Spectral approximation with matrices issued from discretized operators

Ana Luisa Silva Nunes

Universidade do Porto
et
Université Jean Monnet

N° attribué par la bibliothèque
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣

THÈSE

pour obtenir le grade de

Docteur de l'Universidade do Porto et Université Jean Monnet

Spécialité : **Mathématiques Appliquées**

dans le cadre de l' **Ecole Doctorale Sciences, Ingénierie, Santé : ED SIS 488**

présentée et soutenue publiquement
par

Ana Luísa da Silva Nunes

le 11/05/2012

Titre:
Approximation spectrale de matrices issues d'opératreurs discrétisés

Directeurs de thèse: **Paulo Beleza Vasconcelos**
Co-directeur de thèse: **Mario Ahues Blanchait**

Jury

M.	Jorge ROCHA,	Président de Jury
Mme	Rekha KULKARNI,	Rapporteur
Mme	Maria Joana SOARES,	Examinateur, Rapporteur
Mme	Laurence GRAMMONT,	Examinateur
Mme	Teresa DIOGO,	Examinateur
M.	Paulo BELEZA VASCONCELOS,	Directeur de thèse
M.	Mario AHUES BLANCHAIT,	Co-directeur de thèse

To my parents

Acknowledgments

Arriving to this stage, I am glad to complete it by remembering many wonderful people who became involved with this thesis in various ways.

First and foremost I want to thank my thesis supervisors. I am extremely grateful for the guidance and energetic support provided by Professor Paulo Vasconcelos and for all the insightful and always fast feedback whenever required, despite his many other academic commitments. I deeply acknowledge Professor Mario Ahues for all the time he spent providing very important suggestions and comments when it was most necessary. I also thank him for his generous and tireless assistance in my stays in Saint-Etienne, as well as in all my registations at the Université Jean Monnet. It has been an honor to work with both of them, and I am deeply grateful to them for their scientific and general support and encouragement throughout my study as a PhD. student.

I express my sincere thanks to Professor J. E. Roman from Universidad Politecnica de Valencia (Spain), for all his support in the joint computacional work held in this thesis.

I wish to honor the memory of Professor Alain Largillier, with whom I had some scientific profitable talks about my work and helped to make my first stay in Saint-Etienne a pleasant experience.

I am very grateful to my colleagues of Université Jean Monnet, specially to Rola Fares and Taline Boyajian, for the hospitality I experienced in my first stay there.

I also deeply appreciate all the support shown by my friends and IPCA colleagues.

I wish to express my warm thanks to my beloved husband, Paolo, for his love and understanding in every situation.

Last but not least, a very special thank to my wonderful parents for their endless love, patience and encouragement.

Without you all, I could not have reached this far. Thank you!

Ana Luísa

Spectral approximation with matrices issued from discretized operators

Abstract

In this thesis, we consider the numerical solution of a large eigenvalue problem in which the integral operator comes from a radiative transfer problem.

It is considered the use of hierarchical matrices, an efficient data-sparse representation of matrices, especially useful for large dimensional problems. It consists on low-rank subblocks leading to low memory requirements as well as cheap computational costs.

We discuss the use of the hierarchical matrix technique in the numerical solution of a large scale eigenvalue problem arising from a finite rank discretization of an integral operator. The operator is of convolution type, it is defined through the first exponential-integral function and hence it is weakly singular.

We access HLIB (Hierarchical matrices LIBrary) that provides, among others, routines for the construction of hierarchical matrix structures and arithmetic algorithms to perform approximative matrix operations. Moreover, it is incorporated the matrix-vector multiply routines from HLIB, as well as LU factorization for preconditioning, into SLEPc (Scalable Library for Eigenvalue Problem Computations) in order to exploit the available algorithms to solve eigenvalue problems.

It is also developed analytical expressions for the approximate degenerate kernels and deducted error upper bounds for these approximations.

The numerical results obtained with other approaches to solve the problem are used to compare with the ones obtained with this technique, illustrating the efficiency of the techniques developed and implemented in this work.

Keywords with AMS classification - 2010 Mathematics Subject Classification, (MSC[2010]):

33 Special functions; 33F Computational aspects; 33F05 Numerical approximation and evaluation; 45 Integral Equations; 45C Eigenvalue problems; 45C05 Eigenvalue problems; 45E Singular integral equations; 65 Numerical analysis; 65F Numerical linear algebra; 65F15 Eigenvalues, eigenvectors; 65Y Computer aspects of numerical algorithms; 65Y20 Complexity and performance of numerical algorithms.

Aproximação espectral com matrizes e operadores discretizados

Resumo

Nesta dissertação considera-se a solução numérica de um problema de valores próprios de grandes dimensões, no qual o operador provém de um problema de transferência radiativa.

Procede-se ao estudo do uso de matrizes hierárquicas, uma representação eficiente de matrizes, bastante interessante para o uso em problemas de grandes dimensões. Matrizes hierárquicas são representações eficientes de estruturas de dados esparsas de matrizes densamente povoadas, sendo a ideia básica a de dividir uma determinada matriz numa hierarquia de blocos e aproximar determinados blocos por uma matriz de característica pequena. A sua utilização vem permitir, para além da diminuição da memória requerida, a redução dos custos computacionais.

A aplicação do uso das matrizes hierárquicas é analisada no contexto da solução numérica de um problema de valores próprios de grandes dimensões que resulta da discretização de um operador integral. O operador é de convolução e é definido através da primeira função exponencial-integral sendo, desta forma, fracamente singular.

Para o cálculo computacional, acede-se à HLIB (Hierarchical matrices LIBrary) que fornece rotinas para a construção da estrutura das matrizes hierárquicas, bem como algoritmos para operações aproximadas com estas

matrizes. Acrescenta-se que se incorporam algumas rotinas da HLIB, como multiplicação matriz-vector ou a decomposição LU, na SLEPc (Hierarchical matrices LIBrary) de forma a explorar os algoritmos existentes para a resolução de problemas de valores próprios.

Desenvolvem-se ainda expressões analíticas para a aproximação dos núcleos degenerados utilizados na tese e deduzem-se também limites superiores de erros para estas aproximações.

Os resultados numéricos obtidos com outras abordagens para solucionar o problema em questão são utilizados para comparação com os obtidos com a nova técnica, vindo ilustrar a eficiência desta última.

Palavras Chave com classificação MAS - 2010 Mathematics Subject Classification, (MSC[2010]):

33 Special functions; 33F Computational aspects; 33F05 Numerical approximation and evaluation; 45 Integral Equations; 45C Eigenvalue problems; 45C05 Eigenvalue problems; 45E Singular integral equations; 65 Numerical analysis; 65F Numerical linear algebra; 65F15 Eigenvalues, eigenvectors; 65Y Computer aspects of numerical algorithms; 65Y20 Complexity and performance of numerical algorithms.

Approximation spectrale de matrices issues d'opérateurs discrétisés

Résumé

Cette thèse considère la solution numérique d'un problème aux valeurs propres de grandes dimensions, dans lequel l'opérateur est dérivé d'un problème de transfert radiatif.

Ainsi, cette thèse étudie l'utilisation de matrices hiérarchiques, une représentation efficace de tableaux, très intéressante pour une utilisation avec des problèmes de grandes dimensions. Les matrices sont des représentations hiérarchiques de structures de données efficaces pour les matrices denses, l'idée de base étant la division d'une matrice en une hiérarchie de blocs et l´approximation de certains blocs par une matrice de petite caractéristique. Son utilisation permet de diminuer la mémoire nécessaire tout en réduisant les coûts informatiques.

L'application de l'utilisation de matrices hiérarchique est analysée dans le contexte de la solution numérique d'un problème aux valeurs propres de grandes dimensions résultant de la discrétisation d'un opérateur intégral. L'opérateur est de convolution et est défini par la première fonction exponentielle intégrale, donc faiblement singulière.

Pour le calcul informatique, nous avons accès à HLIB (Hierarchical matrices LIBrary) qui fournit des routines pour la construction de la structure hiérarchique des matrices et des algorithmes pour les opérations approxima-

tive avec ces matrices. Nous incorporons certaines routines comme la multiplication matrice-vecteur ou la décomposition LU, en SLEPc (Hierarchical matrices LIBrary) pour explorer les algorithmes existants afin de résoudre les problèmes de valeur propre.

Nous développons aussi des expressions analytiques pour l'approximation des noyaux dégénérés utilisés dans la thèse et déduire ainsi les limites supérieures d'erreur pour ces approximations.

Les résultats numériques obtenus avec d'autres techniques pour résoudre le problème en question sont utilisés pour la comparaison avec ceux obtenus avec la nouvelle technique, illustrant l'efficacité de ce dernier.

Mots-clés avec classification AMS - 2010 Mathematics Subject Classification, (MSC[2010]):

33 Special functions; 33F Computational aspects; 33F05 Numerical approximation and evaluation; 45 Integral Equations; 45C Eigenvalue problems; 45C05 Eigenvalue problems; 45E Singular integral equations; 65 Numerical analysis; 65F Numerical linear algebra; 65F15 Eigenvalues, eigenvectors; 65Y Computer aspects of numerical algorithms; 65Y20 Complexity and performance of numerical algorithms.

Contents

List of Figures

List of Tables

List of Symbols

We now describe the main notations used in this thesis.

T	integral operator
E_i	exponential-integral function of order i
λ	eigenvalue
T_n	finite rank operator
π_n	projection operator
$(\tau_{n,j})_{j=0}^{n}$	family of grids
A	coefficient matrix of the system of linear equations
$A_{i,j}$	matrix entry in the row i and column j
A^T	transpose of matrix A
A^{-1}	inverse of matrix A
$\mathcal{H}$-matrix	hierarchical matrix
$\mathcal{I}$	index set of the basis function
$T_{\mathcal{I}}$	cluster tree
$T_{\mathcal{I}\times\mathcal{I}}$	block cluster tree
$g(.,.)$	kernel function
$\tilde{g}(.,.)$	approximation of the kernel function
diam	diameter
dist	distance
$\mathcal{L}_\rho$	Lagrange polynomial
$\mathrm{sp}\,(L)$	spectrum of the operator L
$\mathrm{sp}_R(F)$	Rhodius spectrum of the operator F

$\mathrm{sp}_N(F)$	Neuberger spectrum of the operator F
$\mathrm{sp}_K(F)$	Kachurovskij spectrum of the operator F
$\mathrm{sp}_D(F)$	Dörfner spectrum of the operator F
$\mathrm{sp}_{FMV}(F)$	Furi-Martelli-Vignoli spectrum of the operator F
$\mathrm{sp}_F(F)$	Feng spectrum of the operator F
$\phi(F)$	Väth phantom of the operator F
$\mathrm{sp}_{CFV}(F,p)$	Calamai-Furi-Vignoli spectrum

Introduction

Eigenvalue problems are considered to be a very important subject of linear algebra, as they appear in many pratical applications in science and engineering. For example, in stability and control, eigenvalues give important information about damping, phase and magnitude of oscillations [MDH98, MMH95].
In [ZLM05] we can perceive some applications where the computation of eigenvalues is of crucial importance:

> "*The computation of eigenvalues and eigenvectors is an important and often time-consuming phase in computer simulations. Without being exhaustive, eigenvalues and eigenvectors are used in the study of nuclear reactor dynamics (stability of neutron fluxes [...]), in finite element dynamic analysis of structural models (e.g., seismic simulations of civil infrastructure [...]), in the design of the next generation of particle accelerators [...], in the definition of a set of eigenfaces in biometric-based identification systems [...], in the solution of Schrödinger's equation in chemistry and physics [...], in the design of microelectromechanical systems (MEMS [...]), and in the study of conformational changes of proteins [...]. Because of the need for higher levels of simulation details and accuracy, the size and complexity of the computations grow as fast as the advancement of the computer hardware. In order to cope with the increasing need of solving eigenvalue problems, various useful numerical algorithms that are suitable for*

solving large-scale eigenvalue problems are developed."

In the last decades, very effective methods have been devised to compute some or all the eigenvalues of a matrix, such as the QR method for the latter case, a standard method for dense matrices of moderate size. With this sophisticated technique, the complete set of eigenvalues and eigenvectors is computed. Considering a real nonsingular square matrix A, an orthogonal matrix Q and an upper triangular matrix R, the QR method consists, briefly, in iterating the following steps: transform A into a hessenberg (tridiagonal if symmetric) matrix H (using e.g. Householder reflections), decompose H in QR, multiply Q and R together in reverse order (i.e., RQ) to form a new matrix H and the diagonal of H will converge to the eigenvalues, for more see [vdV02, GVL96].

In this context, we refer to LAPACK (Linear Algebra PACKage) [ABB99], a library with subroutines to solve the most frequent problems that come about in numerical linear algebra: linear equations, linear least squares problems, singular value decomposition and eigenvalue problems. LAPACK uses for computation calls as much as possible from the Basic Linear Algebra Subprograms (BLAS), that is a library for vector and matrix operations, adapted to the hierarchical memory of today's computers. LAPACK, along with general (dense) structures, has special implementations of the algorithms to deal with special structures, e.g. band matrices.

For large, usually sparse matrices, very effective iterative methods have been devised to approximate the eigenspace associated to any part of the spectrum. Examples of such techniques are the restarted Krylov methods and preconditioned eigensolvers such as Jacobi-Davidson [BDD$^+$00]. Also, these methods are progressively taking shape as high-quality implementations in software libraries such as SLEPc, the Scalable Library for Eigenvalue Problem Computations [HRV05], thus enabling application programmers to cope with challenging problems coming from a wide range of applications. Libraries such as SLEPc try to make problems computationally tractable by combining two main ingredients: (i) exploiting sparsity of the matrices, and

(ii) exploiting parallelism.

The sparsity of matrices is a desirable property that appears, for instance, in the context of partial differential equations with standard discretization techniques such as the finite element method. This situation is very common in practice, and allows iterative methods to be competitive by benefiting from the cheap, linear-cost matrix-vector products. However, there are cases in which the problem is formulated as an integral equation, either from the very nature of the problem or from some special treatment of PDEs such as the boundary element method.

Discretisation schemes like Galerkin or collocation methods are very popular techniques used for solving integral equations numerically. The integral equation is solved by these methods using a system of linear equations, but sparsity of matrices is not guaranteed, so in principle full (dense) storage must be used, making impossible the numerical treatment of large dimensional problems (consequent blow-up in computational cost). Furthermore, large memory requirements and large time consuming are expected even for moderate size ones. Some methods have been developed in order to avoid dealing with discretized dense matrices, for example, using compactly supported orthonormal wavelets to represent the integral operator [BCR91] or hierarchical matrices, $\mathcal{H}$-matrices for brief, introduced by Hackbush and his collaborators [Hac99, HK00], being this last approach our choice in this thesis.

HLIB (Hierarchical matrices LIBrary) is a software that provides, among others, routines for the construction of hierarchical matrix structures and arithmetic algorithms to perform approximate matrix operations. The idea of using $\mathcal{H}$-matrices, is that they provide an inexpensive representation of large densely populated matrices, by means of a decomposition into a hierarchy of blocks, in which most of those, usually far from the diagonal, are computed using a low-rank approximation in factorized form, resulting in a data-sparse storage scheme. It is meant by data-sparse that few data is needed for the

matrix representation, that is, for an $n \times n$-matrix, instead of n^2 entries, only $\mathcal{O}(nk \log n)$ data may be required for a good approximation, where k is the rank of the new representation (it also determines the accuracy of the approximation) [Bör09].

The low-rank blocks' approximation is based on geometrical considerations, i.e., on an admissibility condition involving the notion of distance between two subsets of an index set; testing this condition, it is decided if the blocks of the adjacency matrix are to be approximated, which significantly reduces not only storage requirements but also computational costs. The admissibility condition results naturally from the derivation of the error bound in order to produce convergent approximations.

Being SLEPc already an effcient tool in parallel computing for large and sparse eigenvalue problems, being sparsity of matrices a desirable property and $\mathcal{H}$-matrices an excellent way to achieve that, this thesis accesses all the power of HLIB and incorporates the matrix-vector multiply routines from HLIB into SLEPc.

The main goal of this thesis is to solve an eigenvalue problem arising from an integral formulation of a radiative transfer problem in stellar atmospheres. The kernel of the integral operator under study is weakly singular and the discretization procedure gives rise to large dimensional problems, either due to large integration limits or to fine grids. In the latter, the eigenproblem becomes very hard to solve since the eigenvalues of the discretized operator tend to be clustered. This thesis also aims to show the use of data-sparse structures provided by HLIB in the computation of eigenpairs through the SLEPc library. This is the first time such an integration is done and tested.

The thesis is divided into three parts, each one with two chapters and it is organized as follows. In Chapter 1 is presented the eigenvalue problem where approximate solution is sought, as well as a brief discription of hierarchical matrices along with the state-of-the-art numerical methods to solve large and sparse eigenvalue problems. Chapter 2 considers all the basics about the

software libraries used. These chapters constitute the first part of the thesis, designated by Framework and guidelines. The development of the numerical approximation is done in Part II; in Chapter 3, three possible strategies for obtaining a low-rank approximation are discussed, using $\mathcal{H}$-matrix representation, while on Chapter 4 results of some numerical experiments along with specific details of the implementation are shown. The last two chapters, which embodies Part III, are devoted to conclusions and possible extensions for future research.

Part I

Framework and guidelines

Chapter 1

Theoretical aspects

In this chapter we introduce the eigenvalue problem, its solution and the concept of hierarchical matrices.

1.1 The problem and its discretization

We consider an eigenvalue problem, arising from an integral formulation of a radiative transfer problem in stellar atmospheres [RC04]. The spectral decomposition of an integral operator T has a numerical interest in the integral equation

$$x = f + Tx, \tag{1.1}$$

where x is the source function of the problem and f describes the distribution of internal and external sources [Rut04]. T is a compact Fredholm integral operator from some Banach space X into itself, defined by

$$(Tx)(\tau) := \int_0^{\tau^\star} g(\tau, \sigma) x(\sigma) d\sigma, \quad \tau \in [0, \tau^\star], \tag{1.2}$$

where $\tau^\star < \infty$ is the optical thickness of the atmosphere and g a weakly singular kernel. In particular, the kernel we will deal with is of the form

$g(\tau,\sigma) = k(|\tau-\sigma|)$ and is weakly singular in the following sense [AdL$^+$02]:

$$\begin{aligned}
&\lim_{\tau\to\sigma} k(|\tau-\sigma|) = +\infty;\\
&k \in C^0\left(]0,\tau^\star]\right) \cap L^1([0,\tau^\star]);\\
&\sup_{\tau\in[0,\tau^\star]} \int_0^{\tau^\star} k(|\tau-\sigma|)d\sigma < +\infty;\\
&k(|\tau-\sigma|) > 0 \text{ for all } \tau,\sigma \in]0,\tau^\star];\\
&k \text{ is a decreasing function on }]0,\tau^\star].
\end{aligned}$$

We remark that for simplification, in what follows we will use the notation $g(\tau,\sigma)$ for the operator's kernel, and the one we will work with is defined as

$$g(\tau,\sigma) := \frac{\varpi}{2} E_1\left(|\tau-\sigma|\right), \tag{1.3}$$

where $\varpi \in]0,1[$ is the albedo, assumed as a constant. The kernel depends on E_1, the first of a family of functions E_ν, the exponential-integral functions [AS60], defined by

$$E_\nu(\tau) := \int_1^\infty \frac{\exp(-\tau\mu)}{\mu^\nu} d\mu, \quad \tau > 0, \nu \geq 0, \tag{1.4}$$

and $X := L^1([0,\tau^\star])$.

The application of the eigenvalues in physics is under study, since the absence of internal and external sources leads to the nullity of the radiation field ($f = 0$ results in $x = 0$ if $\tau^\star < \infty$, [I.W60]). The numerical application of the eigenvalues is present in the approximated sum of the Neumann series that produces the solution x of (1.1). The largest eigenvalue in magnitude, λ_1, informs us about the speed of convergence of that series and the next greatest eigenvalue, λ_2, about how quickly, starting in a certain term, can be held the replacement of part of the series terms by those of a geometric series of ratio $\varpi\lambda_1$; these ideas can be found in [dH80].

We consider the problem of finding $(\lambda, x) \in \mathbb{C} \times X$, such that

$$Tx = \lambda x \neq 0. \tag{1.5}$$

Solving integral equations is of extreme importance, as they are recursively used to model some physical problems. In order to obtain approximate solutions to these kind of problems, the integral operator T is replaced by a finite rank operator T_n. So, in our case, the eigenvalue problem for the integral operator is replaced by a matrix eigenvalue problem, which can be solved computationally.

