = todoserros(NumSol,ExaSol)
% NumSol: Numerical sol. & ExaSol: exact sol.
 D=zeros(1,size(NumSol,1)) ;
 Err_rel = zeros(1,size(NumSol,1));
 L=0;
 for n=1:size(NumSol,1)
   Vdiff = NumSol(n,1)-ExaSol(n,1);
   D(n)= norm(Vdiff);
   L = L + norm(Vdiff).^2;
```

```
    Err_rel(n) = D(n)./norm(ExaSol(n,1));
 end
end
```

```
function [MPf] = MatrizPF(SPs,PQ,T,k)
% SPs: matriz dos pontos fonte & T: Truncature
Npf = size(SPs,1);
N1 = zeros(1, 1+2*T);
ctt = 2;
MPf = sparse(Npf,(1+2*T));
 for pp = 1:Npf
   for ii = 0:1:T
        if ii == 0
           N1(1) = Hn(ii,k*SPs(pp,1)).*Normalizacao(PQ,ii,k);
        else
        N1(ctt) = 2*Hn(ii,k*SPs(pp,1)).*Normalizacao(PQ,ii,k)
           .*cos(ii*SPs(pp,2));
        N1(ctt+1) = 2*Hn(ii,k*SPs(pp,1)).*Normalizacao(PQ,ii,
           k).*sin(ii*SPs(pp,2));
        ctt = ctt+2;
        end
   end
    ctt = 2;
    MPf(pp:pp, 1:(1+2*T)) = N1 ;
 end
end
```

```
 function Mreg = Regularizacao(Blk,N)
% N: Nr de pontos de fonte
  [~,~,V] = svd(full(Blk));
  VV1 = V';
  Mreg =VV1(1:N,:) ;
```

```
end
```

```
function [MM] = matrizfundreg(W,BPs,SPs,T,k)
% SPs, BPs, T: source, boundary points matrix & truncature
Npc = size(BPs,1);
Npf = size(SPs,1);
MM = sparse(Npc,Npf);
 for pq = 1:Npc
   TEMP = MatrizPC(BPs(pq,:),BPs,T,k);
   Line = W*TEMP;
   MM(pq:pq, 1:Npf) = Line.' ;
 end
end
```

```
function [MPc] = MatrizPC(PP,BPsh,T,k)
%BPs: matriz dos pontos fronteira & T: Truncature of series
N2 = zeros(1, 1+2*T);
MPc = sparse(1+2*T,1);
ct =2 ;
  for jj = 0:1:T
    jnk= Jn(jj,k*PP(1,1));
   if jj == 0
    N2(1) = jnk./Normalizacao(BPsh,jj,k);
    elseif jj < 140
    N2(ct) = jnk*cos(jj*PP(1,2))./Normalizacao(BPsh,jj,k);
    N2(ct+1) =jnk*sin(jj*PP(1,2))./Normalizacao(BPsh,jj,k);
    ct = ct+2;
    else
         Rdd = Normalizacao2(BPsh,jj,k);
      N2(ct) = (PP(1,1).^jj).*cos(jj.*PP(1,2))./Rdd.^jj;
      N2(ct+1) = (PP(1,1).^jj)*sin(jj*PP(1,2))./Rdd.^jj;
      ct = ct+2;
```

```
    end
 end
    MPc(1:(1+2*T), 1:1) = N2.' ;
end
```

```
function [JNK] = Normalizacao(N,nn,k)
% N: matriz dos pontos fronteira & nn: ordem da funcao Jn
maxBP = max(N(:,1));
dl = linspace(0,maxBP,10000);
BPn =dl(:);
jnk= Jn(nn,k.*BPn(:,1));
JNK= max(abs(jnk));
end
```

```
function [Rd] = Normalizacao2(N,nn,k)
maxBP = max(N(:,1));
dl = linspace(0,maxBP,10000);
BPn =dl(:);
jnk2=(1./sqrt(2*pi.*nn)).*(exp(1).*k.*BPn(:,1)./(2.*nn)).^nn;
JNK2= max(abs(jnk2));
idx = find(jnk2==JNK2);
Rd = max(BPn(idx,1));
end
```

```
function Hn = Hn(n,t)
Hn = besselh(n, t);
end
```

```
function Jn = Jn(n,t)
Jn = besselj(n, t);
end
```

Appendix II

MFS-SVD Matlab code for Helmholtz BVP in elliptic coordinates, Example 4.5

Some functions required by this code are the same to the ones presented in Appendix I. So for functions J_n, H_n, *solnumerica*, *matrizfundamental* and *todoserros* refer to that Appendix.

```
V = 5:5:400;
Ptos_colloc = zeros(1, numel(V));
Ptos_fonte = zeros(1, numel(V));
Err_max = zeros(1, numel(V));
Err_relativo = zeros(1, numel(V));
Err_L2 = zeros(1, numel(V));
NrCond = zeros(1, numel(V));
NrPontos = zeros(1, numel(V));
Err_maxsvd = zeros(1, numel(V));
Err_relativosvd = zeros(1, numel(V));
Err_L2svd = zeros(1, numel(V));
NrCondsvd = zeros(1, numel(V));
NrCondPf = zeros(1, numel(V));
NrCondPc = zeros(1, numel(V));
count = 1;
k = 1.0;
alpha = 10;          % magnifying factor
a1 = 6.0;                    % major axis
b1 = 1.0;                    % minor axis
Rho_0 = atanh(b1/a1);
Sig_0 = sqrt(a1^2-b1^2);
OO = fix(max(V)/4)+5;
[NFL1,NFL2,NFL3,NFL4] = Normalizacao(k,Sig_0,OO,Rho_0);
for N = V
  M = 2*N;
```

```
26 if N <= 120
27 O = 30; % Matrix order define the amount of coef.
28 else
29 O = fix(N/4)+5; % Matrix order define the amount of coef.
30 end
31 T = O; % Truncature of addithiom theorem
32 th=(1:M)'/M*2*pi;
33 Rb = Rho_0*ones(size(th));
34 xb = Sig_0.*cosh(Rho_0).*cos(th);
35 yb = Sig_0.*sinh(Rho_0).*sin(th);
36 BP = [xb(:), yb(:)];
37
38 th_sp =(1:N)'/N*2*pi;
39 Rho_1= Rho_0*alpha;
40 Rs = Rho_1*ones(size(th_sp));
41 xs = Sig_0.*cosh(Rho_1).*cos(th_sp);
42 ys = Sig_0.*sinh(Rho_1).*sin(th_sp);
43 SP = [xs(:), ys(:)];
44 sg0= Sig_0*ones(size(th));
45 sg1= Sig_0*ones(size(th_sp));
46 % Definition of f
47 f = @(x,y,s) (1./(sqrt(2).*s)).*sqrt(x.^2+y.^2-s.^2 + sqrt((x
      .^2+y.^2-s.^2).^2+4.*s.^2.*y.^2));
48 qb = BP(:,2)<0;
49 qs = SP(:,2)<0;
50 BPsh =[asinh(f(xb(:),yb(:),sg0(:))), acos(xb(:)./(sg0.*cosh(
      Rb)))];
51 SPsh =[asinh(f(xs(:),ys(:),sg1(:))), acos(xs(:)./(sg1.*cosh(
      Rs)))];
52 idxp = find(qb ==1);
53 idxs = find(qs ==1);
```

