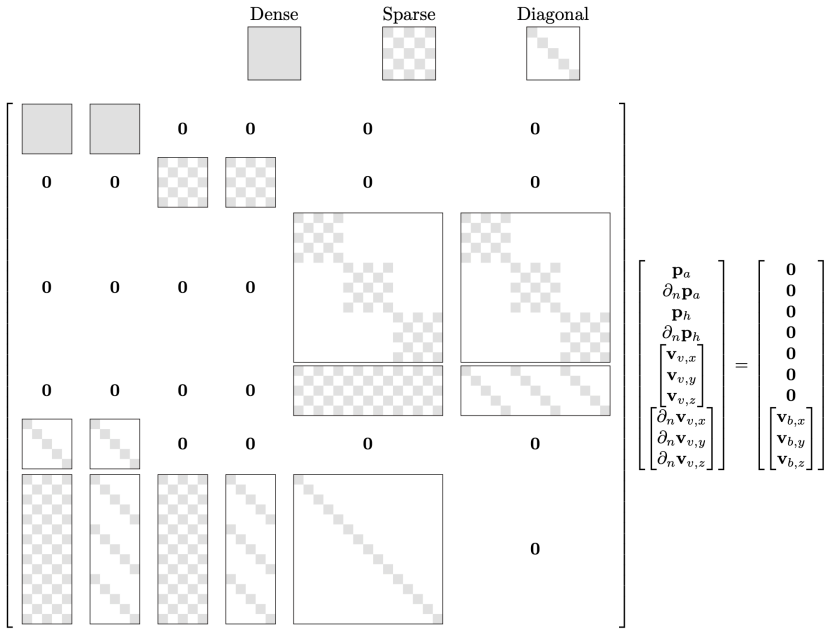
Computational Improvements in the Boundary Element Method for Acoustics including Viscothermal Dissipation

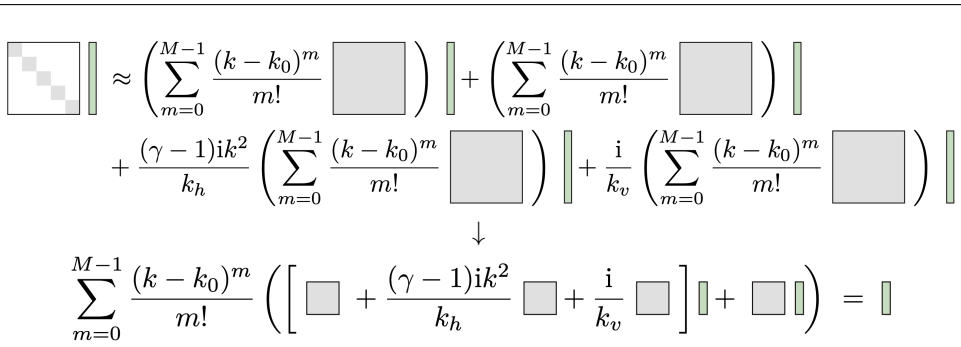
Schmitt, Mikkel Paltorp

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Ph.D. Thesis
Doctor of Philosophy





Computational Improvements in the Boundary Element Method for Acoustics including Viscothermal Dissipation

Mikkel Paltorp

Kgs. Lyngby 2023

DTU Electro
Department of Electrical and Photonics Engineering
Technical University of Denmark

Ørsteds Plads
Building 352
2800, Kgs. Lyngby
Denmark
electro.dtu.dk

Preface

This thesis is submitted to the Technical University of Denmark (DTU) as a partial fulfillment of the requirements for the degree of Doctor of Philosophy (Ph.D.). The work presented in this thesis was completed between September 1st, 2020 and August 31st, 2023 at the Acoustic Technology group, Department of Electrical and Photonics Engineering, DTU, under the supervision of Associate Professor Vicente Cutanda Henríquez and Associate Professor Niels Aage.

Kgs. Lyngby, August 31, 2023

Mikkel Paltorp

Resumé

En lang række problemstillinger indenfor akustisk ingeniørarbejde påkræver inkluderingen af både viskøs og termisk dissipation for nøjagtigt at fange den virkelige fysik. Disse dissipative effekter bliver særligt vigtige, når man har at gøre med mindre geometriske dimensioner i det akustiske domæne, som det ses i anvendelser som akustiske transducere og høreapparater. Beregningsteknikken kendt som randelementmetoden gør det muligt at tage højde for dissipation og samtidig undgå behovet for randlagsmaskering. Dette er i modsætning til den meget anvendte finite element metode, hvor randlagsmaskering er et krav. Derudover er randelementmetoden velegnet til modellering af åbne domæner, hvilket ofte er af interesse i akustisk modellering.

Den nuværende formulering af randelementmetoden for viskøse og termiske tab benyttet i litteraturen, har to væsentlige begrænsninger. Den første begrænsning er at den er baseret på frekvensafhængige randintegraler, hvilket gør formuleringen uegnet til scenarier, der involverer flere frekvenser. Den anden begrænsning er, at formuleringen afhænger af både sparse og tætte matricer, som hver især kommer med sine udfordrigner. Selvom sparsitet oftest er forbundet med behageligheder i numerisk analyse, kan den nuværende formulering ikke udnytte disse egenskaber fuldt ud. Faktisk omfatter løsningsstadiet en række sparse matrix-matrix-produkter, som ikke kan garanteres at være sparse, hvilket igen ødelægger den beregningsfordel, der normalt er forbundet med sparsitet. De tætte matricer er et andet dyr og ville øjeblikkeligt gøre formuleringen ubrugelig til bereningsmæssige store problemer. Men i forbindelse med randelementmatricer findes der måder at tilnærme matrik-vektor-produkterne på en måde, der bereningsmæssigt let kan skaleres til store problemer. Indsigten er her derfor at formulere problemet på en måde, hvortil kun matrix-vektor-produktet kræves, således at tilnærmelserne kan benyttes.

Løsninger på de to begrænsninger er allerede udviklet i for rene akustiske problemer. Denne afhandling udvider løsningerne til også at inkludere tilfælde der inkluderer viskøse og termiske tab. Multifrekvensproblemet løses ved at udvide en kendt reduceret ordens serieekspansionsrandelementmetode til at inkludere randlagstimpedansrandbetingelsen. Denne udvikling krævede afledning af Taylor-udvidelsen af den tangentielle gradient af Green's funktionen. Den resulterende model er mere kompleks end den originale på grund af de frekvensafhængige koefficienter foran integralerne, der stammer fra randlagstimpedansrandbetingelsen. Løsningen i forbindelse med simuleringer i stor skala er en to-trins proces. Det første trin er en omfattende omformulering af den underliggende model, der gør det muligt at udnytte sparsitetsmønstrene i de såkaldte termiske og viskøse matricer i løsningsfasen af problemet. Dette var et nødvendigt første skridt i retning af simulering i stor skala, da det uden dette ikke ville have givet mening at fortsætte med denne formulering. Det sidste trin var så at omformulere problemet på en måde, hvortil kun det tætte matrix-vektor-produkt med de såkaldte akustiske matrices var en nødvendighed. Givet denne omformulering var det så muligt at tilnærme matrix-vektor-produktet ved hjælp af standardteknikker såsom hurtigmultipolmetoden eller $\mathcal{H}$-matricer. Ved at benytte disse teknikker var det muligt at løse problemer af størrelser langt ud over, hvad der tidligere har været muligt.

Keywords: randelementmetoden; hurtigmultipolmetoden; $\mathcal{H}$-matricer; rank-struktur; viskotermiske effekter; Kirchhoff dekomposition; randlagsimpedans; reduceret ordens model.

Abstract

A long range of problems in acoustical engineering necessitate the incorporation of both viscous and thermal dissipation in order accurately capture the real-world physics. These dissipative effects become particularly important when dealing with smaller geometrical dimensions in the acoustic domain, as seen in applications like acoustic transducers and hearing aids. The computational technique known as the boundary element method offers the ability to account for dissipation while avoiding the need for boundary layer meshing. This is opposed to the widely used finite element method for which boundary layer meshing is a must. In addition, the boundary element method is suitable for modeling unbounded domains, which is often of interest in acoustical modeling.

However, the current formulation of the boundary element method including viscous and thermal losses, has two notable drawbacks. The first major limitation is the reliance on frequency-dependent boundary integrals, which makes the formulation unsuitable for scenarios involving multiple frequencies. The second major limitation is that the formulation depends on both sparse and dense matrices, each of which comes with its own problems. Although sparsity is most often associated with pleasantries in numerical analysis, the current formulation cannot utilize its properties to the fullest. In fact, the solution stage includes a series of sparse matrix-matrix products which cannot be guaranteed to be sparse, which in turn ruins the computational advantage that is usually contributed to sparsity. The dense matrices is a different beast and would instantly render the formulation unusable for large-scale problems. However, in the context of boundary element matrices, there exist ways to approximate the matrix-vector products of these types of matrices in a way that scales. The key is to then reformulate the problem in a way for which only the matrix-vector product is required.

Solutions to the two issues at hand have been developed within the context of pure acoustical problems. This thesis extends the solutions in the context of viscous and thermal losses. In particular the multifrequency problem is resolved by extending a known reduced order series expansion boundary element method to include the boundary layer impedance boundary condition. This development required the derivation of the Taylor expansion of the tangential derivative of the Green's function. The resulting model is more complex than the original due to the frequency-dependent coefficients in front of the integrals stemming from the boundary layer impedance condition. The solution in the context of large-scale simulations is a two-step process. The first step is an extensive reformulation of the underlying model that allows one to utilize the sparsity patterns of the so-called thermal and viscous modes in the solution phase. This was a necessary first step in the direction of large-scale simulation, as without this it would have not made sense to continue with this formulation. The final step was then to reformulate the problem in a way for which only the dense matrix-vector product with the so-called acoustical mode was required. Given this reformulation, it was then possible to approximate the matrix-vector product using standard techniques such as the fast multipole method or $\mathcal{H}$-matrices. Using these, it was possible to solve problems of sizes far beyond what was previously possible.

Keywords: boundary element method; fast multipole method; $\mathcal{H}$-matrices; rank-structure; viscous thermal effects; Kirchhoff Decomposition; boundary layer impedance; reduced order model.

Acknowledgements

I want to acknowledge the many great discussions and incredible feedback that I received from my supervisors Vicente and Niels. Additionally, I would like to thank you for giving me your trust and the freedom to explore the things that I found important and interesting - I do not think I would have survied the full PhD without it. I would also like to acknowledge Peter, who for the first two years of my journey acted as a great unofficial supervisor while being an overall amazing desk neighbor. Our discussions on numerical methods, viscothermal losses, and everyday life have made this project possible and joyful.

In addition, I would like to thank my colleagues in the Acoustic Technology group (ACT) as well as the Centre for Acoustic-Mechanical Microsystems (CAMM). Every PhD student, professor, and affiliates has been a pleasure to work with. In particular, I would like to mention the CAMM PhDs Jonathan and Hossein for injecting some fun into my Wednesdays and my ACT colleagues Yorgos, Birgitte, Yauheni, Pierangelo, Zenong, Henrik, Samuel, Sophia, and Antonio for never making me eat lunch alone. Also, I would like to thank the people whom I met mainly virtually, Diana, Daniel P, Daniel G, and Javier, for making my first years enjoyable. Lastly, a big shout-out needs to be given to Nikolas, Xenofon, Franz, and Manuel. Nikolas for our many great talks and the shared long evenings towards the end of our PhDs. Xenofon and Franz for our travels and bikepacking in South Korea after the ICA2022 conference - this was a highlight of a PhD study that was greatly impacted by COVID19 travel restrictions. Franz and Manuel for our many shenanigans including, but not limited to finding the best falafel in Copenhagen and bouldering sessions before, after, but never during work. I am grateful to all of you.

I can not finish this without mentioning the many great people that I met during my visits to the Chair of Vibro-Acoustics of Vehicles and Machines at the Technical University of Munich. Professor Marburg has here assembled an inspiring group of young researchers which I am grateful to have been able to get to know during my stays in Munich. In particular, I would like to mention Simone for her overwhelming hospitality during my visits to Munich and our trip to the Boundary Element summer school in the Alps. I hope you enjoyed your visit to Denmark as much as I did my visits to Munich. Lastly, I would like to stress how important it was to me to have you be part of my project. Having someone with whom to share my frustrations, progress, and ideas was a major reason for the various successes during the project. For this I am grateful.

Finally, but perhaps most importantly, I would like to thank my friends (Wadklubben, MIL-crewet, Mønisterne, Gadevængerne, MatTekkerne, Oskar, Jens, Emma, Krabbe, my brother, and many more) and family (my mom, my dad, my brother, my grandparents, my cousins, my uncle, my aunts, and many more) for continually reminding me that the most important things in life is what happens outside of work - even if science can be really really interesting sometimes.

Publications

The thesis is based on a collection of scientific works in various forms: Paper J1 is an accepted journal paper; Paper J2 is a submitted journal paper; Paper J3 is a manuscript in preparation; Paper C1 and Paper C2 are conference contributions. The list of papers is as follows:

- Paper J1: **M. Paltorp**, V. C. Henríquez, N. Aage, P.R. Andersen. "A Reduced Order Series Expansion for BEM Incorporating the Boundary Layer Impedance Condition.", *Accepted to the Journal of Theoretical and Computational Acoustics* (2023), doi.org/10.1142/S2591728523500123.
- Paper J2: S. Preuss, **M. Paltorp**, A. Blanc, V. C. Henríquez, S. Marburg. "Revising the Boundary Element Method for Thermoviscous Acoustics: An Iterative Approach via Schur Complement." *Accepted to the Journal of Theoretical and Computational Acoustics* (2023).
- Paper J3: **M. Paltorp**, S. Preuss, V. C. Henríquez, S. Marburg. "Large-Scale Boundary Element Computations Including Viscous and Thermal Losses." *Manuscript* (2023).
- Paper C1: **M. Paltorp**, V. C. Henríquez, N. Aage, P. R. Andersen. "A reduced order model including viscothermal losses." In: *Proceedings of 24th International Congress on Acoustics, ICA 2022.* Gyeongju, Republic of Korea (2022).
- Paper C2: **M. Paltorp**, V. C. Henríquez. "An Open-source Boundary Element Framework for Large-scale Viscothermal Acoustics." In: *Proceedings of the 10th convention of the European Acoustic Association, Forum Acusticum 2023.* Turin, Italy (2023).

List of symbols and abbreviations

Abbreviations

$\mathcal{H}$-matrix	Hierarchical Matrix
2D	Two-dimensional
3D	Three-dimensional
BE	Boundary Element
BEM	Boundary Element Method
BLI	Boundary Layer Impedance
DOF	Degrees of Freedom
FEM	Finite Element Method
FLNS	Fully Linearized Navier-Stokes
FMM	Fast Multipole Method
GMRes	Generalized Minimal Residual Method
HPC	High-Performance Computing
IFD	Interpolation Function Derivative
KD	Kirchhoff Decomposition
MOR	Model Order Reduction
PDE	Partial Differential Equation
ROM	Reduced Order Model
ROSEBEM	Reduced Order Series Expansion Boundary Element Method
SEBEM	Series Expansion Boundary Element Method
SVD	Singular Value Decomposition

Symbols (Bold)

$\mathbf{D}_c \in \mathbb{R}^{nd\times n}$	Column collection of interpolation function derivatives, see (4.39)
$\mathbf{D}_r \in \mathbb{R}^{n\times nd}$	Row collection of interpolation function derivatives, see (4.39)
$\mathbf{D}_s \in \mathbb{R}^{n\times n}$	Second tangential interpolation function derivatives
$\mathbf{D}_t \in \mathbb{R}^{n\times n}$	First tangential interpolation function derivatives
$\mathbf{D}_x \in \mathbb{R}^{n\times n}$	x-component of interpolation function derivatives, see (4.21)
$\mathbf{D}_y \in \mathbb{R}^{n\times n}$	y-component of interpolation function derivatives, see (4.21)

$\mathbf{D}_z \in \mathbb{R}^{n\times n}$	z-component of interpolation function derivatives, see (4.21)
$\mathbf{F}_m \in \mathbb{C}^{n\times n}$	Discretized surface integral of the m^{th} derivative of normal Green's function
$\mathbf{F}_{\ell m} \in \mathbb{C}^{\ell\times \ell}$	Projected Discretized surface integral of the m^{th} derivative of Normal Green's function
$\mathbf{G} \in \mathbb{C}^{n\times n}$	Discretized surface integral of the Single Layer Potential
$\mathbf{G}_m \in \mathbb{C}^{n\times n}$	Discretized surface integral of the m^{th} derivative of Green's function
$\mathbf{G}_{\ell m} \in \mathbb{C}^{\ell\times \ell}$	Projected Discretized surface integral of the m^{th} derivative of Green's function
$\mathbf{H} \in \mathbb{C}^{n\times n}$	Discretized surface integral of the Double Layer Potential
$\mathbf{H}_m \in \mathbb{C}^{n\times n}$	Discretized surface integral of the m^{th} derivative of Green's function
$\mathbf{H}_{\ell m} \in \mathbb{C}^{\ell\times \ell}$	Projected Discretized surface integral of the m^{th} derivative of Green's function
$\mathbf{I} \in \mathbb{R}^{n\times n}$	The Identity matrix
$\mathbf{n} \in \mathbb{R}^{d}$	Normal vector
$\mathbf{N} \in \mathbb{R}^{nd\times n}$	Projection onto normal direction, see (4.39)
$\mathbf{n}(\mathbf{y}) \in \mathbb{R}^{d}$	Normal vector at position $\mathbf{y}$
$\mathbf{p} \in \mathbb{C}^{n}$	Pressures
$\mathbf{p}_a \in \mathbb{C}^{n}$	Acoustical pressures
$\mathbf{p}_h \in \mathbb{C}^{n}$	Thermal pressures
$\mathbf{p}_{\text{inc}} \in \mathbb{C}^{n}$	Incident pressures
$\mathbf{p}_{\text{sca}} \in \mathbb{C}^{n}$	Scattered pressures
$\mathbf{R} \in \mathbb{C}^{n\times nd}$	Inner Matrix, see (4.47)
$\mathbf{t} \in \mathbb{R}^{d}$	Target Location
$\mathbf{T}_m \in \mathbb{C}^{n\times n}$	Discretized surface integral of the m^{th} derivative of Tangential Green's function
$\mathbf{T}_{\ell m} \in \mathbb{C}^{\ell\times \ell}$	Projected Discretized surface integral of the m^{th} derivative of Tangential Green's function
$\mathbf{u} \in \mathbb{R}^{d-1}$	Local coordinates
$\mathbf{U}_\ell \in \mathbb{C}^{n\times \ell}$	Projection matrix / Reduced Order Basis
$\mathbf{v} \in \mathbb{R}^{nd}$	(Stacked) Viscous velocity, see (4.7)
$\mathbf{v}_s \in \mathbb{R}^{nd}$	(Stacked) Boundary velocity, see (4.14)
$\mathbf{x} \in \mathbb{R}^{d}$	Target Location
$\mathbf{y} \in \mathbb{R}^{d}$	Source Location
$\mathbf{L}^e \in \mathbb{R}^{n^e\times n}$	Element localization matrix, see (3.10)
$\partial_{\mathbf{n}}\mathbf{p} \in \mathbb{C}^{n}$	Normal derivative of the pressures
$\partial_{\mathbf{n}}\mathbf{p}_a \in \mathbb{C}^{n}$	Normal derivative of the acoustical pressures
$\partial_{\mathbf{n}}\mathbf{p}_h \in \mathbb{C}^{n}$	Normal derivative of the thermal pressures

$\partial_{\mathbf{n}}\mathbf{v} \in \mathbb{R}^{dn}$	(Stacked) Normal derivatives of viscous velocity, see (4.7)

Super- and subscripts

H	Hermitian/Conjugate Transpose
$\parallel$	Tangential direction
$\perp$	Normal direction
$\top$	Transpose
a	Indicates the acoustic mode, see (4.1)
e	Element number
h	Indicates the thermal mode, see (4.2)
v	Indicates the viscous mode, see (4.3)

Functions

$\mathrm{blkdiag}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n)$	Block-diagonal matrix with $\mathbf{A}_i$ in its diagonal (and the rest being zero)
$\mathbf{T}(\mathbf{x})$	Global basis functions (row vector, see (3.1))
$\mathbf{T}^e(\mathbf{x})$	Element basis functions
$\mathbf{v}_v(\mathbf{x})$	Viscous velocity
$\delta(\mathbf{x})$	Dirac delta function
$\delta(\mathbf{x}, \mathbf{y})$	Dirac delta function
$\mathrm{diag}(\mathbf{x})$	Square matrix with $\mathbf{x}$ as its diagonal
$\|\mathbf{x}\|_2$	Euclidean norm of $\mathbf{x}$
$G(\mathbf{x}, \mathbf{y})$	Green's function
$p(\mathbf{x})$	Pressure
$p_a(\mathbf{x})$	Acoustic pressure
$p_h(\mathbf{x})$	Thermal pressure
$p_{\mathrm{inc}}(\mathbf{x})$	Incident pressure
$p_{\mathrm{sca}}(\mathbf{x})$	Scattered pressure

Operators

Δ	Laplacian
$\Delta^{\parallel}$	Tangential (part of) Laplacian
$\frac{\partial}{\partial \mathbf{n}}$	Normal derivative ($\mathbf{n}^\top \nabla$)
∇	Gradient
$\nabla^{\parallel}$	Tangential (part of) Gradient ($\nabla^{\parallel} = (\mathbf{I} - \mathbf{n}\mathbf{n}^\top)\nabla$)
$\nabla^{\perp}$	Normal (part of) Gradient

$\text{oproj}_{\mathbf{n}}(\nabla p_a)$	Vector rejection of p_a from $\mathbf{n}$
$\text{proj}_{\mathbf{n}}(\nabla p_a)$	Vector projection of p_a onto $\mathbf{n}$

Symbols

α	Sign of time dependency
δ_{ij}	Kronecker delta
η	Viscosity of air
Γ	Boundary of PDE domain
γ	Ratio of specific heats
i	Imaginary unit
λ	Thermal conductivity
μ_a	Acoustic Lossy Constant, see (4.42)
μ_h	Thermal Lossy Constant, see (4.42)
Ω	Domain of PDE
ω	Angular frequency of the pressure
ϕ_a	Acoustic No-slip Constant, see (4.11)
ϕ_h	Thermal No-slip Constant, see (4.11)
ρ_0	Equilibrium density
τ_a	Acoustic Isothermal Constant, see (4.9)
τ_h	Thermal Isothermal Constant, see (4.9)
c_0	Equilibrium speed of sound
C_p	Specific heat capacity at constant pressure
C_v	Specific heat capacity at constant volume
d	Dimension (usually we deal with 2- or 3-dimensional computations).
f	Frequency
f_0	Expansion frequency
k	Wavenumber
k_0	Expansion wavenumber
k_a	Acoustic wavenumber, see (4.1)
k_h	Thermal wavenumber, see (4.2)
k_v	Viscous wavenumber, see (4.3)
n	Number of collocation nodes
n^e	Number basis functions on element e
r	Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Motivation

An increasingly important application of acoustics is that of micromechanical systems such as hearing aids and mobile phones. In these devices, the acoustics is propagating in narrow channels and chambers on the millimeter scale. As such, it can be important to include physical phenomena occurring on the microscale, such as e.g. viscous and thermal effects. If these effects are not included in the simulation, the results may be incorrect [28, 36, 50, 63].

1.2 Scope of Thesis

The aim of this PhD study is to develop new methods to efficiently simulate large-scale acoustical problems, including viscous and thermal losses, using the boundary element method (BEM). Given the computational complexity of large-scale problems, thoughts had to be put into the underlying language of implementation. Due to the relatively short timeframe of the project, it was chosen to use the Julia programming language as the compromise between efficiency and ease-of-use was found fitting. The final code is made available through GitHub and an overview of which acoustical problems it can be used to solve is presented in Paper C2. When taking viscothermal simulations large-scale two issues arise. The first issue is that the naive discretization process will result in fully populated matrices, which for even moderately sized problems becomes a memory bottleneck. The second issue is that the frequency dependence of the underlying boundary element integrals makes it so that for every frequency of interest the discretization process needs to be redone. We note that the second issue is an inherent problem of the boundary element method and is therefore also present for computations without viscous and thermal losses. The two approaches, aiming to resolve the two issues, were studied and can be summarized as follows:

- The first approach is based on a new formulation for the inclusion of viscous and thermal losses in boundary element computations. The new model eliminates the reliance of tangential directions, making it easier to implement. More importantly, we show that the new formulation makes it possible to utilize the sparsity in the solution phase of the equations, whereas previously it was only possible to utilize the sparsity in the assembly phase (Paper J2). As a result, the limiting factor of the computation became the two dense acoustical matrices. Fortunately, standard techniques for accelerating the boundary element computations, e.g. the fast multipole method and the $\mathcal{H}$-matrix approach, can be used to speed up this part of the computation. As such, we show that it is possible to solve problems of sizes that far exceed what has previously been possible (Paper J3). The underlying simulation code, written in the Julia programming language, was made available as an open-source package (Paper C2).

- The second approach tackles the computational issues of handling multiple frequencies using the boundary element method. The main idea is rather simple: We combine a known technique for approximating viscothermal losses with a known technique for reducing computational efforts for multifrequency boundary element problems. Doing so, however, is not trivial and requires additional derivations of e.g. the Taylor expansion of the tangential gradient of the Green's function. The resulting model can be used to efficiently solve multifrequency boundary element problems including (approximate) losses. The results show that the introduced model is around 25-100 times faster while using a similar memory footprint (Paper J1 and Paper C1).

1.3 Thesis Structure

The thesis is organized as follows. Chapters 2 and 3 introduce the basic theory of sound propagation and the boundary element method. Chapter 4 deals with the basics of viscous and thermal losses and gives a description of the new formulation. Similarly, Chapter 5 gives a short introduction to the proposed model capable of improving the computational efficiency of multifrequency analysis including viscous and thermal losses. Chapter 6 then draws conclusions from the PhD study. Finally, in Chapter 7, the potential next steps of the research are presented and discussed.

CHAPTER 2

Sound Propagation

2.1 Acoustic Waves

In acoustics, we most often assume that waves are perturbations of the medium density ($\rho(\mathbf{x}, t)$), pressure ($p(\mathbf{x}, t)$), and velocity ($\mathbf{v}(\mathbf{x}, t)$), where t is the time [34]. Furthermore, it is often assumed that the fluid is inviscid and that the perturbations are small, that is, $\rho' \ll \rho_0$, $p' \ll p_0$, and $|\mathbf{v}'| \ll c$ (with c being the wave speed), with the primes denoting the perturbations, and the subscript zero denoting the values when the fluid is at rest. Finally, the density and pressure can be computed as

$$\rho = \rho_0 + \rho', \quad p = p_0 + p'. \tag{2.1}$$

In this case, the linearized mass and momentum conservation equations can be written as

$$\frac{\partial \rho'}{\partial t} + \nabla \cdot (\rho_0 \mathbf{v}') = 0, \quad \rho_0 \frac{\partial \mathbf{v}'}{\partial t} + \nabla p' = 0. \tag{2.2}$$

Differentiating the first equation with respect to t and then utilizing the second equation, it follows that

$$\frac{\partial^2 \rho'}{\partial t^2} - \Delta p' = 0., \tag{2.3}$$

where Δ is the Laplacian. Now the set of equations in (2.2) is not closed since there are only two equations, but more unknowns (density, pressure, and the components of the velocity). The relation needed to close the system of equations is the equation of state, which relates perturbations of the pressure and density. The simplest of such relations is that the pressure depends solely on the density as [40]

$$p = p(\rho). \tag{2.4}$$

Approximating the above using a Taylor expansion, we get the following approximation of the density

$$p = p(\rho_0) + \left.\frac{\mathrm{d}p}{\mathrm{d}\rho}\right|_{\rho=\rho_0} (\rho - \rho_0) + O((\rho - \rho_0)^2). \tag{2.5}$$

Furthermore, using $p(\rho_0) = p_0$, $p = p_0 + p'$, and $\rho' = \rho - \rho_0$ and disregarding the higher-order terms, the above becomes

$$p' \approx c^2 \rho', \quad c^2 = \left.\frac{\mathrm{d}p}{\mathrm{d}\rho}\right|_{\rho=\rho_0}. \tag{2.6}$$

Inserting this back into (2.3) we arrive at the wave equation for pressure perturbations.

$$\frac{1}{c^2}\frac{\partial^2 p'}{\partial t^2} - \Delta p' = 0. \tag{2.7}$$

The relation in (2.6) naturally gives a similar wave equation for the density perturbations. Equivalently, the velocity perturbations satisfy the vector wave equation

$$\frac{1}{c^2}\frac{\partial^2 \mathbf{v}'}{\partial t^2} - \Delta \mathbf{v}' = \mathbf{0}, \tag{2.8}$$

meaning that each of its components satisfy their own scalar wave equation. However, these equations are not independent since the momentum equation shows that there exists some scalar potential ψ', called the velocity potential, such that [35]

$$\mathbf{v}' = \nabla \psi', \quad \frac{1}{c^2}\frac{\partial^2 \psi'}{\partial t^2} - \Delta \psi' = 0, \quad \rho_0 \frac{\partial \psi'}{\partial t} = -p'. \tag{2.9}$$

2.2 Monochromatic waves

For time-harmonic problems the solution can be written as

$$p(\mathbf{x},t) = \mathrm{Re}\left\{p(\mathbf{x})\mathrm{e}^{\alpha \mathrm{i}\omega t}\right\}, \tag{2.10}$$

where $\alpha = \pm 1$ is the sign of the chosen time dependency, $\omega = 2\pi f$ is the angular frequency (and f being the frequency), i is the imaginary unit, and $\mathbf{x}$ is a point in space. Inserting the time-harmonic solution into the wave equation, the Helmholtz equation appears

$$\Delta p(\mathbf{x}) + k^2 p(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \tag{2.11}$$

where $k = \frac{\omega}{c}$ is called the wavenumber. The Helmholtz equation naturally occurs in conservation laws and can be interpreted in the frequency domain as a wave equation for monochromatic waves. The Helmholtz equation can be derived from various partial differential equations such as the diffusion equation, the Schödinger equation, and other wave-like equations [25, 40, 65]. From a mathematical point of view, the Helmholtz equation is an eigenvalue problem. Note that (2.11) appears to be independent of the choice of α, which can cause confusion for cases where the time dependence is not explicitly stated. In many cases the normal velocity will be given as a boundary condition, hence the relation between the normal derivative of the pressure and normal velocity is importantly given as

$$\frac{\partial p(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = -\alpha \mathrm{i}\rho_0 c k v_{\mathbf{n}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{2.12}$$

where α is importantly part of the definition. For simplicity, similarly to [58], we introduce

$$s(\alpha) = -\alpha \mathrm{i}\rho_0 c, \tag{2.13}$$

such that

$$\frac{\partial p(\mathbf{x})}{\partial \mathbf{n}(\mathbf{x})} = s(\alpha) k v_{\mathbf{n}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma. \tag{2.14}$$

2.3 Kirchhoff-Helmholtz Integral Equation

For three-dimensional problems the Green's function, which will come in handy soon enough, is given by

$$G(\alpha, \mathbf{x}, \mathbf{y}) = \frac{\exp\left(-\alpha i k \|\mathbf{x}-\mathbf{y}\|_2\right)}{4\pi \|\mathbf{x}-\mathbf{y}\|_2}, \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^3. \tag{2.15}$$

The Green's function is the solution to the modified Helmholtz equation [58]

$$\Delta G(\alpha, \mathbf{x}, \mathbf{y}) + k^2 G(\alpha, \mathbf{x}, \mathbf{y}) = \alpha \delta(\mathbf{x}, \mathbf{y}), \tag{2.16}$$

where $\delta(\mathbf{x}, \mathbf{y})$ is Dirac delta function. Note that the Green's function takes two inputs, $\mathbf{x}$ and $\mathbf{y}$, which is commonly referred to as the target location ($\mathbf{x}$) and the source location ($\mathbf{y}$), or simply the target(s) and source(s).

The first step in deriving the Kirchhoff-Helmholtz Integral equation is to multiply the Helmholtz equation from (2.11) with a test function q followed by an integration over the domain of interest

$$\int_\Omega q(\mathbf{y}) \left[\Delta p(\mathbf{y}) + k^2 p(\mathbf{y})\right] \, \mathrm{d}\Omega_{\mathbf{y}} = 0. \tag{2.17}$$

Integrating by parts twice it follows that

$$\begin{aligned} \int_\Omega q(\mathbf{y}) \left[\Delta p(\mathbf{y}) + k^2 p(\mathbf{y})\right] \, \mathrm{d}\Omega_{\mathbf{y}} = s(\alpha) k \int_\Gamma q(\mathbf{y}) v_{\mathbf{n}}(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}} - \int_\Gamma \frac{\partial q(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}} \\ + \int_\Omega p(\mathbf{y}) \left[\Delta q(\mathbf{y}) + k^2 q(\mathbf{y})\right] \, \mathrm{d}\Omega_{\mathbf{y}} = 0. \end{aligned} \tag{2.18}$$

Now, setting $q(\mathbf{y}) = G(\alpha, \mathbf{x}, \mathbf{y})$ into (2.18) and utilizing that the Green's function is the solution to the modified Helmholtz equation, it follows that

$$\alpha\zeta(\mathbf{x})p(\mathbf{x}) - \int_\Gamma \frac{\partial G(\alpha, \mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y}) \, \mathrm{d}S_\mathbf{y} + s(\alpha)k \int_\Gamma G(\alpha, \mathbf{x}, \mathbf{y}) v_\mathbf{n}(\mathbf{y}) \, \mathrm{d}S_\mathbf{y} = 0, \tag{2.19}$$

where $\zeta(\mathbf{x})$ is the so-called integral free term depending on the geometry at $\mathbf{x}$. For smooth geometries $\zeta(\mathbf{x})$ is equal to $\frac{1}{2}$, but for corners the values deviate between 0 and 1. Since α is part of every term of (2.19) one often finds that it is completely left out in the literature. The unfortunate side effect of this is that the two terms including the Green's function changes signs depending on the chosen time dependency. Although the above equation could be solved directly, it is common to again multiply the equation with a test function $\phi(\mathbf{x})$ and then apply a surface integral to the domain Γ. The resulting equation is the so-called weak formulation

$$\int_\Gamma \phi(\mathbf{x}) \left(\alpha\zeta(\mathbf{x})p(\mathbf{x}) - \int_\Gamma \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y}) \, \mathrm{d}S_\mathbf{y} + s(\alpha)k \int_\Gamma G(\mathbf{x}, \mathbf{y}) v_\mathbf{n}(\mathbf{y}) \, \mathrm{d}S_\mathbf{y} \right) \mathrm{d}S_\mathbf{x} = 0. \tag{2.20}$$

By default, the solution computed using (2.20) satisfies the Sommerfeld radiation condition

$$\lim_{r \to \infty} \left[r^{\frac{d-1}{2}} \left(\frac{\partial p}{\partial r} + \alpha \mathrm{i} k p \right) \right] = 0. \tag{2.21}$$

As a result, the integral formulation has found a footing in both acoustics and electromagnetics, where this condition is often encountered in specialized measurement facilities.

CHAPTER 3

The Boundary Element Method

The boundary element method (BEM) can be used to numerically approximate solutions to boundary integral equations (BIEs). In the area of acoustics, a common BIE is the Kirchhoff-Helmholtz integral equation used to describe time-harmonic acoustics. The literature of BEMs is vast and giving a full summary is beyond the scope of this thesis. Instead, the following chapter will limit itself to only describing the basics required to understand the fundamental ideas of the research carried out during this PhD study.

3.1 Getting the computer to understand functions

In the most basic terms, computers are only capable of understanding numbers, which means that they are inherently unable to solve equations where the unknowns are functions. This is a problem when trying to solve BIEs. To solve this problem, the functions are instead approximated using parameterizations for which the coefficients (numbers) are unknown. Intuitively, these numbers are exactly what the computer is asked to find. For element methods, this parameterization is chosen to be the simplest possible: A linear combination of functions

$$p(\mathbf{x}) \approx \mathbf{T}(\mathbf{x})\mathbf{p} = \begin{bmatrix} T_1(\mathbf{x}) & T_2(\mathbf{x}) & \dots & T_n(\mathbf{x}) \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}, \tag{3.1}$$

where p is the unknown function being approximated. Note that the linearity is with respect to the unknown parameters $\mathbf{p}$, but not necessarily in the known basis functions $\mathbf{T}(\mathbf{x})$. A useful, but not necessary, property is called the Cardinal property stating that

$$T_i(\mathbf{x}_j) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}. \tag{3.2}$$

The usefulness is due the property making the j^{th} value of $\mathbf{p}$ interpretable as the value of the function at point $\mathbf{x}_j$ as

$$\mathbf{T}(\mathbf{x}_j)\mathbf{p} = p_j. \tag{3.3}$$

One might ask: *How does the above relate to the boundary element method?* The answer is that the functions T_i are chosen to be simpler functions with support equal to only a few subdomains of the original domain (the support is drawn in red in Figure 3.4). These subdomains are commonly referred to as *elements*.

3.2 What is an element?

A key insight is that the element serves two purposes: It represents a subdomain of the original domain (also referred to as the geometry) while also describing parts of the unknown function(s) of interest. In the case of surface elements, which are the ones used for BEMs, the parameterization of the subdomain, i.e. the element, is done as

$$\mathbf{x}^e(\mathbf{u}) = \mathbf{X}^e\mathbf{N}^e(\mathbf{u}) \in \Gamma^e, \quad \forall \mathbf{u} \in \mathcal{L}^e, \tag{3.4}$$

where the superscript e denotes the element number, $\mathbf{X}^e$ is a matrix with columns equal to the interpolation nodes of the geometry, $\mathbf{N}^e(\mathbf{u})$ are the so-called shape functions, Γ^e is the element in global coordinates and $\mathcal{L}^e$ are the local coordinates. The structure of some collection of interpolation nodes ($\mathbf{X}^e$) multiplied by some shape function ($\mathbf{N}^e(\mathbf{u})$) is the same for most geometric elements. For BEMs in 3D, it is common to work with triangular elements, which can be seen in Figure 3.1. In particular, the elements in Figure 3.1 are linear, for which a description can be found in Example 3.1.

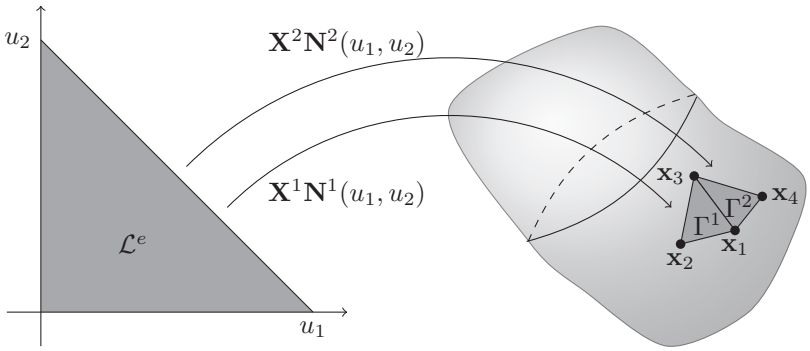


Figure 3.1: The original domain in shown in gray while two (linear) elements are shown in shaded black. The black points denote the interpolation nodes of the elements (the columns of $\mathbf{X}^e$).

Example 3.1. Linear triangular elements

The linear shape functions for a triangular element can have the form

$$\mathbf{N}^e(u_1, u_2) = \begin{bmatrix} 1 - u_1 - u_2 \\ u_1 \\ u_2 \end{bmatrix}, \quad u_1 \in [0, 1], \; u_2 \in [0, 1 - u_1]. \tag{3.5}$$

The choice in the wording *can* is because the ordering of the columns of $\mathbf{X}^e$ can change the ordering rows of $\mathbf{N}^e(\mathbf{u})$ or vice versa. This is something that one should keep in mind in practice when using different mesh file formats. Taking the second element of Figure 3.1 as an example, it could be that

$$\mathbf{X}^2 = \begin{bmatrix} \mathbf{x}_3 & \mathbf{x}_1 & \mathbf{x}_4 \end{bmatrix}. \tag{3.6}$$

Note that extending the geometric interpolation to higher orders is as simple as adding more rows/functions to $\mathbf{N}^e(u_1, u_2)$ as well as more columns/points to $\mathbf{X}^e$.

In addition to the geometric interpolation of each element, we need to further define interpolations of the unknown functions, which in the case of the Kirchhoff-Helmholtz equation is $p(\mathbf{x})$ and $v_\mathbf{n}(\mathbf{x})$. Taking $p(\mathbf{x})$ as an example, we find that on element e this interpolation can be done as

$$p(\mathbf{x}^e(\mathbf{u})) = \mathbf{T}(\mathbf{x}^e(\mathbf{u}))\mathbf{p} = \underbrace{\mathbf{T}(\mathbf{x}(\mathbf{u}))(\mathbf{L}^e)^\top}_{\mathbf{T}^e(\mathbf{u})} \underbrace{\mathbf{L}^e\mathbf{p}}_{\mathbf{p}^e} = \mathbf{T}^e(\mathbf{u})\mathbf{p}^e, \quad \mathbf{u} \in \mathcal{L}^e, \tag{3.7}$$

where $\mathbf{L}^e$ is a permutation-like matrix that extracts the relevant values of $\mathbf{p}$ and orders them such that they correspond to the local basis functions of $\mathbf{T}^e(\mathbf{u})$. The local basis functions are usually chosen as Lagrange polynomials, but other basis functions, such as, e.g., Legendre polynomials and splines, have also been successfully applied [14, 51]. Figure 3.2, Figure 3.3, and Example 3.2 show the interpolation of (3.7) for triangular elements in the case of continuous interpolation (Figure 3.2) and discontinuous interpolation (Figure 3.3). In short, the difference between continuous and discontinuous interpolations is that the interpolation nodes $\mathbf{p}^e$ are shared between multiple elements (continuous elements) or completely inside the element (discontinuous elements). A consequence of the position

of the interpolation nodes is that the value on the shared edge, highlighted in blue in both figures, is uniquely defined for continuous elements, while it serves as a discontinuity for discontinuous elements.

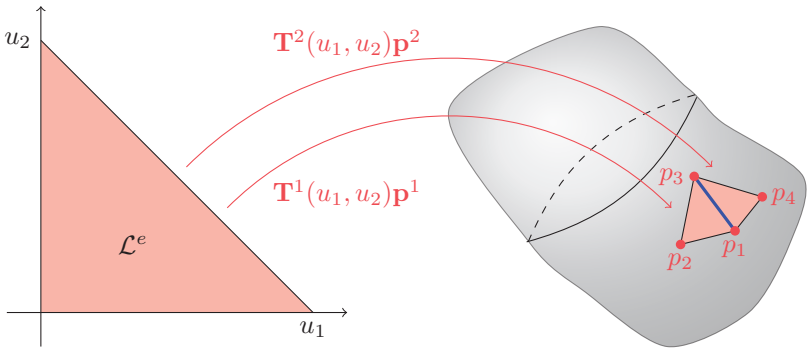


Figure 3.2: The original domain in shown in gray while two continuous linear interpolations on triangular elements are shown in red. The red points denote the interpolation nodes of the elements (where the interpolation of p is equal to a value in $\mathbf{p}^e$).

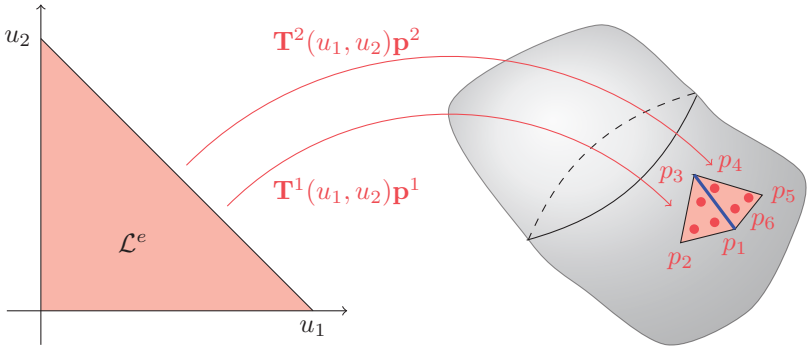


Figure 3.3: The original domain in shown in gray while two discontinuous linear interpolations on triangular elements are shown in red. The red points denote the interpolation nodes of the elements (where the interpolation of p is equal to a value in $\mathbf{p}^e$).

Example 3.2. Basis functions ($\mathbf{T}^e(\mathbf{u})$) for linear interpolation

Continuous linear basis functions on triangular elements are similar to shape functions for a linear triangular element and differ only in the fact that it is the transpose.

$$\mathbf{T}^e_{\text{continuous}}(u_1,u_2) = \begin{bmatrix} 1-u_1-u_2 & u_1 & u_2 \end{bmatrix}, \quad u_1 \in [0,1],\ u_2 \in [0,1-u_1], \tag{3.8}$$

where the subscript "continuous" is only there to highlight that it is a continuous formulation. Again, the ordering of the columns of the row vector depends on the ordering of the element corners. The discontinuous linear interpolation is simply a scaled continuous formulation

$$\mathbf{T}^e_{\text{discontinuous}}(u_1,u_2) = \mathbf{T}^e_{\text{continuous}}\left(\frac{u_1-\beta}{1-3\beta}, \frac{u_2-\beta}{1-3\beta}\right), \tag{3.9}$$

where $\beta \in \left[0, \frac{1}{2}\right[$ is a scalar representing the location of the interpolation nodes in the local $\mathcal{L}^e$ coordinates.

Example 3.3. Element localization matrix

For a continuous linear element e all three corners correspond to a value of the global vector $\mathbf{p}$. For example, the second element in Figure 3.2 has local corner values given by $\mathbf{p}^2 = \begin{bmatrix} p_3 & p_1 & p_4 \end{bmatrix}^\top$. This element would have $\mathbf{L}^e$ given as

$$\mathbf{L}^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \end{bmatrix}, \tag{3.10}$$

so that $\mathbf{p}^2 = \mathbf{L}^2\mathbf{p}$. Note that $\mathbf{L}^e$ is only an artifact of the mathematical description. Any reasonable implementation should use indexing instead of multiplication with $\mathbf{L}^e$.

In the case of the discontinuous description the same element in Figure 3.3 would have $\mathbf{p}^2 = \begin{bmatrix} p_4 & p_5 & p_6 \end{bmatrix}^\top$ meaning that

$$\mathbf{L}^2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}. \tag{3.11}$$

Note here that the discontinuous nature result in $\mathbf{L}^e$ simply picks out three consecutive values.

Note 3.1. Revisiting the geometric interpolation

The element localization matrix can also be used to define the geometric interpolation described in (3.4). Following a similar procedure as for the pressure, we can write the interpolation as

$$\mathbf{x}(\mathbf{u}) = \mathbf{X}\mathbf{N}(\mathbf{u}) = \underbrace{\mathbf{X}\left(\mathbf{L}^e\right)^\top}_{\mathbf{X}^e}\underbrace{\mathbf{L}^e\mathbf{N}(\mathbf{u})}_{\mathbf{N}^e(\mathbf{u})}, \tag{3.12}$$

where $\mathbf{X}$ is a matrix whose columns contain all the interpolation nodes of the geometry and $\mathbf{N}(\mathbf{u})$ are global basis functions. However, in practice, using this formulation does not give an advantage, as $\mathbf{X}$ is fully known. This is opposed to the interpolation of functions for which both $\mathbf{p}$ and $\mathbf{v_n}$ are unknown.

3.3 Applying Quadrature

As we will explore in the next section, the underlying mathematics of boundary element methods is that of boundary integrals. In most cases, it is not possible to compute these integrals analytically. Instead, a quadrature scheme is used to approximate the integrals. To do so, the boundary integrals in global coordinates are transformed into the local element coordinates. This transformation comes with the cost of the need to compute the so-called Jacobian function, which describes the local deformation from the transformation. For 2D and 3D this deformation looks as follows

$$\text{2D:}\quad \text{jacobian}(u) = \left\| \mathbf{X}^e \frac{\mathrm{d}\mathbf{N}^e(u)}{\mathrm{d}u} \right\|_2 \qquad \text{(length deformation)}$$

$$\text{3D:}\quad \text{jacobian}(\mathbf{u}) = \left\| \left(\mathbf{X}^e \frac{\mathrm{d}\mathbf{N}^e(\mathbf{u})}{\mathrm{d}u_1} \right) \times \left(\mathbf{X}^e \frac{\mathrm{d}\mathbf{N}^e(\mathbf{u})}{\mathrm{d}u_2} \right) \right\|_2 \qquad \text{(area deformation)}$$

Applying this the surface integral can be transformed into local coordinates and approximated using a suitable quadrature scheme as

$$\int_{\Gamma^e} f(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} = \int_{\mathcal{L}^e} \text{jacobian}(\mathbf{u}) f(\mathbf{u})\ \mathrm{d}\mathbf{u} \approx \sum_{i=1}^{Q} \text{jacobian}(\mathbf{u}_i) w_i f(\mathbf{u}_i), \tag{3.13}$$

where $\mathbf{u}_i$ is the i^{th} quadrature point with corresponding (quadrature) weight w_i.

3.4 The Boundary Element Method

In simple terms, the boundary element method is a method for solving boundary integral equations through a discretization of both the unknown function and the domain [53, 57, 59, 74]. In this thesis, the main focus is solving the BIE through the so-called collocation approach. This approach is a particular case of the Galerkin approach of (2.20) where the test function is equal to the sum of Dirac-delta functions centered around each of the collocation points, that is,

$$\phi(\mathbf{x}) = \mathbf{a}^\top \begin{bmatrix} \delta(\mathbf{x} - \mathbf{t}_1) \\ \delta(\mathbf{x} - \mathbf{t}_2) \\ \vdots \\ \delta(\mathbf{x} - \mathbf{t}_n) \end{bmatrix}, \tag{3.14}$$

where $\mathbf{a} \in \mathbb{C}^n$ is a vector of arbitrary coefficients and $\mathbf{t}_i \in \mathbb{R}^3$ are the so-called collocation points. For ease of implementation, the collocation points are in most cases chosen as the interpolation nodes of the basis functions. The reason why this result in an easy implementation is that the cardinal property of the basis function can then be easily be used. Inserting this is into (2.20), while setting the constant coefficients $\mathbf{a}^\top$ outside of parentheses it follows that

$$\mathbf{a}^\top \left(\int_\Gamma \begin{bmatrix} \delta(\mathbf{x} - \mathbf{t}_1) \\ \delta(\mathbf{x} - \mathbf{t}_2) \\ \vdots \\ \delta(\mathbf{x} - \mathbf{t}_n) \end{bmatrix} \left(\alpha\zeta(\mathbf{x}) p(\mathbf{x}) - \int_\Gamma \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} + s(\alpha) k \int_\Gamma G(\mathbf{x}, \mathbf{y}) v_{\mathbf{n}}(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} \right) \mathrm{d}S_{\mathbf{x}} \right) = 0. \tag{3.15}$$

By the sifting property of the Dirac-delta function the above can be reduced to

$$\mathbf{a}^\top \left(\alpha \begin{bmatrix} \zeta(\mathbf{t}_1) p(\mathbf{t}_1) \\ \zeta(\mathbf{t}_2) p(\mathbf{t}_2) \\ \vdots \\ \zeta(\mathbf{t}_n) p(\mathbf{t}_n) \end{bmatrix} - \begin{bmatrix} \int_\Gamma \frac{\partial G(\mathbf{t}_1, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} \\ \int_\Gamma \frac{\partial G(\mathbf{t}_2, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} \\ \vdots \\ \int_\Gamma \frac{\partial G(\mathbf{t}_n, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} \end{bmatrix} + s(\alpha) k \begin{bmatrix} \int_\Gamma G(\mathbf{t}_1, \mathbf{y}) v_{\mathbf{n}}(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} \\ \int_\Gamma G(\mathbf{t}_2, \mathbf{y}) v_{\mathbf{n}}(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} \\ \vdots \\ \int_\Gamma G(\mathbf{t}_n, \mathbf{y}) v_{\mathbf{n}}(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}} \end{bmatrix} \right) = 0. \tag{3.16}$$

In its current form, the solution to this equation is a *function* that a computer cannot be tasked to find. Instead, the problem is made computationally tractable by parameterizing the functions p and $v_{\mathbf{n}}$ as

shown in (3.1). As such the above reduces to

$$\mathbf{a}^\top \left(\alpha \begin{bmatrix} \zeta(\mathbf{t}_1)\mathbf{T}(\mathbf{t}_1)\mathbf{p} \\ \zeta(\mathbf{t}_2)\mathbf{T}(\mathbf{t}_2)\mathbf{p} \\ \vdots \\ \zeta(\mathbf{t}_n)\mathbf{T}(\mathbf{t}_n)\mathbf{p} \end{bmatrix} - \begin{bmatrix} \int_\Gamma \frac{\partial G(\mathbf{t}_1,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\mathbf{p}\ \mathrm{d}S_\mathbf{y} \\ \int_\Gamma \frac{\partial G(\mathbf{t}_2,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\mathbf{p}\ \mathrm{d}S_\mathbf{y} \\ \vdots \\ \int_\Gamma \frac{\partial G(\mathbf{t}_n,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\mathbf{p}\ \mathrm{d}S_\mathbf{y} \end{bmatrix} + s(\alpha)k \begin{bmatrix} \int_\Gamma G(\mathbf{t}_1,\mathbf{y})\mathbf{T}(\mathbf{y})\mathbf{v_n}\ \mathrm{d}S_\mathbf{y} \\ \int_\Gamma G(\mathbf{t}_2,\mathbf{y})\mathbf{T}(\mathbf{y})\mathbf{v_n}\ \mathrm{d}S_\mathbf{y} \\ \vdots \\ \int_\Gamma G(\mathbf{t}_n,\mathbf{y})\mathbf{T}(\mathbf{y})\mathbf{v_n}\ \mathrm{d}S_\mathbf{y} \end{bmatrix} \right) \approx 0. \tag{3.17}$$

Setting the parameters $\mathbf{p}$ and $\mathbf{v_n}$ outside the integrals while also using the cardinal property of the basis functions it follows that

$$\mathbf{a}^\top \left(\left(\alpha \mathrm{diag} \left(\begin{bmatrix} \zeta(\mathbf{t}_1) \\ \zeta(\mathbf{t}_2) \\ \vdots \\ \zeta(\mathbf{t}_n) \end{bmatrix} \right) - \begin{bmatrix} \int_\Gamma \frac{\partial G(\mathbf{t}_1,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \int_\Gamma \frac{\partial G(\mathbf{t}_2,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \vdots \\ \int_\Gamma \frac{\partial G(\mathbf{t}_n,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \end{bmatrix} \right) \mathbf{p} + s(\alpha)k \begin{bmatrix} \int_\Gamma G(\mathbf{t}_1,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \int_\Gamma G(\mathbf{t}_2,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \vdots \\ \int_\Gamma G(\mathbf{t}_n,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \end{bmatrix} \mathbf{v_n} \right) \approx 0. \tag{3.18}$$

Since the above has to hold for all $\mathbf{a} \in \mathbb{C}^n$ it follows that the term inside the parentheses is equal to $\mathbf{0}$, i.e.

$$\left(\alpha \mathrm{diag} \left(\begin{bmatrix} \zeta(\mathbf{t}_1) \\ \zeta(\mathbf{t}_2) \\ \vdots \\ \zeta(\mathbf{t}_n) \end{bmatrix} \right) - \begin{bmatrix} \int_\Gamma \frac{\partial G(\mathbf{t}_1,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \int_\Gamma \frac{\partial G(\mathbf{t}_2,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \vdots \\ \int_\Gamma \frac{\partial G(\mathbf{t}_n,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \end{bmatrix} \right) \mathbf{p} + s(\alpha)k \begin{bmatrix} \int_\Gamma G(\mathbf{t}_1,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \int_\Gamma G(\mathbf{t}_2,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ \vdots \\ \int_\Gamma G(\mathbf{t}_n,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \end{bmatrix} \mathbf{v_n} \approx \mathbf{0}. \tag{3.19}$$

Which can be written in matrix form as

$$(\alpha \mathrm{diag}(\boldsymbol{\zeta}) - \mathbf{F})\mathbf{p} + s(\alpha)k\mathbf{G}\mathbf{v_n} = \mathbf{H}\mathbf{p} + s(\alpha)k\mathbf{G}\mathbf{v_n} = \mathbf{0}, \tag{3.20}$$

where we simply defined $\mathbf{H} = \alpha \mathrm{diag}(\boldsymbol{\zeta}) - \mathbf{F}$ for the sake of convenience.