Popular schemes have been around for some time and are used to solve integral equations, for instance, Nyström method or projection methods as Galerkin, Sloan or Kantorovich methods.

Before focusing in our problem, just a brief glimpse to these methods.

In the Nyström method, the integral of the operator is replaced by a numerical quadrature formula; this method is, in principle, applied for solving equations of the second kind, see [Kre99]. Projection methods extend the previous method in the sense that the problem may be set in a wide class of Banach spaces such as L^p spaces ($1 \leq p \leq +\infty$), while Nyström method context is at least that of continuous functions. As examples we have:

- Galerkin Method: $T_n = \pi_n T \pi_n$
- Kantorovich Method: $T_n = \pi_n T$
- Sloan Method: $T_n = T \pi_n$

where $\pi_n : X \to X$ is a projection operator with finite dimensional range $X_n \subset X$ and, for each $x \in X$, as $n \to \infty$

$$\pi_n x \to x.$$

A more detailed explanation of the methods, their convergence and error analysis, may be found e.g. in [Atk97, Kre99].

In this work, we apply a projection method, on a finite dimensional subspace $X_n := \text{Span}\{e_{n,j} : j = 1, ..., n\}$ built as follows: let $(\tau_{n,j})_{j=0}^{n}$ on $[0, \tau^\star]$

be a family of grids, and $e_{n,j}(\tau) := 1$ if $\tau_{n,j-1} < \tau \leq \tau_{n,j}$, and $e_{n,j}(\tau) := 0$ otherwise, and for $x \in X$ define the linear forms

$$\langle x, e^*_{n,j} \rangle := \frac{1}{\tau_{n,j} - \tau_{n,j-1}} \int_{\tau_{n,j-1}}^{\tau_{n,j}} x(\sigma) d\sigma.$$

The bounded $n-$rank projection π_n onto the subspace X_n is defined by

$$\pi_n x := \sum_{j=1}^{n} \langle x, e^*_{n,j} \rangle e_{n,j},$$

and, using the Kantorovich method, the finite rank approximation T_n of T is defined by

$$T_n x = \pi_n T x = \sum_{j=1}^{n} \langle Tx, e^*_{n,j} \rangle e_{n,j}.$$

The spectral problem for the finite rank operator T_n can be solved through an auxiliary $n \times n$ matrix eigenvalue problem

$$A_n x_n = \lambda_n x_n \neq 0, \tag{1.6}$$

where $A_n(i,j) := \langle T e_{n,j}, e^*_{n,i} \rangle$, see [AdLV06], and (λ_n, x_n) is an approximation of (λ, x) for some x properly normalized.

The entries of this matrix are computed explicitly using (1.7), leading to a dense storage, and for large n the matrix's generation has a high computational cost.

$$\begin{aligned} A_n(i,j) =& \frac{\varpi}{2(\tau_{n,i} - \tau_{n,i-1})} \int_{\tau_{n,i-1}}^{\tau_{n,i}} \int_{\tau_{n,j-1}}^{\tau_{n,j}} E_1\left(|\tau - \sigma|\right) d\sigma d\tau \\ =& \begin{cases} \frac{\varpi}{2(\tau_{n,i} - \tau_{n,i-1})} (-E_3(|\tau_{n,i} - \tau_{n,j}|) + E_3(|\tau_{n,i-1} - \tau_{n,j}|) + & \\ \quad + E_3(|\tau_{n,i} - \tau_{n,j-1}|) - E_3(|\tau_{n,i-1} - \tau_{n,j-1}|)), & i \neq j \\ \varpi[1 + \frac{1}{\tau_{n,i} - \tau_{n,i-1}} (E_3\left(\tau_{n,i} - \tau_{n,i-1}\right) - \frac{1}{2})], & i = j \end{cases} \end{aligned} \tag{1.7}$$

For each (i,j), four evaluations of the function E_3 are required. There is a clear decay in magnitude away from the diagonal, depending on $\tau^\star$ and on n: for constant values of the former, smaller values of the latter imply faster

decay from the diagonal. The idea of zeroing out all entries with magnitude less than a certain tolerance to avoid working with dense matrix storage, and a theoretical treatment validating this approach for a required precision, was done respectively in [dTV05] and [Tit04]. Nevertheless, this strategy requires the computation of every matrix entry in order to evaluate its magnitude, since an a priori determination of the maximum bandwidth is not possible.

Besides the high generation cost, the main drawback of dense (or banded) storage is that operations such as matrix-vector product are expensive. Data-sparse representation, such as the general scheme of $\mathcal{H}$-matrices, may be useful to tackle this kind of problems.

1.2 Eigensolvers

In this work we are concerned with the computation of a few eigenvalues and eigenvectors of matrix A_n, that is, to obtain a partial solution of (1.6), by means of iterative eigensolvers. In the following subsections, we describe the methods very briefly, focusing on the required matrix operations that must be available in the implementation of $\mathcal{H}$-matrices.

1.2.1 Iterative eigensolvers

There exist a lot of iterative methods for the partial solution of eigenvalue problems, that is, for computing a subset of the eigenvalues. A detailed discussion can be found in [BDD+00]. Here we restrict our discussion to two families of methods, namely Krylov methods (e.g. Lanczos, Arnoldi or Krylov-Shur methods) and Davidson methods (e.g. Generalized Davidson or Jacobi-Davidson methods).

Given the eigenproblem

$$Ax = \lambda x \neq 0, \tag{1.8}$$

which has n eigenvalues λ counting multiplicities, the goal is to find a subset of the eigenvalues in a given region of the spectrum (for the moment, we

consider the simplest case where we seek the largest magnitude eigenvalues). Iterative eigensolvers are based on iteratively improving a subspace $\mathcal{V}$ in such a way that it eventually contains a good approximation of the eigenspace associated to the wanted eigenvalues.

Let $V \in \mathbb{R}^{n\times m}$ be a basis of $\mathcal{V}$ such that $V^TV = I$. Then the Rayleigh-Ritz projection method computes $H = V^TAV$ and uses its eigendecomposition $HY = Y\Theta$ to obtain approximate eigenpairs $(\theta_i, x_i = Vy_i)$ of A. A more sophisticated alternative is the harmonic Rayleigh-Ritz method [BDD+00], that may provide better approximations in the case of interior eigenvalues.

Krylov methods use so-called Krylov subspaces associated with matrix A and a given initial vector v_1,

$$\mathcal{K}_m(A, v_1) = \text{span}\{v_1, Av_1, A^2v_1, \ldots, A^{m-1}v_1\} \ , \tag{1.9}$$

where without loss of generality we assume that v_1 has unit length and is the first column of V.

The method of Arnoldi is an elegant algorithm that computes an orthonormal basis of the Krylov subspace and at the same time computes the projected matrix H, all this in an efficient and numerically stable way. In brief, the Arnoldi algorithm computes the m columns of V sequentially $V_m = [v_1\ v_2\ ...\ v_m]$, where column v_{j+1} is the result of orthogonalizing Av_j with respect to previous columns, and normalizing. The orthogonalization is carried out by means of a Gram-Schmidt procedure or other with better numerical properties, that removes all the components in the directions of $v_1, \ldots, v_j$. The computed quantities satisfy a relation of the form

$$AV_m = V_mH_m + \beta v_{m+1}e_m^T, \tag{1.10}$$

where H_m is an upper Hessenberg matrix, i.e., $h_{ij} = 0$ for $i > j + 1$. The last term of the Arnoldi relation is the residual and gives an indication of how close is $\mathcal{K}_m(A, v_1)$ to an invariant subspace. In particular, β is used to assess the accuracy of the computed Ritz pairs. See [BDD+00] for additional details.

The Lanczos method is related to the Arnoldi method in the sense that Lanczos can be seen as a particular case of Arnoldi when the matrix is symmetric. In this case, the projected matrix is tridiagonal. For more detail see e.g. [BDD+00, vdV02, Par98].

With Arnoldi, Ritz pairs will converge very fast provided that the initial vector v_1 is rich in the direction of the wanted eigenvectors. However, this is usually not the case and consequently many iterations will be required, but this cannot be allowed in a practical implementation in order to keep the storage requirements and the computational cost per iteration bounded. A workaround is to do a *restart* of the algorithm, that is, stop after m iterations and rerun the algorithm with a new v_1 computed from the recently obtained spectral approximations. An added benefit of this strategy is that it can be useful for driving convergence of the eigensolver towards a part of the spectrum different from the one targeted naturally by the method.

A very effective and elegant restart mechanism is the Krylov-Schur method [Ste01]. It is defined by generalizing the Arnoldi decomposition of order m, (1.10), to a Krylov decomposition of order m,

$$AV_m = V_m B_m + v_{m+1} b_{m+1}^T, \tag{1.11}$$

in which matrix B_m is not restricted to be upper Hessenberg and b_{m+1} is an arbitrary vector. Krylov decompositions are invariant under orthogonal similarity transformations, so that

$$AV_m Q = V_m Q(Q^T B_m Q) + v_{m+1} b_{m+1}^T Q, \tag{1.12}$$

where $Q^T Q = I$, is also a Krylov decomposition. In particular, one can choose Q in such a way that $S_m = Q^T B_m Q$ is in (real) Schur form, that is, upper (quasi-)triangular with the eigenvalues in the 1×1 or 2×2 diagonal blocks. This particular class of relation, called Krylov-Schur decomposition, can be written in block form as

$$A \begin{bmatrix} \tilde{V}_1 & \tilde{V}_2 \end{bmatrix} = \begin{bmatrix} \tilde{V}_1 & \tilde{V}_2 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix} + v_{m+1} \begin{bmatrix} \tilde{b}_1^T & \tilde{b}_2^T \end{bmatrix}, \tag{1.13}$$

and has the nice feature that it can be truncated, resulting in a smaller Krylov-Schur decomposition,

$$A\tilde{V}_1 = \tilde{V}_1 S_{11} + v_{m+1}\tilde{b}_1^T, \tag{1.14}$$

that can be extended again to order m. The crux of the Krylov-Schur eigensolver is to carry out this truncation-extension process repeatedly, always keeping the wanted eigenvalues in the leading principal submatrix S_{11}. Therefore, the strategy that is used for sorting the eigenvalues of S_m will have an impact on which part of the spectrum will be approximated by the Krylov-Schur eigensolver.

So far, the only operation required with matrix A is the matrix-vector product, which can be carried out very efficiently in the $\mathcal{H}$-matrix representation.

1.2.2 Computation of eigenvalues around a given target

The Krylov-Schur method could in principle be used to compute any part of the spectrum, by keeping the wanted eigenvalues in the truncated factorization. Discarding the rest of the factorization has the effect of filtering out the information associated to the unwanted eigenvectors. However, when computing eigenvalues in the interior of the spectrum, this filter is not powerful enough, and components associated to extreme eigenvalues keep on appearing, thus hindering convergence to the wanted ones.

The simplest solution to compute eigenvalues closest to a given target, σ, is to use a spectral transformation, in such a way that eigenvalues are mapped to a different position while eigenvectors remain unchanged. One such transformation is the shift-and-invert technique, that solves the problem

$$(A - \sigma I)^{-1}x = \theta x, \tag{1.15}$$

where the transformed eigenvalues satisfy the simple relation $\theta = (\lambda - \sigma)^{-1}$. In this transformation, the eigenvalues θ of the operator that are largest in

magnitude correspond, in the original problem, to the eigenvalues λ that are close, in absolute values, to σ, see Figure 1.1. As eigenvalues λ closest to the target become dominant in the transformed spectrum, so Krylov methods will have a fast convergence. This can be implemented by simply replacing the action of A by the action of $(A-\sigma I)^{-1}$ in the Krylov subspace expansion, that is, by solving linear systems with coefficient matrix $A - \sigma I$ whenever a matrix-vector product is required. These linear systems must be solved very accurately, since Krylov methods can be very sensitive to numerical errors introduced in the computation of the Krylov subspace, so in most applications a direct linear solver will be required, rather than an iterative method. It is generally claimed that the main drawback of the shift-and-invert technique is the high cost associated to direct linear solvers, since the memory requirements and computational effort can be very high for large, sparse matrices. In the case of the $\mathcal{H}$-matrix representation, this downside disappears because computing the factorization has much smaller cost, both in terms of storage and operations, as well as the corresponding triangular solves.

An alternative to the spectral transformation is the use of a preconditioned eigensolver such as Jacobi-Davidson. These methods expand the subspace in a different way, attempting to make the whole computation more robust with respect to numerical error in the application of the operator. This allows to use iterative linear solvers such as GMRES (Generalized Minimal Residual) in the so-called correction equation.

Jacobi-Davidson method for linear problems was proposed by Sleijpen and van der Vorst [SVdV96] combining the Davidson method to expand the subspace, in which eigenvector approximations are constructed, with the Jacobi's idea, of looking for the orthogonal complement of a given eigenvector approximation. The method has been further developed and adapted to generalized eigenproblems [SBFVdV96]. This combined use is fundamental to enhance Davidson's idea. Whereas in Davidson's method accurate preconditioners lead to slow convergence (or even to stagnation), the Jacobi-Davidson

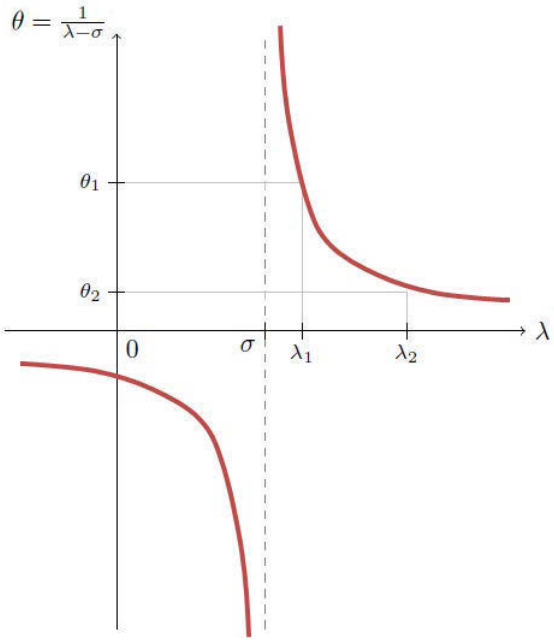

Figure 1.1: Shift-and-invert transformation

method profits from such good preconditioners. Excellent references for the Jacobi-Davidson method are [SVdV00, BDD+00].

This method was motivated by the fact that standard iterative eigensolvers often require an expensive factorization of the matrix when interior eigenvalues are desired (e.g., shift-and-invert Arnoldi with a direct linear solver). Jacobi-Davidson tries to reduce the cost by solving approximately linear systems, generally using iterative methods, without affecting the robustness.

This method usually is aiming at a particular eigenvalue, but if one is interested in more than one (near a specified target), a scheme presented in [FSVdV98] can be used.

Nevertheless, this does not seem to be the best approach in the context of $\mathcal{H}$-matrix representation, in view of the efficiency of matrix factorization. From a practical perspective, to implement this kind of methods it is required to be able to build a preconditioner for matrix A.

1.3 $\mathcal{H}$-matrices

Special mathematical and numerical treatments are required to find a suitable representation of the operator and to improve performance. As previously pointed out, $\mathcal{H}$-matrices are an excellent way to tackle these requirements and overcome the difficulties.

Let $\mathcal{I} = \{1, ..., n\}$ denote the set of the indices of the basis functions $e_{n,i}$, and t and s two subsets of $\mathcal{I}$ where $\alpha = \bigcup_{i \in t} \text{supp } e_{n,i}$ and $\beta = \bigcup_{j \in s} \text{supp } e_{n,j}$ are the corresponding domains.

The notions of a *cluster tree* and of a *block cluster tree* are important components in the construction of $\mathcal{H}$-matrices. While the first describes a hierarchical partitioning over the index $\mathcal{I}$ giving us the candidates for checking an admissibility condition for low-rank approximation, the second contains the $\mathcal{H}$-matrix's structure.

A cluster tree corresponding to the index set $\mathcal{I}$, $T_{\mathcal{I}}$, satisfies the following properties, [BGH03b]:

- Each node of $T_{\mathcal{I}}$ is a subset of $\mathcal{I}$.
- The root of $T_{\mathcal{I}}$ is the index set $\mathcal{I}$.
- A leaf consists of a minor number of indices. A leaf is not more subdivided and this happens when the cardinality of a node is less than or equal to a certain threshold defined *a priori*.
- A node that is not a leaf is subdivided into two sons and is equal to their disjoint union.

The representation of $\mathcal{H}$-matrices uses a tree structure, named block cluster tree and represented by $T_{\mathcal{I}\times\mathcal{I}}$. It describes a hierarchical block partitioning of a matrix. Its root represents the entire matrix and the inner nodes are the matrix sub-blocks that are being partitioned possibly more on the succeeding level, i.e., these blocks can be further subdivisible or not. In the last

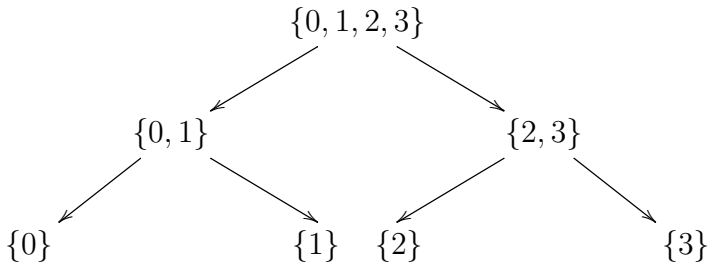

Figure 1.2: Example of a $T_{\mathcal{I}}$.

case, the blocks, named leaves, are represented by either low-rank matrices or full matrices.

In order to ilustrate better these two structures, Figure 1.2 gives an example of a cluster tree $T_{\mathcal{I}}$ with $\mathcal{I} = \{0, 1, 2, 3\}$ and, based upon $T_{\mathcal{I}}$ of this figure, a block cluster tree is constructed in Figure 1.3.

Starting with $\mathcal{I} \times \mathcal{I}$, the root of the so-called cluster tree $T_{\mathcal{I}\times\mathcal{I}}$, a splitting process begins through which each block is subdivided into four successors, until either an admissibility condition is satisfied or the block is already sufficiently small to be still subdivided. This last situation occurs when the refinement has arrived to the leaves, which possesses a certain minimal amount of elements defined *a priori*, as was previoulsy described.

By means of the following admissibility condition, which comes from error bounding reasons as will become apparent later in this thesis, we test if the domain $\alpha \times \beta$ is admissible

$$\operatorname{diam}(\alpha) < \eta \operatorname{dist}(\alpha, \beta), \tag{1.16}$$

for $\eta > 0$ fixed, that is, if the corresponding block of indices $t \times s$ is admissible. Let $M|_{t\times s}$ be the corresponding submatrix. It will be approximated by a low-rank matrix $\tilde{M}|_{t\times s} = AB^T$, where $\tilde{M}|_{t\times s} \in \mathbb{R}^{t\times s}$, $A \in \mathbb{R}^{t\times k}$, $B \in \mathbb{R}^{s\times k}$ and $\operatorname{rank}(\tilde{M}|_{t\times s}) \leq k$. The blocks that do not fulfill the condition (1.16) are said to be inadmissible, stored in the standard way and computed by (1.7). The definition of an $\mathcal{H}$-matrix comes out naturally from what was just described; it can be seen e.g. in [Bor05, GKLB09]:

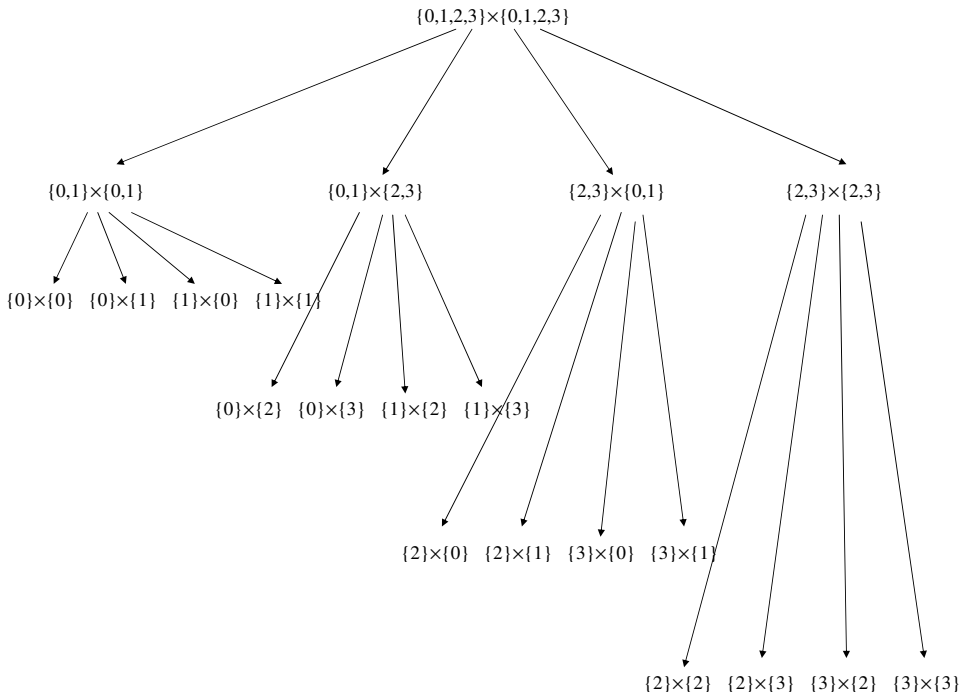

Figure 1.3: Example of a $T_{\mathcal{I}\times\mathcal{I}}$ based upon $T_{\mathcal{I}}$ of Figure 1.2.

