Complexity reduction methods applied to the rapid solution to multi-trace boundary integral formulations.

Alan Ayala Obregón

Thèse présentée pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ SORBONNE

Spécialité : Mathématiques appliquées

par

Alan Ayala Obregón

Complexity reduction methods applied to the rapid solution to multi-trace boundary integral formulations

soutenue le 16 novembre 2018 devant le jury composé de :

Bruno Després	Sorbonne Université - LJLL	président du jury
Patrick Ciarlet	ENSTA, ParisTech	rapporteur
François Alouges	Ecole Polytechnique – CMAP	rapporteur
Eric Darrigrand	Université de Rennes 1	examinateur
Laura Grigori	Sorbonne Université - INRIA Paris	directeur de thèse
Xavier Claeys	Sorbonne Université - INRIA Paris	directeur de thèse

Abstract

The objective of this thesis is to provide complexity reduction techniques for the solution of Boundary Integral Equations (BIE). In particular, we focus on BIE arising from the modeling of acoustic and electromagnetic problems via Boundary Element Methods (BEM). Our approach consists in using the local multi-trace formulation which is friendly to operator preconditioning. We find a closed form inverse of the local multi-trace operator for a particular scattering model problem and then we propose this inverse operator for preconditioning general scattering problems. We numerically show that this preconditioner is efficient and accelerates the solution of the linear system obtained from the discretization of the continuous problem. We also show that the local multi-trace formulation is stable for Maxwell equations posed on a particular domain configuration.

For general problems where BEM are applied, we propose to use the framework of hierarchical matrices, which are constructed using cluster trees and allow to represent the original matrix in such a way that submatrices that admit low-rank approximations (admissible blocks) are well identified. We introduce a technique called geometric sampling which uses cluster trees to sample row and column indices allowing to create accurate linear-time CUR algorithms for the compression and matrix-vector product acceleration of admissible matrix blocks, and which are oriented to develop parallel communication-avoiding algorithms.

For the general framework of low-rank approximations, we study widely used techniques based on QR factorizations and subspace iteration methods; for the former we provide new bounds for the classical column pivoting and general pivoting strategies, and for the later we solve an open question in the literature consisting in proving that the approximation of singular vectors exponentially converges. Finally, we propose a technique called affine low-rank approximation intended to increase the accuracy of classical low-rank approximation methods, in particular for those based on QR and subspace iteration.

Résumé

L'objectif de cette thèse est de fournir des techniques de réduction de complexité pour la solution des équations intégrales de frontière (BIE). En particulier, nous sommes intéressés par les BIE issues de la modélisation des problèmes acoustiques et électromagnétiques via la méthode des éléments de frontière (BEM). Nous utilisons la formulation multi-trace locale pour laquelle nous trouvons une expression explicite pour l'inverse de l'opérateur multi-trace pour un problème modèle de diffusion. Ensuite, nous proposons cet inverse pour préconditionner des problèmes de diffusion plus générales. Nous montrons également que la formulation multi-trace locale est stable pour les équations de Maxwell posées sur un domaine particulier.

Pour les problèmes BEM généraux, nous posons le problème dans le cadre des matrices hiérarchiques, pour lesquelles c'est possible d'identifier sous-matrices admettant des approximations de rang faible (blocs admissibles). Nous introduisons une technique appelée échantillonnage géométrique qui utilise des structures d'arbre pour échantillonner des indices de lignes et de colonnes permettant de créer des algorithmes CUR en complexité linéaire, lesquelles sont orientés pour créer des algorithmes parelles avec communication optimale.

Finalement, nous étudions des méthodes QR et itération sur sous-espaces; pour le premier, nous fournissons de nouvelles bornes pour l'erreur d'approximation, et pour le deuxième nous résolvons une question ouverte dans la littérature consistant à prouver que l'approximation des vecteurs singuliers converge exponentiellement. Enfin, nous proposons une technique appelée approximation affine de rang faible destinée à accroître la précision des méthodes classiques d'approximation de rang faible.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Context of this work

Large scale modeling of real-world problems requires formulations and solvers that go hand in hand with the development of computational resources. Among these problems: acoustics, diffraction, fluid dynamics, electromagnetism, weather prediction, seismic modeling, etc. Boundary Integral Equations (BIE) naturally arise in such applications and have been extensively studied both at the theoretical and practical level. Abel (1826) was one of the first persons to formulate and solve an integral equation, which then lead to the search of integral forms of partial differential equations that govern natural phenomena. Nowadays, the challenge is to develop fast and accurate methods to solve these problems via formulations and algorithms that can be efficiently implemented in large scale computer clusters.

Since most integral equations can not be solved explicitly, BIE are, in general, solved numerically by using a standard approach known as Boundary Element Methods (BEM), which requires a so called integral formulation. Classical formulations for BEM are the electric field integral equation (EFIE), the magnetic field integral equation (MFIE), the combined field integral equation (CFIE), the PMCHW (Poggio, Miller, Chang, Harrington, and Wu) formulation, among others, refer to [JSC02] for a survey. The first part of this thesis is devoted to a formulation for wave scattering problems in acoustics and electromagnetism. Albeit we focus on these particular fields, the presented theory is amenable to be extended to other fields. Wave scattering problems can be solved using the previously mentioned formulations, by reducing the continuous problem to a discrete one represented by a linear system, which is in general ill-conditioned and precondition techniques are necessary. Classical formulations such as the EFIE struggle to admit operator preconditioning when dealing with several scattering domains. For such case, Multi-trace formulations (MTF) [CH11, HJH12] are alternative approaches that

are friendly to operator preconditioning and allow to have a linear system that essentially maintain a constant condition number while refining the problem mesh. The idea behind MTF is the use of domain decompositions techniques which make them amenable for large scale computational models.

Once we reduce a BIE problem to the solution of a linear system, the classical approach is to use iterative solvers such as the Generalized Minimal Residual Method (GMRES), Conjugate Gradient (CG), Bi-Conjugate Gradient (BCG), among others that can be found in [QSS06]. To accelerate such solvers we need a fast way to compute matrix-vector products which can be achieved by classical techniques like the Fast Multipole Method FMM [GR87, Kou95] or the more recent Kernel Independent and Black Box FMM (resp. KIFMM and BBFMM) [MR07, FD09]. Another strategy is to use hierarchical matrices [Beb08, Bör10, Hac15], which partition the matrix associated to the linear system in blocks that admit a low-rank approximation and those who do not. And then approximate low-rank blocks using algorithms such as the Adaptive Cross Approximation (ACA) [Beb00] or its variants, *e.g.* ACA+ [Gra13].

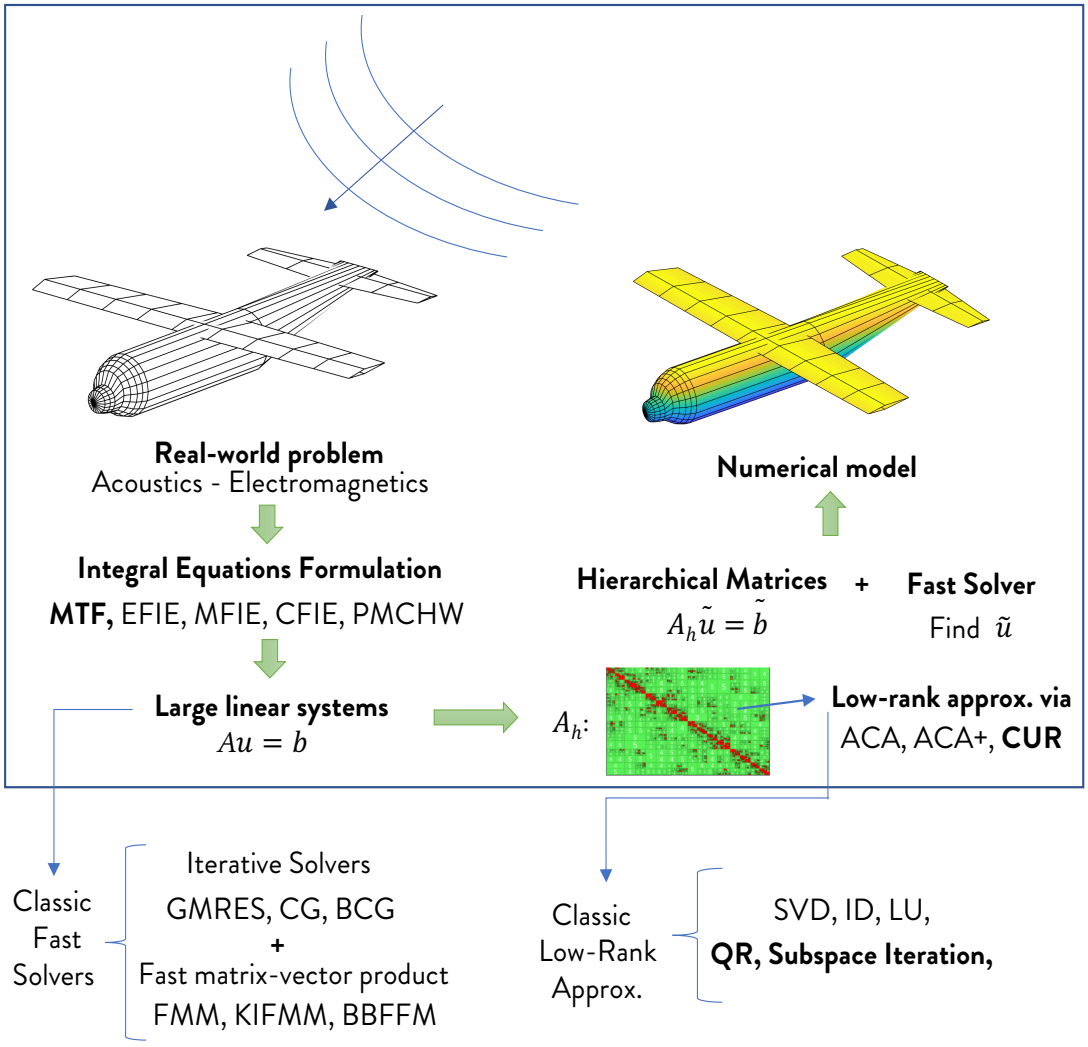


Figure 1.1: Integral approach for the solution of acoustics and electromagnetic problems (enclosed in the box) with classical approaches. In bold text we highlight the methods (formulations and low-rank approximations) that we shall use for the development of this thesis.

In the second part of this thesis, we focus on techniques for low-rank matrix approximations. We start with a general framework for rectangular matrices, and then we analyze the particular case of matrices arising from BEM discretization. Classical low-rank approximations can be constructed via: the Singular Value Decomposition (SVD), QR or LU factorizations, Interpolative Decompositions (ID), Subspace Iteration, CUR decompositions, among others, refer to [KG17] for a survey. The scope of this thesis is summarized in Figure 1.1, which shows classical approaches to handle wave scattering problems (enclosed in a box), and highlights the methodologies that we shall use later on: integral MTF, hierarchical matrices, CUR decompositions, and general low-rank matrix approximations via QR factorizations and subspace iteration.

1.2 Multi-Trace formulations

As mentioned earlier, MTF are formulations that admit efficient operator preconditioning, for a survey on MTF refer to [CHJH13, CHJHP15]. These formulations come in two flavors based on different ideas. To give a simple and comprehensible introduction to MTF, let us consider a scattering problem in acoustics,

$$-\Delta \mathbf{U} - \kappa_i^2 \mathbf{U} = 0, \quad \text{(Helmholtz equation)},$$

or in electromagnetism,

$$-\nabla \times \nabla \times \mathbf{E} - \kappa_i^2 \mathbf{E} = 0, \quad \text{(Maxwell equation)},$$

where we search solutions for $\mathbf{U}$ (resp. $\mathbf{E}$). Let the problem be posed on a domain configuration as shown in Figure 1.2, where Ω_i is a Lipschitz domain and Γ_{ij} refers to the intersection of the boundaries of Ω_i and Ω_j, for $i, j = 0, 1, 2$. The incident wave is given as U_{inc}.

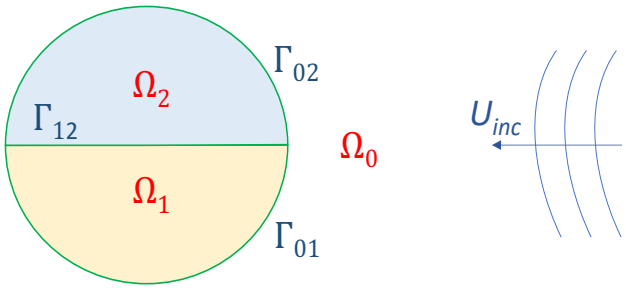


Figure 1.2: Wave scattering model problem

The connection between these domains is stablished by transmission conditions. For acoustics, they consist in continuity of the solution and normal continuity of the gradient. For electromagnetism, tangential continuity of the electric and magnetic fields, *c.f.* chapter 3. By supplementing those conditions with a radiation condition at infinity, we can take for granted existence and uniqueness of solutions in the week sense, see *e.g.* [Pet89, CK13].

In general, integral formulations are oriented to find *trace functions* which are posed on the boundaries Γ_{ij}, classical methods search for two trace functions known as *Dirichlet* and *Neumann* traces. Once these trace functions are obtained, using representation formulae, the solutions for $\mathbf{U}$ and $\mathbf{E}$ can be computed in the whole volume, see *e.g.* [Ste08].

- Global-MTF: Introduced in [CH11], this formulation is based on a so called gap idea, consisting in tearing apart the scattering domains by a small separation $\delta > 0$, see Figure 1.3. Then, proceed to write a classical formulation considering the new interfaces configuration and then make $\delta \longrightarrow 0$.

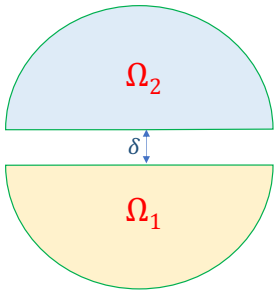


Figure 1.3: Gap Idea of global MTF

  For the global-MTF framework, we get four unknown trace functions on the middle interface Γ_{12}, two coming from the bottom and two coming from the top (instead of two as in the case of classical formulations); hence, they are called multi-trace. Global-MTF is robust for both Helmholtz and Maxwell equations, and admits operator preconditioner (indeed, the global-MTF operator preconditions itself) allowing to deal with mesh refinement without making the condition number of the discrete system to blow up. As a drawback, the global-MTF needs to perform the discretization of remote coupling operators which are non-local and when discretized yield to dense (compressible) matrices.

- Local-MTF: Introduced in [HJH12], it is based on local use of transmission conditions. For each domain Ω_i two traces are obtained and the operator counts the contributions of the others domains by using transmission operators which map traces on $\partial\Omega_j$ into traces on $\partial\Omega_i$. These operators are purely local and, when discretized, yield to sparse matrices. Local-MTF admits a simple and robust preconditioning technique. As a drawback, its stability for Maxwell equation has not yet been proved. We devote the following two chapters to analyze this formulation and make contributions to its developments.

1.3 Matrix-compression and low-rank approximations

In the second part of this thesis, we are interested at first in low-rank approximations for a general rectangular matrix $A \in \mathbb{C}^{m\times n}$, and then we study the particular case where A arises from BEM discretization. Let $A_k \in \mathbb{C}^{m\times n}$ be the rank-k matrix that minimizes the approximation error in the spectral norm, A_k can be obtained by the truncated SVD, [EG36, Mir60], and it holds that $\|A - A_k\|_2 = \sigma_{k+1}$, where σ_{k+1} is the $k+1$ singular value of A, and $\|\cdot\|_2$ is the spectral matrix norm. Computing the truncated SVD is considered expensive in practice and fast techniques are commonly applied to search for good rank-k approximations.

Several low-rank approximations are linked to the *Column Subset Selection Problem* (CSSP), which consists in finding a set of k columns of A given by an index vector J, such that the low-rank matrix ξ_k, obtained by projecting the columns of A onto the space generated by the selected columns, approximates A with a minimal error. For a given choice of J, we have

$$A \approx \xi_k := CC^{\dagger}A, \qquad \|A - \xi_k\|_2 \leq f(k, m, n)\sigma_{k+1}, \tag{1.3.1}$$

where f is a small degree polynomial, $C := A(:, J) \in \mathbb{C}^{m\times k}$ is the matrix formed by the J selected columns, and $C^{\dagger}$ is its classical Moore-Penrose pseudoinverse. Note that we can find the solution of the CSSP by analyzing $\binom{n}{k}$ possible choices of J. However, it would cost $\mathcal{O}(n^k)$ which is prohibitive in practice, and finding such solution is known to be NP-hard [ÇMI13]. Polynomial-cost methods for CSSP are extensively studied in the literature [DRVG06, DV06, BMD09] and there even exists algorithms that find suboptimal approximations in polynomial time [DR10].

Pivoted QR factorization techniques can be efficiently used to approximate the solution of the CSSP [CGMR05], providing efficient low-rank approximations that can even be proved to be suboptimal [GE96]. Low-rank QR based approximations will play a fundamental role in the development of the second part of this manuscript.

Another type of factorization, closely linked to the CSSP, is the CUR low-rank approximation [MD09, DR10, WZ13, VM17], which consists in finding J via an approximation of the CSSP, and then proceed to find a row index vector I of size k, selected such that the submatrix of C with row indices given by I, and denoted by C(I, :), is non-singular. Then A is approximated as

$$A \approx C \cdot U \cdot R,$$

where $C = A(:, J)$, $R := A(I, :) \in \mathbb{C}^{k\times n}$ and $U := C^{-1}(I, :) \in \mathbb{C}^{k\times k}$. When A is a matrix arising from BEM discretization, CUR methods are also known as skeleton approximations [GZT97, Beb00, GT01, GOS+08] and we devote an entire chapter to its numerical and algorithmic analysis.

To conclude, another kind of approximation that we shall use in this thesis, can be obtained via subspace iteration methods, *c.f.* §4.4, which consist in constructing a low-rank matrix by approximating the column space of $Y := (AA^T)^q A\Omega$, where q is a small integer parameter and $\Omega \in \mathbb{R}^{n\times l}$ is a random matrix. We get $A \approx QB_k$, where $Q \in \mathbb{C}^{m\times k}$ is the orthogonal basis of Y and B_k is the rank-k truncated SVD of $Q^T A$, see *e.g.* [Gu15, HMT11]. The bound on the error holds with high probability [HMT11]

$$\mathbb{E}\|A - QB_k\|_2 \leq \left(1 + \left(1 + 4\sqrt{\frac{2\min(m,n)}{k-1}}\right)^{1/(2q+1)}\right)\sigma_{k+1}.$$

1.4 Summary and Contributions

This manuscript is structured in two parts that can be read independently. The first part deals with the solution of boundary integral equations arising from acoustics and electromagnetics problem, for which we use the multi-trace formulation. Then, in the second part of the thesis we propose methods of low-rank compression and approximation of general matrices for accelerating the solution of linear systems arising from BEM discretizations.

Part I

- In **Chapter 2**, we apply the local multi-trace formulation for acoustic scattering problems. We consider a model problem where all scattering domains are composed with an unique homogenous material and then we find a closed form for the inverse of the local-MTF operator corresponding to this configuration. We then use this inverse operator for preconditioning general composite scattering problems.

- In **Chapter 3**, we analyze the stability of the local-MTF for the case of Maxwell equation, which was an open question in the literature of MTF. We prove the injectivity of the local Multi-Trace operator and then a generalized Gårding inequality for the local-MTF formulation on the unit sphere.

Part II

- In **Chapter 4**, we recall classical low-rank approximations for general matrices, focusing in particular on QR and subspace iteration methods. For the former we provide a new bound on the error when the classical column pivoting technique is used; and furthermore, we prove a bound for the case when a general pivoting technique is used. For the latter, we prove exponential convergence on the approximation of singular vectors, which was an open question in the literature.
- **In Chapter 5**, we introduce the concept of affine low-rank approximations providing an algorithm called ALORA that is intended to modify standard approximation algorithms. We then introduce a correlation coefficient to relate the spacial distribution of the columns of a matrix to its low-rank structure, which helps to understand for which matrices an affine low-rank approximation would be advantageous. Finally, we compare the performance of affine low rank-approximation with respect to standard QR and subspace iteration algorithms.
- **In Chapter 6**, we introduce the novel concept of geometric sampling to approximate matrices arising from BEM discretization, to which we refer as to BEM matrices. We provide a framework to constuct linear-time CUR approximations using information from the geometry where the problem is posed. We prove a general bound on the approximation error and provide a CUR algorithm that performs very well in practice using a criterion called gravity centers sampling.

In Appendix A, we present an independent work, albeit related to the low-rank analysis performed in Chapter 4. It consists in a communication avoiding low-rank QR approximation algorithm developed during the first year of this thesis. Note that this work has not received any further development for two years. However, in a future work, this contribution could be optimized for particular applications to accelerate and increase the precision of matrix-compression and low-rank approximations.

Appendix B contains additional theoretical results and MATLAB codes corresponding to the work developed in the second part of this Thesis.

Part I

Muti-trace formulations

CHAPTER 2

Local Multi-Trace formulation

2.1 Preliminaries

The local multi-trace formulation (later abbreviated local-MTF) was introduced in [HJH12] as a means to solve acoustic wave scattering by heterogenous penetrable structures as those found in composite materials. The MTF considers as unknowns Dirichlet and Neumann traces on either side of the subdomains. These traces are then required to satisfy Calderón identities per subdomain and transmission conditions per interface. This last condition forces the appearance of restriction and extension-by-zero operators which entail Petrov-Galerkin variational forms: trial and test functions belong to different functional spaces, $\mathrm{H}^{\pm 1/2}(\Gamma)$ and $\widetilde{\mathrm{H}}^{\mp 1/2}(\Gamma)$, respectively. Consequently, a mismatch between continuity and coercivity spaces takes place and, consequently, the Fredholm alternative argument cannot be used directly. Still, by using Lion's lemma and by proving uniqueness of solutions, it is shown that the formulation is well-posed [HJH12, CHJH13].

For the discretization of local-MTF, one requires a slight increase in regularity and inverse discrete inequalities such as those presented in the original work for 2D (*cf.* [HJHM15] for 3D) to derive discrete stability estimates. However, numerically the method has been successfully shown to be easy to implement with standard codes, with clear parallelization and, though ill-conditioned, obvious preconditioners of algebraic or Calderón type. In recent years, a great deal of work has been devoted to either extend local-MTF [JHPT15], find alternative formulations [CHJH13, CHJHP15], find its connection domain decomposition methods [HJHLP14, DG16, JHPAT17] while with applications even in cellular simulation [HJHA16].

In this chapter we present the local multi-trace formulation, introducing main notations and back-

ground for the theory presented in this and the following chapter. Then, we provide a closed form for the inverse of the local multi-trace operator of a model transmission problem, we posit that this inverse operator can be taken as a preconditioner for general local-MTF of composite scattering in acoustics. The chapter is structured as follows, in Section 2.2 we present technical concepts needed for our heavy analysis later on. Section 2.3 presents the local multi-trace operator and in Section 2.4 we derive its inverse for a model problem. Next, in Section 2.5 we numerically verify the theoretical analysis and show the efficiency of the obtained preconditioner. Finally, Section 2.6 concludes the chapter.

2.2 Functional and trace spaces

Let us consider a partition of the d dimensional space $\mathbb{R}^d := \bigcup_{j=0}^{n} \overline{\Omega}_j$ where each Ω_j is a connected Lipschitz domain. We refer $\Gamma_j := \partial\Omega_j$ as the boundary of Ω_j.

Next, let $\mathrm{L}^2(\Omega)$ be the functional space given by the square integrable functions, and define the following functional spaces,

$$\mathrm{H}^1(\Omega_j) := \left\{ v \in \mathrm{L}^2(\Omega_j) \quad \middle| \quad \|v\|^2_{\mathrm{H}^1(\Omega_j)} := \int_{\Omega_j} |v|^2 + |\nabla v|^2 < +\infty \right\}, \tag{2.2.1}$$

$$\mathrm{H}(\mathrm{div}, \Omega_j) := \left\{ v \in \mathrm{L}^2(\Omega_j) \quad \middle| \quad \|v\|^2_{\mathrm{H}(\Omega_j)} := \int_{\Omega_j} |q|^2 + |\mathrm{div}(q)|^2 < +\infty \right\}, \tag{2.2.2}$$

$$\mathrm{H}(\mathbf{curl}, \Omega_j) := \left\{ v \in \mathrm{L}^2(\Omega_j) \quad \middle| \quad \|v\|^2_{\mathrm{H}(\mathbf{curl},\Omega_j)} := \|v\|_{\mathrm{L}^2(\Omega_j)} + \|\mathbf{curl} v\|_{\mathrm{L}^2(\Omega_j)} < +\infty \right\}. \tag{2.2.3}$$

If $\mathrm{H}(\Omega_j)$ is one of the functional spaces defined above, we define

$$\mathrm{H}_{\mathrm{loc}}(\overline{\Omega}_j) := \left\{ v \quad | \quad \varphi v \in \mathrm{H}(\Omega_j), \quad \forall \varphi \in \mathscr{D}(\mathbb{R}^d) \right\},$$

where $\mathscr{D}(\mathbb{R}^d)$ is the space of functions of class $\mathscr{C}^\infty$ having compact support.

2.2.1 Trace spaces

The space of Dirichlet traces is given as

$$\mathrm{H}^{1/2}(\Gamma_j) := \left\{ v|_{\Gamma_j} \quad \middle| \quad v \in \mathrm{H}^1(\Omega_j) \right\},$$

equipped with the norm

$$\|v\|_{\mathrm{H}^{1/2}(\Gamma_j)} := \min \left\{ \|u\|_{\mathrm{H}^1(\Omega_j)} \quad \middle| \quad u \in \mathrm{H}^1(\Omega_j),\ u|_{\Gamma_j} = v \right\},$$

and the space of Neumann traces, $\mathrm{H}^{-1/2}(\Gamma_j)$, is defined as the dual to $\mathrm{H}^{1/2}(\Gamma_j)$ and is equipped with the corresponding canonical dual norm

$$\|p\|_{\mathrm{H}^{-1/2}(\Gamma_j)} := \sup_{v \in \mathrm{H}^{1/2}(\Gamma_j)\setminus\{0\}} \frac{|\langle p, v\rangle|}{\|v\|_{\mathrm{H}^{1/2}(\Gamma_j)}}.$$

When dealing with Maxwell equation in the next chapter, letting $\Omega \subset \mathbb{R}^3$ be a connected Lipschitz domain with $\Gamma = \partial\Omega$, we shall refer to $\mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)$ as the tangential traces of volume based vector fields belonging to $\mathrm{H}(\mathbf{curl}, \mathbb{R}^3)$,

$$\mathrm{H}^{-1/2}(\mathrm{div},\Gamma) := \left\{ u|_\Gamma \times n_j \quad \middle| \quad u \in \mathrm{H}(\mathbf{curl},\mathbb{R}^3) \right\},$$

note that this definition does not depend on the choice of the normal n_j. This space is put in duality with itself by means of one of the bilinear forms $u, v \mapsto \int_\Gamma n_j \cdot (u \times v) d\sigma$. We also need to introduce duality pairings for $\mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2 = \mathrm{H}^{-1/2}(\mathrm{div},\Gamma) \times \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)$ that is defined by

$$[(u,p),(v,q)]_{\Gamma_j} := \int_{\Gamma_j} n_j \cdot (u \times q + p \times v)\, d\sigma.$$

Multi-trace formulations will be written in a so-called multi-trace space and obtained as the cartesian product of traces on the boundary of each subdomain. It takes the simple form

$$\mathbb{H}(\Sigma) := \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2 \times \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2 = \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^4.$$

This space will be equipped with a bilinear pairing $[\![\cdot,\cdot]\!] : \mathbb{H}(\Sigma) \times \mathbb{H}(\Sigma) \longrightarrow \mathbb{C}$ defined as follows. For any tuples $u = (u_0, u_1), v = (v_0, v_1) \in \mathbb{H}(\Sigma)$ we set

$$[\![u, v]\!] := [u_0, v_0]_{\Gamma_0} + [u_1, v_1]_{\Gamma_1}.$$

Note the identity $[\![u, v]\!] = -[\![v, u]\!]$ for any $u, v \in \mathbb{H}(\Sigma)$. Next we need to introduce three interior trace operators $\gamma_{\mathrm{T}}^j, \gamma_{\mathrm{R}}^j : \mathrm{H}(\mathbf{curl},\overline{\Omega}_j) \longrightarrow \mathrm{H}^{-1/2}(\mathrm{div},\Gamma_j)$ and $\gamma^j : \mathrm{H}(\mathbf{curl},\overline{\Omega}_j) \longrightarrow \mathrm{H}^{-1/2}(\mathrm{div},\Gamma_j)^2$, those traces are taken from the *interior* of Ω_j and defined for all $u \in \mathscr{C}^\infty(\mathbb{R}^3)^3$ by

$$\begin{aligned} \gamma_{\mathrm{T}}^j(u) &:= u|_\Gamma \times n_j, \\ \gamma_{\mathrm{R}}^j(u) &:= \mathbf{curl}(u)|_\Gamma \times n_j, \\ \gamma^j(u) &:= (\gamma_{\mathrm{T}}^j(u), \gamma_{\mathrm{R}}^j(u)). \end{aligned} \tag{2.2.4}$$

The trace operators $\gamma_{\mathrm{T},c}^j$ (resp. $\gamma_{\mathrm{R},c}^j$, γ_c^j) shall refer to exactly the same operators as (2.2.4) but with traces taken from the exterior (with the same direction of normal vector n_j though). Then we shall define jump and averages traces as

$$\begin{aligned} \{\gamma_\star^j\}(u) &:= (\gamma_\star^j(u) + \gamma_{\star,c}^j(u))/2, \\ [\gamma_\star^j](u) &:= \gamma_\star^j(u) - \gamma_{\star,c}^j(u) \quad \text{for } \star = \mathrm{T}, \mathrm{R}. \end{aligned} \tag{2.2.5}$$

We define $\{\gamma^j\}$ and $[\gamma^j]$ accordingly.

2.3 Local Multi-Trace operator

We start by introducing the local multi-trace formulation for a model problem. Consider a partition of the space $\mathbb{R}^d = \overline{\Omega}_0 \cup \overline{\Omega}_1 \cup \overline{\Omega}_2$ as shown in Figure 2.1.

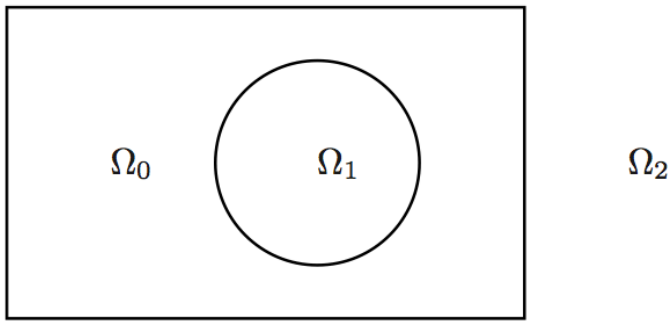


Figure 2.1: Geometrical configuration we consider in the analysis

We assume that Ω_j, $j = 0, 1, 2$ are Lipschitz domains such that $\Omega_j \cap \Omega_k = \emptyset$ for $j \neq k$. Denoting by $\Gamma_j := \partial\Omega_j$, we assume in addition that $\Gamma_1 \cap \Gamma_2 = \emptyset$ and $\Gamma_0 = \Gamma_1 \cup \Gamma_2$. Let n_j be the unit outer normal for Ω_j on its boundary Γ_j. For a sufficiently regular function v we denote by $v|^+_{\Gamma_j}$ the trace of v and by $\partial_{n_j} v|^+_{\Gamma_j}$ the trace of $n_j \cdot \nabla v$ on Γ_j taken from inside of Ω_j. Similarly we define $v|^-_{\Gamma_j}$ and $\partial_{n_j} v|^-_{\Gamma_j}$ but with traces from outside of Ω_j.

The elliptic transmission problem for which we want to study the local multi-trace formulation and its inverse is: find $u \in H^1(\mathbb{R}^d)$ such that

$$\begin{aligned} &-\Delta u + \kappa_j^2 u = 0 \quad \text{in } \Omega_j, \quad j = 0, 1, 2, \\ &[u]_{\Gamma_1} = g_1, \quad [u]_{\Gamma_2} = g_2, \\ &[\partial_n u]_{\Gamma_1} = h_1, \quad [\partial_n u]_{\Gamma_2} = h_2, \end{aligned} \tag{2.3.1}$$

where $\kappa_j > 0$ for $j = 0, 1, 2$, $g_j \in H^{+1/2}(\Gamma_j)$ and $h_j \in H^{-1/2}(\Gamma_j)$ are given data of the transmission problem, and we used the classical jump notation for the Dirichlet and Neumann traces of the solution across the interfaces Γ_j, $j = 1, 2$, i.e. $[u]_{\Gamma_j} := u|^+_{\Gamma_j} - u|^-_{\Gamma_j}$ and $[\partial_n u]_{\Gamma_j} := \partial_{n_j} u|^+_{\Gamma_j} - \partial_{n_j} u|^-_{\Gamma_j}$.

Following [HJHLP14], this problem can be rewritten as a boundary integral local multi-trace formulation, using the Calderón projector: let $\mathbb{H}(\Gamma_j) := H^{1/2}(\Gamma_j) \times H^{-1/2}(\Gamma_j)$; then for $(g, h) \in \mathbb{H}(\Gamma_j)$, the Calderón projector $\mathbb{P}_j : \mathbb{H}(\Gamma_j) \longrightarrow \mathbb{H}(\Gamma_j)$ interior to Ω_j associated to the operator $-\Delta + \kappa_j^2$ is defined by

$$\begin{aligned} &\mathbb{P}_j(g, h) := (v|^+_{\Gamma_j}, \partial_{n_j} v|^+_{\Gamma_j}) \text{ where } v \text{ satisfies} \\ &-\Delta v + \kappa_j^2 v = 0 \quad \text{in } \Omega_j \text{ and in } \mathbb{R}^d \setminus \overline{\Omega}_j, \\ &[v]_{\Gamma_j} = g \quad \text{and} \quad [\partial_n v]_{\Gamma_j} = h, \text{ and} \\ &\limsup_{|x| \to \infty} |v(x)| < +\infty, \end{aligned}$$

and $\mathbb{P}_j$ is known to be a continuous map, see [SS11]. The decomposition $\Gamma_0 = \Gamma_1 \cup \Gamma_2$ induces a natural decomposition of $\mathbb{P}_0$ in the following manner: for any $U \in \mathbb{H}(\Gamma_0)$ set $\rho_j(U) := U|_{\Gamma_j} \in \mathbb{H}(\Gamma_j)$, $j = 1, 2$. In addition, for any $V \in \mathbb{H}(\Gamma_j)$, $j = 1, 2$, define $\rho_j^*(V) \in \mathbb{H}(\Gamma_0)$ by $\rho_j^*(V) = V$ on Γ_j and $\rho_j^*(V) = 0$ on $\Gamma_0 \setminus \Gamma_j$. Then the projector $\mathbb{P}_0$ can be decomposed as

$$\mathbb{P}_0 = \begin{bmatrix} \tilde{\mathbb{P}}_1 & R_{1,2}/2 \\ R_{2,1}/2 & \tilde{\mathbb{P}}_2 \end{bmatrix}, \quad \text{where} \quad \begin{cases} \tilde{\mathbb{P}}_j := \rho_j \cdot \mathbb{P}_0 \cdot \rho_j^*, \\ R_{j,k}/2 := \rho_j \cdot \mathbb{P}_0 \cdot \rho_k^*. \end{cases}$$

The operators $\tilde{\mathbb{P}}_j : \mathbb{H}(\Gamma_j) \longrightarrow \mathbb{H}(\Gamma_j)$ and $R_{j,k} : \mathbb{H}(\Gamma_k) \longrightarrow \mathbb{H}(\Gamma_j)$ are continuous. Following this decomposition, we identify $\mathbb{H}(\Gamma_0)$ with $\mathbb{H}(\Gamma_1) \times \mathbb{H}(\Gamma_2)$. We also introduce the sign switching operator

$\mathrm{X}(v,q) := (v,-q)$, and a relaxation parameter $\sigma \in \mathbb{C}\backslash\{0\}$. The local multi-trace formulation of problem (2.3.1) is then: find $(\mathrm{U}_1, \mathrm{U}_1^{(0)}, \mathrm{U}_2^{(0)}, \mathrm{U}_2) \in \mathbb{H}(\Gamma_1)^2 \times \mathbb{H}(\Gamma_2)^2$ such that

$$\begin{bmatrix} (1+\sigma)\mathrm{Id} - \mathbb{P}_1 & -\sigma\mathrm{X} & 0 & 0 \\ -\sigma\mathrm{X} & (1+\sigma)\mathrm{Id} - \tilde{\mathbb{P}}_1 & -\mathrm{R}_{1,2}/2 & 0 \\ 0 & -\mathrm{R}_{2,1}/2 & (1+\sigma)\mathrm{Id} - \tilde{\mathbb{P}}_2 & -\sigma\mathrm{X} \\ 0 & 0 & -\sigma\mathrm{X} & (1+\sigma)\mathrm{Id} - \mathbb{P}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathrm{U}_1 \\ \mathrm{U}_1^{(0)} \\ \mathrm{U}_2^{(0)} \\ \mathrm{U}_2 \end{bmatrix} = \mathrm{F}, \tag{2.3.2}$$

where $\mathrm{F} \in \mathbb{H}(\Gamma_1)^2 \times \mathbb{H}(\Gamma_2)^2$ is some right-hand side depending on g_j, h_j, σ whose precise expression is not important for our present study, where we want to obtain an explicit expression for the operator in (2.3.2) and its inverse for the special case

$$\kappa_0 = \kappa_1 = \kappa_2. \tag{2.3.3}$$

To simplify the calculations when working with the entries of the operator in (2.3.2), we set $\mathrm{A}_j := -\mathrm{Id} + 2\mathbb{P}_j$ and $\tilde{\mathrm{A}}_j := -\mathrm{Id} + 2\tilde{\mathbb{P}}_j$. The following remarkable identities were established in [CDG18, §4.4] for the special case (2.3.3): $\mathbb{P}_j^2 = \mathbb{P}_j$, $\tilde{\mathbb{P}}_j^2 = \tilde{\mathbb{P}}_j$, $\tilde{\mathbb{P}}_1\mathrm{R}_{1,2} = \tilde{\mathbb{P}}_2\mathrm{R}_{2,1} = 0$, $\mathrm{X}\mathbb{P}_j\mathrm{X} = \mathrm{Id} - \tilde{\mathbb{P}}_j$, and finally $\mathrm{R}_{1,2}\mathrm{R}_{2,1} = \mathrm{R}_{2,1}\mathrm{R}_{1,2} = 0$. These five properties can be reformulated in terms of the operators A_j, namely

$$\begin{aligned} &i) \quad \mathrm{A}_j^2 = \tilde{\mathrm{A}}_j^2 = \mathrm{Id}, \\ &ii) \quad \tilde{\mathrm{A}}_1\mathrm{R}_{1,2} = -\mathrm{R}_{1,2} \text{ and } \tilde{\mathrm{A}}_2\mathrm{R}_{2,1} = -\mathrm{R}_{2,1}, \\ &iii) \quad \mathrm{X}\cdot\mathrm{A}_j\cdot\mathrm{X} = -\tilde{\mathrm{A}}_j, \\ &iv) \quad \mathrm{R}_{1,2}\mathrm{R}_{2,1} = \mathrm{R}_{2,1}\mathrm{R}_{1,2} = 0, \\ &v) \quad \mathrm{R}_{1,2}\tilde{\mathrm{A}}_2 = \mathrm{R}_{1,2} \text{ and } \mathrm{R}_{2,1}\tilde{\mathrm{A}}_1 = \mathrm{R}_{2,1}. \end{aligned} \tag{2.3.4}$$

Let us introduce auxiliary operators $\mathbb{A}, \Pi : \mathbb{H}(\Gamma_1)^2 \times \mathbb{H}(\Gamma_2)^2$ defined by

$$\mathbb{A} := \begin{bmatrix} \mathrm{A}_1 & 0 & 0 & 0 \\ 0 & \tilde{\mathrm{A}}_1 & \mathrm{R}_{1,2} & 0 \\ 0 & \mathrm{R}_{2,1} & \tilde{\mathrm{A}}_2 & 0 \\ 0 & 0 & 0 & \mathrm{A}_2 \end{bmatrix}, \qquad \Pi := \begin{bmatrix} 0 & \mathrm{X} & 0 & 0 \\ \mathrm{X} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathrm{X} \\ 0 & 0 & \mathrm{X} & 0 \end{bmatrix}. \tag{2.3.5}$$

According to property *i)* in (2.3.4), we have $(\mathrm{Id} + \mathbb{A})^2/4 = (\mathrm{Id} + \mathbb{A})/2$, which implies the well known Calderón identity from the boundary integral equation literature, i.e.

$$\mathbb{A}^2 = \mathrm{Id}, \tag{2.3.6}$$

see for example [Néd01, §4.4]. The local multi-trace operator on the left-hand side of Equation (2.3.2) can then be rewritten as

$$\mathrm{MTF}_{\mathrm{loc}} := -\frac{1}{2}\mathbb{A} - \sigma\Pi + (\sigma + \frac{1}{2})\mathrm{Id}. \tag{2.3.7}$$

In (2.3.2), the terms associated with the relaxation parameter σ, namely $\mathrm{Id} - \Pi$, enforce the transmission conditions of problem (2.3.1). For $\sigma = 0$, we have $\mathrm{MTF}_{\mathrm{loc}} = \frac{1}{2}(\mathrm{Id} - \mathbb{A})$, which is a projector, and $\mathrm{MTF}_{\mathrm{loc}}$ is thus not invertible. For $\sigma \neq 0$ however, $\mathrm{MTF}_{\mathrm{loc}}$ was proved to be invertible in [Cla16, Cor. 6.3]. The goal of this chapter is to derive an explicit formula for the inverse of $\mathrm{MTF}_{\mathrm{loc}}$, and we will thus assume $\sigma \neq 0$.