3.4.1 Discretization

While (3.20) is easily stated, it does not mention how to actually compute $\mathbf{G}$ and $\mathbf{H}$. As such, we now briefly explain how to compute the k^{th} row of $\mathbf{F}$ and note that the exact same procedure can be used to compute the k^{th} row of $\mathbf{G}$. The key insight to do this computation is to realize that the interpolation on the e_{th} element is done using (3.7). This means that, for example, the pressure on element e can be described as

$$p(\mathbf{y}^e(\mathbf{u})) \approx \mathbf{T}(\mathbf{y})\mathbf{p} = \underbrace{\mathbf{T}(\mathbf{y}^e(\mathbf{u}))(\mathbf{L}^e)^\top}_{\mathbf{T}^e(\mathbf{u})}\mathbf{L}^e\mathbf{p} \tag{3.21}$$

Approximating the full boundary integral as a sum of each element contribution in addition to inserting (3.21) into the k^{th} row of $\mathbf{F}$ while utilizing the linearity of the sum and integral it follows that

$$\begin{aligned}\int_\Gamma \frac{\partial G(\mathbf{t}_k,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\mathbf{p}\ \mathrm{d}S_{\mathbf{y}} &\approx \sum_{e=1}^{N}\left(\int_{\Gamma^e}\frac{\partial G(\mathbf{t}_k,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})(\mathbf{L}^e)^\top\mathbf{L}^e\ \mathrm{d}S_{\mathbf{y}}\right)\mathbf{p}\\ &\approx \left(\underbrace{\sum_{e=1}^{N}\left(\sum_{i=1}^{Q(\mathbf{t}_k,e)}\frac{\partial G(\mathbf{t}_k,\mathbf{y}^e(\mathbf{u}_i))}{\partial \mathbf{n}(\mathbf{y}^e(\mathbf{u}_i))}\mathrm{jacobian}(\mathbf{u}_i)w_i\mathbf{T}^e(\mathbf{u}_i)\right)\mathbf{L}^e}_{k\text{th row of }\mathbf{F}}\right)\mathbf{p},\end{aligned} \tag{3.22}$$

where it is highlighted that the number of quadrature points, $Q(\mathbf{t}_k, e)$, can be depended on the collocation point $\mathbf{t}_k$ and the e^{th} element. Note that the approximation in the first line of (3.22) is due to the approximation of the geometry using elements while the second approximation is due to the approximation of the integral using a quadrature scheme. The computation of the k^{th} row of $\mathbf{G}$ can be performed analogously.

$$\begin{aligned}\int_\Gamma G(\mathbf{t}_k,\mathbf{y})\mathbf{T}(\mathbf{y})\mathbf{v}_{\mathbf{n}}\ \mathrm{d}S_{\mathbf{y}} &\approx \sum_{e=1}^{N}\left(\int_{\Gamma^e}G(\mathbf{t}_k,\mathbf{y})\mathbf{T}(\mathbf{y})(\mathbf{L}^e)^\top\mathbf{L}^e\ \mathrm{d}S_{\mathbf{y}}\right)\mathbf{v}_{\mathbf{n}}\\ &\approx \left(\underbrace{\sum_{e=1}^{N}\left(\sum_{i=1}^{Q}G(\mathbf{t}_k,\mathbf{y}^e(\mathbf{u}_i))\mathrm{jacobian}(\mathbf{u}_i)w_i\mathbf{T}^e(\mathbf{u}_i)\right)\mathbf{L}^e}_{k\text{th row of }\mathbf{G}}\right)\mathbf{v}_{\mathbf{n}},\end{aligned} \tag{3.23}$$

A considerable downside of the BEM is that the resulting matrices $\mathbf{H}, \mathbf{G} \in \mathbb{C}^{n\times n}$ are dense, meaning that memory used to store the two matrices on a computer scales as $\mathcal{O}(n^2)$, rendering the direct application of the method unusable for large n. However, various methods have been developed that reduce memory consumption to $O(n\log(n))$ or even $\mathcal{O}(n)$ by utilizing the rank-structure of the problem. Many of these methods depend only on defining the matrix-vector products with the matrices, which means that the resulting linear system of equation, like (3.19), must be solved using an iterative scheme. The most common of these schemes is the generalized minimal residual method (GMRes) [60, 70, 71].

3.5 Acceleration Methods

The aim of this section is to show how matrix-vector product of $\mathbf{G}$ and $\mathbf{H}$ (or maybe more precisely $\mathbf{F}$) can be reduced from $O(n^2)$ to $O(n)$ using the fast multipole method (FMM) or hierarchical/$\mathcal{H}$-matrices. To do so, we start by assuming that the integration of each element can be done using the same Q quadrature points[1]. As solving linear systems using an iterative scheme only requires the matrix-vector products, we look into these specifically for $\mathbf{G}$ and $\mathbf{F}$. For simplification purposes, we look only at the k_{th} rows, as extending this to include all rows is trivial. We start by looking at $\mathbf{G}$

$$\begin{aligned}\left(\underbrace{\int_\Gamma G(\mathbf{t}_k,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_{\mathbf{y}}}_{k\text{th row of }\mathbf{G}}\right)\mathbf{z} &\approx \left(\sum_{e=1}^{N}\left(\sum_{i=1}^{Q}G(\mathbf{t}_k,\mathbf{y}^e(\mathbf{u}_i))\mathrm{jacobian}(\mathbf{u}_i)w_i\mathbf{T}^e(\mathbf{u}_i)\right)\mathbf{L}^e\right)\mathbf{z}\\ &= \left(\sum_{j=1}^{NQ}G(\mathbf{t}_k,\mathbf{y}_j)\underbrace{\mathrm{jacobian}(\mathbf{u}_j)w_j\mathbf{T}^{e(j)}(\mathbf{u}_j)\mathbf{L}^{e(j)}}_{j\text{th row of }\mathbf{C}}\right)\mathbf{z}\\ &= \begin{bmatrix}G(\mathbf{t}_k,\mathbf{y}_1) & G(\mathbf{t}_k,\mathbf{y}_2,) & \dots & G(\mathbf{t}_k,\mathbf{y}_{NQ})\end{bmatrix}\mathbf{C}\mathbf{z}\end{aligned} \tag{3.24}$$

[1] This will cause numerical issues that are then fixed by a near field correction step

where the subscript j refers to an ordering of the collection of Gaussian points from all elements and $e(j)$ is a function that returns the element number that Gaussian point j is located on. Furthermore, the matrix $\mathbf{C}$ is *extremely* sparse and can be thought of as a transformation/interpolation of values ($\mathbf{z}$) at the collocation points ($\mathbf{t}_k$) into coefficients $\left(\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \dots & c_{NQ} \end{bmatrix}^\top\right)$ at the Gaussian points ($\mathbf{y}_j$). A similar approach can be applied to $\mathbf{F}$

$$\begin{aligned}\left(\underbrace{\int_\Gamma \frac{\partial G(\mathbf{t}_k, \mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} \mathbf{T}(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}}}_{k\text{th row of } \mathbf{F}}\right) \mathbf{z} &\approx \left(\sum_{e=1}^{N} \left(\sum_{i=1}^{Q} \frac{\partial G(\mathbf{t}_k, \mathbf{y}^e(\mathbf{u}_i))}{\partial \mathbf{n}(\mathbf{y}^e(\mathbf{u}_i))} \mathrm{jacobian}(\mathbf{u}_i) w_i \mathbf{T}^e(\mathbf{u}_i)\right) \mathbf{L}^e\right) \mathbf{z} \\ &= \left(\sum_{j=1}^{NQ} \mathbf{n}(\mathbf{y}_j)^\top \nabla G(\mathbf{t}_k, \mathbf{y}_j) \underbrace{\mathrm{jacobian}(\mathbf{u}_j) w_j \mathbf{T}^{e(j)}(\mathbf{u}_j) \mathbf{L}^{e(j)}}_{j\text{th row of } \mathbf{C}}\right) \mathbf{z} \\ &= \begin{bmatrix} \mathbf{n}(\mathbf{y}_1)^\top \nabla G(\mathbf{t}_k, \mathbf{y}_1) & \dots & \mathbf{n}(\mathbf{y}_{NQ})^\top \nabla G(\mathbf{t}_k, \mathbf{y}_{NQ}) \end{bmatrix} \mathbf{C}\mathbf{z},\end{aligned} \tag{3.25}$$

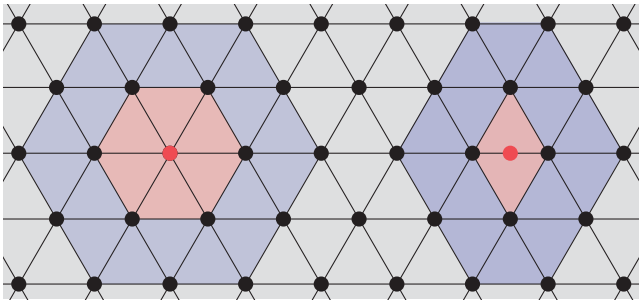
where $\mathbf{C}$ similar to (3.24) transforms $\mathbf{z}$ into some coefficients $\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \dots & c_{NQ} \end{bmatrix}^\top$.

Applying directly (3.24) or (3.25) for all rows of $\mathbf{G}$ and $\mathbf{H}$ will result in products that scale $O(NQn)$, which could hurt more than we gain (as $NQ \gtrapprox n$). However, the fast multipole method or the $\mathcal{H}$-matrix approach can be used to speed up the product with the dense part of the product. However, doing so blindly will cause some numerical errors stemming from the integration of elements close to the collocation point. To resolve this issue, a near-field correction matrix is applied to the product. As such, applying (3.24) and (3.25) for all rows at the same time can be thought of as

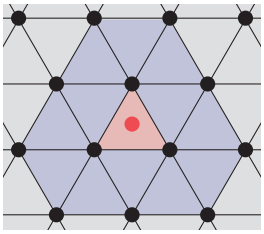
$$\mathbf{A} = \mathbf{B}\mathbf{C} + \mathbf{S}, \quad \mathbf{S}, \mathbf{A} \in \mathbb{C}^{n \times n}, \ \mathbf{B} \in \mathbb{C}^{n \times NQ}, \ \mathbf{C} \in \mathbb{R}^{NQ \times n}, \tag{3.26}$$

where $\mathbf{B}$ is the dense part approximated by either the fast multipole method or a $\mathcal{H}$-matrix (see (3.28) and (3.29) for examples of $\mathbf{B}$), $\mathbf{C}$ is the coefficient map and $\mathbf{S}$ is the near field correction. In short, the near-field correction matrix subtracts the wrong integration done by using only Q Gaussian points and adds the correct integration instead. It is important to note that $\mathbf{S}$ and $\mathbf{C}$ are highly sparse matrices, meaning that both the assembly and the matrix-vector product scale as $\mathcal{O}(n\tau)$ where $\tau \ll n$. This means that using an approximate scheme for $\mathbf{B}$ with a product that scales linearly in time and memory results in a representation of $\mathbf{A}$ that scales similarly.

A priori knowledge of the sparsity pattern of the near-field correction step is needed to compute it efficiently. In the case of collocation, such knowledge could, for example, be the neighboring elements of the interpolation nodes (Figure 3.4). While this approach works well in many cases, it will fail in instances where the elements that are not neighbors are close geometrically, as is the case of e.g. flat structures.



(a) Continuous Elements.



(b) Discontinuous elements.

Figure 3.4: *The figure originally appeared in Paper J3.* Supports of the interpolation functions on the vertex/edge node is shown in red. In some cases, one can also increase the singularity extraction to include the blue areas.

3.5.1 The Fast Multipole Method

The fast multipole method (FMM) can be used to accelerate sums of Green's functions ($\mathbf{G}$) as well as the normal derivative of the Green's function ($\mathbf{H}$) [24, 38, 39]. Throughout the years, many good resources have been written explaining the intricacies of fast multipole methods. For a gentle introduction to the topic, I recommend [12] and [77], while for a more detailed look with an emphasis on boundary element methods, [40] and [76] are good choices. Also, Lexing Ying, the author of [77], has a wonderful introductory talk which can be found online [78]. Given these resources, the specific details of fast multipole methods will be left out here. Instead, we focus on the intuition behind their connection to boundary element methods. For this, it is enough to think of fast multipole methods as a way of speeding up sums of the form

$$u(\mathbf{t}_k) = \sum_{j=1}^{NQ} G(\mathbf{t}_k, \mathbf{y}_j) c_j, \quad u(\mathbf{t}_k) = \sum_{j=1}^{NQ} \mathbf{n}(\mathbf{y}_j)^\top \nabla G(\mathbf{t}_k, \mathbf{y}_j) c_j \tag{3.27}$$

where when $\mathbf{t}_k = \mathbf{y}_j$ the j^{th} term is excluded from the sum. Looking back at (3.24) and (3.25) it can be seen that (3.27) is exactly what is being computed for each of the rows of $\mathbf{G}$ and $\mathbf{F}$ with coefficients given as $\mathbf{c} = \mathbf{C}\mathbf{z}$. Applying the fast multipole method to all n collocation points results in something that scales $O((n + NQ)\log^{3/2}(1/\varepsilon))$ where ε is the relative precision. Many fast multipole software libraries have been developed over the years, such as `ExaFMM` [73], `ScalFMM` [72], and `FMM3D` [4]. Due to the existence of their Julia interface, it was chosen in this study to work with `FMM3D` developed at the Flatiron Institute.

3.5.2 Hierarchical Matrices

The $\mathcal{H}$-matrix approach differs from the fast multipole method by the fact that it does not accelerate the sums of (3.27) directly [61, 62]. Instead, it approximates the matrix that corresponds to performing the sum in (3.27) for all the n collocation points at the same time, that is, for all $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n$. Taking the $\mathbf{G}$ matrix as an example this means approximating a matrix of the form

$$\mathbf{B} = \begin{bmatrix} G(\mathbf{t}_1, \mathbf{y}_1) & G(\mathbf{t}_1, \mathbf{y}_2) & \ldots & G(\mathbf{t}_1, \mathbf{y}_{NQ}) \\ G(\mathbf{t}_2, \mathbf{y}_1) & G(\mathbf{t}_2, \mathbf{y}_2) & \ldots & G(\mathbf{t}_2, \mathbf{y}_{NQ}) \\ \vdots & \vdots & \ddots & \vdots \\ G(\mathbf{t}_n, \mathbf{y}_1) & G(\mathbf{t}_n, \mathbf{y}_2) & \ldots & G(\mathbf{t}_n, \mathbf{y}_{NQ}) \end{bmatrix}, \tag{3.28}$$

based on the fact that subblocks of the matrix is well approximated by low-rank matrices. For the $\mathbf{H}$ matrix the matrix being approximating by low-rank matrices is given by

$$\mathbf{B} = \begin{bmatrix} \mathbf{n}(\mathbf{y}_1)^\top \nabla G(\mathbf{t}_1, \mathbf{y}_1) & \mathbf{n}(\mathbf{y}_2)^\top \nabla G(\mathbf{t}_1, \mathbf{y}_2) & \dots & \mathbf{n}(\mathbf{y}_{NQ})^\top \nabla G(\mathbf{t}_1, \mathbf{y}_{NQ}) \\ \mathbf{n}(\mathbf{y}_1)^\top \nabla G(\mathbf{t}_2, \mathbf{y}_1) & \mathbf{n}(\mathbf{y}_2)^\top \nabla G(\mathbf{t}_2, \mathbf{y}_2) & \dots & \mathbf{n}(\mathbf{y}_{NQ})^\top \nabla G(\mathbf{t}_2, \mathbf{y}_{NQ}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{n}(\mathbf{y}_1)^\top \nabla G(\mathbf{t}_n, \mathbf{y}_1) & \mathbf{n}(\mathbf{y}_2)^\top \nabla G(\mathbf{t}_n, \mathbf{y}_2) & \dots & \mathbf{n}(\mathbf{y}_{NQ})^\top \nabla G(\mathbf{t}_n, \mathbf{y}_{NQ}) \end{bmatrix}, \tag{3.29}$$

Exactly how hierarchical low-rank approximation is computed is, however, outside the scope of this thesis, but the interested reader should look at [6, 13, 42] where I would like to highlight [6] as my preferred introduction. In addition a wonderful lecture series by Alex Barnett, Adrianna Gillman and Per-Gunnar Martinsson from the *2014 CBMS-NSF Conference: Fast Direct Solvers for Elliptic PDEs* can be found online [7]. There exist many implementations of hierarchical matrices such as `HLIBpro` [54], `HLib` [21], and `H2Lib` [22]. In Julia the packages `KernelMatrices.jl` [48], `HssMatrices.jl` [19] and `HMatrices.jl` [32] all implement various hierarchical matrix formats with different properties. For an even longer list of various software, see [23]. In this PhD study, it was chosen to utilize `HMatrices.jl` developed by Luiz M. Faria as it was general, fast, and perhaps even more importantly written with integral equations in mind. The specific format implemented in this library is the standard that scales $O(n\log(n))$ in terms of memory and computation. Note that an often mentioned strength of the $\mathcal{H}$-matrix approach is that it is possible to compute the LU-factorization of the matrix efficiently, which is particularly important for high-frequency problems where an iterative solver might have issues converging. However, in the case of (3.26) an LU-factorization cannot be applied directly. Worse still, when dealing with losses, as seen in (4.50), the matrix of interest has a form even more complicated than in (3.26).

3.5.3 Interpolated Factored Green Function

Recently a new approach called the "Interpolated Factored Green Function" (IFGF) was introduced in [9]. Some advantages of this approach is that it does not require a split on low and high frequencies as needed for the fast multipole method. At the same time, the method is easily parallelized as it does not rely on the fast Fourier transform (FFT) [8]. However, as opposed to the purely algebraic techniques of the $\mathcal{H}$-matrix approach, the IFGF is based on interpolation. As such, it requires that the matrix is generated by a function that can be evaluated at all sets of points $(\mathbf{x}, \mathbf{y})$. Unfortunately, this is not the case for $\mathbf{H}$ as its elements depend on the normal which is computed using the local $\mathbf{u}$ coordinates due to the element description of the geometry. A workaround is to approximate the full gradient of the Green's function using the IFGF, as it can be easily evaluated at any $(\mathbf{x}, \mathbf{y})$, and then compute the normal derivative from this. A simple (single node) implementation of the approach can be found in the `IFGF.jl` package written by Luiz M. Faria [33].

CHAPTER 4

Viscothermal Effects

This chapter introduces the basic concepts of viscous and thermal dissipation. It starts by describing the Kirchhoff decomposition (KD) and its solution through the BEM is explained in two parts. The first part introduces the old state-of-the-art formulation, while the second part explains the new formulation.

4.1 Viscothermal BEM

4.1.1 Kirchhoff Decomposition

An equivalent formulation of the fully linearized Navier-Stokes (FLNS) equation is the Kirchhoff decomposition, which divides the problem into three *modes*, each satisfying its own Helmholtz equation.

$$\text{Acoustic Mode:} \quad (\Delta + k_a^2) p_a(\mathbf{x}) = 0, \tag{4.1}$$

$$\text{Thermal Mode:} \quad (\Delta + k_h^2) p_h(\mathbf{x}) = 0, \tag{4.2}$$

$$\text{Viscous Mode:} \quad (\Delta + k_v^2) \mathbf{v}_v(\mathbf{x}) = \mathbf{0}, \quad \text{with} \quad \nabla \cdot \mathbf{v}_v(\mathbf{x}) = 0. \tag{4.3}$$

The three modal wavenumbers ($k_a,\ k_h$ & k_v) all depend on the lossless wavenumber (k) and the physical properties of the fluid, such as the thermal conductivity, specific heat capacity under constant pressure and the shear/bulk viscosity coefficients [43]. The total pressure and velocity can be extracted as the sum of the contributions of each of the three modes

$$p_t = p_a + p_h, \tag{4.4}$$

$$\mathbf{v}_t = \mathbf{v}_a + \mathbf{v}_h + \mathbf{v}_v. \tag{4.5}$$

Using the standard notation for the BEM discretization, i.e., using (3.20) for the three modes, it must be true that

$$\mathbf{H}_a \mathbf{p}_a + \mathbf{G}_a \frac{\partial \mathbf{p}_a}{\partial \mathbf{n}} = \mathbf{0}, \quad \mathbf{H}_h \mathbf{p}_h + \mathbf{G}_h \frac{\partial \mathbf{p}_h}{\partial \mathbf{n}} = \mathbf{0}, \quad \mathbf{H}_v \mathbf{v} + \mathbf{G}_v \frac{\partial \mathbf{v}}{\partial \mathbf{n}} = \mathbf{0}, \tag{4.6}$$

where

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_{v_x} & \mathbf{v}_{v_y} & \mathbf{v}_{v_z} \end{bmatrix}^\top, \quad \partial_{\mathbf{n}} \mathbf{v} = \begin{bmatrix} \partial_{\mathbf{n}} \mathbf{v}_{v_x} & \partial_{\mathbf{n}} \mathbf{v}_{v_y} & \partial_{\mathbf{n}} \mathbf{v}_{v_z} \end{bmatrix}^\top, \tag{4.7}$$

and

$$\mathbf{H}_v = \text{blkdiag}\left(\widetilde{\mathbf{H}_v},\ \widetilde{\mathbf{H}_v},\ \widetilde{\mathbf{H}_v}\right), \quad \mathbf{G}_v = \text{blkdiag}\left(\widetilde{\mathbf{G}_v},\ \widetilde{\mathbf{G}_v},\ \widetilde{\mathbf{G}_v}\right), \tag{4.8}$$

with $\widetilde{\mathbf{H}_v}$ and $\widetilde{\mathbf{G}_v}$ being the result of a scalar discretization with wavenumber k_v and blkdiag is the block-diagonal operator.

4.1.2 Boundary Coupling of the Modes

In the case of the fluid being air, it is valid to prescribe the so-called isothermal boundary condition due to high higher heat capacity of the boundary material. Mathematically, this condition can be written as

$$p_a(\mathbf{x})\tau_a + p_h(\mathbf{x})\tau_h = 0, \quad \mathbf{x} \in \Gamma, \tag{4.9}$$

where τ_a and τ_h are dependent on the lossless wavenumber as well as the physical properties of fluid. In short, this boundary condition forces the temperature fluctuations to be zero, meaning that there is no exchange of heat between the fluid and the boundary. As such, there is a transition from the physics at the boundary to the bulk where the wave propagation can be considered isentropic (Figure 4.1). This transitional phase is in the literature called the thermal boundary layer and has a thickness that depends on the frequency. In air, the thickness can be approximated as [64]

$$\delta_h \approx 2.5 \frac{1}{\sqrt{f}} \text{ mm}. \tag{4.10}$$

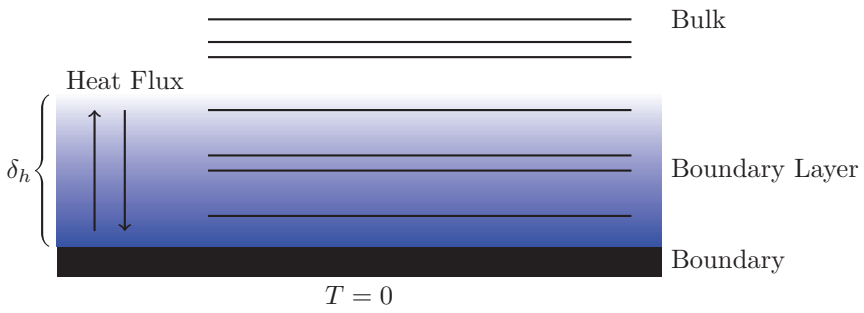


Figure 4.1: Visualization of the thermal boundary layer representing the transitional phase from the boundary to the bulk.

Furthermore, due to frictional forces the air sticks to the surface of objects. As such, it is common to prescribe a no-slip boundary condition

$$\nabla p_a(\mathbf{x})\phi_a + \nabla p_h(\mathbf{x})\phi_h + \mathbf{v}_v(\mathbf{x}) = \mathbf{v}_{\text{boundary}}(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{4.11}$$

Similar to the modal wavenumbers the constants ϕ_a, and ϕ_h are dependent on the lossless wavenumber as well as the physical properties of fluid. Again, the boundary condition results in a transitional phase from the boundary to the bulk (Figure 4.2). The thickness of the viscous boundary layer is of the same order as that of the thermal boundary layer, and can be approximated as [64]

$$\delta_v \approx 2.1 \frac{1}{\sqrt{f}} \text{ mm}. \tag{4.12}$$

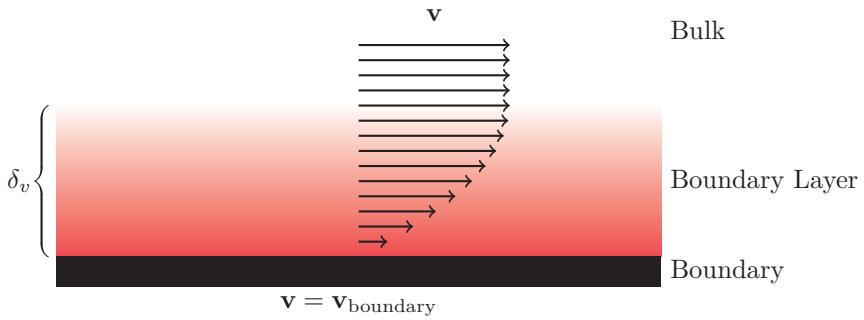


Figure 4.2: Visualization of the viscous boundary layer representing the transitional phase from the boundary to the bulk.

4.1.3 The Lossy System of Equations

The components of the three modes can be computed by solving the large system of equations originating from the combination of the three discretized integral equations, null-divergence constraint and boundary conditions

$$\begin{bmatrix} \text{The three modes on the form described in (4.6)} \\ \text{Null Divergence Constraint} \\ \text{Isothermal Boundary Condition} \\ \text{No-Slip Boundary Condition} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \partial_{\mathbf{n}}\mathbf{p}_a \\ \mathbf{p}_h \\ \partial_{\mathbf{n}}\mathbf{p}_h \\ \mathbf{v} \\ \partial_{\mathbf{n}}\mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_s \end{bmatrix}, \tag{4.13}$$

where

$$\mathbf{v}_s = \begin{bmatrix} \mathbf{v}_{x_0} & \mathbf{v}_{y_0} & \mathbf{v}_{z_0} \end{bmatrix}^\top, \tag{4.14}$$

is the boundary velocities at each of the collocation nodes stacked with respect to the x, y, and z direction. From an implementation point of view, (4.13) poses a challenge: How are the gradients required for the two boundary conditions and the null-divergence constraint computed? The answer lies in the boundary element representation of the underlying variables.

4.1.4 Interpolation Function Derivatives

The interpolation function derivatives (IFDs) are the derivatives of the boundary element interpolation functions. As the boundary element interpolation is defined only on the surface, the interpolation function derivatives contain only the tangential information. The full gradient is then computed by combining the tangential information from the IFDs with the normal information coming directly from the boundary element interpolation of the normal derivative.

Taking the acoustic pressure p_a as an example, it follows from (3.7) that the boundary element representation of the interpolation on element e is given as

$$p_a\left(\mathbf{x}^e(\mathbf{u})\right) = \mathbf{T}^e(\mathbf{u})\mathbf{p}_a^e, \quad \mathbf{u} \in \mathcal{L}^e,\ \mathbf{x} \in \Gamma_e, \tag{4.15}$$

where $\mathbf{p}_a^e$ represent the nodal acoustical pressures of element e while $\mathbf{N}^e(\mathbf{u})$ and $\mathbf{T}^e(\mathbf{u})$ denotes the chosen interpolation schemes for respectively the geometry and the (acoustical) pressure on element e. The resulting gradient of p_a on element e can then be computed as

$$\nabla_{\mathbf{u}} p_a\left(\mathbf{x}^e(\mathbf{u})\right) = \left(\nabla_{\mathbf{u}}\mathbf{T}^e(\mathbf{u})\right)\mathbf{p}_a^e \tag{4.16}$$

since the values $\mathbf{p}_a^e$ are constants. Applying the multivariate chain rule $\nabla_{\mathbf{u}} p_a = (\nabla_{\mathbf{u}}\mathbf{x}^e(\mathbf{u})^\top)\nabla_{\mathbf{x}} p_a$ followed by an isolation with respect to $\nabla_{\mathbf{x}}$ gives the gradient of interest. Since the computations are based on boundary elements we have that the transpose of the Jacobian matrix

$$\nabla_{\mathbf{u}}\mathbf{x}^e(\mathbf{u})^\top = \nabla_{\mathbf{u}}(\mathbf{X}^e\mathbf{N}^e(\mathbf{u}))^\top = \nabla_{\mathbf{u}}\mathbf{N}^e(\mathbf{u})^\top\left(\mathbf{X}^e\right)^\top, \tag{4.17}$$

is a 2×3-matrix which cannot be inverted. It has been shown that it is enough to introduce an artificial u_3 such that $\frac{\partial \mathbf{T}^e}{\partial u_3} = \mathbf{0}$ (with $\mathbf{0}$ being a row vector of appropriate length filled with zeros), while substituting $\frac{\partial \mathbf{x}^e}{\partial u_3} = \frac{\partial \mathbf{x}^e}{\partial u_1} \times \frac{\partial \mathbf{x}^e}{\partial u_2}$ [5]. As a result, the tangential part of the gradient can be computed as

$$\nabla_{\mathbf{x}}^{\parallel} p_a\left(\mathbf{x}^e(\mathbf{u})\right) = \begin{bmatrix} \left(\dfrac{\partial \mathbf{x}^e}{\partial u_1}\right)^\top \\ \left(\dfrac{\partial \mathbf{x}^e}{\partial u_2}\right)^\top \\ \left(\dfrac{\partial \mathbf{x}^e}{\partial u_1} \times \dfrac{\partial \mathbf{x}^e}{\partial u_2}\right)^\top \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{\partial \mathbf{T}^e}{\partial u_1} \\ \dfrac{\partial \mathbf{T}^e}{\partial u_2} \\ \mathbf{0} \end{bmatrix} \mathbf{p}_a^e, \quad \mathbf{x} \in \Gamma_e, \tag{4.18}$$

where the superscript $\|$ is used to explicitly show that this is only the tangential part of the gradient. On element e there is a local coordinate $\mathbf{u}$ for which $\mathbf{x}^e(\mathbf{u})$ is equal to a collocation point. Say that this is the i^{th} collocation point then this means that there is a $\mathbf{u}^{e,i}$ for which $\mathbf{x}^e\left(\mathbf{u}^{e,i}\right) = \mathbf{x}_i$. Using what was described in subsection 3.4.1, namely that $\mathbf{p}_a^e = \mathbf{L}^e \mathbf{p}_a$, it follows that

$$\nabla_{\mathbf{x}}^{\|} p_a\left(\mathbf{x}^e(\mathbf{u}^{e,i})\right) = \nabla_{\mathbf{x}}^{\|} p_a(\mathbf{x}_i) = \begin{bmatrix} \left(\dfrac{\partial \mathbf{x}^e}{\partial u_1}\right)^\top \\ \left(\dfrac{\partial \mathbf{x}^e}{\partial u_2}\right)^\top \\ \left(\dfrac{\partial \mathbf{x}^e}{\partial u_1} \times \dfrac{\partial \mathbf{x}^e}{\partial u_2}\right)^\top \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{\partial \mathbf{T}^e}{\partial u_1} \\ \dfrac{\partial \mathbf{T}^e}{\partial u_2} \\ \mathbf{0} \end{bmatrix} \mathbf{L}^e \mathbf{p}_a = \begin{bmatrix} \mathbf{D}_x^{e,i} \\ \mathbf{D}_y^{e,i} \\ \mathbf{D}_z^{e,i} \end{bmatrix} \mathbf{p}_a. \tag{4.19}$$

In the case of discontinuous elements, the collocation point is only connected to a single element, meaning that in practice the e^{th} superscript is redundant. However, in the case of continuous elements, the collocation point can be connected to multiple elements, and the interpolation function derivative is chosen to be the average contribution from each of the connected elements. As such, the i^{th} rows of the $\mathbf{D}_\bullet$-matrices can be computed as

$$\mathbf{D}_x^i = \frac{1}{N_e(i)} \sum_{e=1}^{N_e(i)} \mathbf{D}_x^{e,i}, \quad \mathbf{D}_y^i = \frac{1}{N_e(i)} \sum_{e=1}^{N_e(i)} \mathbf{D}_y^{e,i}, \quad \mathbf{D}_z^i = \frac{1}{N_e(i)} \sum_{e=1}^{N_e(i)} \mathbf{D}_z^{e,i}, \tag{4.20}$$

where $N_e(i)$ denotes the number of elements that is connected to collocation point i. The different $\mathbf{D}_\bullet^i$ of (4.20) are the collection of the i^{th} rows in three separate matrices, $\mathbf{D}_x, \mathbf{D}_y$ and $\mathbf{D}_z$ so that

$$\frac{\partial \mathbf{p}_a}{\partial x}^{\|} = \mathbf{D}_x \mathbf{p}_a, \quad \frac{\partial \mathbf{p}_a}{\partial y}^{\|} = \mathbf{D}_y \mathbf{p}_a, \quad \frac{\partial \mathbf{p}_a}{\partial z}^{\|} = \mathbf{D}_z \mathbf{p}_a. \tag{4.21}$$

If the chosen discretization of the thermal and viscous modes is the same as for the acoustical mode, then the above interpolation function derivative matrices can be reused to compute $\nabla_{\mathbf{x}}^{\|} p_h$ and $\nabla_{\mathbf{x}}^{\|} \cdot \mathbf{v}_v$ respectively.

4.2 State of the art

In the viscothermal acoustics literature, the observation that the IFD only represented the tangential part of the gradient was only implicitly used [28, 68]. We now briefly explain why this was the case. Previously the normal and tangential components were coupled by transforming the derivatives in the (x, y, z)-coordinates (denoted later by e) into local (t, s, n)-coordinates (denoted later by ℓ) at each collocation point through a change of basis

$$\begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{D}_n \end{bmatrix} = {}_\ell\mathbf{M}_e \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_x \end{bmatrix}, \tag{4.22}$$

where ${}_\ell\mathbf{M}_e$ is an (orthogonal) change of basis matrix given by

$${}_\ell\mathbf{M}_e = \begin{bmatrix} \operatorname{diag}(\mathbf{t}_x) & \operatorname{diag}(\mathbf{t}_y) & \operatorname{diag}(\mathbf{t}_z) \\ \operatorname{diag}(\mathbf{s}_x) & \operatorname{diag}(\mathbf{s}_y) & \operatorname{diag}(\mathbf{s}_z) \\ \operatorname{diag}(\mathbf{n}_x) & \operatorname{diag}(\mathbf{n}_y) & \operatorname{diag}(\mathbf{n}_z) \end{bmatrix} \tag{4.23}$$

$(\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z)$ are respectively the (x, y, z)-coordinates of the first tangential direction (t) and similarly $(\mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z)$ are respectively the (x, y, z)-coordinates of the second tangential direction (s). The argument for why (4.23) is orthogonal comes from the fact that it is a permutation of rows and columns of a block-diagonal matrix for which each block is orthogonal (since they are local orthogonal change-of-basis matrices), meaning that the block-diagonal matrix is orthogonal. As permutations preserve orthogonality, it must be true that ${}_\ell\mathbf{M}_e$ is also orthogonal. The gradient in local coordinates can then be described by combining the first two components of the above, representing the derivatives in the tangential plane, with the normal derivative described directly from the BE discretization. As a result, the local gradient was computed as

$$\nabla_\ell \mathbf{p}_a = \begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{0} \end{bmatrix} \mathbf{p}_a + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} \partial_{\mathbf{n}} \mathbf{p}_a, \tag{4.24}$$

where $\mathbf{0} \in \mathbb{R}^{n \times n}$ is a matrix filled with zeros. While the above is clearly logical, it begs the question of what happened to $\mathbf{D}_n$? The answer is simply that since the IFDs do not contain any information in the normal direction, it must be true that $\mathbf{D}_n = \mathbf{0}$. However, this was never explained before. Now, the sole reason for computing the gradient above is to apply it when asserting the no-slip boundary condition. Unfortunately, inserting (4.24) directly into the no-slip boundary condition is problematic because $\mathbf{v}$ is given in global coordinates. To solve this problem, a change of basis, similar to that in (4.22), is applied to $\mathbf{v}$, resulting in

$$\mathbf{v}_\ell = {}_\ell\mathbf{M}_e \mathbf{v}. \tag{4.25}$$

As such, the no-slip boundary condition could be rewritten as

$$\phi_a \left(\begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{0} \end{bmatrix} \mathbf{p}_a + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} \partial_{\mathbf{n}} \mathbf{p}_a \right) + \phi_h \left(\begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{0} \end{bmatrix} \mathbf{p}_h + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} \partial_{\mathbf{n}} \mathbf{p}_h \right) + {}_\ell\mathbf{M}_e \mathbf{v} = {}_\ell\mathbf{M}_e \mathbf{v}_s, \tag{4.26}$$

where $\mathbf{v}_s$ is the surface velocity in global coordinates.

Computing the divergence was done using a similar approach to that of (4.24) with the problem of $\mathbf{v}$ being given in global coordinates still being an issue. Similar to the gradient computation the solution used was to apply a change of basis to $\mathbf{v}$ and combining the result with the BE normal derivative as shown below

$$\nabla_\ell \cdot \mathbf{v}_\ell = \begin{bmatrix} \mathbf{D}_t & \mathbf{D}_s & \mathbf{0} \end{bmatrix} \mathbf{v}_\ell + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \partial_{\mathbf{n}} \mathbf{v}_\ell, \tag{4.27}$$

where $\mathbf{v}_\ell$ is given in (4.25) and

$$\partial_{\mathbf{n}} \mathbf{v}_\ell = {}_\ell\mathbf{M}_e \partial_{\mathbf{n}} \mathbf{v}. \tag{4.28}$$

Although keeping everything in local ℓ coordinates works in practice, it introduces additional model complexity, as it requires knowledge of the two tangential directions, $\mathbf{t}$ and $\mathbf{s}$ (see the definition of ${}_\ell\mathbf{M}_e$ in (4.23)). This complexity is exactly what the new approach described in section 4.3 avoids by utilizing the fact that the IFDs only contain the information in the tangential directions and directly combine that with the information in the normal direction described by the BEM discretization through a simple projection. Furthermore, previous work tried to simplify the derivation by applying the following relation

$$\text{diag}(\mathbf{x}) \mathbf{D} \text{diag}(\mathbf{x}) = \mathbf{D} \circ (\mathbf{x}^\top \mathbf{x}). \tag{4.29}$$

While mathematically true, a naive implementation of this approach would scale $O(n^2)$ in both computation and storage, as it would require the computation and storage of an outer product. However, this problem could have been resolved by the fact that $\mathbf{D}_t$ and $\mathbf{D}_s$ are sparse and then only compute the parts of the outer product that correspond to the same sparsity pattern.

4.3 A new formulation (Paper J2)

The aim of the new formulation is to eliminate the change-of-basis utilized in both the no-slip and the null-divergence conditions. For the no-slip boundary condition, the solution is simply to realize that the full gradient is the sum of the tangential and normal gradient information. Written out this means that

$$\nabla p_a = \nabla^{\parallel} p_a + \nabla^{\perp} p_a, \tag{4.30}$$

where $\nabla^{\perp} p_a$ is the projection of the gradient onto the normal direction

$$\nabla^{\perp} p_a = \text{proj}_{\mathbf{n}}(\nabla p_a) = \mathbf{n}\frac{\mathbf{n}^{\top}\nabla p_a}{\mathbf{n}^{\top}\mathbf{n}} = \mathbf{n}\partial_{\mathbf{n}} p_a. \tag{4.31}$$

From the knowledge that the IFD only contains the tangential information and that the normal derivative is described by the BE discretization the discrete form of (4.30) can be written as

$$\nabla \mathbf{p}_a = \begin{bmatrix}\mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_z\end{bmatrix} \mathbf{p}_a + \begin{bmatrix}\text{diag}(\mathbf{n}_x) \\ \text{diag}(\mathbf{n}_y) \\ \text{diag}(\mathbf{n}_z)\end{bmatrix} \partial_{\mathbf{n}} \mathbf{p}_a. \tag{4.32}$$

The observant reader will have realized that this is, in fact, equivalent to changing the basis of the local gradient in (4.24) back to the global basis. This can be easily verified as

$$\nabla \mathbf{p}_a = ({}_{\ell}\mathbf{M}_e)^{\top} \left(\begin{bmatrix}\mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{0}\end{bmatrix} \mathbf{p}_a + \begin{bmatrix}\mathbf{0} \\ \mathbf{0} \\ \mathbf{I}\end{bmatrix} \partial_{\mathbf{n}} \mathbf{p}_a \right) = \begin{bmatrix}\mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_z\end{bmatrix} \mathbf{p}_a + \begin{bmatrix}\text{diag}(\mathbf{n}_x) \\ \text{diag}(\mathbf{n}_y) \\ \text{diag}(\mathbf{n}_z)\end{bmatrix} \partial_{\mathbf{n}} \mathbf{p}_a, \tag{4.33}$$

where the last equality comes directly from the fact that the change of basis is orthogonal. For the divergence a similar realization to that of the gradient is needed. That is, the total divergence is equal to the sum of the tangential and normal divergence. Written out, this means that

$$\nabla_{\mathbf{x}} \cdot \mathbf{v}_v = \nabla^{\parallel}_{\mathbf{x}} \cdot \mathbf{v}_v + \nabla^{\perp}_{\mathbf{x}} \cdot \mathbf{v}_v. \tag{4.34}$$

As such, the discrete form of divergence of $\mathbf{v}_v$ can be computed as

$$\nabla \cdot \mathbf{v} = \begin{bmatrix}\mathbf{D}_x & \mathbf{D}_y & \mathbf{D}_z\end{bmatrix} \mathbf{v} + \begin{bmatrix}\text{diag}(\mathbf{n}_x) & \text{diag}(\mathbf{n}_y) & \text{diag}(\mathbf{n}_z)\end{bmatrix} \partial_{\mathbf{n}} \mathbf{v}. \tag{4.35}$$

It turns out that the above is equivalent to (4.27). The reason for this is firstly that

$$\begin{bmatrix}\mathbf{D}_t & \mathbf{D}_s & \mathbf{0}\end{bmatrix} \mathbf{v}_{\ell} = \begin{bmatrix}\mathbf{D}_x & \mathbf{D}_y & \mathbf{D}_z\end{bmatrix} ({}_{\ell}\mathbf{M}_e)^{\top} {}_{\ell}\mathbf{M}_e \mathbf{v} = \begin{bmatrix}\mathbf{D}_x & \mathbf{D}_y & \mathbf{D}_z\end{bmatrix} \mathbf{v}, \tag{4.36}$$

where the last equality is obtained using the orthogonality of ${}_{\ell}\mathbf{M}_e$, and secondly that

$$\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix} {}_{\ell}\mathbf{M}_e \partial_{\mathbf{n}} \mathbf{v} = \begin{bmatrix}\text{diag}(\mathbf{n}_x) & \text{diag}(\mathbf{n}_y) & \text{diag}(\mathbf{n}_z)\end{bmatrix} \partial_{\mathbf{n}} \mathbf{v}. \tag{4.37}$$

As such, it can be seen that (4.27) and (4.35) are, in fact, the same expression. Although this equivalence at first glance might seem strange, it is not. It is simply the discrete form of an old and well-known fact: The divergence is invariant with respect to an orthogonal change-of-basis. Finally, using the discrete forms of the gradients and null-divergence (4.13) can be written as

$$\begin{bmatrix} \mathbf{H}_a & \mathbf{G}_a & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_h & \mathbf{G}_h & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_v & \mathbf{G}_v \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_r & \mathbf{N}^{\top} \\ \tau_a\mathbf{I} & \mathbf{0} & \tau_h\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \phi_a\mathbf{D}_c & \phi_a\mathbf{N} & \phi_h\mathbf{D}_c & \phi_h\mathbf{N} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \partial_{\mathbf{n}}\mathbf{p}_a \\ \mathbf{p}_h \\ \partial_{\mathbf{n}}\mathbf{p}_h \\ \mathbf{v} \\ \partial_{\mathbf{n}}\mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_s \end{bmatrix}, \tag{4.38}$$

where the following notation were introduced

$$\mathbf{N} = \begin{bmatrix}\text{diag}(\mathbf{n}_x) & \text{diag}(\mathbf{n}_y) & \text{diag}(\mathbf{n}_z)\end{bmatrix}^\top, \quad \mathbf{D}_c = \begin{bmatrix}\mathbf{D}_x^\top & \mathbf{D}_y^\top & \mathbf{D}_z^\top\end{bmatrix}^\top, \quad \mathbf{D}_r = \begin{bmatrix}\mathbf{D}_x & \mathbf{D}_y & \mathbf{D}_z\end{bmatrix}. \tag{4.39}$$

As this linear system has 10 DOF pr. collocation node, it becomes problematic to solve directly for even moderately large problems on a desktop machine. Fortunately, the system itself has *structure* which can be utilized. In particular, all the matrices, except $\mathbf{G}_a$ and $\mathbf{H}_a$, are extremely sparse (Figure 4.3). However, due to the ill-conditioning of the system described in (4.38), the structure cannot be easily utilized. The following section explains how the system can be condensed to a well-conditioned linear system with only 1 DOF pr. collocation node.

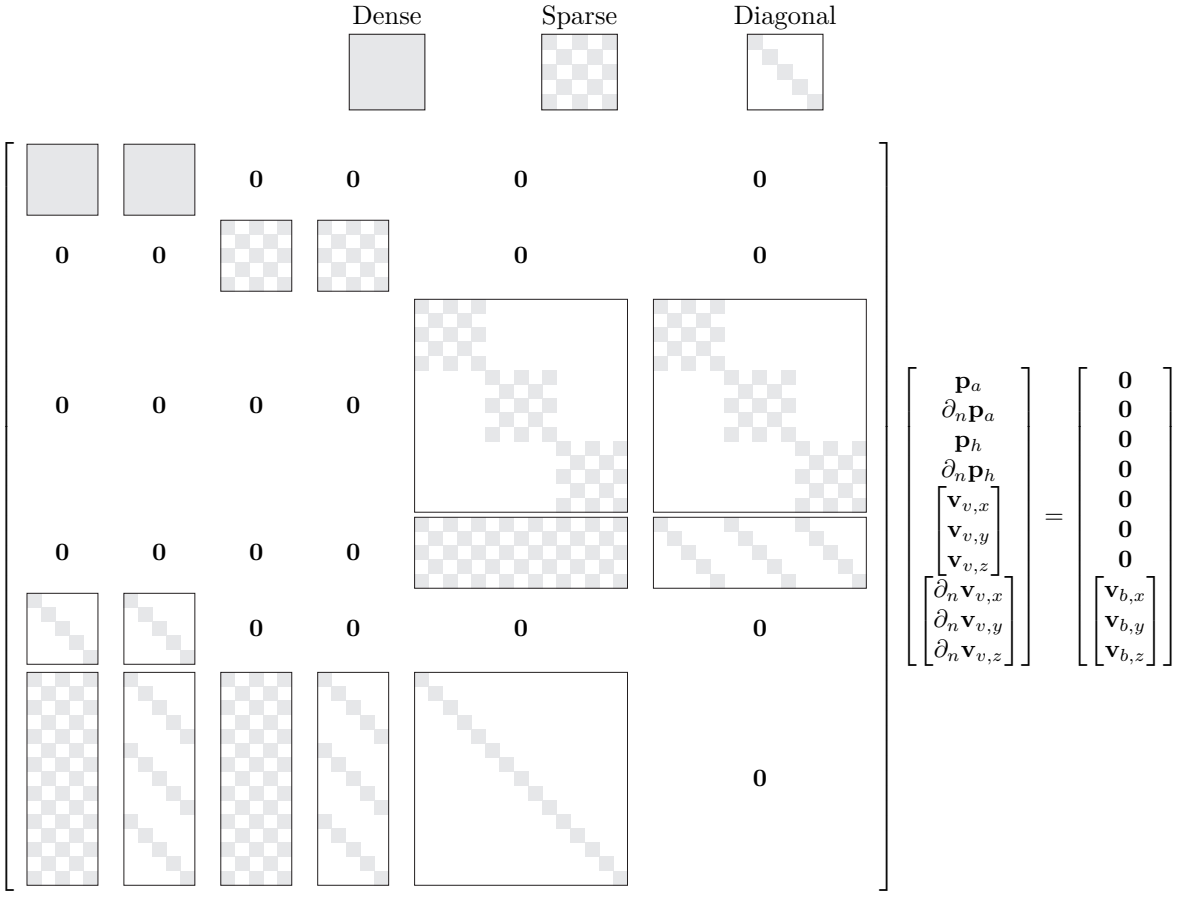


Figure 4.3: Visualization of the sparsity structure of (4.38). Figure originally appeared as an equation in Paper J2.

4.3.1 Condensation

It turns out that the system of (4.38) is ill-conditioned as is. As such, solving it through a conventional iterative solution scheme does not work. To resolve this, a Schur complement preconditioning step is used to reduce to a linear system that only has $\mathbf{p}_a$ as a DOF. The resulting system is well-conditioned and can be solved using an iterative solution scheme. A brief derivation similar to that found in Paper J2 is restated here.

Due to the simplicity of the isothermal boundary condition described in (4.9) the thermal mode can easily be removed using that

$$\mathbf{p}_h = -\frac{\tau_a}{\tau_h}\mathbf{p}_a, \quad \partial_{\mathbf{n}} p_h = \frac{\tau_a}{\tau_h}\mathbf{G}_h^{-1}\mathbf{H}_h\mathbf{p}_a. \tag{4.40}$$

As such (4.38) can be reduced to

$$\begin{bmatrix} \mathbf{H}_a & \mathbf{G}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_v & \mathbf{G}_v \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_r & \mathbf{N}^\top \\ \mu_a\mathbf{D}_c + \mu_h\mathbf{N}\mathbf{G}_h^{-1}\mathbf{H}_h & \phi_a\mathbf{N} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \partial_{\mathbf{n}}\mathbf{p}_a \\ \mathbf{v} \\ \partial_{\mathbf{n}}\mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_s \end{bmatrix}, \tag{4.41}$$

where the following two constants are introduced to ease the notation

$$\mu_a = \phi_a - \frac{\tau_a\phi_h}{\tau_h}, \quad \mu_h = \frac{\tau_a\phi_h}{\tau_h}. \tag{4.42}$$

The normal components of $\mathbf{p}_a$ and $\mathbf{v}$ can further be removed using the discretized integral equations

$$\partial_{\mathbf{n}}\mathbf{p}_a = -\mathbf{G}_a^{-1}\mathbf{H}_a\mathbf{p}_a, \tag{4.43}$$

$$\partial_{\mathbf{n}}\mathbf{v} = -\mathbf{G}_v^{-1}\mathbf{H}_v\mathbf{v}. \tag{4.44}$$

Using the above (4.41) can be reduced to

$$\begin{bmatrix} \mathbf{0} & \mathbf{D}_r - \mathbf{N}^\top\mathbf{G}_v^{-1}\mathbf{H}_v \\ \mu_a\mathbf{D}_c + \mathbf{N}\left(\mu_h\mathbf{G}_h^{-1}\mathbf{H}_h - \phi_a\mathbf{G}_a^{-1}\mathbf{H}_a\right) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_s - \phi_a\mathbf{N}\mathbf{G}_a^{-1}\mathbf{p}_{\text{inc}} \end{bmatrix}. \tag{4.45}$$

Finally, the viscous mode can be completely removed using that

$$\mathbf{v} = \mathbf{v}_s - \left(\mu_a\mathbf{D}_c + \mu_h\mathbf{N}\mathbf{G}_h^{-1}\mathbf{H}_h - \phi_a\mathbf{N}\mathbf{G}_a^{-1}\mathbf{H}_a\right)\mathbf{p}_a. \tag{4.46}$$

To ease the notation the following shorthand is introduced

$$\mathbf{R} = \mathbf{D}_r - \mathbf{N}^\top\mathbf{G}_v^{-1}\mathbf{H}_v. \tag{4.47}$$

As such (4.45) simplifies to

$$\mathbf{R}\left(\mu_a\mathbf{D}_c + \mathbf{N}\left(\mu_h\mathbf{G}_h^{-1}\mathbf{H}_h - \phi_a\mathbf{G}_a^{-1}\mathbf{H}_a\right)\right)\mathbf{p}_a = \mathbf{R}\mathbf{v}_s. \tag{4.48}$$

From a numerical point of view, the above is inconvenient as it requires the inverse of the dense matrix $\mathbf{G}_a$. To remove this requirement the above is left multiplied with

$$\mathbf{G}_a\left(\mathbf{R}\mathbf{N}\right)^{-1}, \tag{4.49}$$

which result in

$$\left[\mathbf{G}_a\left(\mu_a\left(\mathbf{R}\mathbf{N}\right)^{-1}\mathbf{R}\mathbf{D}_c + \mu_h\mathbf{G}_h^{-1}\mathbf{H}_h\right) - \phi_a\mathbf{H}_a\right]\mathbf{p}_a = \mathbf{G}_a\left(\mathbf{R}\mathbf{N}\right)^{-1}\mathbf{R}\mathbf{v}_s. \tag{4.50}$$

As the aim is to solve large problems, we need to be able to solve (4.50) using an iterative scheme. To do so, we must be able to define multiplication with the linear map corresponding to the system matrix. Doing so is fairly straightforward, with the caveat that each multiplication actually requires the solution of additional linear systems. The reason for this is that we need to multiply by both $\mathbf{G}_h^{-1}$, $\mathbf{G}_v^{-1}$, and $\left(\mathbf{R}\mathbf{N}\right)^{-1}$. Given that $\mathbf{G}_h^{-1}$ and $\mathbf{G}_v^{-1}$ are sparse and well-conditioned, they can be applied using a sparse LU-factorization or another iterative scheme. For multiplication with $\left(\mathbf{R}\mathbf{N}\right)^{-1}$, the only option is to use an iterative scheme, as there is no guarantee that the matrix is sparse. The full efficient solution scheme of (4.50) can be found in Algorithm 1.

Algorithm 1 Efficient solution scheme for (4.50).

Require: $\mathbf{G}_a, \mathbf{H}_a, \mathbf{G}_h, \mathbf{H}_h, \mathbf{G}_v, \mathbf{H}_v, \mathbf{D}_r, \mathbf{D}_c, \mathbf{N}, \mu_a, \mu_h, \phi_a$
  procedure MUL_R($\mathbf{z}$)
    return $\mathbf{D}_r\mathbf{z} - \mathbf{N}^\top$ (gmres($\mathbf{G}_v, \mathbf{H}_v\mathbf{z}$))
  end procedure
  procedure MUL_RN($\mathbf{z}$)
    return MUL_R($\mathbf{Nz}$)
  end procedure
  procedure MUL_A($\mathbf{z}$)
    return $\mathbf{G}_a$ (gmres(MUL_RN, MUL_R($\mathbf{D}_c\mathbf{z}$))μ_a + gmres($\mathbf{G}_h, \mathbf{H}_v\mathbf{z}$)) $\mu_h - \phi_a\mathbf{H}_a\mathbf{z}$
  end procedure
  Compute right-hand side: $\mathbf{b} = \mathbf{G}_a$ (gmres(MUL_RN, MUL_R($\mathbf{v}_b$)))
  Solve the linear system: gmres(MUL_A,$\mathbf{b}$)

While the proposed scheme of Algorithm 1 solves some issues regarding the computational scaling, most notably making it possible to utilize the sparsity of the thermal and viscous modes in the solution phase, it does not resolve all problems. In fact, Figure 4.4 shows the computational complexity of the large $10n \times 10n$ system (Figure 4.3) and the $n \times n$ system (Algorithm 1). In particular, we see significant improvements in computational time using the $n \times n$ formulation. Unfortunately, the two dense matrices from the acoustical mode can be seen to ruin the memory scaling of the computation. In the following section, we explain how we in Paper J3 resolved this issue by applying standard the acceleration techniques for boundary element methods.

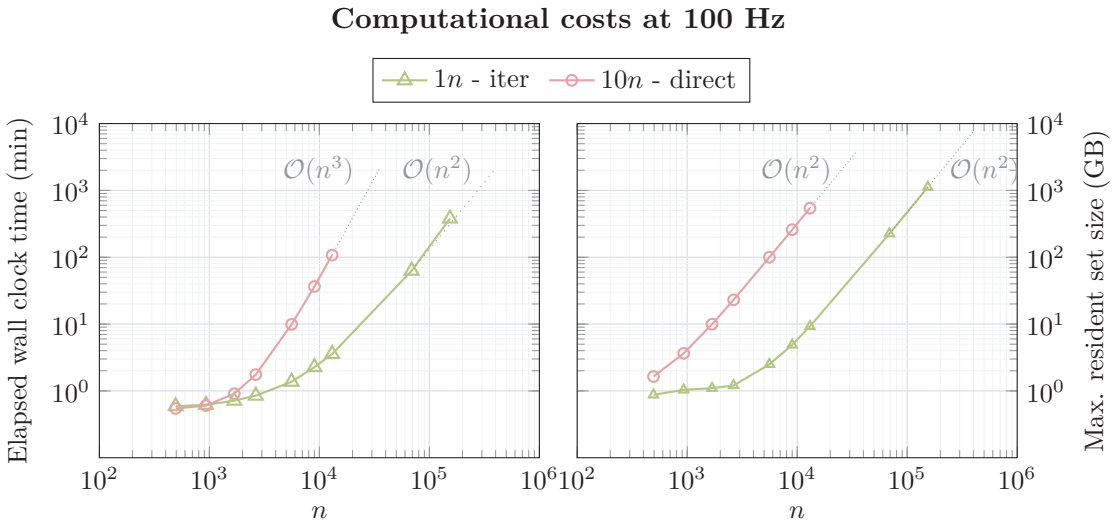


Figure 4.4: Complexity of $10n \times 10n$ system and the $1n \times 1n$ system using either dense matrices and the fast multipole operators. The simulation performed is that of an oscillating sphere in a viscous fluid.