```
54 BPsh(idxp,2)= 2*pi-BPsh(idxp,2);
55 SPsh(idxs,2) = 2*pi-SPsh(idxs,2);
56 % Construindo o sistema e resolvendo-o
57 DM= pdist2(BP,SP);
58 A = matrizfundamental(BP,SP,DM,k);
59 Pf =[20.0, 0]; %  boundary condition  outside
60 Pfs = [sqrt(Pf(1,1).^2 + Pf(1,2).^2 ) atan(Pf(1,2)/Pf(1,1))];
61 DMS= pdist2(BP,Pf);
62 rhsf = rhs(BP,Pf,DMS,k);
63 a=A\rhsf;
64 % Numerical solution
65 qtpts=1000; %numero de divisoes
66 dth = (1:qtpts)'/qtpts*2*pi;
67 sgg0 = Sig_0*ones(size(dth));
68 Rbb = Rho_0*ones(size(dth));
69 xxx = a1.*cos(dth);
70 yyy = b1.*sin(dth);
71 pto_aval_sp =[asinh(f(xxx(:), yyy(:),sgg0(:))), acos(xxx(:)
     ./(sgg0.*cosh(Rbb)))];
72 pto_aval = [xxx(:) yyy(:)];
73 qbb = pto_aval(:,2)<0;
74 idxpp = find(qbb ==1);
75 pto_aval_sp(idxpp,2)= 2*pi-pto_aval_sp(idxpp,2);
76 % Error by MFS Classic
77 DMSS= pdist2(pto_aval,Pf);
78 SolExa =  rhs(pto_aval,Pf,DMSS,k);
79 ExaSol = SolExa; %Sol. Exacta
80 D_eval= pdist2(pto_aval,SP);
81 [AD] = matrizfundamental(pto_aval,SP,D_eval,k);
82 NumSolad = solnumerica(AD,a); % Sol. Numerica
83 [D, L, Err_rel] = todoserros(NumSolad,ExaSol);
```

```
84  Nrcon_bef = cond(full(A));
85  % Output antes da regularizacao
86  fprintf(" MFS solution \n")
87  fprintf(" Nr Colocation points: %d \n",size(BP,1))
88  fprintf(" Nr Source points: %d \n",size(SP,1))
89  fprintf("L2 error: %1.5E \n",max(sqrt(L./size(NumSolad,1))))
90  fprintf(" Cond. No: %1.5E \n",Nrcon_bef)
91  fprintf("No of arbitrary points: %d \n",size(pto_aval,1))
92  Ptos_colloc(1:1, count:count) = size(BP,1);
93  Ptos_fonte(1:1, count:count) = size(SP,1);
94  Err_max(1:1, count:count) = max(D);
95  Err_relativo(1:1, count:count) =  max(Err_rel);
96  Err_L2(1:1, count:count) = max(sqrt(L./size(NumSolad,1)));
97  NrCond(1:1, count:count) = Nrcon_bef;
98  NrPontos(1:1, count:count) = size(pto_aval,1);
99  % Error MFS-SVD
100 MPF = MatrizPFE(SPsh,BPsh,T,k,Sig_0,T,0,NFL1,NFL2,NFL3,NFL4);
101 V1 = Regularizacao(MPF,size(SP,1));
102 MMreg = (1i/4)*matrizfundreg(V1,BPsh,SPsh,T,k,Sig_0,T,0,NFL1,
       NFL2,NFL3,NFL4) ;
103 aa = MMreg\rhsf;
104 [ADreg] = (1i/4)*matrizfundreg(V1,pto_aval_sp,SPsh,T,k,Sig_0,
       T,0,NFL1,NFL2,NFL3,NFL4);
105 NumSolreg = solnumerica(ADreg,aa);
106 [Dreg, Lreg, Err_relreg] = todoserros(NumSolreg,ExaSol);
107 Nrcon_aft = cond(full(MMreg));
108 NrconPf = cond(full(MPF));
109 NrconPc = cond(full(MPF));
110 % Output MFS-SVD
111 fprintf("MFS-SVD solution \n")
112 fprintf("L2 error: %1.5E \n",max(sqrt(Lreg./size(NumSolreg,1)
```

```
   )))
fprintf("Cond. No: %1.5E \n",Nrcon_aft)
Err_maxsvd(1:1, count:count) = max(Dreg);
Err_relativosvd(1:1, count:count) = max(Err_relreg);
Err_L2svd(1:1, count:count) = max(sqrt(Lreg./size(NumSolreg
   ,1)));
NrCondsvd(1:1, count:count) = Nrcon_aft;
NrCondPf(1:1, count:count) = NrconPf;
NrCondPc(1:1, count:count) = NrconPc;
count = count + 1;
Aresults = [Ptos_colloc; Ptos_fonte; Err_max; Err_relativo;
   Err_L2; NrCond; NrPontos ];
Aresultssvd = [Err_maxsvd;Err_relativosvd; Err_L2svd;
   NrCondsvd ];
dlmwrite('myResultsfile_coord_ell_RHS_eli_eli.csv',Aresults);
dlmwrite('myResultsfile_coord_ell_RHS_eli_eli.csv',
   Aresultssvd,'-append','roffset',1);
end
```

```
function [HnE,JnE,CnE] = MathFe(ku,q,n,v1,t1,L,O)
%ku function code
[~, mc, ~] = eig_Spm(ku,q,O);
vy = Jpm(ku,v1,q,mc,L,O);
yv = Hpm1(ku,v1,q,mc,L,O);
yx = Spm(ku,t1,mc,L,O);
nml = Npm(ku,mc,L,O);
JnE = vy(n);
HnE = yv(n)/nml(n);
CnE = yx(n);
end
```

```
function [MM]=matrizfundreg(W,BPs,SPs,T,k,sg,L,O,M1,M2,M3,M4)
```

```
% SPs, BPs: source and boundary points & T: truncature
Npc = size(BPs,1);
Npf = size(SPs,1);
MM = sparse(Npc,Npf);
 for pq = 1:Npc
   TEMP = MatrizPCE(BPs(pq,:),BPs,T,k,sg,L,O,M1,M2,M3,M4);
   Line = W*TEMP;
   MM(pq:pq, 1:Npf) = Line.' ;
 end
end
```

```
function [MPc] = MatrizPCE(PP,BPsh,T,k,sg,L,O,M1,M2,M3,M4)
% BPs: boundary points & T: truncature
N2 = zeros(1, 4*T);
MPc = sparse(4*T,1);
ct = 1 ;
qq = k^2*sg^2/4;
  for jj = 1:1:T
       [~,je1,ce1] = MathFe(1,qq,jj,PP(1,1),PP(1,2),L,O);
       [~,je2,ce2] = MathFe(2,qq,jj,PP(1,1),PP(1,2),L,O);
       [~,je3,ce3] = MathFe(3,qq,jj,PP(1,1),PP(1,2),L,O);
       [~,je4,ce4] = MathFe(4,qq,jj,PP(1,1),PP(1,2),L,O);
       N2(ct) = je1*ce1/M1(jj,1);
       N2(ct+1) = je2*ce2/M2(jj,1);
       N2(ct+2) = je3*ce3/M3(jj,1);
       N2(ct+3) = je4*ce4/M4(jj,1);
       ct = ct+4;
 end
    MPc(1:4*T, 1:1) = N2.' ;
end
```

```
function MPf = MatrizPFE(SPs,PQ,T,k,sg,L,O,M1,M2,M3,M4)
```

```
% SPs: Source points & % T: truncature
Npf = size(SPs,1);
N1 = zeros(1, 4*T);
ctt = 1;
qq = k^2*sg^2/4;
MPf = sparse(Npf,4*T);
%First factor in addithiom teorem decomposition
 for pp = 1:Npf
   for ii = 1:1:T
        [h1,~,u1] = MathFe(1,qq,ii,SPs(pp,1),SPs(pp,2),L,0);
        [h2,~,u2] = MathFe(2,qq,ii,SPs(pp,1),SPs(pp,2),L,0);
        [h3,~,u3] = MathFe(3,qq,ii,SPs(pp,1),SPs(pp,2),L,0);
        [h4,~,u4] = MathFe(4,qq,ii,SPs(pp,1),SPs(pp,2),L,0);
        N1(ctt) = 4*h1*M1(ii,1)*u1;
        N1(ctt+1) = 4*h2*M2(ii,1)*u2;
        N1(ctt+2) = 4*h3*M3(ii,1)*u3;
        N1(ctt+3) = 4*h4*M4(ii,1)*u4;
        ctt = ctt+4;
   end
    ctt = 1;
    MPf(pp:pp, 1:4*T) = N1 ;
 end
end
```