Definition 1. *Let k, $n_{min} \in \mathbb{N}_0$. The $\mathcal{H}$-matrices' set, induced by a block cluster tree $T_{\mathcal{I}\times\mathcal{I}}$ with blockwise rank k and minimum block size n_{min} is defined by*

$$\mathcal{H} = \big\{ M \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}} | \textit{for all } t \times s \textit{ belonging to the leaves' set} : \textit{rank}\,(M|_{t\times s}) \leq k \\ \textit{or } \min\{\#t, \#s\} \leq n_{min} \big\}$$

A matrix $M \in \mathcal{H}$ is said to be given in $\mathcal{H}$-matrix representation if the blocks $M|_{t\times s}$ with $\textit{rank}\,(M|_{t\times s}) \leq k$ are stored in a low-rank representation, whereas the remaining blocks are stored as full matrices.

Following the previous figures of the cluster tree and block cluster tree with $\mathcal{I} = \{0, 1, 2, 3\}$, Figure 1.4 appears to illustrate the respective $\mathcal{H}$-matrix structure, where the red blocks correspond to the inadmissible leaves, while the green ones correspond to the admissible blocks.

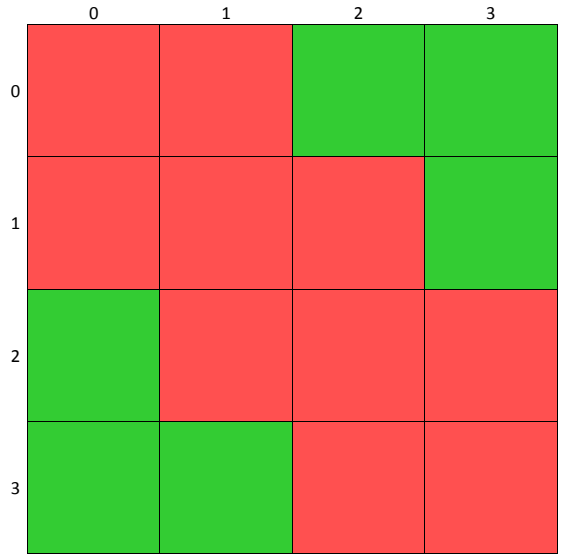

Figure 1.4: Example of an $\mathcal{H}$-matrix structure based upon $T_{\mathcal{I}\times\mathcal{I}}$ of Figure 1.3.

If the rank k is smaller than t and s, corresponding to the matrix M size, considerable savings in the storage and work complexity of a low-rank matrix compared to a full matrix are obtained [GH03].

For the low-rank approximation, the kernel of the integral operator is replaced by a degenerate approximation $\tilde{g}(\tau, \sigma)$, such that the integration with respect to the diferent variables is segregated,

$$\tilde{g}(\tau, \sigma) := \sum_{\rho=0}^{k-1} q_\rho(\tau) p_\rho(\sigma). \tag{1.17}$$

Furthermore, the approximantion $\tilde{g}$ has, naturally, to converge fast to the kernel function g. In this work, the process of building the cluster tree is

undertaken by HLIB [BGH03b, BGH03a] and will be detailed in Subsection 2.1 of this thesis.

The matrix entries $M_{ij} = \frac{1}{h_{n,i}} \int_0^{\tau^\star} \int_0^{\tau^\star} e_{n,i}(\tau) g(\tau,\sigma)\, e_{n,j}(\sigma) d\sigma d\tau$ for $(i,j) \in t \times s$ are then approximately given by

$$\begin{aligned}\left(\tilde{M}|_{t\times s}\right)_{ij} &= \frac{1}{h_{n,i}} \int_0^{\tau^\star} \int_0^{\tau^\star} e_{n,i}(\tau) \tilde{g}(\tau,\sigma)\, e_{n,j}(\sigma) d\sigma d\tau \\ &= \frac{1}{h_{n,i}} \int_0^{\tau^\star} \int_0^{\tau^\star} e_{n,i}(\tau) \sum_{\rho=0}^{k-1} q_\rho(\tau) p_\rho(\sigma) e_{n,j}(\sigma) d\sigma d\tau \\ &= \sum_{\rho=0}^{k-1} \frac{1}{h_{n,i}} \int_0^{\tau^\star} e_{n,i}(\tau) q_\rho(\tau) d\tau \int_0^{\tau^\star} e_{n,j}(\sigma) p_\rho(\sigma) d\sigma, \end{aligned} \tag{1.18}$$

where $h_{n,i} = \tau_{n,i} - \tau_{n,i-1}$.

We remark the importance of using (1.17) regarding the separation of the variables τ and σ, since it allows in (1.18) the possibility to write the double integral as a product of two single integrals, which are the entries of the matrices A and B of the factorized submatrix with rank at most k. So, the entries of those two matrices are defined by

$$\begin{aligned} A_{i\rho} &:= \frac{1}{h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} q_\rho(\tau) d\tau, \ A = (A_{i\rho}) \in \mathbb{R}^{t\times\{0,\ldots,k-1\}} \\ &\text{and} \\ B_{j\rho} &:= \int_{\tau_{n,j-1}}^{\tau_{n,j}} p_\rho(\sigma) d\sigma, \quad B = (B_{j\rho}) \in \mathbb{R}^{s\times\{0,\ldots,k-1\}}. \end{aligned} \tag{1.19}$$

Chapter 2

Numerical libraries

This thesis makes use of a considerable set of software libraries which includes the most recent contributions of software developments based on novel mathematical approaches to numerical computation of eigenvalue problems. These libraries are tuned to explore the modern architecture of today's computers. So, in this chapter we give a short presentation of each one of the main libraries used.

2.1 HLIB library

HLIB [BGH03b, BGH03a] is a library for hierarchical matrices written in C programming language using BLAS and LAPACK libraries to perform lower-level algebraic operations (e.g. dense matrix-matrix multiplication). It includes functions for $\mathcal{H}$ matrix arithmetics, the treatment of partial differential equations and a number of integral operators and in addition support routines for the creation of cluster trees, visualization and numerical quadrature.

In the implementation of an $\mathcal{H}$-matrix, called *supermatrix*, `rkmatrices` are a straightforward representation of low-rank submatrices while `fullmatrices` represent the submatrices corresponding to inadmissible leaves (usually dense matrices although not having to bear any specific structure). This implemen-

tation may be found e.g. in [BGH03a], and it is done in such a way that its structure is similar to the block cluster tree's structure.

This library provides routines for building the structure of an hierarchical matrix, that is, of cluster trees, block cluster trees, low-rank matrices and block matrices; discretization functions that fill these structures by approximations of operators, arithmetic algorithms that produce approximative matrix operations (e.g. addition, multiplication, vector-matrix multiplication, factorizations, inversion); conversion routines that turn sparse and dense matrices into $\mathcal{H}$-matrices; service functions matrix structures, for instance to plot or to handle files. Moreover, based upon basic linear algebra subroutines, it also provides algorithms to compute LU-decompostion and Cholesky decomposition.

So, HLIB provide algorithms that perform matrix operations in the hierarchical matrix format efficiently and the actual proof for the efficiency, namely the complexity estimates, can be found, besides [BGH03a], in [GH03] where the authors analyse the complexity (storage, addition, multiplication and inversion) of the $\mathcal{H}$-matrix arithmetics.

In this work, the process of building the cluster tree is undertaken by HLIB, which, as already mentioned, enables matrix operations of almost linear complexity, being therefore particularly adequate for large dimensional problems. Alternatively, AHMED (Another software library on Hierarchical Matrices for Elliptic Differential equations) could be used. A complete reference for $\mathcal{H}$-matrices as well as for this library is [Beb08]. With the $\mathcal{H}$-matrix representation, it is possible to realize common computations with linear-polylogarithmic complexity rather than quadratic or cubic cost. For instance, the multiplication of an $n \times n$ $\mathcal{H}$-matrix by a vector requires about $4n \log_2 n$ floating-point operations, the (approximate) LU decomposition about $6n \log_2 2n$ operations, and $2n \log_2 n$ operations for the back solves. See [Hac99, HK00] for additional details on $\mathcal{H}$-matrix arithmetic.

2.2 PETSc and SLEPc libraries

Working with sparse representation is not standard, as it happens in dense representation of an operator matrix. This comes from the fact that sparse storage is more complicated, as there may be more variations, so less standardized, and in this way more complex to work.

SLEPc, the Scalable Library for Eigenvalue Problem Computations [HRV05, HRTV10], is a software package for the solution of large-scale eigenvalue problems on parallel computers. It can be used to solve standard and generalized eigenvalue problems, as well as other types of related problems such as the quadratic eigenvalue problem or the singular value decomposition. SLEPc can work with either real or complex arithmetic, in single or double precision, and it is not restricted to symmetric (hermitian) problems. It can be used from code written in C, C++, and FORTRAN. Most of the methods in this library are projection methods, including different variants of Krylov and Davidson iterations.

SLEPc is able to cope with different problem types such as hermitian, non-hermitian (the default case), Generalized hermitian, Generalized non-hermitian, and Generalized non-hermitian with positive (semi-)definite B (a generalized eigenproblem is commonly written as $Ax = \lambda Bx$). The library is very flexible and it is possible to specify how many eigenvalues/eigenvectors to compute. In relation to the eigenvalues of interest, it is possible, in real symmetric problems, to compute the largest or smallest eigenvalues in magnitude, the leftmost or rightmost ones and even those closest to a given target value. For the complex case, more options are available.

SLEPc provides a collection of eigensolvers: Power Iteration with deflation, Subspace Iteration with Rayleigh-Ritz projection, Arnoldi method, Lanczos method, Krylov-Schur and Davidson-type solvers. Most of the eigensolvers are based on the subspace projection paradigm, in particular, it includes a robust and efficient parallel implementation of Krylov-Schur method described in Section 1.2, which is the default solver in SLEPc. Sev-

eral Davidson-type solvers are included as well, in particular Generalized Davidson and Jacobi-Davidson, with various possibilities for the computation of the correction vector. In these solvers, the user can easily select which preconditioner to use.

Apart from eigensolvers, some spectral transformations such as the shift-and-invert technique of (1.15) are available, where the user can compute interior eigenvalues with the help of linear solvers included in PETSc (Portable, Extensible Toolkit for Scientific Computation).

An error bound for the computed solution in hermitian problems is available in literature,

$$\left|\lambda - \tilde{\lambda}\right| \leq \|r\|_2 \tag{2.1}$$

where $r = A\tilde{x} - \tilde{\lambda}\tilde{x}$ is the residual vector, being $\left(\tilde{\lambda}, \tilde{x}\right)$ the computed eigenpair and λ the exact eigenvalue. For the non-hermitian case, such simple relation as (2.1) is not available.

SLEPc is built on top of PETSc [BBE+10], a parallel framework for the numerical solution of partial differential equations, whose approach is to encapsulate mathematical algorithms using object-oriented programming techniques in order to be able to manage the complexity of efficient numerical message-passing codes. It uses primarily the basic data structures such as those for representing vectors and matrices.

PETSc is object-oriented in the sense that all the code is built around a set of data structures and algorithmic objects. The application programmer works directly with these objects rather than concentrating on the underlying data structures. The three basic abstract data objects are *index sets*, *vectors* and *matrices*. Built on top of this foundation are various classes of solver objects, including linear, non-linear and time-stepping solvers.

SLEPc inherits all the good properties of PETSc, including portability to a wide range of parallel platforms, scalability to a large number of processors, and run-time flexibility giving full control over the solution process (one can for instance specify the solver at run time, or change relevant parameters

such as the tolerance or the size of the subspace basis). Also, since PETSc provides a uniform interface with all of its linear solvers and a large family of preconditioners, it is possible to compare diverse combinations of method and preconditioner, just by their specification at execution time.

The solvers in PETSc (and SLEPc) have a data-structure neutral implementation. This means that the computation can be done with different matrix storage formats, and also even with a matrix that is not stored explicitly. By default, a matrix in PETSc is stored in a parallel compressed-row sparse format (called `aij`), where each processor stores a subset of rows. Other formats include the symmetric variant (`sbaij`), where only the upper triangular part is stored, as well as the dense storage (both sequential and parallel).

For implementing a matrix-free solver with so-called *shell* matrices, the application programmer has to create one of such matrices and define its operations, by binding a user-defined subroutine for each operation. Only the operations required by the actual computation need to be set, so in the simplest case it is sufficient to implement the matrix-vector product. For more advanced functionality, *e.g.*, preconditioning, other operations are required as well. We use this feature to interface our code to HLIB.

2.3 BLAS and LAPACK libraries

Among the main numerical linear algebra packages, is certainly the LAPACK and BLAS libraries.

LAPACK (Linear Algebra PACKage) [ABB99] is a library with subroutines for solving the most usual problems appearing in numerical linear algebra, having been conceived to be efficient on various modern high-performance computers. It provides the following routines:

- Linear equations,
- Linear least squares problems,

- Eigenvalue problems,
- Singular value decomposition.

LAPACK can also manage many associated computations such as matrix factorizations or estimating condition numbers.

This library contains: *driver routines* for solving standard types of problems, each solving a complete problem, e.g., solving a system of linear equations; *computational routines* each performing a different computational task, e.g. LU factorization; and *auxiliary routines* to perform a certain subtask or common low-level computation, e.g. computing a matrix-norm. Note that each driver routine calls a sequence of computational routines.

LAPACK routines are design so that the computation is performed by calls, as much as possible, to the Basic Linear Algebra Subprograms (BLAS).

BLAS is a library that provides standard building blocks for performing basic vector and matrix operations. The Level 1 BLAS perform scalar, vector and vector-vector operations, the Level 2 BLAS perform matrix-vector operations, and the Level 3 BLAS perform matrix-matrix operations. A large set of research works, as well as industrial ones, rely on top of this library. The first published paper on the library was [LHKK79] and, maybe, one of the most important references is [DHP02].

Highly efficient machine-specific implementations of the BLAS are available for many modern high-performance computers. The BLAS, being efficient, portable, and widely available, enable LAPACK routines to achieve high performance with portable code, allowing this to be high quality linear algebra software.

2.4 OpenMP library

The OpenMP standard was formulated in 1997 as an Application Program Interface (API) for writing portable, multithreaded applications, shared memory parallelism (note that a shared memory process consists of multiple

threads) [CJVDP07]. It is a specification for a set of *compiler directives*, *runtime library routines*, and *environment variables* that can be used to specify shared memory parallelism in Fortran, C and C++ programs, on a wide variety of architectures.

OpenMP provides a portable (most major platforms have been implemented including Unix/Linux platforms and Windows NT), scalable model for developers of shared memory parallel applications, in which the most of its major features includes its various constructs and directives for specifying parallel regions, work sharing, synchronization and data environment.

This library is the result of efforts for some standardization, providing a standard among a variety of shared memory architectures/platforms, seting up a simple and limited set of directives for programming shared memory machines, be ease to use, have capability to incrementally parallelize a serial program as well as capability to implement both coarse-grain and fine-grain parallelism, and portability.

The OpenMP is based on the existence of multiple threads in the shared memory programming paradigm, being an explicit (i.e. not automatic) programming model allowing the programmer full control over parallelization.

An OpenMP application begins with a single thread (master thread). As the program executes, the application may be faced with parallel regions in which the master thread gives rise to thread teams, including the master thread. At the end of a parallel region, the thread teams are "left" and the master thread continues execution. Inside a parallel region there can be nested parallel regions where each thread of the original parallel region becomes the master of its own thread team. Nested parallelism can continue to additionally nest other parallel regions.

Moreover, OpenMP provides the distinction between data that is shared from data that is private, an important issue in parallel programming (performance). In this context, shared variables are shared by all the threads from the thread team (e.g. a change of the shared variable in one thread may become visible to another thread in the parallel region), while in private

variables this doesn't occur, having private copies made for each thread in the thread team, and in this way, changes made in one thread are not visible in the private variables in the other threads.

Finally, OpenMP provides multiple types of synchronization to help in many different situations (synchronization when multiple threads are running at the same time and synchroniztion between them is needed).

Part II

Development of numerical approximations

Chapter 3

Low-rank approximations

Previously, in (1.7), it was given a dense representation for the matrix A_n. In this chapter, we discuss several strategies for representing A_n by means of data-sparse representation, which can be implemented in various flavours.

Degenerate approximations, generally given by (1.17), can be built in different ways, depending on the properties of the kernel. For instance, they can be based on polynomial interpolation or Taylor series expansion. Another approach consists in computing a low-rank approximation from an explicitly built matrix block by means of a singular value decomposition. We next describe these three approaches in turn.

3.1 Taylor approximation

Using truncated Taylor series, the degenerate kernel expression (1.17) is specifically defined by $q_\rho(\tau) = (\tau - \tau_0)^\rho$ and $p_\rho(\sigma) = \frac{1}{\rho!}\partial_\tau^\rho g(\tau_0, \sigma)$, where τ_0 is taken as the midpoint of α.

The entries of the matrices A and B in (1.19) are analytically determined explicitly as follows.

Proposition 2. *The elements of the matrix A are given by*

$$A_{i\rho} = \frac{1}{h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} q_\rho(\tau) d\tau = \frac{1}{h_{n,i}} \left(\frac{(\tau_{n,i} - \tau_0)^{\rho+1}}{\rho+1} - \frac{(\tau_{n,i-1} - \tau_0)^{\rho+1}}{\rho+1} \right). \quad (3.1)$$

Proof. Immediate. □

In order to get the expression for $\tilde{g}$ and for the entries $B_{j\rho}$, a preceding step must be made, to derive the analytical expression for $\partial_\tau^\rho g(\tau_0, \sigma)$.