4.4 Taking things Large-scale (Paper J3)

Getting rid of the $O(n^2)$ complexity of both memory and computational time of (4.50) can be achieved utilizing the fast multipole method or the $\mathcal{H}$-matrix approach for the dense part of the computation, i.e. for approximating $\mathbf{G}_a$ and $\mathbf{H}_a$. We find that for memory-constrained hardware the fast multipole method is the preferred approach, and reversely the $\mathcal{H}$-approach is preferred when memory is a non-issue (Figure 4.5). This should not come as a surprise, since the added memory is due to more information

being stored of the dense matrices, resulting in less on-the-fly computation, in turn reducing the computational efforts. In Figure 4.5 we repeat the tests of Figure 4.4 using both the fast multipole method and $\mathcal{H}$-matrices to approximate the acoustical mode. From the figure it is clear that the scalability issues have been resolved.

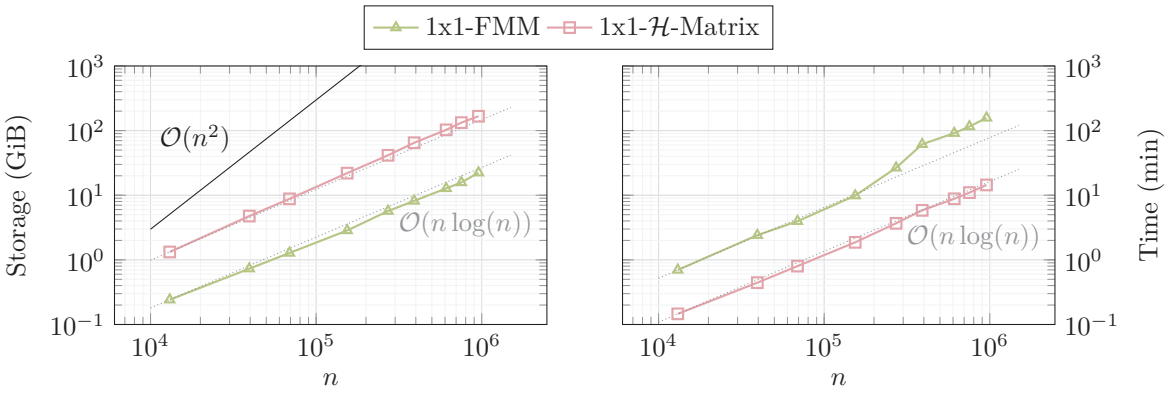


Figure 4.5: *Figure originally appeared in Paper J3.* Left: Complexity of storing all matrices plus performing a single call of `MUL_A` in Algorithm 1. The reason for this addition is that the acceleration techniques include intermediate steps that allocate memory when performing multiplications. Right: Complexity of the total solution time. This includes the assembly of the matrices and the reconstruction of the additional variables ($\partial_{\mathbf{n}}\mathbf{p}_a$, $\mathbf{p}_h$, $\partial_{\mathbf{n}}\mathbf{p}_h$, $\mathbf{v}_v$ & $\partial_{\mathbf{n}}\mathbf{v}_v$).

We note that results seen in Figure 4.5 where all performed using the Julia programming language and the High-Performance Computing (HPC) resources provided by the Technical University of Denmark [17, 16, 31]. The dense linear algebra was handled using LAPACK/BLAS [3, 18] while the sparse computation was handled using SuiteSparse [26, 29, 30].

4.5 Contributions

The papers J2, J3, and C2 contain the methods described in this chapter and further verify the formulations using numerical test setups. The contributions contained in Paper J2 are:

- We derived a new formulation for boundary element simulations including viscous and thermal losses that does not directly rely on the tangential directions. The new formulation is based on two simple facts: First, the gradient can be split into a normal part and a tangential part, and second, that the interpolation function derivatives only give the tangential part of the derivatives. Combining the two facts then makes it possible to get full gradient by adding the boundary element discretization of the normal derivative (containing the normal part of the gradient) and the interpolation function derivatives (containing the tangential part of the gradient).
- We show how the new formulation relates to the old state-of-the-art formulation. In particular, we show that the difference in the no-slip condition between the two formulations is a simple change-of-coordinates and that the null-divergence constraint is equivalent between the two formulations.
- Using the new formulation, we derive an iterative solution strategy that employs nested iterative schemes. This new scheme makes it possible to take advantage of the sparse matrices from the thermal and viscous modes in the solution phase. This is in contrast to previous attempts where the sparsity could only be utilized in the assembly phase. As a result, the solution scheme is significantly more efficient than the original formulation.

- Finally, we show that the new formulation makes it possible to solve problems larger than previously possible. In fact, most of the time is spent reconstructing the normal derivative of the acoustical pressure. As such, a key takeaway is that if it is possible for a specific mesh size to solve a pure acoustical system with Dirichlet boundary conditions, then it should be possible to solve the lossy system for the same mesh/setup.
- Unsurprisingly, however, the formulation is found to still suffer from a computational bottleneck stemming from the two dense matrices originating from the acoustical mode.

The contributions contained in paper J3 are:

- We relieve the computational bottleneck found in paper J2 by utilizing both the fast multipole method and $\mathcal{H}$-matrices. Whether to use one or the other is found to depend on the computational constraints of the chosen hardware. If memory is a problem, then the fast multipole method is favorable, while if memory is not an issue, then the $\mathcal{H}$-matrix approach is favorable.
- We noted that a possible middle ground between memory and computational constraints could be the interpolated factored Green's function (IFGF) as it requires significantly less memory to store than that of a typical $\mathcal{H}$-matrix while possibly being faster than the fast multipole method. At the same time, the IFGF method is known to be easily scaled up to multicore systems.
- Furthermore, it is important to note that parts of the implemented method contain a simplification of the sparsity patterns of the viscous and thermal matrices (Figure 3.4) resulting in a lack of robustness in the implementation, which could result in failures for geometries with overlapping boundary layers.

The contributions contained in paper C2 are:

- A description of an open-source library, written in the Julia programming language, that implements the ideas presented in both Paper J2 and Paper J3. Due to the length limitations of the conference contributions, Paper C2 did not include the ideas of Paper J1, even though the software includes partial support of the method.
- A validation of the software using three test cases: Scattering of a rigid sphere, a plane wave in a duct, and an oscillating sphere in a viscous fluid. The latter test case is equivalent to what was used in Papers J2 and J3. However, the mesh used in Paper C2 is of significantly smaller size.
- The hope is that the software will make the proposed models more accessible for researchers as well as practitioners.

All in all, the contributions serve as important steps in the direction of large-scale boundary element simulations including viscous and thermal losses. A key takeaway is that the computational complexity of the computations including viscous and thermal losses behaves similarly to the computational complexity of the pure acoustical computation.

CHAPTER 5

Reduced Order Boundary Element Models

The frequency-dependent integrals that lays the foundation of the boundary element method make the method infeasible for multifrequency analysis. Several solutions based on making the integrals frequency independent through approximations of the Green's function has been proposed [52, 66, 69, 75]. These techniques differ from other techniques used primarily in finite element analysis, where the frequency response itself is approximated [27, 44, 46, 49, 56]. In addition, there have also been works on approximating the system itself using series expansions [10, 11]. In this chapter, we explain how the first mentioned methods can be extended to handle the boundary layer impedance boundary condition.

5.1 State-of-the-art

In [66] the BE integrals of (2.19) is made frequency independent by utilizing a Taylor expansion of w.r.t the wavenumber for the Green's function and its normal derivative

$$G(\mathbf{x},\mathbf{y}) \approx \sum_{m=0}^{N_v-1} \frac{(k-k_0)^m}{m!}\left[G^{(m)}(\mathbf{x},\mathbf{y},k_0)\right], \quad \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} \approx \sum_{m=0}^{N_p-1} \frac{(k-k_0)^m}{m!}\left[\mathbf{n}(\mathbf{y})^\top \nabla G^{(m)}(\mathbf{x},\mathbf{y},k_0)\right],$$

where N_v and N_p is the number of terms included in the Taylor expansion of respectively the Green's function and its normal derivative while k_0 is the expansion wavenumber. For simplification purposes, it is in the following chosen to set $N_p = N_v = M$. Inserting the above into the Kirchhoff-Helmholtz integral equation the following system of equations will appear

$$\left(\alpha \text{diag}(\boldsymbol{\zeta}) - \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!}\mathbf{F}_m(k_0)\right)\mathbf{p} + s(\alpha)k\left(\sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!}\mathbf{G}_m(k_0)\right)\mathbf{v}_n \approx \mathbf{0}, \tag{5.1}$$

which it sometimes referred to as the series expansion boundary element method or SEBEM for short. The important aspect of (5.1) is that the two series of matrices, $\mathbf{F}_m$ and $\mathbf{G}_m$, depend only on the expansion wavenumber k_0 and can therefore be computed in a so-called offline stage. After applying boundary conditions (5.1) will have the form of

$$\left(\sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!}\mathbf{A}_m(k_0,\alpha)\right)\mathbf{z} \approx \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!}\mathbf{b}_m(k_0,\alpha), \tag{5.2}$$

where $\mathbf{z}$ is a vector containing nodal values of either $\mathbf{p}$ or $\mathbf{v}_n$. Unfortunately, (5.2) does not resolve all the issues, as it only reduces the assembly from a discretization of boundary integrals to a sum of matrices that still scales as $O(n^2)$ in both computation and storage. Additionally, for large systems, the time spent solving the linear system of equations is overshadowed by the time spent assembling the linear system, making the gain of using (5.2) insignificant with respect to the total computational time.

In order to reduce both the memory footprint and the assembly time of (5.2) a Galerkin projection can be applied. The main assumption behind this projection is that the solution, as well as the right-hand side of (5.2) is spanned by a lower dimensional subspace. This means that

$$\mathbf{z} \approx \mathbf{U}_\ell \mathbf{z}_\ell, \quad \mathbf{b}_m \approx \mathbf{U}_\ell \mathbf{b}_{\ell m}, \quad \mathbf{U}_\ell, \in \mathbb{C}^{N\times \ell} \tag{5.3}$$

will be good approximations. For simplification purposes, it is often chosen to let $\mathbf{U}_\ell$ be unitary. An intuitive explanation as to why this must be a good approximation is that most BE systems are well-conditioned and iterative solvers such as the generalized minimal residual method (GMRes) converge quickly. In addition, this motivates a good place to search for both $\mathbf{U}_\ell$ is the ℓ-Krylov subspace defined by [45]

$$\mathcal{K}_\ell\left(\mathbf{A}(k), \mathbf{b}(k)\right) = \text{span}\left\{\mathbf{b}(k),\ \mathbf{A}(k)\mathbf{b}(k),\ \mathbf{A}(k)^2\mathbf{b}(k),\ \ldots,\ \mathbf{A}(k)^{\ell-1}\mathbf{b}(k)\right\}. \tag{5.4}$$

Setting the columns of $\mathbf{U}_\ell$ equal to the Krylov vectors would result in (5.3) being a good approximation for wavenumbers close to k. The subspace can be made to handle a wider range of wavenumbers by combining the Krylov subspace for the wavenumbers $k_1, k_2, \ldots k_L$. As such the final projection matrix, $\mathbf{U}_\ell$, is computed as the singular value decomposition (SVD) of the concatenation of the columns of all the q-Krylov vector subspaces [37, 67]

$$\mathbf{U}_\ell \Sigma_\ell \mathbf{V}_\ell^{\mathsf{H}} = \text{svd}\left(\begin{bmatrix}\mathbf{K}_{k_1} & \mathbf{K}_{k_2} & \ldots & \mathbf{K}_{k_L}\end{bmatrix}\right), \tag{5.5}$$

where $\mathbf{K}_{k_i}$ denotes a matrix with columns equal to some Krylov vectors at wavenumber k_i. Now inserting the approximations introduced in (5.3) into (5.2) while also multiplying both sides with the hermitian transpose of $\mathbf{U}_\ell$ from the left it follows that

$$\left(\sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!}\mathbf{U}_\ell^{\mathsf{H}}\mathbf{A}_m(k_0,\alpha)\mathbf{U}_\ell\right)\mathbf{z}_\ell \approx \sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\mathbf{U}_\ell^{\mathsf{H}}\mathbf{U}_\ell\mathbf{b}_{\ell m}(k_0,\alpha). \tag{5.6}$$

Using that $\mathbf{U}_\ell$ is unitary while defining $\mathbf{A}_{\ell m} = \mathbf{U}_\ell^{\mathsf{H}}\mathbf{A}_m(k_0,\alpha)\mathbf{U}_\ell$ the above becomes

$$\left(\sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!}\mathbf{A}_{\ell m}(k_0,\alpha)\right)\mathbf{z}_\ell \approx \sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\mathbf{b}_{\ell m}(k_0,\alpha). \tag{5.7}$$

The important part of the above is that $\mathbf{A}_{\ell m}(k_0,\alpha) \in \mathbb{C}^{\ell\times\ell}$ and $\mathbf{b}_{\ell m}(k_0,\alpha) \in \mathbb{C}^\ell$, which means that if $\ell \ll n$ this representation requires significantly less memory than the original system. Note, however, that $\mathbf{A}_m(k_0,\alpha)$ should never explicitly be stored. Instead, the rows should be projected during the assembly phase, as described in Algorithm 2.

Algorithm 2 Assembly of the ROSEBEM

Require: $\mathbf{U}_\ell \in \mathbb{C}^{n\times\ell}, k_0 \in \mathbb{R}, M \in \mathbb{N}$
  Preallocate $\mathbf{A}, \in \mathbb{C}^{n\times\ell}$, $\mathbf{b} \in \mathbb{C}^n$
  for $m = 0 : M-1$ **do**
    for $n = 1 : N$ **do**
      Assemble i^{th} row of $\mathbf{A}_m(k_0)$ and $\mathbf{b}_m(k_0)$ (denoted by $\mathbf{a}_m^i$ and b_m^i)
      Save right-hand side: $b_i \leftarrow b_m^i$
      Compress columns: $\mathbf{A}_{[i,:]} \leftarrow \mathbf{a}_m^i\mathbf{U}_\ell$
    end for
    Compress rows: $\mathbf{A}_{\ell m}(k_0) \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{A}$
    Compress rows: $\mathbf{b}_{\ell m}(k_0) \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{b}$
  end for

5.2 The Boundary Layer Impedance Condition

We now leave the realm of series expansions and reduced basis and instead focus on how to include (approximate) viscous and thermal losses using specialized boundary conditions. In particular, we look at the so-called boundary layer impedance (BLI) boundary condition described in [15]. This approximation is derived using a few assumptions such as e.g. low curvature and non-overlapping boundary layers. As introduced in [1] the BLI condition can be written as

$$\frac{\partial p}{\partial \mathbf{n}}(\mathbf{y}) = \left[(\gamma-1)\frac{\mathrm{i}k^2}{k_h} - \frac{\mathrm{i}\Delta^{\parallel}}{k_v}\right]p(\mathbf{y}), \quad \mathbf{y}\in\Gamma \tag{5.8}$$

where $\Delta^{\parallel}$ is the tangential Laplacian, γ is the ratio of specific heat and k, k_v, k_h are respectively the isentropic, viscous and thermal wavenumbers. The two additional wavenumbers are calculated using the physical properties of the fluid, such as thermal conductivity, specific heat capacity under constant pressure, and shear and bulk viscosity coefficients [20].

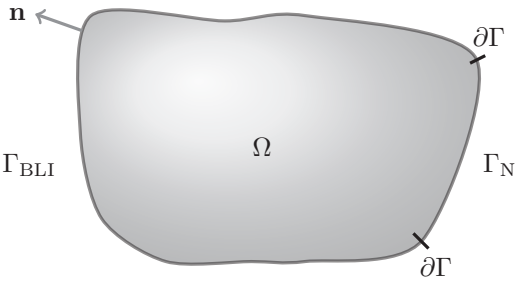


Figure 5.1: *Figure originally appeared in Paper J1.* Domain of an interior problem.

For the sake of simplicity the following derivation is done in the case where the boundary (Γ) is split into a part applied with the BLI condition (Γ_{BLI} in Figure 5.1) and a part applied a Neumann condition (Γ_{N} in Figure 5.1). The reason for this simplification is that it represents the numerical cases investigated in Paper J1. Inserting (5.8) into the boundary integral equation of (2.19)

$$\begin{aligned}\zeta(\mathbf{x})p(\mathbf{x}) = &\int_{\Gamma_{\mathrm{N}}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\,\mathrm{d}S_{\mathbf{y}} - \int_{\Gamma}\frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}p(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}}\\ &+\int_{\Gamma_{\mathrm{BLI}}} G(\mathbf{x},\mathbf{y})\left[\frac{(\gamma-1)\mathrm{i}k^2}{k_h} - \frac{\mathrm{i}\Delta^{\parallel}}{k_v}\right]p(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}}.\end{aligned} \tag{5.9}$$

Applying integration by parts to the term including $\Delta^{\parallel}$ in (5.9) while assuming [55]

$$\int_{\partial\Gamma_{\mathrm{BLI}}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\,\mathrm{d}S_{\mathbf{y}} = 0, \tag{5.10}$$

it follows that

$$\begin{aligned}\zeta(\mathbf{x})p(\mathbf{x}) = &\int_{\Gamma_{\mathrm{N}}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\,\mathrm{d}S_{\mathbf{y}} - \int_{\Gamma}\frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}p(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}}\\ &+\frac{(\gamma-1)\mathrm{i}k^2}{k_h}\int_{\Gamma_{\mathrm{BLI}}} G(\mathbf{x},\mathbf{y})p(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}} + \frac{\mathrm{i}}{k_v}\int_{\Gamma_{\mathrm{BLI}}}\nabla^{\parallel}_{\mathbf{y}}G(\mathbf{x},\mathbf{y})^{\top}\nabla^{\parallel}_{\mathbf{y}}p(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}},\end{aligned} \tag{5.11}$$

where

$$\nabla^{\parallel}_{\mathbf{y}}G(\mathbf{x},\mathbf{y}) = \mathrm{oproj}_{\mathbf{n}}\left(\nabla_{\mathbf{y}}G(\mathbf{x},\mathbf{y})\right) = (\mathbf{I}-\mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y})^{\top})\nabla_{\mathbf{y}}G(\mathbf{x},\mathbf{y}), \tag{5.12}$$

is the tangential gradient of the Green's function. Lastly since $\nabla^{\parallel}_{\mathbf{y}}p(\mathbf{y})$ is the tangential gradient of the pressure it can be computed directly from the BEM discretization using the interpolation function derivatives described in subsection 4.1.4.

5.3 New developments

In contrast to previous works the boundary layer impedance formulation described in (5.11) requires the Taylor series expansion of the tangential gradient of the Green's function. Thankfully, using (5.12), the Taylor expansion of the tangential gradient can easily be obtained as

$$\nabla^{\parallel}_{\mathbf{y}}G(\mathbf{x},\mathbf{y}) \approx \sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\left[\left(\mathbf{I}-\mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y})^{\top}\right)\nabla G^{(m)}(\mathbf{x},\mathbf{y},k_0)\right], \tag{5.13}$$

Now inserting the different Taylor series into (5.11) the expression reduces to

$$\begin{aligned}\operatorname{diag}(\boldsymbol{\zeta})\mathbf{p} \approx &\left(\sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\mathbf{G}_m(k_0)\right)\frac{\partial \mathbf{p}}{\partial \mathbf{n}} + \left(\sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\mathbf{F}_m(k_0)\right)\mathbf{p}\\ &+\frac{(\gamma-1)\mathrm{i}k^2}{k_h}\left(\sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\mathbf{H}_m(k_0)\right)\mathbf{p} + \frac{\mathrm{i}}{k_v}\left(\sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\mathbf{T}_m(k_0)\right)\mathbf{p}.\end{aligned} \tag{5.14}$$

We now again assume that the solution is spanned by lower-dimensional subspace i.e. that the following are good approximations

$$\mathbf{p} \approx \mathbf{U}_\ell \mathbf{p}_\ell, \quad \frac{\partial \mathbf{p}}{\partial \mathbf{n}} \approx \mathbf{U}_\ell \frac{\partial \mathbf{p}_\ell}{\partial \mathbf{n}}. \tag{5.15}$$

Again $\mathbf{U}_\ell$ is computed from the ℓ-Krylov subspace which in this case comes from the linear system generated after applying boundary conditions to the full BLI equations. Now collecting terms in (5.14), adding $\operatorname{diag}(\boldsymbol{\zeta})$ to $\mathbf{F}_0$ and then using the above introduces approximations it follows that

$$\sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\left(\left[\mathbf{F}_m(k_0)+\frac{(\gamma-1)\mathrm{i}k^2}{k_h}\mathbf{H}_m(k_0)+\frac{\mathrm{i}}{k_v}\mathbf{T}_m(k_0)\right]\mathbf{U}_\ell\mathbf{p}_\ell + \mathbf{G}_m(k_0)\mathbf{U}_\ell\frac{\partial \mathbf{p}_\ell}{\partial \mathbf{n}}\right) \approx \mathbf{0}. \tag{5.16}$$

Finally, multiplying $\mathbf{U}_\ell^{\mathsf{H}}$ from the left followed by moving $\mathbf{U}_\ell$ (and $\mathbf{U}_\ell^{\mathsf{H}}$) inside the parentheses followed by collecting terms we reach the final expression of

$$\sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\left(\left[\mathbf{F}_{\ell m}(k_0)+\frac{(\gamma-1)\mathrm{i}k^2}{k_h}\mathbf{H}_{\ell m}(k_0)+\frac{\mathrm{i}}{k_v}\mathbf{T}_{\ell m}(k_0)\right]\mathbf{p}_\ell + \mathbf{G}_{\ell m}(k_0)\frac{\partial \mathbf{p}_\ell}{\partial \mathbf{n}}\right) \approx \mathbf{0}, \tag{5.17}$$

where

$$\begin{aligned}\mathbf{F}_{\ell m}(k_0) &= \mathbf{U}_\ell^{\mathsf{H}}\mathbf{F}_m(k_0)\mathbf{U}_\ell, \quad \mathbf{H}_{\ell m}(k_0) = \mathbf{U}_\ell^{\mathsf{H}}\mathbf{H}_m(k_0)\mathbf{U}_\ell,\\ \mathbf{T}_{\ell m}(k_0) &= \mathbf{U}_\ell^{\mathsf{H}}\mathbf{T}_m(k_0)\mathbf{U}_\ell, \quad \mathbf{G}_{\ell m}(k_0) = \mathbf{U}_\ell^{\mathsf{H}}\mathbf{G}_m(k_0)\mathbf{U}_\ell,\end{aligned}$$

with all the matrices being of size $\ell \times \ell$. This approach is reasonable storage wise if $\ell^2 M \lessapprox n^2$, such that storing all the SEBEM matrices is less than or of the order of storing a single of the original BEM matrices. Additionally, if $\ell \ll n$ solving the resulting linear system of (5.17) will be significantly faster than solving the original system.

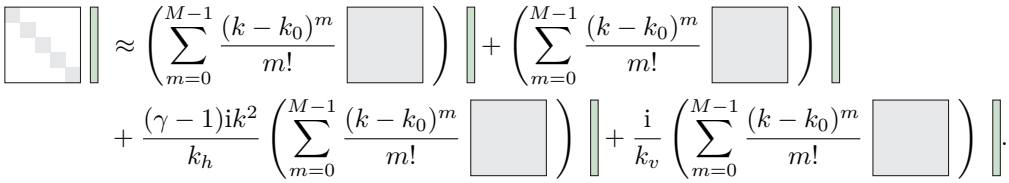
Algorithm 3 Assembly of the ROSEBEM incorporating the BLI boundary condition

Require: $\mathbf{U}_\ell \in \mathbb{C}^{n\times\ell}, k_0 \in \mathbb{R}, M \in \mathbb{N}$
  Preallocate $\mathbf{Z}, \mathbf{F}, \mathbf{H}, \mathbf{T} \in \mathbb{C}^{n\times\ell}$, $\mathbf{b} \in \mathbb{C}^n$
  for $m = 0 : M-1$ **do**
    for $n = 1 : N$ **do**
      if $m = 0$ **then**
        Assemble integral free term for the n_{th} collocation point (denoted by ζ_n)
        Scale row of $\mathbf{U}_\ell$: $\mathbf{Z}_{[n,:]} \leftarrow \zeta_n \mathbf{U}_{\ell,[n,:]}$
      end if
      Assemble i^{th} row of $\mathbf{F}_m(k_0)$, $\mathbf{H}_m(k_0)$, $\mathbf{T}_m(k_0)$ and $\mathbf{b}_m(k_0)$ (denoted by $\mathbf{f}_m^i$, $\mathbf{h}_m^i$, $\mathbf{t}_m^i$, b_m^i)
      Save right-hand side: $b_i \leftarrow b_m^i$
      Compress columns: $\mathbf{F}_{[i,:]} \leftarrow \mathbf{f}_m^i\mathbf{U}_\ell$, $\mathbf{H}_{[i,:]} \leftarrow \mathbf{h}_m^i\mathbf{U}_\ell$, $\mathbf{T}_{[i,:]} \leftarrow \mathbf{t}_m^i\mathbf{U}_\ell$
    end for
    Compress rows: $\mathbf{F}_{\ell m}(k_0) \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{F}$, $\mathbf{H}_{\ell m} \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{H}$, $\mathbf{T}_{\ell m} \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{T}$
    Compress rows: $\mathbf{b}_{\ell m}(k_0) \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{b}$
  end for
  Compress columns: $\mathbf{Z}_\ell \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{Z}$

5.4 Visualizations

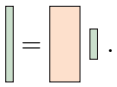
In many cases visualizations of the mathematics gives rise to a deeper understanding. Therefore, a short visual introduction to the ROSEBEM incorporating the BLI boundary condition is given. We start by introducing the following visual representation of the vector spaces

Using this we can visualize (5.14) as



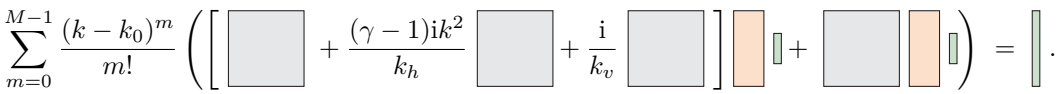
(5.18)

Similarly, the reduced basis assumption, for both $\mathbf{p}$ and $\partial_{\mathbf{n}}\mathbf{p}$, is visualized as



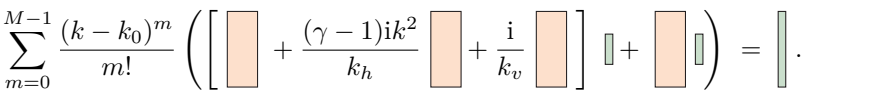
(5.19)

Inserting assumption from above into (5.18) while re-arranging similar to (5.16) it follows that



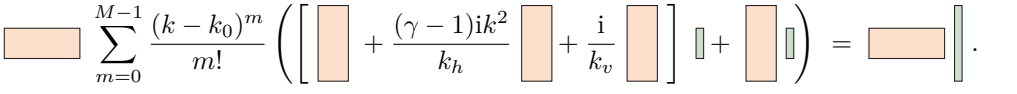
(5.20)

Now by performing the matrix-matrix, i.e. setting the reduced basis matrix inside of the parenthesis, it follows that



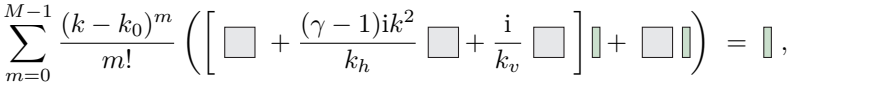
(5.21)

Note that the above is the first step in the assembly process of Algorithm 3 where only a row of original dense matrices are assembled at a time. The next step is to project columns by



(5.22)

Performing the multiplcations it follows that



(5.23)

which is the final ROSEBEM system.

5.5 Contributions

The contributions contained in Papers J1 and C1 are as follows:

- We successfully combined two methods from two different fields of research. That is, we extended the reduced order series expansion boundary element method (ROSEBEM) from [66] to incorporate the boundary layer impedance (BLI) boundary condition described in [15]. This required the derivation of the Taylor expansion of the tangential gradient of the Green's function which was not present in the previous publications. Furthermore, the final model differs from the one found in [66] due to the fact the frequency-dependent coefficients in front of the boundary integrals stemming from the BLI condition (Paper J1 and Paper C1).
- The model was numerically verified using two test cases representing impedance tubes with different setups of resonators at their termination ends. The quantity chosen to evaluate the model was the absorption coefficient (Paper J1) [47].
- The numerical investigation show that the reduced order basis must include information from at least two frequencies in order to accurately represent the solution in the full frequency range of interest. In particular, we show that the solution quickly drifts if only information from only a single frequency is used. This indicates that simply applying multiple ROSEBEMs to capture the full frequency range of interest will generally not work.
- The computational efficiency then the ROSEBEM is shown to be significantly better than the standard full solution. In particular, for the numerical example presented in Paper C1, the ROSEBEM is shown to be more 100 times faster while 1.65 times the storage, while the two setups in Paper J1 are shown to be around 25 timers faster while consuming less memory.
- Additionally, in Paper J1, we present an a priori analysis that is crucial when deciding whether a specific problem would benefit from using the ROSEBEM. The analysis is based on the number of Taylor terms included, the size of the reduced basis, and the number of frequencies of interest.

All in all, the proposed model significantly alleviates the computational bottlenecks in the cases for which the boundary layer impedance boundary condition can be used to approximate the viscous and thermal losses. As such, the results could lay the foundation for broadband shape optimization, including viscothermal losses, where many frequencies must be included into the optimization loop.

CHAPTER 6

Concluding remarks

In this PhD study we investigated various methods for improving the computational performance of the boundary element method including viscous and thermal losses. The purpose of the thesis was to improve the computational efficiency of the underlying simulation, thereby paving the way for both large-scale and broadband shape optimization including viscous and thermal losses. Given the difference in the two challenges, two distinct ideas were explored. The first approach looked at computational improvements in the full boundary element formulation including viscous and thermal losses. Here, the following conclusions could be raised:

- Boundary element simulations including viscous and thermal losses can be formulated in a way for which the tangential directions is not explicitly needed. This simplification makes it possible to exploit in sparsity structure of the thermal and viscous modes in the solution phase (Paper J2). This is opposed to previous formulations, where the sparsity could be utilized only in the assembly phase.
- The new formulation makes it possible to solve the linear system of equations using a nested iterative approach. As such, the formulation makes it possible to solve problems larger than before. However, since the computation still contains two dense matrices coming from the acoustical mode, the memory consumption acts as a computational bottleneck (Paper J2).
- The computational bottleneck that arises from dense acoustical matrices can be alleviated using standard techniques such as the fast multipole method or an $\mathcal{H}$-matrix (Paper J3).
- The findings show that the computation time for large problems is dominated by the reconstruction of the normal derivative of the acoustical pressure. This reconstruction step is equivalent to solving a pure acoustical system with Dirichlet boundary conditions. As such, this indicates that if it is possible to solve a pure acoustical problem using the boundary element method, then it should also be possible to solve the boundary element problem including viscous and thermal losses (Papers J2 and J3).

For the second problem, a reduced order series expansion boundary element method including the losses as boundary conditions was examined. Here, the following conclusions could be raised:

- It is possible to apply the reduced order series expansion boundary element method on the integrals stemming from the boundary layer impedance condition (Papers J1 and C1). Said differently, the method from [66] can be extended to include the boundary conditions of [15].
- The reduced order basis should be computed using information from at least two distinct frequencies in order to accurately capture the solution in the full frequency range of interest (Paper J1).
- The final computational model is significantly faster than the original full solution. In particular, it is shown to be more than 25 times faster while using a similar memory footprint (Paper J1).
- Additionally, we introduce an a priori analysis of when to use the ROSEBEM as opposed to the full formulation depending on the number of frequencies, the size of the reduced basis, and the number of Taylor series terms included (Paper J1).

CHAPTER 7

Outlook

When this PhD study was started, the main objective was to extend previous work on shape optimization in 2D to 3D [1, 2]. To do so required the development of efficient algorithms for larger problems (Papers J2 and J3) as well as methods for faster evaluation of multiple frequencies (Papers J1 and C1). However, a crucial part is missing in order for shape optimization to be possible. How do we efficiently compute the sensitivities?

- In particular, we must answer the question of whether or not it is even possible to compute the sensitives of the iterative scheme presented in Paper J2.
- If the previous statement is true, then the follow-up question would be to ask if it is possible to efficiently compute the sensitivities when acceleration techniques are utilized in the iterative scheme.
- A similar question can be asked for the ROSEBEM presented in Papers J1 and C1. Here, complexity comes into play with how the reduced basis affects the sensitivity calculation.

Furthermore, the following ideas building up the topics presented in this thesis could also be of interest.

- Given that most of the time of the new formulation presented in Paper J2 is spent on the reconstruction of the normal derivative of the acoustical pressure, a question on how to speedup should be asked. An obvious idea, but necessarily the only one, would be to look into the field of preconditioners.
- We mentioned briefly that the sparse assembly of the thermal and viscous modes is based on simple assumptions of contributions only from neighboring elements. This assumption is not, in general, valid and therefore must be considered. Ideas in this area could be looking at tree structures similar to the ones encountered for fast multipole methods.
- A current memory bottleneck of the ROSEBEM comes from the fact that the reduced basis is computed using dense matrices at the primary frequencies. This bottleneck can be relieved by instead utilizing acceleration techniques such as, e.g. the fast multipole method instead, as the Krylov vectors are computed using only matrix-vector products. This idea has already been implemented for scattering problems in the software presented in Paper C2, although left out of the paper due to the limitations of the paper length of the conference. A simple test case can be found in the online documentation of the software.
- The multifidelity approach described in Gurbuz et. al [41] could be used in combination with the ROSEBEM and the acceleration techniques. In particular, the ROSEBEM could be used to easily create data entries for many frequencies for the low-fidelity model, whereas the acceleration techniques could be used to capture the high-fidelity model. This is in contrast to the original paper that utilizes the standard full dense approach to generate the low- and high-fidelity data.
- Finally, as mentioned previously in the thesis, it would be of interest to investigate whether the IFGF method would be a good balance between memory and computational scaling.

Bibliography

[1] Peter Risby Andersen, Vicente Cutanda Henríquez, and Niels Aage. "On the validity of numerical models for viscothermal losses in structural optimization for micro-acoustics." In: *Journal for Sound and Vibration* (2021).

[2] Peter Risby Andersen, Vicente Cutanda Henríquez, and Niels Aage. "Shape optimization of micro-acoustic devices including viscous and thermal losses." In: *Journal of Sound and Vibration* 447 (2019), pages 120–136. ISSN: 10958568. DOI: 10.1016/j.jsv.2019.01.047.

[3] E. Anderson et al. *LAPACK Users' Guide*. Third. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999. ISBN: 0-89871-447-8 (paperback).

[4] Travis Askham et al. *FMM3D*. https://github.com/flatironinstitute/FMM3D. 2023.

[5] Noureddine Atalla and Franck Sgard. *Finite Element and Boundary Methods in Structural Acoustics and Vibration*. CRC Press, 2015. DOI: 10.1201/b18366.

[6] Jonas Ballani and Daniel Kressner. "Matrices with Hierarchical Low-Rank Structures." In: *Lecture Notes in Mathematics* 2173 (2016), pages 161–209.

[7] Alex Barnett et al. *2014 CBMS-NSF Conference: Fast Direct Solvers for Elliptic PDEs*. 2014. URL: https://math.dartmouth.edu/~fastdirect/mat.php (visited on 02 August 2023).

[8] Christoph Bauinger and Oscar P. Bruno. "Massively parallelized interpolated factored Green function method." In: *Journal of Computational Physics* 475 (2023), page 111837. ISSN: 10902716. DOI: 10.1016/j.jcp.2022.111837.

[9] Christoph Bauinger and Oscar P. Bruno. ""Interpolated Factored Green Function" Method for accelerated solution of Scattering Problems." In: *Journal of Computational Physics* 430 (2021), page 110095. ISSN: 10902716. DOI: 10.1016/j.jcp.2020.110095.

[10] Suhaib Koji Baydoun, Matthias Voigt, and Steffen Marburg. "Low-Rank Iteration Schemes for the Multi-Frequency Solution of Acoustic Boundary Element Equations." In: *Journal of Theoretical and Computational Acoustics* 29.3 (2021). DOI: 10.1142/S2591728521500043.

[11] Suhaib Koji Baydoun et al. "A greedy reduced basis scheme for multifrequency solution of structural acoustic systems." In: *International Journal for Numerical Methods in Engineering* 121.2 (2020), pages 187–200. ISSN: 10970207. DOI: 10.1002/nme.6205.

[12] Rick Beatson and Leslie Greengard. *A short course on fast multipole methods*. https://math.nyu.edu/~greengar/shortcourse_fmm.pdf. (Visited on 03 August 2023).

[13] Mario Bebendorf. *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*. Springer, 2008. ISBN: 9783540771470.

[14] Gernot Beer, Benjamin Marussig, and Christian Duenser. *The Isogeometric Boundary Blement Method*. Volume 90. Springer Verlag, 2020. ISBN: 9783030233396. DOI: 10.1007/978-3-030-23339-6_1.

[15] Martin Berggren, Anders Bernland, and Daniel Noreland. "Acoustic boundary layers as boundary conditions." In: *Journal of Computational Physics* 371 (2018), pages 633–650. DOI: 10.1016/j.jcp.2018.06.005.

[16] Jeff Bezanson et al. "Julia: A Fresh Approach to Numerical Computing." In: *SIAM Review* 59.1 (2017), pages 65–98. DOI: 10.1137/141000671.

[17] Jeffrey Bezanson. "Abstraction in Technical Computing." PhD thesis. Massachusetts Institute of Technology, 2015.

[18] L Susan Blackford et al. "An Updated Set of Basic Linear Algebra Subprograms (BLAS)." In: *ACM Transactions on Mathematical Software* 28.2 (2002), pages 135–151.

[19] Boris Bonev. *HssMatrices.jl*. https://github.com/bonevbs/HssMatrices.jl. 2023.

[20] M Brunaeu et al. "General formulation of the dispersion equation in bounded visco-thermal fluid, and application to some simple geometries." In: *Wave Motion* 11.5 (1989), pages 441–451. DOI: 10.1016/0165-2125(89)90018-8.

[21] Steffen Börm and Lars Grasedyck. *HLib*. URL: http://www.hlib.org (visited on 02 August 2023).

[22] Steffen Börm et al. *H2Lib*. URL: http://www.h2lib.org/ (visited on 02 August 2023).

[23] Gustavo Chavez et al. *Curated list of Hierarchical Matrices libraries.* https://github.com/gchavez2/awesome_hierarchical_matrices. 2023.

[24] H. Cheng, L. Greengard, and V. Rokhlin. "A Fast Adaptive Multipole Algorithm in Three Dimensions." eng. In: *Journal of Computational Physics* 155.2 (1999), pages 468–498. ISSN: 10902716. DOI: 10.1006/jcph.1999.6355.

[25] David Colton and Rainer Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer, 1998. ISBN: 9783662035375.

[26] Many Contributors. *SparseArrays.jl*. https://github.com/JuliaSparse/SparseArrays.jl. 2023.

[27] Robert D. Cook and Robert Davis Cook. *Concepts and Applications of Finite Element Analysis*. Wiley, 2002. ISBN: 0471356050.

[28] Vicente Cutanda Henríquez and Peter Risby Andersen. "A three-dimensional acoustic Boundary Element Method formulation with viscous and thermal losses based on shape function derivatives." In: *Journal of Computational Acoustics* 26.3 (2018). DOI: 10.1142/S2591728518500391.

[29] Timothy A. Davis. "Algorithm 1000: SuiteSparse: GraphBLAS: Graph Algorithms in the Language of Sparse Linear Algebra." In: *ACM Trans. Math. Softw.* 45.4 (2019). ISSN: 0098-3500. DOI: 10.1145/3322125.

[30] Timothy Alden Davis. *SuiteSparse*. https://github.com/DrTimothyAldenDavis/SuiteSparse. 2023.

[31] DTU Computing Center. *DTU Computing Center resources*. 2021. DOI: 10.48714/DTU.HPC.0001.

[32] Luiz M. Faria. *HMatrices.jl*. https://github.com/WaveProp/HMatrices.jl. 2023.

[33] Luiz M. Faria. *IFGF.jl*. https://github.com/WaveProp/IFGF.jl. 2023.

[34] R.P. Feynman, R.B. Leighton, and M. Sands. *The Feynman Lectures on Physics, Vol. I: Mainly Mechanics, Radiation, and Heat*. v. 1. Basic Books, 2015. ISBN: 9780465040858. URL: https://www.feynmanlectures.caltech.edu/I_toc.html.

[35] R.P. Feynman, R.B. Leighton, and M. Sands. *The Feynman Lectures on Physics, Vol. II: Mainly Electromagnetism and Matter*. v. 1. Basic Books, 2015. ISBN: 9780465040858. URL: https://www.feynmanlectures.caltech.edu/II_toc.html.

[36] Tobias Frenzel et al. "Three-dimensional labyrinthine acoustic metamaterials." In: *Applied Physics Letters* 103.6 (2013), page 061907. ISSN: 1077-3118. DOI: 10.1063/1.4817934.

[37] G. H. Golub and C. Reinsch. "Singular Value Decomposition and Least Squares Solutions." In: *Numerische Mathematik* 14.5 (1970), pages 403–420. DOI: 10.1007/BF02163027.

[38] Leslie Greengard and V Rokhlin. "A Fast Algorithm for Particle Simulations." In: *Journal of Computational Physics* 73.2 (1987), pages 325–348. ISSN: 10902716, 00219991. DOI: 10.1016/0021-9991(87)90140-9.

[39] Leslie Greengard and Vladimir Rokhlin. "A new version of the Fast Multipole Method for the Laplace equation in three dimensions." In: *Acta Numerica* 6 (1997), pages 229–269. ISSN: 09624929. DOI: 10.1017/S0962492900002725.

[40] Nail A. Gumerov and Ramani Duraiswami. *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*. Elsevier, 2005. ISBN: 0080443710. DOI: 10.1016/B978-0-08-044371-3.X5000-5.

[41] Caglar Gurbuz et al. "A multi-fidelity Gaussian process for efficient frequency sweeps in the acoustic design of a vehicle cabin." In: *Journal of the Acoustical Society of America* 153.4 (2023), pages 2006–2018. ISSN: 15208524, 00014966. DOI: 10.1121/10.0017725.

[42] Wolfgang Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*. Springer, 2015. ISBN: 978-3662473245.

[43] Vicente Cutanda Henriquez. "Numerical Transducer Modeling." PhD thesis. Technical University of Denmark, 2002. ISBN: 8791184096, 9788791184093.

[44] U. Hetmaniuk, R. Tezaur, and C. Farhat. "Review and assessment of interpolatory model order reduction methods for frequency response structural dynamics and acoustics problems." In: *International Journal for Numerical Methods in Engineering* 90.13 (2012), pages 1636–1662. DOI: 10.1002/nme.4271.

[45] Nicholas J. Higham et al., editors. *The Princeton Companion to Applied Mathematics*. Princeton, NJ, USA: Princeton University Press, 2015. ISBN: 978-0-691-15039-0.

[46] Frank Ihlenburg. *Finite Element Analysis of Acoustic Scattering*. Springer, 1998. ISBN: 978-0387983196.

[47] "ISO 10534-2: 1998, Accoustics Determination of Sound Absorption Coefficient and Impedance Tubes Part 2 : Transfer Function Method." In: (1998).

[48] Christopher J. Geoga, Mihai Anitescu, and Michael L. Stein. "Scalable Gaussian Process Computations Using Hierarchical Matrices." In: *Journal of Computational and Graphical Statistics* 29.2 (2020), pages 227–237. DOI: 10.1080/10618600.2019.1652616.

[49] Dan Jiao, Xian Yang Zhu, and Jian Ming Jin. "Fast and accurate frequency-sweep calculations using asymptotic waveform evaluation and the combined-field integral equation." In: *Radio Science* 34.5 (1999), pages 1055–1063. ISSN: 1944799x, 00486604. DOI: 10.1029/1999RS900068.

[50] N. Jiménez et al. "Ultra-thin metamaterial for perfect and quasi-omnidirectional sound absorption." In: *Applied Physics Letters* 109.12 (2016), page 121902. ISSN: 1077-3118. DOI: 10.1063/1.4962328.

[51] Erik Jørgensen et al. "Higher Order Hierarchical Legendre Basis Functions for Electromagnetic Modeling." eng. In: *IEEE Transactions on Antennas and Propagation* 52.11 (2004), pages 2985–2995. ISSN: 15582221. DOI: 10.1109/TAP.2004.835279.

[52] SM Kirkup and DJ Henwood. "Methods For Speeding Up the Boundary Element Solution of Acoustic Radiation Problems." In: *Journal of Vibration and Acoustics-transactions of the Asme* 114.3 (1992), pages 374–380. DOI: 10.1115/1.2930272.

[53] Stephen Kirkup. "The Boundary Element Method in Acoustics: A Survey." In: *Applied Sciences* 9.8 (2019), page 1642. DOI: 10.3390/app9081642.

[54] Ronald Kriemann. *HLIBpro*. URL: https://www.hlibpro.com/doc/ (visited on 02 August 2023).

[55] Peter D. Lax and Maria Shea Terrell. *Multivariable Calculus with Applications*. Springer, 2017. ISBN: 9783319740737.

[56] Quhao Li et al. "Reduced-order methods for dynamic problems in topology optimization: A comparative study." In: *Computer Methods in Applied Mechanics and Engineering* 387 (2021). DOI: 10.1016/j.cma.2021.114149.

[57] Steffen Marburg. "Boundary Element Method for Time–Harmonic Acoustic Problems." In: *Computational Acoustics*. Edited by Manfred Kaltenbacher. CISM International Centre for Mechanical Sciences. Wien: Springer, 2017. Chapter 3, pages 69–159.

[58] Steffen Marburg. "The Burton and Miller Method: Unlocking Another Mystery of its Coupling Parameter." In: *Journal of Computational Acoustics* 24.1 (2016), page 1550016. ISSN: 17936489, 0218396x. DOI: 10.1142/S0218396X15500162.

[59] Steffen Marburg and Bodo Nolte. *Computational Acoustics of Noise Propagation in Fluids - Finite and Boundary Element Methods*. Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-77448-8.

[60] Steffen Marburg and Stefan Schneider. "Performance of iterative solvers for acoustic problems. Part I. Solvers and effect of diagonal preconditioning." In: *Engineering Analysis with Boundary Elements* 27.7 (2003), pages 727–750. ISSN: 0955-7997. DOI: 10.1016/s0955-7997(03)00025-0.

[61] Per-Gunnar Martinsson. *Fast Direct Solvers for Elliptic PDEs*. Society for Industrial and Applied Mathematics, 2019. DOI: 10.1137/1.9781611976045.

[62] Per-Gunnar Martinsson and V. Rokhlin. "A fast direct solver for boundary integral equations in two dimensions." eng. In: *Journal of Computational Physics* 205.1 (2005), pages 1–23. ISSN: 10902716, 00219991. DOI: 10.1016/j.jcp.2004.10.033.

[63] Miguel Molerón, Marc Serra-Garcia, and Chiara Daraio. "Visco-thermal effects in acoustic metamaterials: from total transmission to total reflection and high absorption." In: *New Journal of Physics* 18.3 (2016). DOI: 10.1088/1367-2630/18/3/033003.

[64] P.M. Morse and K.U. Ingard. *Theoretical acoustics*. McGraw-Hill, 1968.

[65] Jean-Claude Nédélec. *Acoustic and Electromagnetic Equations: Integral Representations for Harmonic Problems*. Springer, 2001. ISBN: 9781475743937.

[66] Dionysios Panagiotopoulos, Elke Deckers, and Wim Desmet. "Krylov subspaces recycling based model order reduction for acoustic BEM systems and an error estimator." In: *Computer Methods in Applied Mechanics and Engineering* 359 (2020), page 112755. ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2019.112755.

[67] K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. Version 20121115. 2012. URL: http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html.

[68] Simone Preuss et al. "Recent Advances in Acoustic Boundary Element Methods." In: *Journal of Theoretical and Computational Acoustics* 30.3 (2022), page 2240002. ISSN: 25917811, 25917285. DOI: 10.1142/S2591728522400023.

[69] D. Ryckelynck et al. "An efficient 'a priori' model reduction for boundary element models." In: *Engineering Analysis With Boundary Elements* 29.8 (2005), pages 796–801. DOI: 10.1016/j.enganabound.2005.04.003.

[70] Youcef Saad and Martin H. Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems." In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), pages 856–869. DOI: 10.1137/0907058.

[71] Stefan Schneider and Steffen Marburg. "Performance of iterative solvers for acoustic problems. Part II. Acceleration by ILU-type preconditioner." In: *Engineering Analysis with Boundary Elements* 27.7 (2003), pages 751–757. ISSN: 0955-7997. DOI: 10.1016/s0955-7997(03)00016-x.

[72] Luka Stansic. *ScalFMM*. https://github.com/polatory/ScalFMM. 2023.

[73] Tingyu Wang, Rio Yokota, and Lorena A. Barba. "ExaFMM: a high-performance fast multipole method library with C++ and Python interfaces." In: *Journal of Open Source Software* 6.61 (2021), page 3145. DOI: 10.21105/joss.03145.

[74] T. W. Wu. *Boundary Element Acoustics: Fundamentals and Computer Codes*. WIT, 2000. ISBN: 9781853125201.

[75] Xiang Xie and Yijun Liu. "An adaptive model order reduction method for boundary element-based multi-frequency acoustic wave problems." In: *Computer Methods in Applied Mechanics and Engineering* 373 (2021). DOI: 10.1016/j.cma.2020.113532.

[76] Liu Yijun. *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*. Cambridge University Press, 2009. ISBN: 9780511605345, 0521116597, 9780521116596. DOI: 10.1017/CBO9780511605345.

[77] Lexing Ying. "A pedestrian introduction to fast multipole methods." In: *Science China Mathematics* 55.5 (2012), pages 1043–051. DOI: 10.1007/s11425-012-4392-0.

[78] Lexing Ying. *The Fast Multipole Method*. 2014. URL: https://www.youtube.com/watch?v=qMLIyZi8Sz0 (visited on 02 August 2023).

Paper J1

Journal of Theoretical and Computational Acoustics

A Reduced Order Series Expansion for the BEM Incorporating the Boundary Layer Impedance Condition

Mikkel Paltorp*, Vicente Cutanda Henríquez†, Niels Aage‡
Centre for Acoustic-Mechanical Micro Systems
*,† *Department of Electrical and Photonics Engineering*
‡ *Department of Civil and Mechanical Engineering*
The Technical University of Denmark
Ørsteds Plads, Bygning 352, 2800 Kgs. Lyngby, Denmark
* *mpasc@dtu.dk*

Peter Risby Andersen
Audio Research, GN Audio A/S & Jabra
2750 Ballerup, Denmark

Received (Day Month Year)
Revised (Day Month Year)

When modelling sound waves in fluids it can be important to include the viscous and thermal losses originating from the fluids' interaction with boundaries. In the audible frequency range the thickness of the boundary layers is between a micrometer and a millimeter. As such the viscous and thermal losses are important when simulating the properties of small acoustical devices such as e.g. hearing aids or transducers. However, the inclusion of viscous and thermal losses is a computationally demanding task as it requires a fine discretization of the boundary layer in order to fully capture the complicated physical phenomena happening on the microscale. Recently there has been developments to ease the computational demands using both the Finite Element Method and the Boundary Element Method, by approximating the losses using the Boundary Layer Impedance boundary condition. In this paper we extend previous developments for multifrequency analysis using the Reduced Order Series Expansion Boundary Element Method to handle the BLI condition. This model follows a two-step procedure: Using a series expansion to decrease the assembly time of the BEM matrices and a projection to reduce the overall memory consumption of the model. Results from two acoustic interior problems show that the model decreases the total computational time by around 96% while using less than 15% of the memory. For both test setups the limiting factor of the accuracy was the reduction and not the series expansion.

Keywords: boundary layer impedance; boundary element method and model order reduction

1. Introduction

Sound waves experience losses through viscous and thermal effects, which become relevant i) over long propagation distances, of no relevance for the cases addressed here, and ii) within a thin region next to the boundaries. For audible frequencies, the thickness of these boundary layers is of the order of tens to hundreds of micrometers.[1] These lossy regions, tough small, can become relevant for low-volume, small devices, e.g. hearing aids or transducers.

Traditionally viscous and thermal losses have been modelled using the Full Linearized Navier-Stokes (FNLS) equations and solved using e.g. the Finite Element Method (FEM).[2]

Solving these equations is typically computationally demanding since it introduces the temperature and the velocity as additional Degrees Of Freedom (DOF). Another drawback of this approach is that it requires the use diminutive elements in the close vicinity of the boundary in order to fully capture the underlying microscale physics, which leads to even more DOF. In order to avoid this fine boundary meshing the Boundary Element Method (BEM) can be used as an alternative to the FEM. This approach utilizes the Kirchhoff Decomposition (KD) which separates the FLNS into five Helmholtz equations that is then coupled through the boundary conditions.[3,4]

Recently there has been an interest in approximating the viscous and thermal losses using the Boundary Layer Impedance (BLI) boundary conditions from Berggren *et al.*,[5] as this does not introduce any extra DOF. Instead, this approximation relates the pressure with its normal derivative and tangential Laplacian. This formulation is derived under certain assumptions such as non-overlapping boundary layers and low curvature. These assumptions must be checked to hold by the user or the approximation might not be correct.[6,7]

A general drawback of the BEM is that it requires a recomputation of the discrete form of the underlying integral equation for each frequency of interest. As a result, performing frequency sweep analysis using the BEM is a computationally demanding process. Multiple solutions to this shortcoming of the BEM have been developed. Most of these techniques can be split into two categories: Interpolation of the discrete form of the integral equation[8–10] or an approximation of the Green's function before computing the discrete form of the integral equation.[9,11,12] In both cases the central idea is to reuse computations at certain frequencies (called primary frequencies) at intermediate frequencies (called secondary frequencies). As an alternative to decreasing the assembly time of the discrete form it is possible to instead look for an approximate functional form of the response using e.g. a Padé approximation.[13] This is a standard technique for dynamic FEM problems,[14,15] however it is not directly transferable to the BEM as the boundary integrals are infinitely smooth with respect to the frequency. Worse yet, in the case of the BLI condition the derivatives are non-trivial as the viscous and thermal wavenumbers depend on the frequency.[16]

The main contribution of this paper is a Reduced Order Series Expansion Boundary Element Method (ROSEBEM) for multi-frequency analysis including the Boundary Layer Impedance condition. This is achieved by applying a similar approach to that of Panagiotopoulos *et al.*[11] to the model described by Berggren *et al.*[5] The model differs from the previous works in two ways. Firstly the derivation of a Series Expansion Boundary Element Method (SEBEM) for the BLI condition requires an additional Taylor expansion of the tangential gradient of the Green's function. Secondly, since the coefficients in front of two of the boundary integrals are depended on the wavenumber, the final model requires the storage of three separate reduced matrices, in contrast to the previous model where only one was needed.

The paper is organized as follows: In section 2 an introduction to the Boundary Element Method (BEM) as well as the Boundary Layer Impedance (BLI) boundary condition is given. Next section 3 explains how the Series Expansion Boundary Element Method (SEBEM) can be utilized to remove the frequency dependency of the BLI-BEM integrals. Then in

section 4 a simple Reduced Order Model (ROM) approach is applied to the SEBEM resulting in a Reduced Order Series Expansion Boundary Element Method (ROSEBEM). Lastly in section 5 the model is verified numerically by comparison against the full BEM solution.

2. The Boundary Element Method

In this section a brief introduction to the Boundary Element Method[17–21] (BEM) is given. Afterwards the Boundary Layer Impedance[5] (BLI) boundary condition is summarized. Lastly a short description of how to compute the tangential gradient is given.

2.1. *The Boundary Integral Equation for Acoustics*

A common way of solving time-harmonic acoustical problems is to use the Boundary Element Method (BEM). In the case of the scalar Helmholtz equation this means solving the Kirchhoff-Helmholtz integral given by

$$\phi(\mathbf{x})p(\mathbf{x}) = \int_\Gamma G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\,\mathrm{d}S_\mathbf{y} - \int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}p(\mathbf{y})\,\mathrm{d}S_\mathbf{y}, \quad \mathbf{x}\in\Omega \tag{1}$$

where Ω is a domain with boundary Γ, $G(\mathbf{x},\mathbf{y})$ is the Green's function, $p(\mathbf{y})$ and $\mathbf{n}(\mathbf{y})$ is respectively the pressure and normal at point $\mathbf{y}$ on the surface while $p(\mathbf{x})$ and $\phi(\mathbf{x})$ is respectively the pressure and the integral free form at point $\mathbf{x}$ in the domain. In the following the time-dependency $e^{\mathrm{i}\omega t}$ is used, meaning that[22]

$$G(\mathbf{x},\mathbf{y}) = \frac{\exp\left(-\mathrm{i}k\|\mathbf{x}-\mathbf{y}\|_2\right)}{4\pi\|\mathbf{x}-\mathbf{y}\|_2}. \tag{2}$$

The numerical approximation of (1) then follows from two steps: Firstly a discretization of the boundary (Γ) into elements and secondly setting an interpolation scheme for the pressure (p) and its normal derivative ($\partial_\mathbf{n} p$). Following these two steps the two integrals can be approximated using a quadrature scheme for the integral over each element. What is left is to compute the nodal values of the chosen interpolation nodes of the pressure and its normal derivative. There exist a plethora of ways of doing so with the main three approaches being the Galerkin, Collocation and Nyström approaches. The approach taken here is that of Collocation, which generates an equation for each node by setting $\mathbf{x}$ in (1) equal to all the interpolation nodes of p and $\partial_\mathbf{n} p$. The resulting linear system of equations has the form

$$\mathrm{diag}(\boldsymbol{\phi})\mathbf{p} = \mathbf{G}\frac{\partial \mathbf{p}}{\partial \mathbf{n}} - \mathbf{F}\mathbf{p}. \tag{3}$$

After applying boundary conditions the above system will become consistent, and it can be solved using standard techniques. Since the solution gives an interpolation representation for both p and $\partial_\mathbf{n} p$ on the boundary it can be used to compute (1) for any $\mathbf{x}$ in the domain Ω.