2.4 Inverse of the Local Multi-Trace Operator

We now derive a closed form inverse of the local multi-trace operator in (2.3.7) for the special case (2.3.3). Using that $\Pi^2 = \mathrm{Id}$ and (2.3.6), we obtain

$$\begin{aligned}
&[\,-\mathbb{A}/2 - \sigma\Pi + (\sigma+1/2)\mathrm{Id}\,]\,[\,-\mathbb{A}/2 - \sigma\Pi - (\sigma+1/2)\mathrm{Id}\,] \\
&\qquad = (\mathbb{A}/2 + \sigma\Pi)^2 - (\sigma+1/2)^2\,\mathrm{Id} \\
&\qquad = (\sigma^2 + 1/4 - \sigma^2 - \sigma - 1/4)\mathrm{Id} + \sigma(\mathbb{A}\Pi + \Pi\mathbb{A})/2 \\
&\qquad = -\sigma\mathrm{Id} + \sigma(\mathbb{A}\Pi + \Pi\mathbb{A})/2.
\end{aligned} \tag{2.4.1}$$

Inspired by the calculations in [CDG18, §4.4] as well as [Cla16, Prop. 6.1], we examine more closely $\mathbb{A}\Pi + \Pi\mathbb{A}$. We start by comparing $\mathbb{A}\Pi$ and $\Pi\mathbb{A}$:

$$\mathbb{A}\Pi = \begin{bmatrix} 0 & A_1X & 0 & 0 \\ \tilde{A}_1X & 0 & 0 & R_{1,2}X \\ R_{2,1}X & 0 & 0 & \tilde{A}_2X \\ 0 & 0 & A_2X & 0 \end{bmatrix}, \quad \Pi\mathbb{A} = \begin{bmatrix} 0 & X\tilde{A}_1 & XR_{1,2} & 0 \\ XA_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & XA_2 \\ 0 & XR_{2,1} & X\tilde{A}_2 & 0 \end{bmatrix}. \tag{2.4.2}$$

According to Property *iii)* in (2.3.4), we have $X\tilde{A}_j + A_jX = 0$ and $XA_j + \tilde{A}_jX = 0$, and thus from (2.4.2) we obtain

$$\Pi\mathbb{A} + \mathbb{A}\Pi = \begin{bmatrix} 0 & 0 & XR_{1,2} & 0 \\ 0 & 0 & 0 & R_{1,2}X \\ R_{2,1}X & 0 & 0 & 0 \\ 0 & XR_{2,1} & 0 & 0 \end{bmatrix}.$$

Computing the square of this operator, and taking into account Property *iv*) from (2.3.4), we obtain

$$(\Pi\mathbb{A} + \mathbb{A}\Pi)^2 = \begin{bmatrix} XR_{1,2}R_{2,1}X & 0 & 0 & 0 \\ 0 & R_{1,2}R_{2,1} & 0 & 0 \\ 0 & 0 & R_{2,1}R_{1,2} & 0 \\ 0 & 0 & 0 & XR_{2,1}R_{1,2}X \end{bmatrix} = 0.$$

From this we conclude that $(-\mathrm{Id} + (\mathbb{A}\Pi + \Pi\mathbb{A})/2)^{-1} = -\mathrm{Id} - (\mathbb{A}\Pi + \Pi\mathbb{A})/2$. Coming back to (2.4.1), we obtain a first expression for the inverse of the local multi-trace operator, namely

$$\begin{aligned}
&[\,-\mathbb{A}/2 - \sigma\Pi + (\sigma+1/2)\mathrm{Id}\,]^{-1} \\
&\quad = \sigma^{-1}[\,\mathbb{A}/2 + \sigma\Pi + (\sigma+1/2)\mathrm{Id}\,]\,[\mathrm{Id} + (\mathbb{A}\Pi + \Pi\mathbb{A})/2] \\
&\quad = \sigma^{-1}[\,\tfrac{1}{2}(1+\sigma)\mathbb{A} + (\sigma+1/4)\Pi + (\sigma+1/2)(\mathrm{Id} + (\mathbb{A}\Pi + \Pi\mathbb{A})/2)] \\
&\qquad + \sigma^{-1}[\,\tfrac{\sigma}{2}\Pi\mathbb{A}\Pi + \tfrac{1}{4}\mathbb{A}\Pi\mathbb{A}].
\end{aligned} \tag{2.4.3}$$

The only terms that are not explicitly known yet in (2.4.3) are the last two, $\Pi\mathbb{A}\Pi$ and $\mathbb{A}\Pi\mathbb{A}$. Combining (2.4.2) with the definitions given in (2.3.5), direct calculation yields

$$\Pi\mathbb{A}\Pi = \begin{bmatrix} -A_1 & 0 & 0 & XR_{1,2}X \\ 0 & -\tilde{A}_1 & 0 & 0 \\ 0 & 0 & -\tilde{A}_2 & 0 \\ XR_{2,1}X & 0 & 0 & -A_2 \end{bmatrix},$$

and similarly, we also obtain

$$\mathbb{A}\Pi\mathbb{A} = \begin{bmatrix} 0 & -X & XR_{1,2} & 0 \\ -X & 0 & 0 & -R_{1,2}X \\ -R_{2,1}X & 0 & 0 & -X \\ 0 & XR_{2,1} & -X & 0 \end{bmatrix}.$$

We have now derived an explicit expression for each term in (2.4.3), which leads to a close form matrix expression for the inverse of the local multi-trace operator, namely

$$\mathrm{MTF}_{\mathrm{loc}}^{-1} = (1+\frac{1}{2\sigma})\mathrm{Id} + \frac{1}{\sigma}\begin{bmatrix} \frac{1}{2}\mathrm{A}_1 & \sigma\mathrm{X} & \frac{\sigma+1}{2}\mathrm{X}\mathrm{R}_{1,2} & \frac{\sigma}{2}\mathrm{X}\mathrm{R}_{1,2}\mathrm{X} \\ \sigma\mathrm{X} & \frac{1}{2}\tilde{\mathrm{A}}_1 & \frac{\sigma+1}{2}\mathrm{R}_{1,2} & \frac{\sigma}{2}\mathrm{R}_{1,2}\mathrm{X} \\ \frac{\sigma}{2}\mathrm{R}_{2,1}\mathrm{X} & \frac{\sigma+1}{2}\mathrm{R}_{2,1} & \frac{1}{2}\tilde{\mathrm{A}}_2 & \sigma\mathrm{X} \\ \frac{\sigma}{2}\mathrm{X}\mathrm{R}_{2,1}\mathrm{X} & \frac{\sigma+1}{2}\mathrm{X}\mathrm{R}_{2,1} & \sigma\mathrm{X} & \frac{1}{2}\mathrm{A}_2 \end{bmatrix}. \tag{2.4.4}$$

The expression $\mathrm{MTF}_{\mathrm{loc}} \cdot \mathrm{MTF}_{\mathrm{loc}}^{-1} = \mathrm{Id}$ should not be mistaken for the Calderón identity (2.3.6). The primary difference is that (2.4.4) involves coupling terms between Ω_1 and Ω_2, whereas in (2.3.6), all three subdomains are decoupled.

2.5 Numerical Experiments

2.5.1 Verifying the inversion formula

We now illustrate the closed form inversion formula (2.4.4) for the local multi-trace formulation by a numerical experiment. We consider a three dimensional version of the geometrical setting described at the beginning in Figure 2.1. Here $\Omega_1 := \mathrm{B}(0, 0.5)$ is the open ball centered at 0 with radius 0.5, $\Omega_2 := \mathbb{R}^3 \backslash [-1,+1]^3$, and $\Omega_0 := \mathbb{R}^3 \setminus \overline{\Omega}_1 \cup \overline{\Omega}_2$, see Figure 2.2.

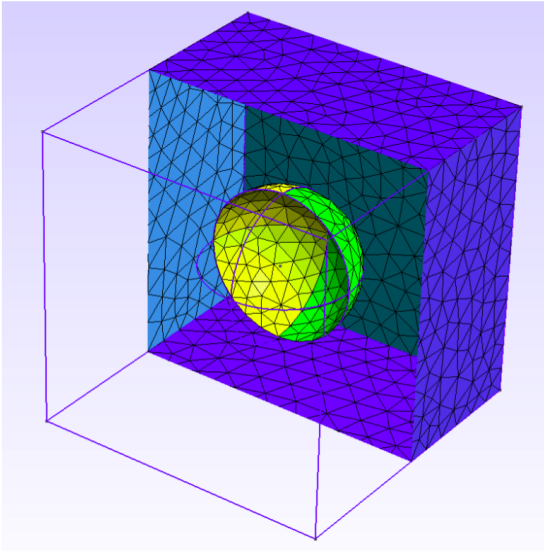


Figure 2.2: 3D geometry for the numerical experiment

For our numerical results, we discretize both $\mathrm{MTF}_{\mathrm{loc}}$ given by (2.3.7) leading to a matrix we denote by $[\mathrm{MTF}_{\mathrm{loc}}]$, and $\mathrm{MTF}_{\mathrm{loc}}^{-1}$ given by (2.4.4) leading to a matrix denoted by $[\mathrm{MTF}_{\mathrm{loc}}^{-1}]$. Our discretization using the code BEMTOOL[1] is based on a Galerkin method where both Dirichlet and Neumann traces are approximated by means of continuous piece-wise linear functions on the same mesh. We use a triangulation with a mesh width $h = 0.35$, and generated the mesh using GMSH, see [GR09].

Let M_h be the mass matrix associated with the duality pairing used to write (2.3.2) in variational form. We represent the spectrum of the matrix $\mathrm{M}_h^{-1} \cdot [\mathrm{MTF}_{\mathrm{loc}}] \cdot \mathrm{M}_h^{-1} \cdot [\mathrm{MTF}_{\mathrm{loc}}^{-1}]$ in Figure 2.3. We see that

[1] available on https://github.com/xclaeys/bemtool under Lesser Gnu Public License.

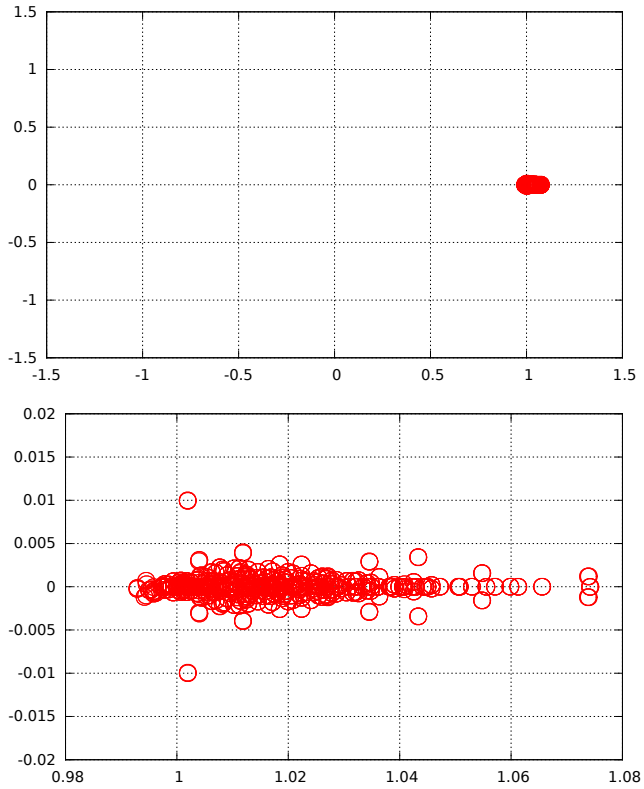


Figure 2.3: Eigenvalues of the matrix $M_h^{-1} \cdot [MTF_{loc}] \cdot M_h^{-1} \cdot [MTF_{loc}^{-1}]$ for $\sigma = -\frac{1}{2}$, with a zoom below around 1.

the eigenvalues are clustered around 1, which agrees well with our analysis at the continuous level.

2.5.2 Preconditioner efficiency

Let us consider the operator $P := MTF_{loc}^{-1}$, obtained by choosing $\sigma = -\frac{1}{2}$ and considering a constant wave number $\kappa_0 = 1$ for all operators in equation (2.4.4). In this section, we evaluate the performance of P for preconditioning the local-MTF operator corresponding to a composite scattering problem in heterogenous media. We use the same domain as in section 2.5.1 and consider the problem of finding the solution u of the following operator equation,

$$M(\kappa_0, \kappa_1, \kappa_2)u = b, \tag{2.5.1}$$

where b is a constant 1-valued function and $M(\kappa_0, \kappa_1, \kappa_2)$ denotes the local-MTF operator considering the wave numbers different in each domain and given as $\kappa_0 = 1$, κ_1 and κ_2 respectively in Ω_0, Ω_1 and Ω_2. Next, we use P for preconditioning (2.5.1). Then, we discretize the problem to obtain a linear system which we solve using GMRES with a restart value of 40 and a tolerance of 10^{-8}.

In figures 2.4 and 2.5 we present representative numerical tests which show that P is an efficient operator preconditioner for the local-MTF operator, and its effect is more notorious when the wave numbers are close to each other, which is in agreement with the theory. We compare our preconditioning technique against the classical block diagonal preconditioning operator D, which was used in the paper introducing local-MTF [HJH12, Sec. 5.3], given as

$$D := \begin{bmatrix} A_1 & 0 & 0 & 0 \\ 0 & \tilde{A}_1 & 0 & 0 \\ 0 & 0 & \tilde{A}_2 & 0 \\ 0 & 0 & 0 & A_2 \end{bmatrix}.$$

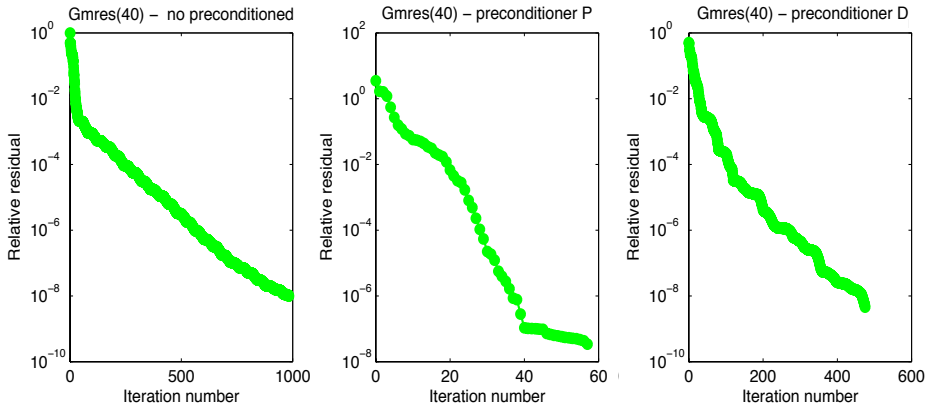


Figure 2.4: Convergence history of GMRES with a restart value of 40, case $\kappa_0 = 1$, $\kappa_1 = 6$, $\kappa_2 = 6$.

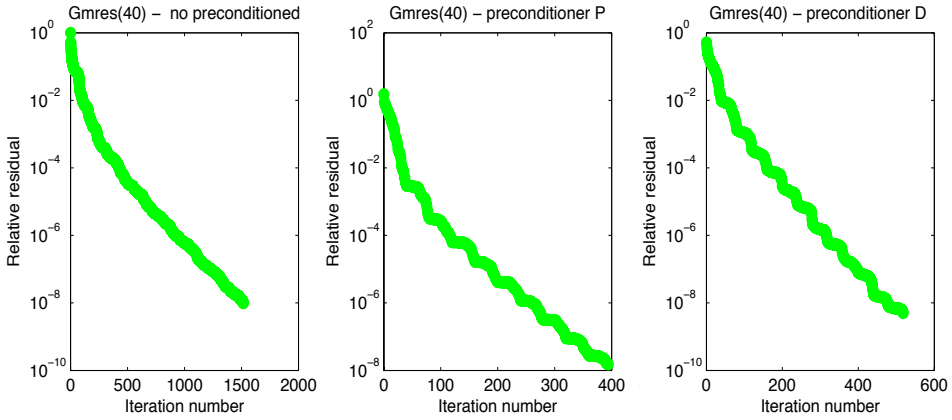


Figure 2.5: Convergence history of GMRES with a restart value of 40, case $\kappa_0 = 1$, $\kappa_1 = 5$, $\kappa_2 = 10$.

2.6 Conclusions of the chapter

We have shown that it is possible for the local multi-trace operator of a model transmission problem to obtain a closed form for the inverse. This would therefore be an ideal preconditioner for local multi-trace formulations. The closed form inverse seems to be inherent to the formulation, and not dependent on the specific form of the partial differential equation. An open question that stems from this chapter is if such closed form inverses can also be obtained for more general situations, where the coefficients are only constant in each subdomain, and in the presence of more subdomains.

CHAPTER 3

Stability of Local-MTF for Maxwell equation

3.1 Preliminaries

In this chapter, we study the local-MTF for an electromagnetic problem posed in a geometric configuration consisting in two subdomains and one interface, where one subdomain is the open unit ball, and the interface is the unit sphere. Our goal is to investigate the stability of the local-MTF for Maxwell in this precise setting. For this, we first prove that the local multi-trace operator is injective and then we write the local-MTF formulation in full detail by means of separation of variables based on vector spherical harmonics. We derive an explicit formula for the corresponding boundary integral operator, relying only on special functions (spherical Bessel and Hankel functions). We end this chapter plotting the first eigenvalues of the operator and commenting on the stability.

This chapter is structured as follows, in Section 3.2 we state the problem and background concepts needed for the analysis later on. Section 3.3 presents the local multi-trace operator for this framework and then a prove that such operator is injective. In Sections 3.4 and 3.5 we use a separation of variables technique based on vector spherical harmonics to derive explicit formulae for the local-MTF operator and the accumulation points of its spectrum. In Section 3.6, we numerically verify the theoretical analysis and in Section 3.7 we discuss on the stability of local-MTF for Maxwell equations. Finally, Section 3.8 concludes the chapter.

3.2 Problem setting

We consider a partition of free space as $\mathbb{R}^3 = \overline{\Omega}_0 \cup \overline{\Omega}_1$ in two smooth open subdomains, Ω_0 and Ω_1, such that $\Omega_0 = \mathbb{R}^3 \setminus \overline{\Omega}_1$. We denote $\Gamma = \partial\Omega_0 = \partial\Omega_1$ and let n_j refer to the unit normal vector to Γ_j directed toward the exterior of Ω_j, so that we have $n_0 = -n_1$. Let $\epsilon_j > 0$ (resp. $\mu_j > 0$) refer to the electric permittivity (resp. magnetic permeability) in domain Ω_j. We are interested in the scattering of an incident electromagnetic wave $(\mathbf{E}_{\text{inc}}, \mathbf{H}_{\text{inc}})$ propagating in harmonic regime at pulsation $\omega > 0$. The equations under consideration then write

$$\begin{cases} \mathbf{curl}(\mathbf{E}) - \imath\omega\mu_j\mathbf{H} = 0 \\ \mathbf{curl}(\mathbf{H}) + \imath\omega\epsilon_j\mathbf{E} = 0 \\ \sqrt{\mu_0}(\mathbf{H} - \mathbf{H}_{\text{inc}}) \times \hat{x} - \sqrt{\epsilon_0}(\mathbf{E} - \mathbf{E}_{\text{inc}}) = \mathcal{O}_{|x|\to\infty}(|x|^{-2}) \end{cases} \tag{3.2.1}$$

$$\begin{cases} n_0 \times \mathbf{E}|_{\Gamma_0} + n_1 \times \mathbf{E}|_{\Gamma_1} = 0 \\ n_0 \times \mathbf{H}|_{\Gamma_0} + n_1 \times \mathbf{H}|_{\Gamma_1} = 0 \end{cases} \tag{3.2.2}$$

where we assume that $\mathbf{curl}(\mathbf{E}_{\text{inc}}) - \imath\omega\mu_0\mathbf{H}_{\text{inc}} = 0$ in $\mathbb{R}^3$ and $\mathbf{curl}(\mathbf{H}_{\text{inc}}) + \imath\omega\epsilon_0\mathbf{E}_{\text{inc}} = 0$ in $\mathbb{R}^3$; the incident field may be, for example, a plane wave. In addition, in the problem above $\hat{x} := x/|x|$. In Equation (3.2.2) the notation "$\mathbf{E}|_{\Gamma_j}$" (resp. "$\mathbf{H}|_{\Gamma_j}$") should be understood as the trace taken at Γ from the interior of Ω_j. Next, let us point out that (3.2.1)-(3.2.2) can be reformulated as a second order transmission boundary value problem, which is the basis of Stratton-Chu potential theory, given as,

$$\begin{cases} \mathbf{curl}^2(\mathbf{E}) - \kappa_j^2\mathbf{E} = 0 \quad \text{in } \Omega_j \\ \mathbf{curl}(\mathbf{E} - \mathbf{E}_{\text{inc}}) \times \hat{x} - \imath\kappa_0(\mathbf{E} - \mathbf{E}_{\text{inc}}) = \mathcal{O}_{|x|\to\infty}(|x|^{-2}) \end{cases} \tag{3.2.3}$$

$$\begin{cases} n_0 \times \mathbf{E}|_{\Gamma_0} + n_1 \times \mathbf{E}|_{\Gamma_1} = 0 \\ \mu_0^{-1} n_0 \times \mathbf{curl}(\mathbf{E})|_{\Gamma_0} + \mu_1^{-1} n_1 \times \mathbf{curl}(\mathbf{E})|_{\Gamma_1} = 0 \end{cases} \tag{3.2.4}$$

In the equations above we adopted the following notations for effective wave number in each subdomain

$$\kappa_j := \omega\sqrt{\mu_j\epsilon_j}, \quad j = 0, 1. \tag{3.2.5}$$

We wish to study the solution of this problem by means of a boundary integral formulation. There are several possible such formulations. We focus here on the local Multi-Trace formulation (local-MTF). Since a complete stability analysis of local-MTF is not presently available, we will concentrate on the following special case.

Assumption: Ω_1 is the unit ball and Γ is the unit sphere.

This will allow explicit calculus by means of separation of variables which will help to investigate and clarify the structure of operators associated to the local-MTF.

3.3 Local multi-trace operator for Maxwell equation

For the mathematical analysis, we heavily rely on potential theory for electromagnetics, *i.e* Statton-Chu theory. On the sequel, $\mathcal{G}_\kappa(x) := \exp(\imath\kappa|x|)/(4\pi|x|)$ will refer to the outgoing Green kernel of the

Helmholtz equation with wave number $\kappa > 0$. Next we define the integral operators: for $u = (u, p) \in \mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)^2$, we set

$$\begin{aligned} &\mathrm{G}_\kappa(u)(x) := \mathrm{DL}_\kappa(u)(x) + \mathrm{SL}_\kappa(p)(x), \\ &\text{where} \quad \mathrm{SL}_\kappa(p)(x) := \kappa^{-2} \int_\Gamma \nabla \mathscr{G}_\kappa(x-y) \mathrm{div}_\Gamma p(y) d\sigma(y) + \int_\Gamma \mathscr{G}_\kappa(x-y) p(y) d\sigma(y), \\ &\qquad \mathrm{DL}_\kappa(u)(x) := \mathbf{curl} \int_\Gamma \mathscr{G}_\kappa(x-y) u(y) d\sigma(y). \end{aligned}$$

The potential operator G_κ continuously maps $\mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)^2$ into $\mathrm{H}_{\mathrm{loc}}(\mathbf{curl}, \Omega_0)$ and it also satisfies $(\mathbf{curl}^2 - \kappa_0^2)\mathrm{G}_\kappa(u) = 0$ in Ω_0 as well as a Silver-Müller radiation condition at infinity, regardless of $u \in \mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)^2$. A similar result also holds in Ω_1. The potential operator plays a central role in the derivation of boundary integral equations as it can be used to represent solution to homogeneous Maxwell equations according to the Stratton-Chu representation theorem given as follows, *c.f.* [BH03, Thm. 6].

Theorem 3.1. *Let* $\mathbf{U} \in \mathrm{H}_{\mathrm{loc}}(\mathbf{curl}, \Omega_j)$ *satisfy* $\mathbf{curl}^2(\mathbf{U}) - \kappa_j^2 \mathbf{U} = 0$ *in* Ω_j. *For* $j = 0$ *assume in addition that* $\mathbf{curl}(\mathbf{U}) \times \hat{x} - \imath\kappa_0 \mathbf{U} = \mathscr{O}(|x|^{-2})$ *for* $|x| \longrightarrow \infty$. *Then,*

$$\mathrm{G}_\kappa(\gamma^j(\mathbf{U}))(x) = 1_{\Omega_j}(x)\mathbf{U}(x),$$

for all $x \in \mathbb{R}^3$.

On the other hand, the jumps of trace, *c.f.* (2.2.5), of the potential operator follow a simple and explicit expression given by the following proposition which can be found in [BH03, Thm. 7].

Proposition 3.1. *For any* $u \in \mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)^2$ *we have* $[\gamma^j] \cdot \mathrm{G}_\kappa(u) = u$.

In the forthcoming analysis, we shall make intensive use of the operator $\mathrm{A}_\kappa^j := 2\{\gamma^j\} \cdot \mathrm{G}_\kappa$. From the classical theory of potentials, it is clear that $\{\gamma_{\mathrm{T}}^j\} \cdot \mathrm{DL}_\kappa = \{\gamma_{\mathrm{R}}^j\} \cdot \mathrm{SL}_\kappa$, see *e.g.* [Ste08, SS11]. On the other hand, using the vector Helmholtz equation satisfied by $\int_\Gamma \mathscr{G}_\kappa(x-y)u(y)d\sigma(y)$, we get also that $\{\gamma_{\mathrm{R}}^j\} \cdot \mathrm{DL}_\kappa = \kappa^2 \{\gamma_{\mathrm{T}}^j\} \cdot \mathrm{SL}_\kappa$. As a consequence the operator A_κ^j can be represented in matrix form as

$$\mathrm{A}_\kappa^j := \begin{bmatrix} \mathrm{K}_\kappa^j & \kappa^{-1}\mathrm{V}_\kappa^j \\ \kappa \mathrm{V}_\kappa^j & \mathrm{K}_\kappa^j \end{bmatrix} \qquad \text{where} \quad \begin{cases} \mathrm{V}_\kappa^j := (2/\kappa)\{\gamma_{\mathrm{R}}^j\} \cdot \mathrm{DL}_\kappa, \\ \mathrm{K}_\kappa^j := 2\{\gamma_{\mathrm{T}}^j\} \cdot \mathrm{DL}_\kappa. \end{cases} \tag{3.3.1}$$

Observe that, for a given κ we have $\mathrm{A}_\kappa^0 = -\mathrm{A}_\kappa^1$ due to the change in the orientation of the normals $n_0 = -n_1$. The following proposition states that the operators in (3.3.1) can be used to characterize solutions of Maxwell equations in a given subdomain.

Proposition 3.2. *The operator* $\gamma^j \mathrm{G}_\kappa = (\mathrm{Id} + \mathrm{A}_\kappa^j)/2$ *is a continuous projector as a mapping from* $\mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)^2$ *into* $\mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)^2$. *Its range is the space of traces* $\gamma^j(\mathbf{U})$, *where* $\mathbf{U} \in \mathrm{H}_{\mathrm{loc}}(\mathbf{curl}, \Omega_j)$ *satisfies*

$$\mathbf{curl}^2(\mathbf{U}) - \kappa^2 \mathbf{U} = 0 \text{ in } \Omega_j,$$

as well as

$$\mathbf{curl}(\mathbf{U}) \times \hat{x} - \imath\kappa_0 \mathbf{U} = \mathscr{O}(|x|^{-2}), \quad \text{for } |x| \longrightarrow \infty \text{ if } j = 0.$$

An easy consequence of the above proposition is that $(\mathrm{A}_\kappa^j)^2 = \mathrm{Id}$ which is known as Calderón's identity. The incident field is solution to Maxwell equations with wave number κ_0 on $\mathbb{R}^3$ including inside Ω_1, so that we get $(\mathrm{A}_{\kappa_0}^1 - \mathrm{Id})\gamma^1(\mathbf{E}_{\mathrm{inc}}) = 0$ according to the proposition above. Since on the other

hand $A^1_{\kappa_0} = -A^0_{\kappa_0}$ and $\gamma^0(\mathbf{E}_{\text{inc}}) = -\gamma^1(\mathbf{E}_{\text{inc}})$ (continuity across interfaces), we conclude that $A^0_{\kappa_0}\gamma^1(\mathbf{E}_{\text{inc}}) = -\gamma^1(\mathbf{E}_{\text{inc}})$. Using Proposition 3.2, we also observe that equations (3.2.3) can be reformulated as

$$(A^1_{\kappa_1} - \mathrm{Id})\gamma^1(\mathbf{E}) = 0,$$

and,

$$(A^0_{\kappa_0} - \mathrm{Id})(\gamma^0(\mathbf{E}) - \gamma^0(\mathbf{E}_{\text{inc}})) = 0,$$

where the later is equivalent to $(A^0_{\kappa_0} - \mathrm{Id})\gamma^0(\mathbf{E}) = -2\gamma^0(\mathbf{E}_{\text{inc}})$.

Next, we need to reformulate the transmission conditions (3.2.4). Since these conditions are weighted with the permeability coefficients μ_j, we need to introduce scaling operators $\tau_\alpha : \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2 \longrightarrow \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2$ defined by $\tau_\alpha(v,q) := (v,\alpha q)$. The transmission conditions then rewrite

$$\tau^{-1}_{\omega\mu_0}\gamma^0(\mathbf{E}) + \tau^{-1}_{\omega\mu_1}\gamma^1(\mathbf{E}) = 0. \tag{3.3.2}$$

For the sake of conciseness, we choose $u_j = \tau^{-1}_{\omega\mu_j}\gamma^j(\mathbf{E})$ as unknowns of our problem. As a consequence, equations (3.2.3)-(3.2.4) rewrite

$$\begin{cases} (A^0_{\kappa_0,\mu_0} - \mathrm{Id})u_0 = -2\tau^{-1}_{\omega\mu_0}\gamma^0(\mathbf{E}_{\text{inc}}), \\ (A^1_{\kappa_1,\mu_1} - \mathrm{Id})u_1 = 0, \\ u_0 + u_1 = 0, \end{cases} \tag{3.3.3}$$

where we systematically denote $\epsilon := \kappa^2/(\omega^2\mu)$ so that $\omega\mu/\kappa = \sqrt{\mu/\epsilon}$, and the scaled operators are defined as

$$A^j_{\kappa,\mu} := \tau^{-1}_{\omega\mu}\cdot A^j_\kappa \cdot \tau_{\omega\mu} = \begin{bmatrix} \mathrm{K}^j_\kappa & \sqrt{\mu/\epsilon}\,\mathrm{V}^j_\kappa \\ \sqrt{\epsilon/\mu}\,\mathrm{V}^j_\kappa & \mathrm{K}^j_\kappa \end{bmatrix}. \tag{3.3.4}$$

With this definition, we have $(A^j_{\kappa,\mu})^2 = \mathrm{Id}$. Now let us rewrite (3.3.3) in a matrix form. We first introduce the continuous map $A_{(\kappa,\mu)} : \mathbb{H}(\Sigma) \longrightarrow \mathbb{H}(\Sigma)$ as a block diagonal operator $A_{(\kappa,\mu)}(u) := (A^0_{\kappa_0,\mu_0}(u_0), A^1_{\kappa_1,\mu_1}(u_1))$ for any $u = (u_0,u_1) \in \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2$. The first two rows of (3.3.3) can be rewritten as

$$(A_{(\kappa,\mu)} - \mathrm{Id})u = f, \tag{3.3.5}$$

where $u = (u_0,u_1)$ and $f = (-2\tau^{-1}_{\omega\mu_0}\gamma^0(\mathbf{E}_{\text{inc}}),0)$. To enforce transmission conditions, we also need to consider an operator $\Pi : \mathbb{H}(\Sigma) \longrightarrow \mathbb{H}(\Sigma)$ whose action consists in inverting traces from both sides of the interface. It is defined by $\Pi(u_0,u_1) := (u_1,u_0)$ for $u_0,u_1 \in \mathrm{H}^{-1/2}(\mathrm{div},\Gamma)$, so that transmission conditions simply rewrite $u = \Pi(u)$. Plugging the transmission operator into (3.3.5) leads to the local Multi-Trace formulation of (3.2.3)-(3.2.4),

$$\begin{cases} \text{Find } u \in \mathbb{H}(\Sigma) \text{ such that} \\ \mathrm{MTF}_{\text{loc}}(u) = f, \end{cases} \tag{3.3.6}$$

where $\mathrm{MTF}_{\text{loc}} := A_{(\kappa,\mu)} + \Pi = \begin{bmatrix} A^0_{\kappa_0,\mu_0} & \mathrm{Id} \\ \mathrm{Id} & A^1_{\kappa_1,\mu_1} \end{bmatrix}$.

As a first result, we prove injectivity of the local Multi-Trace operator, and hence unique solvability of the above equation.

Proposition 3.3. $\mathrm{Ker}(\mathrm{MTF}_{\text{loc}}) = \{0\}$.

Proof:

Assume that $\mathrm{MTF}_{\mathrm{loc}}(u) = 0$ for some $u = (u_0, u_1) \in \mathrm{H}^{-1/2}(\mathrm{div}, \Gamma)^4$ and set $\psi_j := \mathrm{G}_{\kappa_j}(\tau_{\omega\mu_j}(u_j))$ for both $j = 0, 1$. We have $[\gamma^j](\psi_j) = \tau_{\omega\mu_j}(u_j)$ according to the jump formula of Proposition 3.1. Since $2\{\gamma^j\} = 2\gamma^j - [\gamma^j]$, we have

$$2\tau^{-1}_{\omega\mu_j}(\{\gamma^j\}(\psi_j)) = 2\tau^{-1}_{\omega\mu_j}(\gamma^j(\psi_j)) - u_j, \quad j = 0, 1. \tag{3.3.7}$$

Next, from $\mathrm{MTF}_{\mathrm{loc}}(u) = 0$ we directly deduce that $\mathrm{A}^0_{\kappa_0,\mu_0}(u_0) + u_1 = 0$, which rewrites $2\tau^{-1}_{\omega\mu_0}\gamma^0(\psi_0) = u_0 - u_1$, according to (3.3.4) and (3.3.7). Similarly we obtain $2\tau^{-1}_{\omega\mu_1}\gamma^1(\psi_1) = u_1 - u_0$. As a consequence we have $\tau^{-1}_{\omega\mu_0}\gamma^0(\psi_0) + \tau^{-1}_{\omega\mu_1}\gamma^1(\psi_1) = 0$, which rewrites

$$\begin{aligned} &\tau^{-1}_{\omega\mu_0}\gamma^0(\psi) + \tau^{-1}_{\omega\mu_1}\gamma^1(\psi) = 0, \\ &\text{for} \quad \psi := 1_{\Omega_0}\psi_0 + 1_{\Omega_1}\psi_1. \end{aligned}$$

By construction, ψ_0 and thus ψ satisfies the Silver-Müller radiation condition $\mathbf{curl}(\psi) \times \hat{x} - \imath\kappa_0\psi = \mathcal{O}(|x|^{-2})$ for $|x| \longrightarrow \infty$ and $\mathbf{curl}^2(\psi) - \kappa_j^2\psi = 0$ in Ω_j, $j = 0, 1$. As a consequence we conclude that ψ is solution to an homogeneous transmission problem (similar to (3.2.3)-(3.2.4) with an incident field equal to 0). This leads to the conclusion that $\psi = 0$ in $\mathbb{R}^3$. In other words,

$$\psi_j = 0 \quad \text{in} \ \ \Omega_j \ \ \text{for} \ \ j = 0, 1. \tag{3.3.8}$$

According to (3.3.7),this implies that $2\tau^{-1}_{\omega\mu_j}\{\gamma^j\}(\psi_j) = -u_j$ for $j = 0, 1$, and finally using (3.3.4), we obtain $(\Pi - \mathrm{Id})u = 0$ which is equivalent to $u_0 = -u_1$. Besides, $2\{\gamma^j\} = \gamma^j + \gamma^j_c$ so that $2\tau^{-1}_{\omega\mu_j}\{\gamma^j\}(\psi_j) = \tau^{-1}_{\omega\mu_j}\gamma^j_c(\psi_j) = u_j$ for $j = 0, 1$. So we conclude that

$$\begin{aligned} &\tau^{-1}_{\omega\mu_0}\gamma^0(\psi_c) + \tau^{-1}_{\omega\mu_1}\gamma^1(\psi_c) = 0, \\ &\text{for} \quad \psi_c := 1_{\Omega_1}\psi_0 - 1_{\Omega_0}\psi_1. \end{aligned}$$

By construction ψ_1 and thus ψ_c satisfies the Silver-Müller radiation condition $\mathbf{curl}(\psi_c) \times \hat{x} - \imath\kappa_1\psi_c = \mathcal{O}(|x|^{-2})$ for $|x| \longrightarrow \infty$ and $\mathbf{curl}^2(\psi_c) - \kappa_j^2\psi_c = 0$ in $\mathbb{R}^3 \setminus \overline{\Omega}_j$, for $j = 0, 1$. As a consequence, ψ_c is the solution to an homogeneous transmission problem with wave number κ_0 in Ω_1 (resp. κ_1 in Ω_0). We conclude that $\psi_c = 0$ in $\mathbb{R}^3$ i.e. $\psi_j = 0$ in $\mathbb{R}^3 \setminus \overline{\Omega}_j$.