```
function [Lj1, Lj2, Lj3, Lj4] = Normalizacao(k,sg,0,rh)
% nn: order of Jn
H = 10000; % arbitrary to obtain max and min radius
vth=(1:H)'/H*2*pi;
xc = sg.*cosh(rh).*cos(vth);
yc = sg.*sinh(rh).*sin(vth);
sgc= sg*ones(size(vth));
```

```
8  g = @(x,y,s) (1./(sqrt(2).*s)).*sqrt(x.^2+y.^2-s.^2 + sqrt((x
       .^2+y.^2-s.^2).^2+4.*s.^2.*y.^2));
9  BPc =asinh(g(xc(:),yc(:),sgc(:)));
10 minBP = min(BPc(:,1));
11 maxBP = max(BPc(:,1));
12 qq = k^2*sg^2/4;
13 LL = 10000;
14 dl = linspace(minBP,maxBP,LL);
15 BPn =dl(:);
16 Lj1 = zeros(0, LL);
17 Lj2 = zeros(0, LL);
18 Lj3 = zeros(0, LL);
19 Lj4 = zeros(0, LL);
20 [~, mc1, ~] = eig_Spm(1,qq,0);
21 [~, mc2, ~] = eig_Spm(2,qq,0);
22 [~, mc3, ~] = eig_Spm(3,qq,0);
23 [~, mc4, ~] = eig_Spm(4,qq,0);
24  for mj = 1:1:LL
25          je1 = Jpm(1,BPn(mj,1),qq,mc1,0,0);
26          je2 = Jpm(2,BPn(mj,1),qq,mc2,0,0);
27          je3 = Jpm(3,BPn(mj,1),qq,mc3,0,0);
28          je4 = Jpm(4,BPn(mj,1),qq,mc4,0,0);
29       % arrumando todos os vectores numa matriz
30       LIj1(1:0, mj:mj) = je1.';
31       LIj2(1:0, mj:mj) = je2.';
32       LIj3(1:0, mj:mj) = je3.';
33       LIj4(1:0, mj:mj) = je4.';
34  end
35  Abs1 = abs(LIj1);
36  Abs2 = abs(LIj2);
37  Abs3 = abs(LIj3);
```

```
 Abs4 = abs(LIj4);
 Lj1 = max(Abs1,[],2); %determinando o maximo para cada linha
 Lj2 = max(Abs2,[],2);
 Lj3 = max(Abs3,[],2);
 Lj4 = max(Abs4,[],2);
end
```