2.2. *Boundary Layer Impedance*

The inclusion of viscothermal losses in acoustical computations can be approximated using the so-called Boundary Layer Impedance (BLI) boundary condition.[5] Here the viscothermal effects are approximated by applying a Wentzell type boundary condition of the form[5,7]

$$\frac{\partial p}{\partial \mathbf{n}}(\mathbf{x}) = \left[(\gamma - 1)\frac{\mathrm{i}k^2}{k_h} - \frac{\mathrm{i}\Delta^{\parallel}}{k_v}\right] p(\mathbf{x}), \tag{4}$$

where $\Delta^{\parallel}$ is the tangential Laplacian, γ is the heat capacity ratio and k, k_v, k_h are respectively the isentropic, viscous and thermal wavenumbers. The two additional wavenumbers are computed using the physical properties of the fluid such as the thermal conductivity, specific heat capacity under constant pressure and the shear and bulk viscosity coefficients.[3]

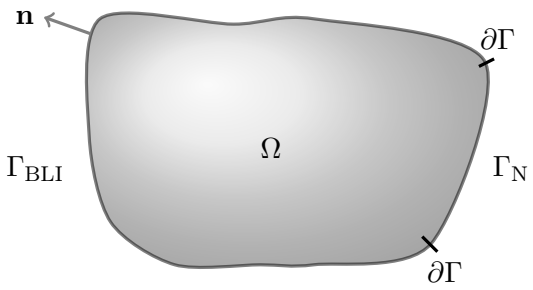


Fig. 1: Domain of an interior problem.

For the sake of simplicity the following derivation is done in the case of Neumann condition, but a similar approach can be applied in the case of e.g. an impedance boundary condition. The first step is to split the boundary (Γ) into a part with a Neumann boundary condition (Γ_{N} in Figure 1) and a part with a BLI condition (Γ_{BLI} in Figure 1). Applying this to (1) one finds that it can be written as

$$\begin{aligned}\phi(\mathbf{x})p(\mathbf{x}) = &\int_{\Gamma_{\mathrm{N}}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\,\mathrm{d}S_{\mathbf{y}} - \int_{\Gamma} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}} \\ &+ \int_{\Gamma_{\mathrm{BLI}}} G(\mathbf{x},\mathbf{y})\left[\frac{(\gamma-1)\mathrm{i}k^2}{k_h} - \frac{\mathrm{i}\Delta^{\parallel}_{\mathbf{y}}}{k_v}\right] p(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}}.\end{aligned} \tag{5}$$

Applying integration by parts to the term including $\Delta^{\parallel}_{\mathbf{y}}$ in (5) while assuming

$$\int_{\partial\Gamma_{\mathrm{BLI}}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\,\mathrm{d}S_{\mathbf{y}} = 0, \tag{6}$$

it follows that

$$\begin{aligned}\phi(\mathbf{x})p(\mathbf{x}) = &\int_{\Gamma_\mathrm{N}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\,\mathrm{d}S_\mathbf{y} - \int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y})\,\mathrm{d}S_\mathbf{y} \\ &+ \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\int_{\Gamma_\mathrm{BLI}} G(\mathbf{x},\mathbf{y})p(\mathbf{y})\,\mathrm{d}S_\mathbf{y} + \frac{\mathrm{i}}{k_v}\int_{\Gamma_\mathrm{BLI}} \nabla^{\parallel}_\mathbf{y} G(\mathbf{x},\mathbf{y})\cdot\nabla^{\parallel}_\mathbf{y} p(\mathbf{y})\,\mathrm{d}S_\mathbf{y},\end{aligned} \tag{7}$$

where $\nabla^{\parallel}_\mathbf{y}$ denotes the tangential gradient. The tangential gradient of the Green's function can be computed as the vector rejection of the gradient of the Green's function onto the normal direction i.e.

$$\nabla^{\parallel}_\mathbf{y} G(\mathbf{x},\mathbf{y}) = (\mathbf{I} - \mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y})^\top)\nabla_\mathbf{y} G(\mathbf{x},\mathbf{y}), \tag{8}$$

where $\mathbf{I}$ is the identity matrix. In order to discretize (7) a discrete form of $\nabla^{\parallel}_\mathbf{y} p(\mathbf{y})$ must be introduced. The following subsection explains how.

2.3. *Computing the Tangential Gradient*

As the Boundary Element (BE) interpolation only describes the surface values it does not contain derivative information orthogonal to the surface. As such the tangential gradient of the pressure can be computed as the gradient of the chosen BE interpolation. In the following derivation the notation $\mathbf{x}(\mathbf{u}) = \begin{bmatrix} x(\mathbf{u}) & y(\mathbf{u}) & z(\mathbf{u}) \end{bmatrix}^\top$ with $\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top$ is used. On element j the BE interpolation is given as

$$p\left(\mathbf{x}(\mathbf{u})\right) = p\left(\mathbf{N}_x(\mathbf{u})^\top \mathbf{X}_j\right) = \mathbf{N}_p(\mathbf{u})^\top \mathbf{p}^j, \quad \mathbf{x}\in\Gamma_j, \tag{9}$$

where $\mathbf{X}_j$ and $\mathbf{p}^j$ contains respectively the nodal positions and pressures of element j, and $\mathbf{N}_x$ and $\mathbf{N}_p$ denotes the chosen interpolation scheme for respectively the geometry and pressure. In order for the gradient to be non-zero the interpolation of the pressure must at least be linear. The resulting gradient of p with respect to $\mathbf{u}$ on element j is given by

$$\nabla_\mathbf{u} p\left(\mathbf{x}(\mathbf{u})\right) = \left(\nabla_\mathbf{u}\mathbf{N}_p(\mathbf{u})^\top\right)\mathbf{p}^j, \tag{10}$$

since the values in $\mathbf{p}^j$ are constant. The chain rule for gradients it states that $\nabla_\mathbf{u} p = (\nabla_\mathbf{u}\mathbf{x}^\top)\nabla_\mathbf{x} p$.[23] From this the tangential gradient can be isolated. Note that since the computations are based on Boundary Elements the isolation poses an initial problem as $(\nabla_\mathbf{u}\mathbf{x}^\top)$ is a 2×3-matrix which is not directly inverted. It has been shown that it is enough to introduce an artificial variable, u_3, for which $\frac{\partial \mathbf{N}_p}{\partial u_3}^\top = \mathbf{0}^\top$ (where $\mathbf{0}$ is a vector of appropriate length filled with zeros) while substituting $\frac{\partial \mathbf{x}}{\partial u_3} = \frac{\partial \mathbf{x}}{\partial u_1}\times\frac{\partial \mathbf{x}}{\partial u_2}$.[19] As such it follows that

$$\nabla^{\parallel}_\mathbf{x} p\left(\mathbf{x}(\mathbf{u})\right) = \left(\nabla_\mathbf{u}\mathbf{x}(\mathbf{u})^\top\right)^{-1}\left(\nabla_\mathbf{u}\mathbf{N}_p(\mathbf{u})^\top\right)\mathbf{p}^j = \begin{bmatrix} \frac{\partial x}{\partial u_1} & \frac{\partial y}{\partial u_1} & \frac{\partial z}{\partial u_1} \\ \frac{\partial x}{\partial u_2} & \frac{\partial y}{\partial u_2} & \frac{\partial z}{\partial u_2} \\ \frac{\partial \mathbf{x}}{\partial u_1}\times\frac{\partial \mathbf{x}}{\partial u_2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathbf{N}_p}{\partial \xi_1}^\top \\ \frac{\partial \mathbf{N}_p}{\partial \xi_2}^\top \\ \mathbf{0}^\top \end{bmatrix}\mathbf{p}^j, \tag{11}$$

where the superscript $^{\parallel}$ indicates that the gradient only contains the tangential information.

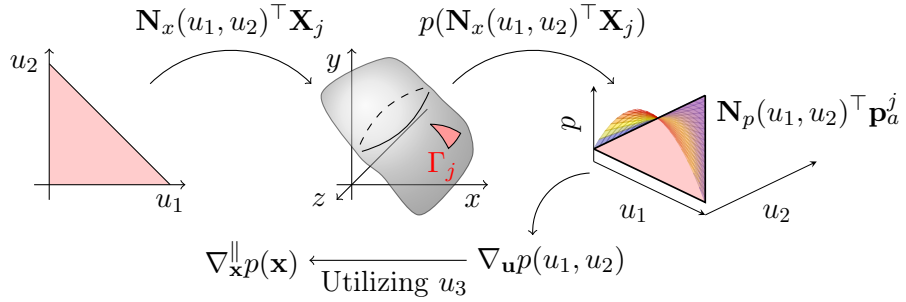


Fig. 2: Visualization of the interpolation function derivative approach.

3. Series Expansion Boundary Element Method

A weakness of the Boundary Element Method (BEM) is that it requires a recomputation of the integrals for every frequency of interest. In order to alleviate this issue the so-called Series Expansion Boundary Element Method (SEBEM) is utilized. The basic idea of the SEBEM is to assemble a few BEM matrices at specified frequencies (called primary frequencies) and then using a polynomial interpolation as approximations of the BEM matrices at intermediate frequencies (called secondary frequencies). Common choices of polynomials are the Taylor polynomials,[11] Chebyshev polynomials[10] and even simple q-order polynomials.[8] The approach taken here is that of expanding the Green's function and its derivatives in terms of their respective Taylor series. This idea is far from new and was already commonly used in the early 90s,[9] but has recently regained interest.[11]

3.1. *The Series Expansion*

In contrast to previous works the formulation in (7) requires the Taylor series expansion of the tangential gradient of the Green's function

$$\nabla_{\mathbf{y}}^{\parallel}G(\mathbf{x},\mathbf{y}) = \sum_{m=0}^{\infty}\frac{(k-k_0)^m}{m!}\left(\mathbf{I}-\mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y})^\top\right)\frac{(\mathbf{x}-\mathbf{y})\,(-\mathrm{i}r)^m(1+\mathrm{i}k_0r-m)\exp\left(-\mathrm{i}k_0r\right)}{4\pi r^3},$$

where $r=\|\mathbf{x}-\mathbf{y}\|_2$ is introduced in order to ease the size of the above expression. Expansions for the remaining kernels can be found in Appendix A. By inserting the four Taylor series expansions into (7) the wavenumber can be decoupled such that its interaction with the integrating variable ($\mathbf{y}$) is linear. As such the discrete form of (7) reduces to

$$\begin{aligned}\operatorname{diag}(\boldsymbol{\phi})\mathbf{p} = &\left(\sum_{m=0}^{\infty}\frac{(k-k_0)^m}{m!}\mathbf{G}_m\right)\frac{\partial\mathbf{p}}{\partial\mathbf{n}} - \left(\sum_{m=0}^{\infty}\frac{(k-k_0)^m}{m!}\mathbf{F}_m\right)\mathbf{p}\\ &+\frac{(\gamma-1)\mathrm{i}k^2}{k_h}\left(\sum_{m=0}^{\infty}\frac{(k-k_0)^m}{m!}\mathbf{H}_m\right)\mathbf{p}+\frac{\mathrm{i}}{k_v}\left(\sum_{m=0}^{\infty}\frac{(k-k_0)^m}{m!}\mathbf{T}_m\right)\mathbf{p}.\end{aligned} \tag{12}$$

A short description on how to compute the matrices above can be found in Appendix B. After applying the Neumann boundary condition the above becomes

$$\left(\mathrm{diag}(\boldsymbol{\phi}) + \sum_{m=0}^{\infty} \frac{(k-k_0)^m}{m!}\left[\mathbf{F}_m - \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\mathbf{H}_m - \frac{\mathrm{i}}{k_v}\mathbf{T}_m\right]\right)\mathbf{p} = \sum_{m=0}^{\infty}\frac{(k-k_0)^m}{m!}\mathbf{b}_m. \quad (13)$$

In any practical setting, however, one can only include a finite number of terms of the above sums. It has been suggested that the Lagrange form of the remainder of the Taylor series can be used as an a priori estimation of the required order achieving a specified desired maximum error.[11] Alternatively the integration error of the Lagrange remainder on the largest element on the mesh gives s similar estimate.[24] In practice, however, one should not use these suggestions blindly as they are only based on the Taylor expansion of the Green's function itself which is only relevant for pure Dirichlet boundary conditions. At the same time the correspondence between the error of the integral of the Green's function and the error of the solution to the linear system are not one-to-one. A rigorous analysis would require the usage of the condition number of the linear system described in (13), however, this will result in the error estimator losing its main purpose as a simple a priori estimate for the order of the Taylor series.

4. Computing a Projection Basis

An essential consideration when applying the Series Expansion Boundary Element Method (SEBEM) is the large amount of memory required if applied directly. Specifically the memory scales $\mathcal{O}(N^2M)$, where N is the number of Degrees Of Freedom (DOF) and $M-1$ is the order of the Taylor series applied. This means that for even reasonably large problems the total memory usage will be the bottleneck of this approach. From a computational point of view the method should only be applied when the number of frequencies of interests (N_f) is larger than the order of the Taylor series ($M \leq N_f$). To resolve the memory issue of the SEBEM a Galerkin projection method can be utilized.[11] This approach reduces the memory scaling to $\mathcal{O}(\ell^2 M)$, which is a very practical improvement in the case of $\ell \ll N$.

4.1. *Galerkin Projection*

The main assumption behind the projection method is that the solution, as well as the right-hand side of (13), for a certain frequency range is spanned by a lower dimensional subspace. This means that

$$\mathbf{z} \approx \mathbf{U}_\ell \mathbf{z}_\ell, \quad \mathbf{b}_m \approx \mathbf{W}_\ell \mathbf{b}_{\ell m}, \quad \mathbf{U}_\ell, \mathbf{W}_\ell \in \mathbb{C}^{N\times\ell} \quad (14)$$

will be good approximations within this frequency range. This assumption is valid for BE systems which are well-conditioned. In particular, for BE systems for which the Generalized Minimal Residual Method (GMRes) converges quickly, the ℓ-Krylov subspace

$$\mathcal{K}_\ell(\mathbf{A}, \mathbf{b}) = \mathrm{span}\left\{\mathbf{b},\ \mathbf{A}\mathbf{b},\ \mathbf{A}^2\mathbf{b},\ \ldots,\ \mathbf{A}^{\ell-1}\mathbf{b}\right\}, \quad (15)$$

is naturally a suitable subspace. Setting the columns of $\mathbf{U}_\ell$ equal to these Krylov vectors would result in (14) being a good approximation - at least for a single frequency. In subsection 4.2 we discuss how to make a projection matrix fill the full frequency range of interest. The choice for $\mathbf{W}_\ell$ is not as straight forward as the choice for $\mathbf{U}_\ell$. However, since $\mathbf{b}_m$ is computed as the sum of particular columns of the BE matrices, a Galerkin projection, i.e. setting $\mathbf{W}_\ell = \mathbf{U}_\ell$, is natural. In order to make the derivation simpler it is chosen to let $\mathbf{U}_\ell$ be unitary, as this results in

$$\mathbf{b}_{\ell m} \approx \mathbf{U}_\ell^{\mathsf{H}} \mathbf{b}_m, \tag{16}$$

where the H superscript denotes the Hermitian transpose. Applying the projection to (13) it follows that

$$\left(\operatorname{diag}(\boldsymbol{\phi}) + \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \left[\mathbf{F}_m - \frac{(\gamma-1)\mathrm{i}k^2}{k_h} \mathbf{H}_m - \frac{\mathrm{i}}{k_v} \mathbf{T}_m \right] \right) \mathbf{U}_\ell \mathbf{z}_\ell = \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \mathbf{U}_\ell \mathbf{b}_{\ell m}.$$

Now multiplying with $\mathbf{U}_\ell^{\mathsf{H}}$ from the left followed by moving $\mathbf{U}_\ell$ (and $\mathbf{U}_\ell^{\mathsf{H}}$) inside the parentheses the above simplifies to

$$\left(\boldsymbol{\Phi}_\ell + \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \left[\mathbf{F}_{\ell m} - \frac{(\gamma-1)\mathrm{i}k^2}{k_h} \mathbf{H}_{\ell m} - \frac{\mathrm{i}}{k_v} \mathbf{T}_{\ell m} \right] \right) \mathbf{z}_\ell = \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \mathbf{b}_{\ell m}, \tag{17}$$

where $\boldsymbol{\Phi}_\ell, \mathbf{F}_{\ell m}, \mathbf{H}_{\ell m}, \mathbf{T}_{\ell m} \in \mathbb{C}^{\ell \times \ell}$ and $\mathbf{b}_{\ell m} \in \mathbb{C}^\ell$. This approach is reasonable if $\ell M \lessapprox N$, such that storing all the SEBEM matrices is less than or of the order of storing a single of the original BEM matrices.

4.2. *The Arnoldi Algorithm*

In order to make a projection matrix that covers the full frequency band of interest, information from multiple frequencies within this band needs to be combined. The chosen frequencies for which this information is extracted will be referred to as *primary frequencies* (denoted by f_i). At each primary frequency the full BEM system $(\mathbf{A}_{f_i}, \mathbf{b}_{f_i})$ are assembled and the q-Krylov subspace computed using the Arnoldi Algorithm (Algorithm 1). Under the assumption that the solution changes smoothly with frequency the collection of the Krylov vectors from each of the Krylov subspaces can be expected to span the solution space for the full frequency band. The final projection matrix, $\mathbf{U}_\ell$, is then computed as the Singular Value Decomposition (SVD) of the concatenation of the columns of all the q-Krylov vector subspaces

$$\mathbf{U}_\ell \Sigma_\ell \mathbf{V}_\ell^{\mathsf{H}} = \operatorname{svd}\left(\left[\mathbf{K}_{f_1} \, \mathbf{K}_{f_2} \, \ldots \, \mathbf{K}_{f_L} \right] \right). \tag{18}$$

The reason the SVD is important is that the derivation of (17) required that $\mathbf{U}_\ell$ was a unitary matrix which would not be true for the concatenation of the columns of the q-Krylov vector subspaces.

Algorithm 1 Arnoldi Algorithm

Assemble $\mathbf{A}_{f_i}$, $\mathbf{b}_{f_i}$
$\mathbf{k}_1 \leftarrow \mathbf{b}_{f_i}/\|\mathbf{b}_{f_i}\|_2$
$\mathbf{K}_{f_i} \leftarrow \left[\mathbf{k}_1\right]$
for $j = 2 : q$ **do**
  $\mathbf{k}_j \leftarrow \mathbf{A}_{f_i}\mathbf{k}_{j-1} - \sum_{l=1}^{j-1}\left(\mathbf{k}_l^{\mathsf{H}}\mathbf{A}_{f_i}\mathbf{k}_{j-1}\right)\mathbf{k}_l$
  $\mathbf{k}_j \leftarrow \mathbf{k}_j/\|\mathbf{k}_j\|_2$
  $\mathbf{K}_{f_i} \leftarrow \left[\mathbf{K}_{f_i}\ \mathbf{k}_j\right]$
end for

4.3. *Implementation Details*

While the previous sections explain the mathematical background of the computations, a straightforward implementation would result in unnecessary memory allocations, as it would require the full assembly of $\mathbf{F}_m, \mathbf{H}_m$ and $\mathbf{T}_m$ before applying the projection matrices. Instead, only a single row gets assembled at the time, which then gets projected directly (Algorithm 2). This assembly strategy requires only $\mathcal{O}(\ell N)$ memory compared to the standard $\mathcal{O}(N^2)$. While this is an improvement, it is important to note that the overall approach still suffers from the $\mathcal{O}(N^2)$ memory requirement stemming from the computation of the projection matrix $\mathbf{U}_\ell$. This issue could possibly be resolved by utilizing an approximate method such as the Fast Multipole method (FMM) when computing the projection basis, however, this is left for future work as the current study concerns the reduced order method itself.[25]

Algorithm 2 Assembly

Require: $\mathbf{U}_\ell \in \mathbb{C}^{N\times\ell}, k_0 \in \mathbb{R}, M \in \mathbb{N}$
Preallocate $\mathbf{\Phi}, \mathbf{F}, \mathbf{H}, \mathbf{T} \in \mathbb{C}^{N\times\ell}$, $\mathbf{b} \in \mathbb{C}^N$
for $m = 0 : M-1$ **do**
  for $n = 1 : N$ **do**
    if $m = 0$ **then**
      Assemble integral free term for the nth collocation point (denoted by ϕ_n)
      Scale row of $\mathbf{U}_\ell$: $\mathbf{\Phi}_{[n,:]} \leftarrow \phi_n \mathbf{U}_{\ell,[n,:]}$
    end if
    Assemble nth row of $\mathbf{F}_m$, $\mathbf{H}_m$, $\mathbf{T}_m$ and $\mathbf{b}_m$ (denoted by $\mathbf{f}_n^{(m)}$, $\mathbf{h}_n^{(m)}$, $\mathbf{t}_n^{(m)}$, $b_n^{(m)}$)
    Save right-hand side: $b_n \leftarrow b_n^{(m)}$
    Compress columns: $\mathbf{F}_{[n,:]} \leftarrow \mathbf{f}_n^{(m)}\mathbf{U}_\ell$, $\mathbf{H}_{[n,:]} \leftarrow \mathbf{h}_n^{(m)}\mathbf{U}_\ell$, $\mathbf{T}_{[n,:]} \leftarrow \mathbf{t}_n^{(m)}\mathbf{U}_\ell$
  end for
  Compress rows: $\mathbf{F}_{\ell m} \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{F}$, $\mathbf{H}_{\ell m} \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{H}$, $\mathbf{T}_{\ell m} \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{T}$
  Compress rows: $\mathbf{b}_{\ell m} \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{b}$
end for
Compress columns: $\mathbf{\Phi}_\ell \leftarrow \mathbf{U}_\ell^{\mathsf{H}}\mathbf{\Phi}$

5. Numerical Evaluation

In this section the Reduced Order Series Expansion Boundary Element Method (ROSEBEM) including the Boundary Layer Impedance (BLI) boundary condition is applied to two acoustic interior problems with increasing complexity. In particular the first setup is chosen such that the frequency response contain only a single resonance in the frequency range of interest while the frequency response of the second setup contain four resonances. As such the first setup gives insights into various aspects of the computational model while the latter setup highlights what knowledge from the first setup generalizes to more complex setups.
Both test setups are based on an impedance tube which can be used to measure the sound absorption of the material/object placed at the termination end (Figure 3). In a real-world scenario a plane wave is created by a loudspeaker placed opposite of the termination end. This end will be referred to as the source end. In the simulations the loudspeaker excitation will be mimicked by imposing a Neumann boundary condition. The sound properties of what is placed at the termination end is then evaluated by measuring the pressure at the two distinct points (denoted here by $\mathbf{x}_1$ and $\mathbf{x}_2$).

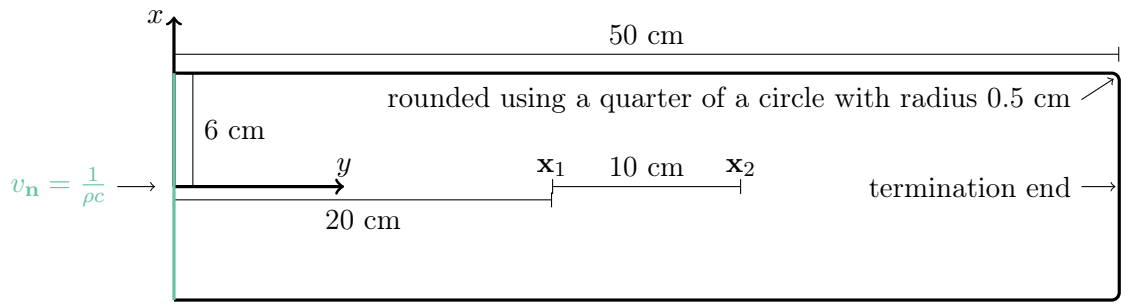


Fig. 3: Two-dimensional slice of the impedance tube (symmetric around the y-axis). The source end, drawn in light green, is prescribed a velocity condition while the remaining boundary, drawn in black, is prescribed the BLI condition.

In the two test cases the termination end will contain resonators for which a good measure of effectiveness is the absorption coefficient,[26]

$$\alpha(k, \mathbf{x}_1, \mathbf{x}_2) = 1 - \left| \frac{\frac{p(\mathbf{x}_1)}{p(\mathbf{x}_2)} - \exp\left(-\mathrm{i}k\|\mathbf{x}_1 - \mathbf{x}_2\|_2\right)}{\frac{p(\mathbf{x}_1)}{p(\mathbf{x}_2)} - \exp(\mathrm{i}k\|\mathbf{x}_1 - \mathbf{x}_2\|_2)} \right|^2, \tag{19}$$

where k is the wavenumber and $p(\mathbf{x}_1)$ and $p(\mathbf{x}_2)$ are computed using (1).

For both test setups the frequency band spans from 200 Hz to 600 Hz and the same Neumann condition of $v_{\mathbf{n}} = \frac{1}{\rho c}$ was applied at the source end while the remaining boundaries, including those of the resonators, were fitted with the Boundary Layer Impedance condition. The two points, $\mathbf{x}_1$ and $\mathbf{x}_2$, were put on the y-axis with respectively a distance of 20 cm and 30 cm to the origin. The main analysis in both setups is the investigation of the importance

of the Krylov subspace. As such it was chosen to truncate the Taylor series at $M = 20$ as this bounds the pointwise error of the Green's function on the order of 10^{-11} making the projection the limiting factor of the accuracy of the ROSEBEM. See Appendix C for a short derivation for the multiple resonator test case. This error bound also holds for the single resonator geometry as the frequency range for both setups is the same and the geometry of the multiple resonator design is larger than the single resonator design.

The simulation code was implemented in the `Julia` programming language and run on the High-Performance Computers provided by the DTU Computing Center (DCC) using `Julia 1.7`[27,28]. The Boundary Element part of the code was inspired by the `OpenBEM` package.[29] The two boundary meshes were created using `COMSOL multiphysics`® and all corners, except those at the source end, were rounded using the `fillet` functionality with a radius of 0.5 cm.

5.1. *Single resonator test case*

The first test case is an impedance tube with a single Helmholtz resonator attached to the center of the termination end. The two tubes making up the Helmholtz resonator have diameters of 2 and 4 cm and lengths 2 and 7.5 cm respectively (Figure 4). The boundary mesh has 2530 isoparametric quadratic elements resulting in a total of 5062 DOF and $h_{\max} \approx 2.9\,\text{cm}$ (Figure 5). In the following the maximum frequency is 600 Hz, meaning that the smallest wavelength is approximately 55 cm. As such the mesh has more than nineteen elements per wavelength for all frequencies of interest, making its size sufficient for its purpose.[30]

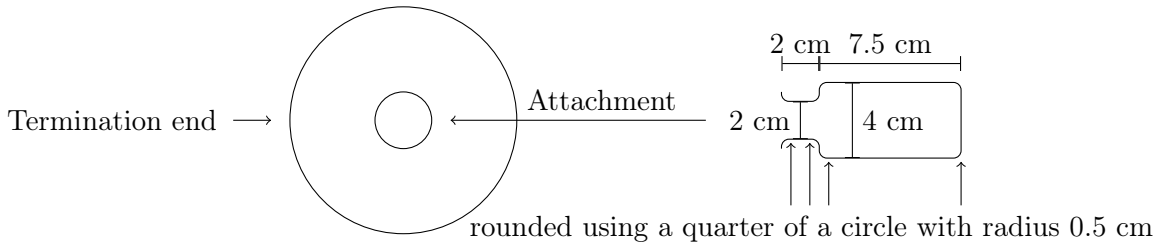


Fig. 4: Left: Resonator placement on the termination end. Right: Two-dimensional cut of the resonator and its dimensions. All corners were rounded using a quarter of a circle with radius 0.5 cm (right).

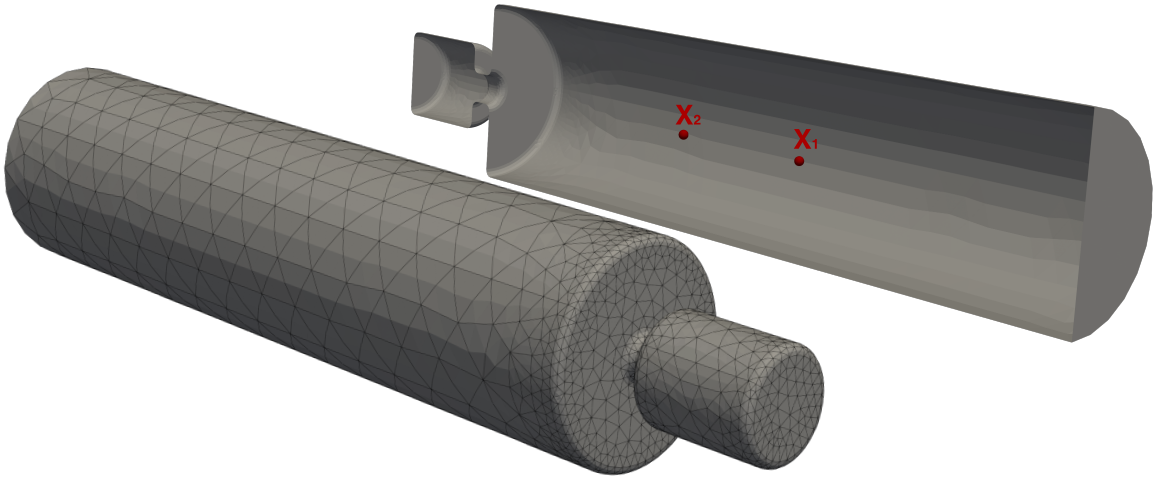


Fig. 5: Three-dimensional visualization of the single resonator setup.

5.1.1. *Creating a Suitable Projection Basis*

The simplest approach when creating a projection basis is to use only the information available at the expansion frequency. From a computational point of view this would be preferable, as this information is readily available from the matrices corresponding to the zeroth order terms in the Taylor expansions. To evaluate this approach three distinct Taylor expansions are created using the three expansions frequencies 200 Hz, 400 Hz and 600 Hz. At every frequency a projection matrix using 170-Krylov vectors of the zeroth order Taylor system is created. This leads to a reduction of the overall DOF of around 97%. Figure 6 show that this approach captures the resonance effects only if the expansion frequency of the Taylor series is chosen sufficiently close to the resonance frequency. For all three setups the error is only small in the close vicinity of their respective the expansion frequency.

Next we examine the influence of adding a single additional frequency when creating the projection matrix. In order to keep the size of the projection matrix constant the 85-Krylov vectors will be computed per primary frequency. Using the same frequencies as before, three distinct projection matrices using two primary frequencies can be computed. Moreover, since the Taylor series can be expanded at either primary frequency, the total number of distinct setups is six. The results show that including information from just one additional frequency results in a good approximation for the full frequency band (Figure 7). Interestingly, as seen from the pairwise identical errors (Figure 7), the same projection gives the same accuracy for different expansion frequencies. This indicates that accuracy of the ROSEBEM, for the single resonator problem, is limited by the resolution of the projection rather than the truncation of the Taylor series.

The errors in Figure 7 are significantly smaller than those previously seen in Figure 6. It can therefore be concluded, for this particular test setup, that it is better to expand around a single frequency and use a more precise projection basis than using two expansion frequencies

with a projection basis computed using only the information available at that frequency.

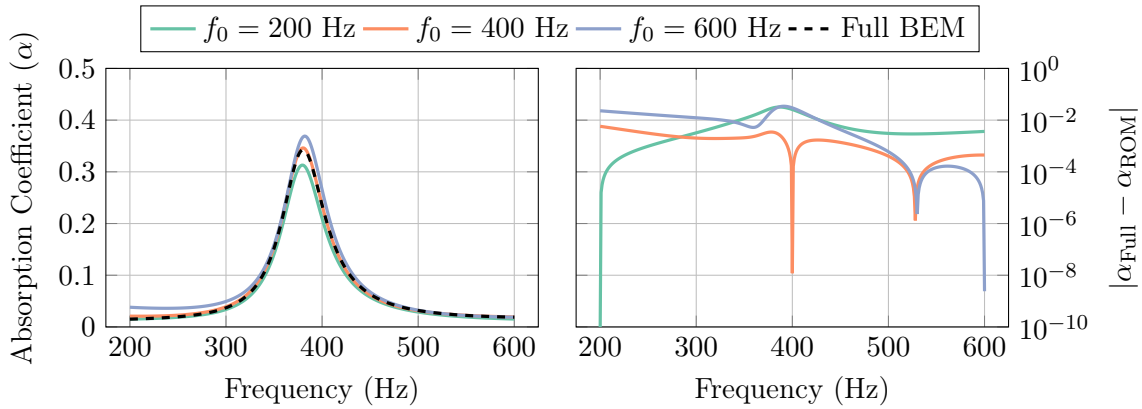


Fig. 6: The absorption coefficient computed using the ROSEBEM with only the information available at the expansion frequencies 200 Hz, 400 Hz & 600 Hz. The ROSBEM expanding close to the resonance is the only model which gives reasonable results (left). The error dip around the expansion frequency is due to the Taylor expansion being exact at these frequencies (right).

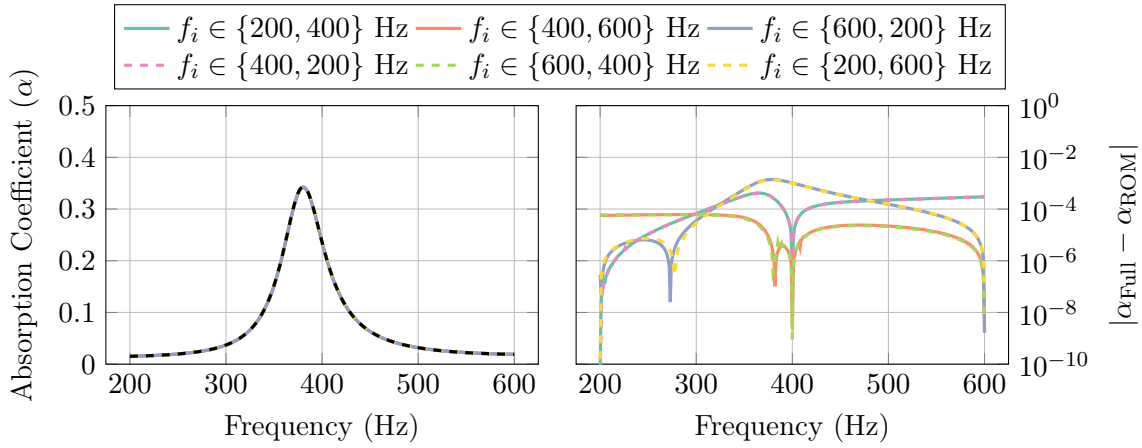


Fig. 7: Legend explanation: $f_i \in \{f_0, f_1\}$ where f_0 is the expansion frequency and f_1 is the additional primary frequency. Using information from two primary frequencies is enough to capture the correct response over the full frequency band (left). The error is dominated by the projection as it is small close to the two primary frequencies (right).

5.2. *Multiple resonator test case*

The following is based on the same impedance tube, but this time four Helmholtz resonators are attached at the termination end (Table 1, Figure 8). The boundary mesh has 4554 isoparametric quadratic elements with 9110 DOF and $h_{\max} \approx 2.9\,\text{cm}$ (Figure 9). The frequency band of interest remains the same as before, meaning that this mesh is sufficient for the frequency range of interest. The Krylov subspace was computed using $\ell \approx 425$, resulting in a reduction of the DOF of around 95%.

Table 1: Dimensions of the four Helmholtz resonators.

	Neck_1	Cavity_1	Neck_2	Cavity_2	Neck_3	Cavity_3	Neck_4	Cavity_4
Length	2 cm	7.5 cm	6 cm	5 cm	8 cm	5 cm	10 cm	5 cm
Radius	1 cm	2 cm	1 cm	2 cm	1 cm	2 cm	1 cm	2 cm

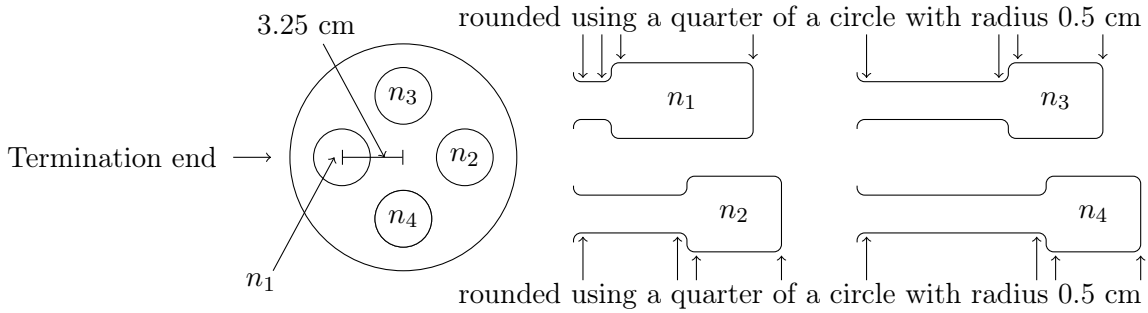


Fig. 8: The resonators are placed along the x and z axes with a distance of 3.25 cm from center to center (left). The dimensions of the two-dimensional cuts of the resonators can be found in Table 1. All corners were rounded using a quarter of a circle with radius 0.5 cm.

5.2.1. *Creating a Suitable Projection Basis*

From the single resonator test case it was seen that expanding close to the resonance frequency resulted in a good fit for the full frequency range. This is not the case for the more complicated geometry. All solutions, no matter the chosen expansion frequency, drift as seen by the error only being small in the close vicinity of the expansion frequency (Figure 10).

Adding information from just a single additional frequency is, however, still enough to create a projection basis with a resolution sufficient to accurately capture the response on the full domain (Figure 11). The errors for the six setups are all on the same scale, with a slight edge to the four setups that includes information from the frequency in the middle of the domain (400 Hz). Additionally, it can be seen that the same projection matrix, but with

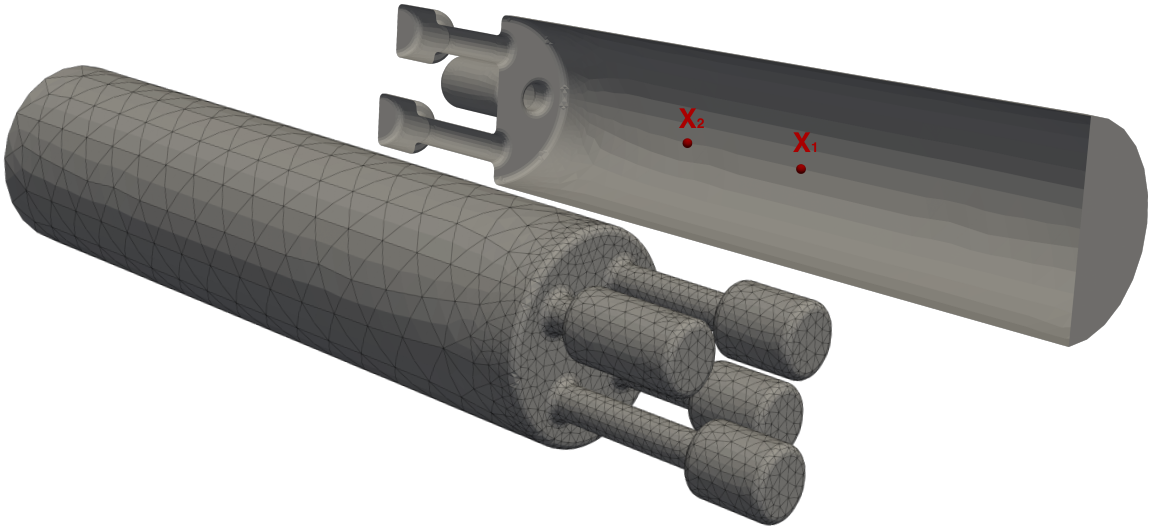


Fig. 9: Three-dimensional visualization of the multiple resonator setup.

different expansion frequencies, result in almost identical errors. The conclusion is henceforth the same as for the simple geometry: The error is bounded by the resolution of the projection matrix and not by the approximation of the Green's function. Given that the dimensions of the two geometries are almost identical, this should not come as a surprise, as the error of the Taylor expansion is bounded by the size of the geometry.

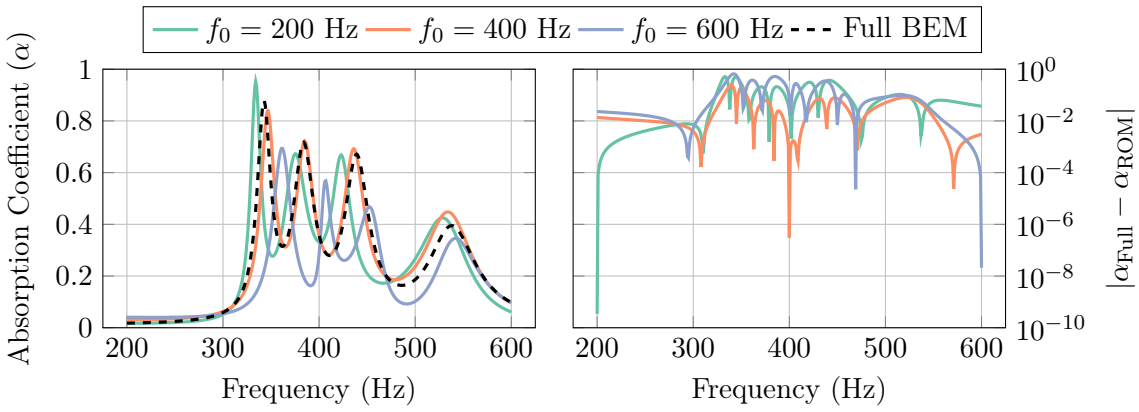


Fig. 10: The absorption coefficient computed using the ROSEBEM with only the information available at the expansion frequencies 200 Hz, 400 Hz & 600 Hz. Not even expanding close to the resonances result in good performance over the full frequency band (left). The absolute error dip around the expansion frequency is due to the Taylor expansion being exact at these frequencies (right).

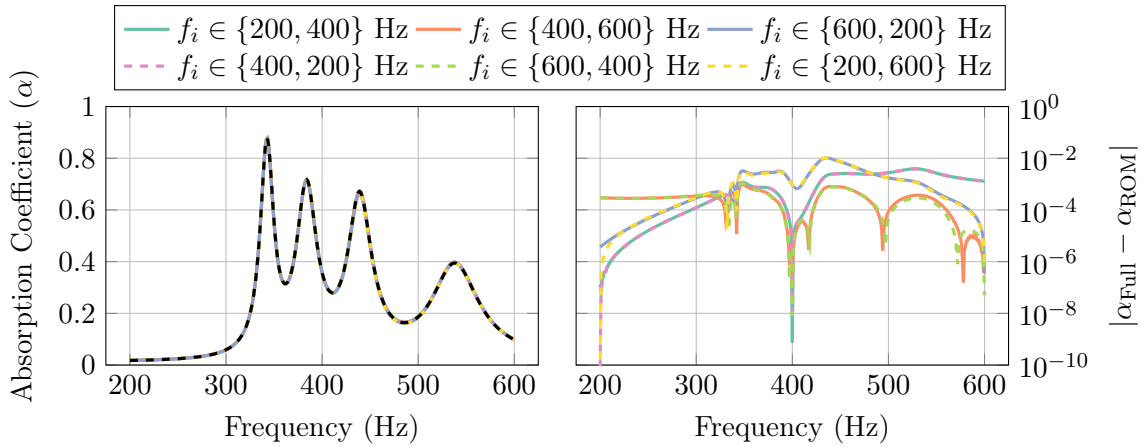


Fig. 11: Legend explanation: $f_i \in \{f_0, f_1\}$ where f_0 is the expansion frequency and f_1 is the additional primary frequency. Using information from two primary frequencies is enough to capture the correct response over the full frequency band (left). The error is still dominated by the projection as the absolute error dip around the two primary frequencies (right).

5.3. *Speedup*

The desirable outcome of using the ROSEBEM as opposed to the full solution is to reduce the total computational time of frequency responses. As such it is germane to figure out a priori when to use either of the two approaches. A simplified analysis of the speedup is as follows

$$\frac{\mathrm{T}_{\text{ROSEBEM}}(N_f)}{\mathrm{T}_{\text{Full}}(N_f)} = \frac{\mathrm{T}_{\text{Krylov}} + \mathrm{T}_{\text{Taylor}}}{(\mathrm{T}_{\text{Full-A}} + \mathrm{T}_{\text{Full-S}})\, N_f} + \frac{(\mathrm{T}_{\text{ROSEBEM-A}} + \mathrm{T}_{\text{ROSEBEM-S}})\, N_f}{(\mathrm{T}_{\text{Full-A}} + \mathrm{T}_{\text{Full-S}})\, N_f}, \tag{20}$$

where N_f denotes the number of frequencies of interest while -A and -S indicates assembly and solution time respectively. From (20) it becomes clear that when $N_f \to \infty$ the speedup tends towards the fraction between the time of assembling and solving the ROSEBEM and the full model respectively. These results are expected since numerous evaluations will make the time spent computing the Krylov vectors and Taylor matrices negligible compared to the frequency sweep itself. To further the analysis two assumptions will be made: The time spent assembling the Krylov vectors is negligible compared to assembling the BEM matrix at the primary frequency and the time spend assembling a single derivative matrix is equal to that of assembling the regular BEM system. Using these assumptions it follows that

$$\frac{\mathrm{T}_{\text{ROSEBEM}}(N_f)}{\mathrm{T}_{\text{Full}}(N_f)} = \frac{\mathrm{T}_{\text{Full-A}}(L+M)}{(\mathrm{T}_{\text{Full-A}} + \mathrm{T}_{\text{Full-S}})N_f} + \frac{3\mathrm{T}_{\ell^2} M + R^3 \mathrm{T}_{\text{Full-S}}}{\mathrm{T}_{\text{Full-A}} + \mathrm{T}_{\text{Full-S}}}, \tag{21}$$

where $R = \frac{\ell}{N}$ is the ratio between the number of DOF in the reduced system and the number of DOF in the original system, L is the number of primary frequencies used and T_{ℓ^2} is the

time to add two $\ell \times \ell$-matrices. For the second test setup ($L = 2$, $M = 20$ and $N_f = 400$) it is found that the speedup is close to 25 (orange, Figure 12). Disregarding the latter term in (21) it can be seen that the cutoff of when to use either the ROSEBEM or the full model will be when $N_f = L + M$. For any other setup the cutoff will lie at a larger N_f and even in some circumstances, when e.g. the reduction of DOF is not large enough and the latter term is larger than 1, the cutoff does not exist.

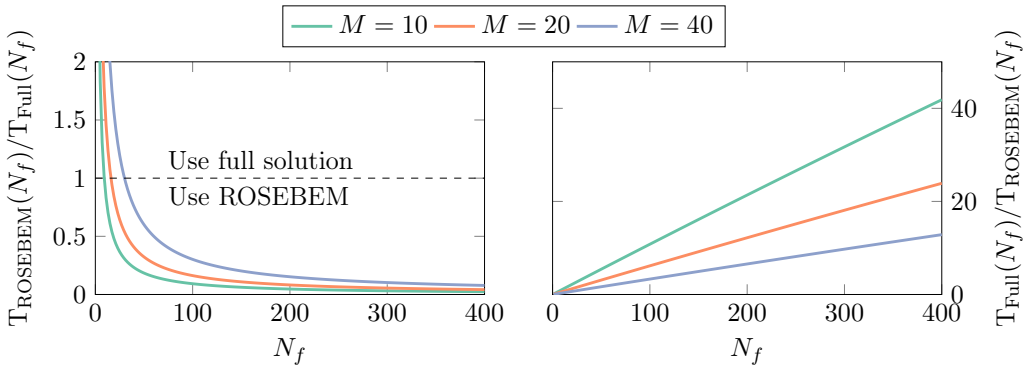


Fig. 12: The cutoff for when to the use the ROSEBEM roughly doubles when M doubles (left). The speedup approximately halves when M doubles (right).

A similar a priori analysis can be performed for the memory usage of the ROSEBEM as compared to the full model. In both cases the memory required to store the right-hand side will not be included as it is negligible compared to the system matrices. Since the assembly of the full system can be implemented such that only a single matrix needs to be stored, while the ROSEBEM requires the storage of three separate matrices, it follows that

$$\frac{\mathrm{MEM_{ROSEBEM}}(N, M, R)}{\mathrm{MEM_{Full}}(N)} = \frac{3(RN)^2 M}{N^2} = 3R^2 M. \tag{22}$$

For the second test setup $R \approx 0.05$ and $M = 20$, meaning that the ROSEBEM used approximately 15% of the memory the full model required (Figure 13).

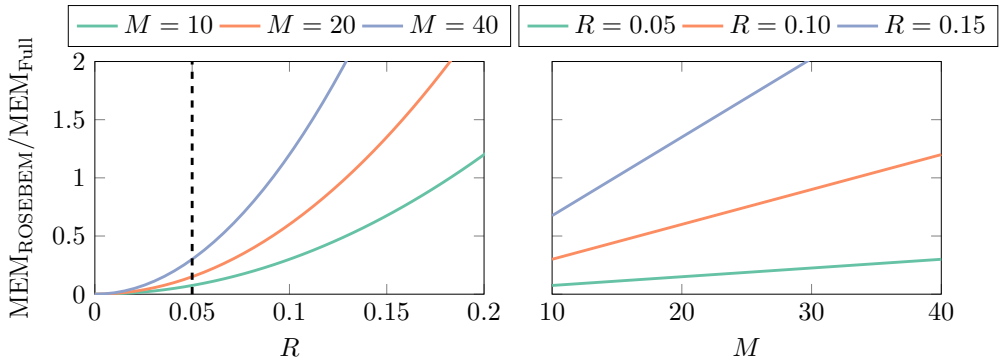


Fig. 13: The memory scaling of the ROSBEM as a function of R and M. The second test setup with $R \approx 0.05$ and $M = 20$ can be seen to have used approximately 15% of the memory of the full model (left). A linear correspondence between the memory scaling and the order of the Taylor expansion can be seen (right).

6. Discussion

In this paper previous ideas of a Reduced Order Series Expansion Boundary Element Method (ROSEBEM) are extended to handle the Boundary Layer Impedance (BLI) boundary condition. Similar to previous works, the model order reduction technique used is the Galerkin projection based on Krylov subspace recycling. The addition of the BLI condition required a derivation of both the tangential gradient of the pressure and the Taylor series of the tangential gradient of the Green's function.

The proposed model was numerically investigated using two geometries representing impedance tubes with resonators attached at their termination end. As such the study concerned interior problems only. The results showed that in order to capture the response accurately in the full frequency band of interest one should create a Galerkin projection matrix based on information from at least two frequencies. Further analysis showed that the error, for both test setups, was dominated by the limitations of the projection rather than the truncation of the Taylor series. As such, it was concluded that it is best to increase the resolution of the projection matrix rather than applying multiple Taylor expansions.

Lastly a short description of the computational and memory scaling of the underlying model, with respect to the various hyperparameters, was presented. Such an analysis is an important a priori tool when applying the ROSEBEM on future problems. In the case of the two numerical investigations presented in this paper the speedup using the ROSEBEM was found to be approximately 25 times while using less than 15% of the memory.

Reference List

References

1. M. Bruneau, *Fundamentals of Acoustics* (Wiley-ISTE, 2010).

2. M. Malinen, M. Lyly, P. Råback, A. Kärkkäinent and L. Kärkkäinen, A finite element method for the modeling of thermo-viscous effects in acoustics, *Eccomas 2004 - European Congress on Computational Methods in Applied Sciences and Engineering* .
3. M. Brunaeu, P. Herzog, J. Kergomard and J. Polack, General formulation of the dispersion equation in bounded visco-thermal fluid, and application to some simple geometries, *Wave Motion* **11** (1989) 441–451.
4. V. Cutanda Henriquez and P. R. Andersen, A three-dimensional acoustic Boundary Element Method formulation with viscous and thermal losses based on shape function derivatives, *Journal of Computational Acoustics* **26**.
5. M. Berggren, A. Bernland and D. Noreland, Acoustic boundary layers as boundary conditions, *Journal of Computational Physics* **371** (2018) 633–650.
6. M. J. Cops, J. G. McDaniel, E. A. Magliula, D. J. Bamford and M. Berggren, Estimation of acoustic absorption in porous materials based on visco-thermal boundary layers modeled as boundary conditions, *Journal of the Acoustical Society of America* **148** (2020) 1624–1635.
7. P. R. Andersen, V. Cutanda Henriquez and N. Aage, On the validity of numerical models for viscothermal losses in structural optimization for micro-acoustics, *Journal of Sound and Vibration* **547** (2022) 117455.
8. S. K. Baydoun, M. Voigt and S. Marburg, Low-rank Iteration Schemes for the Multi-Frequency Solution of Acoustic Boundary Element Equations, *Journal of Theoretical and Computational Acoustics* **29**.
9. S. Kirkup and D. Henwood, Methods For Speeding Up the Boundary Element Solution of Acoustic Radiation Problems, *Journal of Vibration and Acoustics-transactions of the Asme* **114** (1992) 374–380.
10. D. Panagiotopoulos, E. Deckers and W. Desmet, A two-step reduction method for acoustic BEM systems, in *Forum Acusticum* (Lyon, France, 2020), pp. 2291–2298.
11. D. Panagiotopoulos, E. Deckers and W. Desmet, Krylov subspaces recycling based model order reduction for acoustic bem systems and an error estimator, *Computer Methods in Applied Mechanics and Engineering* **359** (2020) 112755.
12. T. W. Wu, W. L. Li and A. F. Seybert, An efficient boundary element algorithm for multi-frequency acoustical analysis, *Journal of the Acoustical Society of America* **94** (1993) 447–452.
13. D. Jiao, X. Y. Zhu and J. M. Jin, Fast and accurate frequency-sweep calculations using asymptotic waveform evaluation and the combined-field integral equation, *Radio Science* **34** (1999) 1055–1063.
14. U. Hetmaniuk, R. Tezaur and C. Farhat, Review and assessment of interpolatory model order reduction methods for frequency response structural dynamics and acoustics problems, *International Journal for Numerical Methods in Engineering* **90** (2012) 1636–1662.
15. Q. Li, O. Sigmund, J. S. Jensen and N. Aage, Reduced-order methods for dynamic problems in topology optimization: A comparative study, *Computer Methods in Applied Mechanics and Engineering* **387**.
16. V. C. Henriquez, Numerical transducer modeling, Ph.D. thesis, Technical University of Denmark, 2002.
17. S. Kirkup, The Boundary Element Method in Acoustics: A Survey, *Applied Sciences* **9** (2019) 1642.
18. Y. Liu, On the BEM for acoustic wave problems, *Engineering Analysis With Boundary Elements* **107** (2019) 53–62.
19. N. Atalla and F. Sgard, *Finite Element and Boundary Methods in Structural Acoustics and Vibration* (CRC Press, 2015).
20. S. Marburg and B. Nolte, *Computational Acoustics of Noise Propagation in Fluids - Finite and Boundary Element Methods* (Springer Berlin Heidelberg, 2008).

21. S. Preuss, C. Gurbuz, C. Jelich, S. K. Baydoun and S. Marburg, Recent Advances in Acoustic Boundary Element Methods, *Journal of Theoretical and Computational Acoustics* **30** (2022) 2240002.
22. S. Marburg, The Burton and Miller Method: Unlocking Another Mystery of Its Coupling Parameter, *Journal of Computational Acoustics* **24** (2016) 1550016.
23. P. D. Lax and M. S. Terrell, *Multivariable Calculus with Applications* (Springer, 2017).
24. X. Xie and Y. Liu, An adaptive model order reduction method for boundary element-based multi-frequency acoustic wave problems, *Computer Methods in Applied Mechanics and Engineering* **373**.
25. L. Yijun, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering* (Cambridge University Press, 2009).
26. Iso 10534-2: 1998, accoustics - determination of sound absorption coefficient and impedance tubes part 2: Transfer function method, .
27. J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah, Julia: A Fresh Approach to Numerical Computing, *SIAM Review* **59** (2017) 65–98.
28. DTU Computing Center, DTU Computing Center resources, 2021.
29. V. C. Henríquez and P. M. Juhl, OpenBEM - An open source Boundary Element Method software in Acoustics, *Proceedings of Internoise 2010* **7** (2010) 5796–5805.
30. S. Marburg, Six boundary elements per wavelength: Is that enough?, *Journal of Computational Acoustics* **10** (2002) 25–51.