To summarize we have established that $\psi_j = 0$ in both Ω_j and $\mathbb{R}^3 \setminus \overline{\Omega}_j$. Taking the jump trace we conclude $u_j = \tau^{-1}_{\omega\mu_j}[\gamma^j](\psi_j) = 0$, which finishes the proof. □

3.4 Separation of variables

We are interested in deriving an explicit expression of operator (3.3.6). As the present geometrical setting admits spherical symmetry, this can be obtained by means of separation of variables based on spherical harmonics. Any tangential vector field $u \in \mathrm{L}^2_{\mathrm{T}}(\Gamma) := \{v : \Gamma \longrightarrow \mathbb{C}, \ v(x) \cdot x = 0 \, \forall x \in \Gamma, \ \|v\|^2_{\mathrm{L}^2_{\mathrm{T}}(\Gamma)} := \int_\Gamma |v|^2 d\sigma < +\infty\}$ can be decomposed as

$$\begin{aligned} u(x) &= \sum_{n=0}^{+\infty} \sum_{|m|\le n} u^{\parallel}_{n,m}\mathbf{X}^{\parallel}_{n,m}(x) + u^{\times}_{n,m}\mathbf{X}^{\times}_{n,m}(x), \\ \text{with} \quad \mathbf{X}^{\parallel}_{n,m} &:= \frac{1}{\sqrt{n(n+1)}}\nabla_\Gamma \mathrm{Y}^m_n, \quad \mathbf{X}^{\times}_{n,m} := n_1 \times \mathbf{X}^{\parallel}_{n,m}, \end{aligned} \tag{3.4.1}$$

where ∇_Γ is the surface gradient. Denoting $(\theta,\varphi)\in[0,\pi]\times[0,2\pi]$ the spherical coordinates on Γ, the spherical harmonics (see e.g [OLBC10, §14.30]) are defined by

$$\mathrm{Y}_n^m(\theta,\varphi) := \sqrt{\frac{2n+1}{4\pi}}\sqrt{\frac{(n-|m|)!}{(n+|m|)!}}\,\mathrm{P}_n^{|m|}(\cos\theta)\exp(\imath m\varphi). \tag{3.4.2}$$

In the definition above, the functions $\mathrm{P}_n^m(t)$, $m\geq 0$, $t\in[0,1]$ refer to the associated Legendre functions, see e.g. [Leb72, §7.12]. The tangent fields $\mathbf{X}^{\shortparallel}_{n,m},\mathbf{X}^{\times}_{n,m}$ form an orthonormal Hilbert basis of $\mathrm{L}^2_{\mathrm{T}}(\Gamma)$. Let us denote $\mathbf{X}_{n,m}(x) := [\mathbf{X}^{\shortparallel}_{n,m}(x),\mathbf{X}^{\times}_{n,m}(x)]$ so that Expansion (3.4.1) can be rewritten in the more compact form

$$u(x)=\sum_{n=0}^{+\infty}\sum_{|m|\leq n}\mathbf{X}_{n,m}(x)\cdot u_{n,m}, \tag{3.4.3}$$

for a collection of coordinate vectors $u_{n,m}=[u^{\shortparallel}_{n,m},u^{\times}_{n,m}]^\top\in\mathbb{C}^2$. The operators coming into play in the expression of the local multi-trace operator (3.3.6) are actually (block) diagonalized by this basis. Define $\mathrm{J}_n(t) := \sqrt{\pi t/2}\mathrm{J}_{n+1/2}(t)$ where $\mathrm{J}_\nu(t)$ are Bessel functions of the first kind of order ν (see [OLBC10, §10.2]) and $\mathrm{H}_n(t) := \sqrt{\pi t/2}\mathrm{H}^{(1)}_{n+1/2}(t)$ where $\mathrm{H}^{(1)}_\nu(t)$ are Hankel functions of the first kind of order ν (see [OLBC10, §10.2 & §10.4]). Then, according to Lemma 1 in [VGG14], using notations (3.3.1), we have

$$\begin{aligned}&\mathrm{V}^0_\kappa\cdot\mathbf{X}_{n,m}(x)=\mathbf{X}_{n,m}(x)\cdot\mathrm{V}^0_\kappa[n],\quad\text{where}\\ &\mathrm{V}^0_\kappa[n] := \begin{bmatrix}0 & +2\imath\,\mathrm{J}_n(\kappa)\mathrm{H}_n(\kappa)\\ -2\imath\,\mathrm{J}'_n(\kappa)\mathrm{H}'_n(\kappa) & 0\end{bmatrix}\in\mathbb{C}^{2\times2}.\end{aligned} \tag{3.4.4}$$

Since $\mathrm{V}^1_\kappa=-\mathrm{V}^0_\kappa$ we have $\mathrm{V}^1_\kappa\cdot\mathbf{X}_{n,m}(x)=\mathbf{X}_{n,m}(x)\cdot\mathrm{V}^1_\kappa[n]$ by setting $\mathrm{V}^1_\kappa[n] := -\mathrm{V}^0_\kappa[n]$. According to Lemma 1 in [VGG14], we also have the explicit expression

$$\begin{aligned}&\mathrm{K}^0_\kappa\cdot\mathbf{X}_{n,m}(x)=\mathbf{X}_{n,m}(x)\cdot\mathrm{K}^0_\kappa[n]\quad\text{where}\\ &\mathrm{K}^0_\kappa[n] := \imath(\,\mathrm{J}_n(\kappa)\mathrm{H}'_n(\kappa)+\mathrm{J}'_n(\kappa)\mathrm{H}_n(\kappa)\,)\begin{bmatrix}-1 & 0\\ 0 & +1\end{bmatrix}\in\mathbb{C}^{2\times2}.\end{aligned} \tag{3.4.5}$$

Here again, defining $\mathrm{K}^1_\kappa[n]=-\mathrm{K}^0_\kappa[n]$ we obtain $\mathrm{K}^1_\kappa\cdot\mathbf{X}_{n,m}(x)=\mathbf{X}_{n,m}(x)\cdot\mathrm{K}^1_\kappa[n]$. From (3.4.4) and (3.4.5) we deduce an explicit expression for the operators $\mathrm{A}^j_{\kappa,\mu}$. First of all define the function $\mathbf{X}^{\#2}_{n,m}$ by the expression

$$\begin{aligned}\mathbf{X}^{\#2}_{n,m}(x) &:= [\mathbf{X}_{n,m}(x),\mathbf{X}_{n,m}(x)]\\ &= [\mathbf{X}^{\shortparallel}_{n,m}(x),\mathbf{X}^{\times}_{n,m}(x),\mathbf{X}^{\shortparallel}_{n,m}(x),\mathbf{X}^{\times}_{n,m}(x)]\end{aligned}$$

Then any element $u=(u,p)\in\mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2$ decomposes as $u(x)=\sum_{n,m}\mathbf{X}^{\#2}_{n,m}(x)\cdot u_{n,m}$ where $u_{n,m}\in\mathbb{C}^4$ are coordinate vectors that do not depend on x. In this basis, the operator $\mathrm{A}^j_{\kappa,\mu}$ admits the following matrix form

$$\begin{aligned}&\mathrm{A}^j_{\kappa,\mu}\cdot\mathbf{X}^{\#2}_{n,m}(x)=\mathbf{X}^{\#2}_{n,m}(x)\cdot\mathrm{A}^j_{\kappa,\mu}[n]\quad\text{where}\\ &\mathrm{A}^j_{\kappa,\mu}[n] := \begin{bmatrix}\mathrm{K}^j_\kappa[n] & \sqrt{\mu/\epsilon}\,\mathrm{V}^j_\kappa[n]\\ \sqrt{\epsilon/\mu}\,\mathrm{V}^j_\kappa[n] & \mathrm{K}^j_\kappa[n]\end{bmatrix}\in\mathbb{C}^{4\times4}.\end{aligned} \tag{3.4.6}$$

We can reiterate the notations we used above, and introduce the fields $\mathbf{X}^{\#4}_{n,m}(x) := [\mathbf{X}^{\#2}_{n,m}(x),\mathbf{X}^{\#2}_{n,m}(x)]$. Then, any $u=(u^0,u^1)\in\mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2\times\mathrm{H}^{-1/2}(\mathrm{div},\Gamma)^2$ can be decomposed as $u(x)=\sum_{n,m}\mathbf{X}^{\#4}_{n,m}(x)\cdot u_{n,m}$ where $u_{n,m}\in\mathbb{C}^8$ are coordinate vectors that do not depend on x. Then, the multi-trace operator (3.3.6)

is reduced to matrix form in this basis

$$\begin{aligned} &\mathrm{MTF}_{\mathrm{loc}} \cdot \mathbf{X}^{\#4}_{n,m}(x) = \mathbf{X}^{\#4}_{n,m}(x) \cdot \mathrm{MTF}_{\mathrm{loc}}[n], \quad \text{where} \\ &\mathrm{MTF}_{\mathrm{loc}}[n] := \begin{bmatrix} \mathrm{A}^0_{\kappa_0,\mu_0}[n] & \mathrm{Id} \\ \mathrm{Id} & \mathrm{A}^1_{\kappa_1,\mu_1}[n] \end{bmatrix} \in \mathbb{C}^{8\times 8}. \end{aligned} \tag{3.4.7}$$

3.5 Computation of accumulation points

In this section, we study in more detail the symbol of the boundary integral operators introduced in the previous section. To be more precise, we examine their behavior for $n \longrightarrow +\infty$. First of all, from the series expansion of spherical Bessel functions given by [OLBC10, §10.53], we deduce that for any fixed $t > 0$ we have

$$\begin{aligned} \mathrm{J}_n(t) &= t^{n+1}\frac{n!\,2^n}{(2n+1)!}\left\{1 - \frac{t^2}{4n} + \mathcal{O}\left(\frac{1}{n^2}\right)\right\}, \\ \mathrm{H}_n(t) &= -\imath t^{-n}\frac{(2n)!}{n!\,2^n}\left\{1 + \frac{t^2}{4n} + \mathcal{O}\left(\frac{1}{n^2}\right)\right\}. \end{aligned} \tag{3.5.1}$$

Since Bessel functions are expressed in terms of convergent series of analytic functions, we can derive the above asymptotics. This leads to the following behaviors for the derivatives,

$$\begin{aligned} \mathrm{J}'_n(t) &= t^{n}\frac{n!\,2^n}{(2n+1)!}\left\{n + 1 - \frac{t^2}{4} + \mathcal{O}\left(\frac{1}{n}\right)\right\}, \\ \mathrm{H}'_n(t) &= \imath t^{-(n+1)}\frac{(2n)!}{n!\,2^n}\left\{n + \frac{t^2}{4} + \mathcal{O}\left(\frac{1}{n}\right)\right\}. \end{aligned} \tag{3.5.2}$$

Next, we can combine these asymptotics to obtain the predominant behavior of the functions coming into play in the expression of the integral operators of the previous section. We have the following three elementary behaviors

$$\begin{aligned} &-2\imath \mathrm{J}_n(t)\mathrm{H}_n(t) \underset{n\to\infty}{\sim} -t/n, \\ &+2\imath \mathrm{J}'_n(t)\mathrm{H}'_n(t) \underset{n\to\infty}{\sim} -n/t, \\ &\imath\big(\,\mathrm{J}'_n(t)\mathrm{H}_n(t) + \mathrm{J}_n(t)\mathrm{H}'_n(t)\,\big) \underset{n\to\infty}{\sim} 1/(2n). \end{aligned} \tag{3.5.3}$$

Define $\mathrm{T}_n \in \mathbb{C}^{2\times 2}$ by $\mathrm{T}_n(u_1, u_2) := (u_1, u_2/n)$. From this we conclude that, as $n \longrightarrow +\infty$, we have $\mathrm{K}^0_\kappa[n] \sim (2n)^{-1}\widetilde{\mathrm{K}}^0_\kappa$ and $\mathrm{V}^0_\kappa[n] \sim \mathrm{T}_n^{-1} \cdot \widetilde{\mathrm{V}}^0_\kappa \cdot \mathrm{T}_n$ where $\widetilde{\mathrm{V}}^0_\kappa, \widetilde{\mathrm{K}}^0_\kappa \in \mathbb{C}^{2\times 2}$ are constant matrices independent of n given by

$$\widetilde{\mathrm{V}}^0_\kappa := \begin{bmatrix} 0 & \kappa \\ 1/\kappa & 0 \end{bmatrix} \quad \text{and} \quad \widetilde{\mathrm{K}}^0_\kappa := \begin{bmatrix} -1 & 0 \\ 0 & +1 \end{bmatrix}.$$

Next, define $\mathrm{T}^{\#2}_n \in \mathbb{C}^{4\times 4}$ by $\mathrm{T}^{\#2}_n(u_1, u_2) = (\mathrm{T}_n(u_1), \mathrm{T}_n(u_2))$ for any pair $u_1, u_2 \in \mathbb{C}^2$. Then, using the above results, the asymptotic behavior of the matrix $\mathrm{A}^0_{\kappa,\mu}[n]$ is given by $\mathrm{A}^0_{\kappa,\mu}[n] \sim (\mathrm{T}^{\#2}_n)^{-1}\tilde{\mathrm{A}}^0_{\kappa,\mu}\mathrm{T}^{\#2}_n$ where

$$\tilde{\mathrm{A}}^0_{\kappa,\mu} := \begin{bmatrix} 0 & \sqrt{\mu/\epsilon}\,\widetilde{\mathrm{V}}^0_\kappa \\ \sqrt{\epsilon/\mu}\,\widetilde{\mathrm{V}}^0_\kappa & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \omega\mu \\ 0 & 0 & (\omega\epsilon)^{-1} & 0 \\ 0 & \omega\epsilon & 0 & 0 \\ (\omega\mu)^{-1} & 0 & 0 & 0 \end{bmatrix}.$$

On the other hand, we also have $A^1_{\kappa,\mu}[n] \sim (T^{\#2}_n)^{-1}\tilde{A}^1_{\kappa,\mu}T^{\#2}_n$, where $\tilde{A}^1_{\kappa,\mu} := -\tilde{A}^0_{\kappa,\mu}$. Finally, define $T^{\#4}_n \in \mathbb{C}^{8\times 8}$ by $T^{\#4}_n(u_1,u_2) = (T^{\#2}_n(u_1), T^{\#2}_n(u_2))$ for any $u_1, u_2 \in \mathbb{C}^4$. Then we have the asymptotic behavior $\mathrm{MTF}_{\mathrm{loc}}[n] \sim (T^{\#4}_n)^{-1}\mathrm{MTF}^{\infty}_{\mathrm{loc}}T^{\#4}_n$ with

$$\mathrm{MTF}^{\infty}_{\mathrm{loc}} := \begin{bmatrix} \tilde{A}^0_{\kappa_0,\mu_0} & \mathrm{Id} \\ \mathrm{Id} & \tilde{A}^1_{\kappa_1,\mu_1} \end{bmatrix} \in \mathbb{C}^{8\times 8}. \tag{3.5.4}$$

It is important to observe that $\mathrm{MTF}^{\infty}_{\mathrm{loc}}$ does *not* depend on n. Since the eigenvalues of $\mathrm{MTF}_{\mathrm{loc}}[n]$ coincide with the eigenvalues of $T^{\#4}_n \cdot \mathrm{MTF}_{\mathrm{loc}}[n] \cdot (T^{\#4}_n)^{-1}$, this shows that the spectrum of $\mathrm{MTF}_{\mathrm{loc}}[n]$ converges toward the spectrum of $\mathrm{MTF}^{\infty}_{\mathrm{loc}}$.

Now let us investigate in detail the spectrum of the matrix $\mathrm{MTF}^{\infty}_{\mathrm{loc}}$. First, as an intermediate step, we examine the spectrum of the matrix $B := \tilde{A}^0_{\kappa_0,\mu_0} + \tilde{A}^1_{\kappa_1,\mu_1}$. A thorough examination shows that it takes the form

$$B := \tilde{A}^0_{\kappa_0,\mu_0} + \tilde{A}^1_{\kappa_1,\mu_1} = \begin{bmatrix} 0 & 0 & 0 & \alpha_\mu \\ 0 & 0 & \beta_\epsilon & 0 \\ 0 & \alpha_\epsilon & 0 & 0 \\ \beta_\mu & 0 & 0 & 0 \end{bmatrix}, \quad \text{with} \quad \begin{cases} \alpha_\epsilon = \omega\epsilon_0 - \omega\epsilon_1 \\ \alpha_\mu = \omega\mu_0 - \omega\mu_1 \\ \beta_\epsilon = (\omega\epsilon_0)^{-1} - (\omega\epsilon_1)^{-1} \\ \beta_\mu = (\omega\mu_0)^{-1} - (\omega\mu_1)^{-1} \end{cases} \tag{3.5.5}$$

Trying to compute directly the eigenvalues of the above matrix leads to the conclusion that any eigenvalue λ satisfies $\lambda^2 = \alpha_\epsilon\beta_\epsilon = -(\sqrt{\epsilon_1/\epsilon_0} - \sqrt{\epsilon_0/\epsilon_1})^2$, or $\lambda^2 = \alpha_\mu\beta_\mu = -(\sqrt{\mu_1/\mu_0} - \sqrt{\mu_0/\mu_1})^2$. As a consequence the spectrum of (3.5.5) is given by

$$\mathcal{S}(B) := \{\pm\imath\Lambda_\mu, \pm\imath\Lambda_\epsilon\} \quad \text{with} \quad \begin{cases} \Lambda_\mu = \left|\sqrt{\dfrac{\mu_1}{\mu_0}} - \sqrt{\dfrac{\mu_0}{\mu_1}}\right| \\ \Lambda_\epsilon = \left|\sqrt{\dfrac{\epsilon_1}{\epsilon_0}} - \sqrt{\dfrac{\epsilon_0}{\epsilon_1}}\right| \end{cases} \tag{3.5.6}$$

Now let us come back to $\mathrm{MTF}^{\infty}_{\mathrm{loc}}$. Recalling that $(\tilde{A}^j_{\kappa,\mu})^2 = \mathrm{Id}$, we obtain directly the following identity

$$(2\mathrm{Id} - (\mathrm{MTF}^{\infty}_{\mathrm{loc}})^2)^2 = \begin{bmatrix} B^2 & 0 \\ 0 & B^2 \end{bmatrix},$$

and taking into account (3.5.6), we finally obtain the following expression for the accumulation points of the spectrum of the local multi-trace operator $\mathrm{MTF}^{\infty}_{\mathrm{loc}}$,

$$\{\pm\sqrt{2 \pm \imath\Lambda_\mu}, \pm\sqrt{2 \pm \imath\Lambda_\epsilon}\}. \tag{3.5.7}$$

3.6 Numerical results

In the present section we numerically examine the spectrum of the operator $\mathrm{MTF}_{\mathrm{loc}}$. An explicit expression of the eigenvectors is provided by the vector spherical harmonics $\mathbf{X}^{\parallel}_{n,m}$ and $\mathbf{X}^{\times}_{n,m}$, so that we aim to graphically see the eigenvalue distribution of $\bigcup_{n=0}^{+\infty}\mathcal{S}(\mathrm{MTF}_{\mathrm{loc}}[n])$. Each $\mathcal{S}(\mathrm{MTF}_{\mathrm{loc}}[n])$ consists in 8 eigenvalues. On each picture below, we plot $\bigcup_{n=0}^{100}\mathcal{S}(\mathrm{MTF}_{\mathrm{loc}}[n])$ for various choices of parameters.

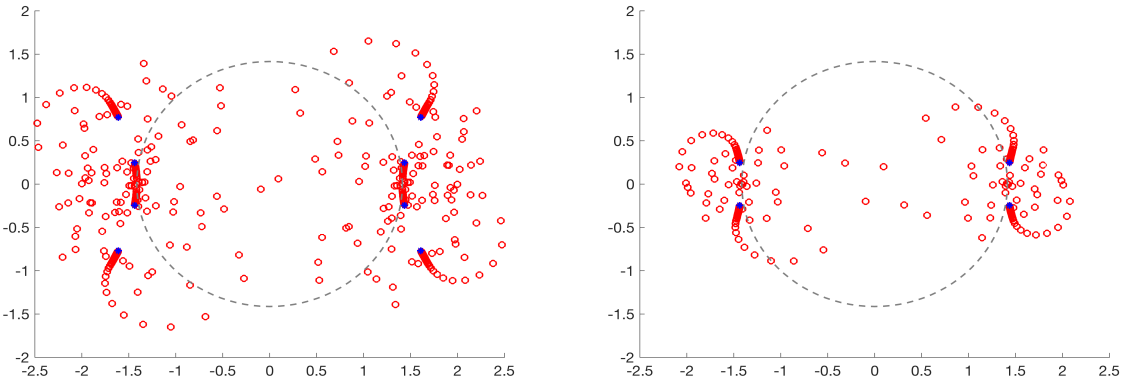


Figure 3.1: Eigenvalue distribution with $\kappa_0 = \kappa_1 = 2\pi/\lambda$ with $\lambda = 0.5$, $\mu_0 = 1, \mu_1 = 2$ (left) and $\mu_0 = 2, \mu_1 = 1$ (right).

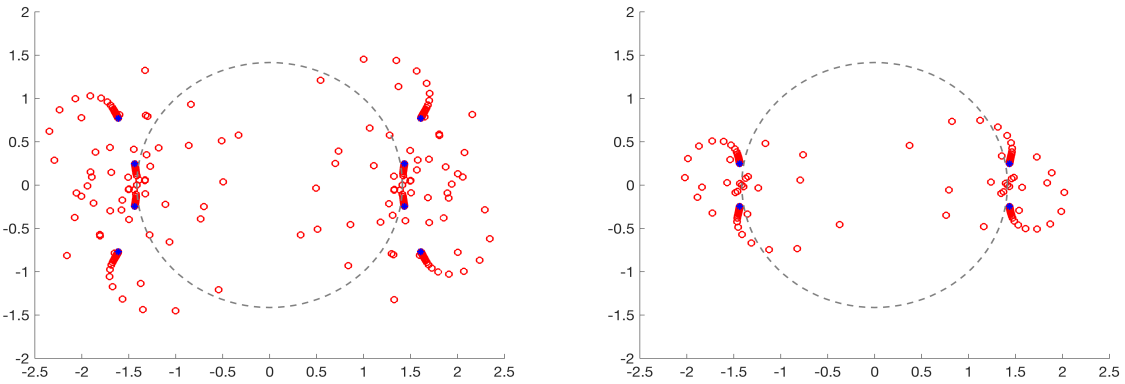


Figure 3.2: Eigenvalue distribution with $\kappa_0 = \kappa_1 = 2\pi/\lambda$ with $\lambda = 1$, $\mu_0 = 1, \mu_1 = 2$ (left) and $\mu_0 = 2, \mu_1 = 1$ (right).

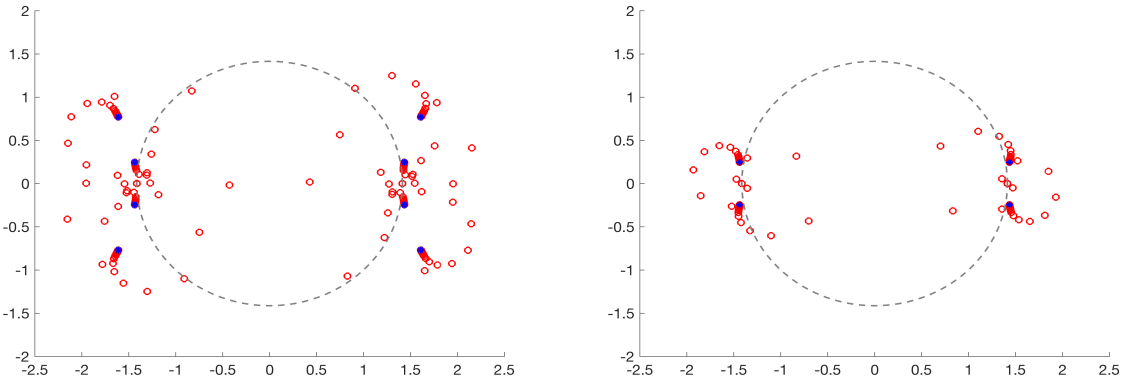


Figure 3.3: Eigenvalue distribution with $\kappa_0 = \kappa_1 = 2\pi/\lambda$ with $\lambda = 10$, $\mu_0 = 1, \mu_1 = 2$ (left) and $\mu_0 = 2, \mu_1 = 1$ (right).

These plots clearly confirm that the spectrum has no more than 8 accumulation points that systematically admit a modulus greater than $\sqrt{2}$.

3.7 Stability of local MTF

In this section, we establish a generalized Gårding inequality for the local Multi-Trace formulation on the unit sphere, by means of separation of variables. First of all, let us derive an expression of the norm on $\mathrm{H}^{-1/2}(\mathrm{div},\Gamma)$ in vector spherical harmonics. Such an expression can be obtained by noting that the dissipative counterpart of the EFIE operator (i.e. associated to a purely imaginary wave number) is continuous and coercive on $\mathrm{H}^{-1/2}(\mathrm{div},\Gamma)$ so that the corresponding bilinear form

$$(u,v)_{-1/2,\mathrm{div}} := \int_{\Gamma\times\Gamma} \mathscr{G}_\imath(x-y)\left(\mathrm{div}_\Gamma u(x)\mathrm{div}_\Gamma \overline{v}(y) + u(x)\cdot\overline{v}(y)\right)d\sigma(x,y), \tag{3.7.1}$$

yields a scalar product. Here $\mathscr{G}_\imath(x) = \exp(-|x|)/(4\pi|x|)$ and $\imath = \sqrt{-1}$ is the imaginary unit. The vector fields $\mathbf{X}^{\parallel}_{n,m}$ and $\mathbf{X}^{\times}_{n,m}$ form an orthogonal family with respect to this scalar product. As a consequence, to obtain an expression of a norm over $\mathrm{H}^{-1/2}(\mathrm{div},\Gamma)$, one can rely on the decomposition of the dissipative EFIE on vector spherical harmonics. First observe that $(u,v)_{-1/2,\mathrm{div}} = \int_\Gamma (n_0\times\gamma^0_\mathrm{T}\cdot\mathrm{SL}_\kappa(u))\cdot v d\sigma$. As a consequence, using (3.4.4) we obtain

$$\begin{aligned}
&(u,v)_{-1/2,\mathrm{div}} = \overline{v}^\top\cdot \mathrm{D}_n\cdot u,\\
&\text{where}\quad \mathrm{D}_n = \mathrm{diag}(\mathrm{J}'_n(\imath)\mathrm{H}'_n(\imath), \mathrm{J}_n(\imath)\mathrm{H}_n(\imath))\\
&\text{for}\quad u(x) = \mathbf{X}_{n,m}(x)\cdot u,\ v(x) = \mathbf{X}_{n,m}(x)\cdot v \quad u,v\in\mathbb{C}^2.
\end{aligned} \tag{3.7.2}$$

From this we deduce the asymptotic behavior $\mathrm{D}_n \sim \widetilde{\mathrm{D}}_n := \mathrm{diag}(1+n, 1/(1+n))$ for $n\longrightarrow\infty$, which yields the expression of an equivalent norm which is explicit when decomposed in spherical harmonics

$$\begin{aligned}
&c_-\|u\|^2_{-1/2,\mathrm{div}} \le (u,u)_{-1/2,\mathrm{div}} \le c_+\|u\|^2_{-1/2,\mathrm{div}},\\
&\|u\|^2_{-1/2,\mathrm{div}} := \sum_{n=0}^{+\infty}\sum_{|m|\le n} \overline{u}^\top_{n,m}\cdot\widetilde{\mathrm{D}}_n\cdot u_{n,m},\\
&\text{where}\quad \widetilde{\mathrm{D}}_n := \mathrm{diag}(1+n, 1/(1+n)).
\end{aligned}$$

From this we easily deduce the expression of an explicit norm for $\mathbb{H}(\Sigma)$, using the matrix $\mathrm{D}^{\#4}_n := \mathrm{diag}(\mathrm{D}_n,\mathrm{D}_n,\mathrm{D}_n,\mathrm{D}_n)$. Next, we need to introduce intermediate notations for the predominant behavior of two key matrices coming into play in the local-MTF formulation, namely

$$\begin{aligned}
&\mathrm{MTF}^\infty_{\mathrm{loc}}[n] := (\mathrm{T}^{\#4}_n)^{-1}\mathrm{MTF}^\infty_{\mathrm{loc}}\mathrm{T}^{\#4}_n \underset{n\longrightarrow+\infty}{\sim} \mathrm{MTF}_{\mathrm{loc}}[n],\\
&\tilde{\mathrm{A}}^j_{\kappa_j,\mu_j}[n] := (\mathrm{T}^{\#2}_n)^{-1}\tilde{\mathrm{A}}^j_{\kappa_j,\mu_j}\mathrm{T}^{\#2}_n \underset{n\longrightarrow+\infty}{\sim} \mathrm{A}^j_{\kappa_j,\mu_j}[n].
\end{aligned} \tag{3.7.3}$$

Since we need to rewrite this formulation variationally, we start by inspecting how the duality pairing decomposes on spherical harmonics. First of all, according to (3.4.1), observe that $\int_\Gamma (n_j\times\mathbf{X}^{\parallel}_{n,m})\cdot\mathbf{X}^{\parallel}_{n,m}d\sigma = \int_\Gamma(n_j\times\mathbf{X}^{\times}_{n,m})\cdot\mathbf{X}^{\times}_{n,m}d\sigma = 0$ and $\int_\Gamma(n_0\times\mathbf{X}^{\times}_{n,m})\cdot\mathbf{X}^{\parallel}_{n,m}d\sigma = 1$. As a consequence, considering the vector fields $u(x) := \mathbf{X}^{\#2}_{n,m}(x)\cdot u$ and $v(x) := \mathbf{X}^{\#2}_{n,m}(x)\cdot v$ where $u,v\in\mathbb{C}^2$, we have

$$[u,v]_{\Gamma_0} = v^\top \mathrm{M}\, u,$$
$$\text{and} \quad \mathrm{M} := \begin{bmatrix} 0 & 0 & 0 & +1 \\ 0 & 0 & -1 & 0 \\ 0 & +1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}. \tag{3.7.4}$$

Observe that $\mathrm{M}^\top = \mathrm{M}$. Since $[u,v]_{\Gamma_0} = -[u,v]_{\Gamma_1}$, we obtain a global matrix expression for the pairing on the multi-trace space: for $u(x) := \mathbf{X}^{\#4}_{n,m}(x)\cdot u$ and $v(x) := \mathbf{X}^{\#4}_{n,m}(x)\cdot v$ where $u, v \in \mathbb{C}^4$, we have

$$[\![u,v]\!]_\Sigma = v^\top \mathbb{M}\, u \quad \text{and} \quad \mathbb{M} := \begin{bmatrix} +\mathrm{M} & 0 \\ 0 & -\mathrm{M} \end{bmatrix}. \tag{3.7.5}$$

To examine coercivity of local MTF on the sphere, we need to study the coercivity of the matrix $\mathbb{M}\cdot \mathrm{MTF}_{\mathrm{loc}}[n]$ as $n \longrightarrow \infty$. If we look at the asymptotic behavior of this matrix, taking account of the results of Section 3.5, we obtain the expression

$$(-1)^j\, \mathrm{M}\cdot \tilde{\mathrm{A}}^j_{\kappa_j,\mu_j}[n] = \begin{bmatrix} \dfrac{n}{\omega\mu_j} & 0 & 0 & 0 \\ 0 & -\dfrac{\omega\epsilon_j}{n} & 0 & 0 \\ 0 & 0 & \dfrac{n}{\omega\epsilon_j} & 0 \\ 0 & 0 & 0 & -\dfrac{\omega\mu_j}{n} \end{bmatrix}. \tag{3.7.6}$$

Let us introduce a diagonal matrix $\theta \in \mathbb{R}^{2\times 2}$ defined by $\theta = \mathrm{diag}(+1,-1)$, and define the matrix $\Theta := \mathrm{diag}(\theta,\theta,\theta,\theta) \in \mathbb{R}^{4\times 4}$. From Expression (3.7.6), it clearly follows that $(-1)^j\, \mathrm{M}\cdot \tilde{\mathrm{A}}^j_{\kappa_j,\mu_j}[n]\cdot\Theta$ is a real valued diagonal positive definite matrix. On the other hand, it holds $(\mathbb{M}\cdot\Theta)^\top = -\mathbb{M}\cdot\Theta$. As a consequence, we finally conclude that there exists $c > 0$ independent of n such that

$$\Re e\{\mathrm{U}^\top \cdot \mathrm{MTF}^\infty_{\mathrm{loc}}[n]\cdot\Theta\cdot\overline{\mathrm{U}}\} \geq c\, \mathrm{U}^\top\cdot \mathrm{D}^{\#4}_n \cdot \overline{\mathrm{U}},$$

for all $\mathrm{U}\in\mathbb{C}^8$ and $n \geq 0$. Since the constant $c > 0$ is independent of n, summing such inequality over n, and taking account that $\mathrm{MTF}^\infty_{\mathrm{loc}}$ is the asymptotic behavior of $\mathrm{MTF}^\infty_{\mathrm{loc}}[n]$, we finally obtain the following coercivity statement.

Theorem 3.2. *There exists a compact operator $\mathscr{K} : \mathbb{H}(\Sigma) \longrightarrow \mathbb{H}(\Sigma)$ and a constant $\mathrm{C} > 0$ such that*

$$\Re e\{[\![(\mathrm{MTF}_{\mathrm{loc}} + \mathscr{K})u, \Theta(\overline{u})]\!]\} \geq \mathrm{C}\|u\|^2_{\mathbb{H}(\Sigma)},$$

for all $u \in \mathbb{H}(\Sigma)$.

3.8 Conclusions of the chapter

In this chapter we have studied the multi-trace formulation for Maxwell equation posed on a particular domain setting. We have proven that the local multi-trace operator is injective and derived an explicit formula for the corresponding boundary integral operator. For our case study, we have shown that the spectrum of the local-MTF operator is located in accumulation clusters, which is in agreement to the results obtained for the local-MTF operator in the previous chapter. The theory and numerical

experiments suggest that local-MTF is a stable formulation suitable for Maxwell equations, we have established this by proving a generalized Gårding inequality for the local-MTF formulation on the unit sphere. The results from this chapter shed light for the search of efficient preconditioners and generalization of the work done in this thesis for general problems in electromagnetism.

Part II

Low-rank approximations

CHAPTER 4

Introduction to Low-Rank approximations

4.1 Preliminaries

Many applications in linear algebra, matrix analysis, and statistics require to approximate a given matrix $A \in \mathbb{R}^{m\times n}$ by a rank-k matrix with $k \ll \min(m, n)$. The best approximation can be computed via the singular value decomposition (SVD) by using state-of-the-art routines, such as `dgesvj` (or `cgesvj` for complex matrices) from LAPACK [ABB+99] which has low backward error and high relative accuracy [DV08a, DV08b]. The main drawback of the SVD is that it requires $14mn^2 + 8n^3$ arithmetic operations [GVL96], and modern attempts to construct faster and accurate low-rank approximations have been made using deterministic and randomized algorithms; among them: QR-based factorizations [GE96, DGGX15], subspace iteration [Gu15], Monte-Carlo algorithms [FKV04] and random projections [VM17, Mar18]. The work by Halko, Martisson, and Tropp [HMT11] unifies several randomized approximation methods and presents state-of-the-art algorithms for approximating the SVD.
In this chapter we present classical low-rank approximation techniques to which we shall refer in the following two chapters. We are mainly interested in two methods from which several kind of algorithms for low-rank approximation and matrix compression can be derived: the truncated QR factorization and randomized subspace iteration. On the first hand, a truncated QR factorization produces a low-rank approximation (to which we refer to as low-rank QR approximation), of the form (see §4.3 for details)

$$A \approx QRP^{T}, \tag{4.1.1}$$

where $P \in \mathbb{R}^{n\times n}$ is a permutation matrix, $Q \in \mathbb{R}^{m\times k}$ has orthonormal columns and $R \in \mathbb{R}^{k\times n}$ is upper trapezoidal. The most common method to obtain P is via the classical truncated QR with column pivoting (QRCP) factorization [GVL96, Sec. 5.4.1]], which is based on interchanging columns for maximizing

the diagonal entries of R. On the other hand, low-rank randomized subspace iteration methods have recently gained attention for their good performance to approximate a matrix and its singular values with tight probabilistic bounds [Gu15, HMT11], they produce low-rank approximations of the form

$$\mathrm{A} \approx \mathrm{Q}\mathrm{B}_k, \tag{4.1.2}$$

where $\mathrm{Q} \in \mathbb{R}^{m\times l}$, with $k < l < \min(m,n)$, is the orthogonal basis of matrix $(\mathrm{A}\mathrm{A}^{\mathrm{T}})^q\mathrm{A}\Omega$, q is a small integer parameter, $\Omega \in \mathbb{R}^{n\times l}$ is a random matrix, and B_k is the rank-k truncated SVD of $\mathrm{Q}^{\mathrm{T}}\mathrm{A}$, see §4.4 for details.

Algorithms to construct low-rank approximations via QR factorizations and subspace iteration, intrinsically inherit the properties of a complete matrix factorization, therefore they can take the desired approximation rank k as input, see *e.g.* [GE96, Alg. 4], or they can return k as output by using a stopping criterion to reveal the rank of A; for instance, by requiring $\|\mathrm{A} - \mathrm{Q}\mathrm{R}\mathrm{P}^{\mathrm{T}}\|_2 \le \epsilon$, where ϵ is the machine epsilon, see *e.g.* [GVL96, Alg. 5.4.1] and [GE96, Alg. 5].

A few words on notations

In this chapter, $\mathrm{A} \in \mathbb{R}^{m\times n}$ refers to a (not necessarily square $m \neq n$) real matrix. However, the basic background and main results also hold true for $\mathrm{A} \in \mathbb{C}^{m\times n}$ to which we shall refer in Chapter 6. In the following, we use MATLAB notation for representing matrix and vector operations. The $m \times m$ identity matrix shall be denoted by I_m. We consider the following matrix norms,

$$\|\mathrm{A}\|_2 := \sup\{\|\mathrm{A}x\|_2 \,:\, x \in \mathbb{R}^n \quad \text{with} \quad \|x\|_2 = 1\}, \qquad \text{Spectral norm},$$

$$\|\mathrm{A}\|_{\mathrm{F}} := \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |\mathrm{A}(i,j)|^2}, \qquad \text{Frobenius norm},$$

$$\|\mathrm{A}\|_{\max} := \max_{i,j} |\mathrm{A}(i,j)|, \qquad \text{Maximum (or Chebyshev) norm}.$$

A matrix norm $\|\cdot\|$ is said to be unitarily invariant if $\|\mathrm{A}\| = \|\mathrm{U}\mathrm{A}\mathrm{V}\|$ for all orthogonal matrices $\mathrm{U} \in \mathbb{R}^{m\times m}$ and $\mathrm{V} \in \mathbb{R}^{n\times n}$, *e.g.* $\|\cdot\|_2$ and $\|\cdot\|_{\mathrm{F}}$ hold this condition.

Finally, let $\mathrm{ran}(\mathrm{A})$ denote the *range* of A, given as the vector subspace spanned by the columns of A. Given two matrices $\mathrm{W}_1, \mathrm{W}_2 \in \mathbb{R}^{m\times k}$ with orthonormal columns, consider $\mathrm{S}_i := \mathrm{ran}(\mathrm{W}_i)$ for $i = 1, 2$. Then, the angle between these two spaces is defined as $\angle(\mathrm{S}_1, \mathrm{S}_2) := \arcsin(\|\mathrm{W}_1\mathrm{W}_1^{\mathrm{T}} - \mathrm{W}_2\mathrm{W}_2^{\mathrm{T}}\|_2)$.

4.2 Best Low-rank Approximation

Definition 4.1. The rank of a matrix $\mathrm{A} \in \mathbb{R}^{m\times n}$ is defined as the maximal number of linearly independent rows or columns of A, we denote it as $r := \mathrm{rank}(\mathrm{A})$.

Definition 4.2. We denote

$$\mathbb{R}_k^{m\times n} := \{\mathrm{B} \in \mathbb{R}^{m\times n} \,:\, \mathrm{rank}(\mathrm{B}) \le k\}, \tag{4.2.1}$$

the set of real matrices having at most rank-k.

The SVD decomposition states that A can be decomposed into a sum of rank-one matrices, see *e.g.* [HJ91, Thm. 3.1.1], this is

$$\mathrm{A} = \sum_{i=1}^{r} u_i \sigma_i v_i^{\mathrm{T}} \equiv \mathrm{U}_r \Sigma_r \mathrm{V}_r^{\mathrm{T}}, \tag{4.2.2}$$

where the matrices $\mathrm{U}_r = [u_1, \cdots, u_r] \in \mathbb{R}^{m\times r}$ and $\mathrm{V}_r = [v_1, \cdots, v_r] \in \mathbb{R}^{n\times r}$ are orthogonal, and their columns are the left and right singular vectors respectively. For matrix $\Sigma_r := \mathrm{diag}(\sigma_1, \cdots, \sigma_r) \in \mathbb{R}^{r\times r}$, we assume $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ so that Σ is uniquely determined by A. The values σ_i are the singular values of A. Next, assuming that $\mathrm{rank}(\mathrm{A}) \geq k$, the k-truncated singular value approximation of A is given as

$$\mathrm{A}_k := \sum_{i=1}^{k} u_i \sigma_i v_i^{\mathrm{T}} \equiv \mathrm{U}_k \Sigma_k \mathrm{V}_k^{\mathrm{T}}. \tag{4.2.3}$$

and the Moore-Penrose pseudoinverse of A_k is given as

$$\mathrm{A}_k^{\dagger} := \sum_{i=1}^{k} v_i \sigma_i^{-1} u_i^{\mathrm{T}} \equiv \mathrm{V}_k \Sigma_k^{-1} \mathrm{U}_k^{\mathrm{T}}. \tag{4.2.4}$$

Note that for the spectral and Frobenius norms it holds

$$\|\mathrm{A}_k - \mathrm{A}\|_2 = \sigma_{k+1}, \qquad \|\mathrm{A}_k - \mathrm{A}\|_{\mathrm{F}} = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_r^2}. \tag{4.2.5}$$

Theorem 4.1. *(Mirsky, [Mir60, Thm. 2]) Consider the matrix* $\mathrm{A} \in \mathbb{R}^{m\times n}$*, with singular triplets* (u_i, σ_i, v_i) *for* $i = 1, \cdots, \min(m, n)$*. Then,* $\mathrm{A}_k = \sum_{i=1}^{k} u_i \sigma_i v_i^{\mathrm{T}}$ *is a solution of the following problem*

$$\begin{cases} \text{Find } \mathrm{B} \in \mathbb{R}_k^{m\times n} \text{ such that} \\ \qquad \|\mathrm{A} - \mathrm{B}\| \leq \|\mathrm{A} - \mathrm{C}\|, \quad \forall\, \mathrm{C} \in \mathbb{R}_k^{m\times n}, \end{cases} \tag{4.2.6}$$

where $\|\cdot\|$ *stands for any unitarily invariant norm.*

Albeit optimal, the truncated SVD is expensive to obtain, indeed it costs $\mathcal{O}(\min(mn^2, m^2 n))$ arithmetic operations. In the following sections, we present QRCP and subspace iteration methods which are $\mathcal{O}(\min(m, n))$ asymptotically faster to obtain than the truncated SVD; and in Chapter 6, we present CUR approximations having linear $\mathcal{O}(\max(m, n))$ cost to compute low-rank approximations of BEM matrices to be introduced later on.