```
function [RHS] = rhs(BP,SP,DM,k)
% BP, SP boundary and source poits & DM matrix of distance
MM = size(BP,1);
NN = size(SP,1);
RHS = zeros(MM,NN);
for p=1:MM
   for j=1:NN
      RHS(p,j)= (-1/4).*bessely(0,k*DM(p,j));
  end
end
end
```

From this point all the functions being transcribed here are part of original Matlab toolbox from Cojocaru, [32, 33]. As mentioned in Remark 4.2 the code was slightly modified to allow double precision.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   eig_Spm FUNCTION       [p,m = e,o (even,odd)]
%   [va,mv,vt]=eig_Spm(KF,q,O)
%
%   INPUTS:       -q= scalar, elliptic parameter value (q > 0)
%                -KF= function code:  KF=1 even-even
%                                     KF=2 even-odd
%                                     KF=3 odd-even
%                                     KF=4 odd-odd
%                          -O = order of the matrix (maximum
   number of coefficients)
```

```
%   OUTPUTS:     -va=  vector of eigenvalues 'a'
%                -mv= matrix of expansion coefficients
%                -vt= values of order n:
%                        KF=1   vt=[0 2 4 6 ... (2*ncoeffs-2)]
%                        KF=2   vt=[1 3 5 7 ... (2*ncoeffs-1)]
%                        KF=3   vt=[2 4 6 8 ... (2*ncoeffs)]
%                        KF=4   vt=[1 3 5 7 ... (2*ncoeffs-1)]
%   Default value for number of coefficients: ncoeffs=25
%   'va' and 'mv' are determined for all 25 orders specified
   in 'vt'
%   (The size of 'va' is [ncoeffs 1], the size of 'vt' is [1
   ncoeffs],
%    and the size of 'mv' is [ncoeffs ncoeffs])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    E. Cojocaru, revised by H. Calunga in January 2024
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   eig_Spm FUNCTION CALL
function [va,mv,vt]=eig_Spm(KF,q,O)
%---------------even-even---KF == 1----------------------
if KF == 1
%   COMPUTE THE MATRIX M
    ik = 0:(O-1);
    off_diag = q*ones(1,length(ik)-1);
    off_diagd= off_diag; off_diagd(1)=2*off_diag(1);
    M= diag((2*ik).^2) + diag(off_diagd, -1) + diag(off_diag,
        1);
%   COMPUTE EIGENVALUES AND EIGENVECTORS
    [u,d]    = eig(M,'nobalance');
    [va,num]= sort(diag(d));    % va is vector of eigenvalues
       'a'
    v= u(:,num);                % matrix of eigenvectors
```

```
38      vt=2*ik;                    % even values of order n
39  %   PROCESS COEFFICIENTS
40  %   (For a given order n, sum of coefficients = 1 )
41      nc=size(v,2);          % number of columns in v
42      sum_v=sum(v);          % sum_vector over columns in v
43  %   Compute matrix mv of processed eigenvectors
44      mv=[];
45      for k=1:nc
46        cn=sum_v(k);       % constant of processing on column k
47        ncv=v(:,k)/cn;     % processed column vector
48        mv=[mv ncv];       % matrix mv of processed eigenvectors
49      end
50  %---------------even-odd---KF == 2 ----------------------
51  elseif KF == 2
52  %   COMPUTE THE MATRIX M
53      ik = 0:(O-1);
54      off_diag = q*ones(1,length(ik)-1);
55      M= diag((2*ik+1).^2) + diag(off_diag, -1) + diag(off_diag
           , 1);
56      M(1,1)= M(1,1) + q;
57  %   COMPUTE EIGENVALUES AND EIGENVECTORS
58      [u,d]    = eig(M,'nobalance');
59      [va,num]= sort(diag(d));  % va is vector of eigenvalues '
           a'
60      v= u(:,num);               % matrix of eigenvectors
61      vt=2*ik+1;                 % odd values of order n
62  %   PROCESS COEFFICIENTS
63  %   (For a given order n, sum of coefficients = 1 )
64      nc=size(v,2);     % number of columns in v
65      sum_v=sum(v);     % sum_vector over columns in v
66  %   Compute matrix mv of processed eigenvectors
```

```
67      mv=[];
68      for k=1:nc
69      cn=sum_v(k);      % constant of processing on column k
70      ncv=v(:,k)/cn;    % processed column vector
71      mv=[mv ncv];      % matrix mv of processed eigenvectors
72      end
73  %---------------odd-even--KF == 3 -----------------------
74  elseif KF == 3
75  %   COMPUTE THE MATRIX M
76      ik=1:0;
77      off_diag = q*ones(1,length(ik)-1);
78      M= diag((2*ik).^2) + diag(off_diag, -1) + diag(off_diag,
           1);
79  %   COMPUTE EIGENVALUES AND EIGENVECTORS
80      [u,d]    = eig(M,'nobalance');
81      [va,num]= sort(diag(d));    % va is vector of eigenvalues
           'a'
82      v=u(:,num);                 % matrix of eigenvectors
83      vt=2*ik;                    % even values of order n
84  %   PROCESS COEFFICIENTS
85  %   [For a given order n, (dSpm/dv)=1 at v=0]
86      nc=size(v,2);               % number of columns in v
87  %   Compute matrix mv of processed eigenvectors
88      mv=[];
89      for k=1:nc
90      cv=v(:,k).*vt';
91      cn=sum(cv);           % constant of processing on column
           k
92      ncv=v(:,k)/cn;        % processed column vector
93      mv=[mv ncv];          % matrix mv of processed
           eigenvectors
```

```
    end
 %---------------odd-odd--KF == 4 -------------------------
 elseif KF == 4
 %  COMPUTE THE MATRIX M
    ik=0:(O-1);
    off_diag =q*ones(1,length(ik)-1);
    M=diag((2*ik+1).^2) + diag(off_diag, -1) + diag(off_diag,1);
    M(1,1)=M(1,1)-q;
%   COMPUTE EIGENVALUES AND EIGENVECTORS
    [u,d]   = eig(M,'nobalance');
    [va,num]=sort(diag(d));     % va is vector of eigenvalues 'a'
    v=u(:,num);                 % matrix of eigenvectors
    vt=2*ik+1;                  % odd values of order n
%   PROCESS COEFFICIENTS
%   [For a given order n, (dSpm/dv)=1 at v=0]
    nc=size(v,2);             % number of columns in v
%   Compute matrix mv of processed eigenvectors
    mv=[];
    for k=1:nc
    cv=v(:,k).*vt';
    cn=sum(cv);          % constant of processing on column k
    ncv=v(:,k)/cn;       % processed column vector
    mv=[mv ncv];         % matrix mv of processed eigenvectors
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   RADIAL MATHIEU FUNCTION OF THE THIRD KIND
```

```
3  %    y = Hpm1(KF,u,q,mv,nmax,O)    [p,m = e,o (even,odd)]
4  %
5  %    INPUTS:      -u= value of radial coordinate to compute
   function
6  %                 -q= elliptical parameter (q > O)
7  %                 -mv= matrix of expansion coefficients
8  %                 -nmax= maximum order
9  %                 -KF= function code:  KF=1 even-even
10 %                                       KF=2 even-odd
11 %                                       KF=3 odd-even
12 %                                       KF=4 odd-odd
13 %        -O = order of the matrix (maximum number of
   coefficients)
14 %    OUTPUTS:     -y= vector of function values for all 'nmax'
   orders
15 %