Lemma 3. *The partial derivatives with respect to the variable τ of the kernel g defined in* (1.3)*, are given by*

$$\partial_\tau^\rho g(\tau_0,\sigma) = \begin{cases} -\frac{\varpi}{2}\sum_{k=1}^{\rho}(-1)^{\rho-1}\frac{(\rho-1)!}{(\rho-k)!}\frac{e^{-\tau_0+\sigma}}{(\tau_0-\sigma)^k}, & \sigma<\tau_0 \\ -\frac{\varpi}{2}\sum_{k=1}^{\rho}(-1)^{k-1}\frac{(\rho-1)!}{(\rho-k)!}\frac{e^{\tau_0-\sigma}}{(\tau_0-\sigma)^k}, & \sigma>\tau_0 \end{cases}.$$

Proof. Considering the following results present in [AS60],

$$E_1(z) = \int_1^\infty \frac{e^{-zt}}{t}dt \quad \text{and} \quad \frac{d}{dz}E_1(z) = -E_0(z) = -\frac{e^{-z}}{z},$$

the first derivative of the kernel with respect to τ is given by

$$\begin{aligned}
\partial_\tau g(\tau,\sigma) &= \frac{\varpi}{2}\partial_\tau E_1(|\tau-\sigma|) \\
&= \frac{\varpi}{2}\partial_\tau\left(E_1\circ f(\tau,\sigma)\right) \quad \text{where } f(\tau,\sigma)=|\tau-\sigma| \\
&= \frac{\varpi}{2}\left(-\frac{e^{-f(\tau,\sigma)}}{f(\tau,\sigma)}\right)\times\partial_\tau f(\tau,\sigma) \\
&= \frac{\varpi}{2}\left(-\frac{e^{-|\tau-\sigma|}}{|\tau-\sigma|}\right)\times\partial_\tau f(\tau,\sigma) \\
&= \begin{cases} -\frac{\varpi}{2}\frac{e^{-(\tau-\sigma)}}{\tau-\sigma}, & \sigma<\tau \\ -\frac{\varpi}{2}\frac{e^{-(-\tau+\sigma)}}{\tau-\sigma}, & \sigma>\tau \end{cases} \\
&= -\frac{\varpi}{2}\frac{e^{-|\tau-\sigma|}}{\tau-\sigma}, \quad \text{for } \tau\neq\sigma.
\end{aligned}$$

1. For $\sigma > \tau$, after some initial calculations, we obtain

$$\begin{aligned}
\partial_\tau^2 g(\tau,\sigma) &= -\frac{\varpi}{2}\partial_\tau^1\left(\frac{e^{\tau-\sigma}}{\tau-\sigma}\right) \\
&= -\frac{\varpi}{2}\left(\frac{e^{\tau-\sigma}(\tau-\sigma)-e^{\tau-\sigma}}{(\tau-\sigma)^2}\right) \\
&= -\frac{\varpi}{2}\left(\frac{e^{\tau-\sigma}}{\tau-\sigma}-\frac{e^{\tau-\sigma}}{(\tau-\sigma)^2}\right),
\end{aligned}$$

$$\begin{aligned}\partial_\tau^3 g(\tau,\sigma) &= -\frac{\varpi}{2}\partial_\tau^1\left(\frac{e^{\tau-\sigma}}{\tau-\sigma}-\frac{e^{\tau-\sigma}}{(\tau-\sigma)^2}\right)\\ &= -\frac{\varpi}{2}\left(\frac{e^{\tau-\sigma}}{\tau-\sigma}-\frac{e^{\tau-\sigma}}{(\tau-\sigma)^2}-\partial_\tau^1\left(\frac{e^{\tau-\sigma}}{(\tau-\sigma)^2}\right)\right)\\ &= -\frac{\varpi}{2}\left(\frac{e^{\tau-\sigma}}{\tau-\sigma}-\frac{2e^{\tau-\sigma}}{(\tau-\sigma)^2}+\frac{2e^{\tau-\sigma}}{(\tau-\sigma)^3}\right),\end{aligned}$$

and, generalizing, we reach the expression for any value of ρ

$$\partial_\tau^\rho g(\tau,\sigma) = -\frac{\varpi}{2}\sum_{k=1}^{\rho}(-1)^{k-1}\times\frac{(\rho-1)!}{(\rho-k)!}\times\frac{e^{\tau-\sigma}}{(\tau-\sigma)^k}.$$

2. For $\sigma < \tau$, analogously to the previous case, the following general recursive rule is obtained

$$\partial_\tau^\rho g(\tau,\sigma) = -\frac{\varpi}{2}\sum_{k=1}^{\rho}(-1)^{\rho-1}\frac{(\rho-1)!}{(\rho-k)!}\frac{e^{-\tau+\sigma}}{(\tau-\sigma)^k}.$$

□

Using the previous proposition, the expression for the truncated Taylor series kernel is explicitly given by

$$\begin{aligned}\tilde{g}(\tau,\sigma) &= \sum_{\rho=0}^{k-1} q_\rho(\tau)p_\rho(\sigma)\\ &= \begin{cases} -\frac{\varpi}{2}\sum_{\rho=0}^{k-1}(\tau-\tau_0)^\rho\frac{1}{\rho}\sum_{i=1}^{\rho}(-1)^{\rho-1}\frac{1}{(\rho-i)!}\frac{e^{-\tau_0+\sigma}}{(\tau_0-\sigma)^i}, & \sigma<\tau_0\\ -\frac{\varpi}{2}\sum_{\rho=0}^{k-1}(\tau-\tau_0)^\rho\frac{1}{\rho}\sum_{i=1}^{\rho}(-1)^{i-1}\frac{1}{(\rho-i)!}\frac{e^{\tau_0-\sigma}}{(\tau_0-\sigma)^i}, & \sigma>\tau_0\end{cases}. \quad (3.2)\end{aligned}$$

With the result of Lemma 3, we can readily obtain the expression for the entries of matrix B, by evaluating the derivatives at $\tau = \tau_0$.

Proposition 4. *The entries of matrix B can be computed as*

$$B_{j\rho} = \begin{cases} -\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(-1)^{\rho-1}}{\rho(\rho-k)!}\left[\Gamma\left(1-k,\tau_0-\tau_{n,j}\right)\right. & \\ \qquad\left.-\Gamma\left(1-k,\tau_0-\tau_{n,j-1}\right)\right], & \tau_{n,j-1}<\tau_0,\ \tau_{n,j}<\tau_0\\ \frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{1}{\rho(\rho-k)!}\left[\Gamma\left(1-k,-\tau_0+\tau_{n,j-1}\right)\right. & \\ \qquad\left.-\Gamma\left(1-k,-\tau_0+\tau_{n,j}\right)\right], & \tau_{n,j-1}>\tau_0,\ \tau_{n,j}>\tau_0\\ A_n(i,j)^1, & \tau_{j-1}<\tau_0<\tau_j\end{cases}$$

[1] these entries are given by (1.7)

Proof. First recall the incomplete Gamma function

$$\Gamma(a,x)=\int_x^\infty \frac{e^{-t}}{t^{1-a}}dt.$$

1. For $\tau_{n,j-1}>\tau_0,\ \tau_{n,j}>\tau_0$:

$$\begin{aligned}B_{j\rho}&=\int_{\tau_{n,j-1}}^{\tau_{n,j}}\frac{1}{\rho!}\partial_\tau^\rho g(\tau_0,\sigma)d\sigma\\&=\int_{\tau_{n,j-1}}^{\tau_{n,j}}\frac{1}{\rho!}\left(-\frac{\varpi}{2}\sum_{k=1}^{\rho}(-1)^{k-1}\frac{(\rho-1)!}{(\rho-k)!}\frac{e^{\tau_0-\sigma}}{(\tau_0-\sigma)^k}\right)d\sigma\\&=-\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(-1)^{k-1}}{\rho\,(\rho-k)!}\int_{\tau_{n,j-1}}^{\tau_{n,j}}\frac{e^{\tau_0-\sigma}}{(-1)^k\,(-\tau_0+\sigma)^k}d\sigma\end{aligned}$$

and using integration by substitution $(u=-\tau_0+\sigma)$,

$$\begin{aligned}B_{j\rho}&=-\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(-1)^{k-1}}{\rho\,(\rho-k)!}\times\frac{1}{(-1)^k}\int_{-\tau_0+\tau_{n,j-1}}^{-\tau_0+\tau_{n,j}}\frac{e^{-u}}{u^k}du\\&=\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{1}{\rho\,(\rho-k)!}\times\left(\int_{-\tau_0+\tau_{n,j-1}}^{\infty}\frac{e^{-u}}{u^k}du-\int_{-\tau_0+\tau_{n,j}}^{\infty}\frac{e^{-u}}{u^k}du\right)\\&=\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{1}{\rho\,(\rho-k)!}\times\left[\Gamma\left(1-k,-\tau_0+\tau_{n,j-1}\right)-\Gamma\left(1-k,-\tau_0+\tau_{n,j}\right)\right].\end{aligned}$$

2. For $\tau_{n,j-1}<\tau_0,\ \tau_{n,j}<\tau_0$:

$$\begin{aligned}B_{j\rho}&=\int_{\tau_{n,j-1}}^{\tau_{n,j}}\frac{1}{\rho!}\partial_\tau^\rho g(\tau_0,\sigma)d\sigma\\&=\int_{\tau_{n,j-1}}^{\tau_{n,j}}\frac{1}{\rho!}\left(-\frac{\varpi}{2}\sum_{k=1}^{\rho}(-1)^{\rho-1}\frac{(\rho-1)!}{(\rho-k)!}\frac{e^{-\tau_0+\sigma}}{(\tau_0-\sigma)^k}\right)d\sigma\\&=-\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(-1)^{\rho-1}}{\rho\,(\rho-k)!}\int_{\tau_{n,j-1}}^{\tau_{n,j}}\frac{e^{-(\tau_0-\sigma)}}{(\tau_0-\sigma)^k}d\sigma\end{aligned}$$

doing the substitution $u=\tau_0-\sigma$,

$$\begin{aligned}B_{j\rho}&=-\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(-1)^{\rho-1}}{\rho\,(\rho-k)!}\left(-\int_{\tau_0-\tau_{n,j-1}}^{\tau_0-\tau_{n,j}}\frac{e^{-u}}{u^k}du\right),\ \text{ and as } \tau_0-\tau_{n,j-1}>\tau_0-\tau_{n,j},\\&=-\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(-1)^{\rho-1}}{\rho\,(\rho-k)!}\left(\int_{\tau_0-\tau_{n,j}}^{\tau_0-\tau_{n,j-1}}\frac{e^{-u}}{u^k}du\right)\\&=-\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(-1)^{\rho-1}}{\rho\,(\rho-k)!}\left[\Gamma\left(1-k,\tau_0-\tau_{n,j}\right)-\Gamma\left(1-k,\tau_0-\tau_{n,j-1}\right)\right].\end{aligned}$$

3. It is compulsory to comment about the entries satisfying $\tau_{n,j-1} < \tau_0$ and $\tau_{n,j} > \tau_0$. Although in this situation the previous integral is divergent, those entries do not satisfy the admissibility condition, which implies that they must be computed using (1.7).

□

Proposition 4 can be written using exponential integral functions instead of the incomplete Gamma function, as it is next explained.

Proposition 5.

$$B_{j\rho} = \begin{cases} -\frac{\varpi}{2} \sum_{k=1}^{\rho} \frac{(-1)^{\rho-1}}{\rho(\rho-k)!} \Big[(\tau_0 - \tau_{n,j})^{1-k} E_k (\tau_0 - \tau_{n,j}) - \\ \qquad - (\tau_0 - \tau_{n,j-1})^{1-k} E_k (\tau_0 - \tau_{n,j-1}) \Big], & \tau_{n,j-1} < \tau_0,\ \tau_{n,j} < \tau_0 \\ \frac{\varpi}{2} \sum_{k=1}^{\rho} \frac{1}{\rho(\rho-k)!} \Big[(-\tau_0 + \tau_{n,j-1})^{1-k} E_k (-\tau_0 + \tau_{n,j-1}) - \\ \qquad - (-\tau_0 + \tau_{n,j})^{1-k} E_k (-\tau_0 + \tau_{n,j}) \Big], & \tau_{n,j-1} > \tau_0,\ \tau_{n,j} > \tau_0 \end{cases}$$

Proof. Immediate, using the next equality present in [AS60]:

$$E_n (x) := x^{n-1} \Gamma (1 - n, x).$$

□

3.2 Error bounds for Taylor approximation

In order to develop error bounding proofs, we introduce some technical constants:

Definition 6. *Let*

1. $\varkappa$ *be any constant such that*

$$\forall \tau \in \alpha, \quad |\tau - \tau_0| \leq \operatorname{diam}(\alpha) \leq \varkappa < 1. \tag{3.3}$$

We recall that the Taylor expansion is done around τ_0.

2. γ_0 *be given by*

$$\gamma_0 := \frac{\operatorname{diam}(\alpha)}{\operatorname{dist}(\alpha,\beta)} < 1. \tag{3.4}$$

We recall that

$$\forall \tau \in \alpha,\ \forall \sigma \in \beta, \quad \operatorname{dist}(\alpha,\beta) \leq |\tau - \sigma|. \tag{3.5}$$

Result (3.4) can be rewritten as $\operatorname{diam}(\alpha) < \operatorname{dist}(\alpha,\beta)$ which is, actually, the admissibility condition given in (1.16) (with $\eta = 1$). The parameter η controlls the speed of convergence, i.e., the quality of the approximation, and a usual choice is $\eta = 1$, see e.g. [BGH03b, Hac99, HKK04].

In terms of notation, consider $\epsilon_k = g(\tau,\sigma) - \tilde{g}(\tau,\sigma)$ to be the error when using $\tilde{g}$ as an approximation of g.

In the next proposition we propose an upper bound for the error when $\tilde{g}$ is taken as the truncated Taylor expansion of the kernel.

Proposition 7. *The error of using the truncated Taylor series* $\tilde{g}$*, defined by* (3.2) *as an approximation of the kernel* g*, given in* (1.3)*, can be estimated by*

$$|\epsilon_k| \leq \begin{cases} C_0 \frac{\varkappa^k}{1-\varkappa}, & |\tau_0 - \sigma| > 1 \\ C_1 \left(\frac{\varkappa^{k+1}}{(1-\varkappa^k)^2} + \frac{k\varkappa^k}{1-\varkappa^k} \right), & |\tau_0 - \sigma| = 1 \\ C_1 \frac{\gamma_0^k (k+\gamma_0 - k\gamma_0)}{(1-\gamma_0)^2}, & |\tau_0 - \sigma| < 1 \end{cases}$$

where $k \in \mathbb{N}$*,* $\varkappa$ *is defined in (3.3),* $C_0 := \frac{\varpi}{2} e^{-|\tau_0 - \sigma|} \left(|\tau_0 - \sigma| - 1 \right)^{-1}$ *and* $C_1 := \frac{\varpi}{2} e^{-|\tau_0 - \sigma|}$*.*

Proof. Bearing in mind (3.2), we can write,

$$\begin{aligned} |\epsilon_k| &\leq \left| \sum_{\rho=k}^{\infty} (\tau - \tau_0)^\rho \left(\frac{-\varpi}{2} \right) \sum_{i=1}^{\rho} \frac{1}{(\rho - i)!} \frac{e^{-|\tau_0 - \sigma|}}{(\tau_0 - \sigma)^i} \right| \\ &\leq \sum_{\rho=k}^{\infty} \frac{\varpi}{2} |\tau - \tau_0|^\rho \sum_{i=1}^{\rho} \left| \frac{e^{-|\tau_0 - \sigma|}}{(\tau_0 - \sigma)^i} \right| \\ &= \sum_{\rho=k}^{\infty} \frac{\varpi}{2} |\tau - \tau_0|^\rho \frac{1}{e^{|\tau_0 - \sigma|}} \sum_{i=1}^{\rho} \frac{1}{|\tau_0 - \sigma|^i} \end{aligned}$$

Now three different cases must be studied:

1. If $|\tau_0 - \sigma| > 1$

$$\begin{aligned}
|\epsilon_k| &= \sum_{\rho=k}^{\infty} \frac{\varpi}{2} |\tau - \tau_0|^{\rho} \frac{1}{e^{|\tau_0-\sigma|}} \left(\frac{1 - \frac{1}{|\tau_0-\sigma|^{\rho+1}}}{1 - \frac{1}{|\tau_0-\sigma|}} - 1 \right) \\
&\leq \sum_{\rho=k}^{\infty} \frac{\varpi}{2} |\tau - \tau_0|^{\rho} \frac{1}{e^{|\tau_0-\sigma|}} \left(\frac{1}{1 - \frac{1}{|\tau_0-\sigma|}} - 1 \right) \\
&= \sum_{\rho=k}^{\infty} \frac{\varpi}{2} |\tau - \tau_0|^{\rho} e^{-|\tau_0-\sigma|} \left(|\tau_0 - \sigma| - 1 \right)^{-1}
\end{aligned}$$

and considering $C_0 := \frac{\varpi}{2} e^{-|\tau_0-\sigma|} \left(|\tau_0 - \sigma| - 1 \right)^{-1}$ it results

$$\begin{aligned}
|\epsilon_k| &= C_0 \sum_{\rho=k}^{\infty} |\tau - \tau_0|^{\rho} \\
&= C_0 \left(\sum_{\rho=0}^{\infty} |\tau - \tau_0|^{\rho} - \sum_{\rho=0}^{k-1} |\tau - \tau_0|^{\rho} \right) \\
&= C_0 \left(\frac{1}{1 - |\tau - \tau_0|} - \frac{1 - |\tau - \tau_0|^{k}}{1 - |\tau - \tau_0|} \right) \\
&= C_0 \frac{|\tau - \tau_0|^{k}}{1 - |\tau - \tau_0|} \\
&\leq C_0 \left(\frac{\varkappa^{k}}{1 - \varkappa} \right)
\end{aligned}$$

2. If $|\tau_0 - \sigma| = 1$

$$\begin{aligned}
|\epsilon_k| &\leq \sum_{\rho=k}^{\infty} \frac{\varpi}{2} |\tau - \tau_0|^{\rho} \frac{1}{e^{|\tau_0-\sigma|}} \rho \\
&= \frac{\varpi}{2e^{|\tau_0-\sigma|}} \sum_{\rho=k}^{\infty} \rho |\tau - \tau_0|^{\rho} \\
&\leq \frac{\varpi}{2e^{|\tau_0-\sigma|}} \sum_{\rho=k}^{\infty} \rho \varkappa^{\rho}
\end{aligned}$$

Bearing in mind the two following geometric series and respective sums,

- $S(\varkappa) = \sum_{\rho=0}^{\infty} \varkappa^{\rho} = \frac{1}{1-\varkappa}$

- $S'(\varkappa) = \sum_{\rho=1}^{\infty} \rho \varkappa^{\rho-1} = \frac{1}{(1-\varkappa)^2}$ and remarking that $\sum_{\rho=1}^{\infty} \rho \varkappa^{\rho-1} = \sum_{\rho=0}^{\infty} \rho \varkappa^{\rho-1}$

we take $\rho = k + j$,

$$\begin{aligned}
|\epsilon_k| &\leq \frac{\varpi}{2e^{|\tau_0-\sigma|}} \sum_{j=0}^{\infty} (k+j)\varkappa^{k+j} \\
&= \frac{\varpi}{2e^{|\tau_0-\sigma|}} \left(\sum_{j=0}^{\infty} j\varkappa^{k+j} + \sum_{j=0}^{\infty} k\varkappa^{k+j} \right) \\
&= \frac{\varpi}{2e^{|\tau_0-\sigma|}} \left(\varkappa^{k+1} \sum_{j=0}^{\infty} j\varkappa^{j-1} + k\varkappa^k \sum_{j=0}^{\infty} \varkappa^j \right) \\
&= \frac{\varpi}{2e^{|\tau_0-\sigma|}} \left(\frac{\varkappa^{k+1}}{(1-\varkappa)^2} + \frac{k\varkappa^k}{1-\varkappa} \right) \\
&= C_1 \left(\frac{\varkappa^{k+1}}{(1-\varkappa)^2} + \frac{k\varkappa^k}{1-\varkappa} \right)
\end{aligned}$$

where $C_1 := \frac{\varpi}{2} e^{-|\tau_0-\sigma|}$.