Remark 4.1. Note that problem (4.2.6) has a unique solution when the Frobenius norm is used if and only if $\sigma_k \neq \sigma_{k+1}$, cf. [EG36]. If the spectral norm is used, as explained in [Gu15], the solution of problem (4.2.6) is not unique. For instance, for any $0 \leq \theta \leq 1$ the matrix $\mathrm{B} = \mathrm{A}_k - \theta\sigma_{k+1}\mathrm{U}_k\mathrm{V}_k^{\mathrm{T}}$ is a solution.

Remark 4.2. If the maximum norm is used then A_k is not, in general, a solution of (4.2.6). For this case, in §6.2, we present explicit suboptimal solutions via skeleton approximations, *c.f.* Theorem 6.1, which we will use as a benchmark for comparisons in Chapter 6. Recently, progress on showing the NP-hardness of finding the solution of (4.2.6) has been reported [GV18]. Next figure shows tests performed on a 100×100 random matrix approximated by a hundred rank-k random matrices, $k = 10$. It is shown that there are many rank-k matrices B for which $\|\mathrm{A} - \mathrm{B}\|_{\max} \leq \|\mathrm{A} - \mathrm{A}_k\|_{\max}$.

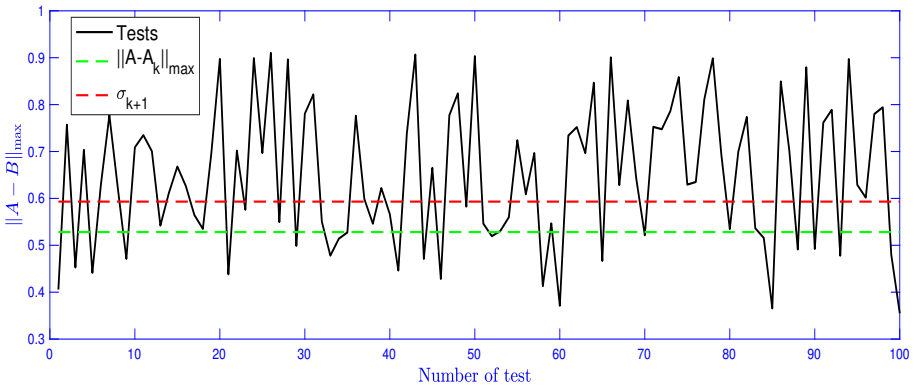


Figure 4.1: Error in maximum norm of rank $k = 10$ approximations of a 100×100 random matrix A.

Next theorem presents some useful inequalities that will be helpful in the following chapters.

Theorem 4.2. *(Horn and Jonhson, [HJ91, Thm. 3.3.16]) Let* $A, B \in \mathbb{R}^{m\times n}$ *and* $q = \min(m, n)$. *Then, for* $1 \le i, j$ *and* $i + j \le q + 1$, *the following inequalities hold,*

$$\sigma_{i+j-1}(AB^T) \le \sigma_i(A)\sigma_j(B), \tag{4.2.7}$$

$$\sigma_{i+j-1}(A + B) \le \sigma_i(A) + \sigma_j(B). \tag{4.2.8}$$

4.3 Low-Rank Approximation using Pivoted QR Factorization

A pivoted truncated QR factorization can be obtained by several algorithms, *e.g.* [GVL96, Alg. 5.4.1], it has the form

$$AP_c = \begin{array}{c} \\ m \end{array} \begin{array}{c} \begin{array}{cc} k & m-k \end{array} \\ \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{array}{c} k \\ m-k \end{array} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \end{array}, \tag{4.3.1}$$

where P_c is a permutation matrix, $Q = [Q_1, Q_2]$ is an orthogonal matrix, R_{11} is an upper triangular matrix, and R_{22} is not necessarily upper triangular. The rank-k approximation is naturally obtained as

$$\xi_k := Q_1 \begin{bmatrix} R_{11} & R_{12} \end{bmatrix} P_c^T, \tag{4.3.2}$$

and the approximation error in the spectral norm (note that it holds for any other unitarily invariant norm) is given as

$$\|A - \xi_k\|_2 = \|Q_2[0 \quad R_{22}]P_c^T\|_2 = \|[0 \quad R_{22}]\|_2 = \|R_{22}\|_2. \tag{4.3.3}$$

Bounds for the error in (4.3.3) depend on the technique used to select P_c. Well established pivoting techniques lead to bounds of the form

$$\|R_{22}\|_2 \le f(k, n)\,\sigma_{k+1}, \tag{4.3.4}$$

$$\|R_{11}^{-1}R_{12}\|_{\max} \le g(k), \tag{4.3.5}$$

$$\sigma_k(A) \le f(k, n)\sigma_k(R_{11}), \tag{4.3.6}$$

where $f(k,n)$ and $g(k)$ are explicitly known functions of k and n, see *e.g.* [DGGX15, GE96, PT99]. For a compilation of some of the different algorithms of this kind and their computational complexity see [BMD09, Tbl. 2]. The permutation P_c can also be found by applying QRCP on ΩA, where $\Omega \in \mathbb{R}^{l\times m}$ is a random matrix, two randomized algorithms for this purpose (without theoretical bounds on the error) are RQRCP [DG17] and HQRRP [MGHV17].

The following table presents two classical QR algorithms that we shall use for the scope of this thesis, the classical QR with column pivoting [GVL96, Alg. 5.4.1] and the strong RRQR [GE96, Alg. 4].

Table 4.1: Error bound for classical QR algorithms for a matrix $A \in \mathbb{R}^{m\times n}$, where k is the truncation rank and ν is a constant.

Algorithm	Reference	$f(k,n)$	$g(k)$	Time
Column Pivoting QRCP	[GVL96, Alg. 5.4.1]	$2^k\sqrt{n-k}$	2^{k-1}	$\mathcal{O}(mnk)$
Strong RRQR	[GE96, Alg. 4]	$\sqrt{1+\nu^2 k(n-k)}$	ν	$\mathcal{O}\left((m+n\log_\nu n)n^2\right)$

In Lemma 4.1 we provide a new bound for QRCP which is 2/3 of the bound presented in Table 4.1, its proof is given in appendix B.2 and it helps to understand the origin of the exponential factor.

Lemma 4.1. *Consider the truncated QRCP factorization (4.3.1). Then,*

$$\|R_{22}\|_2 \le \sqrt{1+2k+\sum_{j=1}^{k-1} 4^j(k-j)}\sqrt{n-k}\,\sigma_{k+1}, \tag{4.3.7}$$

As mentioned above, there exist different algorithms to select the permutation for a truncated QR factorization. We refer the interested reader to Chapter A.4, where we present a communication avoiding parallel algorithm that selects this permutation based on a technique called *tournament pivoting*.

Approximation error for an arbitrary permutation

To conclude this section, we analyze the error of approximation using a low-rank QR approximation with an arbitrary permutation P. Consider a truncated QR factorization of A,

$$AP = QR = \begin{array}{c} \\ m \end{array}\begin{array}{c} \begin{array}{cc} k & m-k \end{array} \\ \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \end{array} \begin{array}{c} \begin{array}{c} k \\ m-k \end{array} \\ \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \end{array}\begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \\ \end{array}. \tag{4.3.8}$$

Next, note that $Q_1R_{11} = AP(:, 1:k)$ and that the error of a QR approximation given in (4.3.3) can also be obtained as

$$\|R_{22}\|_2 = \|(I_m - Q_1Q_1^T)A\|_2, \tag{4.3.9}$$

where $Q_1Q_1^T$ is the orthogonal projector over the subspace generated by the first k columns of AP. This is true since

$$\|(I_m - Q_1Q_1^T)A\|_2 = \|(Q^T - Q^TQ_1Q_1^T)QRP^T\|_2 = \|R - Q^TQ_1Q_1^TQR\|_2,$$

and

$$Q^TQ_1Q_1^TQ = \begin{bmatrix} I_k \\ Q_2^TQ_1 \end{bmatrix} \begin{bmatrix} I_k & Q_1^TQ_2 \end{bmatrix} = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.3.10}$$

which holds since the columns of Q_1 and Q_2 are mutually orthogonal.

Note that from the previous analysis, a simple bound can be obtained for the error using a general permutation, this is

$$\|R_{22}\|_{\max} \le \|R_{22}\|_2 = \|(I_m - Q_1Q_1^T)A\|_2 \le \|I_m - Q_1Q_1^T\|_2\|A\|_2 \le \|A\|_2 \le \sqrt{mn}\|A\|_{\max}. \tag{4.3.11}$$

The following lemma, using an assumption on the right singular vectors, provides a bound of type (4.3.4) for the approximation error when using an arbitrary permutation to compute a low-rank QR approximation.

Lemma 4.2. *Let $A \in \mathbb{R}^{m\times n}$, consider its truncated QR factorization,*

$$AP = QR = \begin{array}{c} \\ m \end{array} \begin{array}{c} \begin{array}{cc} k & m-k \end{array} \\ \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \end{array} \begin{array}{c} \\ \begin{array}{c} k \\ m-k \end{array} \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix} \end{array},$$

where $P \in \mathbb{R}^{n\times n}$ is an arbitrary permutation. Define

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix} := (V^TP)(:, 1 : k),$$

where $V \in \mathbb{R}^{n\times n}$ is the matrix containing the right singular vectors of A, as defined in §4.2. Assuming that Ω_1 is non-singular, then

$$\|R_{22}\|_2 \le \sqrt{1 + \|\Omega_2\Omega_1^{-1}\|_2^2}\,\sigma_{k+1}(A). \tag{4.3.12}$$

Proof. Consider the SVD decomposition $A = U\Sigma V^T$. Define the matrices

$$\bar{A} := AP = U\Sigma\tilde{V}^T, \qquad \text{and} \qquad Y := AP\Omega = U\Sigma\left(\tilde{V}^T(:, 1 : k)\right) \equiv U\Sigma\begin{bmatrix} \Omega_1 \\ \Omega_2 \end{bmatrix}.$$

Next, note that Y is the matrix consisting of the first k columns of $\bar{A}$, and its orthogonal projector is $Q_1Q_1^T$. Then, as showed in (4.3.9) we have $\|R_{22}\|_2 = \|(I_m - Q_1Q_1^T)A\|_2$ and by applying [HMT11, Thm. 9.1] on $\bar{A}$, we get

$$\|(I_m - Q_1Q_1^T)A\|_2 = \|(I_m - Q_1Q_1^T)\bar{A}\|_2 \le \sqrt{1 + \|\Omega_2\Omega_1^{-1}\|_2^2}\,\sigma_{k+1}(A).$$

□

4.4 Low-rank Approximation using Subspace Iteration

Methods based on subspace iteration [GVL96, Ch. 7, 8] have shown to produce good rank-k approximation with cost between $\mathcal{O}(mn\log(k))$ and $\mathcal{O}(mnk)$, see for example [DG17, HMT11, MRT06]. Algorithm 6 presents the basic subspace iteration method. This algorithm is well known in the literature and versions of it have been presented by different authors, see for example [Gu15, HMT11]. It takes as input an $m \times n$ matrix A, a small integer q (that is usually taken as $q = 1$ or $q = 2$), and a random matrix $\Omega \in \mathbb{R}^{n\times l}$; for the scope of this thesis, we create Ω using normal or uniform distributions, *e.g.* using, respectively, routines `randn` or `rand` from MATLAB.

Algorithm 6 iteratively computes $Y = (AA^T)^q A\Omega$ and then gets its orthogonal basis $Q \in \mathbb{R}^{m\times l}$, returning the low-rank matrix QB_k, where B_k is the rank k truncated SVD of Q^TA.

Data: $A \in \mathbb{R}^{m\times n}$, a fixed rank value k, $\Omega \in \mathbb{R}^{n\times l}$, with $\min(m, n) > l \geq k$, and a small integer q
Result: rank-k approximation of A

1 Compute $Y = A\Omega$ and QR factorize $QR = Y$;
2 **for** $j = 1 \longrightarrow q$ **do**
3 $\quad Y = A^*Q$; QR factorize $QR = Y$;
4 $\quad Y = AQ$; QR factorize $QR = Y$;
5 **end**
6 Form $B = Q^TA$;
7 Find B_k, the rank-k truncated SVD of B ;
8 Return $\xi_k := QB_k$.

Algorithm 1: Randomized subspace iteration, SSITER

Note that Algorithm 6 could stop at line 6 and return QQ^TA as the low-rank approximation of A. Indeed it is known in the literature (see e.g. [HMT11]) that for any matrix $B \in \mathbb{R}^{l\times n}$, it holds $\|A - QQ^TA\|_2 \leq \|A - QB\|_2$. Then, $\|A - QQ^TA\|_2 \leq \|A - \xi_k\|_2$. Hence, computing ξ_k provides a less accurate low-rank approximation than QQ^TA, in terms of the norm of the approximation error. However, obtaining ξ_k can provide better approximation of the singular values [Gu15]. In Theorem 4.3, we prove that the first k columns of Q converge to the first k left singular vectors of A at an exponential rate.

Approximation error analysis
Considering that we choose the approximation ξ_k from Algorithm 6 for matrix $A = U\Sigma V^T$ with decreasing singular values $\sigma_1 > \sigma_2 > \cdots > \sigma_k > \sigma_{k+1}$. It is possible to obtain rapidly converging approximations for A and its singular values, provided that the matrix $\widehat{\Omega}$ defined as

$$\widehat{\Omega} := V^T\Omega = \begin{matrix} l-p \\ n-l+p \end{matrix} \begin{bmatrix} \widehat{\Omega}_1 \\ \widehat{\Omega}_2 \end{bmatrix}, \qquad 0 \leq p \leq l-k, \tag{4.4.1}$$

is such that its submatrix $\widehat{\Omega}_1$ is full row rank; p is known as oversampling parameter. In fact, we have the bounds (cf. [Gu15, Thms. 4.3 and 4.4]),

$$\sigma_j \geq \sigma_j(B_k) \geq \frac{\sigma_j}{\sqrt{1 + \psi^2\|\widehat{\Omega}_2\|_2^2\,\|\widehat{\Omega}_1^\dagger\|_2^2}}, \tag{4.4.2}$$

$$\|A - \xi_k\|_2 \leq \sqrt{\sigma_{k+1}^2 + \omega^2 \|\widehat{\Omega}_2\|_2^2 \, \|\widehat{\Omega}_1^\dagger\|_2^2}, \tag{4.4.3}$$

where $\psi = \left(\frac{\sigma_{l-p+1}}{\sigma_j}\right)^{2q+1}$, $\omega = \sqrt{k}\sigma_{l-p+1}\left(\frac{\sigma_{l-p+1}}{\sigma_k}\right)^{2q}$, $0 \leq p \leq l-k$ and $\widehat{\Omega}_1^\dagger$ is the Moore-Penrose pseudoinverse of $\widehat{\Omega}_1$.

When Ω is a random Gaussian matrix, meaning that its entries are independent standard normal variables of unit-variance and zero mean, it holds that $\widehat{\Omega}_1$ is still a Gaussian matrix [HMT11], and it is proven that if $l - p \geq 2$ then $\widehat{\Omega}_1$ has full rank with probability 1 [Gu15, Lem. 5.2]. For this case, setting $l = 2k$, $q = 0$, Algorithm 6 produces a rank-k approximation with expected error [HMT11, Thm.1.2],

$$\mathbb{E}\|A - \xi_k\|_2 \leq \left(2 + 4\sqrt{\frac{2\min(m,n)}{k-1}}\right)\sigma_{k+1}. \tag{4.4.4}$$

Algorithm 6 works very well in practice and has computational complexity of $\mathcal{O}(mnl)$. In the next chapter we construct approximations based on good approximations of the singular vectors; in this context, the following theorem proves a result for the convergence of singular vectors when using Algorithm 6.

Theorem 4.3. *Consider $\Omega \in \mathbb{R}^{m\times l}$ and $A \in \mathbb{R}^{m\times n}$, with SVD decomposition $A = U\Sigma V^T$. Consider the QR factorization $QR = (AA^T)^q A\Omega$ and let $Q_k = [q_1, \cdots, q_k]$ and $U_k = [u_1, \cdots, u_k]$ be matrices constructed with the first k columns of Q and U respectively. Considering the partition*

$$V^T\Omega := \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & l-k \end{array} \\ \begin{bmatrix} \Omega_\alpha & Z_1 \\ \Omega_\beta & Z_2 \end{bmatrix} \end{array},$$

if Ω_α is invertible, defining $\varphi = \angle(\mathrm{ran}(Q_k), \mathrm{ran}(U_k))$, then

$$\sin(\varphi) \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2q+1} \|\Omega_\beta \Omega_\alpha^{-1}\|_2.$$

Proof. First, consider the partitions

$$\Sigma = \begin{array}{c} \\ k \\ n-k \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} D_k & 0 \\ 0 & D_s \end{bmatrix} \end{array}, \quad U = \begin{array}{c} \\ m \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} U_k & U_s \end{bmatrix} \end{array}, \quad Q = \begin{array}{c} \\ m \end{array} \begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{bmatrix} Q_k & Q_s \end{bmatrix} \end{array}.$$

Next, analyzing the QR factorization we get

$$(AA^T)^q A\Omega = U\Sigma^{2q+1}V^T\Omega = QR = \begin{bmatrix} Q_k & Q_s \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix},$$

where $R_{11} \in \mathbb{R}^{k\times k}$. Hence,

$$\begin{bmatrix} U_k & U_s \end{bmatrix} \begin{bmatrix} D_k & 0 \\ 0 & D_s \end{bmatrix}^{2q+1} \begin{bmatrix} \Omega_\alpha & Z_1 \\ \Omega_\beta & Z_2 \end{bmatrix} = \begin{bmatrix} Q_k & Q_s \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}. \tag{4.4.5}$$

Comparing the first k columns of both sides of equation (4.4.5), we get an embedded QR factorization of a matrix W defined as

$$W := \begin{bmatrix} U_k & U_s \end{bmatrix} \begin{bmatrix} D_k & 0 \\ 0 & D_s \end{bmatrix}^{2q+1} \begin{bmatrix} \Omega_\alpha \\ \Omega_\beta \end{bmatrix} = Q_k R_{11}. \tag{4.4.6}$$

Next, note that we search $\sin(\varphi) = \|U_k U_k^T - Q_k Q_k^T\|_2$, which by [GVL96, Thm. 2.6.1] is equivalent to

$$\sin(\varphi) = \|U_s^T Q_k\|_2.$$

Define the matrix

$$X := \Omega_\alpha^{-1} D_k^{-(2q+1)} \in \mathbb{R}^{k\times k}, \tag{4.4.7}$$

which is non-singular by assumption of the theorem, and consider the QR factorization of WX,

$$WX = U \begin{bmatrix} D_k & 0 \\ 0 & D_s \end{bmatrix}^{2q+1} \begin{bmatrix} \Omega_\alpha \\ \Omega_\beta \end{bmatrix} X = \tilde{Q}_k \tilde{R}_{11}. \tag{4.4.8}$$

where $\tilde{R}_{11} \in \mathbb{R}^{k\times k}$. Replacing (4.4.7) in (4.4.8), we get

$$\begin{bmatrix} I_k \\ D_s^{(2q+1)} \Omega_\beta \Omega_\alpha^{-1} D_k^{-(2q+1)} \end{bmatrix} = U^T \tilde{Q}_k \tilde{R}_{11} = \begin{bmatrix} U_k^T \tilde{Q}_k \\ U_s^T \tilde{Q}_k \end{bmatrix} \tilde{R}_{11},$$

from which we deduce that $\tilde{R}_{11}^{-1} = U_k^T \tilde{Q}_k$. Next, let us compute

$$\|U_k U_k^T - \tilde{Q}_k \tilde{Q}_k^T\|_2 \equiv \|U_s^T \tilde{Q}_k\|_2 = \|D_s^{(2q+1)} \Omega_\beta \Omega_\alpha^{-1} D_k^{-(2q+1)} \tilde{R}_{11}^{-1}\|_2. \tag{4.4.9}$$

Equation (4.4.9) is important since by [Gu15, Lem. 4.1] we have that factorizations (4.4.6) and (4.4.8) have the property

$$Q_k Q_k^T = \tilde{Q}_k \tilde{Q}_k^T. \tag{4.4.10}$$

Finally, from (4.4.9), (4.4.10) and the fact that $\|\tilde{R}_{11}^{-1}\|_2 = \|U_k^T \tilde{Q}_k\|_2 \le 1$, we obtain

$$\sin(\varphi) = \|U_k U_k^T - Q_k Q_k^T\|_2 = \|U_k U_k^T - \tilde{Q}_k \tilde{Q}_k^T\|_2 \le \|D_s^{(2q+1)} \Omega_\beta \Omega_\alpha^{-1} D_k^{-(2q+1)}\|_2.$$

Hence,

$$\sin(\varphi) \le \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2q+1} \|\Omega_\beta \Omega_\alpha^{-1}\|_2.$$

□

The previous theorem shows that the subspace generated by the span of the k first columns of Q (obtained by Algorithm 6 applied to a matrix $A \in \mathbb{R}^{m\times n}$) converges exponentially in q to the subspace generated by the first k left singular vectors, provided $\sigma_k > \sigma_{k+1}$. This agrees with the exponential rates obtained for the convergence of singular values and approximation error (4.4.2) and (4.4.3), and was predicted in a previous work [Gu15, Sec. 9].

Remark 4.3. When Ω_α and Ω_β are matrices with independent $N(0,1)$ Gaussian entries, the work developed by Edelman [Ede88] and Szarek [Sza91] tells us that, with high probability, $\|\Omega_\alpha^{-1}\|_2 \leq c_1\sqrt{k}$ and $\|\Omega_\beta\|_2 \leq c_2 \max(\sqrt{m-k}, \sqrt{k})$. Hence, considering $m - k \geq k$, we get that

$$\sin(\varphi) \leq \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2q+1} \|\Omega_\beta \Omega_\alpha^{-1}\|_2 \leq C_\Omega \sqrt{k(m-k)} \left(\frac{\sigma_{k+1}}{\sigma_k}\right)^{2q+1},$$

where $C_\Omega = c1 \cdot c2 > 0$, with $c_1 = \mathcal{O}(1)$, $c_2 = \mathcal{O}(\frac{m-k}{k})$. This shows that the angle converges to zero with an exponential rate up to a small rational factor on m and k. Other bounds can be obtained by using another kind of random matrices such as the centered sub-Gaussian random matrices [Rud08] and the Wigner random matrices [TV12]. Refer to [OVW16] for a recent survey on the different types of random matrices and their spectral properties.

4.5 Conclusions of the chapter

In this chapter we have presented classical methods and algorithms that we shall use in the following chapters. In one hand one, the low-rank QR factorizations, for which we have discussed exiting pivoting techniques and their impact on the approximation error. We have provided a bound for an arbitrary choice of the pivoting strategy and improved the existent bound for QRCP. On the other hand, for low-rank approximations created with subspace iteration methods, we have proved a bound on the convergence of singular vectors which is in agreement with recent bounds obtained for the low-rank approximation error and singular values estimates described in the literature.

CHAPTER 5

Affine low-rank approximations

5.1 Preliminaries

This chapter presents a new approach to construct low-rank approximations using projections into affine subspaces. The objective of such kind of approximations is to increase the accuracy of classical low-rank algorithms by adding a small number of arithmetic operations. An affine approximation of rank at most k has the form

$$A \approx \xi_k := \Big(\sum_{j=1}^{k-1} q_j q_j^T\Big) A(I_n - zz^T) + (Az)z^T, \tag{5.1.1}$$

where $z \in \mathbb{R}^n$ has all entries equal to $1/\sqrt{n}$ and $(Az)z^T$ can be seen as a translation matrix. We justify the choice of z via a geometric argument, and then we bound the approximation error. We provide an algorithm referred to as ALORA (short for affine low-rank approximation) that returns an approximation of type (5.1.1) by modifying any low-rank approximation method; in particular, we call ALORA to modify QR with column pivoting and subspace iteration algorithms.

In the literature, we can find several techniques to increase the accuracy of classical low-rank algorithms. For QRCP, a natural way is to improve the choice of the pivoting technique, see *e.g.* [GE96], while for subspace iteration, we can improve accuracy by increasing l or q or both in Algorithm 6, see *e.g.* [Gu15]. Such techniques typically increase the algorithm's computational cost by a considerable amount. The advantage of using ALORA resides in its capacity to receive any low-rank algorithm and perform inexpensive and simple calculations without changing the permutation technique or adding

post-processing cost.

We analyze the cases where it would behoove to use an affine low-rank approximation, for this aim we provide a correlation coefficient $\mathscr{G}(A)$. We show that matrices with exponentially decreasing singular values have high correlation coefficient and provide simple formulas to approximate their norm and first singular vectors.
For ease of presentation, in this chapter we consider $A \in \mathbb{R}^{m\times n}$. However, the theory and results can directly be extended to rectangular complex matrices, by making small appropriate changes in the definitions, statements, and proofs. We assume that k is given as input, since this choice allows us to show the convergence evolution for increasing rank values in the numerical tests. Note that ALORA can also return k as output, see section 5.2.2 for details.

This chapter is structured as follows. Section 5.2 presents the concept of affine low-rank approximation, it starts by analyzing a general framework for constructing a low-rank approximation via projections of rows and columns. The analysis made in this section leads to the construction of ALORA algorithm. Next, in Section 5.3 we analyze matrices for which an affine approximation would be advantageous, we also introduce a matrix correlation coefficient using statistical tools. Section 5.4 presents and discusses several numerical experiments to validate ALORA algorithm using a set of challenging matrices arising from different interesting fields. Finally, Section 5.5 concludes the chapter.

5.2 Affine Low-rank Approximation

The main objective of this section is to present a low rank approximation of A, which has the form

$$\xi_k := \Big(\sum_{j=1}^{k-1} q_j q_j^{\mathrm{T}}\Big)\mathrm{A}(\mathrm{I}_n - zz^{\mathrm{T}}) + (\mathrm{A}z)z^{\mathrm{T}}, \quad \forall k = \{1, \cdots, \mathrm{rank}(\mathrm{A})\},$$

where $q_j \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ are unitary vectors, i.e. multiplying A by two orthogonal projectors on the left and the right and adding a translation matrix. With this aim, we first review a general framework to construct low rank approximations by projecting the columns and rows of A.

5.2.1 Low-Rank Approximation as Projection of Rows and Columns

Consider the matrix $A \in \mathbb{R}^{m\times n}$, with $\mathrm{rank}(A) > k$, and let $\|\cdot\|$ be any unitarily invariant norm. Then, let us construct a low rank approximation using a truncated QR factorization, *c.f.* (4.3.2), this is

$$\mathrm{A} \approx \bar{\mathrm{Q}}\bar{\mathrm{R}} = \sum_{j=1}^{k} q_j r_j^{\mathrm{T}} =: \xi_k, \tag{5.2.1}$$

where $\bar{\mathrm{Q}} \in \mathbb{R}^{m\times k}$ and $\bar{\mathrm{R}} \in \mathbb{R}^{k\times n}$, and q_j and r_j are the j-th columns of $\bar{\mathrm{Q}}$ and $\bar{\mathrm{R}}^{\mathrm{T}}$ respectively. Note that this approximation can also be written as

$$\xi_k = \bar{\mathrm{Q}}\bar{\mathrm{Q}}^{\mathrm{T}}\mathrm{A} = \Big(\sum_{j=1}^{k} q_j q_j^{\mathrm{T}}\Big)\mathrm{A},$$

and hence, the approximation error is given by

$$\|A - \xi_k\| = \|(I_m - \sum_{j=1}^{k} q_j q_j^T)A\| = \|\prod_{j=1}^{k} (\underbrace{I_m - q_j q_j^T}_{=:\mathscr{P}_j})A\|,$$

where the last equality can be easily proved by induction. Hence, the approximation error can be seen as the norm of the matrix obtained after applying k orthogonal projections, $\mathscr{P}_j$, to the columns of A.

In general, we can consider orthogonal matrices $W = [w_1, \cdots, w_k] \in \mathbb{R}^{m\times k}$ and $Z = [z_1, \cdots, z_k] \in \mathbb{R}^{n\times k}$, and use the orthogonal projectors WW^T and ZZ^T to construct

$$\bar{\xi}_k := WW^T A = (\sum_{j=1}^{k} w_j w_j^T)A, \quad \text{and} \quad \hat{\xi}_k := AZZ^T = A(\sum_{j=1}^{k} z_j z_j^T),$$

for which,

$$\|A - \bar{\xi}_k\| = \|\prod_{j=1}^{k} (I_m - w_j w_j^T)A\| \tag{5.2.2}$$

$$\|A - \hat{\xi}_k\| = \|A\prod_{j=1}^{k} (I_n - z_j z_j^T)\| = \|\prod_{j=1}^{k} (I_n - z_j z_j^T)A^T\|. \tag{5.2.3}$$

Then, the approximation errors (5.2.2) and (5.2.3) are, respectively, the norm of the matrices obtained after applying k orthogonal projections on the columns and rows of A. According to Theorem 4.1, if $w_j = u_j(A)$ or $z_j = v_j(A)$, for $j = 1, \cdots, k$, then the errors (5.2.2) and (5.2.3) attain their minimum among all possible choices of w_j and z_j.

In section 5.2.2 we contruct an approximation by mixing projections of rows and columns. This is, we build an approximation of type

$$\xi_{\bar{r}} := (\sum_{j=1}^{s} w_j w_j^T)A(\sum_{j=1}^{t} z_j z_j^T),$$

where $\xi_{\bar{r}}$ has at most rank $\bar{r} = \min(s, t, \text{rank}(A))$. The analysis of such approximation will make reference to the following lemma, in which we explore the relationship between the singular values of A and those of the matrix obtained by projecting the columns of A. Note that Lemma 5.1 still holds when considering projection of rows instead, by simply applying the same arguments on $Y^T = (I_n - ZZ^T)A^T$.

Lemma 5.1. *Consider* $A \in \mathbb{R}^{m\times n}$ *and an orthogonal matrix* $Z \in \mathbb{R}^{n\times t}$*, with* $t < \min(m, n)$*. Define the matrix* $Y = A(I_n - ZZ^T)$*, constructed by orthogonally projecting the columns of* A*. Then,*

$$\sigma_{k+t}(Y) \le \sigma_{k+t}(A) \le \sigma_k(Y).$$

Proof. The left inequality is verified by applying Theorem 4.2 on the product $A(I_n - ZZ^T)$ with $i = k + t$ and $j = 1$, since an orthogonal projection has unitary norm. To prove the right inequality, define $F := AZ \in \mathbb{R}^{m\times t}$, so that $Y = A - FZ^T$. Next, let Y_{k-1} be the rank $k - 1$ truncated SVD approximation of Y, hence

$$\sigma_k(\mathrm{Y}) = \|\mathrm{Y} - \mathrm{Y}_{k-1}\|_2 = \|\mathrm{A} - (\mathrm{Y}_{k-1} + \mathrm{F}\mathrm{Z}^{\mathrm{T}})\|_2 \geq \sigma_{k+t}(\mathrm{A}),$$

the last inequality holds since $\mathrm{Y}_{k-1} + \mathrm{F}\mathrm{Z}^{\mathrm{T}}$ is a matrix of rank at most $k + t - 1$. □

Corollary 5.2. *If $t = 1$, i.e. $\mathrm{Y} = \mathrm{A}(\mathrm{I}_n - zz^{\mathrm{T}})$, where $z \in \mathbb{R}^n$ is a unit vector, then*

$$\sigma_{k+1}(\mathrm{Y}) \leq \sigma_{k+1}(\mathrm{A}) \leq \sigma_k(\mathrm{Y}),$$

$$\mathrm{rank}(\mathrm{A}) - 1 \leq \mathrm{rank}(\mathrm{Y}) \leq \mathrm{rank}(\mathrm{A}).$$

5.2.2 Getting an Affine Low-Rank Approximation

Define the gravity center of A as

$$g := \frac{1}{n}\sum_{j=1}^{n} a_j, \tag{5.2.4}$$

where a_j is the j-th column of A. Also define the following matrix

$$\mathrm{Y} := \mathrm{A} - gc^{\mathrm{T}} = \mathrm{A}\underbrace{(\mathrm{I}_n - \frac{1}{n}cc^{\mathrm{T}})}_{=:\mathscr{P}}, \tag{5.2.5}$$

where $c = [1, \cdots, 1]^{\mathrm{T}} \in \mathbb{R}^n$. Next note that $\mathscr{P}$ is a rank $n - 1$ orthogonal projector, and let Y_k and A_k be the rank-k truncated SVD approximations of Y and A respectively. By applying Corollary 5.2, we get

$$\|\mathrm{A} - gc^{\mathrm{T}} - \mathrm{Y}_k\|_2 = \|\mathrm{Y} - \mathrm{Y}_k\|_2 = \sigma_{k+1}(\mathrm{Y}) \leq \sigma_{k+1}(\mathrm{A}) = \|\mathrm{A} - \mathrm{A}_k\|_2. \tag{5.2.6}$$

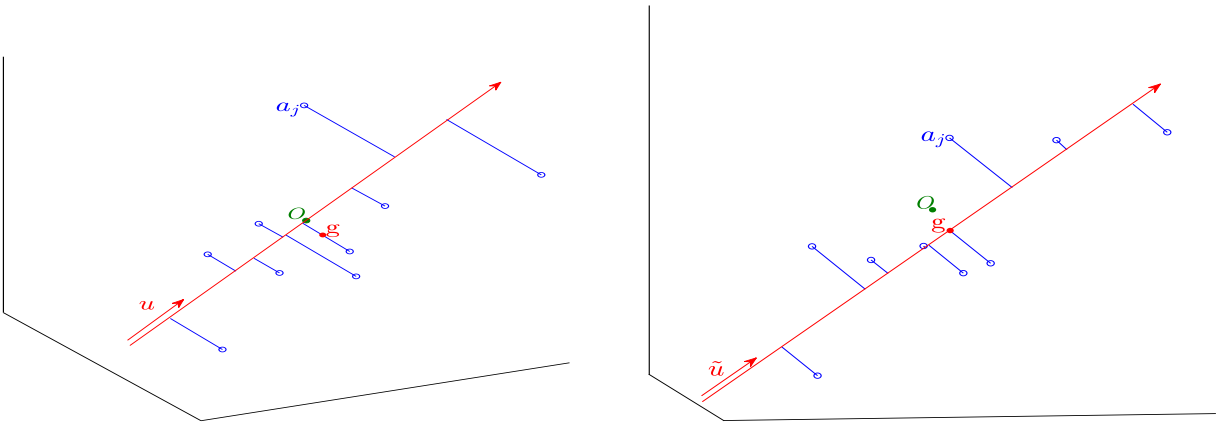
We can geometrically interpret (5.2.6) by defining the lines

$$\mathscr{L}_{\mathrm{A}}(\tau) = \tau u, \quad \tau \in \mathbb{R}, \tag{5.2.7}$$

$$\mathscr{L}_g(\tau) = g + \tau\tilde{u}, \quad \tau \in \mathbb{R}, \tag{5.2.8}$$

where u and $\tilde{u}$ are the first left singular vectors of A and Y respectively. Line $\mathscr{L}_g$ is the one that best fits the columns of A among all lines in $\mathbb{R}^m$, [SE03, Sec.A.7] (see also appendix B.1). And $\mathscr{L}_{\mathrm{A}}$ is the best fitting line among all lines passing through the origin in $\mathbb{R}^m$, see Figure 5.1. And since the misfit of $\mathscr{L}_g$ is smaller than the misfit of $\mathscr{L}_{\mathrm{A}}$, then (5.2.6) holds.

Algorithm 2 constructs an affine approximation of rank at most k. We name Algorithm 2 *ALORA* (short for Affine Low-Rank Approximation). Its computational complexity is $\mathcal{O}(mnk)$ with a constant factor depending on the use of QRCP or subspace iteration in line 3 of the algorithm. The value of k is given as input; however, it can also be found adaptively by imposing a stopping criterium on the algorithm used to compute ξ_{k-1} in line 3. For instance, for the case of QRCP the classical implementation for this aim is provided in [GVL96, Alg. 5.4.1].



(a) Line $\mathscr{L}_A(\tau) = \{\tau u \in \mathbb{R}^3 | \ \tau \in \mathbb{R}\}$ passing through the origin, $u = u_1(A)$.

(b) Line $\mathscr{L}_g(\tau) = \{g + \tau\tilde{u} \in \mathbb{R}^3 | \ \tau \in \mathbb{R}\}$ passing through the gravity center of A, $\tilde{u} = u_1(Y)$.

Figure 5.1: Best fitting lines (represented as arrows) of a matrix $A = [a_1, \cdots, a_n] \in \mathbb{R}^{3\times n}$. The small circles represent the columns a_j's, for $j = 1, \cdots, n$, and their projections over the lines are also showed. The gravity center g and the matrix Y are defined in (5.2.4) and (5.2.5) respectively.

Data: $A \in \mathbb{R}^{m\times n}$ and a fixed rank k
Result: A rank-k approximation of A
1 Define $c = (1, \cdots, 1)^T \in \mathbb{R}^n$;
2 Compute $Y = A\left(I_n - \frac{1}{n}cc^T\right)$;
3 Find ξ_{k-1}: a rank-$(k-1)$ approximation of Y ;
4 Return $\xi_k = gc^T + \xi_{k-1}$

Algorithm 2: Affine low-rank approximation, ALORA

Error analysis

Below we present how to easily derive a bound for the approximation error of an affine approximation. First, consider that

$$\|Y - \xi_{k-1}\|_2 \leq f(k, n)\sigma_k(Y),$$

where $f(k, n)$ is a function depending on the low rank method used, *c.f.* Table 4.1. Then, since $\xi_k = gc^T + \xi_{k-1}$, we obtain

$$\|A - \xi_k\|_2 = \|Y - \xi_{k-1}\|_2 \leq f(k, n)\sigma_k(A), \tag{5.2.9}$$

where we use the fact that $\sigma_k(Y) \leq \sigma_k(A)$ ensured by Corollary 5.2.

Next, note that geometrically the approximation ξ_k can be interpreted as fitting the columns of the matrix into an affine subspace of dimension $k-1$. And since the rank-k truncated SVD can be seen as fitting into a subspace of dimension k, we might also use an affine subspace of dimension k. In terms of matrices, it means that Y is approximated by a rank-k matrix ξ_k, and the affine approximation for A is constructed as

$$A \approx \xi_{k+1} := gc^T + \xi_k \quad \text{with} \quad \|A - \xi_{k+1}\|_2 = \|Y - \xi_k\|_2 \leq f(k, n)\sigma_{k+1}(A), \tag{5.2.10}$$

where it should be noted that the rank of ξ_{k+1} is bounded by $k+1$.

In Section 5.4 we plot the approximation errors when A is approximated by ξ_k and ξ_{k+1}, showing that in many cases they both overcome the QRCP approximation of rank-k.