16 %    The Radial Mathieu Function of the Third Kind is defined
   by analogy
17 %    with the Hankel Function of the First Kind:
18 %      Hpm1(KF,u,q,mv,nmax)=Jpm(KF,u,q,mv,nmax,O)+i*Ypm(KF,u,q
   ,mv,nmax,O)
19 %    'mv' is determined beforehand with function 'eig_Spm'
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21 %     E. Cojocaru, revised November 2008
22 %     Observations, suggestions, and recommendations are
   welcome at e-mail:
23 %     ecojocaru@theory.nipne.ro
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 %    Hpm1 FUNCTION CALL
26 function y = Hpm1(KF,u,q,mv,nmax,O)
27    y1 = Jpm(KF,u,q,mv,nmax,O);
```

```
    y2 = Ypm(KF,u,q,mv,nmax,0);
    y = y1 + 1i*y2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   RADIAL MATHIEU FUNCTION OF THE FIRST KIND
%   y = Jpm(KF,u,q,mv,nmax,0)        [p,m = e,o (even,odd)]
%
%   INPUTS:      -u= value of radial coordinate to compute
    function
%                -q= elliptical parameter (q > 0)
%                -mv= matrix of expansion coefficients
%                -nmax= maximum order
%                -KF= function code:  KF=1 even-even
%                                     KF=2 even-odd
%                                     KF=3 odd-even
%                                     KF=4 odd-odd
%                          -O = order of the matrix (maximum
    number of coefficients)
%   OUTPUTS:     -y= vector of function values for all 'nmax'
    orders
%   'mv'is determined beforehand with function 'eig_Spm'
%   This Radial Mathieu Function is approximated by an
    expansion of
%   products of Bessel functions. It is related to Spm(KF,i*u
    ,mv,nmax,0) by
%   Spm(KF,i*u,mv,nmax,0)=sqrt(2*pi)*gpm(KF,q,mv,nmax,0)*Jpm(
    KF,u,q,mv,nmax,0)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    E. Cojocaru, revised by H. Calunga in January 2024
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Jpm FUNCTION CALL
function y = Jpm(KF,u,q,mv,nmax,0)
```

```
53
54      v1=sqrt(q)*exp(-u);
55      v2=sqrt(q)*exp(u);
56  %---------------even-even---KF == 1--------------------
57  if KF == 1
58      ik=0:(O-1); vt=2*ik;
59      ncoeffs=length(vt);
60      y=[];
61      for k=1:nmax
62      Apm=mv(:,k);  in=vt(k);
63      A0=Apm(1);
64      aApm=abs(Apm);
65      ss1 = find(aApm==max(aApm));
66      A0S=Apm(ss1);
67      s1=ss1-1;
68           yc=A0*(besselj(0-s1,v1)*besselj(0+s1,v2)+besselj(0+
                s1,v1)*besselj(0-s1,v2));
69              for j=2:ncoeffs
70      jm=fix(j-1);
71      yc = yc + ((-1)^jm)*Apm(j)*(besselj(jm-s1,v1)*besselj(jm+
           s1,v2)+ ...
72                           besselj(jm+s1,v1)*besselj(jm-s1,v2));
73              end
74      r=fix(in/2);
75          if s1 == 0
76          coef=((-1)^r)*sqrt(pi/2)/(2*A0S);
77          else
78             coef=((-1)^r)*sqrt(pi/2)/A0S;
79          end
80      yc=yc*coef;
81      y=[y yc];
```

```
82      end
83  %---------------even-odd---KF == 2---------------------
84  elseif KF == 2
85      ik=0:(O-1);     vt=2*ik+1;
86      ncoeffs=length(vt);
87      y=[];
88      for k=1:nmax
89      Apm=mv(:,k);  in=vt(k);
90      A1=Apm(1);
91      aApm=abs(Apm);
92      ss1 = find(aApm==max(aApm));
93      A1S=Apm(ss1);
94      s1=ss1-1;
95           yc=A1*(besselj(1+s1,v2)*besselj(O-s1,v1)+besselj(1+
                s1,v1)*besselj(O-s1,v2));
96              for j=2:ncoeffs
97      jm=fix(j-1);
98      yc = yc + ((-1)^jm)*Apm(j)*(besselj(jm-s1,v1)*besselj(j+
           s1,v2)+ ...
99                          besselj(jm-s1,v2)*besselj(j+s1,v1));
100             end
101     r=fix((in-1)/2);
102     coef=((-1)^r)*sqrt(pi/2)/A1S;
103     yc=yc*coef;
104     y=[y yc];
105     end
106 %---------------odd-even--KF == 3---------------------
107 elseif KF == 3
108     ik=1:O;    vt=2*ik;
109     ncoeffs=length(vt);
110     y=[];
```

```
111     for k=1:nmax
112     Apm=mv(:,k);  in=vt(k);
113     A2=Apm(1);
114     aApm=abs(Apm);
115     s1 = find(aApm==max(aApm));
116     A2S=Apm(s1);
117         yc = A2*(besselj(1-s1,v2)*besselj(1+s1,v1)-besselj(1+
                s1,v2)*besselj(1-s1,v1));
118              for j=2:ncoeffs
119     jm=fix(j-s1);
120     jp=fix(j+s1);
121     yc = yc + ((-1)^j)*Apm(j)*(besselj(jp,v2)*besselj(jm,v1)-
            ...
122                         besselj(jm,v2)*besselj(jp,v1));
123             end
124     r=fix(in/2);
125     coef=((-1)^r)*sqrt(pi/2)/A2S;
126     yc=yc*coef;
127     y=[y yc];
128     end
129 %---------------odd-odd--KF == 4----------------------
130 elseif KF == 4
131     ik=0:(O-1);   vt=2*ik+1;
132     ncoeffs=length(vt);
133     y=[];
134     for k=1:nmax
135     Apm=mv(:,k);  in=vt(k);
136     A1=Apm(1);
137      aApm=abs(Apm);
138     ss1 = find(aApm==max(aApm));
139     A1S=Apm(ss1);
```

```
    s1=ss1-1;
         yc = A1*(besselj(1+s1,v2)*besselj(0-s1,v1)-besselj
            (0-s1,v2)*besselj(1+s1,v1));
            for j=2:ncoeffs
    jm=fix(j-1);
    yc = yc + ((-1)^jm)*Apm(j)*(besselj(j+s1,v2)*besselj(jm-
       s1,v1)- ...
                        besselj(jm-s1,v2)*besselj(j+s1,v1));
           end
    r=fix((in-1)/2);
    coef=((-1)^r)*sqrt(pi/2)/A1S;
    yc=yc*coef;
    y=[y yc];
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   NORMALIZATION FACTOR OF ANGULAR MATHIEU FUNCTIONS
%   y = Npm(KF,mv,nmax,O)       [p,m = e,o (even,odd)]
%
%   INPUTS:      -mv= matrix of expansion coefficients
%                -nmax= maximum order
%                -KF= function code:  KF=1 even-even
%                                     KF=2 even-odd
%                                     KF=3 odd-even
%                                     KF=4 odd-odd
%                          -O = order of the matrix (maximum
   number of coefficients)
%   OUTPUTS:     -y= vector of normalization factors for all '
   nmax' orders
%   'mv' is determined beforehand with function 'eig_Spm'
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    E. Cojocaru, revised by H. Calunga in January 2024
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Npm FUNCTION CALL
function y = Npm(KF,mv,nmax,O)
%---------------even-even---KF == 1--------------------
if KF == 1
    ik=0:(O-1);  vt=2*ik;
    ncoeffs=length(vt);
    y=[];
    for k=1:nmax
    Apm=mv(:,k);
    A0=Apm(1);
         yc = 2*pi*A0^2;
            for j = 2:ncoeffs
                yc = yc + pi * Apm(j) * Apm(j);
            end
    y=[y yc];
    end
%---------------even-odd----KF == 2-----------------------
%---------------odd-even ---KF == 3-----------------------
%---------------odd-odd ----KF == 4-----------------------
elseif KF ~= 1
    y=[];
    for k=1:nmax