3. If $|\tau_0 - \sigma| < 1$

$$\begin{aligned}
|\epsilon_k| &\leq \frac{\varpi}{2e^{|\tau_0-\sigma|}} \sum_{\rho=k}^{\infty} |\tau - \tau_0|^{\rho} \left(\sum_{i=0}^{\rho} \frac{1}{|\tau_0-\sigma|^i} - 1 \right) \\
&= C_1 \sum_{\rho=k}^{\infty} |\tau - \tau_0|^{\rho} \sum_{i=1}^{\rho} \frac{1}{|\tau_0-\sigma|^i} \\
&\leq C_1 \sum_{\rho=k}^{\infty} |\tau - \tau_0|^{\rho} \frac{\rho}{|\tau_0-\sigma|^{\rho}} \\
&= C_1 \sum_{\rho=k}^{\infty} \rho \left(\frac{|\tau - \tau_0|}{|\tau_0-\sigma|} \right)^{\rho}
\end{aligned}$$

using (3.5) and (3.3),

$$|\epsilon_k| \leq C_1 \sum_{\rho=k}^{\infty} \rho \left(\frac{\operatorname{diam}(\alpha)}{\operatorname{dist}(\alpha, \beta)} \right)^{\rho}$$

and remembering (3.4),

$$
\begin{aligned}
|\epsilon_k| &= C_1 \sum_{\rho=k}^{\infty} \rho \gamma_0^{\rho} \\
&= C_1 \left(\sum_{\rho=0}^{\infty} \rho \gamma_0^{\rho} - \sum_{\rho=0}^{k-1} \rho \gamma_0^{\rho} \right) \\
&= C_1 \left(\gamma_0 \sum_{\rho=0}^{\infty} \rho \gamma_0^{\rho-1} - \gamma_0 \sum_{\rho=0}^{k-1} \rho \gamma_0^{\rho-1} \right) \\
&= C_1 \left(\gamma_0 \sum_{\rho=0}^{\infty} \frac{\mathrm{d}}{\mathrm{d}\gamma_0} \gamma_0^{\rho} - \gamma_0 \sum_{\rho=0}^{k-1} \frac{\mathrm{d}}{\mathrm{d}\gamma_0} \gamma_0^{\rho} \right) \\
&= C_1 \left(\gamma_0 \frac{\mathrm{d}}{\mathrm{d}\gamma_0} \sum_{\rho=0}^{\infty} \gamma_0^{\rho} - \gamma_0 \frac{\mathrm{d}}{\mathrm{d}\gamma_0} \sum_{\rho=0}^{k-1} \gamma_0^{\rho} \right) \\
&= C_1 \left(\gamma_0 \frac{\mathrm{d}}{\mathrm{d}\gamma_0} \left(\frac{1}{1-\gamma_0} \right) - \gamma_0 \frac{\mathrm{d}}{\mathrm{d}\gamma_0} \left(\frac{1-\gamma_0^k}{1-\gamma_0} \right) \right) \\
&= C_1 \left(\gamma_0 \left(\frac{1}{(1-\gamma_0)^2} \right) - \gamma_0 \left(\frac{-k\gamma_0^{k-1} + k\gamma_0^k + 1 - \gamma_0^k}{(1-\gamma_0)^2} \right) \right) \\
&= C_1 \left(\frac{\gamma_0}{(1-\gamma_0)^2} - \frac{\gamma_0 - k\gamma_0^k + (k-1)\gamma_0^{k+1}}{(1-\gamma_0)^2} \right) \\
&= C_1 \frac{\gamma_0^k (k + \gamma_0 - k\gamma_0)}{(1-\gamma_0)^2}.
\end{aligned}
$$

We remark that the three error bounds have exponential convergence. □

3.3 Interpolation approximation

Since Taylor expansion involves the computation of the derivative of the kernel function and a recursive rule for it, only seldom can be developed. Polynomial interpolation appears as a good alternative; it requires only the evaluations of the kernel function.

Using polynomial interpolation, the degenerate kernel expression (1.17) is now given by $q_\rho(\tau) = \mathcal{L}_\rho(\tau)$ and $p_\rho(\sigma) = g(x_\rho, \sigma)$.

Consider $(x_\rho)_{\rho=0}^{k-1}$ a family of interpolation points in the subdomain α (for β the procedure is similar) and $(\mathcal{L}_\rho)_{\rho=0}^{k-1}$ the corresponding Lagrange

polynomials:

$$\mathcal{L}_\rho(x) = \prod_{\theta=0,\theta\neq\rho}^{k-1} \frac{x - x_\theta}{x_\rho - x_\theta},$$

for all x in an interval $[a_1, a_2]$. For this interval, the set of interpolant points $(x_\rho)_{\rho=0}^{k}$ can be chosen as the Chebyshev points of order $k-1$:

$$x_\rho := \frac{a_2 + a_1}{2} + \frac{a_2 - a_1}{2} \cos\left(\frac{2\rho + 1}{2k}\pi\right).$$

To minimize the error in the approximation, for the reason to be soon explained, the degenerate kernel is chosen as

$$\tilde{g}(\tau, \sigma) = \begin{cases} \sum_{\rho=0}^{k-1} g(x_\rho, \sigma)\, \mathcal{L}_\rho(\tau), & \text{if } \operatorname{diam}(\alpha) \leq \operatorname{diam}(\beta), \\ \sum_{\rho=0}^{k-1} g(\tau, x_\rho)\, \mathcal{L}_\rho(\sigma), & \text{otherwise,} \end{cases} \tag{3.6}$$

and the admissibility condition (1.16) is adapted to

$$\min\{\operatorname{diam}(\alpha), \operatorname{diam}(\beta)\} < \eta \operatorname{dist}(\alpha, \beta). \tag{3.7}$$

This implies that the computation of the entries of matrices A and B from (1.19) is now done as follows: if $\operatorname{diam}(\alpha) \leq \operatorname{diam}(\beta)$,

$$A_{i\rho} = \frac{1}{h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} \mathcal{L}_\rho(\tau)\, d\tau \quad \text{and} \quad B_{j\rho} = \int_{\tau_{n,j-1}}^{\tau_{n,j}} g(x_\rho, \sigma)\, d\sigma,$$

while if $\operatorname{diam}(\beta) \leq \operatorname{diam}(\alpha)$,

$$A_{i\rho} = \frac{1}{h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} g(\tau, x_\rho)\, d\tau \quad \text{and} \quad B_{j\rho} = \int_{\tau_{n,j-1}}^{\tau_{n,j}} \mathcal{L}_\rho(\sigma)\, d\sigma.$$

All integrals can be obtained using numerical quadrature formulae, nevertheless, it is possible to have a probable more accurate result by computing analytically the integrals involving the kernel. For that, first recall the result in [AS60],

$$\frac{dE_n(z)}{dz} = -E_{n-1}(z) \quad \text{for} \quad n \in \{1, 2, 3, ...\}$$

leading immediately to

$$\int E_{n-1}(z) dz = -E_n(z). \tag{3.8}$$

Proposition 8. *When* $\operatorname{diam}(\beta) \leq \operatorname{diam}(\alpha)$*, the entries* $A_{i\rho}$ *are explicitly given by*

$$A_{i\rho} = \begin{cases} -\frac{\varpi}{2h_{n,i}} \left[E_2\left(\tau_{n,i} - x_\rho\right) - E_2\left(\tau_{n,i-1} - x_\rho\right)\right], & \tau_{n,i-1} \geq x_\rho, \tau_{n,i} \geq x_\rho \\ \frac{\varpi}{2h_{n,i}} \left[E_2\left(-\tau_{n,i} + x_\rho\right) - E_2\left(-\tau_{n,i-1} + x_\rho\right)\right], & \tau_{n,i-1} < x_\rho, \tau_{n,i} < x_\rho \\ \frac{\varpi}{2h_{n,i}} \left[-E_2\left(-\tau_{n,i-1} + x_\rho\right) - E_2\left(\tau_{n,i} - x_\rho\right) + 2\right], & \tau_{n,i-1} < x_\rho, \tau_{n,i} \geq x_\rho \end{cases}$$

Proof. Recalling that

$$\begin{aligned} g\left(\tau, x_\rho\right) &= \frac{\varpi}{2} E_1\left(\left|\tau - x_\rho\right|\right) \\ &= \begin{cases} \frac{\varpi}{2} E_1\left(\tau - x_\rho\right), & \tau \geq x_\rho \\ \\ \frac{\varpi}{2} E_1\left(-\tau + x_\rho\right), & \tau < x_\rho \end{cases} \end{aligned}$$

using (3.8) and integration by substitution, we get

1. if $\tau_{n,i-1} \geq x_\rho$ and $\tau_{n,i} \geq x_\rho$,

$$\begin{aligned} A_{i\rho} &= \frac{\varpi}{2h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} E_1\left(\tau - x_\rho\right) d\tau \\ &= -\frac{\varpi}{2h_{n,i}} \left[E_2\left(\tau_{n,i} - x_\rho\right) - E_2\left(\tau_{n,i-1} - x_\rho\right)\right] \end{aligned}$$

2. if $\tau_{n,i-1} < x_\rho$ and $\tau_{n,i} < x_\rho$,

$$\begin{aligned} A_{i\rho} &= \frac{\varpi}{2h_{n,i}} \int_{\tau_{n,i-1}}^{\tau_{n,i}} E_1\left(-\tau + x_\rho\right) d\tau \\ &= \frac{\varpi}{2h_{n,i}} \left[E_2\left(-\tau_{n,i} + x_\rho\right) - E_2\left(-\tau_{n,i-1} + x_\rho\right)\right] \end{aligned}$$

3. if $\tau_{n,i-1} < x_\rho$ and $\tau_{n,i} \geq x_\rho$,

$$\begin{aligned} A_{i\rho} &= \frac{\varpi}{2h_{n,i}} \left(\int_{\tau_{n,i-1}}^{x_\rho} E_1\left(-\tau + x_\rho\right) d\tau + \int_{x_\rho}^{\tau_{n,i}} E_1\left(\tau - x_\rho\right) d\tau \right) \\ &= \frac{\varpi}{2h_{n,i}} \left[-E_2\left(-\tau_{n,i-1} + x_\rho\right) - E_2\left(\tau_{n,i} - x_\rho\right) + 2\right] \end{aligned}$$

□

The entries of the matrix B are given in Proposition 9.

Proposition 9. *When* $\operatorname{diam}(\alpha) \leq \operatorname{diam}(\beta)$*, the entries of the matrix* B *are explicitly given by*

$$B_{j\rho} = \begin{cases} \frac{\varpi}{2}\left[E_2\left(x_\rho - \tau_{n,j}\right) - E_2\left(x_\rho - \tau_{n,j-1}\right)\right], & \tau_{n,j-1} \leq x_\rho, \tau_{n,j} \leq x_\rho \\ -\frac{\varpi}{2}\left[E_2\left(-x_\rho + \tau_{n,j}\right) - E_2\left(-x_\rho + \tau_{n,j-1}\right)\right], & \tau_{n,j-1} > x_\rho, \tau_{n,j} > x_\rho \\ \frac{\varpi}{2}\left[-E_2\left(x_\rho - \tau_{n,j-1}\right) - E_2\left(-x_\rho + \tau_{n,j}\right) + 2\right], & \tau_{n,j-1} \leq x_\rho, \tau_{n,j} > x_\rho \end{cases}$$

Proof. Using a similar procedure to the one done in Proposition 8, we obtain

1. if $\tau_{n,j-1} \leq x_\rho$ and $\tau_{n,j} \leq x_\rho$,

$$\begin{aligned} B_{j\rho} &= \frac{\varpi}{2}\int_{\tau_{n,j-1}}^{\tau_{n,j}} E_1\left(x_\rho - \sigma\right) d\sigma \text{ taking } u = x_\rho - \sigma, \\ &= \frac{\varpi}{2}\int_{x_\rho - \tau_{n,j-1}}^{x_\rho - \tau_{n,j}} E_1\left(u\right)\left(-du\right) \text{ and as } x_\rho - \tau_{n,j} \leq x_\rho - \tau_{n,j-1}, \\ &= \frac{\varpi}{2}\int_{x_\rho - \tau_{n,j}}^{x_\rho - \tau_{n,j-1}} E_1\left(u\right) du \\ &= \frac{\varpi}{2}\left[-E_2\left(x_\rho - \tau_{n,j-1}\right) + E_2\left(x_\rho - \tau_{n,j}\right)\right] \\ &= \frac{\varpi}{2}\left[E_2\left(x_\rho - \tau_{n,j}\right) - E_2\left(x_\rho - \tau_{n,j-1}\right)\right] \end{aligned}$$

2. if $\tau_{n,j-1} > x_\rho$ and $\tau_{n,j} > x_\rho$,

$$\begin{aligned} B_{j\rho} &= \frac{\varpi}{2}\int_{\tau_{n,j-1}}^{\tau_{n,j}} E_1\left(-x_\rho + \sigma\right) d\sigma \\ &= -\frac{\varpi}{2}\left[E_2\left(-x_\rho + \tau_{n,j}\right) - E_2\left(-x_\rho + \tau_{n,j-1}\right)\right] \end{aligned}$$

3. if $\tau_{n,j-1} \leq x_\rho$ and $\tau_{n,j} > x_\rho$,

$$\begin{aligned} B_{j\rho} &= \frac{\varpi}{2}\left(\int_{\tau_{n,j-1}}^{x_\rho} E_1\left(x_\rho - \sigma\right) d\sigma + \int_{x_\rho}^{\tau_{n,j}} E_1\left(-x_\rho + \sigma\right) d\sigma\right) \\ &= \frac{\varpi}{2}\left(\lim_{x \to x_\rho}\int_{\tau_{n,j-1}}^{x} E_1\left(x_\rho - \sigma\right) d\sigma + \lim_{y \to x_\rho}\int_{y}^{\tau_{n,j}} E_1\left(-x_\rho + \sigma\right) d\sigma\right) \end{aligned}$$

taking $u = x_\rho - \sigma$ and $v = -x_\rho + \sigma$,

$$B_{j\rho} = \frac{\varpi}{2}\left(\lim_{x\to x_\rho}\int_{x_\rho-\tau_{n,j-1}}^{x_\rho-x} E_1(u)(-du) + \lim_{y\to x_\rho}\int_{y-x_\rho}^{\tau_{n,j}-x_\rho} E_1(v)\,dv\right)$$

and as $x_\rho - x \leq x_\rho - \tau_{n,j-1}$,

$$\begin{aligned} B_{j\rho} &= \frac{\varpi}{2}\left(\lim_{x\to x_\rho}\int_{x_\rho-x}^{x_\rho-\tau_{n,j-1}} E_1(u)\,du + \lim_{y\to x_\rho}\int_{y-x_\rho}^{\tau_{n,j}-x_\rho} E_1(v)\,dv\right) \\ &= \frac{\varpi}{2}\left(\lim_{x\to x_\rho}\left[-E_2(u)\right]_{x_\rho-x}^{x_\rho-\tau_{n,j-1}} + \lim_{y\to x_\rho}\left[-E_2(v)\right]_{y-x_\rho}^{\tau_{n,j}-x_\rho}\right) \\ &= \frac{\varpi}{2}\left[-E_2(x_\rho-\tau_{n,j-1}) + E_2(0) - E_2(-x_\rho+\tau_{n,j}) + E_2(0)\right] \\ &= \frac{\varpi}{2}\left[-E_2(x_\rho-\tau_{n,j-1}) - E_2(-x_\rho+\tau_{n,j}) + 2\right] \end{aligned}$$

□

3.4 Error bounds for interpolation approximation

Inspired by [BG04], we give upper bounds for the error when polynomial interpolation is chosen to obtain the degenerate approximation of the kernel function.

The next Lemma introduces an upper bound for the derivative of the kernel of the integral operator we are dealing with.

Lemma 10. *The derivative of the kernel function* (1.3) *with respect to the first argument of the kernel is bounded by*

$$\left|\frac{\partial^\rho g}{\partial\tau^\rho}(\tau,\sigma)\right| \leq \frac{\varpi}{2}(\rho-1)!\frac{1}{e^{\mathrm{dist}(\alpha,\beta)}}\sum_{k=1}^{\rho}\frac{1}{[\mathrm{dist}(\alpha,\beta)]^k}. \tag{3.9}$$

Proof. From Lemma 3 and using (3.5)

$$\begin{aligned}\left|\frac{\partial^\rho g}{\partial \tau^\rho}(\tau,\sigma)\right| &\leq \left|-\frac{\varpi}{2}\sum_{k=1}^{\rho}\frac{(\rho-1)!}{(\rho-k)!}\frac{e^{-|\tau-\sigma|}}{(\tau-\sigma)^k}\right| \\ &\leq \frac{\varpi}{2}(\rho-1)!\frac{1}{e^{|\tau-\sigma|}}\sum_{k=1}^{\rho}\frac{1}{|\tau-\sigma|^k} \\ &\leq \frac{\varpi}{2}(\rho-1)!\frac{1}{e^{\mathrm{dist}(\alpha,\beta)}}\sum_{k=1}^{\rho}\frac{1}{[\mathrm{dist}(\alpha,\beta)]^k}\end{aligned}$$

□

In the next Lemma, an expression for the error bound when the interpolation points are the zeros of the Chebyshev polynomial is given:

Lemma 11. *In [SM03], for $f \in C^{k+1}[a,b]$ and using Chebyshev interpolation, a bound for the approximation error is given by*

$$\|\epsilon_k\|_\infty \leq \frac{(b-a)^{k+1}}{2^{2k+1}(k+1)!}\left\|f^{(k+1)}\right\|_\infty. \tag{3.10}$$

Applying interpolation distinctly to the different arguments of the kernel, the error bound for the approximation of the kernel g by its interpolant $\tilde{g}$, when interpolation is done with respect to the first argument of the kernel (given by the following proposition (12)), is majorized.

Proposition 12.

$$\left|\epsilon_{(k-1)_\alpha}\right| \leq \begin{cases} \frac{\varpi}{e^{\mathrm{dist}(\alpha,\beta)}k4^k}\left(\frac{1-\mathrm{diam}(\alpha)^k}{1-\mathrm{diam}(\alpha)}\right), & \mathrm{diam}(\alpha)<1 \text{ or } 1<\mathrm{diam}(\alpha)<4 \\ \frac{\varpi}{e^{\mathrm{dist}(\alpha,\beta)}4^k}, & \mathrm{diam}(\alpha)=1 \end{cases}$$

Proof. Using (3.10), (3.9) and (1.16)

$$\begin{aligned}
\left|\epsilon_{(k-1)_\alpha}\right| &\leq \frac{\operatorname{diam}(\alpha)^k}{2^{2k-1}k!}\left|\frac{\partial^k g}{\partial \tau^k}(\tau,\sigma)\right| \\
&\leq \frac{\operatorname{diam}(\alpha)^k}{2^{2k-1}k!}\frac{\varpi(k-1)!}{2e^{\operatorname{dist}(\alpha,\beta)}}\sum_{i=1}^{k}\frac{1}{[\operatorname{dist}(\alpha,\beta)]^i} \\
&= \frac{\varpi \operatorname{diam}(\alpha)^k}{4^k k}\frac{1}{e^{\operatorname{dist}(\alpha,\beta)}}\sum_{i=1}^{k}\frac{1}{[\operatorname{dist}(\alpha,\beta)]^i} \\
&= \frac{\varpi}{e^{\operatorname{dist}(\alpha,\beta)}k4^k}\sum_{i=1}^{k}\frac{\operatorname{diam}(\alpha)^k}{[\operatorname{dist}(\alpha,\beta)]^i} \\
&\leq \frac{\varpi}{e^{\operatorname{dist}(\alpha,\beta)}k4^k}\sum_{i=0}^{k-1}\operatorname{diam}(\alpha)^i \\
&= \begin{cases} \frac{\varpi}{e^{\operatorname{dist}(\alpha,\beta)}k4^k}\left(\frac{1-\operatorname{diam}(\alpha)^k}{1-\operatorname{diam}(\alpha)}\right), & \operatorname{diam}(\alpha)\neq 1 \\ \frac{\varpi}{e^{\operatorname{dist}(\alpha,\beta)}4^k}, & \operatorname{diam}(\alpha)=1 \end{cases}
\end{aligned}$$

But to ensure convergence when k tends to infinity, in the case $\operatorname{diam}(\alpha) \neq 1$, regarding the presence of the factors 4^k and $\operatorname{diam}(\alpha)^k$, respectively in the denominator and numerator, we must impose the additional condition $\operatorname{diam}(\alpha) < 4$. □

Similarly, when interpolation is done with respect to the second argument of the kernel, we get the following upper bound to the approximation error.

Proposition 13.

$$\left|\epsilon_{(k-1)_\beta}\right| \leq \begin{cases} \frac{\varpi}{e^{\operatorname{dist}(\alpha,\beta)}k4^k}\left(\frac{1-\operatorname{diam}(\beta)^k}{1-\operatorname{diam}(\beta)}\right), & \operatorname{diam}(\beta)<1 \text{ or } 1<\operatorname{diam}(\beta)<4 \\ \frac{\varpi}{e^{\operatorname{dist}(\alpha,\beta)}4^k}, & \operatorname{diam}(\beta)=1 \end{cases}$$

Proof. Analogous to Proposition 12. □

It is clear now that, in order to minimize the approximating error, interpolation should be applied distinctively to both arguments of the kernel g, taking into account the diameter of α and β, that is why for polynomial interpolation the kernel is written as (3.6). Still related with the two previous propositions, we can state

Proposition 14. *For* $D = \min\{\mathrm{diam}(\alpha), \mathrm{diam}(\beta)\}$*, we get*

$$|\epsilon_{k-1}| \leq \begin{cases} \frac{\varpi}{e^{\mathrm{dist}(\alpha,\beta)} k 4^k} \left(\frac{1-D^k}{1-D}\right), & D < 1 \text{ or } 1 < D < 4 \\ \frac{\varpi}{e^{\mathrm{dist}(\alpha,\beta)} 4^k}, & D = 1 \end{cases}$$

Proof. Take the minimum of $\mathrm{diam}(\alpha)$ and $\mathrm{diam}(\beta)$ to minimize the error bound. □

This justifies the changing of the left hand side of the admissibility condition (1.16) into (3.7) in the present approach, that is, for polynomial interpolation; the replacement with D, defined in Proposition 14, takes place for minimization of the error bound.