Finally, note that a derivation of a bound for the approximation error can be obtained for any low-rank approximation method by using the fact that $\|A-\xi_{k+1}\|_2 = \|Y-\xi_k\|_2$. And depending on the method, we can obtain a bound depending on $\sigma_{k+1}(A)$. For instance, for a QR based approximation, simply by replacing $f(k,n)$ by its appropriate value, see Table 4.1. And for subspace iteration, simply by using the fact that $\sigma_{k+1}(Y) \leq \sigma_{k+1}(A)$ when bounding $\|Y-\xi_k\|_2$ using (4.4.3).

5.3 Correlation of Matrices Using their Gravity Center

In the previous section we have shown how to construct an affine low rank approximation for any matrix $A \in \mathbb{R}^{m\times n}$. In this section we explore the structural relation of a matrix and their best fitting lines $\mathscr{L}_A$ and $\mathscr{L}_g$ studied in the previous section. This allows us to understand for which kind of matrices an affine low rank approximation could be better than the non-affine one. We define a correlation coefficient that helps to understand the matrix structure seeing its columns as spatial points in $\mathbb{R}^m$. We start by analyzing the particular case of matrices with exponentially decreasing singular values, which have high correlation coefficient according to our definitions as we will show later on, and this helps us to provide simple formulas to obtain estimates of the matrix norm and the first left and right singular vectors.

5.3.1 Matrices with Exponentially Decreasing Singular Values

In many important problems of linear algebra oriented to mathematical modeling, matrix compression and related subjects, we handle a matrix $A \in \mathbb{R}^{m\times n}$ with singular values that decrease exponentially. This means that if A has singular triplets (u_j, v_j, σ_j) for $j = 1, \cdots, r = \operatorname{rank}(A)$, then

$$\sigma_j \leq \rho^j \sigma_1, \tag{5.3.1}$$

where $0 < \rho < 1$. Such matrix arises, for example, as an "admissible" block in the context of discretization of boundary integral operators [Beb08]. They are also interesting in merely theoretical and testing problems such as the Kahan matrix [DB08]. In order to see if an affine low-rank approximation would be useful for these kind of matrices, let us write the gravity center of A using its singular triplets,

$$g = \frac{1}{n}\sum_{j=1}^{n} a_j = \frac{1}{n}\sum_{l=1}^{r} u_l\sigma_l\left(\sum_{j=1}^{n} v_l(j)\right),$$

then,

$$g = \sigma_1\left(\sum_{l=1}^{r} \tilde{v}_l u_l\right), \quad \text{with} \quad \tilde{v}_l = \frac{\sigma_l}{n\sigma_1}\left(\sum_{j=1}^{n} v_l(j)\right),$$

and since $\sum_{j=1}^{n} |v_l(j)| \leq \sqrt{n}$, then $|\tilde{v}_l| \leq \frac{\rho^l}{\sqrt{n}}$. Hence, if the singular values of A decrease as in equation (5.3.1), then the unitary vector in the direction of g would be a good approximation u_1, and this approximation gets better when ρ gets smaller. In other words, the matrix A is such that its best fitting

lines, $\mathscr{L}_A$ and $\mathscr{L}_g$, almost overlap.
Note that applying Algorithm 2 to a matrix with rapidly singular values can produce an increase on the precision as in the case of Figure 5.2, and for some cases as in Figures 5.5 and 5.6 it may not produce good results. However, in all the cases of matrices with exponentially decreasing singular values, we will get interesting characterizations of their singular triplets, as it is shown in the next subsection.

Finally, a useful observation, to which we will refer later, is that if A has exponentially decreasing singular values, then the cosine of the angle made by the gravity center and its j-th column is closer to 1 when ρ gets small, this is true since

$$\frac{g^{\mathrm{T}}a}{\|g\|_2\|a_j\|_2} = \frac{\tilde{v}_1 v_1(j) + \sum_{l=2}^{r}\left(\frac{\sigma_l}{\sigma_1}\right)^2 \tilde{v}_l v_l(j)}{\sqrt{\tilde{v}_1^2 + \sum_{l=2}^{r}\left(\frac{\sigma_l}{\sigma_1}\right)^2 \tilde{v}_l^2}\sqrt{v_1(j)^2 + \sum_{l=2}^{r}\left(\frac{\sigma_l}{\sigma_1}\right)^2 v_l(j)^2}}. \tag{5.3.2}$$

5.3.2 Characterization of Matrices using their Gravity Center

Consider $A \in \mathbb{R}^{m\times n}$, from the previous best fitting line analysis, it is clear that a sufficient condition for the lines $\mathscr{L}_A$ and $\mathscr{L}_g$ to coincide, is that $g = 0$. Let us consider the reverse case, i.e. if $\mathscr{L}_A$ and $\mathscr{L}_g$ are identical, then what can we say about the matrix A? The following theorem provides the answer.

Theorem 5.1. *Consider $A \in \mathbb{R}^{m\times n}$, with $r = \operatorname{rank}(A)$ and singular triplets (u_j, v_j, σ_j), for $j = 1, \cdots, r$. Let its best fitting lines be $\mathscr{L}_A$ and $\mathscr{L}_g$. Consider the vector of ones $c = (1, \cdots, 1)^{\mathrm{T}} \in \mathbb{R}^n$. Then, both lines are identical if and only if*

$$A = B + \|g\|_2 u_1 c^{\mathrm{T}},$$

where $B \in \mathbb{R}^{m\times n}$ is a matrix for which the gravity center of its columns is zero. Furthermore, if $\mathscr{L}_A$ and $\mathscr{L}_g$ are identical, then the norm of A is bounded as

$$\|A\|_2 \geq \sqrt{n}\|g\|_2, \tag{5.3.3}$$

and if $\|g\|_2 \neq 0$, we get

$$u_1 = \frac{g}{\|g\|_2},$$

and the right singular vectors hold

$$v_1^{\mathrm{T}} c = \sum_{i=1}^{n} v_1(i) = \frac{n\|g\|_2}{\sigma_1}, \tag{5.3.4}$$

$$v_j^{\mathrm{T}} c = \sum_{i=1}^{n} v_j(i) = 0, \quad \text{for } j = 2, \cdots, r. \tag{5.3.5}$$

Proof. If $g = 0$, the first statement follows straightforwardly. Hence, let us consider the non-trivial case when $g \neq 0$. If $A = B + \|g\|_2 u_1 c^{\mathrm{T}}$, then clearly both lines coincide, since for this case when computing g we obtain

$$u_1 = \frac{g}{\|g\|_2}, \tag{5.3.6}$$

where we use the fact that the gravity center of B is zero. To prove the reverse statement, assume both lines are identical, i.e. assume that (5.3.6) holds. Then, define $\mathrm{B} := \mathrm{A} - gc^{\mathrm{T}}$, where clearly the gravity center of B is zero. And using (5.3.6) we can write

$$\mathrm{A} = \mathrm{B} + \|g\|_2 u_1 c^{\mathrm{T}},$$

which proves the first statement of the theorem.

Next, to prove the second statement of the theorem, write the j-th column of A using its singular triplets, this is

$$a_j = \sum_{l=1}^{r} u_l \sigma_l v_l(j), \quad \text{for } j = \{1, \cdots, n\}. \tag{5.3.7}$$

By definition of the gravity center and (5.3.6), we get

$$g = \frac{1}{n} \sum_{j=1}^{n} a_j = \|g\|_2 u_1, \tag{5.3.8}$$

and combining (5.3.7) and (5.3.8), we get

$$\sum_{l=1}^{r} u_l \sigma_l \left(\sum_{j=1}^{n} v_l(j) \right) = n\|g\|_2 u_1,$$

$$\underbrace{\left(\sigma_1 \left(\sum_{j=1}^{n} v_1(j) \right) - n\|g\|_2 \right)}_{\beta_1} u_1 + \sum_{l=2}^{r} \underbrace{\left(\sigma_l \sum_{j=1}^{n} v_l(j) \right)}_{\beta_l} u_l = 0, \tag{5.3.9}$$

and since (5.3.9) is a linear combination of linearly independent vectors, then $\beta_1 = \beta_2 = \cdots = \beta_r = 0$, which proves (5.3.4) and (5.3.5). Finally, by the Cauchy-Schwartz inequality, we have that

$$|v_1^{\mathrm{T}} c| \leq \|c\|_2 = \sqrt{n}, \tag{5.3.10}$$

and (5.3.3) follows by replacing (5.3.4) in (5.3.10). □

Next, let us explore a direct consequence of the previous theorem. First, note that

$$\frac{\sqrt{n}}{\sqrt{r}} \|a_{\triangle}\|_2 \leq \|\mathrm{A}\|_2 \leq \sqrt{n} \|a_{\star}\|_2,$$

where $a_{\triangle}$ and $a_{\star}$ are, respectively, the columns of A with smallest and largest norm respectively. These inequalities follow from the fact that $\frac{1}{\sqrt{r}} \|\mathrm{A}\|_{\mathrm{F}} \leq \|\mathrm{A}\|_2 \leq \|\mathrm{A}\|_{\mathrm{F}}$.

Hence, when A is such that its best fitting lines, $\mathscr{L}_{\mathrm{A}}$ and $\mathscr{L}_g$, are identical, then we can obtain a narrow bound for the matrix norm, given as

$$\sqrt{n}\|g\|_2 \leq \|\mathrm{A}\|_2 \leq \sqrt{n}\|a_{\star}\|_2,$$

and we can obtain an estimate of the norm that becomes more precise when the columns of the matrix have similar norm. However, it is not evident when A is such that $\mathscr{L}_{\mathrm{A}}$ and $\mathscr{L}_g$ are identical, we explore this in the next subsection.

Finally, gathering the results from this and the previous subsection, we get that an affine approximation should not be used when the gravity center of the columns of the matrix is very small, since for this case both best fitting lines coincide, e.g. the matrix A = `randn`(n) constructed with MATLAB, has as entries normally distributed random numbers having mean zero, so an affine approximation would not make sense. For all other cases, an affine approximation might increase the precision as it is shown in Section 5.4.

5.3.3 Measuring the Correlation of Matrices

We can obtain insights about the geometrical distribution of the columns of a matrix by using formal concepts from statistics, as the correlation of a matrix.

The correlation of a matrix $\mathrm{A} \in \mathbb{R}^{m\times n}$ is typically expressed using the pairwise correlation of its columns, this is, consider the columns a_j and a_l with means $\bar{g}_j := \frac{1}{m}\sum_{i=1}^{m} a_j(i)$ and $\bar{g}_l := \frac{1}{m}\sum_{i=1}^{m} a_l(i)$ respectively. Then, we can obtain the *Pearson correlation* coefficient defined as

$$\rho_{jl} = \frac{\sum_{i=1}^{n}(a_j(i)-\bar{g}_j)(a_l(i)-\bar{g}_l)}{\sqrt{\sum_{i=1}^{n}(a_j(i)-\bar{g}_j)^2}\sqrt{\sum_{i=1}^{n}(a_l(i)-\bar{g}_l)^2}} = \frac{\bar{a}_j^{\mathrm{T}}}{\|\bar{a}_j\|_2}\frac{\bar{a}_l}{\|\bar{a}_l\|_2} = \cos(\angle(\bar{a}_j,\bar{a}_l)),$$

where $\bar{a}_j := a_j - \bar{g}_j\bar{c}$ and $\bar{a}_l = a_l - \bar{g}_l\bar{c}$ are obtained by centering a_j and a_l with respect to their mean, with $\bar{c} = (1,\cdots,1)^{\mathrm{T}} \in \mathbb{R}^m$. This provides a symmetric $(m \times n)$ matrix of coefficients ρ_{jl} having ones on the diagonal. For example, the function `corr` from MATLAB gives exactly this matrix. And since the pairwise interaction of distinct columns can provide at most $n(n-1)/2$ different values, then we can define the correlation of a matrix as a real number, given as

$$\mathscr{C}(\mathrm{A}) := 2\frac{\sum_{i<j}|\rho_{ij}|}{n(n-1)}.$$

Note that $0 \le \mathscr{C}(\mathrm{A}) \le 1$. It is clear that at a given stage of the approximation, computing $\mathscr{C}(\mathrm{A})$ for all the columns would provide an accurate stopping criterium. For instance, at the step $k-1$ of one approximation algorithm, consider

$$\mathrm{F} = \mathrm{A} - \xi_{k-1},$$

then, theoretically if $\mathscr{C}(\mathrm{F}) = 1$, then F is a rank-one matrix and the algorithm should stop at the step k. This could be replaced by the weaker condition $1-\delta < \mathscr{C}(\mathrm{F})$. This technique could be used as a stopping criterium, however costly, indeed it would cost $\mathcal{O}(mn^2k)$.

Next, we propose a cheaper way to measure the correlation of A by defining the correlation vector $\tilde{\rho}_{\mathrm{A}} \in \mathbb{R}^n$ as

$$\tilde{\rho}_{\mathrm{A}}(j) := \frac{g^{\mathrm{T}}a_j}{\|g\|_2\|a_j\|_2}, \tag{5.3.11}$$

which costs $\mathcal{O}(mn)$ to compute, and the correlation coefficient

$$\mathcal{G}(\mathrm{A}) := \frac{\max(\tilde{\rho}_{\mathrm{A}}) - \min(\tilde{\rho}_{\mathrm{A}})}{2}. \tag{5.3.12}$$

Note that $0 \leq \mathcal{G}(\mathrm{A}) \leq 1$ and that $\mathcal{G}(\mathrm{A})$ is a good indicator of the spacial distribution of the columns of the matrix with respect to its gravity center, and in section 5.3.4 we show a very good rank-one approximation for matrices having high correlation coefficient. A way to reduce the computation cost of obtaining $\mathcal{G}(\mathrm{A})$ is to approximate it using only $\mathcal{O}(ml)$ arithmetic cost, where $l < n$. For this we need to use a randomized approach. For instance, the randomized version of QRCP [DG17] obtains its permutation by applying the classic QRCP on the smaller matrix $\Omega_r \mathrm{A} \in \mathbb{R}^{l\times n}$, where $\Omega_r \in \mathbb{R}^{l\times m}$ is a random compression matrix. Hence, we can use (or reuse) a random compression matrix to approximate the gravity center of A by the gravity center of $\mathrm{A}\Omega$, where $\Omega \in \mathbb{R}^{n\times l}$, and make the approximation $\mathcal{G}(\mathrm{A}) \approx \tilde{\rho}(\mathrm{A}\Omega)$. Moreover, note that the approximation of g holds

$$\|g - \tilde{g}\|_2 = \|\frac{1}{n}\mathrm{A}c - \frac{1}{n}\mathrm{A}\Omega c\|_2 \leq \frac{\|\mathrm{A}\|_2(1 + \|\Omega\|_2)}{\sqrt{n}},$$

where $\tilde{g}$ is the gravity center of $\mathrm{A}\Omega$, and $c = (1, \cdots, 1)^{\mathrm{T}} \in \mathbb{R}^n$, and this approximation is justified when the norms of A and Ω are small.

5.3.4 Matrices with High Correlation

We consider that a matrix A has high correlation if the mean of the correlation vector $\tilde{\rho}_{\mathrm{A}}$ defined in (5.3.11) is close to 1, or if the correlation coefficient $\mathcal{G}(\mathrm{A})$, defined in (5.3.12), is close to 0. In order to find a representation of matrices with high correlation, let us consider a rank-one matrix A. From the linear dependency of its columns, it is clear that its correlation coefficients, $\mathcal{C}(\mathrm{A})$ and $\mathcal{G}(\mathrm{A})$, are equal to 1. Furthermore it is clear that A can be written as $\mathrm{A} = [\beta_1 \texttt{ones}(m, n_1), \cdots, \beta_k \texttt{ones}(m, n_k)]$, with appropriate coefficients β_j and $n_1 + \cdots + n_k = n$. Next lemma gives us an useful representation of A.

Lemma 5.3. *Consider* $\mathrm{A} = [\beta_1 \texttt{ones}(m, n_1), \cdots, \beta_k \texttt{ones}(m, n_k)] \in \mathbb{R}^{m\times n}$*, where* $\beta_j \in \mathbb{R}$ *for* $j = 1, \cdots, k$*, and* $n_1 + \cdots + n_k = n$*. Then,*

$$\texttt{abs}(u_1(\mathrm{A})) = \texttt{abs}\left(\frac{g}{\|g\|_2}\right), \quad \texttt{abs}(v_1(\mathrm{A})) = \texttt{abs}\left(\frac{g_t^{\mathrm{T}}}{\|g_t\|_2}\right), \tag{5.3.13}$$

and,

$$\sigma_1(\mathrm{A}) = \sqrt{m(\sum_{j=1}^{k} n_j \beta_j^2)}, \tag{5.3.14}$$

where g *and* g_t *are the gravity centers of the columns of* A *and* A^{T} *respectively.*

Proof. First, note that the line passing through the origin of $\mathbb{R}^m$ in the direction of $c_1 = \texttt{ones}(m, 1) \in \mathbb{R}^m$ is the best fitting line of the columns of A. Since clearly g also belongs to this line, it means that both best fitting lines of A, i.e. $\mathscr{L}_{\mathrm{A}}$ and $\mathscr{L}_g$, coincide, and using Theorem 5.1 we get the left equality of (5.3.13). Analogously, to obtain the right equality of (5.3.13), observe that the line passing through the origin of $\mathbb{R}^n$ in the direction of the vector $c_1 = (\beta_1 \texttt{ones}(1, n_1), \cdots, \beta_k \texttt{ones}(1, n_k))^{\mathrm{T}} \in \mathbb{R}^n$ is the best fitting line of the columns of A^{T} and it contains g_t, then apply Theorem 5.1 on A^{T}.

Next, note that A has rank-one, this is $\sigma_j(A) = 0$ for $j \geq 2$, hence both spectral and Frobenius norm coincide and a simple calculus shows that $\sigma_1(A) = \|A\|_F = \sqrt{m(\sum_{j=1}^{k} n_j \beta_j^2)}$. □

Next, let us propose a rank-one approximation for matrices having high correlation. Note that in particular, from equation (5.3.2) we know that a matrix A with exponentially decreasing singular values tends to have high correlation (just by using the definition of the correlation vector in (5.3.11)). We define the rank-one approximation of these kind of matrices as

$$A \approx \xi_1 := \frac{g}{\|g\|_2} \tilde{\sigma}_1 \frac{g_t^T}{\|g_t\|_2}, \tag{5.3.15}$$

where $\tilde{\sigma}_1$ approximates $\sigma_1(A)$, and according to Lemma 5.3 it can be taken as in (5.3.14). However, our experiments show that a better approximation of the first singular value is

$$\sigma_1 \approx \tilde{\sigma}_1 := \|g\|_2 \sqrt{n}, \tag{5.3.16}$$

which is justified by the analysis made in section 5.3.2.

In Section 5.4.2, we show that $\tilde{\sigma}_1$ approximates very well $\sigma_1 = \|A\|_2$ for most of the test matrices, even though most of them do not have singular values decreasing exponentially.

Next, according to Lemma 5.3 we can approximate the directions of the first left and right singular vectors as

$$u_1(A) = +\frac{g}{\|g\|_2} \quad \text{or} \quad u_1(A) = -\frac{g}{\|g\|_2}, \quad \text{and} \tag{5.3.17}$$

$$v_1(A) = +\frac{g_t}{\|g_t\|_2} \quad \text{or} \quad v_1(A) = -\frac{g_t}{\|g_t\|_2}, \tag{5.3.18}$$

where g and g_t are the gravity centers of A and A^T respectively.

Finally, let us introduce the following definition that will allow to measure the error of approximating $u_1(A)$ and $v_1(A)$ as in (5.3.17) and (5.3.18) respectively, *c.f.* Figure 5.10.

Definition 5.1. For a vector $w \in \mathbb{R}^m$ we define its sign as

$$S(v) := \texttt{sign}\left(\sum_{i=1}^{m} \texttt{sign}\big(w(i)\big)\right),$$

where `sign` is the standard function for real numbers.

5.4 Numerical Experiments

5.4.1 Low-rank Approximation of Challenging Matrices

In this section we numerically show the benefits of Algorithm 2 on a set of challenging matrices with $m = n = 256$, given in Table 5.1. Most of the matrices from Table 5.1 have been previously used in experiments with QR factorizations [DGGX15, GCD18]. These matrices have been constructed using MATLAB and they are easy to replicate for testing and verification. Some of the test matrices, have

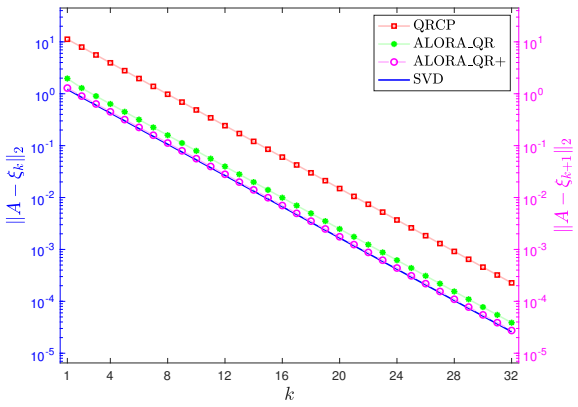
the form $A = U\Sigma V^T$ where, when it is not specified, U and V are random orthogonal matrices and $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_n)$ is a diagonal matrix containing prescribed singular values, the machine epsilon is given as $\epsilon = 2.22E-16$.

Table 5.1: Test matrices

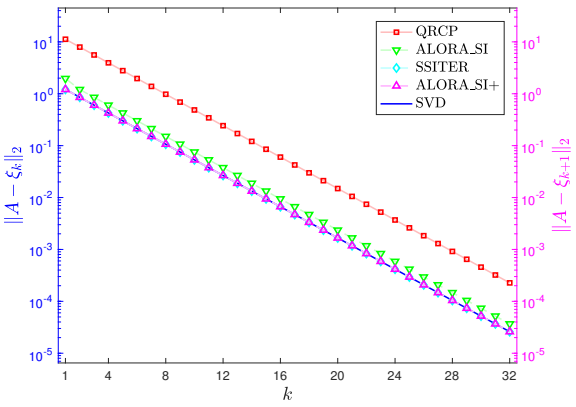
No.	Matrix	Description
1	BAART	Coming from the discretization of the first kind Fredholm integral equation [Han].
2	BREAK-1	$A = U\Sigma V^T$, where $\sigma_1 = \cdots = \sigma_{n-1} = 1$, and $\sigma_n = 10^{-9}$ [Bis91].
3	BREAK-9	$A = U\Sigma V^T$, where $\sigma_1 = \cdots = \sigma_{n-9} = 1$, and $\sigma_{n-8} = \cdots = \sigma_n = 10^{-9}$ [Bis91].
4	DERIV2	Coming from the computation of the second derivative [Han].
5	EXPON	$A = U\Sigma V^T$, where $\sigma_1 = 1$, and $\sigma_i = \alpha^{i-1}$ for $i = 2, \cdots, n$ [Bis91].
6	FOXGOOD	Coming from the discretization of the first kind Fredholm integral equation of a severely ill-posed problem, first used by Fox and Goodwin [Han].
7	GKS	Upper-triangular matrix whose j-th diagonal element is $1/\sqrt{j}$ and whose (i, j) element is $-1/\sqrt{j}$ for $j > i$ [GE96, GKS76].
8	GRAVITY	Coming from the discretization of a one-dimensional model problem in gravity surveying [Han].
9	HC	$A = U\Sigma V^T$, where Σ has diagonal entries 100, 10, and the following $n-2$ are evenly spaced between 10^{-2} and 10^{-8} [HT05].
10	HEAT	Inverse heat equation [Han].
11	PHILLIPS	Phillips test problem [Han].
12	RANDOM	Random matrix $A = 2 * \texttt{rand}(n) - 1$ [GE96].
13	SCALE	Random matrix whose i-th row is scaled by the factor $\eta^{i/n}$, with $\eta = 10\epsilon$ [GE96].
14	SHAW	1D image restoration model [Han].
15	SPIKES	Test problem with a "spiky" solution [Han].
16	STEWART	Matrix $A = U\Sigma V^T + 0.1\sigma_n * \texttt{rand}(n)$, where Σ has first half of the diagonals decreasing geometrically from 1 to $\sigma_n = 10^{-3}$, and the last half of the diagonals being set to zero [Ste99].
17	URSELL	Coming from the discretization of an integral equation with no square integrable solution [Han].
18	WING	Coming from a test problem with a discontinuous solution [Han].
19	KAHAN	The Kahan matrix [Kah66].
20	DEVIL	Devil stairs matrix, a matrix with gaps in its singular values [Ste99].
21	RAND-UNIF	Random matrix with uniformly distributed entries, $A = \texttt{rand}(n)$.
22	3D-ADM	An admissible submatrix of a hierarchical matrix, see description in the text.
23	3D-LAP-NADM	A non-admissible submatrix of a hierarchical matrix, see description in the text.

Matrices 22 and 23 are submatrices of a global hierarchical matrix coming from the discretization of an integral operator on the domain defined in section 6.4.1, matrix 22 correspond to an admissible block created with the admissibility condition reported in Figure 6.4, while matrix 23 corresponds to a non-admissible block obtained from the interaction of adjacent subdomains. Matrix 22 is ensure to have singular values decreasing exponentially, while it is not the case for matrix 23.

Next, we present the error for a rank-k approximation of different test matrices, for $k = 1, \cdots, 32$. We compare ALORA with QRCP and subspace iteration. Figures 5.2 to 5.6 show the approximation errors for some of the test matrices, in order to appreciate the cases where an affine low rank approximation is advantageous or disadvantageous. The labels ALORA_QR and ALORA_SI refer to ALORA using QRCP and subspace iteration (using just small parameters $q = 1$ and $l = k + 3$) to produce the rank $k-1$ approximation needed in line 3 of Algorithm 2. All figures include a right Y-axis where the values ALORA_QR+ and ALORA_SI+ are plotted, they are obtained by plotting for a given k, the error made by approximating A by the matrix ξ_{k+1} defined in (5.2.10). Note that the curves of the SVD, SSITER and ALORA_SI+ almost overlap each other.

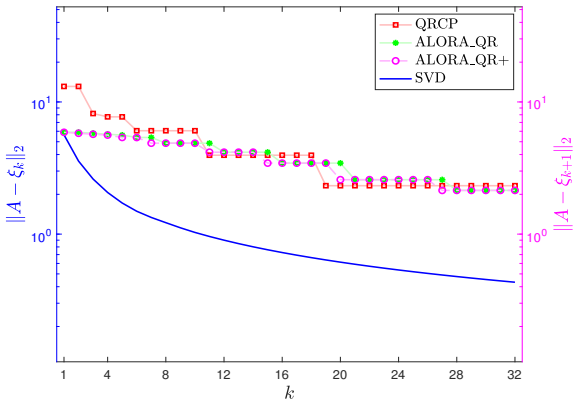


(a) Approximation error of ALORA created with QRCP.

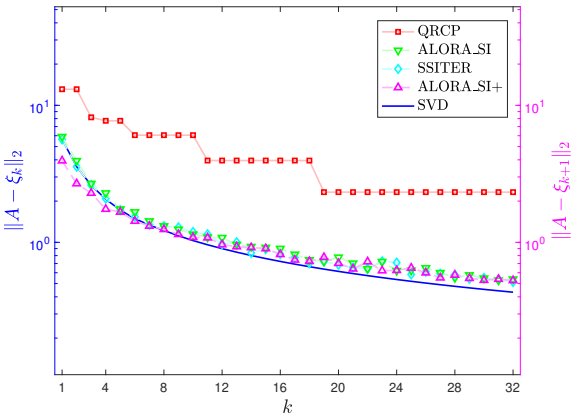


(b) Approximation error of ALORA created with subspace iteration.

Figure 5.2: Convergence curves of the approximation error for the KAHAN matrix.

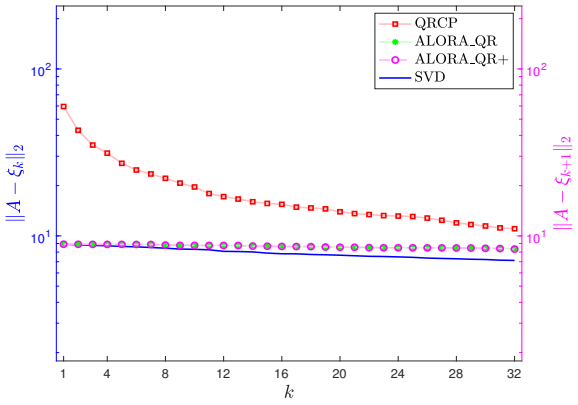


(a) Approximation error of ALORA created with QRCP.

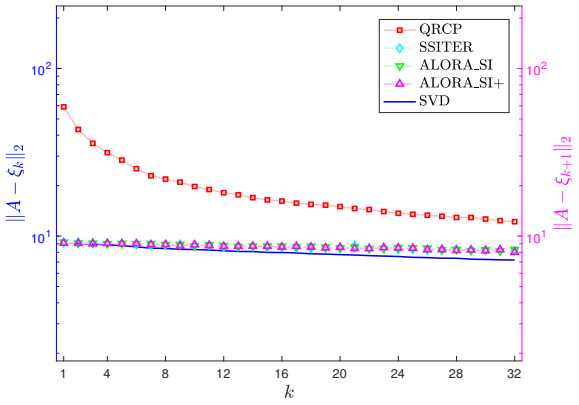


(b) Approximation error of ALORA created with subspace iteration.

Figure 5.3: Convergence curves of the approximation error for the GKS matrix.

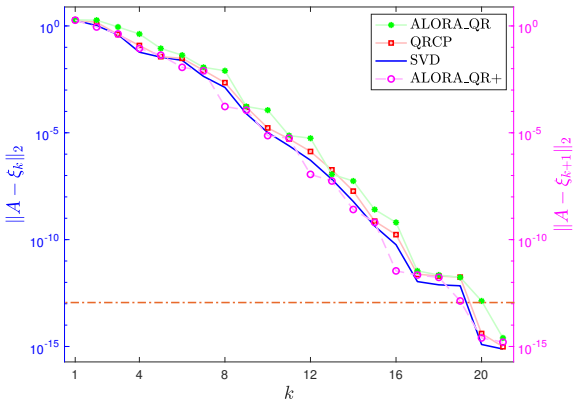


(a) Approximation error of ALORA created with QRCP.

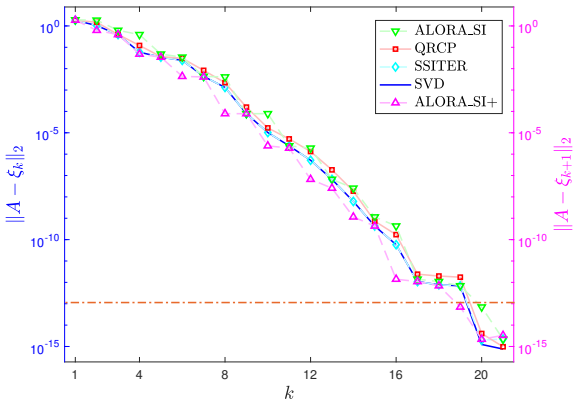


(b) Approximation error of ALORA created with subspace iteration.

Figure 5.4: Convergence curves of the approximation error for the RAND-UNIF matrix.

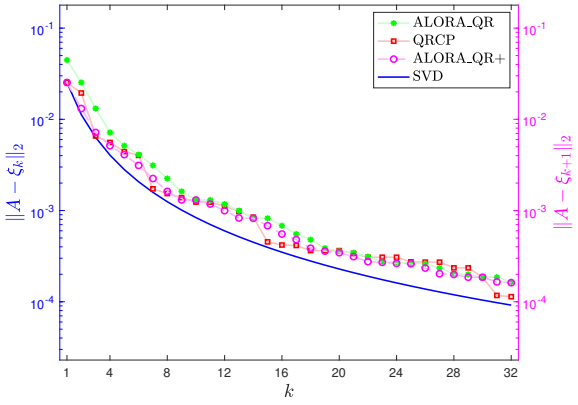


(a) Approximation error of ALORA created with QRCP.

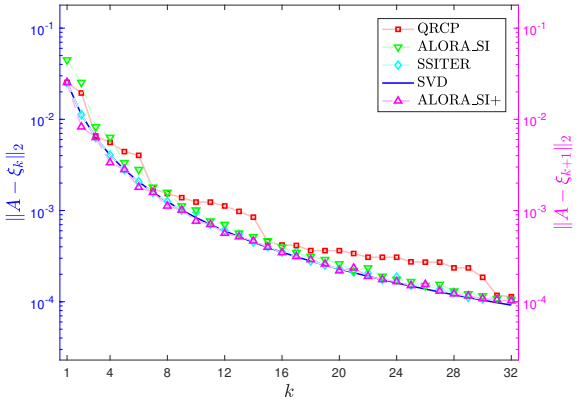


(b) Approximation error of ALORA created with subspace iteration.

Figure 5.5: Convergence curves of the approximation error for the SHAW matrix. The horizontal line is the threshold value, $\epsilon \max(m, n)\|A\|_2$, beyond which the singular values are considered as zero.



(a) Approximation error of ALORA created with QRCP.



(b) Approximation error of ALORA created with subspace iteration.

Figure 5.6: Convergence curves of the approximation error for the DERIV2 matrix.

Note that for the matrices with slowly decreasing singular values, GKS and RAND-UNIF, we have that ALORA improves the approximation for k small. While for the other cases, when the matrices have rapidly decreasing singular values, as studied in Section 5.3.1, their best fitting lines tend to overlap each other and hence an affine approximation may increase considerably the precision as in the case of Figure 5.2, and for some cases as in Figures 5.5 and 5.6 it may not produce good results since the rank-one approximation gc^{T}, used by the ALORA algorithm, might be far from the optimal. For this case it would be more suitable to use the rank-one approximation from (5.3.15) to start the approximation, see Figure 5.9.

Next, we compute the approximation errors for all the matrices described in Table 5.1. Considering an approximation of rank $k = 1, \cdots, \min(\mathrm{rank}(A), 16)$, we compute the errors

$$\mathrm{E}_{\mathrm{QRCP}}(k) = \frac{\|A - \xi_k\|_2}{\sigma_{k+1}(A)}, \tag{5.4.1}$$

$$\mathrm{E}_{\mathrm{ALORA\_QR+}}(k) = \frac{\|A - gc^{\mathrm{T}} - \bar{\xi}_k\|_2}{\sigma_{k+1}(A)}, \tag{5.4.2}$$

$$\mathrm{E}_{\mathrm{ALORA\_SI+}}(k) = \frac{\|A - gc^{\mathrm{T}} - \tilde{\xi}_k\|_2}{\sigma_{k+1}(A)}, \tag{5.4.3}$$

where ξ_k and $\bar{\xi}_k$ are rank-k approximations of A and Y respectively constructed using QRCP, and $\tilde{\xi}_k$ is a rank-k approximation of Y constructed using subspace iteration (Algorithm 6). Figure 5.7 plots the average and variances of these values for all the matrices from Table 5.1.

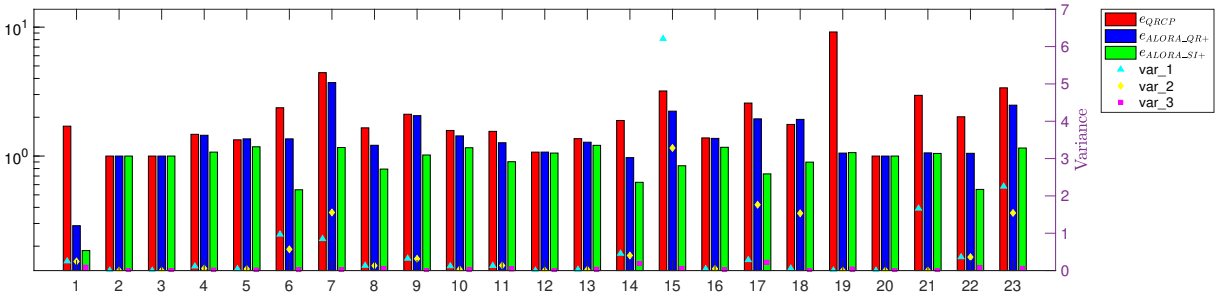


Figure 5.7: Mean of the ratios of the errors of rank-k approximations created by ALORA_QR+ and ALORA_SI+ to the optimal error. For each matrix, e_{QRCP}, $e_{\mathrm{ALORA\_QR+}}$ and $e_{\mathrm{ALORA\_SI+}}$ are, respectively, the mean of the vectors $\mathrm{E}_{\mathrm{QRCP}}$, $\mathrm{E}_{\mathrm{ALORA\_QR+}}$ and $\mathrm{E}_{\mathrm{ALORA\_SI+}}$ defined in (5.4.1), (5.4.2) and (5.4.3); and var_1, var_2 and var_3 are their variances.

We can see that in many cases ALORA, in its both versions ALORA_QR+ and ALORA_SI+, improves the accuracy of QRCP (up to 10 times) . Note that ALORA_QR+ performs, in average, better than QRCP, while ALORA_SI+ overpasses the accuracy of the other methods. Hence, constructing the rank-k approximation of a matrix as fitting its columns into a k-dimensional affine subspace can improve the accuracy of the approximation.

5.4.2 Approximation of the Matrix Norm

Using the analysis done in §5.3, we show that our estimate $\tilde{\sigma}_1 = \|g\|_2\sqrt{n}$, given in (5.3.16), for the norm of a given matrix A, works quite good for most of the test matrices from Table 5.1. We compare this estimate with the one obtained performing a truncated QRCP Factorization of A, i.e. $A = QRP^{\mathrm{T}}$, where

generally authors approximates the i-th singular value as $|R(i,i)|$, see e.g. [DGGX15, Sec. 4], [GCD18, Ste99]. However, this estimate is rough and more precisely viewing QRCP as the decomposition of type (5.2.1), an estimate of σ_i can also be taken as $\|R(i,:)\|_2$. Note that there are more precise ways to approximate the norm using a QR based method, for example we can use the L-values (or the more efficient algorithms) proposed by Stewart [Ste99, Sec. 6]. In Figure 5.8 we plot the ratios of the values $|R(i,i)|$, $\|R(i,:)\|_2$ and $\tilde{\sigma}_1$ to the exact norm.

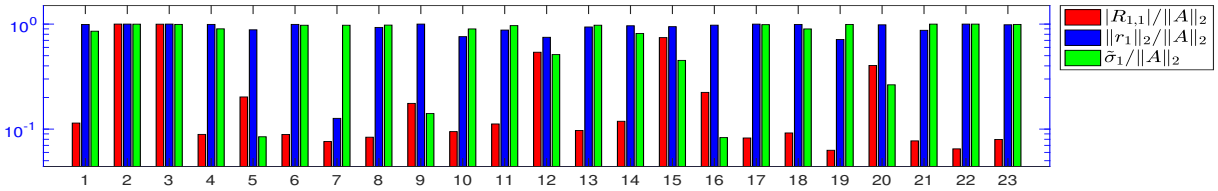


Figure 5.8: Ratios of the approximated matrix norm, we compare $|R(1,1)|$, $\|R(1,:)\|_2$ and $\tilde{\sigma}_1$ to $\|A\|_2$.

In Figure 5.9 we show that the rank-one approximation ξ_1 from (5.3.15) provides a very good rank-one approximation, we consider the following two approximations

$$A \approx q_1 r_1^T, \qquad \text{and} \qquad A \approx \xi_1 = \frac{g}{\|g\|_2}\tilde{\sigma}_1\frac{g_t^T}{\|g_t\|_2}, \quad \tilde{\sigma}_1 = \|g\|_2\sqrt{n},$$

where the first is the classical QRCP rank-one approximation.