Appendix

A. List of Taylor Expansions of Relevant Kernels

In the following $r = \|\mathbf{x} - \mathbf{y}\|_2$, k_0 denotes the expansion wavenumber and α denotes the sign of the chosen harmonic time-dependency. Previously in the text α was set to 1.

A.1. *Taylor Expansion of Green's function*

$$G(\mathbf{x},\mathbf{y}) \approx \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} G^{(m)}(\mathbf{x},\mathbf{y},k_0) = \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \left[\frac{(-\alpha \mathrm{i} r)^m \exp\left(-\alpha \mathrm{i} k_0 r\right)}{4\pi r} \right] \quad \text{(A.1)}$$

A.2. *Taylor Expansion of Gradient of Green's function*

$$\nabla_{\mathbf{y}} G(\mathbf{x},\mathbf{y}) \approx \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \nabla_{\mathbf{y}} G^{(m)}(\mathbf{x},\mathbf{y},k_0), \quad \text{(A.2)}$$

where $\nabla_{\mathbf{y}} G^{(m)}(\mathbf{x},\mathbf{y},k)$ is the gradient of the mth term in (A.1) which can be computed as

$$\begin{aligned}
\nabla_{\mathbf{y}} G^{(m)}(\mathbf{x},\mathbf{y},k_0) &= \nabla_{\mathbf{y}} \left((-\alpha \mathrm{i} r)^m G(\mathbf{x},\mathbf{y},k_0)\right) \\
&= (-\alpha \mathrm{i})^m \nabla_{\mathbf{y}} \left(r^m\right) G(\mathbf{x},\mathbf{y},k_0) + (-\alpha \mathrm{i} r)^m \nabla_{\mathbf{y}} G(\mathbf{x},\mathbf{y},k_0) \\
&= (-\alpha \mathrm{i} r)^m \left(\frac{m}{r} \nabla_{\mathbf{y}} \left(r\right) \frac{\exp\left(-\alpha \mathrm{i} k_0 r\right)}{4\pi r} + \frac{(\mathbf{x}-\mathbf{y})\left(1+\alpha \mathrm{i} k_0 r\right)\exp\left(-\alpha \mathrm{i} k_0 r\right)}{4\pi r^3} \right) \\
&= (-\alpha \mathrm{i} r)^m \left(\frac{(\mathbf{y}-\mathbf{x}) m \exp\left(-\alpha \mathrm{i} k_0 r\right)}{4\pi r^3} + \frac{(\mathbf{x}-\mathbf{y})(1+\alpha \mathrm{i} k_0 r)\exp\left(-\alpha \mathrm{i} k_0 r\right)}{4\pi r^3} \right) \\
&= \frac{(\mathbf{x}-\mathbf{y})(-\alpha \mathrm{i} r)^m (1+\alpha \mathrm{i} k_0 r - m)\exp\left(-\alpha \mathrm{i} k_0 r\right)}{4\pi r^3}
\end{aligned}$$

A.3. *Taylor Expansion of normal derivative of Green's function*

$$\frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} \approx \sum_{m=0}^{M-1} \left[\frac{(k-k_0)^m}{m!} \mathbf{n}(\mathbf{y})^\top \nabla_{\mathbf{y}} G^{(m)}(\mathbf{x},\mathbf{y},k_0) \right] \tag{A.3}$$

A.4. *Taylor Expansion of tangential derivative of Green's function*

$$\nabla_{\mathbf{y}}^{\parallel} G(\mathbf{x},\mathbf{y}) \approx \sum_{m=0}^{M-1} \left[\frac{(k-k_0)^m}{m!} \left(\mathbf{I} - \mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y})^\top \right) \nabla_{\mathbf{y}} G^{(m)}(\mathbf{x},\mathbf{y},k_0) \right] \tag{A.4}$$

B. List of Integrals

On the surface the chosen interpolation of the pressure and its normal derivative is given by

$$p(\mathbf{x}) = \mathbf{B}(\mathbf{x})^\top \mathbf{p}, \quad \frac{\partial p}{\partial \mathbf{n}}(\mathbf{x}) = \mathbf{B}(\mathbf{x})^\top \frac{\partial \mathbf{p}}{\partial \mathbf{n}}, \quad \mathbf{x} \in \Gamma. \tag{B.1}$$

This means that $\mathbf{B}_j(\mathbf{x})$ is the global basis function for the jth node. As such the integrals of (12) can be computed as

$$[\mathbf{G}_m]_{ij} = \int_{\Gamma_{\mathrm{N}}} G^{(m)}(\mathbf{x}_i,\mathbf{y},k_0) \mathbf{B}_j(\mathbf{y})^\top \, \mathrm{d}S_{\mathbf{y}}, \tag{B.2}$$

$$[\mathbf{F}_m]_{ij} = \int_{\Gamma} \left[\mathbf{n}(\mathbf{y})^\top \nabla_{\mathbf{y}} G^{(m)}(\mathbf{x},\mathbf{y},k_0) \right] \mathbf{B}_j(\mathbf{y})^\top \, \mathrm{d}S_{\mathbf{y}}, \tag{B.3}$$

$$[\mathbf{H}_m]_{ij} = \int_{\Gamma_{\mathrm{BLI}}} G^{(m)}(\mathbf{x}_i,\mathbf{y},k_0) \mathbf{B}_j(\mathbf{y})^\top \, \mathrm{d}S_{\mathbf{y}}, \tag{B.4}$$

$$[\mathbf{T}_m]_{ij} = \int_{\Gamma_{\mathrm{BLI}}} \left[\left(\mathbf{I} - \mathbf{n}(\mathbf{y})\mathbf{n}(\mathbf{y})^\top \right) \nabla_{\mathbf{y}} G^{(m)}(\mathbf{x},\mathbf{y},k_0) \right]^\top \nabla_{\mathbf{y}}^{\parallel} \mathbf{B}_j(\mathbf{y})^\top \, \mathrm{d}S_{\mathbf{y}}, \tag{B.5}$$

where $\nabla^{\parallel}\mathbf{B}_j(\mathbf{y})^{\top}$ can be computed using the interpolation function derivative described in (11) on each element. Note that while each integral of (B.2)-(B.5) is defined on the full part of their respective boundaries the integrals need only to be computed where $\mathbf{B}_j(\mathbf{y})$ has support.

C. Error bounds of Taylor series

The Lagrange remainder theorem of the Taylor series says that the pointwise error is bounded as[11]

$$\left|\frac{(r_{\max})^M}{M!}(k-k_0)^M\right|, \tag{C.1}$$

where $k \in [k_{\min}, k_{\max}]$, $r_{\max}$ is the diameter of the smallest sphere containing the geometry and $M-1$ is the order of the Taylor series. To investigate the maximum error we must pick k the furthest away from k_0 while finding a bound on $r_{\max}$. The first part is solved by setting $k = k_{\max}$ as k_0 is set in the middle of the frequency domain for both cases. For $r_{\max}$ we see that the longest resonator has length 15 cm meaning that the maximum distance can be bounded as

$$r_{\max} \leq \sqrt{(0.50\,\mathrm{m} + 0.15\,\mathrm{m})^2 + 0.12\,\mathrm{m}^2} \approx 0.66\,\mathrm{m}. \tag{C.2}$$

Using this one find that

$$\left|\frac{(0.66)^{20}}{20!}(11.09 - 7.39)^{20}\right| \leq 2.33 \cdot 10^{-11}. \tag{C.3}$$

Paper J2

Journal of Theoretical and Computational Acoustics

Revising the Boundary Element Method for Thermoviscous Acoustics: An Iterative Approach via Schur Complement

Simone Preuss[a,*], Mikkel Paltorp[b,†], Alexis Blanc[a], Vicente Cutanda Henríquez[b], Steffen Marburg[a]

[a] *Chair of Vibroacoustics of Vehicles and Machines*
Department of Engineering Physics and Computation
TUM School of Engineering and Design
Technical University of Munich
Boltzmannstraße 15, 85748 Garching, Germany
**simone.preuss@tum.de*

[b] *Acoustic Technology Group*
Department of Electrical and Photonics Engineering
The Technical University of Denmark
Kgs. Lyngby, DK 2800, Denmark
[†] *mpasc@dtu.dk*

Received (Day Month Year)
Revised (Day Month Year)

The Helmholtz equation is a reliable model for acoustics in inviscid fluids. Real fluids, however, experience viscous and thermal dissipation that impact the sound propagation dynamics. The viscothermal losses primarily arise in the boundary region between the fluid and solid, the acoustic boundary layers. To preserve model accuracy for structures housing acoustic cavities of comparable size to the boundary layer thickness, meticulous consideration of these losses is essential. Recent research efforts aim to integrate viscothermal effects into acoustic boundary element methods (BEM). While the reduced discretization of BEM is advantageous over finite element methods, it results in fully populated system matrices whose conditioning deteriorates when extended with additional degrees of freedom to account for viscothermal dissipation. Solving such a linear system of equations becomes prohibitively expensive for large-scale applications, as only direct solvers can be used. This work proposes a revised formulation for the viscothermal BEM employing the Schur complement and a change of basis for the boundary coupling. We demonstrate that static condensation significantly improves the conditioning of the coupled problem. When paired with an iterative solution scheme, the approach lowers the algorithmic complexity and thus reduces the computational costs in terms of runtime and storage requirements. The results demonstrate the favorable performance of the new method, indicating its usability for applications of practical relevance in thermoviscous acoustics.

Keywords: boundary element method; acoustic boundary layers; viscous and thermal losses; Schur complement; static condensation; iterative solvers

1. Introduction

Standard numerical methods in acoustics are typically based on the wave equation and its time-harmonic counterpart, the Helmholtz equation. The derivation of these equations from the set of conservation laws employs simplifications and idealizes the physical field as inviscid and adiabatic.[1] While restricting the model to a linear stationary regime covers a variety of

applications, recent studies question the general disregard of dissipative effects.[2–8] Acoustic waves traveling through fluids are affected by viscous and thermal losses. The viscothermal losses manifest in the bulk of the fluid cavity and in the vicinity of its solid frame in the acoustic boundary layers. The boundary layers form due to viscous friction hindering the sound waves to freely propagate and due to heat exchange between the adiabatic bulk and isothermal wall.[9] Whereas the bulk losses only take effect on propagation over long distances, the relative importance of the boundary losses varies with the properties of the propagation medium, the characteristic size of the fluid domain, and the wavelength.[10] For air in a frequency range audible by humans, the viscous and thermal boundary layers comprise a region spanning from a few micrometers to a fraction of a millimeter.[11] Hence, the dissipation effects are small, and omitting them usually does not notably reduce the model accuracy. However, for applications such as hearing aids, condenser and MEMS (micro-electromechanical system) microphones, or small-scale acoustic metamaterials, the fluid enclosures become similarly sized as the boundary layers. The viscothermal damping consequently dominates the acoustic field and needs to be included in the modeling process.

In order to numerically analyze the aforementioned test cases, research efforts brought forth finite (FEM)[12,13] and boundary element methods (BEM)[11,14,15] that fully incorporate the dissipation in the bulk and boundary layers. Moreover, simplified models were proposed, cf. [16–21] These models resolve the losses in an approximate and therefore efficient manner but are only suited for specific geometries, frequency ranges, or boundary layer constitutions.[22] Whereas volumetric domain methods like the lossy FEM offer extensive modeling capabilities, they also prove computationally demanding. To capture the small-scale dissipation effects, fine meshing of the associated boundary layers is required.[12] In addition to the large mesh, each discretization point comes with extra degrees of freedom (DOFs) for the temperature- and viscosity-related quantities, resulting in ten times more unknowns than for a conventional acoustic problem. The boundary element method, however, only requires discretization of the surfaces bounding the fluid domain.[1,14] This avoids costly meshing of the boundary layers and reduces the overall problem size. Furthermore, the BEM formulation implicitly satisfies the far-field radiation condition, making it a good fit for exterior acoustics.[23,24]

Starting from the linearized Navier-Stokes, continuity, and Kirchhoff-Fourier equations describing the conservation of momentum, mass, and energy,[25,26] Cutanda Henríquez et al.[14] developed a first direct collocation BEM including the viscous and thermal effects. While this early version of the viscothermal BEM was tailored to axisymmetric geometries, later publications generalize the concept for arbitrary two- and three-dimensional problems.[11,15] The state-of-the-art lossy boundary element methods have shown promising results when applied to practical problems but still suffer from shortcomings rooted in the inherent properties of the BEM.[24] Restricting the discretization process to the boundary comes at the cost of dense non-symmetric coefficient matrices.[1] The assembly of those matrices has a time and space complexity of quadratic order. The computational costs of the algorithm unfold even worse when considering the increased number of matrix coefficients stemming from the

viscothermal degrees of freedom. As a first remedy, Andersen et al.[8] suggested exploiting the spatially limited significance of the viscothermal effects by truncating the Green's function at a distance of twenty times the boundary layer thicknesses. As a result, the coefficient matrices for the viscous and thermal wave fields become sparse. The sparsity, however, cannot be further utilized when plugged into the coupled global system. The matrix blocks show vastly different characteristics that cause the system to be ill-posed. Thus, solving the linear system of equations by means of classical iterative solvers like the generalized minimal residual method (GMRes)[27–29] proves impossible. Instead, the system is solved directly. When using an LU - factorization, the asymptotic solution complexity with respect to execution time and memory usage stands as $\mathcal{O}\left(N^3\right)$ and $\mathcal{O}\left(N^2\right)$, respectively, with N being the dimension of the discrete boundary problem.[30] As the problem dimensions increase, the cubic time and quadratic memory effort of the dense factorization makes it prohibitively costly to handle large-scale viscothermal problems.

In order to alleviate the computational burden, we propose a revised version of the viscothermal BEM. First, a change of basis for the coupling conditions is introduced. By defining the no-slip and null-divergence equations in global coordinates, the formulation for the lossy BEM becomes simpler and less cumbersome to implement. To counteract the ill-conditioning, we further apply the Schur complement and condense the formulation sequentially until solely the acoustic pressure remains as an unknown. The resulting system is well conditioned and of reduced size, thus avoiding the necessity for direct solvers while requiring less computational effort for the solution. Furthermore, separation of the dense acoustic matrices from the sparse viscothermal terms sets the way for future use of fast BEM methods. Finally, the new formulation is evaluated through a simple academic test case comparing it to an analytical reference model and the original formulation from Cutanda Henríquez et al.[11]

2. Viscothermal Boundary Element Formulation

The present work relies on a previously published three-dimensional BEM with losses. An in-depth description can be found in Ref. 11. This section gives a short overview of the underlying formulation, focusing on the issues motivating this paper.

2.1. *Kirchhoff Decomposition and Coupling Conditions*

Unlike the finite element method, the BEM cannot directly evaluate the fundamental equations governing the acoustic wave propagation in the presence of viscosity and thermal conduction. Instead, Kirchhoff's dispersion relation[26] and its later extension[25] is adopted. Kirchhoff proposed to decompose the physical problem into three modal wave fields coupled solely on the domain boundary. The acoustic, entropy, and vorticity mode, which correspond to the perturbation of the scalar-valued acoustic and thermal pressure p_a and p_h and the vector-valued viscous velocity $\mathbf{v}_v$, each satisfies a continuous equation of the Helmholtz form. The set of partial differential equations (PDEs) for linear time-harmonic acoustics can be

written as

$$\Delta p_a(\mathbf{x}) + k_a^2 p_a(\mathbf{x}) = 0\,, \tag{1}$$
$$\Delta p_h(\mathbf{x}) + k_h^2 p_h(\mathbf{x}) = 0\,, \tag{2}$$
$$\Delta \mathbf{v}_v(\mathbf{x}) + k_v^2 \mathbf{v}_v(\mathbf{x}) = \mathbf{0} \quad \text{with} \quad \mathbf{x} \in \Omega\,, \tag{3}$$

where $\Omega \subset \mathbb{R}^d$ is the computational domain. Whereas Eq. (1) models the sound propagation, the latter two cover the thermal and viscous diffusion, respectively. The complex-valued modal wavenumbers k_a, k_h, and k_v are functions of the isentropic wavenumber k and the fluid properties of the propagation medium. This includes, among others, the thermal conductivity, specific heat capacity, and the shear and bulk viscosity.[25] The formulas for the wavenumbers are outlined in Appendix A. For simplicity reasons, we omit sources that might be added as extra terms to the right-hand sides of Eqs. (1)-(3). Superposition of the modal contributions yields the total velocity and pressure in the fluid as

$$p(\mathbf{x}) = p_a(\mathbf{x}) + p_h(\mathbf{x})\,, \tag{4}$$
$$\mathbf{v}_f(\mathbf{x}) = \mathbf{v}_a(\mathbf{x}) + \mathbf{v}_h(\mathbf{x}) + \mathbf{v}_v(\mathbf{x}) \quad \text{with} \quad \mathbf{x} \in \Omega\,. \tag{5}$$

Further introduction of coupling conditions on the boundary Γ enables the solution of the PDEs with a direct collocation BEM. Whereas the viscous coupling is usually realized by a no-slip condition,

$$\mathbf{v}_b(\mathbf{x}) = \mathbf{v}_f(\mathbf{x}) = \phi_a \nabla p_a(\mathbf{x}) + \phi_h \nabla p_h(\mathbf{x}) + \mathbf{v}_v(\mathbf{x}) \quad \text{with} \quad \mathbf{x} \in \Gamma\,, \tag{6}$$

matching the boundary velocities in the fluid $\mathbf{v}_f$ and structure $\mathbf{v}_b$, the thermal coupling is accounted for by an isothermal condition as

$$T_b(\mathbf{x}) = \tau_a p_a(\mathbf{x}) + \tau_h p_h(\mathbf{x}) = 0 \quad \text{with} \quad \mathbf{x} \in \Gamma\,. \tag{7}$$

The latter assumes that due to the higher heat capacity and thermal conductivity of the structure bounding the fluid domain, any temperature fluctuation T associated with the traveling acoustic wave vanishes at the boundary (subscript b).[25] Similar to the wavenumbers in Eqs. (1)-(3), the coefficients τ_a, τ_h, ϕ_a, and ϕ_h represent constant complex-valued parameters that depend on the frequency and fluid properties.[14] Calculation of the parameters follows the expressions given in Appendix A. Furthermore, the null-divergence characteristic of the viscous velocity, which marks the rotational part of the total velocity field, is exploited. To compute the divergence, a dot product, denoted by the "$\cdot$" operator, is performed. Setting the divergence to zero as

$$\nabla \cdot \mathbf{v}_v(\mathbf{x}) = 0 \quad \text{with} \quad \mathbf{x} \in \Omega\,, \tag{8}$$

and using it as a coupling condition on Γ completes the full description of the viscothermal acoustic field.

2.2. *Acoustic Boundary Element Method*

Except for a few academic test cases, the coupled system of equations cannot be solved analytically. Instead, numerical methods are tasked with finding a discrete approximation to the solution.[31] In order to apply the boundary element method to the viscothermal Helmholtz problem, the governing PDEs from Eqs. (1)-(3) are converted to boundary integral equations (BIEs). Taking the acoustical mode as an example, the integral formulation reads

$$c(\mathbf{y})p_a(\mathbf{y}) + \int_\Gamma \frac{\partial G(k_a, \mathbf{x}, \mathbf{y})}{\partial n(\mathbf{x})} p_a(\mathbf{x}) d\Gamma(\mathbf{x}) = \int_\Gamma G(k_a, \mathbf{x}, \mathbf{y}) \frac{\partial p_a(\mathbf{x})}{\partial n(\mathbf{x})} d\Gamma(\mathbf{x}) \,, \tag{9}$$

where $\mathbf{x} \in \Gamma$ and $\mathbf{y} \in \Gamma$ denote the field and source points, respectively, and n is the normal direction.[1] We establish the normal vector to point outward from domain Ω. The constant $c(\mathbf{y}) \in [0, 1]$ as part of the integral-free term depends on the boundary geometry and positioning of $\mathbf{y}$ relative to the computational domain. Equally to the conventional acoustic BEM, the integral kernel is built from the Green's function G and its normal derivative $\partial_n G$. Here, the shorthand notation $\partial_\bullet$ replaces a partial derivative defined as $\frac{\partial}{\partial \bullet}$. Assuming a harmonic time dependence of $\mathrm{e}^{-\mathrm{i}\omega t}$ with ω denoting the angular frequency, the fundamental solution for the 3D acoustic Helmholtz operator is given by

$$G(k_a, \mathbf{x}, \mathbf{y}) = \frac{1}{4\pi} \frac{\mathrm{e}^{\mathrm{i}k_a r(\mathbf{x},\mathbf{y})}}{r(\mathbf{x}, \mathbf{y})} \quad \text{with} \quad r = \|\mathbf{x} - \mathbf{y}\|_2 \,, \tag{10}$$

in which r represents the Euclidean distance.[1] The same procedure is applied to the governing equations (2) and (3) describing the thermal pressure and viscous velocity. The resulting two BIEs are of the same form as Eq. (9) but carry k_h and k_v as wavenumbers in their kernel functions.[14] For solving the system numerically, the continuous integral equations are discretized using the collocation approach. The physical quantities are formulated in terms of interpolation polynomials. Although not a requirement, we utilize the same set of Lagrangian polynomials to approximate all unknowns. Furthermore, the nodes for the piece-wise approximation coincide with the collocation points. Division of the geometrical boundary into surface elements sized according to the problem characteristics eventually allows calculating the integrals via a Gaussian quadrature scheme.[1] In matrix notation, the BEM formulation for the acoustic, thermal, and viscous mode are written as

$$\mathbf{H}_a \mathbf{p}_a - \mathbf{G}_a \, \partial_n \mathbf{p}_a \, = \mathbf{0} \,, \tag{11}$$

$$\mathbf{H}_h \mathbf{p}_h - \mathbf{G}_h \, \partial_n \mathbf{p}_h = \mathbf{0} \,, \tag{12}$$

$$\mathbf{H}_v \mathbf{v}_v - \mathbf{G}_v \, \partial_n \mathbf{v}_v \, = \mathbf{0} \,. \tag{13}$$

The variables $\mathbf{p}_a$, $\mathbf{p}_h$, and $\mathbf{v}_v$ and their derivative counterparts constitute arrays holding the nodal results - one for each of the n collocation points.[11] The matrices $\mathbf{G}_\bullet$ and $\mathbf{H}_\bullet$ replace the discretized integrals on the right-hand and left-hand side of the BIEs, also known as single layer and double layer potentials, respectively.[1] The latter include the contribution from the integral free term. As the velocity has three directional components, defined in

global coordinates as

$$\mathbf{v}_v = \begin{bmatrix} \mathbf{v}_{v,x} \\ \mathbf{v}_{v,y} \\ \mathbf{v}_{v,z} \end{bmatrix}, \quad \partial_n \mathbf{v}_v = \begin{bmatrix} \partial_n \mathbf{v}_{v,x} \\ \partial_n \mathbf{v}_{v,y} \\ \partial_n \mathbf{v}_{v,z} \end{bmatrix}, \tag{14}$$

the coefficient matrices in Eq. (13) represent block-diagonal matrices of size $3n \times 3n$ built from the component-wise (superscript c) viscous potentials as $\mathbf{G}_v = \text{blkdiag}(\mathbf{G}_v^c, \mathbf{G}_v^c, \mathbf{G}_v^c)$ and $\mathbf{H}_v = \text{blkdiag}(\mathbf{H}_v^c, \mathbf{H}_v^c, \mathbf{H}_v^c)$. Further discretization of the coupling conditions yields the final linear system of equations solvable by means of a direct solver. Whereas the isothermal boundary coupling from Eq. (7) appears straight-forward,

$$\mathbf{p}_a \tau_a + \mathbf{p}_h \tau_h = \mathbf{0}\,, \tag{15}$$

the no-slip (Eq. (6)) and null-divergence (Eq. (8)) equations contain gradients that need further clarification and pose a challenge for the viscothermal BEM setup.

2.3. *Previous Developments*

In previous publications on the viscothermal BEM, both coupling conditions were defined in a nodal coordinate basis such that the included differential operator conveniently connects to the ∂_n - unknowns from the discrete BIEs in Eqs. (11)-(13), cf.[11,14,15] When split into its vectorial components, the no-slip boundary condition states

$$v_{b,t} = \phi_a\, \partial_t p_a + \phi_h\, \partial_t p_h + v_{v,t}\,, \tag{16}$$

$$v_{b,s} = \phi_a\, \partial_s p_a + \phi_h\, \partial_s p_h + v_{v,s}\,, \tag{17}$$

$$v_{b,n} = \phi_a\, \partial_n p_a + \phi_h\, \partial_n p_h + v_{v,n}\,. \tag{18}$$

Similarly, the component-wise notation of the null-divergence characteristic is given by

$$0 = \partial_t v_{v,t} + \partial_s v_{v,s} + \partial_n v_{v,n}\,. \tag{19}$$

Here, the subscripts (t, s, n) indicate the contributions in first and second tangential and in normal direction, respectively. While early implementations of the lossy BEM calculated the surface tangential derivatives via central[14] or Voronoi cell finite differences[32] that are cumbersome to implement and numerically unstable for nearly uniform pressure distributions, the latest version circumvents the critical differentiation by utilizing analytical derivatives of the interpolation functions.[11] Thus, the ∇ - operator is shifted from the physical quantities to their piece-wise approximation.

2.3.1. *Interpolation Function Derivatives in Global and Local Space*

In the following, a detailed description of the interpolation function derivatives (IFDs) is given since a change of formulation will be proposed later on. As the interpolation in the boundary element method is conducted on surface elements, the interpolation functions are defined in two-dimensional space. As a consequence, differentiating the basis functions

only yields the in-plane component of the gradient. The full gradient is then computed by combining the tangential information from the IFDs with the normal derivatives from the discrete BE formulation.

To compute the boundary integral at any position $\mathbf{x} = \begin{bmatrix} x\ y\ z \end{bmatrix}^\top$ all continuously defined variables are reformulated in terms of their element-wise basis functions evaluated based on their parametric coordinates $\boldsymbol{\xi} = \begin{bmatrix} \xi_1\ \xi_2 \end{bmatrix}^\top$ on a reference element.[1] Taking the acoustic pressure p_a as an example, the discrete BE representation is given by

$$p_a\left(\mathbf{x}(\boldsymbol{\xi})\right) = \boldsymbol{\phi}(\boldsymbol{\xi})^\top \mathbf{p}_a^j \quad \text{with} \quad \mathbf{x} \in \Gamma_j\,, \tag{20}$$

where $\mathbf{p}_a^j$ represents the acoustic pressure in the physics nodes of element j and $\boldsymbol{\phi}$ denotes the physical interpolation polynomials. These discrete variables constitute column vectors with as many entries as physical nodes on the element. Since $\mathbf{p}_a^j$ has only constant values, the gradient of p_a with respect to $\boldsymbol{\xi}$ can be replaced by the equivalent gradient of its interpolation functions as

$$\nabla_{\boldsymbol{\xi}} p_a\left(\mathbf{x}(\boldsymbol{\xi})\right) = \left(\nabla_{\boldsymbol{\xi}} \boldsymbol{\phi}(\boldsymbol{\xi})^\top\right) \mathbf{p}_a^j \quad \text{with} \quad \mathbf{x} \in \Gamma_j\,. \tag{21}$$

In the case of collocation on continuous elements, the physical and geometrical nodes coincide. As nodes are shared between elements, the interpolation function derivatives are taken as the average of the contributions from all the elements connected to a node. Further explanations on the procedure are given in Appendix B.

Applying the multivariate chain rule $\nabla_{\boldsymbol{\xi}} p_a = (\nabla_{\boldsymbol{\xi}} \mathbf{x}^\top) \nabla_{\mathbf{x}} p_a$ followed by a reformulation to isolate $\nabla_{\mathbf{x}} p_a$ gives the gradient in a global coordinate basis. Since the computations are conducted on boundary elements, the $f : \mathbb{R}^3 \mapsto \mathbb{R}^2$ mapping via $\nabla_{\boldsymbol{\xi}} \mathbf{x}^\top$ manifests as a 2×3 Jacobian matrix which cannot be inverted. As a remedy, an artificial third coordinate direction $\frac{\partial \mathbf{x}}{\partial \xi_3} = \frac{\partial \mathbf{x}}{\partial \xi_1} \times \frac{\partial \mathbf{x}}{\partial \xi_2}$ is introduced such that $\frac{\partial \boldsymbol{\phi}}{\partial \xi_3} = \mathbf{0}$.[33] Inversion of the square Jacobian eventually delivers the pressure gradient in global space as

$$\nabla_{\mathbf{x}}^{\parallel} p_a\left(\mathbf{x}(\boldsymbol{\xi})\right) = \begin{bmatrix} \frac{\partial x}{\partial \xi_1} & \frac{\partial y}{\partial \xi_1} & \frac{\partial z}{\partial \xi_1} \\ \frac{\partial x}{\partial \xi_2} & \frac{\partial y}{\partial \xi_2} & \frac{\partial z}{\partial \xi_2} \\ \left(\frac{\partial \mathbf{x}}{\partial \xi_1} \times \frac{\partial \mathbf{x}}{\partial \xi_2}\right)^\top \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \boldsymbol{\phi}}{\partial \xi_1}^\top \\ \frac{\partial \boldsymbol{\phi}}{\partial \xi_2}^\top \\ \mathbf{0}^\top \end{bmatrix} \mathbf{p}_a^j \quad \text{with} \quad \mathbf{x} \in \Gamma_j\,, \tag{22}$$

where the superscript $\parallel$ highlights that this is only the in-plane component of the gradient. To compute the coefficients of the Jacobian matrix, we use the discrete representation of the position vector $\mathbf{x}$ as

$$\nabla_{\boldsymbol{\xi}} \mathbf{x}^\top = \left(\nabla_{\boldsymbol{\xi}} \mathbf{N}_{\mathbf{x}}(\boldsymbol{\xi})^\top\right) \mathbf{X}_j \quad \text{with} \quad \mathbf{x} \in \Gamma_j\,, \tag{23}$$

where the rows of $\mathbf{X}_j$ contain the global coordinates of the geometry nodes of element j and $\mathbf{N}_{\mathbf{x}}$ holds the geometrical basis functions. Storing the discrete nodal values requires three and one columns in $\mathbf{X}_j$ and $\mathbf{N}_{\mathbf{x}}$, respectively. The row count is dictated by the number of

geometrical nodes per element.

Splitting the tangential gradient in Eq. (22) into its directional components (x, y, z) and further applying a coordinate transformation allows to formulate the coupling conditions from Eqs. (16)-(19) in a discrete manner. The global derivative matrices $\mathbf{D}_\bullet$ are built such that multiplying one row i of the matrices with the array of nodal values equals the interpolation function derivative at node i (see also Appendix B). Written out this means that

$$\partial_x \mathbf{p}_a{}^{\parallel} = \mathbf{D}_x \mathbf{p}_a \,, \quad \partial_y \mathbf{p}_a{}^{\parallel} = \mathbf{D}_y \mathbf{p}_a \,, \quad \partial_z \mathbf{p}_a{}^{\parallel} = \mathbf{D}_z \mathbf{p}_a \,. \tag{24}$$

If the chosen discretization of the thermal and viscous mode are the same as for the acoustical mode, then the above interpolation function derivative matrices can be reused to compute $\nabla^{\parallel}_{\mathbf{x}} p_h$ and $\nabla^{\parallel}_{\mathbf{x}} \cdot \mathbf{v}_v$, respectively. We further introduce a transformation matrix that performs a change of basis from global (x, y, z) - coordinates (denoted by subscript g) to local node-based (t, s, n) - coordinates (denoted by subscript ℓ) as

$${}_\ell\mathbf{M}_g = \begin{bmatrix} \mathrm{diag}(\mathbf{t}_x) & \mathrm{diag}(\mathbf{t}_y) & \mathrm{diag}(\mathbf{t}_z) \\ \mathrm{diag}(\mathbf{s}_x) & \mathrm{diag}(\mathbf{s}_y) & \mathrm{diag}(\mathbf{s}_z) \\ \mathrm{diag}(\mathbf{n}_x) & \mathrm{diag}(\mathbf{n}_y) & \mathrm{diag}(\mathbf{n}_z) \end{bmatrix} , \tag{25}$$

where $(\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z)$ are respectively the directional components of the first and $(\mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z)$ of the second unit tangential vectors. Similarly, $(\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z)$ hold the nodal values of the unit normal vector in (x, y, z) - direction. The matrix in Eq. (25) is a permutation of a block-diagonal matrix whose blocks resemble the local orthogonal change of basis matrices. As permutations preserve orthogonality, it must be true that ${}_\ell\mathbf{M}_g$ is also orthogonal. Using the change-of-basis matrix, the derivatives of Eq. (24) can be cast to the local space as

$$\begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{D}_n \end{bmatrix} = {}_\ell\mathbf{M}_g \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_x \end{bmatrix} . \tag{26}$$

It is apparent that a matrix $\mathbf{D}_n$ representing the normal derivatives is included. However, due to the IFDs only containing the in-plane information, this part disappears up until machine precision, resulting in $\mathbf{D}_n \approx \mathbf{0}$. Instead, the out-of-plane information is added through a combination with the ∂_n - components of the discrete BIEs.

2.3.2. *Discrete Coupling Conditions and Final Formulation*

While the local gradients in the coupling conditions are now accessible via Eq. (26), the mismatch between the locally defined velocities in the no-slip and null-divergence equation (Eqs. (16)-(19)) and the global definition in the discrete BEM equations (Eq. (14)) remains. To resolve this issue, a change-of-basis is applied to $\mathbf{v}_v$ and its normal derivative using once again ${}_\ell\mathbf{M}_g$ as

$$\mathbf{v}_{v\ell} \quad = {}_\ell\mathbf{M}_g \mathbf{v}_v \,, \tag{27}$$

$$\partial_n \mathbf{v}_{v\ell} = {}_\ell\mathbf{M}_g \, \partial_n \mathbf{v}_v \,. \tag{28}$$

Finally, the no-slip and null-divergence conditions are computed numerically via

$$\phi_a \left(\begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{0} \end{bmatrix} \mathbf{p}_a + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} \partial_n \mathbf{p}_a \right) + \phi_h \left(\begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{0} \end{bmatrix} \mathbf{p}_h + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} \partial_n \mathbf{p}_h \right) + {}_\ell\mathbf{M}_g \mathbf{v}_v = {}_\ell\mathbf{M}_g \mathbf{v}_b \,, \tag{29}$$

$$\begin{bmatrix} \mathbf{D}_t \; \mathbf{D}_s \; \mathbf{0} \end{bmatrix} {}_\ell\mathbf{M}_g \mathbf{v}_v + \begin{bmatrix} \mathbf{0} \; \mathbf{0} \; \mathbf{I} \end{bmatrix} {}_\ell\mathbf{M}_g \, \partial_n \mathbf{v}_v = \mathbf{0} \,, \tag{30}$$

where $\mathbf{v}_b = \begin{bmatrix} \mathbf{v}_{b,x} \; \mathbf{v}_{b,y} \; \mathbf{v}_{b,z} \end{bmatrix}^\top$ is the boundary velocity in global coordinates. Putting together the discrete formulations of the BIEs (Eqs. (11)-(13)) and the coupling conditions (Eqs. (15),(29),(30)), the viscothermal BEM system reads

$$\begin{bmatrix} \mathbf{H}_a & -\mathbf{G}_a & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_h & -\mathbf{G}_h & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_v & -\mathbf{G}_v \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\mathbf{D}_{r\ell}\, {}_\ell\mathbf{M}_g) & \left(\mathbf{N}_\mathbf{I}^\top {}_\ell\mathbf{M}_g\right) \\ \tau_a \mathbf{I} & \mathbf{0} & \tau_h \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \phi_a \mathbf{D}_{c\ell} & \phi_a \mathbf{N}_\mathbf{I} & \phi_h \mathbf{D}_{c\ell} & \phi_h \mathbf{N}_\mathbf{I} & {}_\ell\mathbf{M}_g & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \partial_n \mathbf{p}_a \\ \mathbf{p}_h \\ \partial_n \mathbf{p}_h \\ \mathbf{v}_v \\ \partial_n \mathbf{v}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ {}_\ell\mathbf{M}_g \mathbf{v}_b \end{bmatrix} , \tag{31}$$

where the following row (subscript r) and column (subscript c) matrices where introduced to shorten the notation

$$\mathbf{N}_\mathbf{I} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix} , \quad \mathbf{D}_{c\ell} = \begin{bmatrix} \mathbf{D}_t \\ \mathbf{D}_s \\ \mathbf{0} \end{bmatrix} , \quad \mathbf{D}_{r\ell} = \begin{bmatrix} \mathbf{D}_t \; \mathbf{D}_s \; \mathbf{0} \end{bmatrix} . \tag{32}$$

While keeping the formulation in local space works in practice and has been the go-to strategy previously, it also introduces additional model complexity as knowledge of the nodal tangentials $\mathbf{t}$ and $\mathbf{s}$ is required to perform the coordinate transformation via ${}_\ell\mathbf{M}_g$, cf. Eq. (25). In the following, we avoid said complexity by updating the formulation to a global basis. In doing so, the gradient can be put together from a combination of the global in-plane IFDs and a normal projection.

3. Global Formulation of the Viscothermal BEM

Since the IFDs only replace the in-plane differential operator, we propose to decompose the gradients of the (acoustic and thermal) pressure $p_\bullet$ such that

$$\nabla p_\bullet = \nabla^\parallel p_\bullet + \nabla^\perp p_\bullet \,, \tag{33}$$

where $\nabla^\perp p_\bullet$ is the vector projection of the pressure gradient onto the out-of-plane normal direction described by

$$\nabla^\perp p_\bullet = \mathrm{proj}_\mathbf{n} \left(\nabla p_\bullet\right) = \mathbf{n} \left(\nabla p_\bullet \cdot \mathbf{n}\right) = \mathbf{n} \, \partial_n p_\bullet \,. \tag{34}$$

Here, $\partial_n p_\bullet$ is the result of the scalar projection of $\nabla p_\bullet$ onto the unit normal vector $\mathbf{n}$. Given that the IFDs contain the tangential contribution and that the normal derivative is part of the BIEs, the discrete form of Eq. (33) can be rewritten as

$$\nabla \mathbf{p}_\bullet = \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_z \end{bmatrix} \mathbf{p}_\bullet + \begin{bmatrix} \mathrm{diag}(\mathbf{n}_x) \\ \mathrm{diag}(\mathbf{n}_y) \\ \mathrm{diag}(\mathbf{n}_z) \end{bmatrix} \partial_n \mathbf{p}_\bullet \,. \tag{35}$$

Using this decomposed formulation of the gradients, we redefine the no-slip condition from Eq. (29) in global space as

$$\phi_a \left(\begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_z \end{bmatrix} \mathbf{p}_a + \begin{bmatrix} \mathrm{diag}(\mathbf{n}_x) \\ \mathrm{diag}(\mathbf{n}_y) \\ \mathrm{diag}(\mathbf{n}_z) \end{bmatrix} \partial_n \mathbf{p}_a \right) + \phi_h \left(\begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_z \end{bmatrix} \mathbf{p}_h + \begin{bmatrix} \mathrm{diag}(\mathbf{n}_x) \\ \mathrm{diag}(\mathbf{n}_y) \\ \mathrm{diag}(\mathbf{n}_z) \end{bmatrix} \partial_n \mathbf{p}_h \right) + \mathbf{v}_v = \mathbf{v}_b \,. \tag{36}$$

Similarly, the divergence splits into a tangential and normal component as

$$\nabla_{\mathbf{x}} \cdot \mathbf{v}_v = \nabla_{\mathbf{x}}^{\parallel} \cdot \mathbf{v}_v + \nabla_{\mathbf{x}}^{\perp} \cdot \mathbf{v}_v \,. \tag{37}$$

Using once more that the tangential information comes directly from the IFDs and that the normal information is included in the discrete BIEs, Eq. (30) converts to

$$\nabla \cdot \mathbf{v}_v = \begin{bmatrix} \mathbf{D}_x \, \mathbf{D}_y \, \mathbf{D}_z \end{bmatrix} \mathbf{v}_v + \begin{bmatrix} \mathrm{diag}(\mathbf{n}_x) \, \mathrm{diag}(\mathbf{n}_y) \, \mathrm{diag}(\mathbf{n}_z) \end{bmatrix} \partial_n \mathbf{v}_v \,. \tag{38}$$

Finally, by replacing the original no-slip and null-divergence conditions with their globally defined counterparts, the updated formulation of the viscothermal BEM reads

$$\begin{bmatrix} \mathbf{H}_a & -\mathbf{G}_a & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_h & -\mathbf{G}_h & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}_v & -\mathbf{G}_v \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_r & \mathbf{N}^\top \\ \tau_a \mathbf{I} & \mathbf{0} & \tau_h \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \phi_a \mathbf{D}_c & \phi_a \mathbf{N} & \phi_h \mathbf{D}_c & \phi_h \mathbf{N} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \partial_n \mathbf{p}_a \\ \mathbf{p}_h \\ \partial_n \mathbf{p}_h \\ \mathbf{v}_v \\ \partial_n \mathbf{v}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_b \end{bmatrix} , \tag{39}$$

where the following shorthand notation has been used

$$\mathbf{N} = \begin{bmatrix} \mathrm{diag}(\mathbf{n}_x) \\ \mathrm{diag}(\mathbf{n}_y) \\ \mathrm{diag}(\mathbf{n}_z) \end{bmatrix} , \quad \mathbf{D}_c = \begin{bmatrix} \mathbf{D}_x \\ \mathbf{D}_y \\ \mathbf{D}_z \end{bmatrix} , \quad \mathbf{D}_r = \begin{bmatrix} \mathbf{D}_x \, \mathbf{D}_y \, \mathbf{D}_z \end{bmatrix} . \tag{40}$$

As this viscothermal system comes with 10 degrees of freedom for each of the n collocation points, solving the linear system of equations directly on a desktop machine proves impractical even for moderately large problems. Fortunately, the system matrix shows a distinct block structure that can be exploited to reduce the problem size. In particular, all matrices except for $\mathbf{G}_a$ and $\mathbf{H}_a$ are sparse. The following section explains how to take advantage of the sparsity pattern via a Schur complement reduction. Additionally, applying the Schur complement can be a measure to counteract the ill-conditioning that occurs in coupled systems with quantities of different orders of magnitude. Similar concepts are used in elastoacoustics to improve the convergence of iterative solvers.[34]

4. Static Condensation via Schur Complement

Applying the Schur complement resembles a Gaussian elimination at matrix block level.[35] By eliminating internal unknowns, we sequentially reduce the system's size until only n DOFs remain. As the pressure field is of primary interest for engineering applications, we keep the nodal acoustic pressure as the solution vector. The simplicity of the isothermal boundary condition allows for an initial removal of the thermal mode. By solving Eqs. (7) and (12) with respect to $\mathbf{p}_a$ and $\partial_n \mathbf{p}_h$ as

$$\mathbf{p}_h = -\frac{\tau_a}{\tau_h}\mathbf{p}_a\,, \quad \partial_n \mathbf{p}_h = \mathbf{G}_h^{-1}\mathbf{H}_h\mathbf{p}_h = -\frac{\tau_a}{\tau_h}\mathbf{G}_h^{-1}\mathbf{H}_h\mathbf{p}_a\,, \tag{41}$$

the matrix equation (39) is reduced to

$$\begin{bmatrix} \mathbf{H}_a & -\mathbf{G}_a & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_v & -\mathbf{G}_v \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_r & \mathbf{N}^\top \\ \mu_a\mathbf{D}_c - \mu_h\mathbf{N}\mathbf{G}_h^{-1}\mathbf{H}_h & \phi_a\mathbf{N} & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \partial_n\mathbf{p}_a \\ \mathbf{v}_v \\ \partial_n\mathbf{v}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_b \end{bmatrix}, \tag{42}$$

where the following constants are introduced to ease the notation

$$\mu_a = \phi_a - \frac{\tau_a\phi_h}{\tau_h}\,, \quad \mu_h = \frac{\tau_a\phi_h}{\tau_h}\,. \tag{43}$$

Further, the normal derivatives of $\mathbf{p}_a$ and $\mathbf{v}_v$ are eliminated by utilizing reformulations of the discretized integral equations (11) and (13) as

$$\partial_n\mathbf{p}_a = \mathbf{G}_a^{-1}\mathbf{H}_a\mathbf{p}_a\,, \tag{44}$$

$$\partial_n\mathbf{v}_v = \mathbf{G}_v^{-1}\mathbf{H}_v\mathbf{v}_v\,. \tag{45}$$

Plugging the above into Eq. (42) condenses the system to

$$\begin{bmatrix} \mathbf{0} & \mathbf{D}_r + \mathbf{N}^\top\mathbf{G}_v^{-1}\mathbf{H}_v \\ \mu_a\mathbf{D}_c + \mathbf{N}\left(\phi_a\mathbf{G}_a^{-1}\mathbf{H}_a - \mu_h\mathbf{G}_h^{-1}\mathbf{H}_h\right) & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \mathbf{v}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_b \end{bmatrix}. \tag{46}$$

Finally, the viscous mode is taken out using that

$$\mathbf{v}_v = \mathbf{v}_b - \left(\mu_a\mathbf{D}_c + \phi_a\mathbf{N}\mathbf{G}_a^{-1}\mathbf{H}_a - \mu_h\mathbf{N}\mathbf{G}_h^{-1}\mathbf{H}_h\right)\mathbf{p}_a\,. \tag{47}$$

To simplify the formulation, the following shorthand notation is applied

$$\mathbf{R} = \mathbf{D}_r + \mathbf{N}^\top\mathbf{G}_v^{-1}\mathbf{H}_v\,. \tag{48}$$

As such Eq. (46) simplifies to

$$\mathbf{R}\left(\mu_a\mathbf{D}_c + \mathbf{N}\left(\phi_a\mathbf{G}_a^{-1}\mathbf{H}_a - \mu_h\mathbf{G}_h^{-1}\mathbf{H}_h\right)\right)\mathbf{p}_a = \mathbf{R}\mathbf{v}_b\,. \tag{49}$$

From a numerical point of view, the above formulation is challenging as it includes the inverse of the dense matrix $\mathbf{G}_a$. To remove the necessity for a dense inversion, Eq. (49) is multiplied from the left with $\mathbf{G}_a\left(\mathbf{R}\mathbf{N}\right)^{-1}$ resulting in

$$\left[\mathbf{G}_a\left(\mu_a\left(\mathbf{R}\mathbf{N}\right)^{-1}\mathbf{R}\mathbf{D}_c - \mu_h\mathbf{G}_h^{-1}\mathbf{H}_h\right) + \phi_a\mathbf{H}_a\right]\mathbf{p}_a = \mathbf{G}_a\left(\mathbf{R}\mathbf{N}\right)^{-1}\mathbf{R}\mathbf{v}_b\,. \tag{50}$$

Since the reduced system matrix is of the same size as in the conventional lossless formulation, storing the matrix requires much less memory than experienced for previous versions of the lossy BEM. Furthermore, solving the linear system of equations takes significantly fewer floating point operations since only $1/10^{\text{th}}$ of the former viscothermal unknowns and thus $1/100^{\text{th}}$ of the number of matrix coefficients remain. As the boundary layer contributions are relatively small, the condensed coupled system is expected to be mainly dominated by the dense acoustic matrices. As such, their well-conditioning is inherited, which clears the path for usage of iterative solvers.

5. Nested Iterative Solution Scheme

While the Schur complement formulation reduces the system's overall size, its setup goes along with multiple additional matrix-matrix products, some of which require a matrix inversion. As such, the computational complexity of the condensation scales cubicly. In order to mitigate this performance penalty, we propose combining the Schur complement lossy BEM with a nested solution scheme, which employs iterative solvers in three stages. By performing the math operations in an iterative manner, the sequence of matrix-matrix products in Eq. (50) is replaced by matrix-vector products. Thus, neither any intermediate nor the final system matrix must be stored directly. Instead, the solution for the acoustic pressure is computed using linear operators that access the matrix-building blocks. In doing so, the time and space complexity of the system solve are improved by an order.

Establishing such a nested solution scheme requires propagating the matrix-vector multiplications in Eq. (50) from right to left. Parenthesizing the four components that form the left- and right-hand side of Eq. (50) as

$$\begin{aligned}\mu_a \mathbf{G}_a \left((\mathbf{RN})^{-1} \left(\mathbf{R} (\mathbf{D}_c \mathbf{p}_a) \right) \right) - \mu_h \mathbf{G}_a \left(\mathbf{G}_h^{-1} (\mathbf{H}_h \mathbf{p}_a) \right) \\ + \phi_a \mathbf{H}_a \mathbf{p}_a = \mathbf{G}_a \left((\mathbf{RN})^{-1} (\mathbf{R} \mathbf{v}_b) \right) ,\end{aligned} \tag{51}$$

yields the ordering of the math operations. The products with $\mathbf{D}_c$ and $\mathbf{H}_h$ are evaluated using sparse linear algebra and multiplication with $\mathbf{G}_h^{-1}$ can be done by means of either a sparse direct or an iterative solver, both of which scale linearly. Computing the product with $\mathbf{R}$ and $(\mathbf{RN})^{-1}$, on the other hand, requires some preliminary thoughts. We start by looking at the multiplication of $\mathbf{R}$ with any vector $\bar{\mathbf{x}}$. The matrix-vector product is computed following

$$\mathbf{R} \bar{\mathbf{x}} = \mathbf{D}_r \bar{\mathbf{x}} + \mathbf{N}^\top \left(\mathbf{G}_v^{-1} \left(\mathbf{H}_v \bar{\mathbf{x}} \right) \right) , \tag{52}$$

where $\mathbf{G}_v^{-1}$ is, similarly to $\mathbf{G}_h^{-1}$, handled efficiently using a sparse direct or an iterative solver. For the test case considered in this paper, the viscous and thermal matrices are very sparse and well-conditioned. The resulting quick convergence of the iterative solvers favors the iterative approach over the sparse direct solver. For multiplication with $(\mathbf{RN})^{-1}$, we first evaluate Eq. (52) replacing the generic vector $\bar{\mathbf{x}}$ with the matrix-vector product $\mathbf{N}\bar{\mathbf{x}}$ and

then employ an iterative solver to obtain the result of the multiplication with the inverse. A detailed description of the concept for solving the condensed viscothermal BEM efficiently is given in Algorithm 1. As the matrices are generally non-Hermitian, we pick a robust iterative solver considered appropriate for these circumstances: GMRes.[1,27] For now, we refrain from applying preconditioners.

Algorithm 1 Nested solution scheme employing restarted GMRes solvers on three levels.

Input: $\mathbf{G}_a, \mathbf{H}_a, \mathbf{G}_h, \mathbf{H}_h, \mathbf{G}_v, \mathbf{H}_v, \mathbf{D}_r, \mathbf{D}_c, \mathbf{N}, \mu_a, \mu_h, \phi_a$ ▷ Matrix-building blocks
Output: $\mathbf{p}_a$ ▷ Nodal acoustic pressure
  function ConstructOperator_$\mathbf{R}(\bar{\mathbf{x}})$ ▷ Level 1: Linear operator
    return $\mathbf{D}_r\bar{\mathbf{x}} + \mathbf{N}^\top * \mathrm{gmres}(\mathbf{G}_v, \mathbf{H}_v\bar{\mathbf{x}})$
  end function
  function ConstructOperator_$\mathbf{RN}(\bar{\mathbf{x}})$ ▷ Auxiliary operator
    return $\mathbf{R}(\mathbf{N}\bar{\mathbf{x}})$
  end function
  function ConstructOperator_$\mathbf{A}(\bar{\mathbf{x}})$ ▷ Level 2: Linear operator
    return $\mathbf{G}_a\left(\mu_a * \mathrm{gmres}(\mathbf{RN}, \mathbf{R}(\mathbf{D}_c\bar{\mathbf{x}})) - \mu_h * \mathrm{gmres}(\mathbf{G}_h, \mathbf{H}_h\bar{\mathbf{x}})\right) + \phi_a\mathbf{H}_a\bar{\mathbf{x}}$
  end function
  $\mathbf{b} \leftarrow \mathbf{G}_a * \mathrm{gmres}(\mathbf{RN}, \mathbf{R}(\mathbf{v}_b))$ ▷ Compute right-hand side
  $\mathbf{p}_a \leftarrow \mathrm{gmres}(\mathbf{A}, \mathbf{b})$ ▷ Level 3: Solve linear system

6. Results

6.1. *Geometry and Discretization*

The performance of the global IFD-BEM is studied based on an academic test case. We investigate the sound radiation of a rigid sphere of radius 1 m that is surrounded by air at reference ambient conditions. The corresponding properties of the thermoviscous fluid are provided in Appendix A. The sphere oscillates along its polar axis in z - direction. As boundary condition, a uniform particle velocity of $v_{b,z} = 10^{-2}\,\mathrm{m/s}$ is applied to all surface nodes. This example presents an exterior problem where the tangential movement of the surface induces significant viscous dissipation in the boundary layers. Although analysis of any field point in the exterior domain is possible, only the boundary solution is of interest since the viscothermal effects become increasingly small with a growing distance from the surface. The same application has been previously used to test the axisymmetric,[14] 2D,[15] and 3D formulation[11] of the viscothermal BEM. The test case offers an analytical solution[36] and can therefore be used to validate the numerical results of the revised lossy BEM. Note that the analytical reference only includes the dominant viscous effects but omits the less significant thermal losses.[36] A detailed description of the analytical model can be found in Appendix C. To avoid the non-uniqueness problem of the BEM,[1] the studies are conducted at frequencies sufficiently distant from the internal eigenfrequencies of the sphere. Benchmarking and testing is done at 100 Hz and 1000 Hz. The boundary meshes are

generated using COMSOL Multiphysics[37] and are centered at the origin of the coordinate axes. As shown in Fig. 1, seven different meshes built from quadratic 6-node triangular elements are studied. We employ isoparametric elements using the same set of basis functions for both geometry and physics interpolation.

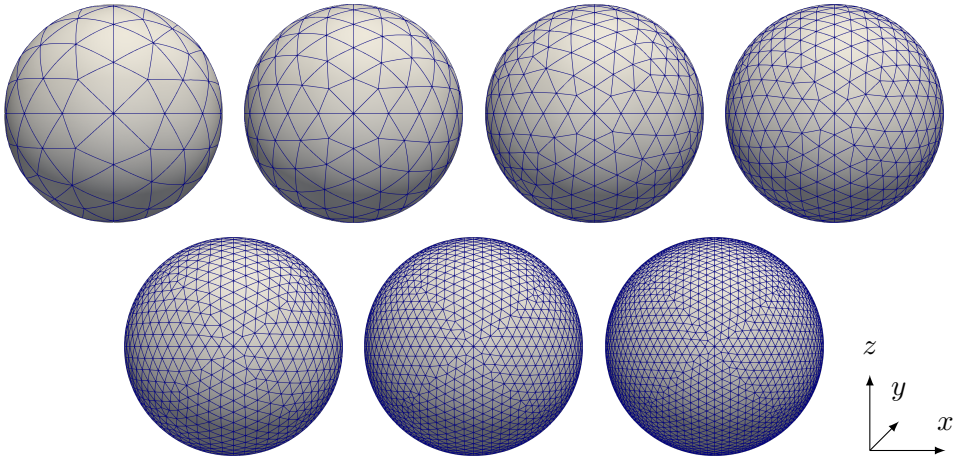


Fig. 1: Boundary element meshes employed for testing the lossy BEM. All meshes are built from quadratic triangular elements. From top left to bottom right, incremental refinement results in meshes composed of 246, 464, 840, 1320, 2794, 4506, and 6530 elements, respectively.

In order to decrease the computational costs for assembling and storing the viscothermal matrices, we predetermine the sparsity pattern of the matrices. Since the boundary losses have limited spatial significance,[8] calculating the boundary layer thicknesses relative to the element size reveals the nonzero nodal interactions, which will contribute to the matrix setup. The boundary layer thicknesses in air can be approximated by[10]

$$\delta_v \approx \frac{2.1}{\sqrt{f/\mathrm{Hz}}}\,\mathrm{mm}\,, \quad \delta_h \approx \frac{2.5}{\sqrt{f/\mathrm{Hz}}}\,\mathrm{mm}\,. \tag{53}$$

Thus, in the frequency range of interest from 100 Hz to 1000 Hz, the maximum space the boundary layers will occupy equals $\delta_{\max} \approx 2.5/\sqrt{f_{\min}/\mathrm{Hz}} = 2.5 \cdot 10^{-4}\,\mathrm{m}$. By ensuring that all elements of the meshes are larger than a predefined multiple of said maximum thickness, we can limit the boundary integration to the element the current collocation point is part of and its direct neighbors. Tab. 1 gives the minimum edge length of each of the boundary meshes and the number of maximum-sized boundary layers that would fit into said element. We rule out any number being smaller than 20 since truncating the viscothermal kernel functions at a distance of 20 boundary layers is regarded as sufficiently accurate.[8]

Table 1: Minimum and maximum element edge lengths ($h_{\min}$ and $h_{\max}$) in meters evaluated according to Ref. 38 for all mesh configurations. Estimate of worst-case scenarios for the number of (abbreviated #) elements per wavelength and the number of vertically stacked boundary layers fitting into an element.