3.5 SVD approximation

This approach consists in computing a low-rank approximation from an explicitly built matrix block by means of a singular value decomposition (SVD).

A SVD on each admissible block, with already generated entries, can be used to preserve the most valuable information and discard the rest. The resulting rank-k approximation can be expressed as

$$AB^T = \sum_{\rho=1}^{k} u_\rho \sigma_\rho v_\rho^T, \tag{3.11}$$

where σ_ρ are the singular values (in descending order of magnitude), and u_ρ and v_ρ are the corresponding left and right singular vectors, respectively. Note that in the SVD, the eigenvectors of $(AB^T)^T(AB^T)$ are the right singular vectors of AB^T, the eigenvectors of $(AB^T)(AB^T)^T$ are the left singular vectors of AB^T and the eigenvalues of $(AB^T)^T(AB^T)$ (or $(AB^T)(AB^T)^T$) are the squares of the singular values of AB^T.

The rank, k, can be chosen to be a fixed value, or alternatively to be set dynamically on each block, based on a prescribed tolerance ϵ. In the latter case, the condition $\sigma_k > \epsilon \geq \sigma_{k+1}$ holds.

However, in computational terms, the cost to obtain the $\mathcal{H}$-matrix is high since the entries of every admissible block must be explicitly generated first, in the present case through (1.7), prior to the singular value decomposition. Nevertheless, all the computations done afterwards can be performed cheaply.

This approach may only be of pratical use if data can be stored, after being generated, for subsequent (and repetitive) use.

Chapter 4

Numerical results

In this chapter we begin by a concise summary of the machine that has been used and present some numerical experiments that aim at illustrating the benefits of the $\mathcal{H}$-matrix representation with respect to conventional storage, both in terms of performance and memory requirements.

4.1 Hardware

The tests in Sections 4.2 and 4.3 have been executed on a Linux workstation with an Intel Core i7 950 processor at 3,06 GHz with 8 MB of L3 cache memory and 8 GB of main memory. This processor has 4 cores with hyper-threading technology (a total of 8 virtual processors).

The less computer intensive tests, in Section 4.4 , have been performed on a Linux Ubuntu 11.04 desktop edition with an Intel Core i5 M450 at 2.40 GHz with 4GB of main memmory (4 cores).

4.2 Serial approach

For all the tests, in Sections 4.2 and 4.3, we chose to use a fixed value of the $\tau^\star$ parameter, in particular $\tau^\star = 4000$. We also set a constant value for

the rank and minimum size of admissible blocks (`degree`=6 and `bound`=80, respectively), as well as $\eta = 1$ in (1.16).

For building the $\mathcal{H}$-matrix we have used HLIB's *supermatrix* data structure, and recursively computed the cluster tree and populated it with admissible or inadmissible blocks, as described in Section 1.3.

In our code, we have implemented three operations of the *shell* matrix: matrix-vector multiplication, shift of origin $A := A + \sigma I$, and extraction of the diagonal. These are simply calls to the corresponding HLIB functions, appropriately wrapped according to PETSc convention. In addition, we have also implemented a *shell* spectral transformation in SLEPc that similarly implements a specialized version of the shift-and-invert technique of (1.15) by means of HLIB's LU decomposition.

A final note about the implementation is that both the exponential-integral and incomplete Gamma functions are available via GSL, the GNU Scientific Library.

In Table 4.1 we show the CPU time required for the matrix generation phase with dimension n varying from 4000 to 256000. With a uniform grid the resulting matrix is symmetric and the code takes this fact into consideration: the generation time for the symmetric case is almost half of the time required for the unsymmetric counterpart. The $\mathcal{H}$-matrix approach, either with Taylor or Lagrange for computing the admissible blocks, represents a significant gain in generation time compared with the version with conventional sparse storage (note that in the sparse version we compute all matrix elements and then decide whether they are too small to be stored). The time reported for the SVD version includes the computation of matrix elements as in the sparse version as well as the time required for low-rank approximation through SVD decomposition (with LAPACK). Although this variant is the most expensive one, if the problem is to be solved several times one may consider this approach since it allows both for a fixed rank-k and for a rank satisfying a prescribed tolerance, as mentioned in Section 3.5. For large values of n the CPU time required to compute the entries is prohibitive for the

Table 4.1: CPU time (in seconds) for the generation phase for $\tau^\star = 4000$ and varying n, with a uniform (left) and non-uniform (right) grid. The results correspond to `bound`=80 and `degree`=6.

	Uniform				Non-uniform			
n	Taylor	Lagrange	SVD	Sparse	Taylor	Lagrange	SVD	Sparse
4000	2.1	2.0	20.1	15.3	3.9	3.5	38.6	30.9
8000	6.1	5.7	99.6	61.8	10.4	9.4	182.7	123.3
16000	16.7	15.6	536.0	245.4	29.5	26.9	927.2	496.4
32000	49.6	47.0	3505.0	982.1	85.4	78.9	5630.7	1972.0
64000	140.6	133.5	–	–	245.4	227.2	–	–
128000	394.4	374.4	–	–	690.3	637.9	–	–
256000	1019.8	958.3	–	–	1783.4	1616.1	–	–

sparse implementation. The slight differences reported for Taylor and Lagrange result from the fact that in the latter case we implemented numerical quadrature while for Taylor we used the formulae presented in Section 3.1; the computation of the incomplete Gamma function at the required points is skewing the results a bit. For increasing problem size there is a constant growth factor less than three for these two approaches while the problem size is quadrupling. The growth factor respects the estimated $n \log(n)$ asymptotic cost, in contrast with the sparse version that follows n^2. Some values were not reported due to their high value.

Table 4.2 complements the previous comments, showing the number of stored elements for all approaches. Note that the actual memory requirements for sparse storage are quite larger, since the space needed for indices is considerable, while for the $\mathcal{H}$-matrix representation the overhead in negligible. The last column shows the compression factor for $\mathcal{H}$-matrix format as a percentage of the full (dense) storage, revealing noteworthy gains for increasing values of n.

Table 4.3 reports on the CPU time for the solution phase using Taylor, Lagrange and SVD data-sparse representation as well as the sparse approach.

Table 4.2: Number of stored elements in the case of dense, sparse and $\mathcal{H}$-matrix representation, for the non-uniform grid case in Table 4.1.

n	Dense	Sparse	$\mathcal{H}$-matrix	Compression
4000	$1.6 \cdot 10^7$	$2.9 \cdot 10^5$	$1.9 \cdot 10^6$	12.0%
8000	$6.4 \cdot 10^7$	$1.2 \cdot 10^6$	$4.2 \cdot 10^6$	6.5%
16000	$2.6 \cdot 10^8$	$4.5 \cdot 10^6$	$9.1 \cdot 10^6$	3.5%
32000	$1.0 \cdot 10^9$	$1.8 \cdot 10^7$	$1.9 \cdot 10^7$	1.9%
64000	$4.1 \cdot 10^9$	–	$4.1 \cdot 10^7$	1.0%
128000	$1.6 \cdot 10^{10}$	–	$8.6 \cdot 10^7$	0.5%
256000	$6.6 \cdot 10^{10}$	–	$1.9 \cdot 10^8$	0.28%

Since the spectrum is tightly clustered (see Table 4.4 for the five largest eigenvalues with relative tolerance on the residual of 10^{-7}), the shift-and-invert technique is required to enable convergence of the Krylov-Schur method. In the following, an LU factorization on the $\mathcal{H}$-matrix representation is used in the linear solver required in the application of the shift-and-invert operator. The factorization is the most costly operation but is performed only once, while triangular solves are required at each iteration of the eigensolver. In Table 4.3 we show the factorization time as well as the total solution time. For these tests, we used a Krylov basis of dimension 16, and with this size the method does not need to restart (except for the matrix of $n = 256000$ where two restarts are required).

As expected, the computation of eigenpairs with the implementations of the data-sparse representation is very fast compared to the sparse storage, which shows a fast degradation in performance for increasing dimension. The SVD approach is competitive with Lagrange and Taylor approximations, and results for SVD on large values of n are not reported only due to the high generation time. As mentioned above, the present problem is hard to solve since for incresing values of n, and for fixed $\tau^\star$, the eigenvalues tend to become more and more clustered. For problems with better separation of the spectrum, the shift-and-invert step can be avoided and consequently its

Table 4.3: CPU time (in seconds) for the solution phase for $\tau^\star = 4000$ and varying n, with a non-uniform grid (non-symmetric case). The results correspond to `bound`=80 and `degree`=6.

	Solution				Factorization	
n	Taylor	Lagrange	SVD	Sparse	Taylor	Lagrange
4000	0.3	0.3	0.3	0.2	0.3	0.3
8000	0.5	0.5	0.5	1.1	0.4	0.4
16000	1.0	1.1	1.2	8.2	0.8	0.9
32000	2.6	2.7	2.9	66.3	2.1	2.3
64000	6.5	6.8	–	–	5.6	5.9
128000	16.4	17.3	–	–	14.5	15.4
256000	47.5	49.3	–	–	39.3	41.3

Table 4.4: Computed eigenvalues for the case of a uniform grid with $n = 16000$ and $\tau^\star = 4000$.

Eigenvalue	Taylor	Lagrange	SVD
λ_1	0.749999843422	0.749999843459	0.749999843598
λ_2	0.749999374216	0.749999374253	0.749999374391
λ_3	0.749998592208	0.749998592245	0.749998592383
λ_4	0.749997497401	0.749997497438	0.749997497576
λ_5	0.749996089801	0.749996089838	0.749996089976

computational cost.

As mentioned at the end of Section 1.2, Davidson methods do not seem appropriate in the context of hierarchical matrices, since the shift-and-invert technique is very cheap in this case. However, we wanted to do some experiments. With Jacobi (diagonal) preconditioning, we were able to solve the problem (although after many iterations) by tuning the parameters of SLEPc's Davidson solver. For instance, for $n = 8000$ with uniform grid, the response time is 15.5 seconds, as opposed to 0.5 seconds with shift-and-invert Krylov-Schur. A much powerful preconditioner is to use the LU factoriza-

Table 4.5: Execution time (in seconds) for the matrix generation in parallel (for different number of threads, p) corresponding to the two longest times in Table 4.1 (non-uniform grid, Taylor with $n = 256000$ and SVD with $n = 32000$).

	Taylor 256000		SVD 32000	
p	Time	Speedup	Time	Speedup
1	1783.4	–	5630.7	–
2	891.9	1.99	3097.2	1.82
4	446.0	3.99	1983.4	2.83
6	381.0	4.68	1875.1	3.00
8	332.9	5.35	1834.0	3.07

tion, but then the behaviour is almost identical to shift-and-invert. Again, we remark that in other applications with a different spectrum, Davidson solvers could be more useful than in this case.

4.3 Parallel approach (threads)

We have also developed a straightforward parallel version of the $\mathcal{H}$-matrix generation, based on the OpenMP API for shared-memory parallel programming. In particular, we follow a *tasking* approach with the OpenMP `task` directive [ACD+09], where each recursive call constitutes a new task.

Regarding parallelization of the generation phase, the multi-threaded version was analyzed up to 8 threads.

Table 4.5 shows the measured execution times along with the achieved speedups, for the two longest times in Table 4.1 (non-uniform grid, Taylor with $n = 256000$ and SVD with $n = 32000$). In the case of the Taylor approximation, speedup is virtually ideal up to 4 threads and decays significantly later. This can be attributed in part to the fact that only 4 physical cores are available. In the case of SVD generation, speedup is much worse, thus revealing a problem with load imbalance, due to the fact that the high cost of

the decompostion (cubic in the matrix block size) makes parallel tasks differ wildly in duration.

4.4 Sensitivity analysis

We consider now a smaller problem, $\tau^\star = 1000$ and $n = 2000$, to assess the sensitivity of the proposed resolution method regarding to some of the parameters.

SVD

degree/bound	5	10	20	40	80
3	2,53E-09	2,33E-10	1,18E-11	1,18E-11	1,18E-11
6	1,18E-11	1,18E-11	1,18E-11	1,18E-11	1,18E-11
9	8,39E-01	1,18E-11	1,18E-11	1,18E-11	1,18E-11

Taylor

degree/bound	5	10	20	40	80
3	2,46E-03	2,98E-04	5,05E-06	1,92E-09	1,18E-11
6	1,71E-03	8,40E-05	1,73E-06	1,53E-09	1,18E-11
9	1,72E-03	8,33E-05	4,61E-07	7,62E-10	1,18E-11

Lagrange

degree/bound	5	10	20	40	80
3	3,13E-04	9,15E-05	2,68E-06	1,15E-09	1,18E-11
6	1,19E-06	1,15E-06	1,42E-07	2,45E-10	1,18E-11
9	4,62E-09	4,52E-09	2,55E-09	1,94E-11	1,18E-11

Figure 4.1: Norm of the difference between matrix A_n and its $\mathcal{H}$-matrix representation, for several values of the degree k and minimum bound.

The two main parameters under review are k, the degree on the Taylor or Lagrange approximations, and the minimum size for the blocks, named `bound`. The former is already well understood, larger values of k should improve the quality of the approximation but at a higher cost. The latter is a numerical way to stop the splitting process before reaching sets containing only one element; it defines which sets are small enough.

In Figure 4.1 we show the distance from the $\mathcal{H}$-matrix approximation to the original one, A_n. For most part of the combinations of both parameters,

Eigenvalue	Sparse	Taylor	Lagrange
λ_1	0.749997399534	0.749997056849	0.749997402836
λ_2	0.749989598317	0.749989290172	0.749989601309
λ_3	0.749976596889	0.749976284686	0.749976599928
λ_4	0.749958396151	0.749958079593	0.749958399195
λ_5	0.749934997365	0.749934670108	0.749935000520

Table 4.6: List of the 5 largest eigenvalues produced by the sparse version and Taylor and Lagrange approximations, using degree 6, bound 20

the distance between the matrices is sufficiently small, providing good perspectives for the success of the numerical approximations. It is important to note that the eigenvalues are computed with a tolerance of 10^{-7}. Smaller values for `bound` reduce the minimum sizes of the blocks thus increasing the number of blocks, particularly the low-rank approximation ones, and the $\mathcal{H}$-matrix becomes sparser. With the decrease of `bound` the admissibility condition becomes eager. Similarly, lower values of `degree` may produce worst but cheaper approximations. These two parameters must be considered together, since it may not make sense, for instance, to consider a `degree` greater than a `bound`. Very low values of either parameters may lead to wrong answers (values reported in red and orange color show a significative difference between A_n and its $\mathcal{H}$-matrix approximation). Moreover, this analysis is, naturally, problem dependent, although there are geometrical considerations of general nature.

For the problem under study, Lagrange approximations seem to be slightly better than Taylor ones, allowing best approximations either for fixed values of `bound or degree`. This can be justified by the fact that the former is a global approximation while the latter is of local nature. Furthermore, Lagrange is more flexible since it does not require the computation of derivatives. As stated previously, SVD decomposition allows for the best approximation but it is very expensive to compute.

Table 4.6 shows the computed eigenvalues for the Sparse, Taylor and

Lagrange approximations, for `degree` 6 and `bound` 20. The relative error (distance from the approximation solution to the one provided by the Sparse version (reference)) is of $O(10^{-7})$ for Taylor and of $O(10^{-9})$ for Lagrange. One should bear in mind that the required precision for the solution was 10^{-7}, so both solutions should be considered correct under the precision of data. For the same `bound` but reducing the `degree` to 3, both approximations deliver a relative error of $O(10^{-7})$. For larger values of `bound`, the number of correct digits of the approximate solutions with respect to the reference case increases, so results are insensitive to this parameter. To explore further this aspect, one should produce more refined eigenvalue approximate solutions. On the other hand, smaller values of bound, 10 or less, begin to produce solutions under the required precision, so the relative error, as mesured in this section, becomes larger.

For SVD, the only case where solutions don't match at least at $O(10^{-7})$ is for `degree` 9 and `bound` 5 (indeed, as explained, this combination of parameters is no sense). Except for the previous case, singular value decomposition captures extremely well the relevant information for the eingenvalue computation.

To better clarify the impact of the `bound` parameter on the $\mathcal{H}$-matrix, we plot in Figures 4.2 to 4.5 the structure for a small size problem, $\tau^\star = n = 100$, and for a fixed `degree` equal to 6. It is clear, Figure 4.5, that a larger `bound`, is not adequate for such a small size problem. At the other extreme, a small `bound`, Figure 4.2, produces very small inadmissible (and admissible) blocks, giving rise to important information, green shapes, in blocks very far from the diagonal; for such a small bound, maybe a higher degree would be recommended. Figure 4.4 illustrates how this type of approximation can be used as a clever and cheap way to treat the problem at a cost of a block-band matrix eigenvalue computation. Probably the best choice would be Figure 4.3, combining a moderate size for the inadmissible blocks and setting the admissible blocks neither in an atomized form nor too much aggregated.

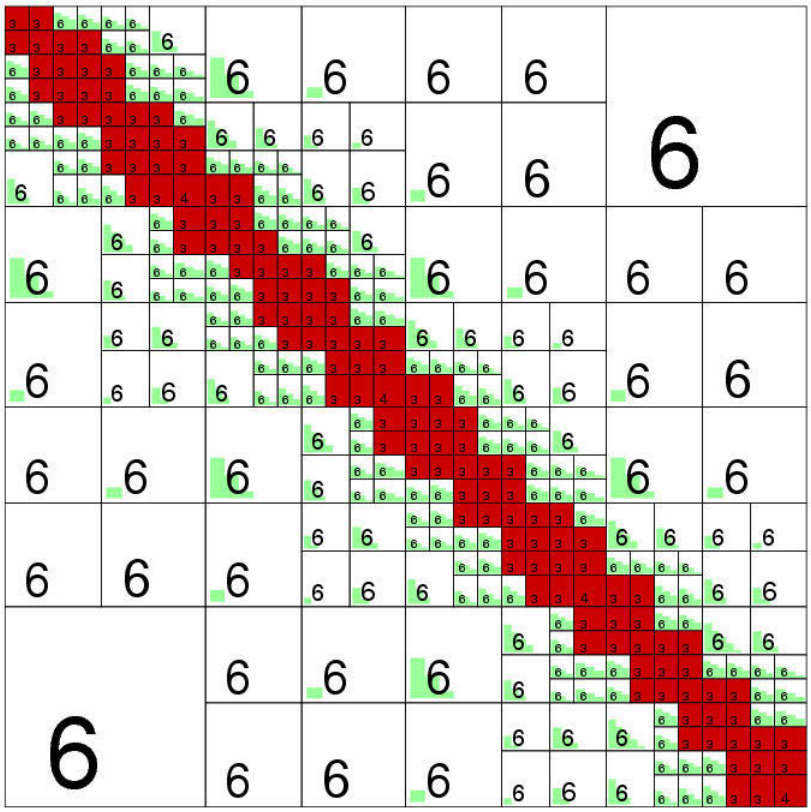

Figure 4.2: $\mathcal{H}$-matrix structure for Lagrange approximation with `degree` 6 and `bound` 5 ($\tau^\star = 100,\ \ n = 100$).

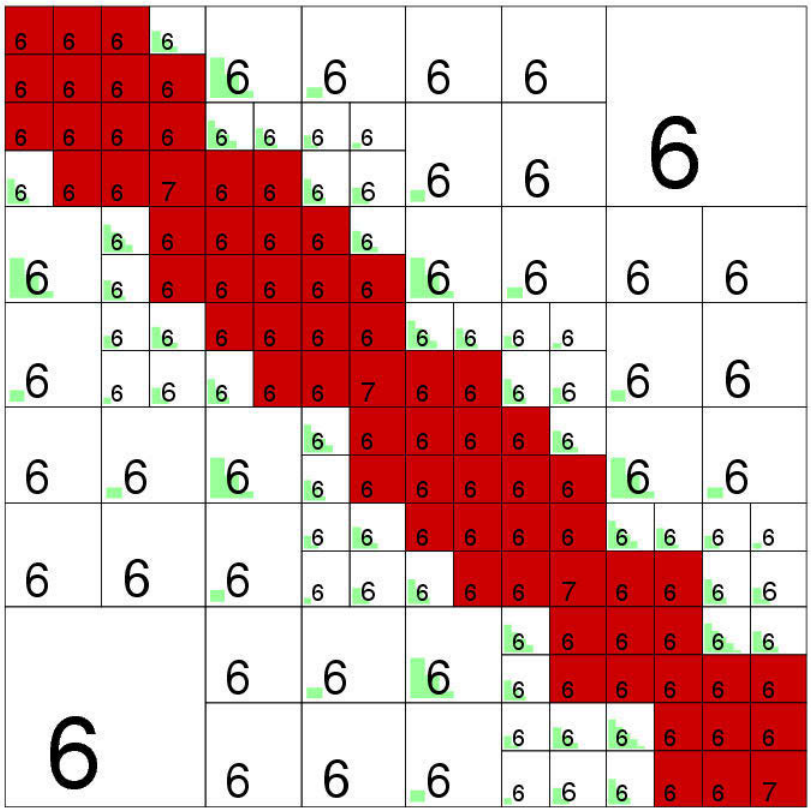

Figure 4.3: $\mathcal{H}$-matrix structure for Lagrange approximation with `degree` 6 and `bound` 10 ($\tau^\star = 100,\ \ n = 100$).