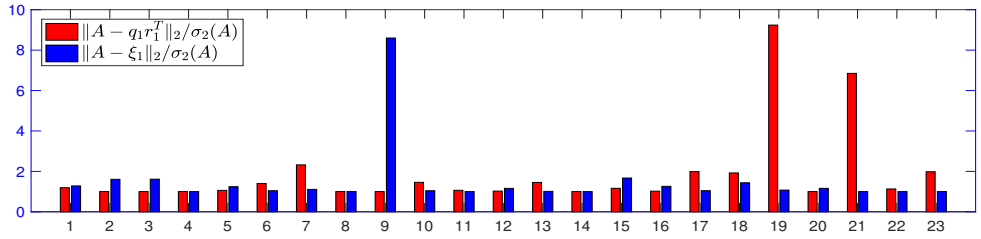
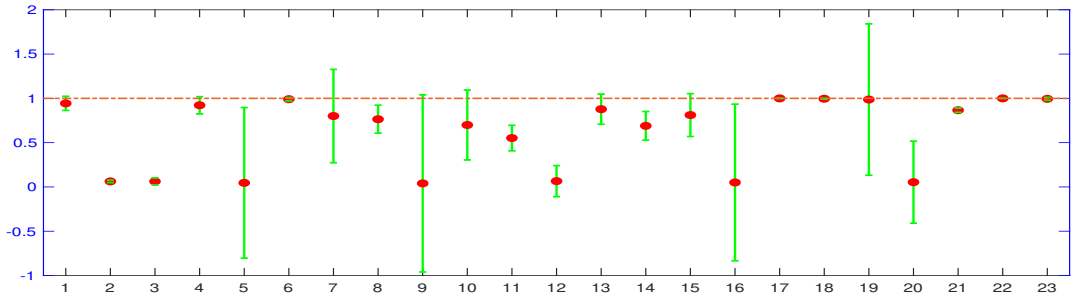


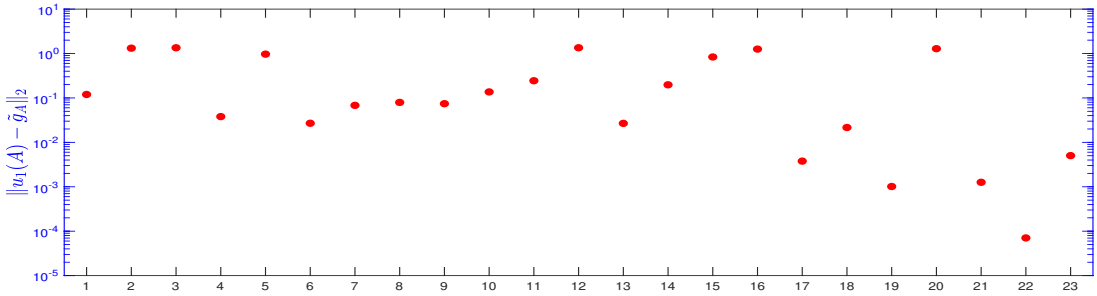
Figure 5.9: Ratios of the error of rank-one approximation obtained by QRCP and ξ_1 from (5.3.15) to the optimal error.

5.4.3 Analyzing the Correlation Coefficient

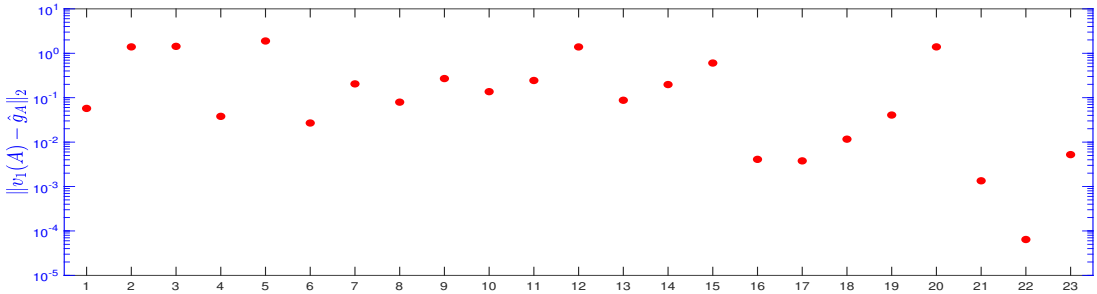
In Figure 5.10 we numerically study the correlation of a matrix by using the vector $\tilde{\rho}(A)$, defined in (5.3.11), and the correlation coefficient $\mathcal{G}(A)$, defined in (5.3.12), as indicators of when the best fitting lines of A tend to overlap each other, and hence provide an easy way to approximate $u_1(A)$ and $v_1(A)$ according to Theorem 5.1. The matrix A stands for one of the 23 matrices from Table 5.1. We present three subfigures aligned in such a way that we can see that for matrices with high correlation we can approximate the first left and right singular vectors by using information of the spatial distribution of the columns and rows of A, more precisely, the gravity centers of its columns and rows.



(a) Letting A be one of the 23 matrices from Table 5.1, the circle corresponds to the mean of the correlation vector $\tilde{\rho}_A$ defined in (5.3.11), and the length of the bar is equal to the correlation coefficient $\mathscr{G}(A)$ defined in (5.3.12).



(b) Error to approximate the first left singular vector $u_1(A)$ by $\tilde{g} = \tilde{s}\frac{g}{\|g\|_2}$, where g is the gravity center of the columns of A and $\tilde{s} = S(g)S(u_1(A))$, *c.f.* (5.3.17).



(c) Error to approximate the first right singular vector $v_1(A)$ by $\hat{g} = \hat{s}\frac{g_t}{\|g_t\|_2}$, where g_t is the gravity center the columns of A^T and $\hat{s} = S(g_t)S(v_1(A))$, *c.f.* (5.3.18).

Figure 5.10: Correlation vector and coefficient for the 23 matrices from Table 5.1.

Note that, as expected, for the matrices with singular values decreasing at exponential rate, we have that the mean of the correlation vector $\tilde{\rho}(A)$ is close to 1, while the coefficient $\mathscr{G}(A)$ is close to 0, and their singular vectors $u_1(A)$ and $v_1(A)$ can be safely approximated by the unit vectors in the directions of the gravity centers of the columns of A and A^T respectively, up to a corresponding sign. Moreover, this kind of approximation also works relatively well for some matrices with slowly decreasing singular values, such as matrices 7 and 21 from Table 5.1.

5.5 Conclusions of the chapter

We have presented the concept of affine low-rank approximation for rectangular matrices, which can be interpreted geometrically as fitting the columns of the matrix into an affine subspace. We have showed how to construct an affine approximation by means of orthogonal projections and propose an algorithm named ALORA that can be adapted to any low-rank approximation algorithm. We have derived a bound for the approximation error and analyzed the cases where this approach might be ad-

vantageous by means of a correlation coefficient that we define in order to understand the geometrical structure of a matrix by seeing its columns as points of a high-dimensional space. By looking for matrices with high correlation, in the sense of our definitions, we encountered the case of matrices with exponentially decreasing singular values for which we have proposed simple formulas to obtain good approximations of their norm and first singular vectors, and hence a good rank-one approximation.

We have constructed affine low-rank approximations using ALORA with the classical QRCP and subspace iteration algorithms. The numerical experiments performed on a set of challenging matrices, showed that an affine low-rank approach can increase, in many cases, the accuracy of QRCP and subspace iteration.

CHAPTER 6

Liner-time CUR approximations for BEM matrices

6.1 Preliminaries

In this chapter, we are interested in accelerating the matrix-vector products for matrices arising from the discretization of boundary integral operators, as the ones from Chapter 2. Such matrices are usually referred to as BEM matrices and arise from diverse real problems such as wave propagation, geophysics, scattering in quantum mechanics, among other applications that can be found *e.g.* in [Lon77, Ste08, Waz11].

A BEM matrix has entries of type $\mathcal{G}(x_i, y_j)$, where $\mathcal{G} : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{C}$, is a kernel integral operator and $\mathrm{X} := [x_1, \cdots, x_m]$ and $\mathrm{Y} := [y_1, \cdots, y_n]$ are interaction domains known as source and target domains respectively. For the scope of this work we consider $d = 3$ by default, however the theory straightforwardly holds for higher dimensions. The classical approach to accelerate the matrix-vector products for BEM matrices, is to separate the kernel evaluation into far-field (tailored to low-rank approximation) and near-field (direct evaluation). One of the most prominent methods to approximate the far field interactions is the Fast Multipole Method FMM [GR87, Kou95]; however, it has important drawbacks such as the kernel-dependency, high cost for problems with multiple right-hand sides and its difficult implementation. Remedies to these drawbacks have been and are currently being developed, such as Kernel independent FMM methods [MR07, FD09].

Our approach consists in using the hierarchical form of the BEM matrix to obtain submatrices corresponding to the far-field interaction, which are known as *admissible blocks* and are constructed in a

tree-fashion structure using a geometric admissibility criterion for clustering [Beb08, Bör10, Hac15]. Typically, hierarchical matrices are constructed such that most of its blocks are admissible and hence the cost for compression and matrix-vector product is dominated by the cost of low-rank approximation of admissible blocks.

Let $A \in \mathbb{C}^{m\times n}$ denote one admissible block, a popular algorithm for approximating A is the Adaptive Cross Approximation (ACA), which has $\mathcal{O}(m+n)$ cost and its accuracy is good enough for many practical applications. The methodology performed by ACA can be seen as a CUR (or skeleton) approximation; this is,

$$A \approx \xi_k = CUR,$$

where $C := A(:, J)$, $R := A(I, :)$ and $U := A^{-1}(I, J) \in \mathbb{C}^{k\times k}$, I and J are sets of indices with cardinality k and must ensure that A(I, J) is invertible. For the case of ACA, I and J are selected adaptively based on a greedy approach to make A(I, J) have maximum absolute determinant among all $k \times k$ submatrices of A. Our approach consists in finding such indices using information from the problem geometry, we call our methodology geometric sampling and provide a general bound for the approximation error $\|A - \xi_k\|$. We analyze different methods to select I and J such as the Nearest-Neighbors (NN) criterion, which have recently been evaluated on multiple kernels in high dimensions showing good accuracy [MB17]. We propose a novel criterion called Gravity Centers Sampling (GCS) which in most cases overcomes the accuracy of ACA and the NN criterion, *c.f.* §6.4, having asymptotically $\mathcal{O}((m+n)k)$ cost to compute.

Skeleton approximations are mainly important when structure in the data must be preserved [MD09]. For BEM matrices, preserving data structure is not a priori relevant, our interest on CUR approximations is to achieve linear complexity. Note that preserving data approximations are tailored for the development of supervised machine learning algorithms that can predict the most representative source and target points by simply analyzing properties of the domains where the problem is posed.

There exist randomized approaches to select indices I and J that can achieve linear-time complexity algorithms, *e.g.* via uniformly random selection [MB17], and even sublinear cost algorithms such as the one presented in [CD13], however their accuracy is not always guaranteed, see *e.g.* [MB17]. The methodologies presented in this chapter are purely algebraic (no kernel dependency), deterministic and can be obtained in linear time. Related works are the IE-QR algorithm [OL04], which constructs a low-rank QR approximation using the modified Gram-Schmidt algorithm and costs $\mathcal{O}(N^{3/2})$, with $N = \max(m, n)$; the IES3 algorithm [KL98], a kernel independent method for electromagnetic simulations which costs $\mathcal{O}(N \log(N))$; and Interpolative Decompositions [BMD09, VM17], which rely on rank-revealing QR factorizations [GE96] and cost $\mathcal{O}(mnk)$.

This chapter is structured as follows. Section 6.2 presents classical methods to compute low-rank approximations based on CUR decompositions. Section 6.3 presents the notion of geometric sampling to create a CUR approximation. We provide an algorithm and prove a relative error bound that can be used for any geometric sampling method. Section 6.4 presents and discusses several numerical experiments to validate our algorithm by using different types of geometries and integral kernels. Finally, Section 6.5 concludes the chapter.

6.2 CUR approximations

In this chapter, we consider a complex matrix $A \in \mathbb{C}^{m\times n}$. We consider notations from Chapter 4, and we denote A^* the conjugate transpose of A. Let row and column indices $I = \{i_1, \cdots, i_k\}$ and $J = \{j_1, \cdots, j_k\}$

be chosen such that $A(I, J) \in \mathbb{C}^{k\times k}$ is non-singular. The CUR approximation of A has the form

$$A \approx CUR, \tag{6.2.1}$$

where $C := A(:, J) \in \mathbb{C}^{m\times k}$, $R := A(I, :) \in \mathbb{C}^{k\times n}$, and $U := A(I, J)^{-1}$. Equation (6.2.1) is also known as skeleton approximation [GT01, GOS+08]. The search of I and J is known as *sampling*. Note that if $\operatorname{rank}(A) = k$, then its skeleton approximation is exact, this is A = CUR.

Error of CUR approximation

Let us consider the indices $\tilde{I} := [I, \{1, \cdots, m\}\backslash I]$, and $\tilde{J} := [J, \{1, \cdots, n\}\backslash J]$, such that

$$A(\tilde{I}, \tilde{J}) := \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{6.2.2}$$

where $A_{11} = A(I, J) \in \mathbb{C}^{k\times k}$. A simple decomposition of A follows as

$$A(\tilde{I}, \tilde{J}) = \underbrace{\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}}_{=:C(\tilde{I},:)} \underbrace{A_{11}^{-1}}_{=:U} \underbrace{\begin{bmatrix} A_{11} & A_{12} \end{bmatrix}}_{=:R(:,\tilde{J})} + \begin{bmatrix} 0 & 0 \\ 0 & S(A_{11}) \end{bmatrix} \equiv C(\tilde{I}, :)UR(:, \tilde{J}) + \begin{bmatrix} 0 & 0 \\ 0 & S(A_{11}) \end{bmatrix}, \tag{6.2.3}$$

where $S(A_{11}) := A_{22} - A_{21}A_{11}^{-1}A_{12}$ is known as the Schur complement of A_{11}. Hence, the approximation error is given as

$$\|A - CUR\| = \|A(\tilde{I}, \tilde{J}) - C(\tilde{I}, :)UR(:, \tilde{J})\| = \|S(A_{11})\|, \tag{6.2.4}$$

where $\|\cdot\|$ stands for any unitarily invariant norm and the maximum norm. Hence, to get a good rank-k CUR approximation we need to sample I and J such that the norm of the Schur complement of A(I, J) is small. Next, we consider a sub-optimal sampling technique consisting in finding I and J to make A(I, J) have *maximal volume*, *i.e.* maximal absolute determinant among all $k \times k$ submatrices of A.

Theorem 6.1. *Consider* $A \in \mathbb{C}^{m\times n}$*, and row and column indices* I *and* J *respectively, with* $|I| = |J| = k$*. Define* $G := A(I, J) \in \mathbb{C}^{k\times k}$*. If* G *is non-singular and has maximal volume among all* $k \times k$ *submatrices of* A*, then*

$$\|A - CUR\|_{\max} \le (1 + k)\, \sigma_{k+1}, \tag{6.2.5}$$

$$\|A - CUR\|_{\max} \le (1 + k)^2 \cdot \min_{B\in\mathbb{C}_k^{m\times n}} \|A - B\|_{\max}, \tag{6.2.6}$$

where $C := A(:, J)$, $R := A(I, :)$, *and* $U := G^{-1}$.

Proof. Inequality (6.2.5) is proved in [GT01, Thm. 2.1], and (6.2.6) in [GT11, Thm. 1]. □

Although the sampling from Theorem 6.1 is nearly optimal, finding submatrices of maximal volume is NP-hard [ÇMI13].

Algorithm 3, which is adapted from [Beb08, Alg. 3.1], computes $A(:, J) \cdot A^{-1}(I, J) \cdot A(I, :) \equiv CUR$ as sum of rank-one matrices. The advantages of this form is that we can update the choice of the selected rows and columns adaptively, and it also allows to monitor the evolution of the determinant of the submatrix formed by the selected indices at a given rank of approximation.

Data: An integral kernel $\mathcal{G} : \mathbb{R}^{d\times d} \longrightarrow \mathbb{C}$, Indices: I and J, each of size k,
Source and target points: $X = [x_1, \cdots, x_m]$ and $Y = [y_1, \cdots, y_n]$
Result: A matrix ξ_k of rank at most k and given as sum of rank-one matrices

```
for h = 1 ⟶ k do
    Set i = I(h) and j = J(h) ;
    ṽ_h := [G(x_i, y_1), ⋯, G(x_i, y_n)];
    u_h := [G(x_1, y_j), ⋯, G(x_m, y_j)]^T;
    for l = 1 ⟶ h − 1 do
        ṽ_h := ṽ_h − u_l(i) v_l
    end
    if ṽ_h(j) vanishes then
        Update column index j = argmax_{s=1,⋯,n} |ṽ_h(s)|
    end
    Set δ(h) = ṽ_h(j) ;
    Normalize v_h := ṽ_h/δ(h) ;
    for l = 1 ⟶ h − 1 do
        u_h := u_h − v_l(j) u_l
    end
end
```

Algorithm 3: Skeleton approximation with fixed pivots

Algorithm 3 requires $(m + n)k$ evaluations of kernel function $\mathcal{G}$ and $\mathcal{O}\left((m + n)k^2\right)$ complex operations. When it halts, we get a rank-k matrix

$$\xi_k := \sum_{h=1}^{k} u_h v_h \equiv \mathrm{CUR}. \tag{6.2.7}$$

This approximation only requires $(m + n)k$ units of storage. Defining $M_k := A(I, J)$, we can also obtain the volume of the submatrix obtained by our choice of row and column indices, it is given as [Beb00, Lem. 2],

$$|\det(M_k)| = \left|\prod_{i=1}^{k} \delta(i)\right|. \tag{6.2.8}$$

In Section 6.4 we plot the value $|\det(M_k)|$ for different sampling techniques, to analyze its impact on increasing the approximation accuracy.

6.3 Linear-time CUR approximation via Geometric Sampling

In this section, we present the concept of *geometric sampling* to select row and column indices I and J by using information from the geometry of the source and target points. Then, a CUR approximation directly follows by using the theory from the previous section.

6.3.1 Geometrical sampling

Algorithm 4 shows our sampling technique. We select $t > k$ (oversampling) points from the target domain and store them into an index vector $\tilde{J}$ which defines a matrix $\tilde{C} := A(:, \tilde{J}) \in \mathbb{C}^{m\times t}$ of sampled

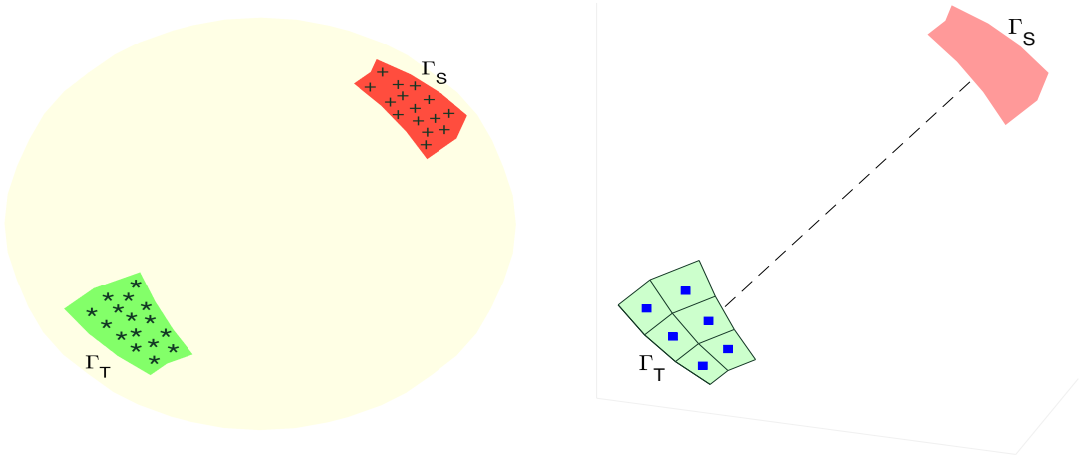
columns. Then, we work on the m-dimensional space, selecting a set of k column indices J corresponding to the *most significant* columns of $\tilde{C}$, we do this by computing the pivoted QR factorization $\tilde{C}(:, p_c) = \hat{Q}\hat{R}$. Then, we set $Q = \hat{Q}(:, 1 : k)$ and perform a truncated pivoted QR factorization on matrix Q^T to obtain a permutation vector p_r. Finally we return $I = p_r(1 : k)$ and $J = p_c(1 : k)$ from which a CUR approximation directly follows as done in §6.2, *c.f.* Algorithm 3.

Data: Approximation Rank k; Source and target points: $X = [x_1, \cdots, x_m]$ and $Y = [y_1, \cdots, y_n]$
Result: Low-rank CUR approximation of A

1 Set oversampling: $t = 2^l$ such that $2^l > k > 2^{l-1}$;
2 Decompose Y into t subdomains, see Appendix B.3.2 ;
3 Form J with the t indices of target points closest to the gravity centers of subdomains;
4 Set $C = A(:, J)$ and compute its pivoted QR factorization $\tilde{C}(:, p_c) = \hat{Q}\hat{R}$;
5 Set $Q = \hat{Q}(:, 1 : k)$ and compute the truncated QR factorization of Q^T to get permutation p_r ;
6 Set $J = p_c(1 : k)$ and $I = p_r(1 : k)$.
7 Return $CUR = A(:, J) \cdot A^{-1}(I, J) \cdot A(I, :)$, which can be computed via Algorithm 3.

Algorithm 4: CUR with gravity centers sampling, CUR_GCS

Remark 6.1. Note that the permutation vector p_r from Algorithm CUR_GCS, would be the same if we instead perform the QR factorization of the conjugate transpose Q^*. This is true since a simple algebraic effort shows that for any $M \in \mathbb{C}^{m\times n}$ with QR factorization $MP = QR$, it holds that $\overline{M}P = \overline{Q}\,\overline{R}$ is the pivoted QR factorization of $\overline{M}$ (matrix with complex conjugated entries).



(a) Interaction of source (Γ_S) and target (Γ_T) points, discretized by $X = \{x_1, \cdots, x_n\}$ (marked with +) and $Y = \{y_1, \cdots, y_m\}$ (marked with $*$) respectively.

(b) Selection of representative targets points (squares), as points closest to the gravity centers of subdomains of Γ_T.

Figure 6.1: Interaction of distant subdomains on a sphere, and selection of representative target points.

The computational cost of CUR_GCS is given as: $\mathcal{O}(n \log_2(t))$ floating point operations to obtain J (see appendix B.3.2), $\mathcal{O}(mt^2)$ complex operations to perform a truncated QR factorization on C, $\mathcal{O}(nk^2)$ complex operations to perform a truncated QR factorization on Q^T and $\mathcal{O}\left((m+n)k^2\right)$ complex operations to get the CUR approximation. Thus, the total cost is $\mathcal{O}(mt^2 + nk^2)$. Also, note that we do not

need to form the whole matrix A, we only need $mt + nk$ evaluations of the kernel function.

Figure 6.1 illustrates the procedure of algorithm CUR_GCS, it displays a spherical domain with source and target distant subdomains. We use the *geometrically balanced* partition technique, *c.f.* [Beb08, Sec. 1.4.1] and [Bör10, Alg.2], to decompose the target domain into $t = 6$ subdomains, and then six target points (blue squares) are selected as the ones closest to the gravity centers of the subdomains. In appendix B.3.1 we provide a MATLAB code for algorithm CUR_GCS, and in appendix B.3.2 we provide the code for the gravity centers sampling technique.

Note that we can easily modify Algorithm 4 to get alternative CUR approximations, by changing the partition technique. For instance, if using the Nearest-Neighbors criterion instead of the gravity centers criterion in line 2 of Algorithm 4, we obtain a new algorithm to which we refer to as CUR_NNS, *c.f.* appendix B.3.3. The Nearest-Neighbors technique selects t target points as the ones closest to the source domain, this has been recently studied in [MB17]. In next subsection, we prove a bound on the CUR approximation error for an arbitrary domain partitioning technique, and in section 6.3.3 we discuss the advantages of the partitioning technique of our algorithm CUR_GCS over CUR_NNS.

6.3.2 Bound on the error of CUR approximation with geometric sampling

Consider that geometric sampling has been performed selecting indices I and J, with $|\mathrm{I}| = |\mathrm{J}| = k$, such that $\mathrm{A}(\mathrm{I},\mathrm{J})$ is non-singular. Let $\tilde{\mathrm{I}} := [\mathrm{I}, \{1,\cdots,m\}\backslash\mathrm{I}]$, and $\tilde{\mathrm{J}} := [\mathrm{J}, \{1,\cdots,n\}\backslash\mathrm{J}]$ and let us apply a truncated-QR factorization in the permuted matrix $\mathrm{A}(\tilde{\mathrm{I}},\tilde{\mathrm{J}})$, this is

$$\mathrm{A}(\tilde{\mathrm{I}},\tilde{\mathrm{J}}) := \begin{bmatrix} \mathrm{A}_{11} & \mathrm{A}_{12} \\ \mathrm{A}_{21} & \mathrm{A}_{22} \end{bmatrix} = \check{\mathrm{Q}}\check{\mathrm{R}} \equiv \begin{bmatrix} \mathrm{Q}_{11} & \mathrm{Q}_{12} \\ \mathrm{Q}_{21} & \mathrm{Q}_{22} \end{bmatrix} \begin{bmatrix} \mathrm{R}_{11} & \mathrm{R}_{12} \\ 0 & \mathrm{R}_{22} \end{bmatrix}. \tag{6.3.1}$$

Next, we use an idea presented in a previous paper [GCD18] for the case of real matrices, where the authors observed that $\mathrm{S}(\mathrm{A}_{11}) = \mathrm{S}(\mathrm{Q}_{11})\mathrm{R}_{22}$, with

$$\mathrm{S}(\mathrm{Q}_{11}) := \mathrm{Q}_{22} - \mathrm{Q}_{21}\mathrm{Q}_{11}^{-1}\mathrm{Q}_{12} = \mathrm{Q}_{22}^{*-1}, \tag{6.3.2}$$

where Q_{22}^{*} is the conjugate transpose of Q_{22}, and the last equality can be verified by computing $\mathrm{Q}_{22}^{*}\mathrm{S}(\mathrm{Q}_{11})$ and using the fact that $\check{\mathrm{Q}}\check{\mathrm{Q}}^{*} = \mathrm{I}_m$. Then, Equation (6.2.4) is rewritten

$$\|\mathrm{A} - \mathrm{CUR}\|_2 = \|\mathrm{S}(\mathrm{A}_{11})\|_2 \leq \|\mathrm{Q}_{22}^{*-1}\|_2 \|\mathrm{S}(\mathrm{R}_{22})\|_2, \tag{6.3.3}$$

where $\mathrm{C} = \mathrm{A}(:,\mathrm{J})$, $\mathrm{R} = \mathrm{A}(\mathrm{I},:)$ and $\mathrm{U} = \mathrm{A}^{-1}(\mathrm{I},\mathrm{J})$ and by using the CS decomposition [GVL96, Thm.2.6.3], which tells us that $\sigma_{\min}(\mathrm{Q}_{11}) = \sigma_{\min}(\mathrm{Q}_{22})$, finally we get the bound

$$\|\mathrm{A} - \mathrm{CUR}\|_2 \leq \frac{1}{\sigma_{\min}(\mathrm{Q}_{22})}\|\mathrm{R}_{22}\|_2 = \frac{1}{\sigma_k(\mathrm{Q}_{11})}\|\mathrm{R}_{22}\|_2. \tag{6.3.4}$$

Error of column sampling

We first state a theorem to bound the error of column sampling. This bound involves C and its QR factorization. From this we then derive a bound for CUR approximation.

Theorem 6.2. *Consider* $\mathrm{A} \in \mathbb{C}^{m\times n}$ *and a set of indices* J, *with* $|\mathrm{J}| = t$. *Let* $\mathrm{C} := \mathrm{A}(:,\mathrm{J})$ *be at least rank-k and consider its QR column pivoted factorization,*

$$\mathrm{C}(:,p_c) = \hat{\mathrm{Q}}\hat{\mathrm{R}} \equiv \hat{\mathrm{Q}} \begin{matrix} \\ k \\ m-k \end{matrix} \begin{matrix} \begin{matrix} k & t-k \end{matrix} \\ \begin{bmatrix} \hat{\mathrm{R}}_{11} & \hat{\mathrm{R}}_{12} \\ 0 & \hat{\mathrm{R}}_{22} \end{bmatrix} \end{matrix} \tag{6.3.5}$$

where $\hat{Q} \in \mathbb{C}^{m\times m}$ *is unitary,* $\hat{R} \in \mathbb{C}^{m\times t}$*, and* p_c *is a permutation vector of size* t*. Define* $Q = \hat{Q}(:, 1 : k)$*, then*

$$E := \|(I_m - QQ^*)A\|_2 \le \sqrt{f^2(k,t) + k\left(\frac{2^{k-1}\|A\|_F}{\mu}\right)^2} \cdot \sigma_{k+1}(A). \tag{6.3.6}$$

where μ *is the minimum, in absolute value, of the first* k *diagonal entries of* $\hat{R}$ *and* $f(k,t)$ *is defined in Table 4.1.*

Note that $\|R_{22}\|_2 = \|(I_m - QQ^*)A\|_2$ according to (4.3.9), hence one of the factors of the bound on the CUR approximation error (6.3.4) follows from the proof of Theorem 6.2. This bound can also be interpreted as the error of a rank-k truncated QR approximation with geometric sampling as pivoting technique, *c.f.* §4.3.

Proof. Let us consider $\hat{J} = J(p_c)$ and define $p = [\hat{J}, \{1, \cdots, n\}\backslash\hat{J}]$, we get

$$\hat{Q}^T A(:,p) = \underbrace{\begin{array}{c} \\ k \\ m-k \end{array}\begin{array}{c} \begin{array}{ccc} k & t-k & n-t \end{array} \\ \begin{bmatrix} \hat{R}_{11} & \hat{R}_{12} & \hat{B}_1 \\ 0 & \hat{R}_{22} & \hat{B}_2 \end{bmatrix} \end{array}}_{=:\tilde{R}}, \tag{6.3.7}$$

where $\hat{B}_1 \in \mathbb{C}^{k\times(n-t)}$ and $\hat{B}_2 \in \mathbb{C}^{(m-k)\times(n-t)}$. Note that approximating A by $QQ^*A = Q[\hat{R}_{11}, \hat{R}_{12}, \hat{B}_1]$, we get (*c.f.* (4.3.9))

$$\|(I_m - QQ^*)A\|_2 = \|[\hat{R}_{22}, \hat{B}_2]\|_2,$$

and bounding the right hand side of the previous equation would give us the desired bound of the theorem. However, we do not want to compute $\hat{B}_2$ and also, this form does not allow to directly get a bound as in (6.3.6). Hence, we proceed to use a technique developed by Gu and Eisenstat [GE96, Thm. 3.2]. Let us define the following block diagonal matrix,

$$Z := \begin{bmatrix} \alpha\hat{R}_{11} & \\ & [\hat{R}_{22}, \hat{B}_2] \end{bmatrix} = \begin{bmatrix} \hat{R}_{11} & \hat{R}_{12} & \hat{B}_1 \\ 0 & \hat{R}_{22} & \hat{B}_2 \end{bmatrix} \underbrace{\begin{bmatrix} \alpha I_k & -\hat{R}_{11}^{-1}[\hat{R}_{12}, \hat{B}_1] \\ & I_{n-k} \end{bmatrix}}_{=:W} \equiv \tilde{R}W$$

where $\alpha = \sigma_{\max}([\hat{R}_{22}, \hat{B}_2])/\sigma_{\min}(\hat{R}_{11})$. Note that this choice of α ensures that $\sigma_{k+1}(Z) = \sigma_1([\hat{R}_{22}, \hat{B}_2])$. Next, using Theorem 4.2 we get,

$$E = \|[\hat{R}_{22}, \hat{B}_2]\|_2 = \sigma_{k+1}(Z) \le \sigma_{k+1}(\tilde{R})\|W\|_2 = \sigma_{k+1}(A)\|W\|_2, \tag{6.3.8}$$

where the last equality holds since $\hat{Q}$ is unitary. Then, to complete the proof, it remains to bound $\|W\|_2$. We proceed as follows,

$$\begin{aligned}
\|W\|_2^2 &\le 1 + \|\hat{R}_{11}^{-1}[\hat{R}_{12}, \hat{B}_1]\|_2^2 + \alpha^2 && (6.3.9)\\
&= 1 + \|[\hat{R}_{11}^{-1}\hat{R}_{12}, \hat{R}_{11}^{-1}\hat{B}_1]\|_2^2 + \|\hat{R}_{11}^{-1}\|_2^2\left(\|[\hat{R}_{22}, \hat{B}_2]\|_2^2\right) && (6.3.10)\\
&\le 1 + \|\hat{R}_{11}^{-1}\hat{R}_{12}\|_F^2 + \|\hat{R}_{11}^{-1}\hat{B}_1\|_F^2 + \|\hat{R}_{11}^{-1}\|_F^2\left(\|\hat{R}_{22}\|_F^2 + \|\hat{B}_2\|_F^2\right) && (6.3.11)\\
&\le \left(1 + \|\hat{R}_{11}^{-1}\hat{R}_{12}\|_F^2 + \|\hat{R}_{11}^{-1}\|_F^2\|\hat{R}_{22}\|_F^2\right) + \|\hat{R}_{11}^{-1}\|_F^2\left(\|\hat{B}_1\|_F^2 + \|\hat{B}_2\|_F^2\right). && (6.3.12)
\end{aligned}$$

From the QRCP factorization (6.3.5), we get that (*c.f.* proof of [GE96, Thm. 7.2]),

$$1 + \|\hat{R}_{11}^{-1}\hat{R}_{12}\|_F^2 + \|\hat{R}_{11}^{-1}\|_F^2 \|\hat{R}_{22}\|_F^2 \leq f^2(k,t). \tag{6.3.13}$$

Hence,

$$\|W\|_2^2 \leq f^2(k,t) + \|\hat{R}_{11}^{-1}\|_F^2 \left(\|\hat{B}_1\|_F^2 + \|\hat{B}_2\|_F^2\right). \tag{6.3.14}$$

Next, we observe that

$$\|\hat{R}_{11}^{-1}\|_F \leq \sqrt{k}\|\hat{R}_{11}^{-1}\|_2 \leq \sqrt{k}\frac{2^{k-1}}{\mu},$$

where for the first inequality we use a classic relationship between the spectral and Frobenius norms, and for the last inequality we use a theorem from [Hig02, Thm. 8.14].

From (6.3.7) we get that $\|\hat{B}_1\|_F^2 + \|\hat{B}_2\|_F^2 \leq \|A\|_F^2$. Hence,

$$\|W\|_2^2 \leq f^2(k,t) + k\left(\frac{2^{k-1}\|A\|_F}{\mu}\right)^2, \tag{6.3.15}$$

and the result follows by replacing (6.3.15) in (6.3.8).

□

Remark 6.2. The value μ in the previous theorem depends on k and its inverse can be bounded. Consider $D := \mathrm{diag}(\mathrm{diag}(\hat{R}))$ and define Y such that $R = DY$. Then, using Theorem 4.2 we get $\sigma_i(\hat{R}) \leq \sigma_i(D)\|Y\|_2 = \hat{R}(i,i)\|Y\|_2$. Also, since $\sigma_i(C) = \sigma_i(\hat{R})$ we get

$$\frac{1}{\mu} = \frac{1}{\hat{R}(k,k)} \leq \frac{\|Y\|_2}{\sigma_k(\hat{R})} = \frac{\|Y\|_2}{\sigma_k(C)} \leq \sqrt{\frac{t(t+1)}{2\sigma_k(C)}},$$

where the last inequality holds since all entries of Y are smaller than 1.

Remark 6.3. A bound can also be obtained when the strong rank-revealing factorization is used to factor C. From (4.3.6) we get $\sigma_i(\hat{R}_{11}) \leq \sigma_i(C) \leq f(k,t)\sigma_i(\hat{R}_{11})$, then $\|\hat{R}_{11}^{-1}\|_F \leq \sqrt{k}\|\hat{R}_{11}^{-1}\|_2 \leq \sqrt{k}\frac{f(k,t)}{\sigma_k(C)}$. Hence, by using (6.3.8) and (6.3.14) as done in the proof of the theorem, we obtain

$$E \leq f(k,t)\sqrt{1 + k\left(\frac{\|A\|_F}{\sigma_k(C)}\right)^2} \cdot \sigma_{k+1}(A), \tag{6.3.16}$$

where $f(k,t) = \sqrt{1 + \nu^2 k(t-k)}$ and ν is a parameter of the strong rank revealing QR factorization.

Error of row sampling

Next, let us complete the bound (6.3.4). We use a simple technique found in [GCD18], which is described as follows. Once the set of column indices J, with $|J| = k$, is obtained, define $C := A(:,J)$ with QR factorization $C = QR_{11}$, where $Q \in \mathbb{C}^{m\times k}$ and $R_{11} \in \mathbb{C}^{k\times k}$. According to Algorithm 4, we apply row sampling by performing a truncated pivoted QR on Q^*, *c.f.* Remark 6.1, this is

$$Q^*(:,\tilde{I}) \equiv [Q_{11}^*, Q_{21}^*] = \tilde{Q}[\tilde{R}_1, \tilde{R}_2]. \tag{6.3.17}$$

where $\tilde{Q} \in \mathbb{C}^{k\times k}$ is unitary and $\tilde{R}_1 \in \mathbb{C}^{k\times k}$ is upper triangular. Using (4.3.6), we get

$$1 = \sigma_k(Q) \leq f(k,m)\sigma_k(\tilde{R}_1) \leq f(k,m)\sigma_k(Q_{11}), \tag{6.3.18}$$

where $f(k,m)$ can be obtained from Table 4.1 depending on the algorithm chosen. Hence,

$$\frac{1}{\sigma_k(Q_{11})} \leq f(k,m). \tag{6.3.19}$$

CUR error bound

Finally, let $C := A(:, J(1:k))$, $R := A(I,:)$ and $U = A^{-1}(I,J) \in \mathbb{C}^{k\times k}$. The final bound on the error $\|A - CUR\|_2$ is obtained by replacing the bound from Theorem 6.2 and (6.3.19) in (6.3.4).

6.3.3 Discussion on geometric sampling technique

As presented in Section 6.2, the accuracy of a rank-k CUR approximation greatly depends on the choice of row and column indices I, J, with $|I| = |J| = k$. We need to ensure that matrix $A(I,J)$ is as well conditioned as possible; and we know that if it has maximal volume, then we get a suboptimal approximation.

By construction, our CUR approximation first computes J and then using $C = A(:,J)$ it finds I. Hence, finding I can be performed suboptimally in linear-time by using routines such as the Strong RRQR (see Table 4.1) or *maxvol* [GOS+08] to find k most representative rows of C. Therefore, it is most important to select a good set of column indices and geometric sampling allows to find them in linear-time.

To show why the gravity center criterion from Algorithm CUR_GCS is a good choice, let c_j be the j-th column of C, and by seeing the columns of C as points in $\mathbb{C}^m$, let us compute the volume of the simplex formed by these points. For this, we use the Cayley-Menger determinant [Som58, Pag. 24], the volume of such simplex is given as

$$\mathscr{V}_k := \mu \left| \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & d_{12}^2 & \cdots & d_{1k}^2 \\ 1 & d_{21}^2 & 0 & \cdots & d_{2k}^2 \\ \vdots & & & \ddots & \vdots \\ 1 & d_{k1}^2 & d_{k2}^2 & \cdots & 0 \end{bmatrix} \right|,$$

where $\mu = \frac{(-1)^k}{2^{k-1}(k-1)!^2}$ and $d_{jl} = \|c_l - c_j\|_2$ for $j,l = 1,\cdots,k$.

First, note that $\mathscr{V}_k = 0$ if and only if there are at least two linearly dependent columns. Hence, our selection of J can be seen as an approach to obtain a value of $\mathscr{V}_k$ as large as possible while keeping d_{ij} of the same order of magnitude (this is different from the approach that finds maximal projective volume rectangular submatrices [OZ18] and closely related to the approach of volume sampling [DRVG06]). For the sake of simplicity, let us consider a smooth kernel function $\mathscr{G} : \mathbb{R}\times\mathbb{R} \longrightarrow \mathbb{R}$ to construct A (and hence C). Then, by using the mean value theorem, we get

$$d_{jl}^2 = |y_j - y_l|^2 \sum_{i=1}^{m} |\partial_y \mathscr{G}(x_i, \psi_{lj})|^2,$$

where ψ_{lj} is a real number that lies between y_l and y_j. Hence, the values of d_{jl}^2 are directly related to the distance between the selected target points y. Then, if the selected target points are very close to each other (a behavior that is commonly observed for nearest-neighbors criterion) we get a small

value $\mathscr{V}_k$, while the gravity centers criterion is an approach created to maintain d_{jl}^2 different from zero and to keep the rows linearly independent. According to our experiments, the value of $\mathscr{V}_k$ for matrix C = A(:, J), when J is obtained by the gravity center criterion, is in general greater than the case when the nearest-neighbors or uniformly random selection are used, in some cases by one or two orders of magnitude.