    Apm=mv(:,k);
    yc = pi * sum(Apm.*Apm);
    y=[y yc];
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   ANGULAR MATHIEU FUNCTION
%   y = Spm(KF,v,mv,nmax,O)    [p,m = e,o (even,odd)]
%
%   INPUTS:      -v= value of angular coordinate in radians
%                -mv= matrix of expansion coefficients
%                -nmax= maximum order
%                -KF= function code:  KF=1 even-even
%                                     KF=2 even-odd
%                                     KF=3 odd-even
%                                     KF=4 odd-odd
%                          -O = order of the matrix (maximum
   number of coefficients)
%   OUTPUTS:     -y= vector of function values for all 'nmax'
   orders
%   'mv' is determined beforehand with function 'eig_Spm'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    E. Cojocaru, revised by H. Calunga in January 2024
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Spm FUNCTION CALL
function y = Spm(KF,v,mv,nmax,O)
%---------------even-even---KF == 1--------------------
if KF == 1
    ik=0:(O-1);  vt=2*ik;
    ncoeffs=length(vt);
    y=[];
    for k=1:nmax
    Apm=mv(:,k);
         yc = Apm(1);
            for j= 2:ncoeffs
                jp = fix(2*(j-1));
```

```
                yc= yc + Apm(j)*cos(jp*v);
            end
    y=[y yc];
    end
%---------------even-odd---KF == 2-------------------
elseif KF == 2
    ik=0:(O-1);  vt=2*ik+1;
    ncoeffs=length(vt);
    y=[];
    for k=1:nmax
    Apm=mv(:,k);
    yc = 0;
            for j = 1:ncoeffs
                yc = yc + Apm(j)*cos((2*j-1)*v);
            end
     y=[y yc];
     end
%---------------odd-even---KF == 3----------------
elseif KF == 3
    ik=1:O;  vt=2*ik;
    ncoeffs=length(vt);
    y=[];
    for k=1:nmax
    Apm=mv(:,k);
    yc = 0;
            for j = 1:ncoeffs
                yc = yc + Apm(j)*sin(2*j*v);
            end
     y=[y yc];
     end
%---------------odd-odd---KF == 4-----------------
```

```
elseif KF == 4
    ik=0:(O-1);  vt=2*ik+1;
    ncoeffs=length(vt);
    y=[];
    for k=1:nmax
    Apm=mv(:,k);
    yc = 0;
            for j = 1:ncoeffs
                yc = yc + Apm(j)*sin((2*j-1)*v);
            end
     y=[y yc];
     end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   RADIAL MATHIEU FUNCTION OF THE SECOND KIND
%   y = Ypm(KF,u,q,mv,nmax,O)     [p,m = e,o (even,odd)]
%
%   INPUTS:      -u= value of radial coordinate to compute
   function
%               -q= elliptical parameter (q > O)
%               -mv= matrix of expansion coefficients
%               -nmax= maximum order
%               -KF= function code:  KF=1 even-even
%                                    KF=2 even-odd
%                                    KF=3 odd-even
%                                    KF=4 odd-odd
%                        -O = order of the matrix (maximum
   number of coefficients)
%   OUTPUTS:     -y= vector of function values for all 'nmax'
   orders
```

```
15 %   'mv' is determined beforehand with function 'eig_Spm'
16 %   The Radial Mathieu Function is approximated by an
       expansion
17 %   of product of Bessel functions.
18 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
19 %    E. Cojocaru, revised by H. Calunga in January 2024
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21 %   Ypm FUNCTION CALL
22 function y = Ypm(KF,u,q,mv,nmax,0)
23     v1=sqrt(q)*exp(-u);
24     v2=sqrt(q)*exp(u);
25 %---------------even-even---KF == 1---------------
26 if KF == 1
27     ik=0:(0-1);    vt=2*ik;
28     ncoeffs=length(vt);
29     y=[];
30     for k=1:nmax
31     Apm=mv(:,k);  in=vt(k);
32     A0=Apm(1);
33     aApm=abs(Apm);
34     ss1 = find(aApm==max(aApm));
35     A0S=Apm(ss1);
36     s1=ss1-1;
37          yc = A0*(besselj(0-s1,v1)*bessely(0+s1,v2) + besselj
               (0+s1,v1)*bessely(0-s1,v2));
38             for j=2:ncoeffs
39     jm=fix(j-1);
40     yc = yc + ((-1)^jm)*Apm(j)*(besselj(jm-s1,v1)*bessely(jm+
          s1,v2)+...
41                          besselj(jm+s1,v1)*bessely(jm-s1,v2));
42             end
```

```
43      r=fix(in/2);
44          if s1 == 0
45                  coef=((-1)^r)*sqrt(pi/2)/(2*A0S);
46          else
47                  coef=((-1)^r)*sqrt(pi/2)/A0S;
48          end
49      yc=yc*coef;
50      y=[y yc];
51      end
52  %---------------even-odd---KF == 2-----------------
53  elseif KF == 2
54      ik=0:(O-1);  vt=2*ik+1;
55      ncoeffs=length(vt);
56      y=[];
57      for k=1:nmax
58      Apm=mv(:,k);  in=vt(k);
59      A1=Apm(1);
60      aApm=abs(Apm);
61      ss1 = find(aApm==max(aApm));
62      A1S=Apm(ss1);
63      s1=ss1-1;
64          yc = A1*(besselj(0-s1,v1)*bessely(1+s1,v2)+bessely(0-
               s1,v2)*besselj(1+s1,v1));
65              for j=2:ncoeffs
66      jm=fix(j-1);
67      yc = yc + ((-1)^jm)*Apm(j)*(besselj(jm-s1,v1)*bessely(j+
           s1,v2)+ ...
68                          bessely(jm-s1,v2)*besselj(j+s1,v1));
69              end
70      r=fix((in-1)/2);
71      coef=((-1)^r)*sqrt(pi/2)/A1S;
```

```
72      yc=yc*coef;
73      y=[y yc];
74      end
75  %----------------odd-even--KF == 3------------------
76  elseif KF == 3
77      ik=1:0;  vt=2*ik;
78      ncoeffs=length(vt);
79      y=[];
80      for k=1:nmax
81      Apm=mv(:,k);  in=vt(k);
82      A2=Apm(1);
83       aApm=abs(Apm);
84      s1 = find(aApm==max(aApm));
85      A2S=Apm(s1);
86          yc = A2*(bessely(1-s1,v2)*besselj(1+s1,v1)-bessely(1+
               s1,v2)*besselj(1-s1,v1));
87               for j=2:ncoeffs
88      jm=fix(j-s1);
89      jp=fix(j+s1);
90      yc = yc + ((-1)^j)*Apm(j)*(bessely(jp,v2)*besselj(jm,v1)-
            ...
91                          bessely(jm,v2)*besselj(jp,v1));
92              end
93      r=fix(in/2);
94      coef=((-1)^r)*sqrt(pi/2)/A2S;
95      yc=yc*coef;
96      y=[y yc];
97      end
98  %----------------odd-odd--KF == 4---------------------
99  elseif KF == 4
100     ik=0:(0-1);  vt=2*ik+1;
```

```
101     ncoeffs=length(vt);
102     y=[];
103     for k=1:nmax
104     Apm=mv(:,k);  in=vt(k);
105     A1=Apm(1);
106     aApm=abs(Apm);
107     ss1 = find(aApm==max(aApm));
108     A1S=Apm(ss1);
109     s1=ss1-1;
110         yc = A1*(bessely(1+s1,v2)*besselj(0-s1,v1)-bessely(0-
               s1,v2)*besselj(1+s1,v1));
111              for j=2:ncoeffs
112     jm=fix(j-1);
113     yc = yc + ((-1)^jm)*Apm(j)*(bessely(j+s1,v2)*besselj(jm-
           s1,v1)- ...
114                         bessely(jm-s1,v2)*besselj(j+s1,v1));
115             end
116     r=fix((in-1)/2);
117     coef=((-1)^r)*sqrt(pi/2)/A1S;
118     yc=yc*coef;
119     y=[y yc];
120     end
121 end
```