# elements n_e	246	464	840	1320	2794	4506	6530
# nodes n	494	930	1682	2642	5590	9014	13062
$h_{\min}$	0.1981	0.1352	0.0925	0.0718	0.0462	0.0373	0.0243
$h_{\max}$	0.2554	0.1823	0.1521	0.1139	0.0790	0.0618	0.0508
Min. # elements per wavelength	1.34	1.88	2.26	3.02	4.35	5.56	6.77
Min. # boundary layers per element	792.4	540.8	370.0	287.2	184.7	149.2	97.1

Furthermore, the maximum element edge length and the minimum wavelength of $\lambda_{\min} = c/f_{\max} = 0.344\,\mathrm{m}$ are put in relation to give a worst-case estimate for the number of elements per wavelength for all test cases. As demonstrated in Tab. 1, the coarser boundary meshes do not comply with common rules for discretization,[39] which might lead to inaccurate results for the 1000 Hz studies.

6.2. *Validation*

For validating the revised BEM with losses, we reproduce the oscillating sphere test case from Ref. 11. The sphere mesh with $n_e = 1320$ elements ($n = 2642$ nodes) serves as geometry approximation for a study at 1000 Hz. Although the new Schur complement formulation exclusively solves for the acoustic pressure, we retrieve the condensed velocity vector using Eq. (47). As such, a comparison to the full analytical solution is possible. The code base used for all simulations in this work has been developed by the Acoustics Technology Group at DTU jointly with the Chair of Vibroacoustics of Vehicles and Machines at TUM. The software is written in the Julia Programming Language,[40] which constitutes a good compromise between speed and ease of implementation and usage. All results are generated running Julia version 1.9.1.

Fig. 2 shows the analytical and numerical results for the complex-valued acoustic pressure and viscous velocity on the boundary. Depicted are the discrete nodal values at the continuous surface nodes. Aiming for validation of the new lossy BEM, we consult an appropriate analytical model.[36] Since the reference model, as discussed in Appendix C, returns the viscous velocity in radial direction (denoted by superscript r) and along the polar angle (denoted by superscript θ), which align with the normal and tangential directions, we convert

the solution vector to the same spherical coordinate basis through

$$\mathbf{v}_v^r = -\mathbf{v}_{v,n} = -\mathbf{N}^\top \mathbf{v}_v \,, \tag{54}$$

$$\mathbf{v}_v^{\parallel} = \mathbf{v}_v - \mathbf{v}_v^{\perp} = \mathbf{v}_v - \mathbf{N}\mathbf{v}_{v,n} \,, \tag{55}$$

$$\mathbf{v}_v^{\theta} = \operatorname{diag}(\cos\boldsymbol{\theta}) \operatorname{diag}(\cos\boldsymbol{\varphi})\, \mathbf{v}_{v,x}^{\parallel} + \operatorname{diag}(\cos\boldsymbol{\theta}) \operatorname{diag}(\sin\boldsymbol{\varphi})\, \mathbf{v}_{v,y}^{\parallel} - \operatorname{diag}(\sin\boldsymbol{\theta})\, \mathbf{v}_{v,z}^{\parallel} \,. \tag{56}$$

Eq. (55) makes use of the assumption that each quantity can be split into its tangential in-plane component and a normal out-of-plane part, cf. Sec. 3. Note that this transformation does not require explicit knowledge of the nodal tangentials. The polar angle in Eq. (56) marks the inclination from the positive z - axis as specified in ISO standard 80000-2:2019. Following this convention, the polar and azimuth angles θ and φ for collocation point i depend on its respective Cartesian coordinates as

$$\theta_i = \arccos \frac{z_i}{\sqrt{x_i^2 + y_i^2 + z_i^2}} \,, \quad \varphi_i = \operatorname{sgn}(y_i) \arccos \frac{x_i}{\sqrt{x_i^2 + y_i^2}} \,. \tag{57}$$

Since we deal with discrete results, all mathematical operations are performed entry-wise, meaning separately for each entry of the discrete arrays. Consequently, we evaluate the sine and cosine in Eq. (56) for each of the nodal values stored in $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$.

As depicted in Fig. 2, the transverse movement of the sphere generates a viscous mode prevalent at its equator. While the magnitude of the viscous velocity peaks at around $\theta = 90°$ in the tangent space, the normal component has its maxima at the poles where the boundary excitation aligns with the direction of the normal vectors. The curves for the acoustic pressure follow a similar pattern, with the sound pressure magnitude being largest at $\theta = 0°$ and $180°$ and zero at the equator. Due to the test case resembling a dipole source, sound is primarily radiated in positive and negative z - direction. In general, the numerical and analytical results are in good agreement. But similar to previous versions of the lossy BEM,[11] the newly developed formulation fails to meet the reference curves when the values become very small. Quantities of small order of magnitude, like the normal viscous velocity, seem more likely to be affected by discretization inaccuracies. Even when using quadratic elements, the nodal positions may deviate from the theoretical radius of the sphere. A small study focusing on the impact of geometrical inaccuracies introduced during the meshing process has revealed that relocation of the nodes to match the ideal radius does not change the results significantly. Hence, we conclude that the geometry error only slightly contributes to the deviation from the analytical solution. The error associated with the physics discretization, especially when considering discontinuous instead of continuous elements, remains unclear and will be part of future contributions by the authors.

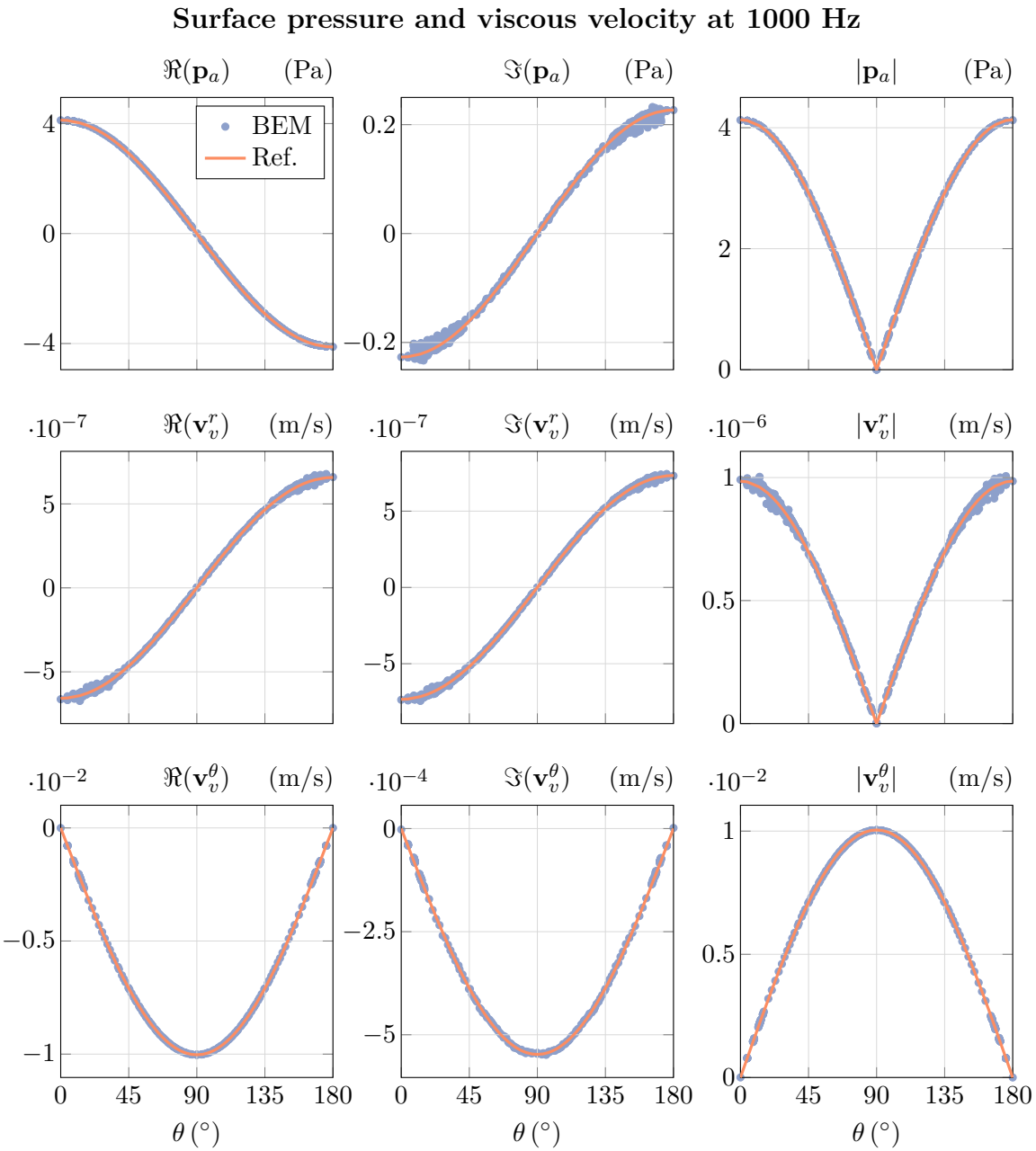


Fig. 2: Nodal acoustic pressure (top), viscous velocity in radial direction (middle), and viscous velocity along the polar angle (bottom) on the discrete surface of a sphere with $n_e = 1320$ elements, oscillating at 1000 Hz. Shown are the real (left), imaginary (middle), and absolute values (right) of the analytical reference (solid line) and the numerical solution calculated using the lossy iterative BEM (dots). The curves follow an arc from one pole of the sphere to the other ($z = 1\,\mathrm{m}$ to $-1\,\mathrm{m}$). Note the different orders of magnitude for the velocity plots.

For further clarification of the model accuracy, we evaluate the relative surface error as[1]

$$e_2^\Gamma = \frac{\|e^\Gamma\|_2}{\|p_{a,\mathrm{ref}}^\Gamma\|_2} = \sqrt{\frac{\sum_{i=1}^{n} |p_a(\mathbf{x}_i) - p_{a,\mathrm{ref}}(\mathbf{x}_i)|^2}{\sum_{i=1}^{n} |p_{a,\mathrm{ref}}(\mathbf{x}_i)|^2}}, \tag{58}$$

for all mesh configurations and both frequencies. Here, we focus on the acoustic pressure p_a at the nodal positions $\mathbf{x}_i$ as this is the solution quantity of the global Schur complement BEM. We analyze the convergence with respect to h-refinement, expecting a steady decrease of the error. Similarly, considering a larger frequency ideally leads to larger approximation errors when the mesh is kept constant. Both effects can be observed in Fig. 3. For the 1000 Hz study, a bend in the convergence curve occurs when approaching the coarser discretization levels. As the error surpasses 1 %, we conclude that the number of elements per wavelength in these meshes does not suffice to accurately model the physics.

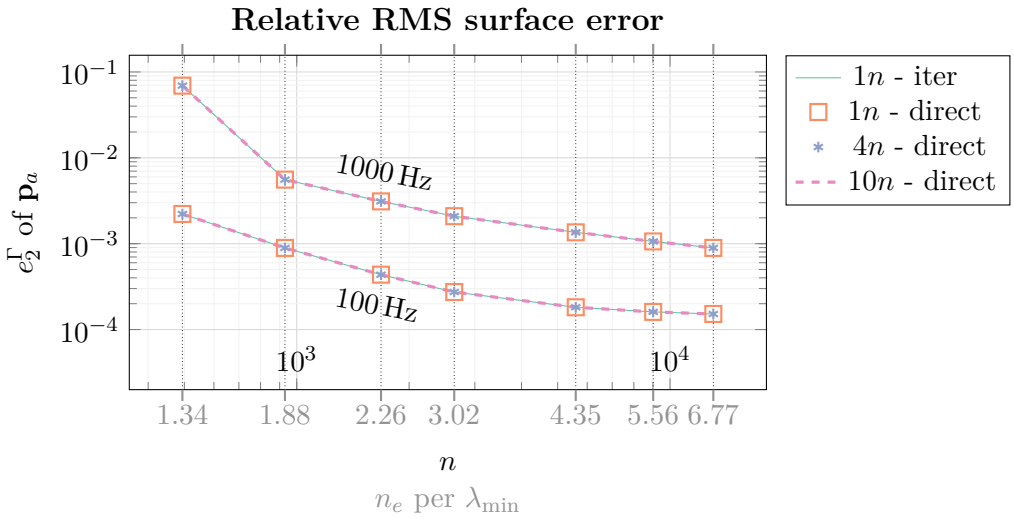


Fig. 3: Euclidean error norm[1] in terms of the node count n and the corresponding minimum number of elements per wavelength (n_e per $\lambda_{\min}$) evaluated on the discrete sphere surfaces. The relative error comparing the numerical and analytical solution for the acoustic pressure at 100 Hz and 1000 Hz is shown. The numerical results are computed using the global lossy BEM at different condensation levels: "10n" denotes the uncondensed, "4n" the partially condensed, and "1n" the fully condensed formulation. While the first two only allow a direct solution ("direct"), the last one can be solved directly or iteratively ("iter"). As all variants of the lossy BEM lead - except for the error introduced by GMRes - to nearly the same solution, the curves fall on top of each other.

Comparison of the error at different condensation levels further reveals the same model accuracy for the full model as defined in Eq. (39), the reduced model lacking the normal derivatives as described by Eq. (46), and the final model as stated in Eq. (50), which keeps only $\mathbf{p}_a$ as unknown. All variants of the system formulation have been solved directly using an LU - factorization. Furthermore, the iterative approach has been tested, cf. Algorithm 1. As shown in Fig. 3, the nested iterative solution scheme does not introduce significant additional errors.

6.3. *Computational Scaling*

After confirming the validity of the new formulation, we evaluate the performance gain achieved through the Schur complement reduction and iterative solution. First, we assess the method in terms of its theoretical algorithmic complexities. Then, we measure the actual performance with respect to the execution time and memory usage. To maintain consistency, all simulations for benchmarking the model are carried out on the Linux Cluster of the Leibniz Supercomputing Centre, specifically on the single-node Teramem partition, which is tailored to applications with extreme memory requirements. The linear-algebra operations are performed on 32 cores using the Julia link to the Linear Algebra PACKage (LAPACK)[41] and the multi-threaded BLAS library.[42]

All four versions of the lossy BEM rely on a precomputation of the building blocks for setting up the global system matrix. Both the assembly time (denoted by T) and the memory requirements (denoted by M) for this precomputation scale as $\mathcal{O}(2n^2 + sn)$ with n^2 emerging from the two dense acoustic matrices and n from the sparse matrices with matrix count s. After the computation of the matrix-building blocks, the asymptotic space complexity for setting up the dense linear systems of equations generally unfolds as $\mathcal{O}(N^2)$ with N being either $10n$, $4n$, or $1n$ depending on the system formulation. The time complexity heavily depends on the intermediate computations which are required to perform the condensation. Whereas the $10n$ - system from Eq. (39) does not introduce any extra steps, the $4n$ - formulation from Eq. (46) relies on the matrix products $\mathbf{G}_a^{-1}\mathbf{H}_a$, $\mathbf{G}_h^{-1}\mathbf{H}_h$, and $\mathbf{G}_v^{-1}\mathbf{H}_v$ with the dense multiplication setting the upper bound of the complexity as $\mathcal{O}(n^3)$. The same holds for the assembly of the $1n$ - system from Eq. (50) as it involves multiplying $\mathbf{G}_a$ with $(\mathbf{RN})^{-1}\mathbf{RD}_c$. When solving the system of equations directly via LU - factorization, the algorithmic time complexity peaks at $\mathcal{O}(N^3)$. Although of the same order, we expect a shift in performance for the three condensation levels due to the different system sizes and amount of additional matrices that need to be computed and kept in storage. When considering the cubic scaling for the direct solution, the necessary floating point operations multiply by a factor of 1000, 64, or 1. A similar but less pronounced shift is to be expected in the space complexity as the quadratic scaling for storing the matrix arises from 100, 16, or $1n^2$ memory allocations. Unlike the direct approach, the nested iterative solution scheme does not incorporate any operations of cubic order but is dominated by the dense matrix-vectors products scaling with $\mathcal{O}(n^2)$. By employing a restarted GMRes that utilizes the matrix-building blocks without explicitly storing the full matrix, the assembly and solution

time is bound by $\mathcal{O}(pn^2)$ and the memory by $\mathcal{O}(p_r n)$.[27] Here, $p = ij(3k)$ is the product of the total number of iterations on the outer (i), mid (j), and inner level (k) of the iterative scheme, and $p_r = i_r j_r(3k_r)$ is the corresponding iteration count before restart. As such, the computational scaling of the iterative approach strongly depends on the system conditioning. An in-depth analysis of the conditioning follows in Sec. 6.4. Tab. 2 summarizes the theoretical estimates for the asymptotic space and time complexities. In order to distinguish the various cases, we include the multiplication factor of the highest-order term.

Table 2: Theoretical estimate for the asymptotic time and space complexity comprising the system assembly with precomputed matrix building blocks and the direct or iterative solution with respect to the acoustic pressure.

System	$10n$ - direct	$4n$ - direct	$1n$ - direct	$1n$ - iter	Common offset
Space	$\mathcal{O}(100n^2)$	$\mathcal{O}(25n^2)$	$\mathcal{O}(8n^2)$	$\mathcal{O}(p_r n)$	$+M$
Time	$\mathcal{O}(1000n^3)$	$\mathcal{O}(65n^3)$	$\mathcal{O}(3n^3)$	$\mathcal{O}(pn^2)$	$+T$

For benchmarking the different versions of the lossy BEM, we track the elapsed wall clock time and the memory allocations from 20 simulation runs. The measurement results with respect to the system assembly and solution for $\mathbf{p}_a$ are visualized in Fig. 4. The computational effort for setting up the matrix-building blocks is not included, as it remains constant throughout the variants. The memory usage is evaluated as the maximum allocations that occur either during the matrix setup or during the solution while keeping the system matrix in storage. The assembly makes up the primary costs for the $4n$ - and $1n$ - direct variants, whereas the solution dominates for the others. Overall, the condensation and, if applicable, the iterative solution significantly improve both the time and space complexity. Taking the finest mesh as an example, the iterative approach proves more than 10^4 times faster at 100 Hz and $2.7 \cdot 10^3$ times faster at 1000 Hz than the conventional uncondensed version of the viscothermal BEM. Memory-wise, the performance gain is significant but less distinctive. At 100 Hz, 1169 times less memory is allocated and at 1000 Hz, 328 times less.

The measured memory consumption matches the predicted spatial complexities of linear and quadratic order. The runtime analysis, however, reveals fewer computational costs than estimated. As the values in Tab. 2 represent upper bounds, we expect the algorithms to somewhat outperform these theoretical complexities. The scaling behavior observed in Fig. 4 exceeds said expectations. Possible explanations might lead back to the utilized software and hardware architecture. The Julia packages are mainly written for scientific computing purposes and therefore performance-optimized with respect to math operations.[40] Some measures implemented are, for example, partial pivoting in the LU - decomposition[43] or restarting the GMRes algorithm.[27] Furthermore, in case of high constant factors, the lower-order terms can dominate the scaling until the matrices become very large. As such, the number of DOFs considered here might not suffice since some curves still experience a change

in slope, cf. the $4n$ - direct timings. Lastly, as more computations fit into the processor cash, which has a higher memory bandwidth than RAM, the reduction in memory allocations might result in speedups up to super-linear order. In general, all scaling measurements disclose a decrease in order for the new iterative Schur complement BEM, making it better suited for acoustic applications of larger size.

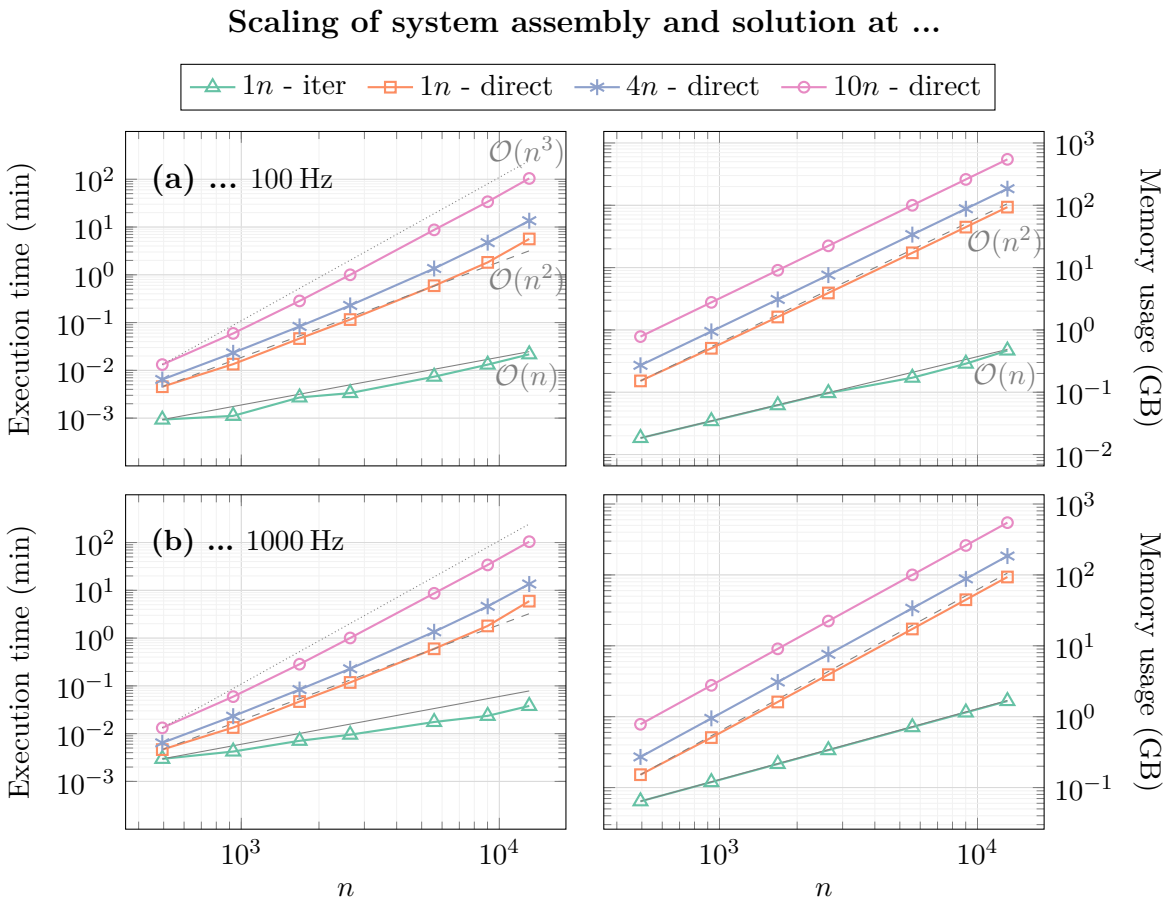


Fig. 4: Maximum memory consumption (right) and lower bound of wall clock time (left) measured from 20 evaluations of each of the lossy BEM algorithms. Depicted is the computational scaling for 100 Hz (top) and 1000 Hz (bottom) comprising the system assembly with precomputed matrix building blocks and the direct or iterative solution with respect to the acoustic pressure. Any steps to compute the building blocks or to reconstruct the condensed unknowns are excluded. The marks show the measurement results from the benchmarking runs, and the gray curves indicate matching theoretical complexities of linear (solid), quadratic (dashed), or cubic order (dotted).

When comparing the top and bottom plots of Fig. 4, the offset between the curves representing the iterative approach and the direct methods appears smaller in the 1000 Hz study. This dependence on the frequency can be attributed to a different convergence rate of the iterative solvers. Similar to conventional acoustics problems, we expect GMRes to take more iterations to converge when the frequency increases.[28] The following Tab. 3 lists the number of matrix-vector products performed when solving the viscothermal problem with respect to the acoustic pressure. While the solution at 100 Hz takes 6 to 7 matrix-vector products, at 1000 Hz 24 to 96 products are required. This only holds for the outmost GMRes loop. At the other levels of the nested solution scheme, 3 to 4 iterations suffice no matter the test case. The occurring decline in matrix-vector products with growing mesh size impacts the execution time at 1000 Hz. Although the scaling order remains the same, the shrinking multiplication factor flattens the time curve.

Table 3: The top rows list the number of matrix-vector products that were performed in the outer (level 3) GMRes-loop to solve the $1n$ - system iteratively. Only the convergence rate of the outmost GMRes varies with frequency and system size. Within every outer iteration step, 4 to 5 matrix-vector multiplications are performed in each of the mid (level 2) and inner loops (level 1). The bottom rows contain the number of matrix-vector products needed in the extension step for recovering $\mathbf{v}_v$ with GMRes. The relative error tolerance is set to 1e-7 for all steps.

n	494	930	1682	2642	5590	9014	13062
100 Hz	7	7	7	7	6	6	6
1000 Hz	96	73	50	42	35	27	24
100 Hz	18	17	19	19	18	18	17
1000 Hz	520	224	297	322	310	339	380

As the last part of the runtime analysis, we look into the computational costs for recovering the unknowns that are eliminated during the system reduction via Schur complement. Since the viscothermal BEM is mainly utilized in engineering acoustics, this step is generally not needed. The focus is on the pressure field as the solution quantity of the numerical method. Nevertheless, some applications might require additional knowledge of the velocity field. Extending the condensed system to retrieve the nodal viscous velocities follows Eq. (47), where the multiplication with an inversion of the dense acoustic matrices stands out as bottleneck. To relieve the computational burden, we employ an iterative solver, specifically restarted GMRes without preconditioning, for calculating $\mathbf{G}_a^{-1}(\mathbf{H}_a\mathbf{p}_a)$. This computation resembles the classical system setup we encounter in isentropic acoustics. Fig. 5 gives an outlook on how much effort the extension step takes when compared to the system assembly and solution, as well as the construction of the right-hand side. A tendency is evident in the $1n$ - iter case, where recovering the velocities becomes more costly with increasing problem

size. For 13062 nodes, most of the total execution time is spent on the extension. This is a consequence of the convergence rate for the GMRes solver, which proves worse than the ones observed in the nested scheme for calculating $\mathbf{p}_a$. The corresponding iteration count for the extension step is given in Tab. 3. As the increasing number of iterations does not change the order of the overall algorithmic complexity but only impacts the multiplication factor, we do not expect significant drawbacks for larger problems.

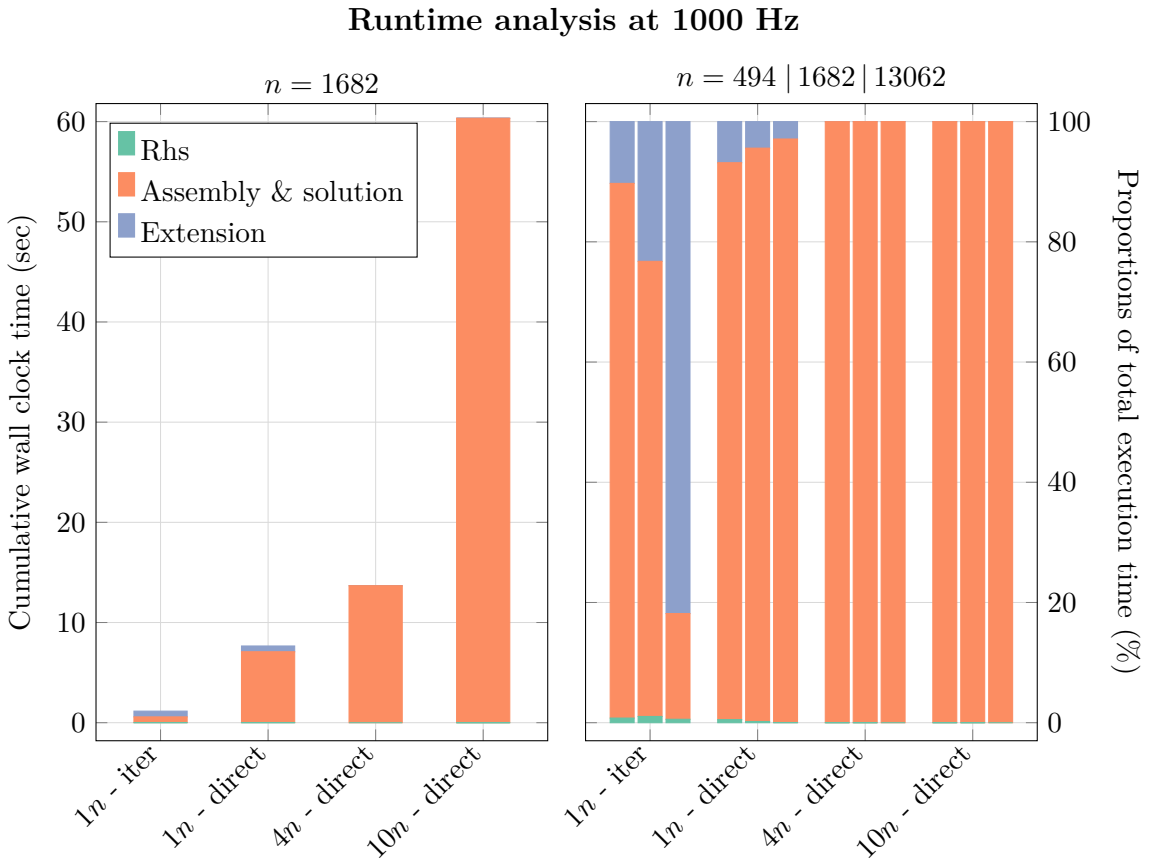
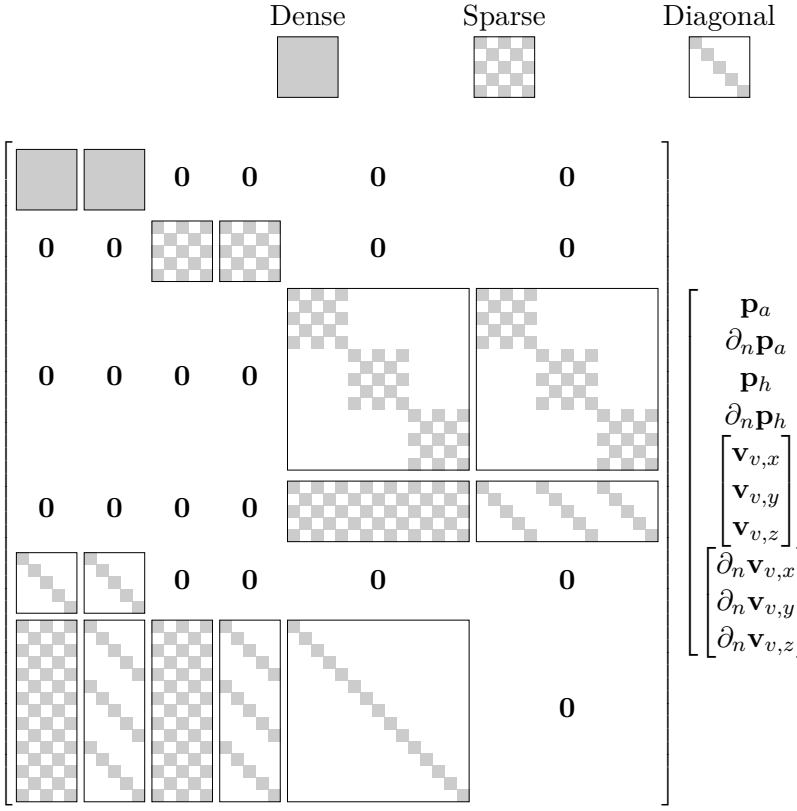


Fig. 5: Elapsed wall clock times measured from 20 evaluations of each of the lossy BEM algorithms. The bars indicate a cumulation of the runtimes for the system assembly and solution, the setup of the right-hand side, and the steps required to extend the system for recovering the condensed viscous velocities. Shown are the total runtimes for a specific mesh (left), and, for three different meshes, the relative share of each of the three computation steps (right). Only the maximum frequency of 1000 Hz is looked into as this will require the most GMRes iterations and therefore have a lesser time advantage when compared to the direct approaches.

As shown in the left plot of Fig. 5, even when including the costly extension step, the runtime reduction due to condensation and iterative solution improving the scalability remains strong when compared to the other lossy BEM variants. An opposite trend regarding the relative importance of the extension step is observed for the $1n$ - direct case. Although the absolute costs for recovering the viscous velocities stay the same, the solution via factorization scales worse and therefore dominates the time effort. The remaining variants of the lossy BEM spend almost all runtime on the assembly and solution.

6.4. *Conditioning*

To understand the reason behind the changing convergence for different condensation levels, we analyze the conditioning of the systems. Tab. 4 features the condition numbers κ of the system matrices for each of the condensation levels and test cases up to 5590 discretization nodes. Due to the high computational costs for calculating the condition numbers and the problem size having no visible impact, we refrain from extending the study to finer meshes. While the original uncondensed system suffers from ill-conditioning with κ being of the order $> 10^9$, the system reduction via Schur complement relieves this issue gradually. Unfolding the matrix structure from Eq. (39) as



(59)

visualizes the cause for the initial ill-conditioning. Coupling the acoustic, viscous, and thermal modes results in an inhomogeneous matrix structure with building blocks of dense,

sparse, and diagonal nature. Note that the diagonal matrices stemming from the coordinate transformation with $\mathbf{N}$ can have close-to-zero entries on the diagonal if the nodal normal vectors align with the coordinate axes. As the solution quantities are of vastly different orders of magnitude, so are the matrix coefficients. For the $4n$ - system as defined in Eq. (46), we can already see an improvement to $\kappa < 10^7$ caused by the changing structure as we apply the Schur complement. While the first reduction does not guarantee convergence of iterative solvers yet, the final stage of the global lossy BEM formulation, as stated in Eq. (50), shows favorable condition numbers in accordance with the convergence behavior observed during the benchmarking runs, cf. Tab. 3. Furthermore, we notice an adverse influence of the frequency for the fully condensed system. As such, the number of iterations required to find a solution increases for higher frequencies.

Table 4: Condition numbers of global system matrix for different condensation levels at frequencies of 100 Hz and 1000 Hz.

		Number of nodes n				
System size	f/Hz	494	930	1682	2642	5590
$10n \times 10n$	100	$1.55 \cdot 10^{10}$	$1.57 \cdot 10^{10}$	$1.58 \cdot 10^{10}$	$1.55 \cdot 10^{10}$	$1.53 \cdot 10^{10}$
	1000	$3.57 \cdot 10^{9}$	$3.66 \cdot 10^{9}$	$3.88 \cdot 10^{9}$	$3.66 \cdot 10^{9}$	$3.77 \cdot 10^{9}$
$4n \times 4n$	100	$2.99 \cdot 10^{6}$	$3.01 \cdot 10^{6}$	$3.03 \cdot 10^{6}$	$3.03 \cdot 10^{6}$	$3.04 \cdot 10^{6}$
	1000	$9.89 \cdot 10^{7}$	$3.12 \cdot 10^{7}$	$2.42 \cdot 10^{7}$	$2.44 \cdot 10^{7}$	$2.57 \cdot 10^{7}$
$1n \times 1n$	100	2.71	2.71	2.71	2.70	2.71
	1000	36.31	34.45	35.64	35.85	36.35

When looking deeper into the characteristics of the matrix-building blocks, we see a correlation between the dense matrices and the overall conditioning of the $1n$ - system. For $n = 2642$ nodes and a frequency of 1000 Hz, the discrete single and double layer potentials show condition numbers of

$$\kappa(\mathbf{G}_a) \approx 62.9413\,, \quad \kappa(\mathbf{H}_a) \approx 35.8337\,, \tag{60}$$

$$\kappa(\mathbf{G}_h) \approx 1.0024\,, \quad \kappa(\mathbf{H}_h) \approx 1.0004\,, \tag{61}$$

$$\kappa(\mathbf{G}_v) \approx 1.0020\,, \quad \kappa(\mathbf{H}_v) \approx 1.0004\,. \tag{62}$$

As such, the properties of the final system matrix seem to be mainly dominated by the acoustic double-layer potential. Note that all $\mathbf{H}_\bullet$ matrices include the contribution from the integral free term. The matrices for the viscous and thermal modes have condition numbers of close to one, explaining the fast convergence of the inner GMRes loops.

7. Conclusions and Future Work

A modified formulation of the direct three-dimensional boundary element method for time-harmonic acoustics with viscothermal losses has been presented in this paper. By redefining the coupling of the viscous, thermal, and acoustic wave fields on the boundary in global space, a previously needed coordinate transformation[11] becomes redundant. As such, the change of basis facilitates a software implementation of the method and removes the necessity for computing the nodal tangential vectors. The new method is validated, and its performance is analyzed based on an academic test case that shows a notable impact from boundary layer effects: an oscillating sphere. We investigate seven different mesh configurations at two frequencies and compare the results to an analytical solution.

To counteract the system's ill-conditioning, which has been observed for both the former local formulation and the proposed global one, we apply the Schur complement. An analysis of the matrix properties reveals that the ill-conditioning can be traced back to the inhomogeneous block structure and different orders of magnitude of the solution quantities. A systematic condensation of the internal unknowns proves an effective measure for adjusting the conditioning of the system matrix. A complete condensation with the acoustic pressure as remaining unknown results in a system matrix whose properties are mainly dominated by the dense acoustic double-layer potentials. Furthermore, the reduction in system size carries a memory advantage. To mitigate the computational costs of solving the coupled system directly, which has been inevitable as of now, we set up a nested iterative solution scheme that employs a restarted GMRes solver in three stages. The inner loops of the multilevel solver operate on the viscothermal matrices that show sparse matrix patterns due to the limited spatial reach of the boundary layer effects. The iteration counts remain constant throughout the test cases. The outer loop performs a dense matrix-vector product, and its convergence varies with frequency and mesh discretization. In general, a fast convergence for the solution scheme is observed, resulting from the improved conditioning via Schur complement. Evaluation of the discrete surface error yields the expected convergence behavior, confirming that the new global Schur complement BEM is stable and sufficiently accurate when solved iteratively.

Furthermore, an in-depth analysis of the performance with respect to execution time and memory consumption reveals the computational scaling of the newly developed method. The benchmarking measurements underline that combining the static condensation and iterative scheme reduces the order of the algorithmic complexity for the system assembly and solution. As such, the new method is computationally less expensive and therefore better suited for practical applications in acoustics. The next steps should include testing the method for larger applications relevant in viscothermal acoustics, such as metamaterials[6,7] and sound barriers.[44] Of particular interest are structures composed of narrow gaps and small acoustic cavities where the spatial significance of the losses extends beyond the neighboring elements. We consequently expect a change in the sparsity pattern of the viscothermal matrices.

While the iterative fully condensed system reduces the computational complexity significantly by avoiding any dense matrix inversions as well as dense matrix-matrix products, it still suffers from computational drawbacks rooted in the characteristics of the acoustic single- and double-layer potentials. Computing the acoustic part of the matrix-building blocks for assembling the coupled system requires $\mathcal{O}(n^2)$ operations, resulting in quadratic time and space complexities. A similar effort is spent on the dense algebraic operations included in the outer linear operator for the iterative solution. To tackle this issue, future work should focus on incorporating acceleration techniques like the fast multipole method,[45] that improve the scalability by approximating the matrix-vector products. The new formulation is already set up such that the acoustic matrices are isolated and can easily be treated with a low-rank approximation that does not require an explicit assembly and storage of the full matrix. Furthermore, we suggest generalizing the method formulation to guarantee applicability for all kinds of test cases, such as non-smooth geometries and studies at eigenfrequencies of the corresponding interior problems. As such, the use of discontinuous elements and the method by Burton and Miller should be evaluated.[46] Additionally, a close examination of the current technique for nearly singular and singular integration might reveal alternative methods better suited for the fast decaying viscothermal kernel functions. Lastly, applying preconditioners like the ILU - type preconditioner[29] or the diagonal preconditioner[28] can help with the slower convergence of the extension step for recovering the velocities.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the grant agreement No 468269586 and by the Technical University of Denmark through a joint research project as part of the EuroTech Universities Alliance. Furthermore, the authors gratefully acknowledge the computational and data resources provided by the Leibniz Supercomputing Centre.

References

1. S. Marburg, Boundary Element Method for Time–Harmonic Acoustic Problems, in *Computational Acoustics*, ed. M. Kaltenbacher (Springer, 2017), CISM International Centre for Mechanical Sciences, ch. 3, pp. 69–159.
2. T. Frenzel, J. David Brehm, T. Bückmann, R. Schittny, M. Kadic and M. Wegener, Three-dimensional labyrinthine acoustic metamaterials, *Applied Physics Letters* **103** (2013) 061907.
3. M. D. Guild, V. M. García-Chocano, W. Kan and J. Sánchez-Dehesa, Acoustic metamaterial absorbers based on multilayered sonic crystals, *Journal of Applied Physics* **117** (2015) 114902.
4. N. Jiménez, W. Huang, V. Romero-García, V. Pagneux and J.-P. Groby, Ultra-thin metamaterial for perfect and quasi-omnidirectional sound absorption, *Applied Physics Letters* **109** (2016) 121902.
5. M. Molerón, M. Serra-Garcia and C. Daraio, Visco-thermal effects in acoustic metamaterials: from total transmission to total reflection and high absorption, *New Journal of Physics* **18** (2016) 033003.

6. V. Cutanda Henríquez, P. R. Andersen, J. S. Jensen, P. M. Juhl and J. Sánchez-Dehesa, A Numerical Model of an Acoustic Metamaterial Using the Boundary Element Method Including Viscous and Thermal Losses, *Journal of Computational Acoustics* **25** (2017) 1750006.
7. V. Cutanda Henríquez, V. M. García-Chocano and J. Sánchez-Dehesa, Viscothermal losses in double-negative acoustic metamaterials, *Physical Review Applied* **8** (2017) 014029.
8. P. R. Andersen, V. Cutanda Henríquez and N. Aage, Shape optimization of micro-acoustic devices including viscous and thermal losses, *Journal of Sound and Vibration* **447** (2019) 120–136.
9. A. D. Pierce, *Acoustics: An Introduction to Its Physical Principles and Applications* (Springer International Publishing, Cham, 2019), 3rd edition.
10. K. U. I. Philip M. Morse, *Theoretical Acoustics* (Princeton University Press, 1987).
11. V. Cutanda Henríquez and P. R. Andersen, A three-dimensional acoustic boundary element method formulation with viscous and thermal losses based on shape function derivatives, *Journal of Theoretical and Computational Acoustics* **26** (2018) 1850039.
12. W. R. Kampinga, Y. H. Wijnant and A. de Boer, Performance of several viscothermal acoustic finite elements, *Acta Acustica united with Acustica* **96** (2010) 115–124.
13. W. R. Kampinga, Y. H. Wijnant and A. de Boer, An efficient finite element model for viscothermal acoustics, *Acta Acustica united with Acustica* **97** (2011) 618–631.
14. V. Cutanda Henríquez and P. M. Juhl, An axisymmetric boundary element formulation of sound wave propagation in fluids including viscous and thermal losses, *The Journal of the Acoustical Society of America* **134** (2013) 3409–3418.
15. P. R. Andersen, V. Cutanda Henríquez, N. Aage and S. Marburg, A Two-Dimensional Acoustic Tangential Derivative Boundary Element Method Including Viscous and Thermal Losses, *Journal of Theoretical and Computational Acoustics* **26** (2018) 1850036.
16. H. Tijdeman, On the propagation of sound waves in cylindrical tubes, *Journal of Sound and Vibration* **39** (1975) 1–33.
17. W. M. Beltman, P. J. M. van der Hoogt, R. M. E. J. Spiering and H. Tijdeman, Implementation and experimental validation for a new viscothermal acoustic finite element method for acousto-elastic problems, *Journal of Sound and Vibration* **216** (1998) 159–185.
18. R. Bossart, N. Joly and M. Bruneau, Hybrid numerical and analytical solutions for acoustic boundary problems in thermo-viscous fluids, *Journal of Sound and Vibration* **263** (2003) 69–84.
19. M. Berggren, A. Bernland and D. Noreland, Acoustic boundary layers as boundary conditions, *Journal of Computational Physics* **371** (2018) 633–650.
20. K. Schmidt and A. Thöns-Zueva, Impedance boundary conditions for acoustic time-harmonic wave propagation in viscous gases in two dimensions, *Mathematical Methods in the Applied Sciences* **45** (2022) 7404–7425.
21. M. Paltorp, V. Cutanda Henríquez, N. Aage and P. R. Andersen, A reduced order series expansion for the BEM incorporating the boundary layer impedance condition, *Journal of Theoretical and Computational Acoustics* .
22. P. R. Andersen, V. Cutanda Henríquez and N. Aage, On the validity of numerical models for viscothermal losses in structural optimization for micro-acoustics, *Journal of Sound and Vibration* **547** (2023) 117455.
23. F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*, volume 132 of *Applied Mathematical Sciences* (Springer Verlag, 1998).
24. S. Preuss, C. Gurbuz, C. Jelich, S. K. Baydoun and S. Marburg, Recent Advances in Acoustic Boundary Element Methods, *Journal of Theoretical and Computational Acoustics* **30** (2022) 2240002.
25. M. Bruneau, P. Herzog, J. Kergomard and J. D. Polack, General formulation of the dispersion equation in bounded visco-thermal fluid, and application to some simple geometries, *Wave Motion* **11** (1989) 441–451.

26. G. Kirchhoff, Über den Einfluss der Wärmeleitung in einem Gase auf die Schallbewegung, *Annalen der Physik und Chemie* **210** (1868) 177–193.
27. Y. Saad and M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM Journal on Scientific and Statistical Computing* **7** (1986) 856–869.
28. S. Marburg and S. Schneider, Performance of iterative solvers for acoustic problems. part I. Solvers and effect of diagonal preconditioning, *Engineering Analysis with Boundary Elements* **27** (2003) 727–750.
29. S. Schneider and S. Marburg, Performance of iterative solvers for acoustic problems. part II. Acceleration by ILU-type preconditioner, *Engineering Analysis with Boundary Elements* **27** (2003) 751–757.
30. G. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, 2013), 4th edition.
31. C. Brebbia, *The Boundary Element Method for Engineers* (Pentech Press, 1984).
32. V. Cutanda Henríquez and P. M. Juhl, Implementation of an acoustic 3D BEM with visco-thermal losses, in *Proceedings of the 42nd International Congress on Noise Control Engineering (INTERNOISE 2013)*, ed. W. Talasch (Austrian Noise Abatement Association, Innsbruck, Austria, 2013), pp. 5576–5583.
33. N. Atalla and F. Sgard, *Finite Element and Boundary Methods in Structural Acoustics and Vibration* (CRC Press, 2015).
34. D. Brunner, M. Junge and L. Gaul, A comparison of FE-BE coupling schemes for large-scale problems with fluid-structure interaction, *International Journal for Numerical Methods in Engineering* **77** (2009) 664–688.
35. F. Zhang, ed., *The Schur Complement and Its Applications* (Springer-Verlag, 2005).
36. S. Temkin and J. D. N. Cheeke, *Elements of Acoustics* (American Association of Physics Teachers (AAPT), 1982).
37. COMSOL AB, Stockholm, Sweden, COMSOL Multiphysics® v. 6.0, 2022.
38. O. Steinbach, *Numerical Approximation Methods for Elliptic Boundary Value Problems: Finite and Boundary Elements* (Springer, 2008).
39. S. Marburg, Six boundary elements per wavelength: Is that enough?, *Journal of Computational Acoustics* **10** (2002) 25–51.
40. J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah, Julia: A fresh approach to numerical computing, *SIAM Review* **59** (2017) 65–98.
41. 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* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999), 3rd edition.
42. L. S. Blackford, A. Petitet, R. Pozo, K. Remington, R. C. Whaley, J. Demmel, J. Dongarra, I. Duff, S. Hammarling, G. Henry et al., An Updated Set of Basic Linear Algebra Subprograms (blas), *ACM Transactions on Mathematical Software* **28** (2002) 135–151.
43. N. J. Higham, *Functions of Matrices* (Society for Industrial and Applied Mathematics, 2008).
44. C. Jelich, M. Karimi, N. Kessissoglou and S. Marburg, Efficient solution of block Toeplitz systems with multiple right-hand sides arising from a periodic boundary element formulation, *Engineering Analysis with Boundary Elements* **130** (2021) 135–144.
45. N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions* (Elsevier, 2004).
46. S. Marburg, The Burton and Miller Method: Unlocking Another Mystery of its Coupling Parameter, *Journal of Computational Acoustics* **24** (2016) 1550016.
47. C. Karra and M. B. Tahar, An integral equation formulation for boundary element analysis of propagation in viscothermal fluids, *The Journal of the Acoustical Society of America* **102** (1997)

1311–1318.
48. K. Rasmussen, Calculation methods for the physical properties of air used in the calibration of microphones, Technical Report PL-11b, Department of Acoustic Technology, Technical University of Denmark, 1997.

Appendix

A. Viscothermal Coefficients and Material Model

The following section elaborates on the modal wavenumbers $k_\bullet$ and model parameters $\phi_\bullet$ and $\tau_\bullet$ that are needed to construct the viscothermal BEM model. We focus on the expressions corresponding to a negative harmonic time dependence, as has been assumed throughout this work. The equivalent expressions for $\mathrm{e}^{\mathrm{i}\omega t}$ are outlined in Ref. 14. The viscothermal parameters emerge from the conservation laws for mass, momentum, and energy, which constitute the starting point for the model derivation. When cast into the form of time-harmonic Helmholtz equations as given in Eqs. (11)-(13), the material properties originally included in the governing equations become part of complex-valued wavenumbers. As detailed in Refs. 25 and 47, the dispersion relations for the acoustic, thermal, and viscous modes read

$$k_a^2 = \frac{k^2}{1 - \mathrm{i}k\left[l_v + (\gamma-1)\, l_h\right] - k^2 l_h\,(\gamma-1)\,(l_h - l_v)}\,, \tag{A.1}$$

$$k_h^2 = \frac{\mathrm{i}k}{l_h\left[1 + \mathrm{i}k\,(\gamma-1)\,(l_h - l_v)\right]}\,, \tag{A.2}$$

$$k_v^2 = \frac{\mathrm{i}\rho_0\,\omega}{\mu}\,, \tag{A.3}$$

with the thermal and viscous characteristic lengths defined as

$$l_v = \frac{\eta + \dfrac{4}{3}\mu}{\rho_0\, c}\,, \quad l_h = \frac{\lambda_h}{\rho_0\, c\, c_p}\,. \tag{A.4}$$

Here, ω is the angular frequency, c the speed of sound, $k = \omega/c$ the isentropic wavenumber, ρ_0 the static density, λ_h the thermal conductivity, μ the shear viscosity, $\eta \approx 0.6\mu$[9] the bulk viscosity, c_p the specific heat capacity at constant pressure, c_v the specific heat capacity at constant volume, and $\gamma = c_p/c_v$ the specific heat ratio. We further define the coefficients appearing in the coupling conditions (Eqs. (6) and (7)) as

$$\tau_a = \frac{\gamma-1}{\beta\gamma}\frac{1}{1 + \mathrm{i}l_h\dfrac{k_a^2}{k}}\,, \qquad \tau_h = \frac{\gamma-1}{\beta\gamma}\frac{1}{1 + \mathrm{i}l_h\dfrac{k_h^2}{k}}\,, \tag{A.5}$$

$$\phi_a = \frac{-\mathrm{i}}{\rho_0\,\omega}\frac{1}{1 + \mathrm{i}l_v\dfrac{k_a^2}{k}}\,, \qquad \phi_h = \frac{-\mathrm{i}}{\rho_0\,\omega}\frac{1}{1 + \mathrm{i}l_v\dfrac{k_h^2}{k}}\,, \tag{A.6}$$

where $\beta = \rho_0(c_p - c_v)$ denotes the thermal expansion coefficient. The material properties in this study have been evaluated following the description in Ref. 48. For air as propagation medium, assuming reference ambient conditions of

- Static temperature: $T_0 = 293.15\,\mathrm{K}$
- Static pressure: $p_0 = 101\,325\,\mathrm{Pa}$
- Relative humidity: $\varphi = 50\,\%$

we assign the following material properties

- Static density: $\rho_0 = 1.199\,\mathrm{kg/m^3}$
- Speed of sound: $c = 343.986\,\mathrm{m/s}$
- Thermal conductivity: $\lambda_h = 251.778 \cdot 10^{-4}\,\mathrm{W/(m\,K)}$
- Shear viscosity: $\mu = 181.267 \cdot 10^{-7}\,\mathrm{Pa\,s}$
- Heat capacity at const. pressure: $c_p = 1012.368\,\mathrm{J/(kg\,K)}$
- Heat capacity at const. volume: $c_v = 722.552\,\mathrm{J/(kg\,K)}$

to compute the solution of all test cases considered in this work.

B. Further Explanations on the Interpolation Function Derivatives

On all boundary elements and their respective parametric representation, there is a local coordinate $\boldsymbol{\xi}$ for which $\mathbf{x}(\boldsymbol{\xi})$ marks the location of a collocation point in global space. Say that this is the i^{th} collocation point being part of the j^{th} element, then $\boldsymbol{\xi}^{j,i}$ evaluated on the parametric reference element determines the global position of this collocation point as $\mathbf{x}^j\left(\boldsymbol{\xi}^{j,i}\right) = \mathbf{x}_i$. From this discrete representation of the nodal position, together with assuming $\mathbf{p}_a^j = \mathbf{L}^j \mathbf{p}_a$, where $\mathbf{L}^j$ represents a sparse matrix that extracts the correct values of $\mathbf{p}_a$, it follows that

$$\nabla^{\parallel}_{\mathbf{x}} p_a\left(\mathbf{x}^j(\boldsymbol{\xi}^{j,i})\right) = \nabla^{\parallel}_{\mathbf{x}} p_a(\mathbf{x}_i) = \begin{bmatrix} \frac{\partial x}{\partial \xi_1} & \frac{\partial y}{\partial \xi_1} & \frac{\partial z}{\partial \xi_1} \\ \frac{\partial x}{\partial \xi_2} & \frac{\partial y}{\partial \xi_2} & \frac{\partial z}{\partial \xi_2} \\ (\frac{\partial \mathbf{x}}{\partial \xi_1} \times \frac{\partial \mathbf{x}}{\partial \xi_2})^\top \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \boldsymbol{\Phi}}{\partial \xi_1}^\top \\ \frac{\partial \boldsymbol{\Phi}}{\partial \xi_2}^\top \\ \mathbf{0}^\top \end{bmatrix} \mathbf{L}^j \mathbf{p}_a = \begin{bmatrix} \mathbf{D}_x^{j,i} \\ \mathbf{D}_y^{j,i} \\ \mathbf{D}_z^{j,i} \end{bmatrix} \mathbf{p}_a \,. \tag{B.1}$$

In the case of continuous elements, the collocation point is connected to multiple elements, and the interpolation function derivative is chosen to be the average contribution from each of the connected elements. As such, the i^{th} rows of the $\mathbf{D}_\bullet$ - matrices can be computed via

$$\mathbf{D}_x^i = \frac{1}{n_e(i)} \sum_{j=1}^{n_e(i)} \mathbf{D}_x^{j,i} \,, \quad \mathbf{D}_y^i = \frac{1}{n_e(i)} \sum_{j=1}^{n_e(i)} \mathbf{D}_y^{j,i} \,, \quad \mathbf{D}_z^i = \frac{1}{n_e(i)} \sum_{j=1}^{n_e(i)} \mathbf{D}_z^{j,i} \,, \tag{B.2}$$

where $n_e(i)$ denotes the number of elements that belong to collocation point i. Note that in the case of discontinuous elements, the collocation point is only part of a single element, meaning that no averaging is needed and Eq. (B.1) directly computes the rows of the $\mathbf{D}_\bullet$ - matrices.