6.4 Numerical Experiments

In this section we numerically show the benefits of our algorithms. We consider the following three kernels encountered in the discretization of elliptic partial differential equations by means of integral equations techniques, see *e.g.* [Ste08, Beb08],

$$\mathscr{G}_{\mathrm{g}}(x,y) = \frac{1}{4\pi\|x-y\|_2}, \quad \mathscr{G}_{\mathrm{e}}(x,y) = \frac{\exp(\imath\|x-y\|_2)}{\|x-y\|_2}, \quad \mathscr{G}_{l}(x,y) = -\frac{1}{2\pi}\log(\|x-y\|_2), \tag{6.4.1}$$

where $\imath$ is the imaginary unit. We construct a matrix $\mathrm{A} \in \mathbb{C}^{m\times n}$ by evaluating one of the above kernels on three-dimensional interaction points, *i.e.* $\mathrm{A}(i,j) = \mathscr{G}(x_i, y_j)$, where $\mathrm{X} := [x_1, \cdots, x_m]$ (red points) are known as sources and $\mathrm{Y} := [y_1, \cdots, y_n]$ (green points) as targets. Domains X and Y hold an admissibility condition given as

$$\min(\mathrm{diam}(\mathrm{X}), \mathrm{diam}(\mathrm{Y})) \leq \eta\, \mathrm{dist}(\mathrm{X}, \mathrm{Y}), \tag{6.4.2}$$

with $\eta < 1$, ensuring that singular values of A exponentially decrease [Beb00, Beb08]. In our plots we report the value η that makes (6.4.2) an equality.

Comparing linear-time algorithms

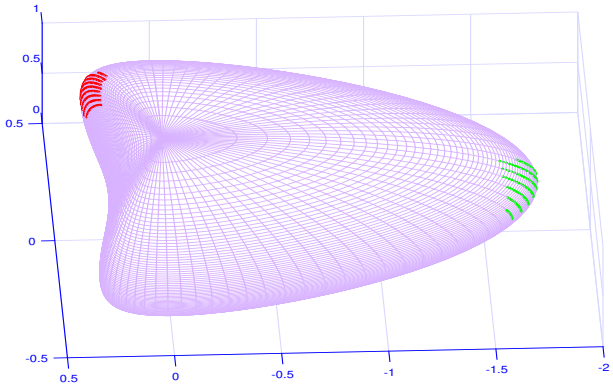
Our first experiments are performed on admissible submatrices taken from global hierarchical matrices. We compare algorithms CUR_GCS and CUR_NNS, introduced in section 6.3.1 to ACA with partial pivoting (ACAp) [Beb00, Alg. 2], for which we only modify the first row pivot by an efficient one proposed in [Beb08, Sec. 3.4.3]. For all three methods we also plot values $\delta(k)$ and $\det(\mathrm{M}_k)$ to show that not necessarily we need to approximate maxvol submatrices to get higher accuracy. In order to show that CUR_GCS produces a quasi-optimal approximation, we also display a line tagged Bound_MaxVol, corresponding to the value $(k+1)\sigma_{k+1}$ given in eq. (6.2.5). In all plots we also show the optimal error obtained by the truncated SVD as a reference curve. These plots are displayed in figures 6.4, 6.7, 6.10 and 6.13.

We also plot the performance of our main algorithm CUR_GCS, and compare it with quadratic cost algorithms QRCP and ACA with full pivoting (ACAf). The latter is a quadratic cost implementation of ACA consisting in iteratively sampling rows and columns using the maximum element of residual matrices, see *e.g.* [Rja02]. We show that our linear-time CUR_GCS algorithm has accuracy comparable with these methods and in some cases even overcomes them. This can be seen in figures 6.5, 6.8, 6.11 and 6.14.

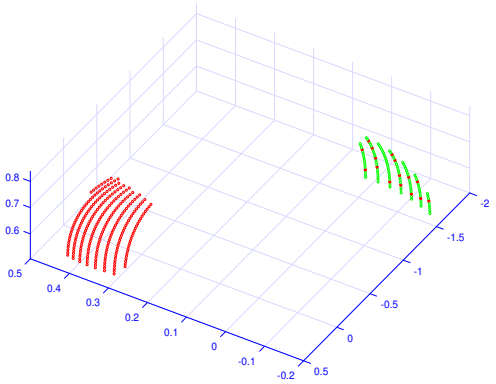
To conclude, we compare the performance of CUR_GCS against ACAp for approximating an entire hierarchical matrix, see section 6.4.5.

6.4.1 BEM matrix from Laplacian kernel

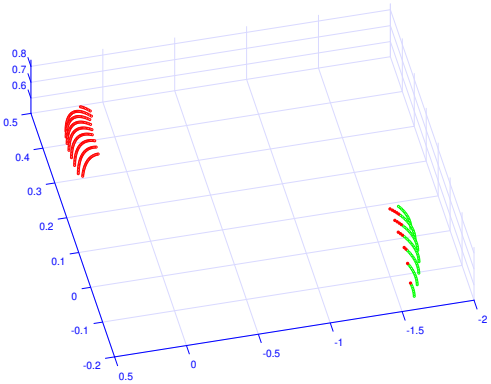
Using the kernel function $\mathcal{G}_l$, we construct matrix A which entries are obtained by evaluating $\mathcal{G}_l$ on source and target points located on a 3D surface proposed in [Beb00], which is shown in Figure 6.2 together with target points sampled by the gravity centers and nearest-neighbors methodologies.



(a) Admissible subdomains interaction



(b) Sampling targets with the gravity centers criterion



(c) Sampling targets with the nearest-neighbors criterion

Figure 6.2: Surface from [Beb00], with admissible subdomains created with $\eta = 0.15$.

In Figure 6.4, we observe that even when the value of $\delta(k)$ corresponding to CUR_GCS is in many cases smaller than the other methods, however we still get better accuracy. For reference, we also show the optimal error obtained by the truncated SVD. Moreover, Figure 6.5 shows that the accuracy of CUR_GCS is comparable to those of quadratic cost algorithms QRCP and ACAf.

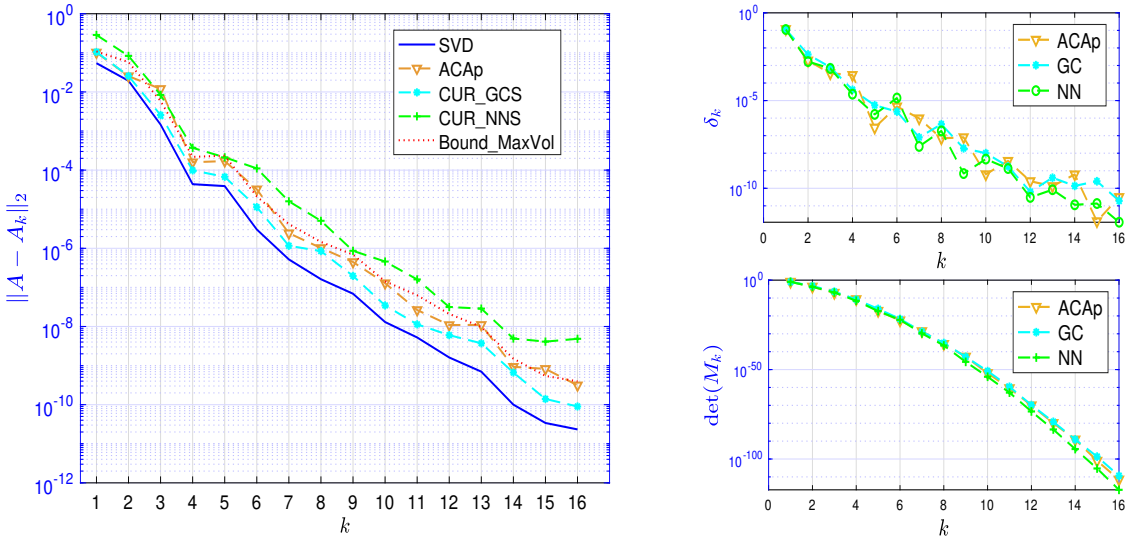


Figure 6.4: Error convergence of CUR approximation with geometric sampling. The values of $\delta(k)$ and $\det(M_k)$ allow to show the method that better approaches a maximal volume submatrix.

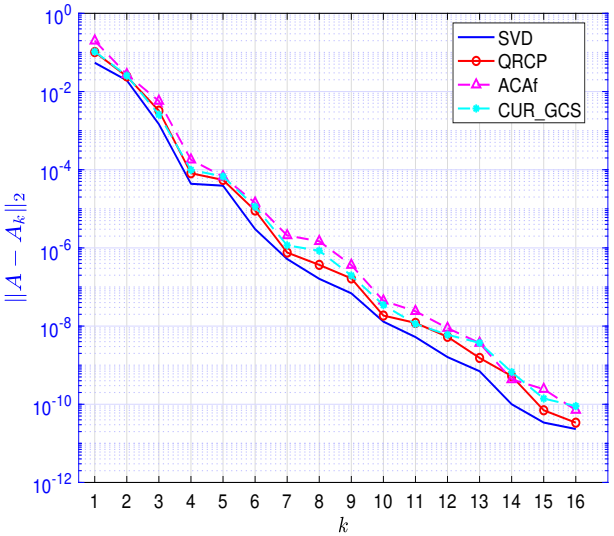


Figure 6.5: Comparison of our linear cost method CUR_GS versus $\mathcal{O}(mnk)$ cost methods QRCP and ACAf.

6.4.2 BEM matrix from Exponential kernel

We use kernel $\mathcal{G}_e$ to construct a complex BEM matrix A using a 3D airplane surface that we construct using MATLAB, see Figure 6.6. Analogously to previous subsection, we show the error convergence for CUR_GCS and compare it with classical methods, showing in all cases a clear improvement, see Figures 6.7 and 6.8 respectively.

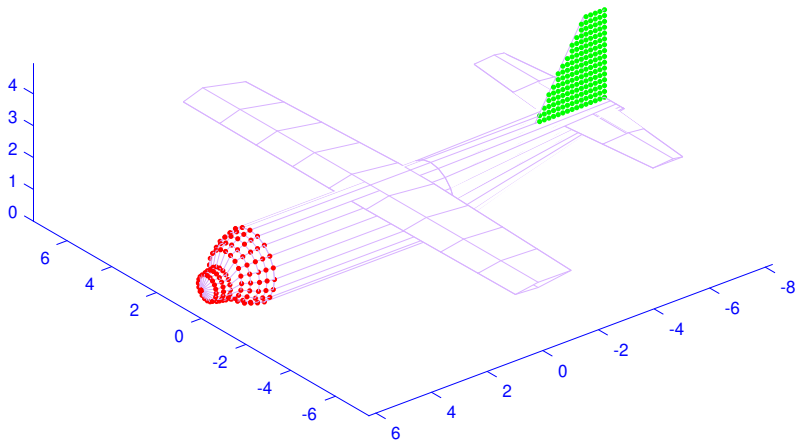


Figure 6.6: Airplane surface with admissible subdomains created with $\eta = 0.22$.

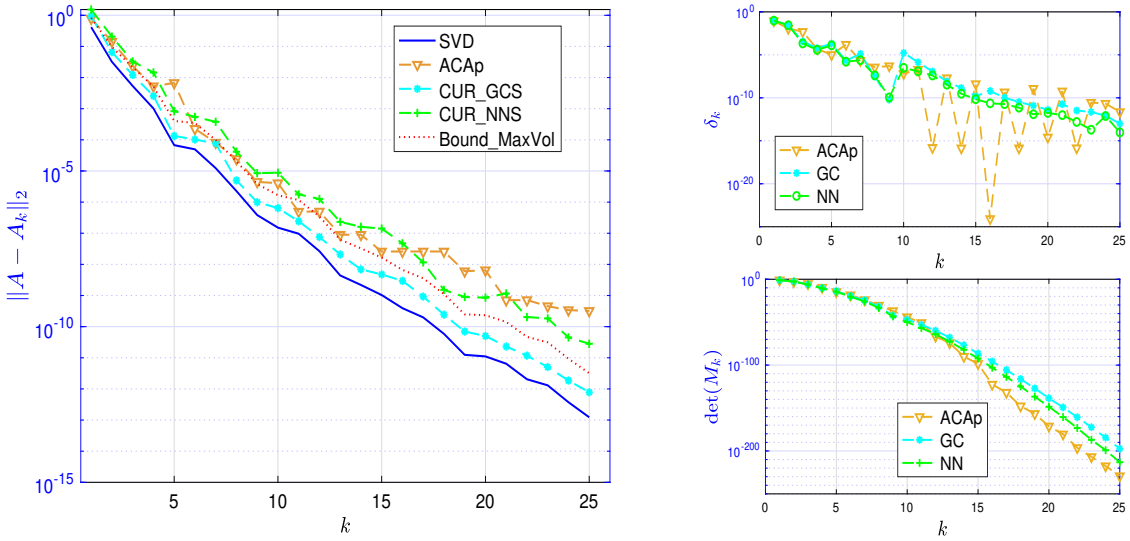


Figure 6.7: Error convergence of CUR approximation with geometric sampling. The values of $\delta(k)$ and $\det(M_k)$ allow to show the method that better approaches a maximal volume submatrix.

In Figure 6.8 we compare our linear cost method CUR_GCS versus quadratic cost methods QRCP and ACAf, showing comparable accuracy and even improving them in some cases.

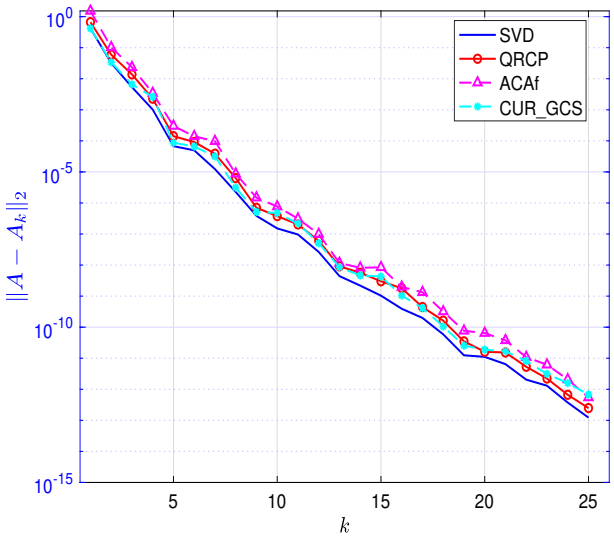


Figure 6.8: Comparison of our linear cost method CUR_GS versus $\mathcal{O}(mnk)$ cost methods QRCP and ACAf.

6.4.3 BEM matrix from Gravity kernel

We use kernel $\mathscr{G}_g$ to construct matrix A using a toroid surface that we construct using MATLAB, see Figure 6.9.

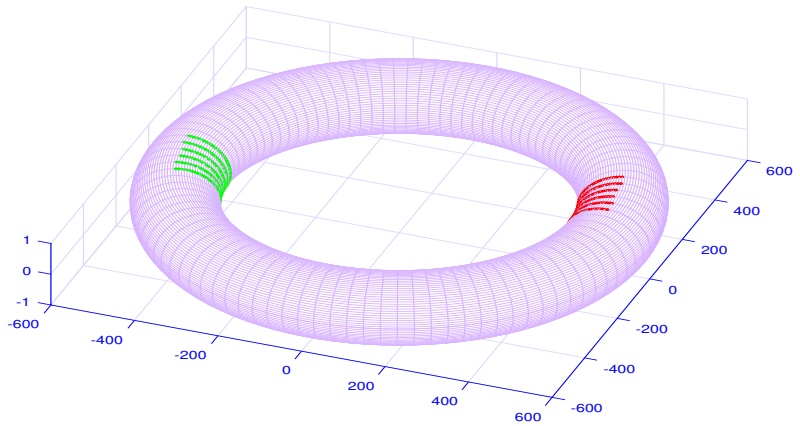


Figure 6.9: Toroid surface with admissible subdomains created with $\eta = 0.22$.

In Figure 6.10 we plot convergence curves for linear-time algorithms, showing that CUR_GCS has better accuracy than ACAp (about one order of magnitude). In fact, for this case study, CUR_GCS has practically the same accuracy as quadratic cost algorithms, see Figure 6.11.

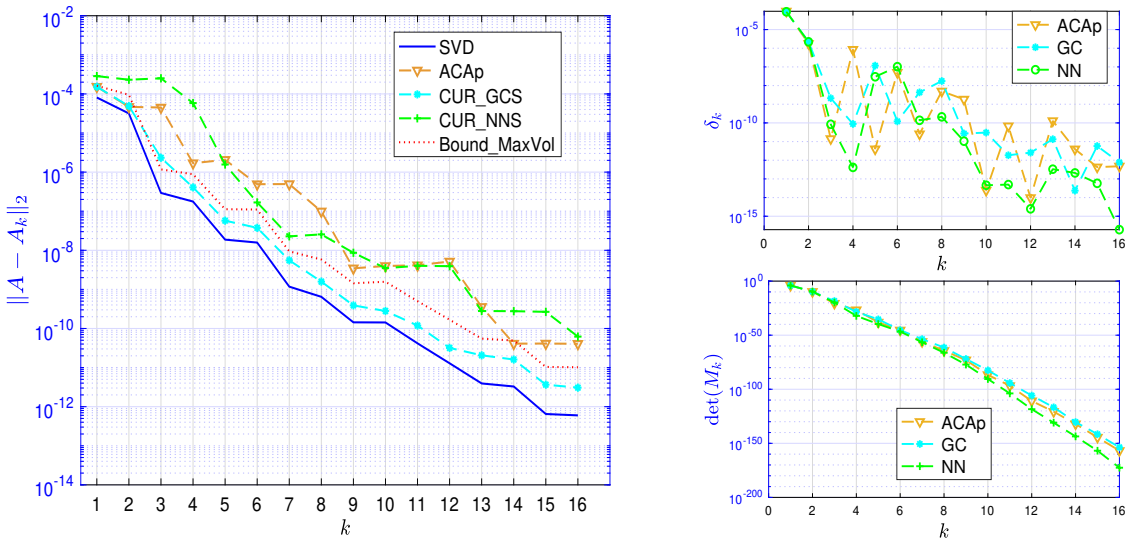


Figure 6.10: Error convergence of CUR approximation with geometric sampling. The values of $\delta(k)$ and $\det(M_k)$ allow to show the method that better approaches a maximal volume submatrix.

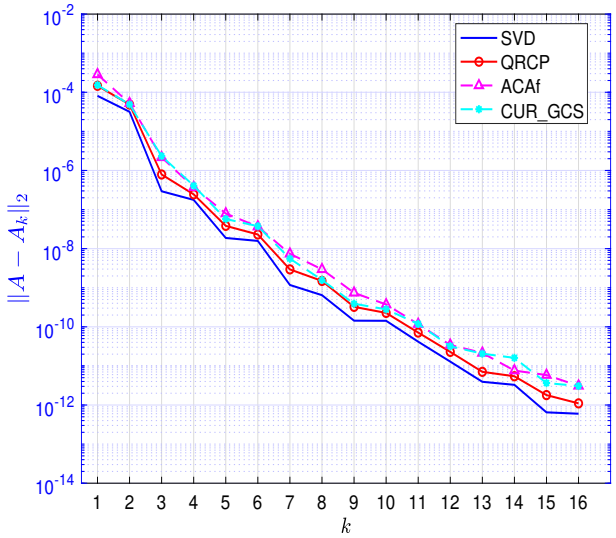


Figure 6.11: Comparison of our linear cost method CUR_GS versus $\mathcal{O}(mnk)$ cost methods QRCP and ACAf.

6.4.4 When ACA with partial pivoting fails

Next, we evaluate our algorithms on a challenging problem reported in [Beb08, Sec. 3.4.3]. We build matrix A with a kernel given as

$$\mathcal{G}_b(x, y) = \frac{(x - y) \cdot n_x}{4\pi \|x - y\|_2}, \tag{6.4.3}$$

where n_x is a unit vector normal to Γ_X at point x, and Γ_X is a surface from where the discretization points X are taken, see Figure 6.12.

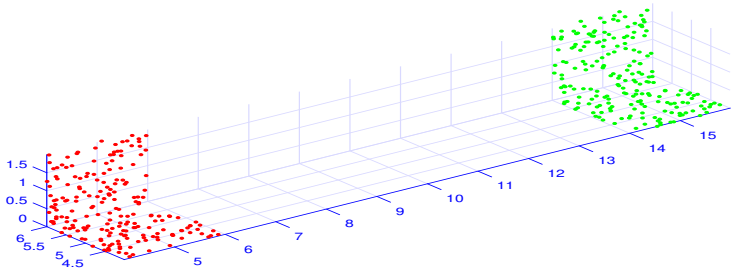


Figure 6.12: Two admissible subdomains, created with $\eta = 0.39$. By computing their interaction via the kernel function (6.4.3), they produce a matrix of type (6.4.4).

When such kernel is evaluated in domains from Figure 6.12, we can get a matrix A of type

$$\begin{bmatrix} 0 & A_{12} \\ A_{21} & 0 \end{bmatrix}, \tag{6.4.4}$$

and a simple analysis shows that under this configuration ACAp fails to converge. Even though there are improvements of ACA sampling to ensure convergence, see *e.g. [Beb08, Sec. 3.4.3]*, our methodology is accurate and much simpler, see Figure 6.13.

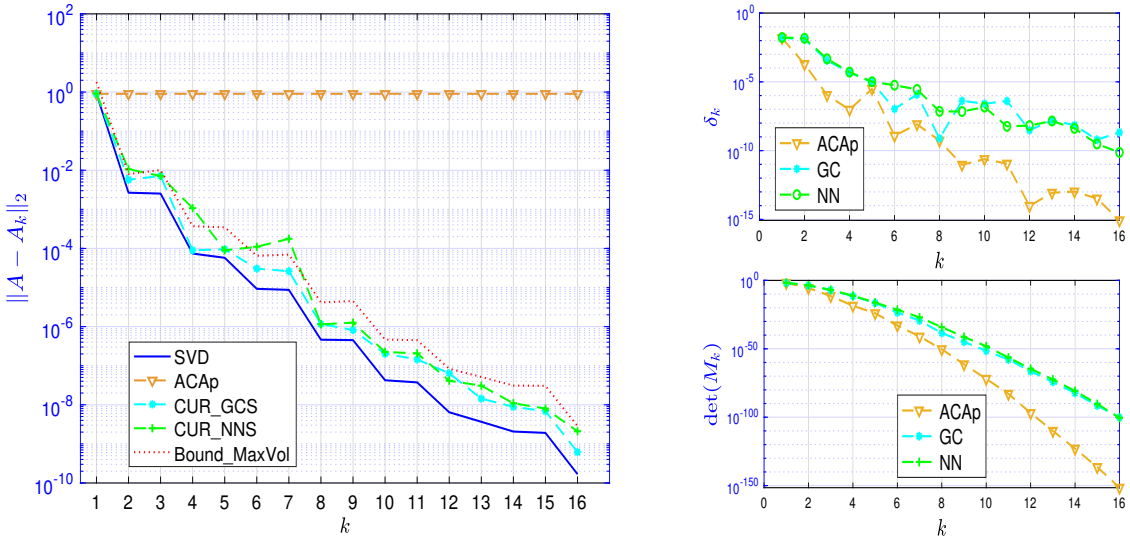


Figure 6.13: Error convergence of CUR approximation with geometric sampling. The values of $\delta(k)$ and $\det(M_k)$ allow to show the method that better approaches a maximal volume submatrix.

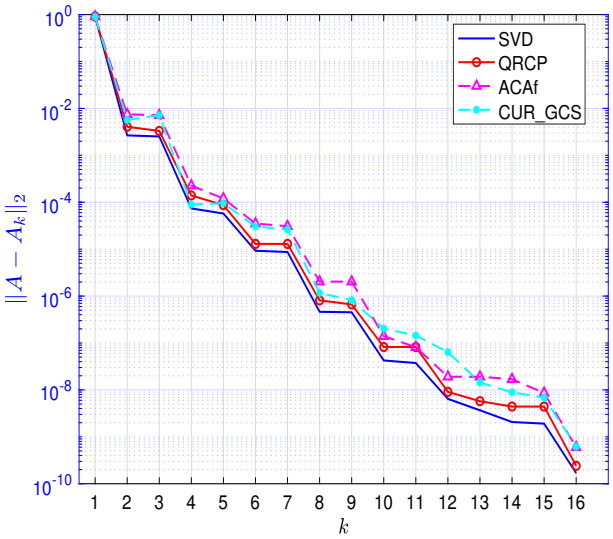


Figure 6.14: Comparison of our linear cost method CUR_GS versus $\mathcal{O}(mnk)$ cost methods QRCP and ACAf.

6.4.5 *Approximating a Hierarchical matrix*

To finalize our numerical experiments, we compare the performance of CUR_GCS and ACAp to approximate all the admissible blocks of a hierarchical matrix, obtained from the discretization of the integral

$$\frac{1}{4\pi}\int_{\Gamma}\int_{\Gamma}\frac{1}{\|x-y\|_2}\mathrm{dxdy},$$

where Γ is the surface of a cavity domain, see figure below.

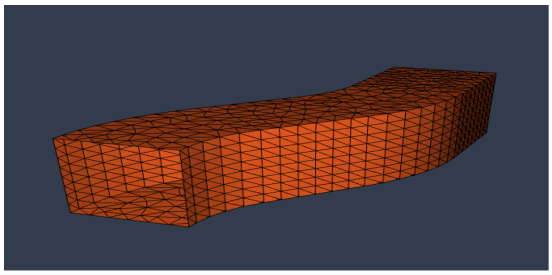


Figure 6.15: 3D cavity domain.

We use Galerkin discretization using a triangular mesh as in Figure 6.15, obtaining a square matrix $\mathrm{A} \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$, with entries given as

$$\mathrm{A}(i,j) = \frac{1}{4\pi}\int_{\tau_i}\int_{\tau_j}\frac{1}{\|x-y\|_2}\varphi_i(x)\mathrm{dx}\varphi_j(y)\mathrm{dy},$$

where φ_i and φ_j are polynomials of degree one and τ_i, τ_j are triangular elements from the discretization mesh.

The following figure shows the approximation error and execution time to form the hierarchical matrix corresponding to A, where the admissible blocks are approximated by low-rank matrices created with ACAp and CUR_GCS. We can confirm the linear behavior of the computational cost of our algorithm CUR_GCS as presented in the theory. We can clearly see the tradeoff between amount of computation and accuracy. For the experiment, we have used C++ libraries `HTool`[1] and `BemTool`[2]. We have run the experiment using 4 MPI processes on a MacBook Pro with 4 cores and frequency of 2.5 GHz.

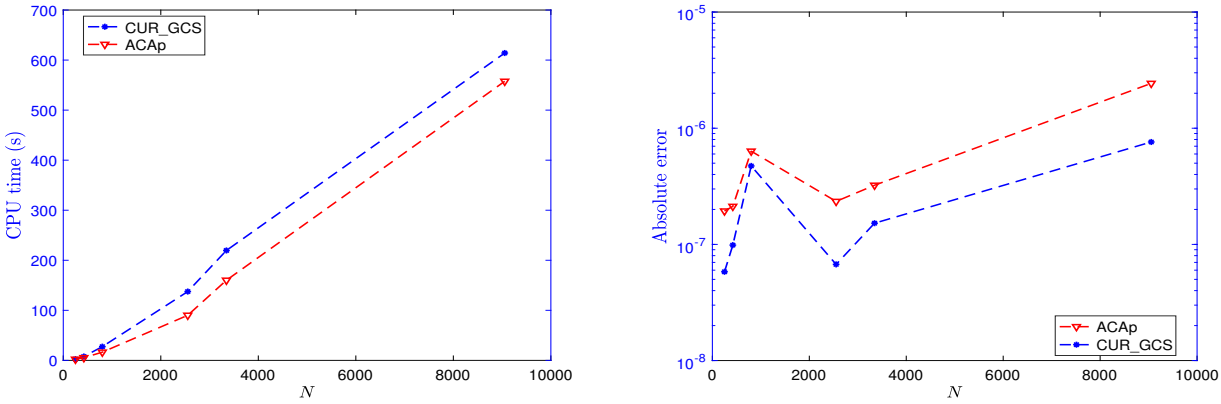


Figure 6.16: Comparison of the execution time and absolute approximation error between ACAp and CUR_GCS .

From Figure 6.16, we clearly see the improvement in the approximation error when using CUR_GCS with respect to ACAp, at the expense of performing more arithmetics. We believe that for large scale matrices, an optimized parallel implementation of CUR_GCS (or another CUR created with geometric sampling) would be faster and more accurate than current parallel implementations of ACAp, see *e.g.* [Beb08, Sec. 3.4.6]. This is because CUR_GCS depends on QR truncated factorizations that can be computed with small communication cost, see *e.g.* our algorithm CALRQR in appendix A.1. And communication between processors is known to be a bottleneck for large scale problems running on computer clusters, and optimizing communication leads to considerable speed-ups [DGHL08, DGHL12].

6.5 Conclusions of the chapter

We have presented a technique called geometric sampling to construct linear-time CUR algorithms for admissible blocks of a hierarchical matrix coming from the discretization of a BEM problem. We have presented a relative error bound for geometric column sampling, which we then extended to a bound for a CUR approximation. Also, this bound can directly be used for truncated QR factorizations, interpolative decompositions and other methods that involve the selection of representative columns. Numerical experiments showed good performance for different integral kernels evaluated on challenging domains. We compared two CUR algorithms created with geometric sampling against ACA with partial pivoting technique. The results showed that our main algorithm CUR_GCS is very efficient and even can handle convergence issues of ACA with partial pivoting, having accuracy comparable with quadratic cost algorithms QRCP and ACA with full pivoting.

[1]Developed by Pierre Marchand, https://github.com/PierreMarchand20/htool
[2]Developed by Xavier Claeys, https://github.com/xclaeys/BemTool

CHAPTER 7

Conclusion

In this thesis, we have first contributed to the development of the local multi-trace formulation, we have obtained a closed form for the inverse of the local multi-trace operator of a model transmission problem and shown that this inverse operator can be used for preconditioning a general transmission problem. Then, we have proven that the local multi-trace formulation is stable for Maxwell equations on a model domain, which was an open question in the literature.

Then, we have extended the literature on classical low-rank approximations for general rectangular matrices. For approximations based on QR factorizations, we have proven a new bound for the classical column pivoting technique and for the case where an arbitrary pivoting strategy is used. For approximations based on subspace iteration, we have proven that the approximations of singular vectors converge exponentially. And then, we have presented affine low-rank approximation techniques to increase the accuracy of QR and subspace iteration algorithms.

To relate both parts of this thesis, we have presented a methodology based on hierarchical matrices and generalized CUR approximations for accelerating the solution of linear systems arising from the discretization of boundary integral equations. We have introduced the concept of geometric sampling that in linear time allows to obtain an accurate low-rank CUR approximation that can be used for compression and matrix-vector product acceleration. Our general bound on the error of geometric sampling, allows to use this technique for pivoted QR factorizations and interpolative decomposition methods, allowing to reduce their computational cost with provable convergence.

Open questions and future work

- An open question for MTF is if these formulations can be extended by using other transmission

conditions, in the sense of optimal Schwarz iterations.

- For local multi-trace formulations, it is an open question to find closed form inverses for more general situations and explore its performance in other engineering and mathematical fields.
- The stability of the local multi-trace formulation for arbitrary Lipschitz domains with junction points is also an open question.
- For geometric sampling, it remains as an open research topic to evaluate its efficiency on unsupervised feature selection strategies for matrices that not necessarily have exponentially decreasing singular values, such as non-admissible blocks of hierarchical matrices.
- The development of geometric sampling techniques to deal with highly oscillatory kernels is an interesting research topic, where more sophisticated geometric properties of the surfaces, containing source and target points, need to be explored.

Bibliography

[ABB+99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, 1999.

[BCC+97] L. Blackford, J. Choi, A. Cleary, E. E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henri, A. Petitet, K. Stanley, D. Walker, and R. Whaley. *ScaLAPACK Users' Guide*. SIAM, Philadelphia, 1997.

[BDHO11] G. Ballard, J. Demmel, O. Holtz, and Schawartz O. Minimizing communication in numerical linear algebra. *SIAM J. Matrix Anal. Appl.*, 32(3):866–901, 2011.

[Beb00] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.

[Beb08] M. Bebendorf. *Hierarchical Matrices*. Springer, Leipzig, Germany, 2008.

[BH03] A. Buffa and R. Hiptmair. Galerkin boundary element methods for electromagnetic scattering. *Topics in computational wave propagation*, 31:83–124, 2003.

[Bis91] C. Bischof. A parallel QR factorization algorithm with controlled local pivoting. *SIAM Journal on Scientific and Statistical Computing*, 12(1):36–57, 1991.

[BMD09] C. Boutsidis, M. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977, 2009.

[Bör10] S. Börm. *Efficient Numerical Methods for Non-local Operators: H-2 matrix Compression, Algorithms and Analysis*. European Mathematical Society, 2010.

[CD13] J. Chiu and L. Demanet. Sublinear randomized algorithms for skeleton decompositions. *SIAM Journal on Matrix Analysis and Applications*, 34(3):1361–1383, 2013.

[CDG18] X. Claeys, V. Dolean, and M. Gander. An introduction to multi-trace formulations and associated domain decomposition solvers. *Applied Numerical Mathematics*, 2018.

[CGMR05] H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin. On the compression of low rank matrices. *SIAM Journal on Scientific Computing*, 26(4):1389–1404, 2005.

[CH11] X. Claeys and R. Hiptmair. Multi-trace boundary integral formulation for acoustic scattering by composite structures. *Comm. Pure. Applied Math.*, 66:1163–1201, 2011.

[CHJH13] X. Claeys, R. Hiptmair, and C. Jerez-Hanckes. Multitrace boundary integral equations. In *Direct and inverse problems in wave propagation and applications*, volume 14 of *Radon Ser. Comput. Appl. Math.*, pages 51–100. De Gruyter, Berlin, 2013.

[CHJHP15] X. Claeys, R. Hiptmair, C. Jerez-Hanckes, and S. Pintarelli. Novel multi-trace boundary integral equations for transmission boundary value problems. In A. S. Fokas and B. Pelloni, editors, *Unified Transform for Boundary Value Problems: Applications and Advances*, chapter Novel Multi-Trace Boundary Integral Equations for Transmission Boundary Value Problems. SIAM, 2015.

[CK13] D. Colton and R. Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93 of *Applied Mathematical Sciences*. Springer, New York, 3rd edition, 2013.

[Cla16] X. Claeys. Essential spectrum of local multi-trace boundary integral operators. *IMA Journal of Applied Mathematics*, 81(6):961–983, 2016.

[ÇMI13] A. Çivril and M. Magdon-Ismail. Exponential inapproximability of selecting a maximum volume sub-matrix. *Algorithmica*, 65(1):159–176, 2013.

[Dav08] T. A. Davis. Algorithm 8xx: SuiteSparseQR, a multifrontal multithreaded sparse QR factorization package. *ACM Trans. Math. Software*, 2008.

[DB08] Z. Drmač and Z. Bujanović. On the failure of rank-revealing QR factorization software – a case study. *ACM Trans. Math. Softw.*, 35(2):12:1–12:28, 2008.

[DF08] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing (STOC)*, pages 537–546, 2008.

[DG16] V. Dolean and Martin J. Gander. Multitrace formulations and Dirichlet-Neumann algorithms. In *Domain decomposition methods in science and engineering XXII*, volume 104 of *Lect. Notes Comput. Sci. Eng.*, pages 147–155. Springer, Cham, 2016.

[DG17] J. Duersch and M. Gu. Randomized QR with column pivoting. *SIAM J. Sci. Comput.*, 39(4):C263–C291, 2017.

[DGGX15] J. Demmel, L. Grigori, M. Gu, and H. Xiang. Communication avoiding rank revealing QR factorization with column pivoting. *SIAM J. Matrix Anal. Appl.*, 2015.

[DGHL08] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou. Communication-optimal parallel and sequential QR and LU factorizations. *Technical Report No. UCB/EECS-2008-89*, 2008.

[DGHL12] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou. Communication-optimal parallel and sequential QR and LU factorizations. *SIAM J. Sci. Comput.*, 34(1):A206–A239, 2012.

[DR10] A. Deshpande and L. Rademacher. Efficient volume sampling for row/column subset selection. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 329–338. IEEE Computer Society, 2010.

[DRVG06] A. Deshpande, L. Rademacher, S. Vempala, and G.Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2:225–247, 2006.

[DV06] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation, approximation, randomization and combinatorial optimization, algorithms and techniques. *Lecture notes in Comp. Sci.*, 2:292–303, 2006.

[DV08a] Z. Drmač and K. Veselić. New fast and accurate Jacobi SVD algorithm. I. *SIAM J. Matrix Anal. Appl.*, 29(4):1322–1342, 2008.

[DV08b] Z. Drmač and K. Veselić. New fast and accurate Jacobi SVD algorithm. II. *SIAM J. Matrix Anal. Appl.*, 29(4):1343–1362, 2008.

[DY11] T. Davis and H. Yifan. The university of florida sparse matrix collection. *ACM Trans. Math. Software*, 1(162), 2011.

[Ede88] A. Edelman. Eigenvalues and condition numbers of random matrices. *SIAM J. Matrix Anal. Appl.*, 9(4):543–560, 1988.

[EG36] G. Eckart and Young G. The approximation of one matrix by another of lower rank. *Psychometrica*, 1:211–218, 1936.

[FD09] W. Fong and E. Darve. The black-box fast multipole method. *Journal of Computational Physics*, 228:8712–8725, 2009.

[FKV04] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004.

[GA01] A. Gray and Moore A. N-body problems in statistical learning. *Adv. Neural Inf. Process. Syst.*, pages 521–527, 2001.

[GCD18] L. Grigori, S. Cayrols, and J. Demmel. Low rank approximation of a sparse matrix based on lu factorization with column and row tournament pivoting. *SIAM J. Sci. Comput.*, 40(2):C181–C209, 2018.

[GE96] M. Gu and S. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Matrix Anal. Appl.*, 17(4):848–869, 1996.

[GKS76] G. Golub, V. Klema, and G. Stewart. Rank degeneracy and least squares problems. *Tech. Report TR-456, Department of Computer Science, University of Maryland, College Park*, MD, 1976.

[GOS+08] S. Goreinov, I. Oseledets, D. Savostyanov, E. Tyrtyshnikov, and N. Zamarashkin. How to find a good submatrix. *Research Report 08-10, ICM HKBU, Kowloon Tong, Hong Kong*, 2008.

[GR87] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.

[GR09] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Internat. J. Numer. Methods Engrg.*, 79(11):1309–1331, 2009.

[Gra13] L. Grasedyck. Adaptive recompression of H-matrices for BEM. *Computing*, 74:205–223, 2013.

[GT01] S. Goreinov and E. Tyrtyshnikov. The maximal-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, (280):47–51, 2001.

[GT11] S. Goreinov and E. Tyrtyshnikov. Quasioptimality of skeleton approximation of a matrix in the chebyshev norm. *Doklady Mathematics*, 83(3):374–375, 2011.

[Gu15] M. Gu. Subspace iteration randomization and singular value problems. *SIAM J. Sci. Comput.*, 37(3):A1139–A1173, 2015.

[GV18] N. Gillis and S. Vavasis. On the Complexity of Robust PCA and l_1-Norm Low-Rank Matrix Approximation. *Mathematics of Operations Research*, pages 1–13, 2018.

[GVL96] G. Golub and C. Van Loan. *Matrix Computations*. Jonhs Hopkins University Press, Baltimore, 3rd edition, 1996.

[GZT97] S. Goreinov, N. Zamarashkin, and E. Tyrtyshnikov. Pseudo-skeleton approximations by matrices of maximal volume. *Mathematical Notes*, 62(4):515–519, 1997.

[Hac15] W. Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*. Springer Series in Computational Mathematics, Baltimore, 3rd edition, 2015.

[Han] P. Hansen. *Regularization tools version 4.1 for MATLAB 7.3.* `http://www.imm.dtu.dk/~pcha/Regutools`. Accessed 10 Mar 2018.

[Hig02] N. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, 2nd edition, 2002.

[HJ91] R. Horn and C. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, New York, USA, 1991.

[HJH12] R. Hiptmair and C. Jerez-Hanckes. Multiple traces boundary integral formulation for Helmholtz transmission problems. *Advances in Computational Mathematics*, 37(1):39–91, 2012.

[HJHA16] F. Henríquez, C. Jerez-Hanckes, and F. Altermatt. Boundary integral formulation and semi-implicit scheme coupling for modeling cells under electrical stimulation. *Numerische Mathematik*, 136(1):101–145, 2016.

[HJHLP14] R. Hiptmair, C. Jerez-Hanckes, J.-F. Lee, and Z. Peng. Domain decomposition for boundary integral equations via local multi-trace formulations. In *Domain decomposition methods in science and engineering XXI*, volume 98 of *Lect. Notes Comput. Sci. Eng.*, pages 43–57. Springer, Cham, 2014.