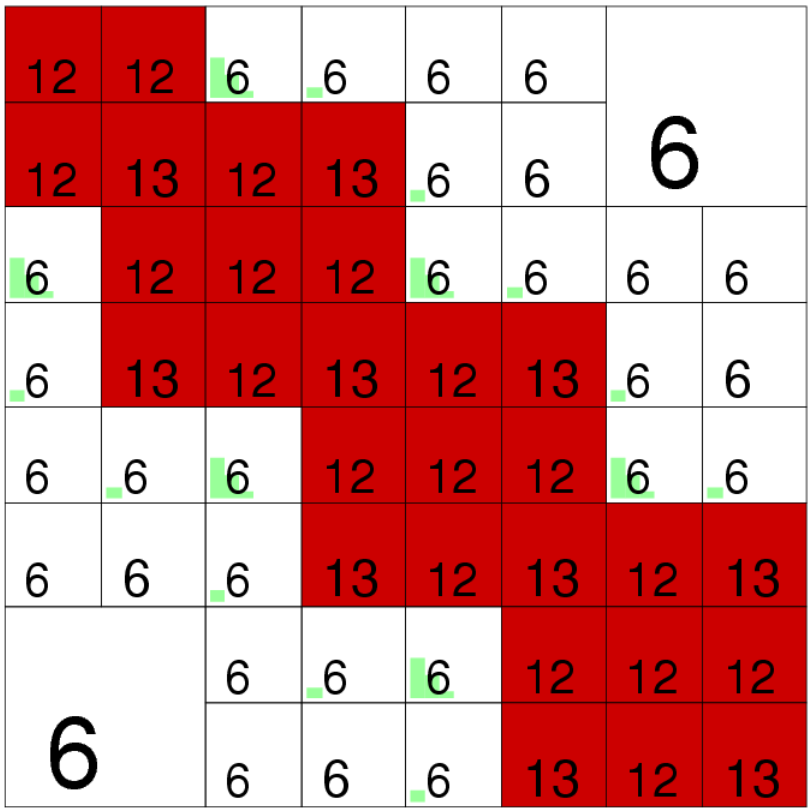

Figure 4.4: $\mathcal{H}$-matrix structure for Lagrange approximation with `degree` 6 and `bound` 20 ($\tau^\star = 100,\ \ n = 100$).

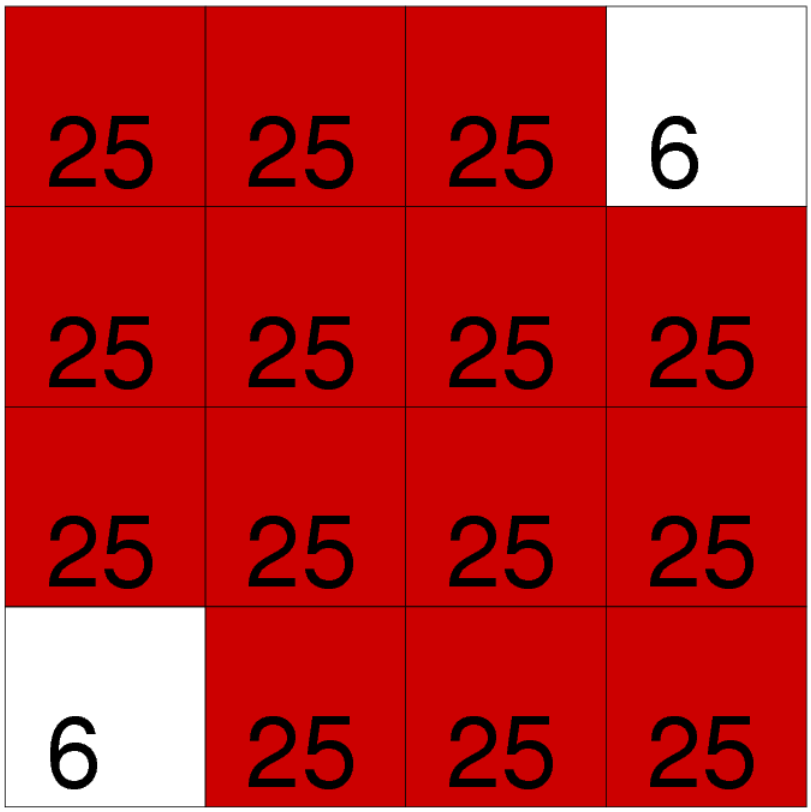

Figure 4.5: $\mathcal{H}$-matrix structure for Lagrange approximation with `degree` 6 and `bound` 40 ($\tau^\star = 100,\ \ n = 100$).

Part III

Conclusions and future research

Chapter 5

Conclusions

The $\mathcal{H}$-matrix technique has its origin in the panel clustering [HN89, Hac90] where the kernel of the operator is approximated by a degenerate one. However, the panel clustering matrices cannot cheaply be multiplied or inverted, and we arrive to one great advantage of using $\mathcal{H}$-matrices: the arithmetic, that is, matrix-vector multiplication, matrix-matrix multiplication, inversion, matrix functions, can be performed in almost linear complexity. So, new applications as the solution of matrix equations or just the construction of very efficient preconditioners for linear systems can be handled with efficacy using this technique, [Bör09].

In this work we solve a large eigenvalue problem issued from a radiative transfer equation in stellar atmospheres. The computation is time intensive both in memory requirements, since large dimensional cases are to be treated, and in the solution phase, due to the clustering of the eigenvalues. Sophisticated numerical algorithm implementations for the solution phase are thus required. The $\mathcal{H}$-matrix data representation provided by HLIB was integrated in the SLEPc and PETSc frameworks to tackle those two difficulties.

We report on the numerical low-rank approximations developed, on the details of the integration of the libraries under use, as well as presented

a brief explanation of the problem and its interesting characteristics. The combined use of numerical solution methods optimized for high-end computer environments and of the clever data storage representations, although not with ease, provide an efficient and fast answer capable of solving even larger dimensional problems, maybe requiring the use of distributed processing. The numerical tests illustrate the success of the proposed solutions.

Moreover, error bounds for the proposed low-rank approximations were developed, ensuring that under the stated conditions, these approximations are mathematically supported.

This thesis also presents a useful summary on numerical state-of-art methods for the solution of algebraic eigenvalue problems and on some of today's most important software libraries for high-end scientific computing.

Intentionally, some important computational skills to fullfil the aims of this work were hidden to keep the reading as much clear and fluent as possible. Indeed, dealing with such huge number of software libraries, knowing in depth the merits and limitations of the numerical algorithms involved, requires a high level of specialization. Furthermore, being familiar with computer languages is crucial.

Chapter 6

Future research

In this chapter, we make some comments on future research directions that may amplify the application of the $\mathcal{H}$-matrix technique to other cases, which would provide the next steps along the path to a more wide application of it. Moreover, in Section 6.2 it is described some state-of-art about some topics in nonlinear eigenproblems.

6.1 Some limitations on $\mathcal{H}$-matrix technique

We have started out dealing with an integral operator defined from $L^1([0,1])$ into itself, and with kernel

$$f(\tau,\sigma) = \ln\left(1 - \cos 2\pi\left(\tau - \sigma\right)\right), \tag{6.1}$$

aiming to compute some few eigenvalues and eigenvectors of the discretized matrix of the operator, with an analogous aproach as it was done with the operator T in (1.2) with kernel (1.3). The kernel (6.1) was purposely chosen as it is a wealky singular kernel for which the exact eigenvalues are known:

- simple eigenvalue: $\lambda_0 = -\ln 2$,
- double eigenvalues: $\lambda_k = -\frac{1}{k}$, $k \geq 1$.

It would be an interesting work to compute a few eigenvalues using the technique used in the thesis, followed by the comparsion between them and the corresponding exact ones.

Nevertheless, the study done with this kernel showed us the apparent restriction of the use of the $\mathcal{H}$-matrix technique, similiar to what was done with the kernel (1.3) in this thesis. We have encountered some difficulties: we did not manage to find a solution to the computation of the double integral of the kernel (6.1), necessary to achieve the formula for the inadmissible leaves of the $\mathcal{H}$-matrix structure; we were unable to find a recursive rule for the derivative of the kernel, essential to the construction of the Taylor series expansion of the kernel, one of the options for the low-rank construction; we did not manage to write the kernel with both variables separated, prerequisite to construct a degenerate approximation of the kernel, see (1.17), for the low-rank approximation.

Despite the previous facts, polynomial interpolation is an optimal option for the low-rank approximation in this case, using a similar procedure to what was done in Section 3.3, except for the cases of low-rank approximation close to singularities, besides that the question remains how to compute the inadmissible leaves. These two points appear to be a good point of research in an immediate future work, and we leave here a hint of the study we are having in hands.

One possibility is to approximate the new operator T by $\tilde{T}_n$ obtained from T by means of a particular quadrature approximation. First we rewrite the kernel as a function of one variable, $f(\tau, \sigma) = \tilde{f}(r)$, with $r = |\tau - \sigma|$. The essence of this procedure is using a truncated kernel function $\tilde{f}_n$ instead of the kernel itself. The truncation is done close enough to the singularity, that is, given a point c_n of the same order of the weights of the quadrature formula (e.g. for trapezoidal quadrature rule it is of order $\frac{1}{n}$), for the points of the domain located between c_n and the singularity, the kernel is evaluated as $\tilde{f}_n(c_n)$; in the other points of the domain, $\tilde{f}_n$ is defined to be $\tilde{f}$. These ideas may be found in [LL93].

6.2 First glance on nonlinear eigenvalue problems

At the begining of the research project, we had thought to make some contributions in nonlinear eigenvalue problems, but in fact the research followed other directions. Nevertheless, some investigation about this subject was done, speaking in terms of state-of-the-art, and in this section we present a summary of it as we consider nonlinear eigenproblems one topic for research in the future.

Regarding the importance of spectral theory in the case of linear operators, for instance, spectra of differential operators applied to the theory of elliptic boundary value problems or the application in classical quantum mechanics, it is quite natural that several efforts have been made to study and define spectra in the nonlinear case.

In order to recall the definition of spectrum in the linear case, let X and Y be two Banach spaces over $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$ and an operator $L \in BL(X)$, where $BL(X)$ is the algebra of all bounded linear operators from X into itself. The `resolvent set` of L is defined by

$$\mathrm{re}\,(L) = \{\lambda \in \mathbb{K} : \lambda I - L \text{ is a bijection}\} \tag{6.2}$$

and the `spectrum` of the operator by

$$\mathrm{sp}\,(L) = \mathbb{K}\backslash \mathrm{re}(L) = \{\lambda \in \mathbb{K} : \lambda I - L \text{ is not a bijection}\}, \tag{6.3}$$

where I is the identity operator.

The study of nonlinear eigenvalues is older than the nonlinear spectral theory, remarking that initially, the word *spectrum* was used in the sense of the set of eigenvalues, the *point spectrum*. Considering $F : X \longrightarrow X$ a continuous nonlinear operator, we note that the definition of the *point spectrum* of F coincides with the one of the linear case

$$\mathrm{sp}_p\,(F) = \{\lambda \in \mathbb{K} : F(x) = \lambda x \text{ for some } x \neq 0\}. \tag{6.4}$$

It was just in the late 60's that Kachurovskij and Neuberger independently studied spectral values that were not necessarily eigenvalues. Then, serveral different spectral "theories" started to appear, aiming to preserve the properties considered useful regarding the linear case, as well as trying to extend them to a possible wider range of nonlinear problems.

In [ADPV04], the authors describe that it was assumed that a spectrum of a continuous nonlinear operator should satisfy some essential requirements, as reducing to the familiar spectrum in case of linear operators; keeping some of the usual properties of the linear spectrum (e.g. compactness, nonemptiness), containing the eigenvalues of the operator considered, as well as having applications (e.g. existence, uniqueness).

On the contrary of what could be possibly expected, nonlinear spectral theory have faced some "disappointing" facts. For example, in oposition to the linear case: the spectrum of a nonlinear operator has almost no information about the operator itself; in general the properties of boundedness, closedness, or nonemptiness fail; the spectra can be disjoint from the set of eigenvalues.

Next we give a glimpse over some spectra that have appeared over the last forty years:

- Rhodius spectrum [Rho77],
  Consider $F : X \longrightarrow X$ a continuous operator such that $F(0) = 0$. The set

  $$\begin{aligned} \mathrm{sp}_R(F) &= \mathbb{K} \backslash \{\lambda \in \mathbb{K} : \lambda I - F \text{ homeomorphism on } X\} \\ &= \mathbb{K} \backslash \left\{\lambda \in \mathbb{K} : \lambda I - F \text{ is a bijection and } (\lambda I - F)^{-1} \text{ continuous}\right\} \end{aligned}$$

  is the *Rhodius spectrum* of F.

- Neuberger spectrum [Neu69],
  Consider $F : X \longrightarrow X$ a continuously differentiable operator such that $F(0) = 0$. The set

  $$\begin{aligned} \mathrm{sp}_N(F) &= \mathbb{K} \backslash \{\lambda \in \mathbb{K} : \lambda I - F \text{ diffeomorphism on } X\} \\ &= \mathbb{K} \backslash \left\{\lambda \in \mathbb{K} : \lambda I - F \text{ is a bijection and } (\lambda I - F)^{-1} \text{ of class } C^1\right\} \end{aligned}$$

is the *Neuberger spectrum* of F.

- Kachurovskij spectrum [Kac69],
  Consider $F : X \longrightarrow X$ belonging to the class $Lip(X)$ such that $F(0) = 0$. $F \in Lip(X)$ means that F is Lipschitz continuous on X, that is,

$$[F]_{Lip} = \sup_{x \neq y} \frac{\|F(x) - F(y)\|}{\|x - y\|} < \infty.$$

  The set

$$\begin{aligned} \mathrm{sp}_K(F) &= \mathbb{K} \backslash \{\lambda \in \mathbb{K} : \lambda I - F \text{ lipeomorphism on } X\} \\ &= \mathbb{K} \backslash \left\{\lambda \in \mathbb{K} : \lambda I - F \text{ is a bijection and } (\lambda I - F)^{-1} \in Lip(X)\right\} \end{aligned}$$

  is the *Kachurovskij spectrum* of F.

- Dörfner spectrum [Dör97],
  Consider $F : X \longrightarrow X$ linearly bounded on X, i.e.

$$[F]_B = \sup_{x \neq 0} \frac{\|F(x)\|}{\|x\|} < \infty.$$

  The set

$$\begin{aligned} \mathrm{sp}_D(F) = &\mathbb{K} \backslash \left\{\lambda \in \mathbb{K} : \lambda I - F \text{ is a bijection and } (\lambda I - F)^{-1} \text{ is} \right. \\ &\left. \text{linearly bounded on } X\right\} \end{aligned}$$

  is the *Dörfner spectrum* of F.

- Furi-Martelli-Vignoli spectrum [FMV78],
  Before the definition of this spectrum, some other definitions have to be presented. The (Karatowski) *measure of compactness* $\alpha(A)$ of a bounded set $A \subset X$ is defined as the infimum of real numbers $\delta > 0$ such that A admites a finite covering by sets of diameter less than δ. Considering $F : X \longrightarrow Y$ continuous, the following numbers are defined

$$[F]_\alpha = \sup \left\{k > 0 : \alpha\left(F\left(A\right)\right) \geq k\alpha\left(A\right) \text{ for every bounded } A \subset X\right\},$$

$$[F]_Q = \lim_{\|x\|\to\infty} \sup \frac{\|F(x)\|}{\|x\|}, \tag{6.5}$$

and

$$[F]_q = \lim_{\|x\|\to\infty} \inf \frac{\|F(x)\|}{\|x\|}. \tag{6.6}$$

The operator F is said to be *stably solvable* if, given any compact operator G in X with $[G]_Q = 0$, the equation $F(x) = G(x)$ has a solution $x \in X$.

Finally, F is said to be *FMV-regular* if F is stably solvable, $[F]_\alpha > 0$ and $[F]_q > 0$. Now we have all the notations needed to define the *Furi-Martelli-Vignoli spectrum* (FMV spectrum):

$$\mathrm{sp}_{FMV}(F) = \mathbb{K}\backslash \{\lambda \in \mathbb{K} : \lambda I - F \text{ is } FMV\text{-regular}\}.$$

- Feng spectrum [Fen97],
  Consider $F : X \longrightarrow Y$ continuous on X. We give the following notations to achive the definition of the Feng spectrum:

$$[F]_A = \inf \{k > 0 : \alpha(F(A)) \leq k\alpha(A) \text{ for every bounded } A \subset X\}, \tag{6.7}$$

$$[F]_b = \inf_{x\neq 0} \frac{\|F(x)\|}{\|x\|}.$$

  Considering $G : B_r \longrightarrow Y$ a continuous operator, with $B_r = \{x \in X : \|x\| \leq r\}$, it is defined

$$\begin{aligned}\nu_r(F) = &\inf \{k > 0, \text{ there exists } G \text{ satisfying } [G]_A \leq 0, G\mid_{\partial B_r} \equiv 0,\\ &\text{and } F(x) \neq G(x), \forall x \in B_r\}\end{aligned}$$

  and

$$\nu(F) = \inf \{\nu_r(F), r > 0\}.$$

  Finally, F is said to be *F-regular* if F is stably solvable, $[F]_\alpha > 0$ and $[F]_b > 0$ and $\nu(F) > 0$. Now we have all the notations needed to define the *Feng spectrum*:

$$\mathrm{sp}_F(F) = \mathbb{K}\backslash \{\lambda \in \mathbb{K} : \lambda I - F \text{ is } F\text{-regular}\}.$$

- Väth phantom [SV00],
  Let $U \subset X$ be open, bounded, connected and with $0 \in U$. Consider $F : \overline{U} \longrightarrow Y$ continuous.

  F is *epi* on $\overline{U}$ if $F(x) \neq 0$ on ∂U and the equation $F(x) = G(x)$ has a solution in U for any compact operator $G : \overline{U} \longrightarrow Y$ that satisfies $G(x) \equiv 0$ on ∂U.

  F is *strictly epi* on U if

  $$\inf \{\|F(x)\| : x \in \partial U\} > 0,$$

  and there exists some $k > 0$ such that for any operator $G : \overline{U} \longrightarrow Y$ that satisfies $G(x) \equiv 0$ on ∂U and $[G]_A \leq k$, the equation $F(x) = G(x)$ has a solution in U.

  Finally, $F : X \longrightarrow Y$ is said to be *v-regular* if it is strictly epi on some U. Now we have all the notations needed to define the *Väth phantom*:

  $$\phi(F) = \{\lambda \in \mathbb{K} : \lambda I - F \text{ is not v-regular}\}.$$

- Calamai-Furi-Vignoli spectrum [CFV10],
  Consider $U \subset X$ open, the continuous operator $F : U \longrightarrow Y$ and $p \in U$. Let U_p denote the open neighborhood $\{x \in X : p + x \in U\}$ of $0 \in X$ and consider $F_p : U_p \longrightarrow Y$ continuous with $F_p(x) = F(p + x) - F(p)$.

  Recalling (6.7) and (6.6), considering $B(p, r) \subset X$ an open ball with center at p and radius $r > 0$, $B(p, r) \subset U$, it is defined now

  $$[F]_{A_p} = \lim_{r \to 0} [F \mid_{B(p,r)}]_A,$$

  $$[F]_{q_p} = \lim_{r \to 0} [F \mid_{B(p,r)}]_q,$$

  and, for $y \in Y$, F is *y-admissible at* p if $F(p) = y$ and $F(x) \neq y$ for every x in a constricted neighborhood of p. Moreover, F is said to be *y-epi at* p if it is y-admissible at p and y-epi on any small enough neighborhood of p (the definition of F y-epi can be seen, e.g. in [CFV09]).