C. Analytical Model of a Transversely Oscillating Sphere

For the sake of completeness, we include the analytical model that serves as a reference for the test case studied in this work. The analytical solution considers boundary losses due to viscosity but omits effects that arise from heat conduction. The formulas, as stated below, closely follow the derivation in Sec. 6.9 of Ref. 36. We consider a rigid sphere of radius a whose center oscillates harmonically along the z - axis. The transverse motion in the viscous fluid can be described conveniently in terms of polar-spherical coordinates (r, θ, φ), where the polar axis of the sphere ($\theta = 0°$) matches the z - direction of the global coordinate system. We follow the convention specified by ISO 80000-2:2019 to define the spherical coordinate system. The boundary condition at the sphere's surface is defined as

$$v^r(r = a, \theta) = v_0 \cos\theta\,, \tag{C.1}$$

$$v^\theta(r = a, \theta) = -v_0 \sin\theta\,, \tag{C.2}$$

where v_0 denotes the amplitude of excitation. Since we assume a uniform excitation in z - direction without slip, v_0 is equivalent to $v_{b,z}$. A solution of the linear acoustic equations satisfying the boundary condition is given by

$$\begin{aligned} p(r,\theta) &= \mathrm{i}\rho_0\omega \sum_{n=0}^{\infty} \mathrm{i}^n (2n+1) A_n h_n^{(1)}(kr) P_n(\cos\theta) \\ &\approx p_a\,, \end{aligned} \tag{C.3}$$

$$\begin{aligned} v^r(r,\theta) &= \sum_{n=0}^{\infty} \mathrm{i}^n (2n+1) k \left(A_n h_n^{(1)\prime}(kr) - n(n+1)\frac{B_n}{kr} h_n^{(1)}(k_v r) \right) P_n(\cos\theta) \\ &\approx v_a^r + v_v^r\,, \end{aligned} \tag{C.4}$$

$$\begin{aligned} v^\theta(r,\theta) &= \sum_{n=0}^{\infty} \mathrm{i}^n (2n+1) \frac{1}{r} \left(A_n h_n^{(1)}(kr) - B_n \left(k_v r h_n^{(1)\prime}(k_v r) + h_n^{(1)}(k_v r) \right) \right) P_n^1(\cos\theta) \\ &\approx v_a^\theta + v_v^\theta\,. \end{aligned} \tag{C.5}$$

Given that the external medium is unbounded, the acoustic field around the sphere is symmetric about the polar axis, and the solution therefore independent from the azimuth angle φ. Since the analytical model generally neglects heat transfer, the solution quantities built from the superimposed modal wave fields lack the thermal contributions and only depend on the isentropic and viscous wavenumbers k and k_v. Eqs. (C.3)-(C.5) include the spherical Hankel function of the first kind and n^{th} order $h_n^{(1)}$, its first derivative with respect to the argument $h_n^{(1)\prime}$, and the coefficients A_n and B_n. Furthermore, the solutions depend on P_n and P_n^1 representing the Legendre polynomial of degree n and the associated Legendre polynomial of first order, respectively. We truncate the series at $n = 1$, focusing on the

dominating first-order terms. As such, the solution for p_a and v_v simplifies to

$$p_a(r,\theta) \approx -3\rho_0\omega A_1 h_1^{(1)}(kr)\cos\theta\,, \tag{C.6}$$

$$v_v^r(r,\theta) \approx -6\mathrm{i}\frac{B_1}{r}h_1^{(1)}(k_v r)\cos\theta\,, \tag{C.7}$$

$$v_v^\theta(r,\theta) \approx \frac{3\mathrm{i}}{r}B_1\left(k_v r h_1^{(1)\prime}(k_v r) + h_1^{(1)}(k_v r)\right)\sin\theta\,. \tag{C.8}$$

For arbitrary arguments α, the spherical Hankel function of first order and its derivative are calculated via

$$h_1^{(1)}(\alpha) = -\frac{\mathrm{e}^{\mathrm{i}\alpha}}{\alpha^2}(\alpha+\mathrm{i})\,, \tag{C.9}$$

$$h_1^{(1)\prime}(\alpha) = \frac{\mathrm{e}^{\mathrm{i}\alpha}}{\alpha^3}\left(2\alpha+\mathrm{i}(2-\alpha^2)\right)\,, \tag{C.10}$$

respectively. Constructing a system of equations that computes the boundary solution allows to find the values of the coefficients A_1 and B_1. Setting Eqs. (C.3)-(C.5) at $r=a$ and for $n=1$ equal to the boundary conditions in Eqs. (C.1) and (C.2) and reformulating with respect to the solution vector $[A_1, B_1]^\top$ leads to

$$\begin{bmatrix} h_1^{(1)\prime}(ka) & -\dfrac{2}{ka}h_1^{(1)}(k_v a) \\ h_1^{(1)}(ka) & -\left(k_v a h_1^{(1)\prime}(k_v a) + h_1^{(1)}(k_v a)\right) \end{bmatrix} \begin{bmatrix} A_1 \\ B_1 \end{bmatrix} = \begin{bmatrix} \dfrac{v_0}{3\mathrm{i}k} \\ \dfrac{a v_0}{3\mathrm{i}} \end{bmatrix}. \tag{C.11}$$

Solving the above naively will be numerically unstable due to the large values of k_v. Instead, we insert the expressions for the Hankel functions as given in Eqs. (C.9) and (C.10) and apply Cramer's rule to find A_1. After calculating A_1 via

$$A_1 = -\frac{v_0}{3\mathrm{i}}k^2 a \mathrm{e}^{-\mathrm{i}ka}\frac{3k_v a + 3\mathrm{i} - \mathrm{i}k_v^2 a^2}{k_v^2(k^2a^2-2) - k^2 + \mathrm{i}k_v ka(k+2k_v)}\,, \tag{C.12}$$

we can plug the solution for the coefficient into Eq. (C.11) and rearrange as

$$v_v^r(r=a,\theta) \approx \left(v_0 - 3\mathrm{i}kA_1 h_1^{(1)\prime}(ka)\right)\cos\theta\,, \tag{C.13}$$

$$v_v^\theta(r=a,\theta) \approx \left(-v_0 + \frac{3\mathrm{i}}{a}A_1 h_1^{(1)}(ka)\right)\sin\theta\,, \tag{C.14}$$

to determine the solution for the viscous velocity on the boundary. Hence, the viscous component containing the unknown coefficient B_1 and the problematic Hankel function $h_1^{(1)}(k_v a)$ is alternatively evaluated as the difference between the total velocity prescribed on the surface and the acoustic velocity depending on $h_1^{(1)}(ka)$ and A_1. Eqs. (C.13), (C.14), and (C.6) at $r=a$ provide a solution that can be used for validating the numerical method developed in this work, cf. Fig. 2.

Paper J3

Journal of Theoretical and Computational Acoustics

Large-Scale Boundary Element Computations Including Viscothermal Losses

Mikkel Paltorp[a,*], Simone Preuss[b,†], Vicente Cutanda Henríquez[b], Steffen Marburg[b]

[a] *Acoustic Technology Group*
Department of Electrical and Photonics Engineering
The Technical University of Denmark
Kgs. Lyngby, DK 2800, Denmark
[b] *Chair of Vibroacoustics of Vehicles and Machines*
Department of Engineering Physics and Computation
TUM School of Engineering and Design
Technical University of Munich
Boltzmannstraße 15, 85748 Garching, Germany
**mpasc@dtu.dk*
† *simone.preuss@tum.de*

Received (Day Month Year)
Revised (Day Month Year)

Accurate acoustical simulations in the vicinity of boundaries require the inclusion of viscous and thermal effects resulting from fluid-boundary interactions. The additional complexity of the model comes with an expected increase in the computational demands of the simulation. Recently a paper by the authors has shown improvements in the boundary element description of the Kirchhoff decomposition used to include the viscous and thermal effects. The new formulation, as opposed to previous formulations, made it possible to utilize the sparsity structure present in the system of equations during the solution phase. As a result, the proposed formulation was significantly more computationally efficient than previous formulations. However, two fully populated matrices arising from the acoustical mode were unsurprisingly shown to be a computational bottleneck. This study shows that it is possible to eliminate the computational bottleneck by employing either the fast multipole method or hierarchical matrices as approximations to the acoustical matrices. The computational scaling of the new formulation is validated using meshes of various refinements. The findings show significant improvements in both the required memory consumption and computational effort, making it possible to solve problems of sizes that far exceed anything previously possible. As such, the ideas presented in this paper serve as important first steps in the direction of large-scale boundary element simulations, including viscous and thermal losses, using the Kirchhoff decomposition.

Keywords: boundary element method and Kirchhoff decomposition and viscothermal losses and fast multipole method and hierarchical matrices

1. Introduction

When performing acoustical simulations in the vicinity of boundaries, it is crucial to consider the impact of viscous and thermal effects of the fluid resulting from its interaction with the boundary.[1] The dissipation occurs primarily in the transitional phase, between the boundary and the bulk, called the boundary layer. For audible frequencies, the thickness of the boundary layers is approximately one hundred micrometers, making it significant when

modeling acoustic micro-devices like hearing aids or transducers.

Traditionally, viscous and thermal losses have been addressed using the fully linearized Navier-Stokes (FNLS) equations for which a numerical solution can be obtained using the finite element method (FEM).[2–4] However, solving these equations can be computationally intensive, because of the inclusion of additional degrees of freedom (DOF) for the temperature and the velocity. Another limitation is the requirement for fine meshing near the boundary in order to accurately capture the microscale physics, which further increases the number of DOF. To overcome these challenges and avoid the need for fine boundary meshing, an alternative to the finite element method known as the boundary element method (BEM) can be used.[5–7] This method is based on the Kirchhoff decomposition (subsection 2.1), which separates the FLNS equations into one vectorial and two scalar Helmholtz equations that are then coupled through a set of boundary conditions (subsection 2.2).[8]

This approach was initially developed for axisymmetrical problems, but was later generalized to full three-dimensional objects.[9,10] The underlying model included fully populated and sparse matrices of which the sparse structure was only later utilized to increase the computational efficiency of the assembly phase.[11] Recently, the authors showed that the sparsity structure can be further utilized to increase the computational efficiency of the solution phase.[12] In this paper we show that the new formulation can be further improved by exploiting the structure of the fully populated matrices. In particular, we apply both the fast multipole method (FMM) and hierarchical matrices ($\mathcal{H}$-matrices) to accelerate the matrix-vector products of the two otherwise fully populated matrices.[13–16] The results show that the memory and computational complexity of the model follow that of pure acoustical simulations.

The paper is structured as follows: In section 2 a short introduction to the boundary element method and the Kirchhoff decomposition is given. Then in section 3 we give a description of how the fast multipole method and hierarchical matrices can be applied to relieve the computational bottleneck arising from the acoustical mode. Finally, in section 4, we investigate the scalability of the computational model on meshes of various refinements.

2. Boundary Element Method Formulation including Losses

2.1. *The Kirchhoff Decomposition*

An equivalent formulation of the fully linearized Navier-Stokes equations is the Kirchhoff decomposition, which splits the equation into three so-called modes, each satisfying its own Helmholtz equation

$$\text{Acoustic Mode:} \quad (\Delta + k_a^2) p_a(\mathbf{x}) = 0, \tag{1}$$

$$\text{Thermal Mode:} \quad (\Delta + k_h^2) p_h(\mathbf{x}) = 0, \tag{2}$$

$$\text{Viscous Mode:} \quad (\Delta + k_v^2) \mathbf{v}_v(\mathbf{x}) = \mathbf{0}, \quad \text{with} \quad \nabla \cdot \mathbf{v}_v(\mathbf{x}) = 0. \tag{3}$$

The three modal wavenumbers (k_a, k_h, and k_v) all depend on the lossless wavenumber (k) and the physical properties of the fluid, such as the thermal conductivity, specific heat capacity under constant pressure and the shear/bulk viscosity coefficients.[8] The total pressure and velocity can be extracted as the sum of the contributions of each of the three modes

$$p_t = p_a + p_h, \tag{4}$$

$$\mathbf{v}_t = \mathbf{v}_a + \mathbf{v}_h + \mathbf{v}_v. \tag{5}$$

2.2. *Coupling the Modes Through Boundary Conditions*

In the case of the fluid being air, it is valid to prescribe the so-called isothermal boundary condition due to the higher heat capacity of the boundary material. Mathematically, this condition can be written as

$$p_a(\mathbf{x})\tau_a + p_h(\mathbf{x})\tau_h = 0, \quad \mathbf{x} \in \Gamma, \tag{6}$$

where τ_a and τ_h are complex and depend on the frequency and properties of the fluid and Γ is the boundary of the domain. Furthermore, due to frictional forces the air sticks to the surface of objects. As such, it is common to prescribe a no-slip boundary condition which states that the total velocity of the fluid on the boundary is equal to the velocity of the boundary. Mathematically, this can be written as

$$\mathbf{v}_b(\mathbf{x}) = \phi_a \nabla p_a(\mathbf{x}) + \phi_h \nabla p_h(\mathbf{x}) + \mathbf{v}_v(\mathbf{x}), \quad \mathbf{x} \in \Gamma, \tag{7}$$

where $\mathbf{v}_b(\mathbf{x})$ is the surface velocity at $\mathbf{x}$ while ϕ_a and ϕ_h are complex and depend on the frequency as well as the physical properties of the fluid.

2.3. *The Boundary Element Method*

The boundary element method can be used to solve the Helmholtz equations through the discretization of the Kirchhoff-Helmholtz boundary integral equation. Taking the acoustical mode as an example, this boundary integral has the form

$$\zeta(\mathbf{x})p_a(\mathbf{x}) = \int_\Gamma G(k_a, \mathbf{x}, \mathbf{y})\partial_{\mathbf{n}} p(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}} - \int_\Gamma \frac{\partial G(k_a, \mathbf{x}, \mathbf{y})}{\partial \mathbf{n}_{\mathbf{y}}} p_a(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}}, \quad \mathbf{x} \in \Omega \tag{8}$$

where $\mathbf{x}$ is a point in the domain Ω, $\mathbf{y}$ represent a point on the boundary Γ, $p_a(\mathbf{x})$ is the acoustical pressure, $\zeta(\mathbf{x})$ is the integral free form depending only on the geometry at $\mathbf{x}$ and $G(k_a, \mathbf{x}, \mathbf{y})$ is the Green's function. In the following text, the time dependence $e^{-\mathrm{i}\omega t}$ is chosen. As such the Green's function in three-dimensional space is given as[17]

$$G(k_a, \mathbf{x}, \mathbf{y}) = \frac{\exp\left(\mathrm{i}k_a \|\mathbf{x} - \mathbf{y}\|_2\right)}{4\pi \|\mathbf{x} - \mathbf{y}\|_2}. \tag{9}$$

The Kirchhoff-Helmholtz equation looks similar to (8) for the thermal and viscous modes, but with respectively the thermal and viscous wavenumbers in the Green's function. The numerical approximation of (8) requires two steps: First, a discretization of the boundary

into surface elements, and second, the establishment of an interpolation scheme for p_a and $\partial_{\mathbf{n}} p_a$. The first step starts with defining a set of elements (Γ^e) such that $\Gamma \approx \cup_{e=1}^{N} \Gamma^e$. These geometric elements are often parametrized by

$$\mathbf{x}^e(\mathbf{u}) = \mathbf{X}^e \mathbf{N}^e(\mathbf{u}) \in \Gamma^e, \quad \forall \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \in \mathcal{L}^e \subset \mathbb{R}^2, \tag{10}$$

where $\mathbf{X}^e$ is a matrix with columns equal to the so-called interpolation nodes of the geometry, $\mathbf{N}^e(\mathbf{u})$ is a column-vector containing the shape functions and $\mathcal{L}^e$ is the local element coordinates of the e^{th} element. Using this formulation, it becomes possible to approximate the boundary integral using a quadrature scheme as

$$\int_{\Gamma^e} f(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}} = \int_{\mathcal{L}^e} \mathcal{J}(\mathbf{u}) f(\mathbf{u}) \, \mathrm{d}\mathbf{u} \approx \sum_{j=1}^{Q} w_j \mathcal{J}(\mathbf{u}_j) f(\mathbf{u}_j), \tag{11}$$

where $(w_i, \mathbf{u}_i)$ is the i^{th} quadrature weight and node while $\mathcal{J}$ is the Jacobian of the element parametrization. In 3D the Jacobian can be thought of as a pointwise area deformation and is computed as

$$\mathcal{J}(\mathbf{u}) = \left\| \left(\mathbf{X}^e \frac{\mathrm{d}\mathbf{N}^e(\mathbf{u})}{\mathrm{d}u_1} \right) \times \left(\mathbf{X}^e \frac{\mathrm{d}\mathbf{N}^e(\mathbf{u})}{\mathrm{d}u_2} \right) \right\|_2. \tag{12}$$

The second step begins by introducing approximations of the pressure and its normal derivative as follows

$$p_a(\mathbf{x}) = \mathbf{T}(\mathbf{x}) \mathbf{p}_a, \quad \partial_{\mathbf{n}} p_a(\mathbf{x}) = \mathbf{T}(\mathbf{x}) \partial_{\mathbf{n}} \mathbf{p}_a, \quad \mathbf{x} \in \Gamma \tag{13}$$

where $\mathbf{T}(\mathbf{x})$ is a row-vector containing the global basis functions while $\mathbf{p}_a$ and $\partial_{\mathbf{n}} \mathbf{p}_a$ contain the nodal values of respectively the pressure and its normal derivative. In general, however, its easier to work with local basis functions meaning that the interpolation on the e^{th} element is instead given by

$$p_a(\mathbf{x}^e(\mathbf{u})) = \mathbf{T}^e(\mathbf{u}) \mathbf{L}^e \mathbf{p}_a, \quad \partial_{\mathbf{n}} p_a(\mathbf{x}^e(\mathbf{u})) = \mathbf{T}^e(\mathbf{u}) \mathbf{L}^e \partial_{\mathbf{n}} \mathbf{p}_a, \tag{14}$$

where $\mathbf{T}^e(\mathbf{u})$ is a row-vector containing the local basis functions and $\mathbf{L}^e$ is a sparse matrix that extracts and orders the values of $\mathbf{p}_a$ and $\partial_{\mathbf{n}} \mathbf{p}_a$ in a way that matches with the ordering of the local basis functions. Note that any reasonable implementation would never form $\mathbf{L}^e$ but would instead use the indexing of the vectors. Determining the values of $\mathbf{p}_a$ and $\partial_{\mathbf{n}} \mathbf{p}_a$ can be done in a variety of ways, the most common methods being the Nyström, collocation, and Galerkin approaches. The approach taken in this text is that of collocation. Here, a linear system of equations is generated by setting $\mathbf{x}$ in (8) equal to the so-called collocation points $(\{\mathbf{x}_k\}_{k=1}^{n}$. As such, for the collocation approach, $\mathbf{p}_a$ and $\partial_{\mathbf{n}} \mathbf{p}_a$ contain the values of the pressure and the normal derivative at the collocation points. In section 3 we explain in detail how the linear system is generated, but for now it is enough, think of the discrete form of (1) as

$$\mathbf{H}_a \mathbf{p}_a + \mathbf{G}_a \partial_{\mathbf{n}} \mathbf{p}_a = \mathbf{0}, \tag{15}$$

where $\mathbf{H}_a = -\mathbf{F}_a - \text{diag}(\boldsymbol{\zeta})$. Following a similar procedure, the discrete form of (2) and (3) becomes

$$\mathbf{H}_h \mathbf{p}_h + \mathbf{G}_h \partial_{\mathbf{n}} \mathbf{p}_h = \mathbf{0}, \tag{16}$$

$$\mathbf{H}_v \mathbf{v}_v + \mathbf{G}_v \partial_{\mathbf{n}} \mathbf{v}_v = \mathbf{0}. \tag{17}$$

Note that as opposed to the dense matrices in the discrete acoustical mode the matrices in the discrete thermal and viscous modes are sparse. The mathematical reason for this is due to the thermal and viscous wavenumbers being complex, with a large imaginary component, resulting in rapidly decaying Green's functions. A physical intuition here is that the physics that the two modes represent are local effects. Furthermore, the discrete form of the viscous mode is special as it comes from a vector Helmholtz equation. In (17) it was chosen to stack the components of the viscous velocity such that

$$\mathbf{v}_v = \left[\mathbf{v}_{v,x}\ \mathbf{v}_{v,y}\ \mathbf{v}_{v,z}\right]^\top, \quad \partial_{\mathbf{n}} \mathbf{v}_v = \left[\partial_{\mathbf{n}} \mathbf{v}_{v,x}\ \partial_{\mathbf{n}} \mathbf{v}_{v,y}\ \partial_{\mathbf{n}} \mathbf{v}_{v,z}\right]^\top, \tag{18}$$

meaning that

$$\mathbf{H}_v = \text{blkdiag}\left(\widetilde{\mathbf{H}_v}, \widetilde{\mathbf{H}_v}, \widetilde{\mathbf{H}_v}\right), \quad \mathbf{G}_v = \text{blkdiag}\left(\widetilde{\mathbf{G}_v}, \widetilde{\mathbf{G}_v}, \widetilde{\mathbf{G}_v}\right), \tag{19}$$

where blkdiag represent the block-diagonal operator while $\widetilde{\mathbf{H}_v}$ and $\widetilde{\mathbf{G}_v}$ is the discrete BEM equations for a scalar Helmholtz equation with wavenumber k_v, i.e., they are computed similarly to $\mathbf{H}_a$ and $\mathbf{G}_a$ but with wavenumber k_v.

2.4. *The Lossy System of Equations*

The values at the collocation points for each of the three modes can be computed by solving the large system of equations originating from the combination of the three discretized integral equations, the null-divergence constraint, and the boundary conditions as seen below

$$\begin{bmatrix} \text{The three discrete modes as described in (15)-(17)} \\ \text{Null Divergence Constraint} \\ \text{Isothermal Boundary Condition} \\ \text{No-Slip Boundary Condition} \end{bmatrix} \begin{bmatrix} \mathbf{p}_a \\ \partial_{\mathbf{n}} \mathbf{p}_a \\ \mathbf{p}_h \\ \partial_{\mathbf{n}} \mathbf{p}_h \\ \mathbf{v}_v \\ \partial_{\mathbf{n}} \mathbf{v}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{v}_b \end{bmatrix}, \tag{20}$$

where

$$\mathbf{v}_b = \left[\mathbf{v}_{b,x}\ \mathbf{v}_{b,y}\ \mathbf{v}_{b,z}\right]^\top, \tag{21}$$

is the prescribed boundary velocity at each collocation point stacked componentwise. While (20) is ill-conditioned, it turns out that the acoustical pressure, $\mathbf{p}_a$ can be found by solving the following well-conditioned linear system[12]

$$\left[\mathbf{G}_a \left(\mu_a \left(\mathbf{RN}\right)^{-1} \mathbf{RD}_c + \mu_h \mathbf{G}_h^{-1} \mathbf{H}_h\right) - \phi_a \mathbf{H}_a\right] \mathbf{p}_a = \mathbf{G}_a \left(\mathbf{RN}\right)^{-1} \mathbf{R} \mathbf{v}_b, \tag{22}$$

where

$$\mathbf{R} = \mathbf{D}_r - \mathbf{N}^\top \mathbf{G}_v^{-1} \mathbf{H}_v, \quad \mu_a = \phi_a - \frac{\tau_a \phi_h}{\tau_h}, \quad \mu_h = \frac{\tau_a \phi_h}{\tau_h}, \tag{23}$$

with

$$\mathbf{N} = \left[\mathrm{diag}(\mathbf{n}_x)\ \mathrm{diag}(\mathbf{n}_y)\ \mathrm{diag}(\mathbf{n}_z)\right]^\top,\ \mathbf{D}_c = \left[\mathbf{D}_x^\top\ \mathbf{D}_y^\top\ \mathbf{D}_z^\top\right]^\top,\ \mathbf{D}_r = \left[\mathbf{D}_x\ \mathbf{D}_y\ \mathbf{D}_z\right].$$

where in the above, $\mathbf{n}_x$ refers to a vector that collects all the x-components of the normals at the collocation points and similarly for $\mathbf{n}_y$ and $\mathbf{n}_z$. Furthermore, $\mathbf{D}_x$, $\mathbf{D}_y$, and $\mathbf{D}_z$ are the interpolation function derivatives (IFDs) matrices used to compute the gradient of p_a and p_h as well as the divergence of $\mathbf{v}_v$. See Appendix A for a derivation of the IFDs. A derivation of the IFDs can be found in Appendix A. As described in Preuss *et. al* [12] (22) can be solved using nested iterative solution schemes as described in Algorithm 1. In Preuss *et. al* [12] it was found, unsurprisingly, that the acoustical matrices, $\mathbf{G}_a$ and $\mathbf{H}_a$ limited the scalability of the model due to the dense nature of the two matrices. However, given that they are both standard BEM matrices they can be approximated using acceleration techniques such as the fast multipole method and Hierarchical/$\mathcal{H}$-matrices. The following section explains the basics of how these acceleration techniques can be applied in the context of boundary element matrices.

Algorithm 1 Efficient solution scheme for (22).

Require: $\mathbf{G}_a, \mathbf{H}_a, \mathbf{G}_h, \mathbf{H}_h, \mathbf{G}_v, \mathbf{H}_v, \mathbf{D}_r, \mathbf{D}_c, \mathbf{N}, \mu_a, \mu_h, \phi_a$

procedure MUL_R($\mathbf{z}$)

return $\mathbf{D}_r\mathbf{z} - \mathbf{N}^\top \mathrm{gmres}(\mathbf{G}_v, \mathbf{H}_v\mathbf{z})$

end procedure

procedure MUL_RN($\mathbf{z}$)

return MUL_R($\mathbf{N}\mathbf{z}$)

end procedure

procedure MUL_A($\mathbf{z}$)

return $\mathbf{G}_a\left(\mu_a \mathrm{gmres}(\mathrm{MUL\_RN}, \mathrm{MUL\_R}(\mathbf{D}_c\mathbf{z})) + \mu_h \mathrm{gmres}(\mathbf{G}_h, \mathbf{H}_v\mathbf{z})\right) - \phi_a \mathbf{H}_a\mathbf{z}$

end procedure

Compute right-hand side: $\mathbf{b} = \mathbf{G}_a \mathrm{gmres}(\mathrm{MUL\_RN}, \mathrm{MUL\_R}(\mathbf{v}_b))$

Solve the linear system: gmres(MUL_A,$\mathbf{b}$)

3. Acceleration Techniques

The fast multipole method and $\mathcal{H}$-matrices can be used to accelerate the complexity of the boundary element method from $O(n^2)$ to $O(n\log(n))$ or even $O(n)$ in some cases. To achieve this improvement, the matrix-vector products of $\mathbf{G}_a$ and $\mathbf{H}_a$ must be approximated. Matrix-vector products are crucial, as it is the only operation required to solve a linear system of equations using an iterative scheme such as, e.g. the generalized minimal residual method (GMRES). A brief explanation of how the two aforementioned approximation techniques

can be applied to BEM computations is now given. To simplify the derivation, we apply the same Q quadrature points to all elements. This would cause numerical problems that, however, can be fixed by applying a so-called near-field correction matrix (see (26)). We start by showing the matrix-vector product of $\mathbf{G}_a$ with a vector $\mathbf{z}$. For simplicity, we focus on the k^{th} row of $\mathbf{G}_a$

$$\begin{aligned}\left(\underbrace{\int_\Gamma G(k_a,\mathbf{x}_k,\mathbf{y})\mathbf{T}(\mathbf{y})\,\mathrm{d}S_\mathbf{y}}_{k\text{th row of }\mathbf{G}_a}\right)\mathbf{z} &\approx \left(\sum_{e=1}^{N}\left(\sum_{i=1}^{Q} G(k_a,\mathbf{x}_k,\mathbf{y}^e(\mathbf{u}_i))w_i\mathcal{J}(\mathbf{u}_i)\mathbf{T}^e(\mathbf{u}_i)\right)\mathbf{L}^e\right)\mathbf{z}\\ &= \left(\sum_{j=1}^{NQ} G(k_a,\mathbf{x}_k,\mathbf{y}_j)\underbrace{w_j\mathcal{J}(\mathbf{u}_j)\mathbf{T}^{e(j)}(\mathbf{u}_j)\mathbf{L}^{e(j)}}_{j\text{th row of }\mathbf{C}}\right)\mathbf{z}\\ &= \left[G(k_a,\mathbf{x}_k,\mathbf{y}_1)\; G(k_a,\mathbf{x}_k,\mathbf{y}_2,)\;\ldots\; G(k_a,\mathbf{x}_k,\mathbf{y}_{NQ})\right]\mathbf{C}\mathbf{z},\end{aligned} \tag{24}$$

where the subscript j refers to an ordering of the collection of Gaussian points from all elements and $e(j)$ is a function that returns the element number that Gaussian point j is located on. Furthermore, the matrix $\mathbf{C}$ is sparse and can be thought of as a transformation/interpolation of values ($\mathbf{z}$) at the collocation points ($\mathbf{x}_k$) into coefficients ($\mathbf{c}$) at the Gaussian points ($\mathbf{y}_j$). Note that $\mathbf{C}$ is independent of the chosen row, meaning that it only needs to be computed once. The derivation for the multiplication with the k^{th} row of $\mathbf{F}_a$ follows similarly

$$\begin{aligned}\left(\underbrace{\int_\Gamma \frac{\partial G(k_a,\mathbf{x}_k,\mathbf{y})}{\partial\mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\,\mathrm{d}S_\mathbf{y}}_{k\text{th row of }\mathbf{F}_a}\right)\mathbf{z} &\approx \left(\sum_{e=1}^{N}\left(\sum_{i=1}^{Q} \frac{\partial G(k_a,\mathbf{x}_k,\mathbf{y}^e(\mathbf{u}_i))}{\partial\mathbf{n}(\mathbf{y}^e(\mathbf{u}_i))}w_i\mathcal{J}(\mathbf{u}_i)\mathbf{T}^e(\mathbf{u}_i)\right)\mathbf{L}^e\right)\mathbf{z}\\ &= \left(\sum_{j=1}^{NQ} \mathbf{n}(\mathbf{y}_j)^\top\nabla G(k_a,\mathbf{x}_k,\mathbf{y}_j)\underbrace{w_j\mathcal{J}(\mathbf{u}_j)\mathbf{T}^{e(j)}(\mathbf{u}_j)\mathbf{L}^{e(j)}}_{j\text{th row of }\mathbf{C}}\right)\mathbf{z}\\ &= \left[\mathbf{n}(\mathbf{y}_1)^\top\nabla G(k_a,\mathbf{x}_k,\mathbf{y}_1)\;\ldots\;\mathbf{n}(\mathbf{y}_{NQ})^\top\nabla G(k_a,\mathbf{x}_k,\mathbf{y}_{NQ})\right]\mathbf{C}\mathbf{z},\end{aligned} \tag{25}$$

where $\mathbf{C}$ is the same as for $\mathbf{G}_a$ as it still representation the transformation/interpolation of values from collocation points to Gaussian points.

Applying (24) or (25) directly to all rows of $\mathbf{G}_a$ and $\mathbf{H}_a$ will result in a matrix-vector product that scale $O(NQn)$ for which no performance improvement is achieved given that $NQ \gtrapprox n$. However, the fast multipole method or the $\mathcal{H}$-matrix approach can be used to speed up the dense part of the matrix-vector product. However, doing so blindly will cause some numerical errors stemming from the integration of elements close to the collocation point. To solve this issue, a so-called near-field correction matrix is added to the matrix-vector product. As such, applying (24) and (25) for all rows at the same time can be thought of as

$$\mathbf{A} = \mathbf{B}\mathbf{C} + \mathbf{S}, \quad \mathbf{S},\mathbf{A} \in \mathbb{C}^{n\times n},\ \mathbf{B} \in \mathbb{C}^{n\times NQ},\ \mathbf{C} \in \mathbb{R}^{NQ\times n}, \tag{26}$$

where $\mathbf{B}$ is the dense part approximated by e.g. the fast multipole method or a $\mathcal{H}$-matrix (see (28) and (29)), $\mathbf{C}$ is the mapping from values ($\mathbf{z}$) at the collocation points ($\mathbf{x}_k$) to coefficients ($\mathbf{c}$) at the Gaussian points ($\mathbf{y}$), and $\mathbf{S}$ is the sparse near-field correction matrix. The near-field correction matrix is assembled such that it subtracts the incorrect integration made by using only Q Gaussian points in the near-field while adding the correct near-field integration. Note that similar to $\mathbf{C}$ the matrix $\mathbf{S}$ is highly sparse, which means that their assembly and matrix-vector products both scales $\mathcal{O}(n\tau)$ where $\tau \ll n$. As such, using an approximate scheme for $\mathbf{B}$ will result in a representation of $\mathbf{A}$ that is useful for large-scale computations.

3.1. *The Fast Multipole Method*

Throughout the years, many good resources have been written explaining the intricacies of fast multipole methods.[13,18] While the specifics for the different implementations might vary, the ideas are the same, namely that fast multipole methods can be used to accelerate summations. In the context of the acoustical boundary element method, the sums of interests have the form

$$u(\mathbf{x}_k) = \sum_{j=1}^{NQ} G(k_a, \mathbf{x}_k, \mathbf{y}_j) c_j, \quad u(\mathbf{x}_k) = \sum_{j=1}^{NQ} \mathbf{n}(\mathbf{y}_j)^\top \nabla G(k_a, \mathbf{x}_k, \mathbf{y}_j) c_j \tag{27}$$

where if $\mathbf{x}_k = \mathbf{y}_j$ the j^{th} term is excluded from the sums. These sums are of interest, as they exactly represent the multiplication of the k^{th} row of respectively $\mathbf{G}_a$ and $\mathbf{F}_a$ as described in (24) and (25). In both cases, the coefficients are simply computed from the sparse matrix-vector product $\mathbf{c} = \mathbf{C}\mathbf{z} = \begin{bmatrix} c_1 & c_2 & \dots & c_{NQ} \end{bmatrix}^\top$. As such, the fast multipole method can be used to accelerate the multiplication with $\mathbf{G}_a$ or $\mathbf{F}_a$ by accelerating the product with each of their rows.

3.2. *Hierarchical Matrices*

The $\mathcal{H}$-matrix approach differs from the fast multipole method approach in the sense that it does not accelerate the multiplication with each row directly. Instead, it approximates the full dense matrices

$$\mathbf{B} = \begin{bmatrix} G(k_a, \mathbf{x}_1, \mathbf{y}_1) & G(k_a, \mathbf{x}_1, \mathbf{y}_2) & \dots & G(k_a, \mathbf{x}_1, \mathbf{y}_{NQ}) \\ G(k_a, \mathbf{x}_2, \mathbf{y}_1) & G(k_a, \mathbf{x}_2, \mathbf{y}_2) & \dots & G(k_a, \mathbf{x}_2, \mathbf{y}_{NQ}) \\ \vdots & \vdots & \ddots & \vdots \\ G(k_a, \mathbf{x}_n, \mathbf{y}_1) & G(k_a, \mathbf{x}_n, \mathbf{y}_2) & \dots & G(k_a, \mathbf{x}_n, \mathbf{y}_{NQ}) \end{bmatrix}, \tag{28}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{n}(\mathbf{y}_1)^\top \nabla G(k_a, \mathbf{x}_1, \mathbf{y}_1) & \mathbf{n}(\mathbf{y}_2)^\top \nabla G(k_a, \mathbf{x}_1, \mathbf{y}_2) & \dots & \mathbf{n}(\mathbf{y}_{NQ})^\top \nabla G(k_a, \mathbf{x}_1, \mathbf{y}_{NQ}) \\ \mathbf{n}(\mathbf{y}_1)^\top \nabla G(k_a, \mathbf{x}_2, \mathbf{y}_1) & \mathbf{n}(\mathbf{y}_2)^\top \nabla G(k_a, \mathbf{x}_2, \mathbf{y}_2) & \dots & \mathbf{n}(\mathbf{y}_{NQ})^\top \nabla G(k_a, \mathbf{x}_2, \mathbf{y}_{NQ}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{n}(\mathbf{y}_1)^\top \nabla G(k_a, \mathbf{x}_n, \mathbf{y}_1) & \mathbf{n}(\mathbf{y}_2)^\top \nabla G(k_a, \mathbf{x}_n, \mathbf{y}_2) & \dots & \mathbf{n}(\mathbf{y}_{NQ})^\top \nabla G(k_a, \mathbf{x}_n, \mathbf{y}_{NQ}) \end{bmatrix}, \tag{29}$$

by utilizing the hierarchical rank-structure of subblocks of the matrices.[15,16,19] There exist many ways for which the rank-structure is both computed and utilized. However, the idea of approximating the full matrix at the same time is universal for all $\mathcal{H}$-matrix-based methods.

3.3. *Sparsity Patterns*

Many of the matrices in the final linear system described in (22) are sparse, for example, the boundary element matrices for discrete thermal and viscous modes and the interpolation function derivative matrices. In general, efficient construction of sparse matrices requires the need to avoid slow memory allocations. As such, it is important to include a priori knowledge of the sparsity patterns. For the interpolation function derivative matrices, this information is straightforward, as the interpolation functions for each collocation point have compact support equal to the elements connected to the collocation point. As an example, Figure 1 draw a collocation point in red and the support of its interpolation function as the shaded red area. In practice the sparsity patterns of the thermal and viscous matrices will be similar, but for smaller geometries the support might need to be increased slightly. As such, it may be necessary to increase the support to include the areas shown in blue in Figure 1. The sparsity pattern will generally be similar to the one for the singularity extractions applied when using the acceleration methods, and therefore the underlying code can be reused in many cases. The consequence of the sparsity patterns being the result of contributions from only local elements means that the memory scaling for all sparse matrices used must be $\mathcal{O}(\gamma n)$ where $\gamma \ll n$.

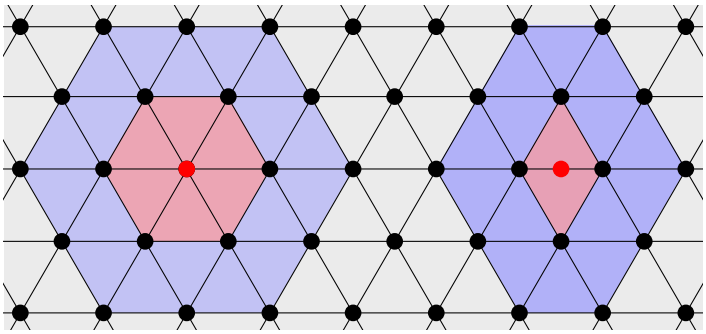


Fig. 1: The red point denotes the collocation points. The area shaded in red is the support of the IFDs while the blue area can be included in the assembly of the discrete form of the thermal and viscous modes.

4. Results

All simulations were performed using a reimplementation of the OpenBEM software written in the Julia programming language and carried out on the High-Performance Computers (HPC) provided by the DTU Computing Center (DCC).[20–22] The cluster used was using the AMD EPYC™ 7543 processor with 64 threads and was equipped with 1008 GB of

RAM. All meshes were generated using COMSOL multiphysics® v. 6.0 and were chosen to include only six-node quadratic triangular elements.[23] For fast multipole acceleration, the Flatiron Institute Fast Multipole Libraries v. 1.0.1 with a precision set to 10^{-6} was used.[24] Acceleration using the $\mathcal{H}$-matrix approach was done using HMatrices.jl v. 0.1.3, with a relative tolerance set at 10^{-6}.[25]

4.1. *Oscillating Sphere*

A widely used test case in the literature is that of a sphere in a viscous fluid oscillating along the z -axis. This test case has previously been used to validate the Kirchhoff decomposition of axisymmetrical, 2D, and 3D formulations.[9,10,26] The main reason for its wide adaptation is that the exterior test case offers an analytical solution for which the numerical results can be validated.[27] In the following we use spheres of radius 1 m and set the vertical a velocity equal to $0.01\,\mathrm{m\,s^{-1}}$. The simulation is run for nine different mesh sizes ranging from around 10^4 to 10^6 degrees of freedom (Figure 2).

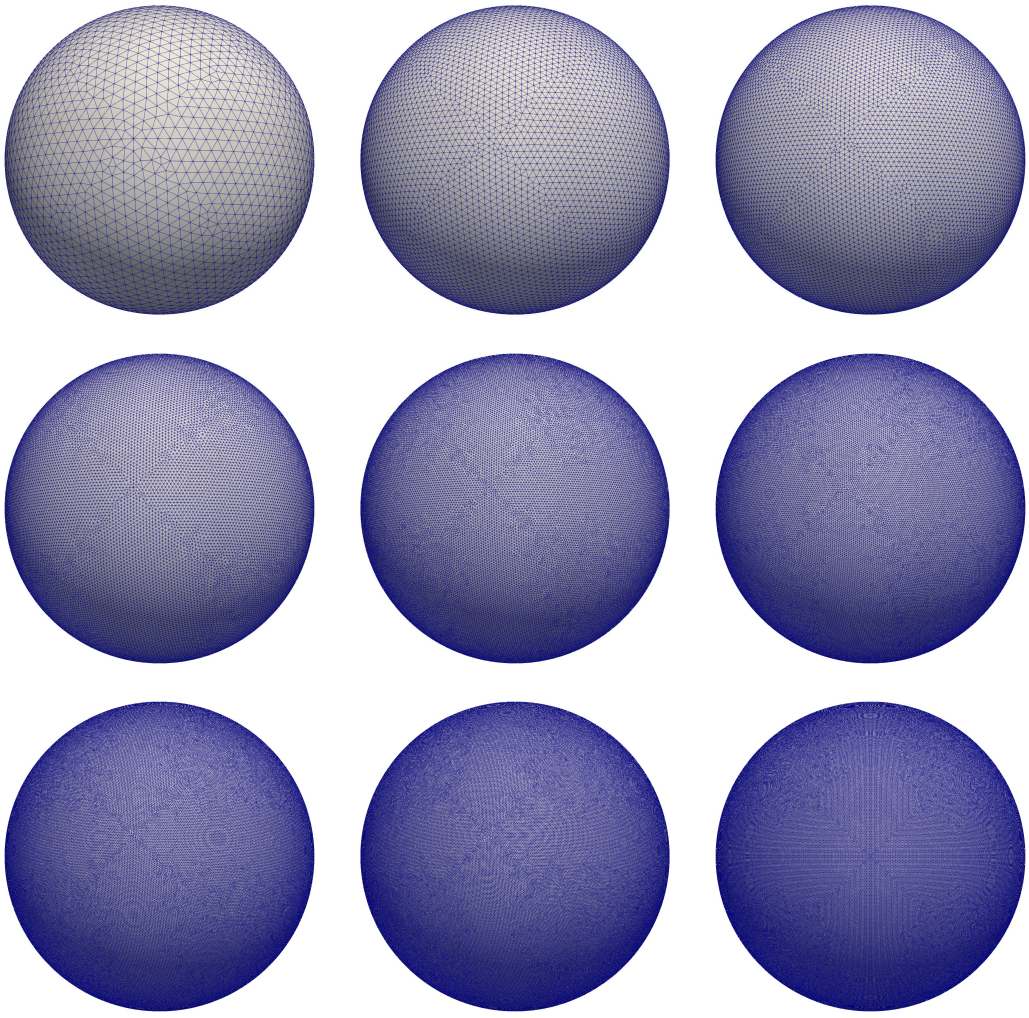


Fig. 2: The nine boundary meshes used in the simulation. From top left to bottom right the incremental refinements are as follows: 13.062, 39.486, 69.038, 153.978, 271.778, 390.890, 610.058, 753.778 and 955.706 DOFs.

The memory and computational complexity of solving (22) at 100 Hz using the acceleration techniques for approximating the discrete acoustical mode follows the theoretical scaling of said acceleration techniques (Figure 3). As such, both acceleration techniques seem to solve the computational problems described in Preuss *et. al* .[12] In particular, the memory complexity is seen to be a substantial improvement, when compared to the $O(n^2)$ scaling of storing $\mathbf{H}_a$ and $\mathbf{G}_a$ as dense matrices. A key observation when looking at Figure 3 is that while the $\mathcal{H}$-matrix approach uses close to an order of magnitude more memory than the fast multipole method (left plot in Figure 3) it decreases the overall computational effort by

close to an order of magnitude (right plot in Figure 3). Therefore, the conclusion must be that if one has the available memory, then the $\mathcal{H}$-matrix approach is preferred; however, in cases where the memory is a constraint, the fast multipole method can still be used to solve the problem at hand.

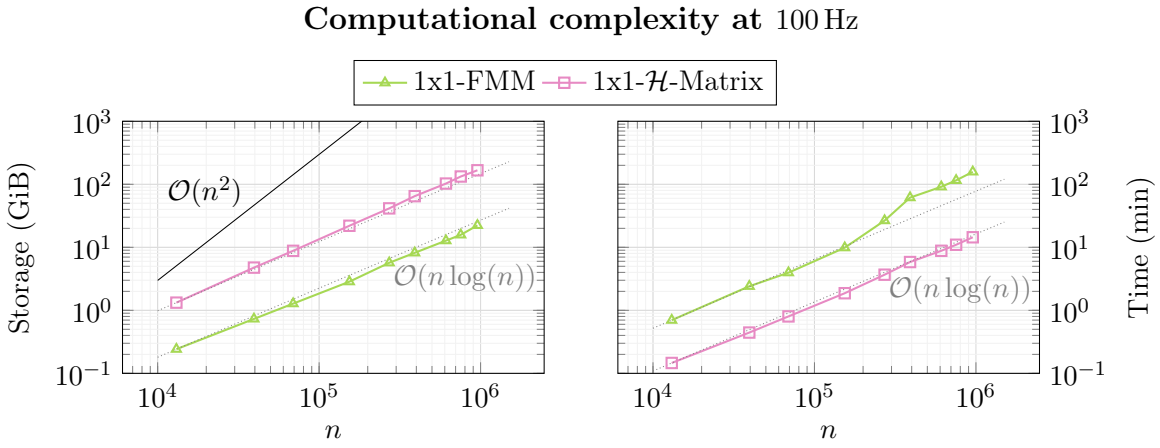


Fig. 3: Left: Complexity of storing all matrices plus performing a single call of `MUL_A` in Algorithm 1. The reason for this addition is that the acceleration techniques include intermediate steps when performing multiplications. Right: Complexity of the total solution time. This includes the assembly of the matrices and the reconstruction of the additional variables ($\partial_{\mathbf{n}}\mathbf{p}_a$, $\mathbf{p}_h$, $\partial_{\mathbf{n}}\mathbf{p}_h$, $\mathbf{v}_v$, & $\partial_{\mathbf{n}}\mathbf{v}_v$).

5. Discussion and Conclusions

A solution to the scalability issues of a recent new formulation for three-dimensional boundary element method for time-harmonic acoustics including viscothermal losses is presented in this paper. The solution utilizes known acceleration techniques for boundary element simulations, namely the fast multipole method and $\mathcal{H}$-matrices. As such, the model is shown to be capable of solving problems of sizes far beyond what was previously possible. In fact, due to most of the time being used in the reconstruction of the normal derivative of the acoustical pressure, this indicates that it is not that much of an extra effort to include the viscous and thermal losses into the computation.

The difference between the fast multipole method and the $\mathcal{H}$-matrix approach is close to similar; however, a significant difference in total memory usage and computational time can be seen. In short, if memory is a constraint, the fast multipole method is favorable and, if not, then the $\mathcal{H}$-matrix approach is favorable. However, this should not come as a shock, as the larger memory footprint of the $\mathcal{H}$-matrix is due to more intermediate information being saved, resulting in less computation on the fly when compared to the fast multipole method.

A possible middle ground between the fast multipole method and the $\mathcal{H}$-matrix, like the interpolated factored Green's function (IFGF), could be the best approach.[28,29] A downside of this method is that it requires the underlying matrix to be generated by a continuous function of $\mathbf{x}$ and $\mathbf{y}$, which is a problem for the $\mathbf{H}_a$ matrix due to its dependence on the normals that are computed only from the local coordinates. The workaround is to apply the IFGF for each of the components of the gradient of the Green's function resulting in three separate matrix approximations, which increases the model complexity substantially.

Acknowledgments

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of research grant no. 468269586 and partly funded by DTU through a joint alliance research project. Additionally, the authors are grateful for the computing resources provided by the DTU Computing Center.

Reference List

References

1. V. Cutanda Henríquez, P. R. Andersen, J. S. Jensen, P. M. Juhl and J. Sánchez-Dehesa, A Numerical Model of an Acoustic Metamaterial Using the Boundary Element Method Including Viscous and Thermal Losses, *Journal of Computational Acoustics* **25** (2017) 1750006.
2. R. D. Cook and R. D. Cook, *Concepts and Applications of Finite Element Analysis* (Wiley, 2002).
3. F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*, volume 132 of *Applied Mathematical Sciences* (Springer Verlag, 1998).
4. W. R. Kampinga, Y. H. Wijnant and A. de Boer, A finite element for viscothermal wave propagation, *23rd International Conference on Noise and Vibration Engineering 2008, Isma 2008* **7** (2008) 4271–4278.
5. N. Atalla and F. Sgard, *Finite Element and Boundary Methods in Structural Acoustics and Vibration* (CRC Press, 2015).
6. S. Marburg, Boundary Element Method for Time–Harmonic Acoustic Problems, in *Computational Acoustics*, ed. M. Kaltenbacher (Springer, 2017), CISM International Centre for Mechanical Sciences, ch. 3, pp. 69–159.
7. S. Preuss, C. Gurbuz, C. Jelich, S. K. Baydoun and S. Marburg, Recent Advances in Acoustic Boundary Element Methods, *Journal of Theoretical and Computational Acoustics* **30** (2022) 2240002.
8. M. Bruneau, P. Herzog, J. Kergomard and J. D. Polack, General formulation of the dispersion equation in bounded visco-thermal fluid, and application to some simple geometries, *Wave Motion* **11** (1989) 441–451.
9. V. Cutanda Henríquez and P. M. Juhl, An axisymmetric boundary element formulation of sound wave propagation in fluids including viscous and thermal losses, *The Journal of the Acoustical Society of America* **134** (2013) 3409–3418.
10. V. Cutanda Henríquez and P. R. Andersen, A Three-Dimensional Acoustic Boundary Element Method Formulation with Viscous and Thermal Losses Based on Shape Function Derivatives, *Journal of Theoretical and Computational Acoustics* **26** (2018) 1850039.

11. P. R. Andersen, V. Cutanda Henríquez and N. Aage, Shape optimization of micro-acoustic devices including viscous and thermal losses, *Journal of Sound and Vibration* **447** (2019) 120–136.
12. S. Preuss, M. Paltorp, V. Cutanda Henríquez and S. Marburg, Revising the Boundary Element Method for Thermoviscous Acoustics: An Iterative Approach via Schur Complement, *Journal of Theoretical and Computational Acoustics* .
13. N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions* (Elsevier, 2005).
14. L. Yijun, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering* (Cambridge University Press, 2009).
15. W. Hackbusch, *Hierarchical Matrices: Algorithms and Analysis* (Springer, 2015).
16. M. Bebendorf, *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems; with 53 tables* (Springer, 2008).
17. S. Marburg, The Burton and Miller Method: Unlocking Another Mystery of Its Coupling Parameter, *Journal of Computational Acoustics* **24** (2016) 1550016.
18. L. Ying, A pedestrian introduction to fast multipole methods, *Science China Mathematics* **55** (2012) 1043–051.
19. J. Ballani and D. Kressner, Matrices with Hierarchical Low-Rank Structures, *Lecture Notes in Mathematics* **2173** (2016) 161–209.
20. V. C. Henríquez and P. M. Juhl, OpenBEM - an open source Boundary Element Method software in Acoustics, *Proceedings of Internoise 2010* **7** (2010) 5796–5805.
21. J. Bezanson, A. Edelman, S. Karpinski and V. B. Shah, Julia: A fresh approach to numerical computing, *SIAM review* **59** (2017) 65–98.
22. DTU Computing Center, DTU Computing Center resources, 2021.
23. COMSOL AB, Stockholm, Sweden, *Acoustics Module Users Guide, COMSOL Multiphysics v. 6.0*, 2021.
24. T. Askham, Z. Gimbutas, L. Greengard, L. Lu, J. Magland, D. Malhotra, M. O'Neil, M. Rachh and V. Rohklin, FMM3D v. 1.0.1, `https://github.com/flatironinstitute/FMM3D/releases/tag/v1.0.1`, 2023.
25. L. M. Faria, HMatrices.jl v. 0.1.3, `https://github.com/WaveProp/HMatrices.jl/releases/tag/v0.1.3`, 2023.
26. P. R. Andersen, V. Cutanda Henríquez, N. Aage and S. Marburg, A Two-Dimensional Acoustic Tangential Derivative Boundary Element Method Including Viscous and Thermal Losses, *Journal of Theoretical and Computational Acoustics* **26** (2018) 1850036.
27. S. Temkin and J. D. N. Cheeke, *Elements of Acoustics* (American Association of Physics Teachers (AAPT), 1982).
28. C. Bauinger and O. P. Bruno, Interpolated Factored Green Function method for accelerated solution of scattering problems, *Journal of Computational Physics* **430** (2021) 110095.
29. C. Bauinger and O. P. Bruno, Massively parallelized interpolated factored Green function method, *Journal of Computational Physics* **475** (2023) 111837.

Appendix

A. Interpolation Function Derivatives

The interpolation function derivatives (IFDs) are the derivatives of the boundary element interpolation functions. As the boundary element interpolation is defined only on the surface, the derivatives of the interpolation function contain only tangential information. The full gradient is then computed by combining the tangential information from the IFDs with

the normal information that comes directly from the boundary element interpolation of the normal derivative.

Taking the acoustic pressure p_a as an example, it follows from (14) that the boundary element representation of the interpolation on element e is given as

$$p_a\left(\mathbf{x}^e(\mathbf{u})\right) = p_a\left(\mathbf{X}^e\mathbf{N}^e(\mathbf{u})\right) = \mathbf{T}^e(\mathbf{u})\mathbf{p}_a^e, \quad \mathbf{u} \in \mathcal{L}^e,\ \mathbf{x} \in \Gamma_e, \tag{A.1}$$

where $\mathbf{p}_a^e$ represent the nodal acoustical pressures of element e while $\mathbf{N}(\mathbf{u})$ and $\mathbf{T}^e(\mathbf{u})$ denotes the chosen interpolation schemes for respectively the geometry and the (acoustical) pressure on element e. The resulting gradient of p_a on element e can be computed as

$$\nabla_{\mathbf{u}} p_a\left(\mathbf{x}^e(\mathbf{u})\right) = \left(\nabla_{\mathbf{u}}\mathbf{T}^e(\mathbf{u})\right)\mathbf{p}_a^e \tag{A.2}$$

since the values $\mathbf{p}_a^e$ are constants. Applying the multivariate chain rule $\nabla_{\mathbf{u}} p_a = (\nabla_{\mathbf{u}}\mathbf{x}^\top)\nabla_{\mathbf{x}} p_a$ followed by isolating with respect to $\nabla_{\mathbf{x}}$ gives the gradient of interest. Since the computations are based on boundary elements we have that the transpose of the Jacobian matrix

$$\nabla_{\mathbf{u}}\mathbf{x}^e(\mathbf{u})^\top = \nabla_{\mathbf{u}}(\mathbf{X}^e\mathbf{N}^e(\mathbf{u}))^\top = \nabla_{\mathbf{u}}\mathbf{N}^e(\mathbf{u})^\top\left(\mathbf{X}^e\right)^\top, \tag{A.3}$$

is a 2×3-matrix which cannot be inverted. It has been shown that it is enough to introduce an artificial u_3 such that $\frac{\partial \mathbf{T}^e}{\partial u_3} = \mathbf{0}$ (with $\mathbf{0}$ being a row vector of appropriate length filled with zeros), while substituting $\frac{\partial \mathbf{x}}{\partial u_3} = \frac{\partial \mathbf{x}}{\partial u_1} \times \frac{\partial \mathbf{x}}{\partial u_2}$.[5] As a result, the tangential part of the gradient can be computed as

$$\nabla_{\mathbf{x}}^{\parallel} p_a\left(\mathbf{x}^e(\mathbf{u})\right) = \begin{bmatrix} \frac{\partial x}{\partial u_1} & \frac{\partial y}{\partial u_1} & \frac{\partial z}{\partial u_1} \\ \frac{\partial x}{\partial u_2} & \frac{\partial y}{\partial u_2} & \frac{\partial z}{\partial u_2} \\ \left(\frac{\partial \mathbf{x}}{\partial u_1} \times \frac{\partial \mathbf{x}}{\partial u_2}\right)^\top \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathbf{T}^e}{\partial u_1} \\ \frac{\partial \mathbf{T}^e}{\partial u_2} \\ \mathbf{0} \end{bmatrix} \mathbf{p}_a^e, \quad \mathbf{x} \in \Gamma_e, \tag{A.4}$$

where the superscript $\parallel$ is used to explicitly show that this is only the tangential part of the gradient. On element e there is a local coordinate $\mathbf{u}$ for which $\mathbf{x}^e(\mathbf{u})$ is equal to a collocation point. If this is the i^{th} collocation point, then this means that there is a $\mathbf{u}^{e,i}$ for which $\mathbf{x}^e\left(\mathbf{u}^{e,i}\right) = \mathbf{x}_i$. Using what was described in (14), namely that $\mathbf{p}_a^e = \mathbf{L}^e\mathbf{p}_a$, it follows that

$$\nabla_{\mathbf{x}}^{\parallel} p_a\left(\mathbf{x}^e(\mathbf{u}^{e,i})\right) = \nabla_{\mathbf{x}}^{\parallel} p_a(\mathbf{x}_i) = \begin{bmatrix} \frac{\partial x}{\partial u_1} & \frac{\partial y}{\partial u_1} & \frac{\partial z}{\partial u_1} \\ \frac{\partial x}{\partial u_2} & \frac{\partial y}{\partial u_2} & \frac{\partial z}{\partial u_2} \\ \left(\frac{\partial \mathbf{x}}{\partial u_1} \times \frac{\partial \mathbf{x}}{\partial u_2}\right)^\top \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathbf{T}^e}{\partial u_1} \\ \frac{\partial \mathbf{T}^e}{\partial u_2} \\ \mathbf{0} \end{bmatrix} \mathbf{L}^e\mathbf{p}_a = \begin{bmatrix} \mathbf{D}_x^{e,i} \\ \mathbf{D}_y^{e,i} \\ \mathbf{D}_z^{e,i} \end{bmatrix} \mathbf{p}_a. \tag{A.5}$$

In the case of discontinuous elements, the collocation point is only connected to a single element, meaning that in practice the e^{th} superscript is redundant. However, in the case of continuous elements, the collocation point can be connected to multiple elements, and the

interpolation function derivative is chosen to be the average contribution from each of the connected elements. As such, the i^{th} rows of the $\mathbf{D}_\bullet$-matrices can be computed as

$$\mathbf{D}_x^i = \frac{1}{N_e(i)} \sum_{e=1}^{N_e(i)} \mathbf{D}_x^{e,i}, \quad \mathbf{D}_y^i = \frac{1}{N_e(i)} \sum_{e=1}^{N_e(i)} \mathbf{D}_y^{e,i}, \quad \mathbf{D}_z^i = \frac{1}{N_e(i)} \sum_{e=1}^{N_e(i)} \mathbf{D}_z^{e,i}, \tag{A.6}$$

where $N_e(i)$ denotes the number of elements that is connected to collocation point i. The different $\mathbf{D}_\bullet^i$ of (A.6) are the collection of the i^{th} rows in three separate matrices, $\mathbf{D}_x, \mathbf{D}_y$ and $\mathbf{D}_z$ so that

$$\frac{\partial \mathbf{p}_a}{\partial x}^{\parallel} = \mathbf{D}_x \mathbf{p}_a, \quad \frac{\partial \mathbf{p}_a}{\partial y}^{\parallel} = \mathbf{D}_y \mathbf{p}_a, \quad \frac{\partial \mathbf{p}_a}{\partial z}^{\parallel} = \mathbf{D}_z \mathbf{p}_a. \tag{A.7}$$

If the chosen discretization of the thermal and viscous modes are the same as for the acoustical mode, then the above interpolation function derivative matrices can be reused to compute $\nabla_{\mathbf{x}}^{\parallel} p_h$ and $\nabla_{\mathbf{x}}^{\parallel} \cdot \mathbf{v}_v$ respectively.