[HJHM15] R. Hiptmair, C. Jerez-Hanckes, and S. Mao. Extension by zero in discrete trace spaces: inverse estimates. *Math. Comp.*, 84(296):2589–2615, 2015.

[HMT11] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[HT05] D. Huckaby and Chan T. Stewart's pivoted QLP decomposition for low-rank matrices. *Numer. Linear Algebra Appl.*, 12(4):153–159, 2005.

[JHPAT17] C. Jerez-Hanckes, C. Pérez-Arancibia, and C. Turc. Multitrace/singletrace formulations and Domain Decomposition Methods for the solution of Helmholtz transmission problems for bounded composite scatterers. *J. Comput. Phys.*, 350:343–360, 2017.

[JHPT15] C. Jerez-Hanckes, J. Pinto, and S. Tournier. Local multiple traces formulation for high-frequency scattering problems. *Journal of Computational and Applied Mathematics*, 289:306–321, 2015.

[JSC02] B. Jung, T. Sarkar, and Y. Chung. A survey of various frequency domain integral equations for the analysis of scattering from three-dimensional dielectric objects. *Progress In Electromagnetics Research, PIER*, 36:193–246, 2002.

[Kah66] W. Kahan. Numerical linear algebra. *Canadian Math. Bull*, (9):757–801, 1966.

[KG17] N. Kumar and Schneider G. Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra*, 65(11):2212–2244, 2017.

[KL98] S. Kapur and D. Long. N-body problems: IES3: Efficient electrostatic and electromagnetic simulation. *IEEE Computational Science and Engineering*, 5(4):60–67, 1998.

[Kou95] P. Koumoutsakos. Fast multipole methods for three-dimensional N-body problems. *Center for Turbulence Research, NASA Ames/Stanford Univ.*, pages 377–390, 1995.

[Leb72] N. Lebedev. *Special functions and their applications*. Dover Publications, Inc., New York, 1972.

[Lon77] A. Lonseth. Sources and applications of integral equations. *SIAM Review*, 19(2):241–278, 1977.

[Mar18] P. G. Martinsson. Randomized methods for matrix computations. *arXiv:1607.01649*, 2018.

[MB17] W. March and G. Biros. Far-field compression for fast kernel summation methods in high dimensions. *Applied and Computational Harmonic Analysis*, 43(1):39–75, 2017.

[MD09] M. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

[MGHV17] P. G. Martinsson, Quintana G., N. Heavner, and R. Van. Householder QR Factorization with Randomization for Column Pivoting (HQRRP). *SIAM Journal on Scientific Computing*, 39(2):C96–C115, 2017.

[Mir60] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math. Oxford Ser.*, 11(2):50–59, 1960.

[MR07] P. G. Martinsson and V. Rokhlin. An accelerated kernel-independent fast multipole method in one dimension. *SIAM J. Sci. Comput.*, 29(3):1160–1178, 2007.

[MRT06] P. G. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the approximation of matrices. *Technical Report Yale CS research report YALEU/DCS/RR-1361, Yale University, Computer Science Department*, 2006.

[MVH07] I. Markovsky and S. Van Huffel. Overview of total least-squares methods. *Signal Process.*, 87(10):2283–2302, 2007.

[Néd01] J.-C. Nédélec. *Acoustic and electromagnetic equations*, volume 144 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2001.

[OL04] N. A. Ozdemir and J.-F. Lee. A low-rank IE-QR algorithm for matrix compression in volume integral equations. *IEEE Transactions on Magnetics*, 40(2):1017–1020, 2004.

[OLBC10] F. Olver, D. Lozier, R. Boisvert, and C. Clark. *NIST handbook of mathematical functions*. U.S. Department of Commerce, National Institute of Standards and Technology, Washington, DC; Cambridge University Press, Cambridge, 1st edition, 2010.

[OVW16] S. O'Rourke, V. Vu, and K. Wang. Eigenvectors of random matrices: A survey. *J. Comb. Theory Ser. A*, 144:361–442, 2016.

[OZ18] A. Osinsky and N. Zamarashkin. Pseudo-skeleton approximations with better accuracy estimates. *Linear Algebra and its Applications*, 537:221–249, 2018.

[Pet89] T. Von Petersdorff. Boundary integral equations for mixed dirichlet, neumann and transmission problems. *Math. Met. App. Sc.*, 11:185–213, 1989.

[PT99] C.-T. Pan and P.T.P. Tang. Bounds on singular values revealed by QR factorizations. *BIT Numerical Mathematics*, 39(4):740–756, 1999.

[QSS06] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics (Texts in Applied Mathematics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Rja02] S. Rjasanow. Adaptive cross approximation of dense matrices. *International Association for Boundary Element Methods Conference, IABEM*, 2002.

[Rud08] M. Rudelson. Invertibility of random matrices: norm of the inverse. *Annals of Mathematics*, 168:575–600, 2008.

[SE03] P. Schneider and D. Eberly. *Geometric Tools for Computer Graphics*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[Som58] D. Sommerville. *An Introduction to the Geometry of n Dimensions*. Dover, New York, 1958.

[SS11] S. Sauter and C. Schwab. *Boundary element methods*, volume 39 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2011.

[Ste99] G. Stewart. The QLP approximation to the singular value decomposition. *SIAM J. Sci. Comput.*, 20(4):1336–1348, 1999.

[Ste08] O. Steinbach. *Numerical Approximation Methods for Elliptic Boundary Value Problems: Finite and Boundary Elements*. Springer, New York, USA, 2008.

[Sza91] S.J. Szarek. Condition numbers of random matrices. *J. Complexity*, 7:131–149, 1991.

[TV12] T. Tao and V. Vu. Random matrices: Universal properties of eigenvectors. *Random Matrices Theory Appl.*, 1(1):1150001, 2012.

[VGG14] F. Vico, L. Greengard, and Z. Gimbutas. Boundary integral equation analysis on the sphere. *Numer. Math.*, 128(3):463–487, 2014.

[VM17] S. Voronin and P. G. Martinsson. Efficient algorithms for cur and interpolative matrix decompositions. *Advances in Computational Mathematics*, 43(3):495–516, 2017.

[Waz11] A.-M. Wazwaz. *Applications of Integral Equations*, pages 569–595. Springer Berlin Heidelberg, 2011.

[WZ13] S. Wang and Z. Zhang. Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *J. Mach. Learn. Res.*, 14(1):2729–2769, 2013.

APPENDIX A

CALRQR: Communication avoiding low-rank QR approximation

In this supplemental chapter, we present a parallel algorithm called CALRQR which was developed during the first year of this thesis, it is based on QR approximations described in Chapter 4.

Our interest in communication avoiding algorithms is based on the fact that the performance of QR factorizations is highly impacted by the amount of communication performed during its execution, where communication refers to both data transferred between different levels of the memory hierarchy of a processor and data transferred between different processors of a parallel computer. In this context, our algorithm CALRQR can perform a low-rank approximation of a dense and sparse matrix with small communication cost compared to standard algorithms. The methodologies on which CALRQR is based are derived from the communication avoiding pivoted QR factorization CARRQR algorithm from [DGGX15].

We structure this short chapter into two sections, in the first section we present two algorithms that will allow to develop an optimal communication framework. Then, in the last section we present experimental results to evaluate the scalability and performance of our algorithm to produce a low-rank QR approximation. In particular, we compare CALRQR with a modified version of the `pdgeqp` routine from ScaLAPACK [BCC+97], which we refer to as `pdgekqp`, we made this adaptation since ScaLAPACK does not provide a truncated QR factorization. The experiments show good performance of CALRQR in terms of approximation error and scalability.

A.1 Communication avoiding algorithm low-rank QR

We build our algorithm CALRQR using a combination of two fundamental techniques, the first one is the algorithm TSQR which computes a communication avoiding QR factorization of a tall-and-skinny matrix. And the second one, is tournament pivoting, which enables communication between processors with small volume of communication and number of messages.

A.1.1 TSQR: Tall-Skinny QR factorization

In many applications we deal with matrices $A \in \mathbb{R}^{m\times n}$ with $m \gg n$, the Tall Skinny QR (TSQR) algorithm is very well suited for such case. It was introduced in [DGHL08] and its design allows to factorize these kind of matrices in a one-dimensional (1-D) block cyclic row layout, and is optimized in the sense that it attains lower bounds for the communication cost up to polylogarithmic factors.

Considering P processors, and assuming that $m/P \geq n$, Table A.1 shows the cost in arithmetic operations and volume of communication for TSQR and for its equivalent routine `pdgeqrf` from ScaLAPACK. Note that the communication cost of the `pdgeqrf` routine depends on the dimensions of the matrix, and then it becomes a more important drawback when dealing with large scale matrices.

Table A.1: Performance models of parallel TSQR and ScaLAPACK's parallel QR factorization PDGEQRF on a $m \times n$ matrix with P processors, along with lower bounds on the number of flops, words, and messages [DGHL08].

	TSQR	PDGEQRF	Lower bound
# *flops*	$\frac{2mn^2}{P} + \frac{2}{3}n^3 \log_2 P$	$\frac{2mn^2}{P} - \frac{2n^3}{3P}$	$\mathcal{O}(\frac{mn^2}{P})$
# *words*	$\frac{n^2}{2}\log_2 P$	$\frac{n^2}{2}\log_2 P$	$\frac{n^2}{2}\log_2 P$
# *messages*	$\log_2 P$	$2n\log_2 P$	$\log_2 P$

Remark A.1. The parallel TSQR algorithm is described in [DGHL08, Alg. 3], it requires a routine for computing local QR factorizations, for our implementation of TSQR we consider two cases,

- For **dense** matrices, we have implemented a truncated version of the classical truncated pivoted QR algorithm [GVL96, Alg. 5.4.1], by modifying the routine `dgeqp3` from LAPACK [ABB+99], which obtains a full QR matrix factorization. We refer to this modifications as `dgekqp3`.
- For **sparse** matrices, we use the routine `SuiteSparseQR` from SuiteSparse [Dav08] to compute local QR factorizations needed by TSQR.

For both cases, TSQR returns the Q and R factors and a permutation vector.

A.1.2 Tournament pivoting

Tournament pivoting is a pivoting technique that performs a reduction operation on blocks of columns of a matrix to identify b pivot columns. Recently, tournament pivoting was used to construct a communication avoiding rank revealing factorization, the theory and algorithm are very well explained in [DGHL08] and [DGGX15].

To illustrate how tournament pivoting works, let us consider an $m \times n$ matrix distributed block cyclically on a grid of $P = P_r \times P_c$ processors using blocks of size b. Two reduction trees are constructed, one for a local reduction (without communication) and the other for a global reduction (with communication between processors). We consider binary reduction trees (trees of types other than binary often result in better reduction performance, depending on the architecture, see *e.g.* [BDHO11]).

The local reduction tree has depth $\log_2(n/(bP_c))$, at each node a QR factorization is performed on blocks of size $m \times 2b$, using TSQR algorithm [DGHL08, Alg. 3]. Next a rank revealing QR factorization is performed on the small upper triangular factor R to obtain a permutation that gives the final b columns. The global reduction tree has depth $\log_2(P_c)$ and proceeds analogously as the local tree, but the $m \times 2b$ is formed by the b selected columns from 2 different processors, hence communication is required. Figure A.1 illustrates the reduction scheme for $b = 1$ and $P_c = 3$.

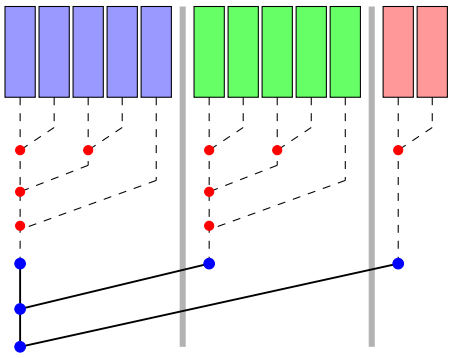


Figure A.1: Illustration of the tournament pivoting scheme on an m-by-10 matrix using 3 processors. The red and blue nodes correspond to reduction trees inside each processor and inter-processors respectively. There are only two inter-processors messages, this number of messages (two) is independent of the number of columns and it is obviously optimal.

The cost of a QR factorization with tournament pivoting is presented in table A.2. Note that these costs are reported for the case where the tournament pivoting algorithm from [DGHL08] is performed as an *all-reduction operation*, *i.e.* all processors participate at all the stages of the reduction and the final result is available on all processors.

Table A.2: Performance model of parallel all-reduction tournament pivoting to compute a full QR factorization [DGGX15].

# *flops*	$\frac{4mn^2-4n^3/3}{P} + \frac{8n^2b}{3P_c}(\log_2 P_r + 2) + \mathcal{O}(mn)$
# *words*	$\frac{n^2}{P_c}\log_2 P_r + \frac{mn-n^2/2}{P_r}(log_2 P_c + 1)$
# *messages*	$\frac{n^2}{2b^2P_c}\log_2 P_r + \frac{n}{b}\log_2 P_c(log_2 P_r + 1)$

Since we are interested in obtaining a truncated QR factorization, in order to use it to derive a low-rank approximation as explained in §4.3, we only need tournament pivoting to select b pivot columns and then the truncated QR factorization easily follows since we get the orthogonal basis of the selected columns, matrix Q of size $m \times b$, and then the rank-b QR approximation is given as

$$A \approx \xi_b := QQ^T A. \tag{A.1.1}$$

The following algorithm named QRTP obtains the selected columns, note that Q is know implicitly when computing tournament pivoting with TSQR, see [DGHL08, DGGX15] for details.

Data: A: the local matrix to factorize of size $m \times n$, b: the column panel size
Result: The QR factor and permutation vector of a selected group of $2b$ columns

$A_1 = A(:, b)$;
for $i = 1$ *to* n/b **do**
  $A_2 = A(:, i * b + 1 : (i + 1) * b)$;
  $[Q, R, p] = TSQR([A_1, A_2])$;
  $A_1 = [A_1, A_2](:, p)$;
  $A_1 = A_1(:, 1 : b)$;
end
while $myRank \in Workers$ **do**
  if $myRank \in Senders$ **then**
    Send A_1 ;
  else
    Receive A_2 ;
    $[Q, R, p] = TSQR([A_1, A_2])$;
    $A_1 = [A_1, A_2](:, p)$;
    $A_1 = A_1(:, 1 : b)$;
  end
  Update(myRank)
end
Return $[Q, R, p]$;

Algorithm 5: Tournament pivoting, QRTP

In Table A.3, we write the performance model of Algorithm 5. This will be useful for writing the performance model of CALRQR algorithm.

Table A.3: Performance model of parallel tournament pivoting performed as a reduction operation to select b pivot columns.

# *flops*	$\frac{8mnb}{P} + \frac{8mb^2}{P_r}(\log_2 P_c - 1) + \frac{16nb^2}{P_c}(\log_2 P_r - 1)$
# *words*	$\frac{2mb \log_2 P_c}{P_r} + \frac{2nb \log_2 P_r}{P_c}$
# *messages*	$\log_2 P_r \log_2 P_c$

A.1.3 CALRQR: Communication avoiding low-rank QR factorization

Using TSQR and tournament pivoting, we construct CALRQR to produce a low-rank QR factorization. Algorithm CALRQR is implemented considering that the matrix can be distributed block cyclically on a two-dimensional grid of $P = P_r \times P_c$ processors such that the local part of the matrix in a local processor fits in its fast memory. Then the b column candidates are selected by computing the first b steps of QRCP.

Algorithm 6 describes CALRQR. For the sake of simplicity, we assume that P_c, P_r, n and b are powers of two, CALRQR performs a local flat tree to select b candidate columns and a global binary tree to

obtain the final b selected columns.

Data: A: matrix of size $m \times n$ to be approximated, b: the rank of approximation
Result: Factors of a truncated rank-b QR approximation of A using tournament pivoting.

Π := I; Q := I ;
Distribute the matrix A block cyclically over $P = P_r \times P_c$ processors using blocks of size b ;
$\mathscr{P}_j$:= processors in the same column of processors led by processor P_j ;
A_j := panel that lies on processors of $\mathscr{P}_j$;
$N_b = n/P_c/b$;

/ Perform the local flat tree */*
$p = 1 : 2b$;
for $j = 2 \longrightarrow N_b$ **do**
  [Q,R,Π_{TP}] = QRTP($A_j(:, p)$,b) ;
  $p = [\Pi_{TP}, 2j + 1 : 2(j + 1)]$;
end
$A_w = A(:, \Pi_{TP}(1 : b))$;

/ Perform the global binary tree */*
for $k = 1 \longrightarrow \log(P_c)$ **do**
  Define a communication pattern splitting between sending and receiving processors ;
  if P_k *is a sender processor* **then**
    sends the local part of the selected columns of A_w to processors in the same row of the grid ;
  else
    A_{wr} := Received selected columns ;
    $A_b = [A_w, A_{wr}]$;
    [Q,R,p] = QRTP(A_b,b) ;
    $A_w = A_b(:, p)$;
  end
end

/ Getting QR factors */*
$Q = Q(:, 1 : k)$;
Update R_{12} considering only the non-pivot columns;
R_{11} = triu(A_w); */* upper triangular part of A_w */*
$R = [R_{11}, R_{12}]$;
Return [Q, R] ;
Verify $A \approx QR$;

Algorithm 6: Communication avoiding Low-Rank QR, CALRQR

Essentially, the cost of CALRQR is the sum of the cost of performing tournament pivoting plus the cost of computing the low-rank QR matrix in (A.1.1). In Table A.4 we detail these costs.

Table A.4: Performance models of the two versions of CALRQR on a rectangular $m \times n$ matrix with P processors, considering the rank of the matrix equal to b. "TP" stands for tournament pivoting.

	CALRQR2D = TP + TSQR
# *flops*	$\frac{12mnb}{P} + \frac{8mb^2}{P_r}(\log_2 P_c - 1) + \frac{16nb^2}{P_c}(\log_2 P_r - 1)$
# *words*	$\frac{2mb \log_2 P_c}{P_r} + \frac{4nb \log_2 P_r}{P_c}$
# *messages*	$\log_2 P_r \log_2 P_c$

A.2 Numerical Results

For the numerical tests, we have used the machine MESU from Sorbonne University in Paris. This machine runs on Linux and has 28 nodes, each node is equipped with a socket having 24 cores based on "Intel Xeon E5-2670", and each core has a frequency of 2.6GHz. We assign one MPI task per core.

We first analyze the approximation error, for which we use some matrices from Table 5.1, we construct these square matrices of size $n = 4096$. Figure A.2 shows the normalized error for a rank $k = 32$ approximation computed with CALRQR and PDGEKQP. We see that our accuracy is comparable with this state-of-the-art algorithm and in some cases we improve its accuracy.

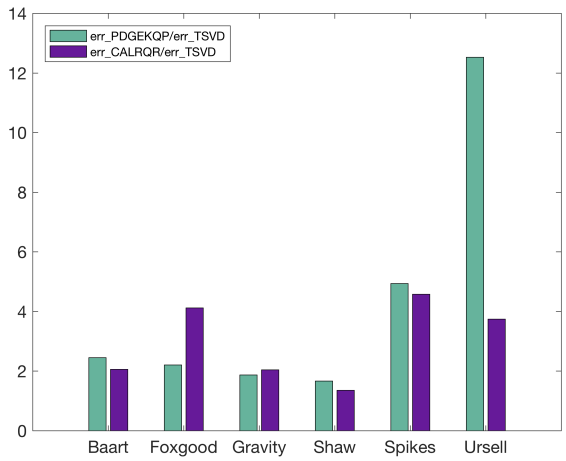


Figure A.2: Error of approximation for PDGEKQP and CALRQR normalized with respect to the truncated SVD error.

We test the scalability of our algorithms using a a set of square sparse matrices of size $n \times n$ taken from the University of Florida sparse matrix collection [DY11]. The results are shown in Figure A.3 for the set of matrices presented in the following table, where we show their number of rows (nrows), columns (ncols), and number of non-zero entries (nnz). Table A.5 also presents the run-time for obtaining a rank-256 QR truncated factorization using different number of MPI processes on the machine previously described.

Table A.5: Time, in seconds, to obtain a rank-256 QR truncated factorization of a set of large matrices taken from the University of Florida sparse matrix collection [DY11].

Matrix	**Dimensions**			**Number of MPI processes**				
	nrows	ncols	nnz	32	64	128	256	512
parabolic_fem	525825	525825	3674625	57.66	44.0	25.7	12.4	6.9
mac_econ_fwd500	206500	206500	1273389	94.0	55.1	28.2	13.1	7.2
atmosmodd	1270432	1270432	8814880	370.3	203.3	150.1	86.0	44.0
circuit5M_dc	3523317	3523317	19194193	916.0	465.9	245.4	143.1	80.7

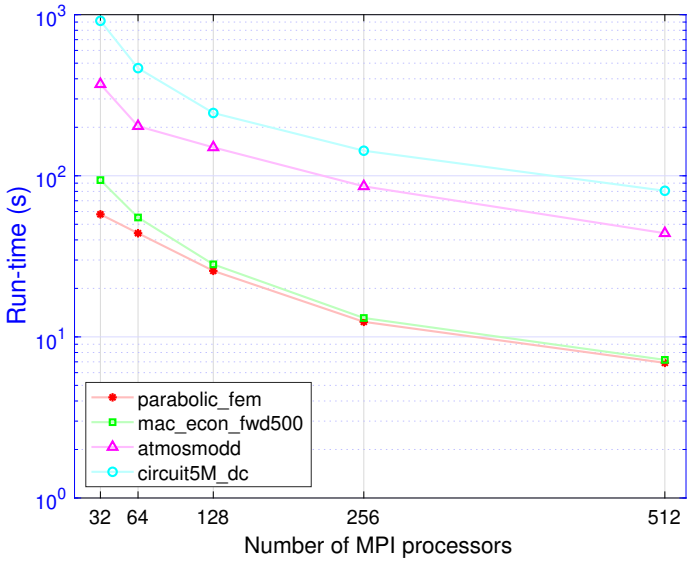


Figure A.3: Scalability of CALRQR algorithm for large matrices, runtime measured assigning one MPI task per core up to 512 cores.

APPENDIX B

Extra proofs and algorithms

B.1 Best Fitting Line Analysis

In this section we analyze the relationship of the best fitting line associated with the columns of a matrix with respect of the singular triplets of the matrix and its covariance matrix, This analysis is more general than the one made for the total least-square problem in [MVH07, Thm. 5]. Let $A \in \mathbb{R}^{m\times n}$, we use the notation $A := [a_j]$, where a_j is its j-th column. By considering the vectors a_j as points on the space $\mathbb{R}^m$, we are interested in the problem of finding the line that fits the best to all these points, we write this line as

$$\mathscr{L}_A(\tau) = w + \tau u, \quad \forall \tau \in \mathbb{R}, \tag{B.1.1}$$

where $w, u \in \mathbb{R}^m$ and u is unitary.

In order to find $\mathscr{L}_A$, let us write the n points as $a_j = w + \rho_j u + \delta_j u_{\perp j}$, where $\rho_j = u^T(a_j - v)$ and $u_{\perp j}$ is a unit vector perpendicular to u with an appropriate coefficient δ_j. Also define $y_j := a_j - w$ and its corresponding matrix $Y := [y_j] \in \mathbb{R}^{m\times n}$.

Next, we write the error as a functional, depending on w and u, which measures sum of the squared distances from a_j to $\mathscr{L}_A$, for all $j = 1, \cdots, n$. This is,

$$E(w, u) = \sum_{j=1}^{n} \delta_j^2 = \sum_{j=1}^{n} \|y_j - \rho_j u\|_2^2 = \sum_{j=1}^{n} y_j^T(I_m - uu^T)y_j. \tag{B.1.2}$$

Existence of the solution
Fist, to find u that minimizes E, let us rewrite (B.1.2) as

$$\mathrm{E}(w,u) = u^t \underbrace{\sum_{j=1}^{n} \left((y_j^t y_j)\mathrm{I}_m - y_j y_j^{\mathrm{T}} \right)}_{\mathrm{X}} u. \tag{B.1.3}$$

Then, it is clear that E attains its minimum when u corresponds to the eigenvector associated to the smallest eigenvalue of X or, equivalently, to the greatest eigenvalue of $\mathrm{C} := \sum_{j=1}^{n} y_j y_j^{\mathrm{T}} = \mathrm{Y}\mathrm{Y}^{\mathrm{T}} \in \mathbb{R}^{m\times m}$. Hence, the first singular vector of Y is a solution for u, this is

$$u = u_1(\mathrm{Y}). \tag{B.1.4}$$

Next, in order to find w, simply set the derivative of E with respect to w equal to zero, this is

$$\frac{\partial \mathrm{E}}{\partial w} = -2(\mathrm{I}_m - uu^{\mathrm{T}})(\sum_{j=1}^{n} y_j) = 0, \tag{B.1.5}$$

where the equality trivially holds when $\sum_{j=1}^{n} y_j = 0$, or equivalently when

$$w = \frac{1}{n}\sum_{j=1}^{n} a_j =: g, \tag{B.1.6}$$

where g is known as the gravity center of matrix A.

Uniqueness of the solution
Clearly the choice of w is not unique, since the pair $(w + \theta u, u)$, for all $\theta \in \mathbb{R}$, also defines the same line $\mathscr{L}_{\mathrm{A}}$ as the pair (w, u). Hence, we set $w = g$.

It is much more interesting to analyze if the solution for u is unique. For this case, we have that u is the eigenvector corresponding to the largest eigenvalue of $\mathrm{C} = \mathrm{Y}\mathrm{Y}^{\mathrm{T}}$, named λ_1. Then, $\mathrm{E}(w,u)$ attains a minimum if and only if $u = u_1(\mathrm{Y})$, provided λ_1 has algebraic multiplicity equal to 1, since its geometric multiplicity is also going to be 1 (see e.g. [QSS06, Sec.1]). Equivalently, the solution $u = u_1(\mathrm{Y})$ is unique provided $\sigma_1(\mathrm{Y}) \neq \sigma_2(\mathrm{Y})$.

B.2 Proof of Lemma 4.1

For ease of notation, let us consider $\mathrm{X} = \mathrm{R}_{11}$ and $\tilde{\mathrm{X}} = \mathrm{X}^{-1}$. Next, let us rewrite the truncated QR from (4.3.1) as

$$\mathrm{AP}_c = \begin{array}{c} \\ m \end{array} \begin{array}{c} \begin{array}{cc} k & m-k \end{array} \\ \begin{bmatrix} \mathrm{Q}_1 & \mathrm{Q}_2 \end{bmatrix} \end{array} \underbrace{\begin{array}{c} \begin{array}{cc} k & n-k \end{array} \\ \begin{array}{c} k \\ m-k \end{array} \begin{bmatrix} \mathrm{X} & \mathrm{R}_{12} \\ 0 & \mathrm{R}_{22} \end{bmatrix} \end{array}}_{=:\tilde{\mathrm{R}}}, \tag{B.2.1}$$

where P_c is the permutation matrix obtained from QRCP. Then, we use Gu and Eisenstat's technique [GE96, Thm. 3.2] as done for the proof of Theorem 6.2, we define

$$Z := \begin{bmatrix} \alpha X & \\ & R_{22} \end{bmatrix} = \begin{bmatrix} X & R_{12} \\ 0 & R_{22} \end{bmatrix} \underbrace{\begin{bmatrix} \alpha I_k & -X^{-1}R_{12} \\ & I_{n-k} \end{bmatrix}}_{=:W} \equiv \tilde{R}W, \tag{B.2.2}$$

where $\alpha = \sigma_{\max}(R_{22})/\sigma_{\min}(X) = \|R_{22}\|\|\tilde{X}\|_2$. Note that this choice of α ensures that $\sigma_{k+1}(Z) = \sigma_1(R_{22})$. Next, using Theorem 4.2 in (B.2.2) we get $\sigma_{k+1}(Z) \le \|W\|_2\sigma_{k+1}(\tilde{R})$. And since from (B.2.1) we have that the singular values of $\tilde{R}$ are equal to those of A, then

$$\|R_{22}\|_2 \le \|W\|_2\sigma_{k+1}(A). \tag{B.2.3}$$

Hence, it remains to bound $\|W\|_2$. Our proof goes as follows. We first show that $\|\tilde{X}\|_F \le \rho\tilde{f}(k)$, where $\rho := \frac{1}{|X(k,k)|}$ and $\tilde{f}$ is a function to be defined later on. Then, we shall show that $\|W\|_2^2 \le \tilde{f}(k+1)(n-k)$.

To bound $\|\tilde{X}\|_F$, we proceed to bound each of its entries. Next, we compare the i-th row of the equality $X\tilde{X} = I_k$, we get $|\tilde{X}(i,i)| = |\frac{1}{X(i,i)}| \le \rho$, where the right inequality holds by construction of QRCP. Also,

$$\sum_{l=i}^{h} X(i,l)\tilde{X}(l,h) = 0,$$

for $k \ge h > i$. By using the previous equality, replacing $h = i + j$, we get that

$$\tilde{X}(i,i+j) = \frac{1}{X(i,i)}\left(-\sum_{l=i+1}^{i+j} X(i,l)\tilde{X}(l,i+j)\right), \tag{B.2.4}$$

for $1 \le j \le k - i$.

Next, since $|X(i,i+j)| \le |X(i,i)|$ for all $j \ge 1$ (by construction of QRCP), we get

$$|\tilde{X}(i,i+j)| \le \sum_{l=i+1}^{i+j} |\tilde{X}(l,i+j)|. \tag{B.2.5}$$

Next, by recursively applying the previous inequality, we obtain

$$\begin{aligned} |\tilde{X}(i,i+1)| &\le 1\rho, \\ |\tilde{X}(i,i+2)| &\le 2\rho \\ |\tilde{X}(i,i+3)| &\le 2^2\rho \\ &\vdots \\ |\tilde{X}(i,i+j)| &\le 2^{j-1}\rho \\ &\vdots \\ |\tilde{X}(i,k)| &\le 2^{k-i-1}\rho, \end{aligned}$$

since the previous bounds hold for any $1 \le i \le k$, we set $i = 1$ and write

$$\|\tilde{X}\|_F^2 \le k|\tilde{X}(1,1)|^2 + (k-1)|\tilde{X}(1,2)|^2 + \cdots + 2|\tilde{X}(1,k-1)|^2 + |\tilde{X}(1,k)|, \tag{B.2.6}$$

$$\frac{1}{\rho}\|\tilde{X}\|_F^2 \le 2k - 1 + \sum_{j=2}^{k-1} 4^{j-1}(k-j) =: \tilde{f}(k). \tag{B.2.7}$$

As a second step, let c be the j–th column of matrix $Y := \tilde{X}R_{12}$, then $\forall i = 1, \cdots, k$, we get

$$|c(i)| \le \sum_{l=i}^{k} |\tilde{X}(i,l)||R_{12}(l,j)| = 1 + \sum_{h=1}^{k-i} 2^{h-1}\rho\,|R_{12}(i+h,j)|, \tag{B.2.8}$$

since $|R_{12}(i+h,j)| \le 1/\rho$ (by QRCP algorithm). Then,

$$|Y(i,j)| = |c(i)| \le 2^{k-i}, \quad \text{and} \quad \|Y(:,j)\|_F^2 = \sum_{i=1}^{k} 4^{k-i}, \tag{B.2.9}$$

this result coincides with a previous bound found in [GE96, Thm. 7.2].

Next, let us bound $\|W\|_2$, note that

$$\|W\|_2^2 \le 1 + \|Y\|_F^2 + \|\tilde{X}\|_F^2\|R_{22}\|_F^2 = 1 + \sum_{j=1}^{n-k} \left(\|Y(:,j)\|_2^2 + \rho\tilde{f}(k)\|R_{22}(:,j)\|_F^2\right). \tag{B.2.10}$$

Since QRCP also ensures that $\|R_{22}(:,j)\|_F \le 1/\rho$ and replacing (B.2.9) in (B.2.10), we get

$$\|W\|_2^2 \le (n-k)\tilde{f}(k+1), \tag{B.2.11}$$

Finally, replacing (B.2.11) in (B.2.3), we get the bound

$$\|R_{22}\|_2 \le \|W\|_2\sigma_{k+1}(A) = \sqrt{\tilde{f}(k+1)(n-k)}\,\sigma_{k+1}(A). \tag{B.2.12}$$

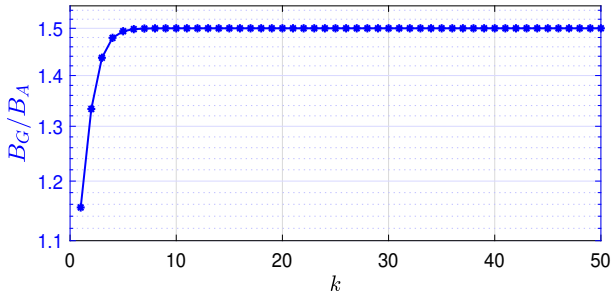


Figure B.1: Ratio of classical bound B_G for QRCP (see Table 4.1) to the new bound B_A from Lemma 4.1.

B.3 Algorithms

B.3.1 CUR via Geometric sampling

We present a MATLAB code for Algorithm 4.

```
%% CUR approximation with Gravity points criterion
% Requires:
% X,Y: Source and target points, given as matrices of size (mxd) and (nxd)
% respectively, where d is the geometric dimension
% k: fixed approximation rank.
% fun: kernel function, e.g. Laplacian kernel: fun = @(x,y) -1/(2*pi)*log(norm(x-y);
% Exponential kernel: fun = @(x,y) exp(1i*norm(x-y))/norm(x-y);
% Gravitation kernel:  fun = @(x,y) 1/(4*pi*norm(x-y));
% Returns:
% CUR: a rank-k approximation of matrix A(i,j)=fun(X(i,:),Y(j,:)).
% A \approx CxUxR, where C, R, U are complex matrices of size (mxk), (kxn), (kxk)

function [CUR] = CUR_GCS(fun,X,Y,k)

m = size(X,1);  n = size(Y,1);

% Finding t: number of sampling columns
l = nextpow2(k);
% t = pow2(l);
if (k > pow2(l-1) && k>2 )
    t = pow2(l+1);
else
    t = pow2(l);
end
if (k==1); t = 1; end

C=zeros(m,t);
R=zeros(k,n);

% Decompose target domain into t subdomains
[J] = GC_Sampling(Y,t);
% [J] = NN_Sampling(Y,X,t); % Alternatively use Nearest-Neighbors sampling

% Form matrix C of sampling columns, C is of size mxt
for i=1:size(X,1)
    for j=1:t
        C(i,j) = fun(X(i,:),Y(J(j),:));
    end
end

[Q,~,p_c]=qr(C,'vector');
Q=Q(:,1:k);
C=C(:,p_c(1:k));
```

```
44
45 % Get column indices
46 [~,~,p_r]=qr(Q.','vector');
47 I=p_r(1:k);
48
49 % Form Matrix R
50 for i=1:k
51     for j=1:size(Y,1)
52         R(i,j) = fun(X(I(i),:),Y(j,:));
53     end
54 end
55
56 % Construct the CUR rank-k approximation
57 G=C(I,:);
58 CUR=C*(G\R); % Use Algorithm 3 for computing this skeleton approximation in order to
       better handle and control the selected indices
59
60 return
```

B.3.2 Selecting columns using Gravity centers

The following algorithm presents a technique to decompose the target domain into t subdomains, in which we select a target point as the one closest to its gravity center, see Figure 6.2b. Partition is made by calling function *BinaryPartition*, which is an approach known as *geometrically balanced* clustering, *c.f.* [Bör10, Alg.2], [Beb08, Sec.1.4.1]. Such partition is generated by using a binary tree in which every non-leaf node, $\mathcal{T} := \{y_1, \cdots, y_h\} \subset Y$ with gravity center $g \in \mathbb{R}^3$, has two sons corresponding to disjoint sets of pints separated by the plane orthogonal to the line having direction given as the first left singular vector of matrix $T := [y_1, \cdots, y_h] - g \in \mathbb{R}^{3\times r}$, and intersecting it at g. The following algorithm, based on a binary tree structure, costs $O(n \log_2(t))$ floating point operations.

```
1  %% Select target points using geometrically balanced partition
2  % Require:
3  % Y: set of n target points, Y is an mxd matrix, d: geometric dimension
4  % t: number of subclusters to obtain from Y
5  % Returns:
6  % J: indices of target points closest to the gravity centers of the t subclusters
7
8  function [J] = GC_Sampling(Y,t)
9
10 % Sanity check
11 l = log(t)/log(2);
12 if (floor(l) ~= l)
13     error('t must be a power of 2!');
14 end
15 if (t==1); l=1; end
16
17 % Get 1st generation of sons
18 [G{1},S{1},GGC] = Geo_Bal_Partition(Y);
19 % GGC: index of target point closest to the gravity center of Y
```

```
20
21 % Sons of further generations
22 for i=2:l
23 S{i} = {};
24 G{i} = {};
25     for j = 1:size(S{i-1},2)    % number of clusters at previous generation
26             [g,s] = Geo_Bal_Partition(S{i-1}{j});
27             S{i} = cat(2,S{i},s);
28             G{i} = cat(2,G{i},g);
29     end
30 end
31
32 % Getting the indices of target points closest to gravity centers
33 for j=1:size(G{l},2)
34     for i=1:size(Y,1)
35     if(G{l}{j}'==Y(i,:))
36     J(j)=i;
37     end
38     end
39 end
40
41 if(t==1)
42 J=GGC;
43 end
44 end
45
46
47 %% Function Geo_Bal_Partition
48 % Performs geometrically ballanced partition to divide a cluster into two clusters son
49 % Requires:
50 % S_y: cluster of points
51 % Returns:
52 % Son: list of two cluster sons
53 % G: contains the gravity centers of cluster sons
54 % GCC: index of target point closest to the gravity center of S_y
55
56 function [G,Sons,GGC] = Geo_Bal_Partition(S_y)
57
58 [n,~] = size(S_y);
59 g = S_y'*ones(n,1)/n;
60 Cov = S_y - g';
61 [~,~,v] = svd(Cov); v=v(:,1);
62
63 L = Cov*v;
64 b_1 = (L>0);
65 b_2 = (L<0);
66 Sons{1} = S_y(b_1,:);
67 Sons{2} = S_y(b_2,:);
68
```

```
% Getting the index of target point closest to the gravity center of S_y
v = (S_y - g.'); z = zeros(n,1);
    for i=1:n
        z(i)=norm(v(i,:));
    end
[~,GGC]=min(z);

% Getting indices of target points closest to the gravity centers of clusters son
for j=1:2
n_s = size(Sons{j},1);  G{j} = Sons{j}'*ones(n_s,1)/n_s;
v = (Sons{j} - G{j}.');
z = zeros(n_s,1);
    for i=1:n_s
        z(i)=norm(v(i,:));
    end
[~,f]=min(z);
G{j}=Sons{j}(f,:)';
end
end
```

B.3.3 Selecting columns using Nearest-Neighbors approach

This approach consists in selecting t target points the closest to the set of source points, see Figure 6.2c. Then, indices corresponding to these points are the selected columns to be used to compute a CUR approximation and we call the resulting algorithm CUR_NNS as mentioned in section 6.3.1.

```
%% Select target points using Nearest-Neighbors
% Require:
% Y: set of n target points, Y is an mxd matrix, d: geometric dimension
% t: number of selected points Y
% Returns:
% J: indices of target points closest to the source domain
function [P] = NN_Sampling(Y,X,t)

% Finding the distance
    DX = bsxfun(@minus,Y(:,1),X(:,1)');
    DY = bsxfun(@minus,Y(:,2),X(:,2)');
    DZ = bsxfun(@minus,Y(:,3),X(:,3)');
    D = sqrt(DX.^2+DY.^2+DZ.^2); % The i-th line of D is the distance from
    % the i-th target point to X
    d = min(D(:));

for i=1:size(D,1)
    [dist(i),~] = min(D(i,:));
end

[~,P] = mink(dist,t); % Find t points on Y closest to X
end
```

Algorithm above costs $\mathcal{O}(mn)$ floating point operations. This non-linear complexity can be reduced by using efficient algorithms as the ones presented in [GA01]. In a recent work, March and Biros [MB17] showed that nearest-neighbors approach works well in practice for matrices created with kernels depending inversely on the distance of interaction points. However, they did not provide an explicit bound for the error, which we provide in Theorem 6.2. For higher dimension problems, nearest neighbors technique is still applicable by applying approximation techniques such as random trees [DF08].