Appendix III

MFS-SVD Matlab code for biharmonic BVP in polar coordinates, Example 4.7 approximation A_2

```
V = 10:10:500;
Ptos_colloc = zeros(1, numel(V));
Ptos_fonte = zeros(1, numel(V));
Err_max = zeros(1, numel(V));
Err_relativo = zeros(1, numel(V));
Err_L2 = zeros(1, numel(V));
NrCond = zeros(1, numel(V));
NrPontos = zeros(1, numel(V));
Err_maxsvd = zeros(1, numel(V));
Err_relativosvd = zeros(1, numel(V));
Err_L2svd = zeros(1, numel(V));
NrCondsvd = zeros(1, numel(V));
NrCondPf = zeros(1, numel(V));
NrCondPc = zeros(1, numel(V));
%%%%%%%%%%%%
count = 1;
for N = V
  M = 2*N;
 alpha = 1.5;       % distance between border and source
    points surface
r = 1.0;                      % radius of circumference
RR = Normalizacao(r);
T = 250; % Truncature of the addition theorem
th=(1:M)'/M*2*pi;
Rb = r*ones(size(th));
xb = Rb.*cos(th);
yb = Rb.*sin(th);
```

```
BP = [xb(:), yb(:)];
th_sp =(1:N)'/N*2*pi;
Rs = r*alpha*ones(size(th_sp));
xs = Rs.*cos(th_sp);
ys = Rs.*sin(th_sp);
SP = [xs(:), ys(:)];
BPsh =[Rb(:), th(:)];
SPsh =[Rs(:), th_sp(:)];
% Construindo o sistema e resolvendo-o
DM= pdist2(BP,SP);
A = matrizfundamental(BP,SP,DM);
rhsf = rhs2(BP);
a=A\rhsf;
% Computation of num. sol.
qtpts=33; %numero de divisoes
dr = linspace(0,r,qtpts);
dth = (1:qtpts)'/qtpts*2*pi;
[Rr, Tt] = meshgrid(dr,dth);
pto_aval_sp =[Rr(:), Tt(:)];
xxx = Rr.*cos(Tt);
yyy = Rr.*sin(Tt);
pto_aval = [xxx(:) yyy(:)];
% Computation of error
SolExa =  exas2(pto_aval);
ExaSol = SolExa; %Sol. Exacta
D_eval= pdist2(pto_aval,SP);
[AD] = MatrizAvaliacaoCl(pto_aval,SP,D_eval);
NumSolad = solnumerica(AD,a); %Sol. Numerica
[D, L, Err_rel] = todoserros(NumSolad,ExaSol);
Nrcon_bef = cond(full(A));
fprintf("MfS Solution \n")
```

```
fprintf(" Nr Colocation points: %d \n",size(BP,1))
fprintf(" Nr Source points: %d \n",size(SP,1))
fprintf("L2 Error: %1.5E \n",max(sqrt(L./size(NumSolad,1))))
fprintf("Cond. No.: %1.5E \n",Nrcon_bef)
fprintf("No arbitrary points: %d \n",size(pto_aval,1))
Ptos_colloc(1:1, count:count) = size(BP,1);
Ptos_fonte(1:1, count:count) = size(SP,1);
Err_max(1:1, count:count) = max(D);
Err_relativo(1:1, count:count) =  max(Err_rel);
Err_L2(1:1, count:count) = max(sqrt(L./size(NumSolad,1)));
NrCond(1:1, count:count) = Nrcon_bef;
NrPontos(1:1, count:count) = size(pto_aval,1);
% Calculo do Erro pelo Teorema da adicao
MPF = MatrizPF(SPsh,T,RR);
V1 = Regularizacao(MPF,size(SP,1));
MMreg = matrizfundreg(V1,BPsh,SPsh,T,RR) ;
aa = MMreg\rhsf;
[ADreg] = MatrizAvalSVD(V1,pto_aval_sp,SPsh,T,RR);
NumSolreg = solnumerica(ADreg,aa);
[Dreg, Lreg, Err_relreg] = todoserros(NumSolreg,ExaSol);
Nrcon_aft = cond(full(MMreg));
fprintf(" MFS-SVD output\n")
fprintf("L2 error: %1.5E \n",max(sqrt(Lreg./size(NumSolreg,1)
   )))
fprintf(" Cond. No: %1.5E \n",Nrcon_aft)
Err_maxsvd(1:1, count:count) = max(Dreg);
Err_relativosvd(1:1, count:count) = max(Err_relreg);
Err_L2svd(1:1, count:count) = max(sqrt(Lreg./size(NumSolreg
   ,1)));
NrCondsvd(1:1, count:count) = Nrcon_aft;
count = count + 1;
```

```
87 Aresults = [Ptos_colloc; Ptos_fonte; Err_max; Err_relativo;
      Err_L2; NrCond];
88 Aresultssvd = [Err_maxsvd;Err_relativosvd; Err_L2svd;
      NrCondsvd ];
89 dlmwrite('myResultsfile_disk_r_1_5_rhs2_out.csv',Aresults);
90 dlmwrite('myResultsfile_disk_r_1_5_rhs2_out.csv',Aresultssvd,
      '-append','roffset',1);
91 end
```

```
1  function [G] = rhs2(BP)
2  % BP: matriz dos pontos fronteira
3  sP=[1.1,0]; %  sP= [3.0,0];
4  MM = size(BP,1);
5  for p=1:MM
6     Rq= norm(BP(p,:)-sP);
7     G1(p,1)= -1/(8*pi).*Rq.^2.*log(Rq);
8     vnxr=[BP(p,1),BP(p,2)];
9     nxr=(1/norm(vnxr)).*vnxr;
10    G2(p,1)=nxr(1,1).*(-1/(8*pi)*(2*Rq*log(Rq)+Rq)).*(BP(p,1)-
         sP(1,1))/Rq + nxr(1,2).*(-1/(8*pi)*(2*Rq*log(Rq)+Rq))
         .*(BP(p,2)-sP(1,2))/Rq;
11 end
12  G =[G1;G2];
13 end
```

```
1  function [G] = exas2(BP)
2  % BP: matriz dos pontos fronteira
3  sP= [1.1,0]; % sP= [3.0,0];
4  MM = size(BP,1);
5  for p=1:MM
6   G1(p,1)=-1/(8*pi).*norm(BP(p,:)-sP).^2.*log(norm(BP(p,:)-sP)
       );
```

```
end
 G = G1;
end
```

```
function [MAC] = MatrizAvaliacaoCl(BP,SP,DM)
% BP, SP boundary and source points & DM: distace
xmenosx0= repmat(BP(:,1),1, size(SP,1))-repmat(SP(:,1)', size
   (BP,1),1);
ymenosy0= repmat(BP(:,2),1, size(SP,1))-repmat(SP(:,2)', size
   (BP,1),1);
delG2dr = @(r) -1./(8*pi).*(2.*r.*log(r) + r);
del2G2dr2 = @(r) -1./(8*pi).*(2.*log(r) + 3);
delrdx = @(r,x) x./r;
delrdy = @(r,y) -y./r;
del2rdxdy = @(r,x) -(x.^2)./r.^3;
del2rdxy = @(r,x,y) (x.*y)./r.^3;
%vector normal para o caso de uma circuferencia ny=(x,y)
N = zeros(size(BP,1),size(SP,1));
P = zeros(size(BP,1),size(SP,1));
for p=1:size(BP,1)
   for j=1:size(SP,1)
        vny=[SP(j,1),SP(j,2)];
        ny=(1/norm(vny)).*vny;
    N(p,j)= -1./(8*pi).*(DM(p,j).^2).*log(DM(p,j));
    P(p,j)= ny(1,1).*delG2dr(DM(p,j)).*delrdy(DM(p,j),
       xmenosx0(p,j)) + ny(1,2).*delG2dr(DM(p,j)).*delrdy(DM(
       p,j),ymenosy0(p,j));
  end
end
MAC =[N,P];
end
```

```
function [MSVD] = MatrizAvalSVD(W,BPs,SPs,T,Rd)
% SPs, BPs: Source and boundary points & T truncature
Npc = size(BPs,1);
Npf = size(SPs,1);
MM = sparse(2*Npc,2*Npf);
NP = sparse(Npc,2*Npf);
 for pq = 1:Npc
    [MF1,~]= MatrizPC(BPs(pq,:),T,Rd);
    TEMP1 = MF1;
    Line1 = W*TEMP1;
    NP(pq:pq, 1:2*Npf) = Line1.';
 end
 MSVD = NP;
end
```

```
function [MG] = matrizfundamental(BP,SP,DM)
% BP,SP boundary and souce points & DM:matrix of distances
% Partial derivatives
xmenosx0= repmat(BP(:,1),1, size(SP,1))-repmat(SP(:,1)', size
   (BP,1),1);
ymenosy0= repmat(BP(:,2),1, size(SP,1))-repmat(SP(:,2)', size
   (BP,1),1);
delG2dr = @(r) -1./(8*pi).*(2.*r.*log(r) + r);
del2G2dr2 = @(r) -1./(8*pi).*(2.*log(r) + 3);
delrdx = @(r,x) x./r;
delrdy = @(r,y) -y./r;
del2rdxdy = @(r,x) -(x.^2)./r.^3;
del2rdxy = @(r,x,y) (x.*y)./r.^3;
%vector normal para o caso de uma circuferencia ny=(x,y)