Finally, F is said to be *regular at* p if $[F]_{A_p} > 0$ and $[F]_{q_p} > 0$ and F_p is 0-epi at 0. Now we are able to define the *spectrum of the map* F *at the point* p, that is, the *Calamai-Furi-Vignoli spectrum*:

$$\mathrm{sp}_{CFV}(F,p) = \{\lambda \in \mathbb{K} : \lambda I - F \text{ is not regular at } p\}\,.$$

Besides the previous references to the papers where the respective spectrum was introduced, in [ADPV04, APV00, App03] just to mention a few, it is possible to find not only the definitions, but also the comparison between some of the spectra previously mentioned. Next, not aiming to be exhaustives, we illustrate some differences between these spectra.

The Rhodius, Neuberg, Kachurovskij and Dörfner spectra contain the eigenvalues of the operator, and 0 if the operator is compact in a Banach space of infinite dimension (this is analogous to the linear case). It can be seen that in these spectra, what was done in their definitions was simply the replacement of $BL(X)$ in (6.2) and (6.3) by other classes of continuous nonlinear operators, in this four cases, respectively by the class of continuous, continuously differentiable, Lipschitz continuous, and linearly bounded operators.

Apparently, the *asymptotic* spectrum ($\|x\| \to \infty$) introduced by Furi, Martelli and Vignoli, the *global* Feng spectrum ($x \in X$), and the *local* spectrum $\left(x \in \overline{U}\right)$ given by Väth are more present in literature, probably because, even though being quite different in nature, are identical for homogeneous operators and when applied to linear operators they give exactly the usual spectrum of the linear theory. Note that usually the word "spectrum" in the case of Väth is replaced by "phantom" because its construction is quite far from what is usually called spectrum.

The fact that FMV spectrum do not contain the set of eigenvalues, has induced Feng to define a new spectrum having the same topological properties of the FMV spectrum, but containing the classical eigenvalues, as happens in the linear case. The Väth phantom is closed and can be in some cases bounded, hence compact.

In Table 6.1, inspired in [CFV09], we give a summary of the comparison of the different spectra described before, from the viewpoint of topological properties (having in mind the properties of the linear spectra).

Table 6.1: A general view of the main properties of some nonlinear spectra

Spectrum	nonempty	closed	bounded	compact
Rhodius	no*	no	no	no
Kachurovskij	no*	yes	yes	yes
Neuberg	yes	no	no	no
Dörfner	no*	no	no	no
FMV	no*	yes	no*	no*
Feng	no*	yes	no*	no*
CFV	no*	yes	no*	no*

(*) yes, in some cases [CFV09].

Considering $F, J : X \longrightarrow Y$ continuous operators, the point spectrum (6.4) can be written in a more general way

$$\mathrm{sp}_p(F, J) = \{\lambda \in \mathbb{K} : F(x) = \lambda J(x) \text{ for some } x \neq 0\}. \tag{6.8}$$

Anyway, the first definition is a particular case of (6.8), that is, $\mathrm{sp}_p(F) = \mathrm{sp}_p(F, I)$, with $X = Y$.

Recalling that, in the linear case, the *approximate point spectrum of* L is defined by

$$\begin{aligned}\mathrm{sp}_{ap}(L) = \{\lambda \in \mathbb{K} : &\text{ exists a sequence } x_n, \|x_n\| \to 0 \\ &\text{such that } \|\lambda x_n - L x_n\| \to 0 \text{ as } n \to \infty\}\end{aligned}$$

in some of the nonlinear spectra a corresponding notion of eigenvalues (point spectrum or approximate point spectrum) was introduced; note that $\mathrm{sp}_p(L) \subseteq \mathrm{sp}_{ap}(L)$ and $\mathrm{sp}(L) = \mathrm{sp}_{ap}(L) \cup \mathrm{sp}_p(L^*) = \mathrm{sp}_p(L) \cup \mathrm{sp}_{ap}(L^*)$, being $L^* \in BL(X^*)$ the adjoint of L.

As was mentioned before, the FMV-spectrum has asymptotic characteristics, and so (6.8) should be replaced by the *asymptotic point spectrum*

$$\mathrm{sp}_q(F,J) = \{\lambda \in \mathbb{K} : [\lambda J - F]_q = 0\},$$

that is always in $\mathrm{sp}_{FMV}(F)$.

In [SV00, ADPV04] it is possible to find *point phantom of* (F,J) defined as

$$\phi_p(F,J) = \{\lambda \in \mathbb{K} : \lambda \text{ connected eigenvalue of}(F,J)\}$$

where by connected eigenvalue of (F,J) is meant the nullset

$$N(\lambda J - F) = \{x \in X : F(x) = \lambda J(x)\}$$

containing an unbounded connected set C with $0 \in C$.

In [CFV10] it is defined the *approximate point spectrum of* F *at* p, a subset of $\mathrm{sp}_{CFV}(F)$

$$\mathrm{sp}_{CFVap}(F,p) = \left\{\lambda \in \mathbb{K} : [F]_{A_p} = 0 \text{ and } [F]_{q_p} = 0\right\}$$

that coincides with the usual approximate point spectrum in the linear case.

Combining some interesting summary schemes in [App03, APV00, ADPV04], relating some possible inclusions between the spectra, we present also the Table 6.2 as a possible summary, for $F, J : X \longrightarrow Y$ nonlinear operators and

Table 6.2: Possible inclusions between some nonlinear spectra

$\phi(F,J)$	$\subseteq$	$\mathrm{sp}_{FMV}(F,J)$	$\subseteq$	$\mathrm{sp}_F(F,J)$
$\cup\!\mid$		$\cup\!\mid$		$\cup\!\mid$
$\phi_p(F,J)$	$\subseteq$	$\mathrm{sp}_q(F,J)$		$\mathrm{sp}_p(F,J)$

when reducing to the linear case, $L \in BL(X)$ with $J = I$, we have the Table 6.3.

What seems it should be done, when we want to apply spectral theory to a particular nonlinear problem, is to choose with care a spectrum having at least some of the needed characteristics.

Table 6.3: Possible inclusions of different spectra applied to the linear case

$$
\begin{array}{ccccc}
 & & \mathrm{sp}(L) & & \\
 & & \| & & \\
\phi(L,I) & = & \mathrm{sp}_{FMV}(L,I) & = & \mathrm{sp}_{F}(L,I) \\
\cup| & & \cup| & & \cup| \\
\phi_p(L,I) & \subseteq & \mathrm{sp}_{q}(L,I) & \supseteq & \mathrm{sp}_{p}(L,I)
\end{array}
$$

In [ADPV04] the authors state that it is not the intrinsic structure of the spectrum itself that results in interesting applications, but considered a tool for solving nonlinear equations. It is also referred that in nonlinear analysis, two techniques have been effectively used for the study of nonlinear eigenproblems: *topological methods* (fixed point theorems, degree theory) and *variational methods* (critical points, nonlinear functionals).

The knowledge of the spectrum of a nonlinear operator seems to be useful in solvability results for nonlinear equations, moreover we emphasise that bifurcation theory, one of the most important fields of nonlinear functional analysis, is closely related to nonlinear eigenvalue problems. Other more applications, as e.g. to boundary value problems, may be found in literature.

Nonlinear eigenvalue problems arise in many applications, e.g. structural dynamics, acoustics, fluid mechanics, control theory and quantum physics. Usually the procedure is to discretize the problem (e.g. Galerkin schemes) and to the resulting matrix problem an iterative projection method may be applied .

It can also be held what is called *linearization*, for example, in polynomial or rational eigenvalue problems in which first it is performed a linearization, resulting in a larger linear eigenvalue problem with the same eigenvalues and then apply a method for linear eigenvalue problems. However, in [MV05] the authors state that this is not ideal, since it makes the problem larger and moreover may considerably increase the conditioning of the problem. Even though, this approach seems to be appropriate for this kind of problems.

There is a vast literature on numerical methods for nonlinear eigenvalue problems. Generally, they are split into dense and large sparse problems. For the last ones, projection methods that work directly for general nonlinear eigenproblems, are a good option. In this case the search spaces have to be expanded by directions that have a high approximation potential for the desired eigenvector.

We have seen in Section 1.2 that for sparse linear eigenproblems iterative projection methods, like e.g. Arnoldi or Jacobi-Davidson, are a tool for the computation of a few eigenpairs of the spectrum. As previously observed, Jacobi-Davidson method for linear problems was proposed by Sleijpen and van der Vorst; the method was later extended to quadratic eigenproblems [BV04]. Voss considered also the use of this method for the nonlinear eigenproblem (nonsymmetric case) [Vos07].

Generally two approaches to subspace expansion may be found in the literature: Jacobi-Davidson [BV04, Vos04b] and nonlinear Arnoldi [Vos04a] type expansion. Both methods approximate inverse iteration, which is known to provide a direction of a high approximating potential to the eigenpair it is aiming for.

A more detailed study of these issues may be found in [MV05, Vos04c].

Bibliography

[ABB99] E. Anderson, Z. Bai, and C. Bischof. *LAPACK Users' guide*, volume 9. Society for Industrial Mathematics, 1999.

[ACD+09] E. Ayguadé, N. Copty, A. Duran, J. Hoeflinger, Y. Lin, F. Massaioli, X. Teruel, P. Unnikrishnan, and G. Zhang. The design of OpenMP tasks. *IEEE Transactions on Parallel and Distributed Systems*, 20(3):404–418, 2009.

[AdL+02] M. Ahues, F. D. d'Almeida, A. Largillier, O. Titaud, and P. Vasconcelos. An L^1 refined projection approximate solution of the radiation transfer equation in stellar atmospheres. *Journal of Computational and Applied Mathematics*, 140(1-2):13–26, 2002.

[AdLV06] M. Ahues, F. D. d'Almeida, A. Largillier, and P. B. Vasconcelos. Defect correction for spectral computations for a singular integral operator. *Communications on Pure and Applied Analysis*, 5(2):241–250, 2006.

[ADPV04] J. Appell, E. De Pascale, and A. Vignoli. *Nonlinear spectral theory*, volume 10. Walter De Gruyter Inc, 2004.

[App03] J. Appell. A la recherche du spectre perdu: An invitation to nonlinear spectral theory. *Nonlinear Analysis, Function Spaces and Applications*, 7:1–20, 2003.

[APV00] J. Appell, E. De Pascale, and A. Vignoli. A comparison of different spectra for nonlinear operators. *Nonlinear Analysis*, (40):73–90, 2000.

[AS60] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions*. Dover, New York, USA, 1960.

[Atk97] K.E. Atkinson. *The numerical solution of integral equations of the second kind*, volume 4. Cambridge Univ Pr, 1997.

[BBE+10] Satish Balay, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matt Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.1, Argonne National Laboratory, 2010.

[BCR91] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, 44(2):141–183, 1991.

[BDD+00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[Beb08] M. Bebendorf. *Hierarchical matrices: a means to efficiently solve elliptic boundary value problems*, volume 63 of Lecture Notes in Computational Science and Engineering. Springer Verlag, 2008.

[BG04] S. Börm and L. Grasedyck. Low-rank approximation of integral operators by interpolation. *Computing*, 72(3):325–332, 2004.

[BGH03a] S. Börm, L. Grasedyck, and W. Hackbusch. *Hierarchical Matrices*. Lecture Note 21 of Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2003.

[BGH03b] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405–422, 2003.

[Bor05] S. Borne. Hierarchical matrices for convection-dominated problems. *Domain Decomposition Methods in Science and Engineering*, pages 631–638, 2005.

[Bör09] S. Börm. Construction of data sparse $\mathcal{H}^2$-matrices by hierarchical compression. *SIAM J. Scientific Computing*, 31(3):1820–1839, 2009.

[BV04] T. Betcke and H. Voss. A Jacobi–Davidson-type projection method for nonlinear eigenvalue problems. *Future Generation Computer Systems*, 20(3):363–372, 2004.

[CFV09] A. Calamai, M. Furi, and A. Vignoli. An overview on spectral theory for nonlinear operators. *Communications in Applied Analysis*, 13(4):509–534, 2009.

[CFV10] A. Calamai, M. Furi, and A. Vignoli. A new spectrum for nonlinear operators in Banach spaces. *Arxiv preprint arXiv:1005.1819*, 2010.

[CJVDP07] B. Chapman, G. Jost, and R. Van Der Pas. *Using OpenMP: portable shared memory parallel programming*, volume 10. The MIT Press, 2007.

[dH80] H.C. Van de Hulst. *Multiple light scattering: Tables, Formula, and applications*. Academic Press, NY, 1980.

[DHP02] I.S. Duff, M.A. Heroux, and R. Pozo. An overview of the sparse basic linear algebra subprograms: The new standard from the BLAS technical forum. *ACM Transactions on Mathematical Software (TOMS)*, 28(2):239–267, 2002.

[Dör97] M. Dörfner. *Spektraltheorie für nichtlineare Operatoren*. PhD thesis, Universität Würzburg, 1997.

[dTV05] F. d'Almeida, O. Titaud, and P. B. Vasconcelos. A numerical study of iterative refinement schemes for weakly singular integral equations. *Applied Mathematics Letters*, 18(5):571–576, 2005.

[Fen97] W. Feng. A new spectral theory for nonlinear operators and its applications. In *Abstract and Applied Analysis*, volume 2, pages 163–183. Hindawi Publishing Corporation, 1997.

[FMV78] M. Furi, M. Martelli, and A. Vignoli. Contributions to the spectral theory for nonlinear operators in banach spaces. *Annali di Matematica Pura ed Applicata*, 118(1):229–294, 1978.

[FSVdV98] D.R. Fokkema, G.L.G. Sleijpen, and H.A. Van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. *SIAM journal on scientific computing*, 20(1):94–125, 1998.

[GH03] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[GKLB09] L. Grasedyck, R. Kriemann, and S. Le Borne. Domain decomposition based-LU preconditioning. *Numerische Mathematik*, 112(4):565–600, 2009.

[GVL96] G.H. Golub and C.F. Van Loan. *Matrix computations*, volume 3. Johns Hopkins Univ Pr, 1996.

[Hac90] W. Hackbusch. The panel clustering algorithm. In *MAFELAP*, pages 339–348, 1990.

[Hac99] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[HK00] W. Hackbusch and B. N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. Part II: application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

[HKK04] W. Hackbusch, B.N. Khoromskij, and R. Kriemann. Hierarchical matrices based on a weak admissibility criterion. *Computing*, 73:207–243, 2004.

[HN89] W. Hackbusch and Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numerische Mathematik*, 54(4):463–491, 1989.

[HRTV10] V. Hernandez, J. E. Roman, A. Tomas, and V. Vidal. SLEPc users manual. Technical Report DSIC-II/24/02 - Revision 3.1, D. Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 2010.

[HRV05] V. Hernandez, J. E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Transactions on Mathematical Software*, 31(3):351–362, 2005.

[I.W60] I.W.Busbridge. *The mathematics of radiative transfer*, volume 369. Cambridge University Press, 1960.

[Kac69] R.I. Kachurovskij. Regular points, spectrum and eigenfunctions of nonlinear operators. *Dokl. Akad. Nauk SSSR*, 188:274–277, 1969.

[Kre99] R. Kress. *Linear integral equations*, volume 82 of *Applied mathematical sciences*. Springer Verlag, 1999.

[LHKK79] C.L. Lawson, R.J. Hanson, D.R. Kincaid, and F.T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Transactions on Mathematical Software (TOMS)*, 5(3):308–323, 1979.

[LL93] A. Largillier and M. Levet. A note on a collectively compact approximation for weakly singular integral operators. *Applied Mathematics Letters*, (4):87–90, 1993.

[MDH98] Y.V. Makarov, Z.Y. Dong, and D.J. Hill. A general method for small signal stability analysis. *Power Systems, IEEE Transactions on*, 13(3):979–985, 1998.

[MMH95] Y.V. Makarov, V.A. Maslennikov, and D.J. Hill. Calculation of oscillatory stability margins in the space of power system controlled parameters. *Proceedings Stockholm Power Tech, Stockholm*, pages 416–421, 1995.

[MV05] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. 2005.

[Neu69] JW Neuberger. Existence of a spectrum for nonlinear transformations. *Pacific J. Math*, 31(1):157–159, 1969.

[Par98] B.N. Parlett. *The symmetric eigenvalue problem*, volume 20. Society for Industrial Mathematics, 1998.

[RC04] B. Rutily and L. Chevallier. The finite Laplace transform for solving a weakly singular integral equation occurring in transfer theory. *Journal of Integral Equations and Applications*, 16(4):389–409, 2004.

[Rho77] A. Rhodius. Der numerische wertebereich und die Lösbarkeit linearer und nichtlinearer operatorengleichungen. *Mathematische Nachrichten*, 79(1):343–360, 1977.

[Rut04] B. Rutily. Multiple scattering theory and integral equations. In C. Constanda, M. Ahues, and A. Largillier, editors, *Integral Methods in Science and Engineering*, pages 211–232. Birkhäuser, 2004.

[SBFVdV96] G.L.G. Sleijpen, A.G.L. Booten, D.R. Fokkema, and H.A. Van der Vorst. Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems. *BIT Numerical Mathematics*, 36(3):595–633, 1996.

[SM03] E. Süli and D.F. Mayers. *An introduction to numerical analysis.* Cambridge University Press, 2003.

[Ste01] G. W. Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2001.

[SV00] P. Santucci and M. Väth. On the definition of eigenvalues for nonlinear operators. *Nonlinear Analysis*, (40):565–576, 2000.

[SVdV96] G.L.G. Sleijpen and H.A. Van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 17(2):401–425, 1996.

[SVdV00] G.L.G. Sleijpen and H.A. Van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM Review*, pages 267–293, 2000.

[Tit04] O. Titaud. Reduction of computation in the numerical resolution of a second kind weakly singular Fredholm equation. In C. Constanda, M. Ahues, and A. Largillier, editors, *Integral Methods in Science and Engineering: Analytic and Numerical Techniques*, pages 255–260. Birkhäuser, 2004.

[vdV02] H.A. van der Vorst. Computational methods for large eigenvalue problems. *Handbook of numerical analysis*, 8:3–179, 2002.

[Vos04a] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT Numerical Mathematics*, 44(2):387–401, 2004.

[Vos04b] H. Voss. A Jacobi–Davidson method for nonlinear eigenproblems. *Computational Science-ICCS 2004*, pages 34–41, 2004.

[Vos04c] H. Voss. Numerical methods for sparse nonlinear eigenvalue problems. *Proc. XVth Summer School on Software and Algorithms of Numerical Mathematics, Hejnice, Czech Republic*, 2004.

[Vos07] H. Voss. A Jacobi-Davidson method for nonlinear and nonsymmetric eigenproblems. *Computers & Structures*, 85(17-18):1284–1292, 2007.

[ZLM05] Y. Zhang, X.S. Li, and O. Marques. Towards an automatic and application-based eigensolver selection. *Lawrence Berkeley National Laboratory: Lawrence Berkeley National Laboratory. LBNL Paper LBNL-58949*, 2005.

Appendix A

Research projects

This thesis is a cotutelle project developed within two Universities: the University of Porto (Portugal) and University of Jean Monnet, St. Etienne (France), and during this reasearch period there were some projects approved by different entities:

- The research project of this thesis was supported by the fellowship PROTEC (Programa de apoio à formação avançada de docentes do Ensino Superior Politécnico) provided by FCT (Fundação para a Ciência e a Tecnologia) under the project SFRH/BD/49394/2009.

- Programa de Acções Universitárias Integradas Luso-Francesas (PAUILF) 2011 and 2012, under the project "Aproximação espectral com matrizes e operadores discretizados", F-TCO3/11. The program PAUILF is jointly supported by Conférence des Présidents d'Université (CPU) and Conselho de Reitores das Universidades Portuguesas (CRUP), being designed to initiate academic and scientific exchanges between the two countries.

- Acções de Mobilidade de Investigadores FCT/MICINN for 2011 under the Memorandum of Understanding between the Ministério da Ciência, Tecnologia e Ensino Superior da República Portuguesa (Portugal) and the Ministério da Ciência e Inovação do Reino de Espanha em Matéria

de Física Nuclear, de Partículas, Astropartículas e e-Ciência, with the project "De plataformas tradicionais paralelas para ambientes GPU e computação em cloud - um estudo de caso para cálculo espectral".