N = zeros(size(BP,1),size(SP,1));
P = zeros(size(BP,1),size(SP,1));
```

```
DN = zeros(size(BP,1),size(SP,1));
DP = zeros(size(BP,1),size(SP,1));
for p=1:size(BP,1)
   for j=1:size(SP,1)
        vny=[SP(j,1),SP(j,2)];
        vnx=[BP(p,1),BP(p,2)];
        ny=(1/norm(vny)).*vny;
        nx=(1/norm(vnx)).*vnx;
    N(p,j)= -1./(8*pi).*(DM(p,j).^2).*log(DM(p,j));
    P(p,j)= ny(1,1).*delG2dr(DM(p,j)).*delrdy(DM(p,j),
       xmenosx0(p,j)) + ny(1,2).*delG2dr(DM(p,j)).*delrdy(DM(
       p,j),ymenosy0(p,j));
    DN(p,j)= nx(1,1).*delG2dr(DM(p,j)).*delrdx(DM(p,j),
       xmenosx0(p,j)) + nx(1,2).*delG2dr(DM(p,j)).*delrdx(DM(
       p,j),ymenosy0(p,j));
    DP(p,j)= nx(1,1).*ny(1,1).*(del2G2dr2(DM(p,j)).*delrdx(DM
       (p,j),xmenosx0(p,j)).*delrdy(DM(p,j),xmenosx0(p,j)) +
       delG2dr(DM(p,j)).*del2rdxdy(DM(p,j),ymenosy0(p,j))) +
       ...
    nx(1,1).*ny(1,2).*(del2G2dr2(DM(p,j)).*delrdx(DM(p,j),
       xmenosx0(p,j)).*delrdy(DM(p,j),ymenosy0(p,j)) +
       delG2dr(DM(p,j)).*del2rdxy(DM(p,j),xmenosx0(p,j),
       ymenosy0(p,j))) + ...
    nx(1,2).*ny(1,1).*(del2G2dr2(DM(p,j)).*delrdx(DM(p,j),
       ymenosy0(p,j)).*delrdy(DM(p,j),xmenosx0(p,j)) +
       delG2dr(DM(p,j)).*del2rdxy(DM(p,j),xmenosx0(p,j),
       ymenosy0(p,j))) + ...
    nx(1,2).*ny(1,2).*(del2G2dr2(DM(p,j)).*delrdx(DM(p,j),
       ymenosy0(p,j)).*delrdy(DM(p,j),ymenosy0(p,j)) +
       delG2dr(DM(p,j)).*del2rdxdy(DM(p,j),xmenosx0(p,j)));
  end
```

```
end
MG =[N,P;DN,DP];
end
```

```
function [MM] = matrizfundreg(W,BPs,SPs,T,Rd)
% SPs,BPs, source, boundary poionts matrix & truncature
Npc = size(BPs,1);
Npf = size(SPs,1);
MM = sparse(2*Npc,2*Npf);
NP = sparse(Npc,2*Npf);
DNDP = sparse(Npc,2*Npf);
 for pq = 1:Npc
    [MF1,MF2]= MatrizPC(BPs(pq,:),T,Rd);
    TEMP1 = MF1;
    TEMP2 = MF2;
    Line1 = W*TEMP1;
    Line2 = W*TEMP2;
    NP(pq:pq, 1:2*Npf) = Line1.';
    DNDP(pq:pq, 1:2*Npf) = Line2.' ;
 end
 MM = [NP; DNDP];
end
```

```
function [MPc1, MPc2] = MatrizPC(PP,T,Rd)
% BPs: boundary points & T: truncature
F1 = zeros(1, 6+4*(T-1));
F2 = zeros(1, 6+4*(T-1));
ct = 7 ;
for jj = 0:1:T
 if jj == 0
  F1(1) = 1;
  F1(2) = PP(1,1).^2/Rd.^2;
```

```
  F2(1) = 0;
  F2(2) = 2*PP(1,1)/Rd.^2;
 elseif jj == 1
  F1(3) = PP(1,1).*cos(PP(1,2))/Rd;
  F1(4) = PP(1,1).*sin(PP(1,2))/Rd;
  F1(5) = PP(1,1).^3.*cos(PP(1,2))/Rd.^3;
  F1(6) = PP(1,1).^3.*sin(PP(1,2))/Rd.^3;
  F2(3) = cos(PP(1,2))/Rd;
  F2(4) = sin(PP(1,2))/Rd;
  F2(5) = 3*PP(1,1).^2*cos(PP(1,2))/Rd.^3;
  F2(6) = 3*PP(1,1).^2*sin(PP(1,2))/Rd.^3;
  else
    F1(ct)= PP(1,1).^jj.*cos(jj.*PP(1,2))/Rd.^jj;
    F1(ct+1)= PP(1,1).^jj.*sin(jj.*PP(1,2))/Rd.^jj;
    F1(ct+2)= PP(1,1).^(jj+2).*cos(jj.*PP(1,2))/Rd.^(jj+2);
    F1(ct+3)= PP(1,1).^(jj+2).*sin(jj.*PP(1,2))/Rd.^(jj+2);
    F2(ct)= jj*PP(1,1).^(jj-1)*cos(jj.*PP(1,2))/Rd.^jj;
    F2(ct+1)= jj*PP(1,1).^(jj-1)*sin(jj.*PP(1,2))/Rd.^jj;
 F2(ct+2)=(jj+2)*PP(1,1).^(jj+1)*cos(jj.*PP(1,2))/Rd.^(jj+2);
 F2(ct+3)=(jj+2)*PP(1,1).^(jj+1)*sin(jj.*PP(1,2))/Rd.^(jj+2);
    ct = ct+4;
  end
 end
      MPc1 = F1.';
      MPc2 = F2.';
end
```

```
function [MPf] = MatrizPF(SPs,T,Rd)
% SPs: source points matrix e & T: truncature
Npf = size(SPs,1);
B1 = sparse(Npf,6+4*(T-1));
```

```
5  B2 = sparse(Npf,6+4*(T-1));
6  bB1 = zeros(1, 6+4*(T-1));
7  bB2 = zeros(1, 6+4*(T-1));
8  ct1 = 7;
9  % First factor of addithim teorem decompostiom
10  for pp = 1:Npf
11     nyy=[cos(SPs(pp,2)),sin(SPs(pp,2))];
12     a_alfa=nyy(1,1).*cos(SPs(pp,2))+nyy(1,2).*sin(SPs(pp,2));
13    for ii = 0:1:T
14     if ii == 0
15      bB1(1) = SPs(pp,1).^2.*log(SPs(pp,1));
16      bB1(2) = (1+log(SPs(pp,1)))*Rd.^2;
17      bB2(1) = a_alfa*(2*SPs(pp,1)*log(SPs(pp,1))+SPs(pp,1));
18      bB2(2) = a_alfa*(1/SPs(pp,1))*Rd.^2;
19     elseif ii==1
20      bB1(3) = -(2*SPs(pp,1).*log(SPs(pp,1)) + SPs(pp,1)).*cos
           (SPs(pp,2))*Rd ;
21      bB1(4) = -(2*SPs(pp,1).*log(SPs(pp,1)) + SPs(pp,1)).*sin
           (SPs(pp,2))*Rd;
22      bB1(5) = -(1/(2*SPs(pp,1))).*cos(SPs(pp,2))*Rd.^3;
23      bB1(6) = -(1/(2*SPs(pp,1))).*sin(SPs(pp,2))*Rd.^3;
24      bB2(3)=-a_alfa*(2*log(SPs(pp,1))+3).*cos(SPs(pp,2))*Rd;
25      bB2(4)=-a_alfa*(2*log(SPs(pp,1))+3).*sin(SPs(pp,2))*Rd;
26      bB2(5)=a_alfa*1/(2*SPs(pp,1).^2).*cos(SPs(pp,2))*Rd.^3;
27      bB2(6)=a_alfa*1/(2*SPs(pp,1).^2).*sin(SPs(pp,2))*Rd.^3;
28     else
29      bB1(ct1) = (1/(ii*(ii-1)*SPs(pp,1).^(ii-2))).*cos(ii*SPs
           (pp,2))*Rd.^ii;
30      bB1(ct1+1) = (1/(ii*(ii-1)*SPs(pp,1).^(ii-2))).*sin(ii*
           SPs(pp,2))*Rd.^ii;
31      bB1(ct1+2) = -(1/(ii*(ii+1)*SPs(pp,1).^(ii))).*cos(ii*
```

```
        SPs(pp,2))*Rd.^(ii+2);
32      bB1(ct1+3) = -(1/(ii*(ii+1)*SPs(pp,1).^(ii))).*sin(ii*
        SPs(pp,2))*Rd.^(ii+2);
33      bB2(ct1) = -a_alfa*((ii-2)/(ii*(ii-1)*SPs(pp,1).^(ii-1))
        ).*cos(ii*SPs(pp,2))*Rd.^(ii);
34      bB2(ct1+1) = -a_alfa*((ii-2)/(ii*(ii-1)*SPs(pp,1).^(ii
        -1))).*sin(ii*SPs(pp,2))*Rd.^(ii);
35      bB2(ct1+2) = a_alfa*(1/((ii+1)*SPs(pp,1).^(ii+1))).*cos(
        ii*SPs(pp,2))*Rd.^(ii+2);
36      bB2(ct1+3) = a_alfa*(1/((ii+1)*SPs(pp,1).^(ii+1))).*sin(
        ii*SPs(pp,2))*Rd.^(ii+2);
37      ct1 = ct1+4;
38     end
39    end
40          ct1 = 7;
41          B1(pp:pp, 1:6+4*(T-1)) = bB1 ;
42          B2(pp:pp, 1:6+4*(T-1)) = bB2 ;
43  end
44  MPf = [B1;B2];
45  end
```

```
1   function [RF] = Normalizacao(rr)
2   % rr:radius of the disk for other domaisn no input
3   H = 1000; % arbitrary to define min and max radius
4   vth=(1:H)'/H*2*pi;
5   xc = rr.*cos(vth); % boundary of domain
6   yc = rr.*sin(vth);
7   BPc=sqrt(xc.^2+yc.^2);
8   RF= max(BPc(:,1));
9   end
```

```
1   function Mreg = Regularizacao(Blk,N)
```

```
% N: No of source points
   [~,~,V] = svd(full(Blk));
   VV1 = V';
   Mreg =VV1(1:2*N,:) ;
end
```

For functions *solnumerica* and *todoserros* refer to Appendix I.