Paper C1

(Including Errata)

ICA 2022
24th INTERNATIONAL CONGRESS ON ACOUSTICS
October 24(Mon) - 28(Fri), 2022 Gyeongju, Korea

PROCEEDINGS of the
24th International Congress on Acoustics

October 24 to 28, 2022 in Gyeongju, Korea

ABS-0839

A reduced order model including viscothermal losses

Mikkel PALTORP SCHMITT[(1)]; Vicente CUTANDA HENRÍQUEZ[(2)]; Niels AAGE[(3)]; Peter RISBY ANDERSEN[(4)]

[(1)]Department of Electrical & Photonics Engineering, Technical University of Denmark, Denmark, mpasc@dtu.dk

[(2)]Centre for Acoustic-Mechanical Microsystems, Department of Electrical & Photonics Engineering, Technical University of Denmark, Denmark, vcuhe@elektro.dtu.dk

[(3)]Centre for Acoustic-Mechanical Microsystems, Department of Civil & Mechanical Engineering, Technical University of Denmark, Denmark, naage@mek.dtu.dk

[(4)]Audio Research, GN Audio A/S & Jabra, Denmark, prandersen@jabra.com

ABSTRACT

The Boundary Element Method (BEM) is a well-known numerical method for solving time harmonic acoustical problems. While the BEM has attractive features e.g. automatically satisfying the free field conditions, the frequency dependence of the Greens function makes is inconvenient for broadband simulations due to excessive computational costs. This problem becomes worse yet when viscous and thermal effects are included in the BE computations. Recently it has been suggested that a series expansion of the Greens function as well as its directional derivatives together with model order reduction techniques can relieve some of the computational demands. This paper applies similar ideas in the setting of the so-called boundary layer impedance boundary conditions used to approximate viscous and thermal effects. The final computational model can be used to efficiently perform broadband simulations including viscothermal losses of devices on the centimeter scale, thereby paving the path towards e.g. broadband shape optimization of small acoustical devices such as transducers, metamaterials and hearing aids.

Keywords: Boundary Element Method, Viscothermal Effects, Reduced Order Model

1 INTRODUCTION

The Boundary Element Method (BEM) is a well-known simulation technique where the underlying (partial) differential equation is solved through the discretization of a boundary integral. It is a popular choice for problems concerning semi-infinite and infinite domains, as the solution computed from the BEM automatically fulfills the so-called free-field conditions. As a result the method is widespread within the fields of electromagnetics and acoustics as these commonly deal with such domains. As opposed to the Finite Element Method (FEM) the discretization of the BEM result in frequency-depended and dense linear systems. The latter problem can be alleviated by utilizing compression techniques such as the Fast Multipole Method[1] (FMM) and $\mathscr{H}$-matrices[2]. While some details of the two approaches differ they are both based on the same ideas of so-called near and far field splits of the underlying integral operators. While these approaches solves the memory issue of the BEM they still require a reevaluation of the underlying boundary integrals for every frequency of interest. As a result the BEM is cumbersome to apply when one is interested in a wide range of frequencies. Several solutions to this problem has been proposed during the years, all with different pros and cons. These techniques are almost exclusively based on interpolations of either the discrete or continuous form of the boundary integral [3, 4, 5]. In this text the interpolation scheme chosen is that of a Taylor series expansion of the (derivatives of the) Green's function. This approach transforms the time-consuming boundary integral discretization into a sum of matrices. For small orders of Taylor expansions this reduces the assembly significantly, however the total memory used to store the derivatives matrices

makes this approach close to unusable in many cases. This memory increase can be diminished by following the approach taken in Panagiotopoulos et al. [6]. Here a projection matrix is created from the collection of Krylov vectors from specified frequencies. The full system is then projected resulting in a reduced number of DOFs. This approach works under the assumption the solution is spanned by said Krylov vectors, which is true for problems for which a Krylov based iterative solver converges quickly.

The main result presented in this paper is an extension of the model proposed in Panagiotopoulos et al. [6]. This extension makes it possible for the model to also handle the so-called Boundary Layer Impedance (BLI) boundary conditions [7]. The text is structured as follows: In section 2 we describe how the discrete BEM system representing the BLI boundary condition can be formulated as a Taylor series of frequency-decoupled matrices. Next, in section 3 the model order reduction procedure is explained. Finally, in section 4 the proposed model is numerically evaluated using a simple test case.

2 BOUNDARY ELEMENT METHOD WITH BLI BOUNDARY CONDITION

In this section a rudimental description of the BEM, the Boundary Layer Impedance (BLI) boundary condition and the Series Expansion Boundary Element Method (SEBEM) is given.

2.1 The Boundary Element method

The BEM for time-harmonic acoustical problems is based on the discretization of the Kirchoff-Helmholtz representation formula for the scalar Helmholtz equation. The representation formula has the following form

$$c(\mathbf{x})p(\mathbf{x}) = \int_\Gamma G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}}\ \mathrm{d}S_\mathbf{y} - \int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}} p(\mathbf{y})\ \mathrm{d}S_\mathbf{y}, \quad \mathbf{x}\in\Omega, \tag{1}$$

where Ω is the domain with boundary Γ, $c(\mathbf{x})$ is the free coefficient depending only on the geometry at point $\mathbf{x}$ and $(\partial_n G)$ G is the (normal derivative of the) Green's function for the scalar Helmholtz equation, with the form

$$G(\mathbf{x},\mathbf{y}) = \frac{\exp\left(-\mathrm{i}k\|\mathbf{x}-\mathbf{y}\|_2\right)}{4\pi\|\mathbf{x}-\mathbf{y}\|_2}, \qquad \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}} = \frac{\exp\left(-\mathrm{i}k\|\mathbf{x}-\mathbf{y}\|_2\right)\left(1+\mathrm{i}k\|\mathbf{x}-\mathbf{y}\|_2\right)(\mathbf{x}-\mathbf{y})^\top \mathbf{n}_\mathbf{y}}{4\pi\|\mathbf{x}-\mathbf{y}\|_2^3}, \tag{2}$$

where $k = \frac{\omega}{c}$ is called the wavenumber. Due to the nonlinear coupling between the frequency-depended wavenumber and the variable of integration inside the exponential, the frequency can not be decoupled from the integration in (1). As a result the integral needs to be recomputed for every frequency of interest, making the use of BEM for large frequency sweeps inconvenient. In subsection 2.3 we describe how these can be decoupled through a Taylor series expansion.

2.2 The Boundary Layer Impedance

The Boundary Layer Impedance (BLI) boundary condition, described in Berggren et al. [7], is an impedance like boundary condition in the way that it linearly relates the pressure and its normal derivative. However, in addition to this relation there is also the addition of the tangential Laplacian

$$\frac{\partial p}{\partial \mathbf{n}} = \left[(\gamma-1)\frac{\mathrm{i}k^2}{k_h} - \frac{\mathrm{i}\Delta_\mathrm{T}}{k_v}\right] p, \tag{3}$$

where k_h and k_v are respectively the thermal and viscous wavenumbers and γ is the ratio of specific heats. The two additional wavenumbers are computed using the frequency and the physical properties of the fluid such as the thermal conductivity, specific heat capacity under constant pressure and the shear and bulk viscosity coefficients [8].

In the following the boundary of the domain is split into two parts. The first will be applied a Neumann boundary condition while the second will be applied the BLI boundary condition (Figure 1).

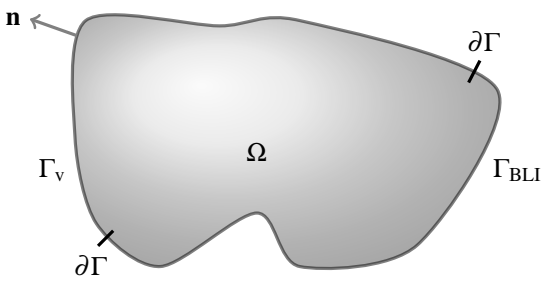


Figure 1. Domain of an interior problem. The boundary is split into two parts: One with a Neumann boundary condition (Γ_v) and one with the BLI boundary condition (Γ_BLI).

Using the boundary splitting and inserting (3) into the representation formula we attain the following

$$c(\mathbf{x})p(\mathbf{x}) = \int_{\Gamma_\mathrm{v}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}}\ \mathrm{d}S_\mathbf{y} + \int_{\Gamma_\mathrm{BLI}} G(\mathbf{x},\mathbf{y})\left[\frac{(\gamma-1)\mathrm{i}k^2}{k_h} - \frac{\mathrm{i}\Delta_\mathrm{T}}{k_v}\right] p(\mathbf{y})\ \mathrm{d}S_\mathbf{y} - \int_{\Gamma} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}} p(\mathbf{y})\ \mathrm{d}S_\mathbf{y}. \tag{4}$$

Due to the polynomial representation of the pressure on each element the tangential Laplacian restricts the elements to be at least second order. To relax this restriction we apply integration by parts on the term including the tangential Laplacian. Setting the additional term equal to 0, similar to what is described in Berggren et al. [7], while rearranging a few terms we end up with the following equation of interest

$$\begin{aligned}\int_{\Gamma_\mathrm{v}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}}\ \mathrm{d}S_\mathbf{y} = c(\mathbf{x})p(\mathbf{x}) + \int_{\Gamma} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}_\mathbf{y}} p(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ - \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\int_{\Gamma_\mathrm{BLI}} G(\mathbf{x},\mathbf{y})p(\mathbf{y})\ \mathrm{d}S_\mathbf{y} - \frac{\mathrm{i}}{k_v}\int_{\Gamma_\mathrm{BLI}} \nabla_\mathrm{T} G(\mathbf{x},\mathbf{y})\cdot\nabla_\mathrm{T} p(\mathbf{y})\ \mathrm{d}S_\mathbf{y}.\end{aligned} \tag{5}$$

In the above only the tangential gradient of the pressure is needed, meaning that elements need only be of at least linear order. Finally, applying the element discretization on (5) the following linear system of equations is computed

$$\left(\mathrm{diag}(\mathbf{c}) + \mathbf{F}(k) - \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\mathbf{H}(k) - \frac{\mathrm{i}}{k_v}\mathbf{T}(k)\right)\mathbf{p} = \mathbf{b}(k). \tag{6}$$

2.3 Series Expansion BEM

As mentioned previously a drawback of the BEM is the frequency-depended integrals. In order to resolve this issue all relevant kernels in (5) is expanded in terms of their respective Taylor series. This approach is sometimes referred to as the Series Expansion BEM (SEBEM), although other series representations of the kernels could have been utilized. The simplest of the Taylor series expansions is that of the Green's function

$$G(\mathbf{x},\mathbf{y}) = \frac{\exp\left(-\mathrm{i}k\|\mathbf{x}-\mathbf{y}\|_2\right)}{4\pi\|\mathbf{x}-\mathbf{y}\|_2} = \frac{1}{4\pi}\sum_{i=0}^{\infty}\frac{(k-k_0)^m}{m!}\frac{(-\mathrm{i}\|\mathbf{x}-\mathbf{y}\|_2)^m}{\|\mathbf{x}-\mathbf{y}\|_2}\exp\left(-\mathrm{i}k_0\|\mathbf{x}-\mathbf{y}\|_2\right), \tag{7}$$

where k_0 is the expansion wavenumber. Applying the analogous on the remaining kernels in (5) we have that the discrete form of (6) can be approximated as

$$\left(\mathrm{diag}(\mathbf{c}) + \sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\left[\mathbf{F}_m - \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\mathbf{H}_m - \frac{\mathrm{i}}{k_v}\mathbf{T}_m\right]\right)\mathbf{p} = \sum_{m=0}^{M-1}\frac{(k-k_0)^m}{m!}\mathbf{b}_m. \tag{8}$$

Note that we already here have truncated the series the Taylor series to be order M (degree $M-1$). As a result the above should strictly speaking be an approximation. While the Series Expansion Boundary Element Method (SEBEM) reduces the assembly time by reducing the integral discretization to matrix additions, the

resulting model have a large memory footprint, stemming from the need to store intermediate matrices. A first fix could be to Taylor expand the full expression in (5) instead of the kernels separately. This would result in only needing to store one derivative matrix (for the total derivative) instead of the 3 required in (8) (one for each integral). However, the implementation of this approach would be significantly more complicated while most likely being less accurate.

3 BOUNDARY ELEMENT METHOD WITH MODEL ORDER REDUCTION

In order to reduce the large memory footprint of the SEBEM a Model Order Reduction (MOR) approach can be utilized. There exist several possible MOR techniques that can be applied with the one used here following that of Panagiotopoulos et al. [6]. In short this approach works under the assumption that the true solution can be well approximated by a lower-dimensional subspace, meaning that the total size of linear system can be reduced by projection onto this subspace.

3.1 Krylov projection

In general a projection based MOR utilizes the assumption that the solution as well as the right-hand side of the linear system of equations can be well-approximated by a lower-dimensional subspace. As such the following are assumed to be good approximations

$$\mathbf{p} \approx \mathbf{U}_\ell \mathbf{p}_\ell, \quad \mathbf{b}_m \approx \mathbf{U}_\ell \mathbf{b}_{\ell m}, \qquad \mathbf{U}_\ell \in \mathbb{C}^{N\times \ell}, \tag{9}$$

where the columns of $\mathbf{U}_\ell$ forms a basis for the lower-dimensional subspace. The Krylov projection technique is based on setting these columns equal to the first ℓ Krylov vectors of the system matrix (subsection 3.2). In order to reduce the system further, one can introduce a projection of the rows. If the projection of the rows is equal to the Hermitian transpose of the column projection, then the approach is called Galerkin. This is the approach taken here. As a result (8) can be approximated as

$$\mathbf{U}_\ell^{\mathsf{H}} \left(\operatorname{diag}(\mathbf{c}) + \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \left[\mathbf{F}_m - \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\mathbf{H}_m - \frac{\mathrm{i}}{k_v}\mathbf{T}_m \right] \right) \mathbf{U}_\ell \mathbf{p}_\ell = \mathbf{U}_\ell^{\mathsf{H}} \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \mathbf{U}_\ell \mathbf{b}_{\ell m}. \tag{10}$$

Moving $\mathbf{U}_\ell$ and $\mathbf{U}_\ell^{\mathsf{H}}$ inside the parentheses and using that $\mathbf{U}_\ell$ is unitary, the above can be written as

$$\left(\mathbf{C}_\ell + \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \left[\mathbf{F}_{\ell m} - \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\mathbf{H}_{\ell m} - \frac{\mathrm{i}}{k_v}\mathbf{T}_{\ell m} \right] \right) \mathbf{p}_\ell = \sum_{m=0}^{M-1} \frac{(k-k_0)^m}{m!} \mathbf{b}_{\ell m}, \tag{11}$$

where all matrices are in $\mathbb{C}^{\ell\times\ell}$ and vectors in $\mathbb{C}^\ell$. Given that $\ell \ll N$ the above is a very efficient way of storing the information. In many practical cases one find that storing the full reduced SEBEM requires less memory than storing just a single of the original $N \times N$ matrices.

3.2 Creating a projection matrix

Computing a projection matrix $\mathbf{U}_\ell$ using just the information available at a single frequency does not result in a good approximation on the full frequency range (Figure 3). To resolve this issue information from multiple frequencies is utilized. The collection of frequencies for which this information is extracted will be referred to as *primary frequencies* (denoted by f_i). The total number of primary frequencies is denoted by L. At each primary frequency the full BEM system $(\mathbf{A}_{f_i}, \mathbf{b}_{f_i})$ is computed and the first q Krylov vectors computed using the Arnoldi Algorithm (Algorithm 1).

Algorithm 1 Arnoldi Algorithm

Assemble $\mathbf{A}_{f_i}$, $\mathbf{b}_{f_i}$
$\mathbf{k}_1 \leftarrow \mathbf{b}_{f_i}/\|\mathbf{b}_{f_i}\|_2$
$\mathbf{K}_{f_i} \leftarrow \begin{bmatrix}\mathbf{k}_1\end{bmatrix}$
for $j = 2 : q$ **do**
  $\mathbf{k}_j \leftarrow \mathbf{A}_{f_i}\mathbf{k}_{j-1} - \sum_{l=1}^{j-1}\left(\mathbf{k}_l^{\mathsf{H}}\mathbf{A}_{f_i}\mathbf{k}_{j-1}\right)\mathbf{k}_l$
  $\mathbf{k}_j \leftarrow \mathbf{k}_j/\|\mathbf{k}_j\|_2$
  $\mathbf{K}_{f_i} \leftarrow \begin{bmatrix}\mathbf{K}_{f_i} & \mathbf{k}_j\end{bmatrix}$
end for

The idea is that by collecting all $\ell = qL$ Krylov vectors will result in a projection matrix that covers the full frequency range of interest. However, doing this without any additional computation will result in a projection matrix that is not necessarily unitary, which was a requirement in order to arrive at (11). In order to arrive at a unitary projection matrix the Singular Value Decomposition (SVD) is applied to the collection of the Krylov subspaces

$$\mathbf{U}_\ell \Sigma_\ell \mathbf{V}_\ell^{\mathsf{H}} = \mathrm{svd}\left(\begin{bmatrix}\mathbf{K}_{f_1} & \mathbf{K}_{f_2} & \dots & \mathbf{K}_{f_L}\end{bmatrix}\right), \tag{12}$$

where $\mathbf{U}_\ell$ is the final projection matrix.

4 NUMERICAL EVALUATION

In this section the accuracy of the Reduced Order Series Expansion Boundary Element Method (ROSEBEM) for the BLI boundary conditions is numerically evaluated. All simulation were performed using the Julia programming language [9, 10] on computational resources provided by the DTU Computing Center [11]. The core of the code was inspired by the OpenBEM software written in MATLAB [12, 13].

4.1 Simulation setups

In this section we numerically study a small tube of length 10 cm and radius 1 cm (Figure 2). The used mesh contained 1916 quadratic elements resulting in a total of 3834 DOFs. The frequency of interest is $[20\,\mathrm{Hz},\ 10\,\mathrm{kHz}]$, meaning that the mesh with $h_{\max} \approx 0.033\,\mathrm{cm}$ had at minimum 10 elements per wavelength, satisfying the standard rule-of-thumb of [14]. The object is excited by applying a Neumann boundary condition at one of the tube ends (Figure 2) while the remaining part of the boundary were applied the BLI boundary condition. In order to evaluate the response the acoustic pressure were evaluated at a point on the opposite side of the Neumann condition.

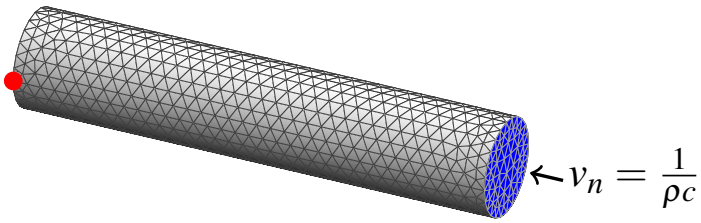


Figure 2. Cylinder with length 10 cm and radius 1 cm. The mesh has 1916 quadratic elements resulting in 3834 DOFs. The pressure is evaluated at the point shown in red, located at $\mathbf{x} = (0\,\mathrm{cm},\ -1\,\mathrm{cm},\ 0\,\mathrm{cm})$.

Two different ROSEBEM with expansion frequency $f_0 = 5\,\mathrm{kHz}$ where evaluated. The first used only the information available at the expansion frequency when creating the projection matrix whereas the second used two additional primary frequencies, meaning that $f_i \in \{1.5\,\mathrm{kHz},\ 5\,\mathrm{kHz},\ 8.5\,\mathrm{kHz}\}$. In both cases the total size of the projection space was 450, resulting in a total reduction of DOFs of approximately 88%.

4.2 Frequency response

The acoustic response is evaluated at increments of 1 Hz in the interval [20 Hz, 10 kHz] resulting in a total of 9981 evaluations. Around the expansion frequency the two ROSEBEM setups are indistinguishable from the full BEM computations (Figure 3). However, at the two ends of the frequency range of interest the setup that only uses the information from the expansion frequency drifts away from the true solution (Figure 3). The pointwise relative error is significantly lower over the full domain when using three primary frequencies as compared to only one (Figure 3), albeit both setups experience large relative errors close to resonances.

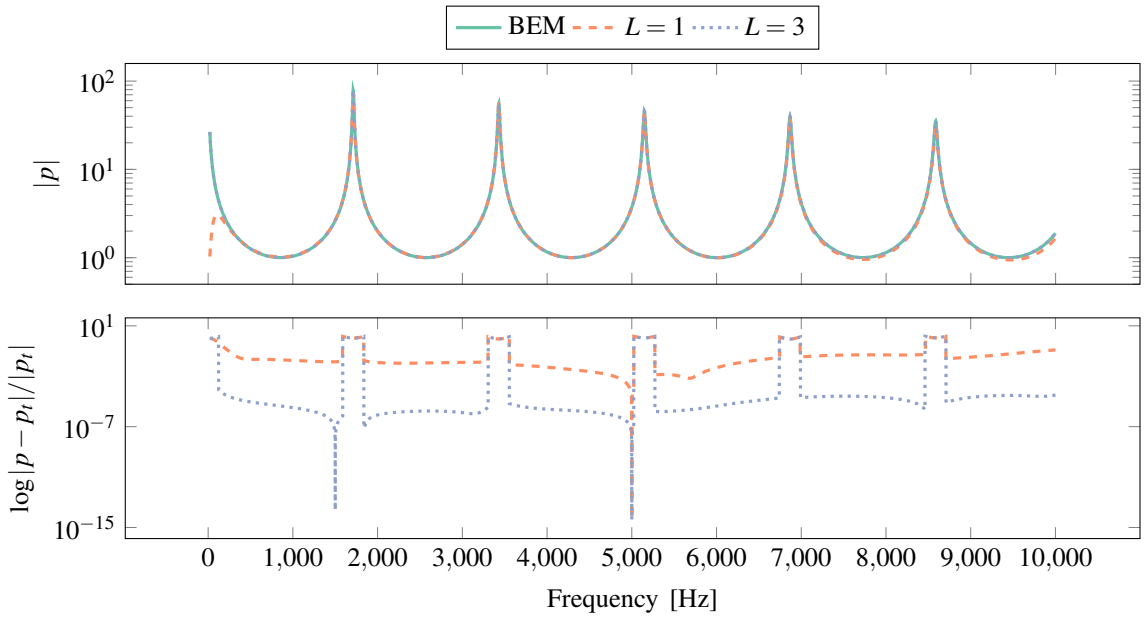


Figure 3. Top: Absolute value of the pressure at point $\mathbf{x}$ as a function of frequency. Bottom: Relative error of the ROSEBEM setups compared to the full BEM solution.

Comparing the computational efforts of the model can easily lead to skewed results as the efficiency greatly depend on the problem at hand. One of the more prominent of these dependencies is the number of frequencies evaluated. The more frequencies evaluated the better the ROSEBEM will perform compared to the full model. It is therefore better to think in terms of the computational and memory scaling of the various components that goes into the ROSEBEM. A good summary of these can be found in Table 1 in Panagiotopoulos et al. [6]. A general rule-of-thumb is that the ROSEBEM should only be utilized if the number of frequencies evaluated is larger than the order of the Taylor expansion.

For our specific setup there were almost 10,000 frequencies evaluated resulting in the ROSEBEM being around 106 times faster than the full solution (Table 1). In practical terms this meant that the total computational time went from close to 5 days to a little over an hour. Comparing the total memory, it was seen that the ROSEBEM used 370.789 MB which is 1.65 times more than the full BEM. Either decreasing or increasing the size of the projection matrix could change these results. However, had the ROM not been applied the SEBEM it would have used close to 27 GB of memory which highlights why the ROM is a crucial part of making the model applicable on regular machines.

Table 1. Computation efforts: Full BEM vs. ROSEBEM.

	Full BEM	ROSEBEM
Computational Time	6.914 min	65 min
Memory Consumption	224.297 MB	370.789 MB

5 CONCLUSIONS

In this short work we extended a Model Order Reduction (MOR) technique for the Series Expansion Boundary Element Method (SEBEM) to handle the Boundary Layer Impedance (BLI) boundary condition. Similar to the previous work the series expansion was based on the Taylor expansion of all relevant kernels and the reduced order model based on the Galerkin projection. Lastly the model was numerically verified and analyzed using a cylinder as a test case. It was found that information from a single frequency was not enough to create a suitable projection basis.

REFERENCES

[1] Liu Yijun. *Fast multipole boundary element method: Theory and applications in engineering*, volume 9780521116596. Cambridge University Press, 2009. ISBN 9780511605345, 0521116597, 9780521116596. doi: 10.1017/CBO9780511605345.

[2] Wolfgang Hackbusch. *Hierarchical Matrices: Algorithms and Analysis*. Springer, 2015. ISBN 3662473232, 3662473240, 9783662473238, 9783662473245.

[3] Suhaib Koji Baydoun, Matthias Voigt, and Steffen Marburg. Low-rank iteration schemes for the multi-frequency solution of acoustic boundary element equations. *Journal of Theoretical and Computational Acoustics*, 29(3), 2021. doi: 10.1142/S2591728521500043.

[4] SM Kirkup and DJ Henwood. Methods for speeding up the boundary element solution of acoustic radiation problems. *Journal of Vibration and Acoustics-transactions of the Asme*, 114(3):374–380, 1992. doi: 10.1115/1.2930272.

[5] Dionysios Panagiotopoulos, Elke Deckers, and Wim Desmet. A two-step reduction method for acoustic BEM systems. In *Forum Acusticum*, pages 2291–2298, Lyon, France, December 2020. doi: 10.48465/fa.2020.0712. URL https://hal.archives-ouvertes.fr/hal-03235474.

[6] Dionysios Panagiotopoulos, Elke Deckers, and Wim Desmet. Krylov subspaces recycling based model order reduction for acoustic bem systems and an error estimator. *Computer Methods in Applied Mechanics and Engineering*, 359, 2020. doi: 10.1016/j.cma.2019.112755.

[7] Martin Berggren, Anders Bernland, and Daniel Noreland. Acoustic boundary layers as boundary conditions. *Journal of Computational Physics*, 371:633–650, 2018. doi: 10.1016/j.jcp.2018.06.005.

[8] Vicente Cutanda Henriquez. *Numerical Transducer Modeling*. PhD thesis, Technical University of Denmark, 2002.

[9] Jeffrey Werner Bezanson, Alan Edelman., Massachusetts Institute of Technology. Dept. of Electrical Engineering, and Massachusetts Institute of Technology. Dept. of Electrical Engineering. Julia : an efficient dynamic language for technical computing - efficient dynamic language for technical computing. 2012.

[10] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. URL 10.1137/141000671.

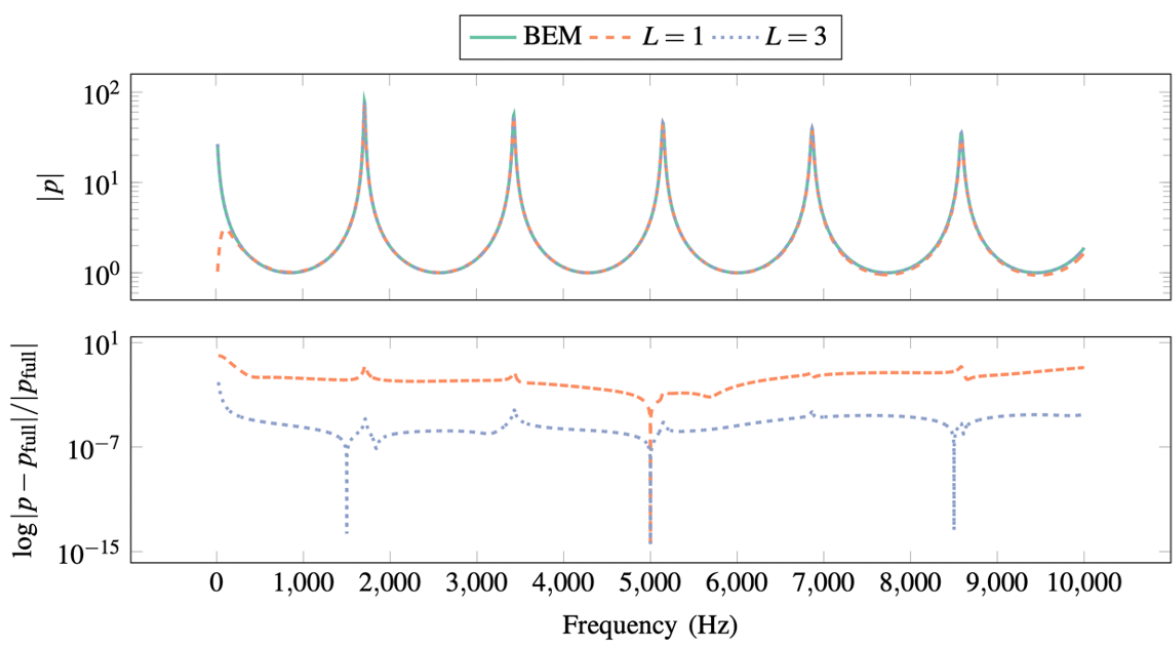
[11] DTU Computing Center. DTU Computing Center resources, 2021. URL https://doi.org/10.48714/DTU.HPC.0001.

[12] Vicente Cutanda Henríquez and Peter Møller Juhl. Openbem - an open source boundary element method software in acoustics. *Proceedings of Internoise 2010*, 7:5796–5805, 2010.

[13] MATLAB. *version 9.10.0 (R2021a)*. The MathWorks Inc., Natick, Massachusetts, 2021.

[14] Steffen Marburg. Six boundary elements per wavelength: Is that enough? *Journal of Computational Acoustics*, 10(1):25–51, 2002. doi: 10.1142/S0218396X02001401.

Errata (Figure 3)



Paper C2

forum acusticum 2023

AN OPEN-SOURCE BOUNDARY ELEMENT FRAMEWORK FOR LARGE-SCALE VISCOTHERMAL ACOUSTICS

Mikkel Paltorp[1,3,*] **Vicente Cutanda Henriquez**[2,3]
[1] Acoustic Technology Group (ACT), 2800 Kgs. Lyngby
[2] Centre for Acoustic-Mechanical Microsystems (CAMM), 2800 Kgs. Lyngby
[3] Department of Electrical and Photonics Engineering, The Technical University of Denmark

ABSTRACT

BoundaryIntegralEquations.jl is an open source software library aimed at solving the Kirchhoff-Helmholtz Integral Equation using the collocation Boundary Element Method (BEM). The software is written in the Julia programming language, making it both easy to use and maintain while also being computationally efficient. The package builds upon the ideas of the OpenBEM software, but adds on additional functionality such as the Fast Multipole Method (FMM) and sparse assembly of all matrices used in the Kirchhoff Decomposition (KD) description of viscous and thermal losses. As such the package takes the first steps towards large-scale BEM computations including viscous and thermal losses. The package is validated using simple geometries, such as cuboids and spheres, where an analytical solution exist.

Keywords: boundary element method, viscothermal losses, open-source, fast multipole method

1. INTRODUCTION

The Helmholtz equation is often met when solving time-harmonic acoustical problems. The analytical solution to this Partial Differential Equation (PDE) is limited to simple geometries such as e.g. spheres, cuboids and cylinders. As the real geometry of the world much more complex than this the solution to the PDE is more often than not approximated using a numerical scheme such as the Finite Element Method (FEM) or the Boundary Element Method (BEM).

This paper focuses on the latter method and introduces the open-source library `BoundaryIntegeralEquations.jl` aimed at solving the Helmholtz using the collocation Boundary Element Method [1]. The design of the package is inspired by the OpenBEM package written in MatLab [2]. The package itself is implemented in the Julia programming language as it was found to be good compromise of efficiency and maintainability [3]. Additionally the Julia language is free and open-source making it accessible to a larger audience than the OpenBEM package.

2. THEORY

The basic idea of the Boundary Element Method is to transform the Helmholtz equation into the Kirchoff-Helmholtz integral equation. The weak form of this integral equation is

$$\int_\Gamma \phi(\mathbf{x}) \left(\alpha\zeta(\mathbf{x})p(\mathbf{x}) - \int_\Gamma \frac{\partial G(\alpha,\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})} p(\mathbf{y})\, \mathrm{d}S_\mathbf{y} - \alpha \mathrm{i}\rho c k \int_\Gamma G(\alpha,\mathbf{x},\mathbf{y}) v_\mathbf{n}(\mathbf{y})\, \mathrm{d}S_\mathbf{y} \right) \mathrm{d}S_\mathbf{x} = 0, \mathbf{x} \in \Omega, \tag{1}$$

where i is the imaginary unit, k is the wavenumber, c is the speed of sound of the medium, ρ is the medium density, Ω is the domain of interest, $\Gamma = \partial\Omega$ is the boundary of the domain, $\zeta(\mathbf{x})$ is the integral free term at point $\mathbf{x}$, $\mathbf{n}(\mathbf{y})$ is the normal vector at point $\mathbf{y}$, $p(\mathbf{y})$ is the pressure at point $\mathbf{y}$, $v_\mathbf{n}(\mathbf{y})$ is the normal velocity at point $\mathbf{y}$, $\phi(\mathbf{x})$ is a so-called test function and $G(\alpha,\mathbf{x},\mathbf{y})$ is the Green's function

*__Corresponding author__: mpasc@dtu.dk.

defined as

$$G(\alpha, \mathbf{x}, \mathbf{y}) = \frac{\exp\left(-\alpha \mathrm{i} k \|\mathbf{x}-\mathbf{y}\|_2\right)}{4\pi \|\mathbf{x}-\mathbf{y}\|_2}, \tag{2}$$

with α being the sign of the chosen time dependency [4]. In the case of collocation BEM the test function is chosen as the sum of Dirac delta functions, i.e. that

$$\phi(\mathbf{x}) = \mathbf{a}^\top \begin{bmatrix} \delta(\mathbf{x}-\mathbf{t}_1) \\ \vdots \\ \delta(\mathbf{x}-\mathbf{t}_n) \end{bmatrix}, \tag{3}$$

where $\mathbf{a} \in \mathbb{C}^n$ is a vector of arbitrary coefficients and $\mathbf{t}_i$ is the i-th collocation point, with the set of collocation points being the collection of all interpolation nodes from the elements describing the geometry (see Figure 1 and Figure 2). The discretization of (1) is a two step process: First a discretization of the boundary into so-called elements and secondly by approximating the pressure and the velocity on each element by simple functions. Part of the first step is to represent the coordinates of element e as

$$\mathbf{x}^e(\mathbf{u}) = \mathbf{X}^e \mathbf{N}^e(\mathbf{u}) \in \Gamma^e, \ \forall \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \in \mathcal{L}^e \subset \mathbb{R}^2, \tag{4}$$

where $\mathbf{X}^e$ is a matrix with columns equal to the interpolation nodes of the geometry, $\mathbf{N}^e(\mathbf{u})$ are the shape functions of the e-th element, Γ^e is the global element coordinates and $\mathcal{L}^e$ represent the local element coordinates of the e-th element. As an example, a flat (linear) triangular element will have

$$\mathbf{N}^e(\mathbf{u}) = \begin{bmatrix} 1-u_1-u_2 \\ u_1 \\ u_2 \end{bmatrix}, \ u_1 \in [0,1], \ u_2 \in [0, 1-u_1], \tag{5}$$

with the columns of $\mathbf{X}^e$ being equal to the corners of the triangle. The integral over the e-th element can be approximated using a quadrature scheme as

$$\begin{aligned} \int_{\Gamma^e} f(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}} &= \int_{\mathcal{L}^e} \mathcal{J}(\mathbf{u}) f(\mathbf{u}) \, \mathrm{d}\mathbf{u} \\ &\approx \sum_{i=1}^{Q} w_i \mathcal{J}(\mathbf{u}_i) f(\mathbf{u}_i), \end{aligned} \tag{6}$$

where $(w_i, \mathbf{u}_i)$ is a set of quadrature weights and points, Q is the number of quadrature weights/points and $\mathcal{J}(\mathbf{u})$ is the Jacobian of the parametrization. For 2D surface in 3D, which is the case for BEM, the Jacobian represents the area distortion from $\mathcal{L}^e$ to Γ^e in (4). This area distortion can be computed as

$$\mathcal{J}(\mathbf{u}) = \left\| \left(\mathbf{X}^e \frac{\mathrm{d}\mathbf{N}^e(\mathbf{u})}{\mathrm{d}u_1} \right) \times \left(\mathbf{X}^e \frac{\mathrm{d}\mathbf{N}^e(\mathbf{u})}{\mathrm{d}u_2} \right) \right\|_2. \tag{7}$$

For the second step the discretization of the pressure and normal velocity is

$$p(\mathbf{x}) \approx \mathbf{T}(\mathbf{x})\mathbf{p}, \quad v_{\mathbf{n}}(\mathbf{x}) \approx \mathbf{T}(\mathbf{x})\mathbf{v}_{\mathbf{n}}, \tag{8}$$

where $\mathbf{T}(\mathbf{x})$ is a row vector containing the global nodal interpolation functions while $\mathbf{p}$ and $\mathbf{v}_{\mathbf{n}}$ contain the nodal values of respectively the pressure and normal velocity. Taking the pressure as an example it follows further that the pressure on the e-th element can be described as

$$p(\mathbf{x}^e(\mathbf{u})) = \mathbf{T}(\mathbf{x}^e(\mathbf{u}))\mathbf{p} = \underbrace{\mathbf{T}(\mathbf{x}(\mathbf{u}))(\mathbf{L}^e)^\top}_{\mathbf{T}^e(\mathbf{u})} \underbrace{\mathbf{L}^e \mathbf{p}}_{\mathbf{p}^e}, \ \mathbf{u} \in \mathcal{L}^e$$

where $\mathbf{L}^e$ is a permutation-like matrix that extracts the relevant rows of $\mathbf{p}$ and orders them into $\mathbf{p}^e$ such that they correspond to the order of the local basis functions of $\mathbf{T}^e(\mathbf{u})$. The advantage of this description is that $\mathbf{T}^e(\mathbf{u})$ can be chosen to the same for all elements. As an example, a continuous linear interpolation of the pressures will have

$$\mathbf{T}^e(\mathbf{u}) = \begin{bmatrix} 1-u_1-u_2 & u_1 & u_2 \end{bmatrix}, \tag{9}$$

where $u_1 \in [0,1]$ and $u_2 \in [0, 1-u_1]$. The normal velocity on the e-th element can be defined analogously. Inserting the element approximations of the pressure, the normal velocity and the test function into (1), while imposing that the equality has to hold for all $\mathbf{a} \in \mathbb{C}^n$, the following linear system is generated

$$\alpha \mathrm{diag}(\zeta)\mathbf{p} - \mathbf{F}\mathbf{p} - \alpha \mathrm{i} \rho c k \mathbf{G} \mathbf{v}_{\mathbf{n}} = \mathbf{0}, \tag{10}$$

where the kth row of $\mathbf{G}$ is approximated using a quadrature scheme as

$$\begin{aligned} &\underbrace{\left(\sum_{e=1}^{N} \left(\int_{\Gamma^e} G(\mathbf{t}_k, \mathbf{y}) \mathbf{T}(\mathbf{y}) \, \mathrm{d}S_{\mathbf{y}} \right) \right)}_{k\text{th row of } \mathbf{G}} \\ &\approx \left(\sum_{e=1}^{N} \left(\sum_{i=1}^{Q(k,e)} G(\mathbf{t}_k, \mathbf{y}^e(\mathbf{u}_i)) w_i \mathcal{J}(\mathbf{u}_i) \mathbf{T}^e(\mathbf{u}_i) \right) \mathbf{L}^e \right), \end{aligned} \tag{11}$$

with a similar approximation of the kth row of $\mathbf{F}$

$$\begin{aligned}&\Bigg(\underbrace{\sum_{e=1}^{N}\left(\int_{\Gamma^e}\frac{\partial G(\mathbf{t}_k,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}}\right)}_{k\text{th row of }\mathbf{F}}\Bigg)\\ &\approx\left(\sum_{e=1}^{N}\left(\sum_{i=1}^{Q(k,e)}\frac{\partial G(\mathbf{t}_k,\mathbf{y}^e(\mathbf{u}_i))}{\partial \mathbf{n}(\mathbf{y}^e(\mathbf{u}_i))}w_i\mathcal{J}(\mathbf{u}_i)\mathbf{T}^e(\mathbf{u}_i)\right)\mathbf{L}^e\right).\end{aligned}\tag{12}$$

In both cases the number of quadrature points, $Q(k,e)$, depends on the collocation point and the element.

While (10) can be solved directly, a naive discretization will result in $\mathbf{G},\mathbf{F}\in\mathbb{C}^{n\times n}$ being fully populated matrices. Due to the $\mathcal{O}(n^2)$ scaling of the memory, as well as the matrix vector products, this naive approach will only be possible for relatively small number of Degrees of Freedom ($n<50.000$). In the following section we show how either the Fast Multipole Method (FMM) [1] or Hierarchical matrices ($\mathcal{H}$-matrices) [5] can be utilized to reduce the scaling respect to both memory and computation from $O(n^2)$ to $O(n\log(n))$.

2.1 Acceleration Methods

In order to simplify the description of how to accelerate the Boundary Element (BE) computations we assume that the integration of each element requires the same Q quadrature points to be approximately correct. This will be the source of numerical error for every element close to the collocation point. This error can, however, be dealt with by a so-called near field correction step (as shown in (15)). As a result of this assumption it is possible to describe the product of $\mathbf{G}$ and $\mathbf{F}$ with a known vector $\mathbf{z}$ easily. In order to avoid too large expressions in the following we again zoom in on the multiplication with the kth rows. Starting with $\mathbf{G}$ it follows that

$$\begin{aligned}&\Bigg(\underbrace{\sum_{e=1}^{N}\left(\int_{\Gamma^e}G(\mathbf{t}_k,\mathbf{y})\mathbf{T}(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}}\right)}_{k\text{th row of }\mathbf{G}}\Bigg)\mathbf{z}\\ &\approx\left(\sum_{j=1}^{NQ}G(\mathbf{t}_k,\mathbf{y}_j)\underbrace{w_j\mathcal{J}(\mathbf{u}_j)\mathbf{T}^{e(j)}(\mathbf{u}_j)\mathbf{L}^{e(j)}}_{j\text{th row of }\mathbf{C}}\right)\mathbf{z}\\ &=\begin{bmatrix}G(\mathbf{t}_k,\mathbf{y}_1) & \dots & G(\mathbf{t}_k,\mathbf{y}_{NQ})\end{bmatrix}\mathbf{C}\mathbf{z},\end{aligned}\tag{13}$$

where the subscript j refers to an ordering of the collection of Gaussian points from all elements and $e(j)$ is a function that returns the element number that Gaussian point j is located on. The matrix $\mathbf{C}$ is sparse and can be thought of as a transformation of $\mathbf{z}$ into coefficients $\mathbf{c}=\begin{bmatrix}c_1 & c_2 & \dots & c_{NQ}\end{bmatrix}^\top$. A similar approach can be applied to $\mathbf{F}$ resulting in

$$\begin{aligned}&\Bigg(\underbrace{\sum_{e=1}^{N}\left(\int_{\Gamma^e}\frac{\partial G(\mathbf{t}_k,\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\mathbf{T}(\mathbf{y})\,\mathrm{d}S_{\mathbf{y}}\right)}_{k\text{th row of }\mathbf{F}}\Bigg)\mathbf{z}\\ &\approx\left(\sum_{j=1}^{NQ}\nabla G(\mathbf{t}_k,\mathbf{y}_j)\cdot\mathbf{n}(\mathbf{y}_j)\underbrace{w_j\mathcal{J}(\mathbf{u}_j)\mathbf{T}^{e(j)}(\mathbf{u}_j)\mathbf{L}^{e(j)}}_{j\text{th row of }\mathbf{C}}\right)\mathbf{z}\\ &=\begin{bmatrix}\nabla G(\mathbf{t}_k,\mathbf{y}_1)\cdot\mathbf{n}(\mathbf{y}_1) & \dots & \nabla G(\mathbf{t}_k,\mathbf{y}_{NQ})\cdot\mathbf{n}(\mathbf{y}_{NQ})\end{bmatrix}\mathbf{C}\mathbf{z},\end{aligned}\tag{14}$$

where $\mathbf{C}$ is the same transform as for the $\mathbf{G}$ operator. In order to fix the numerical inaccuracies caused by the elements close to each collocation we add a so-called near field correction matrix, $\mathbf{S}$, to the matrix product. As such multiplying with either $\mathbf{G}$ and $\mathbf{F}$ is as follows

$$\mathbf{A}=\mathbf{B}\mathbf{C}+\mathbf{S},\quad \mathbf{S},\mathbf{A}\in\mathbb{C}^{n\times n},\ \mathbf{B},\mathbf{C}^\top\in\mathbb{C}^{n\times NQ},\tag{15}$$

where $\mathbf{A}$ represent either $\mathbf{G}$ or $\mathbf{F}$, $\mathbf{B}$ is the fully populated part for which multiplication can be accelerated by the FMM or a $\mathcal{H}$-matrix, $\mathbf{C}$ is the coefficient map and $\mathbf{S}$ is the near field correction. In short $\mathbf{S}$ corrects the computation by subtracting the wrong integration done by using only Q Gaussian points and adds on the correct integration instead. Note that both $\mathbf{S}$ and $\mathbf{C}$ are sparse matrices, meaning that their assembly and matrix-vector-products scale $\mathcal{O}(n\tau)$ where $\tau\ll n$. As such using an approximate scheme for $\mathbf{B}$ is all that is needed to reduce the memory and computational scaling of $\mathbf{G}$ and $\mathbf{F}$.

2.2 Viscous and thermal losses

The inclusion of viscous and thermal losses can be achieved utilizing the Kirchoff Decomposition [6]. This approach splits the problem into three separate Helmholtz equations, commonly referred to as modes, as follows

$$\text{Acoustic Mode:}\quad (\Delta+k_a^2)p_a(\mathbf{x})=0,\tag{16}$$

$$\text{Thermal Mode:}\quad (\Delta+k_h^2)p_h(\mathbf{x})=0,\tag{17}$$

$$\text{Viscous Mode:}\quad (\Delta+k_v^2)\mathbf{v}_v(\mathbf{x})=\mathbf{0},\tag{18}$$

where the viscous mode is divergence free ($\nabla\cdot\mathbf{v}_v(\mathbf{x})=0$). The three modal wavenumbers ($k_a,\ k_h$ & k_v) all depend on the lossless wavenumber (k) as well as the physical properties of the fluid, such as the thermal conductivity, specific heat capacity under constant pressure and

the shear/bulk viscosity coefficients [7]. The three modes are coupled through the isothermal and no-slip boundary conditions. The first boundary condition states that there is no change of the surface temperature while the second boundary condition states that the fluid sticks to the surface. Mathematically the two boundary conditions are given as

$$\begin{aligned} p_a(\mathbf{x})\tau_a + p_h(\mathbf{x})\tau_h = 0, \\ \nabla p_a(\mathbf{x})\phi_a + \nabla p_h(\mathbf{x})\phi_h + \mathbf{v}_v(\mathbf{x}) = \mathbf{v}_s(\mathbf{x}), \end{aligned} \tag{19}$$

where $\mathbf{x} \in \Gamma$, $\mathbf{v}_s(\mathbf{x})$ is prescribed surface velocity at point $\mathbf{x}$ and like the modal wavenumbers the constants τ_a, τ_h, ϕ_a, and ϕ_h depends on the lossless wavenumber and the physical properties of the fluid. The total pressure and velocity can be extracted from the three modes as follows

$$p_t = p_a + p_h, \tag{20}$$

$$\mathbf{v}_t = \mathbf{v}_a + \mathbf{v}_h + \mathbf{v}_v. \tag{21}$$

Each of the three modes give rise to a linear system of equations similar to that of (10). Writing this out it means that the following must be true

$$\begin{aligned} \mathbf{H}_a\mathbf{p}_a + \mathbf{G}_a\frac{\partial \mathbf{p}_a}{\partial \mathbf{n}} = \mathbf{0}, \quad \mathbf{H}_h\mathbf{p}_h + \mathbf{G}_h\frac{\partial \mathbf{p}_h}{\partial \mathbf{n}} = \mathbf{0}, \\ \mathbf{H}_v\mathbf{v} + \mathbf{G}_v\frac{\partial \mathbf{v}}{\partial \mathbf{n}} = \mathbf{0}, \end{aligned} \tag{22}$$

where $\mathbf{H}_i = \alpha\mathrm{diag}(\zeta_i) - \mathbf{F}_i$ and $\mathbf{G}_i = -\alpha \mathrm{i}\rho c k \mathbf{G}_i$ for $i \in \{a, h, v\}$. Note that the viscous mode is special as it is a result of a vector Helmholtz equation. In the above it was chosen to separate the x, y and z components as $\mathbf{v} = \begin{bmatrix}\mathbf{v}_x & \mathbf{v}_y & \mathbf{v}_z\end{bmatrix}^\top$ and $\frac{\partial \mathbf{v}}{\partial \mathbf{n}} = \begin{bmatrix}\frac{\partial \mathbf{v}_x}{\partial \mathbf{n}} & \frac{\partial \mathbf{v}_y}{\partial \mathbf{n}} & \frac{\partial \mathbf{v}_z}{\partial \mathbf{n}}\end{bmatrix}^\top$. Using this description will result in $\mathbf{G}_v$ and $\mathbf{H}_v$ being block diagonal matrices with the three blocks being identical and computed similar to the acoustical and thermal mode but with wavenumber k_v. In order to assert the null-divergence of the viscous mode, as well as applying the no-slip boundary condition, we need to be able to compute the gradient. The idea in both cases is to split the gradient computation into a tangential part (denoted by $^\parallel$) and a orthogonal part (denoted by $^\perp$). In the case of the divergence this means that

$$\nabla \cdot \mathbf{v}_v = \nabla^\parallel \cdot \mathbf{v}_v + \nabla^\perp \cdot \mathbf{v}_v, \tag{23}$$

while for the gradients this means that

$$\nabla p_a = \nabla^\parallel p_a + \nabla^\perp p_a, \quad \nabla p_h = \nabla^\parallel p_h + \nabla^\perp p_h. \tag{24}$$

It turns out that since the Boundary Element discretization only prescribes values on the surface, then the tangential part of the gradient is extracted directly from the interpolation functions. Likewise the orthogonal part can be extracted from the Boundary Element discretization of the normal derivatives by multiplication with the normals. As such the divergence can be computed as

$$\nabla \cdot \mathbf{v} = \mathbf{D}_r\mathbf{v} + \mathbf{N}^\top\frac{\partial \mathbf{v}}{\partial \mathbf{n}}, \tag{25}$$

where $\mathbf{N} = \begin{bmatrix}\mathrm{diag}(\mathbf{n}_x) & \mathrm{diag}(\mathbf{n}_y) & \mathrm{diag}(\mathbf{n}_z)\end{bmatrix}^\top$ with $\mathbf{n}_x$, $\mathbf{n}_y$ and $\mathbf{n}_z$ being respectively the vectors collecting the x, y and z components of the normals at the collocation points and $\mathbf{D}_r = \begin{bmatrix}\mathbf{D}_x & \mathbf{D}_y & \mathbf{D}_z\end{bmatrix}$ with the sparse matrices $\mathbf{D}_x$, $\mathbf{D}_y$ and $\mathbf{D}_z$ being computed directly from the Boundary Element interpolation using the interpolation function derivative approach [6]. Using the same idea for the gradients it is found that

$$\nabla \mathbf{p}_a = \mathbf{D}_c\mathbf{p}_a + \mathbf{N}\frac{\partial \mathbf{p}_a}{\partial \mathbf{n}}, \ \nabla \mathbf{p}_h = \mathbf{D}_c\mathbf{p}_h + \mathbf{N}\frac{\partial \mathbf{p}_h}{\partial \mathbf{n}}, \tag{26}$$

where $\mathbf{D}_c = \begin{bmatrix}\mathbf{D}_x^\top & \mathbf{D}_y^\top & \mathbf{D}_z^\top\end{bmatrix}^\top$.

Putting everything together it turns out that the acoustical pressure can be computed by solving the following linear system of equations

$$\begin{aligned} &\left[\mathbf{G}_a\left(\mu_a\left(\mathbf{RN}\right)^{-1}\mathbf{RD}_c + \mu_h\mathbf{G}_h^{-1}\mathbf{H}_h\right) - \phi_a\mathbf{H}_a\right]\mathbf{p}_a \\ &= \mathbf{G}_a\left(\mathbf{RN}\right)^{-1}\mathbf{R}\mathbf{v}_s. \end{aligned} \tag{27}$$

where $\mu_h = {}^{\tau_a\phi_h}/_{\tau_h}$, $\mu_a = \phi_a - \mu_h$ and $\mathbf{v}_s = \begin{bmatrix}\mathbf{v}_{s_x} & \mathbf{v}_{s_y} & \mathbf{v}_{s_z}\end{bmatrix}^\top$ is the boundary velocities stacked with respect to the x, y, and z direction and

$$\mathbf{R} = \mathbf{D}_r - \mathbf{N}^\top\mathbf{G}_v^{-1}\mathbf{H}_v. \tag{28}$$

An important detail to mention is that only the acoustical mode, corresponding to $\mathbf{G}_a$ and $\mathbf{H}_a$, are fully populated matrices while the remaining matrices in (27) are sparse. This means that simply utilizing one of the acceleration strategies for the acoustical mode is enough to make (27) solvable for even relatively large problems. Using the no-slip boundary condition it is possible to extract the viscous velocity as

$$\mathbf{v} = \mathbf{v}_s - \left(\mu_a\mathbf{D}_c + \mu_h\mathbf{N}\mathbf{G}_h^{-1}\mathbf{H}_h - \phi_a\mathbf{N}\mathbf{G}_a^{-1}\mathbf{H}_a\right)\mathbf{p}_a. \tag{29}$$

The normal and tangential components of the viscous velocity, as described above, will later be used to verify the simulation results (Figure 7).

3. IMPLEMENTATION

This section gives an overview of the currently supported elements types, mesh file formats, and acceleration methods.

3.1 Element Types

The software package currently supports triangular and quadrilateral elements. For the geometry both linear and quadratic elements are supported. Additionally the global interpolation functions for the pressure and normal derivative, as shown in (8), can be chosen to be discontinuous constant, linear and quadratic. For continuous elements interpolation nodes are being put on the boundary of the element while for the discontinuous elements the interpolation nodes are exclusively put inside the domain of the elements. The different setups of geometric and interpolation elements of triangles can be seen in Figure 1 while the elements for the quadrilaterals can be seen in Figure 2.

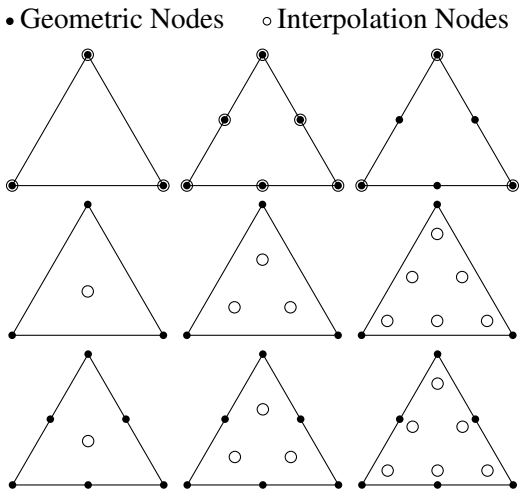


Figure 1. Top row: Linear and quadratic continuous interpolation. Middle row: Linear Geometry with discontinuous Constant, linear and quadratic interpolation. Bottom row: Quadratic Geometry with discontinuous constant, linear and quadratic interpolation.

3.2 Meshes

Currently flat triangular panels can be imported through the `OBJ`, `STL`, `PLY`, `OFF` and `2DM` file formats using the `MeshIO.jl` package. Additionally both triangular and quadrilateral panels of linear and quadratic order can be imported using the `MPHTXT` file format from COMSOL Multiphysics® [8].

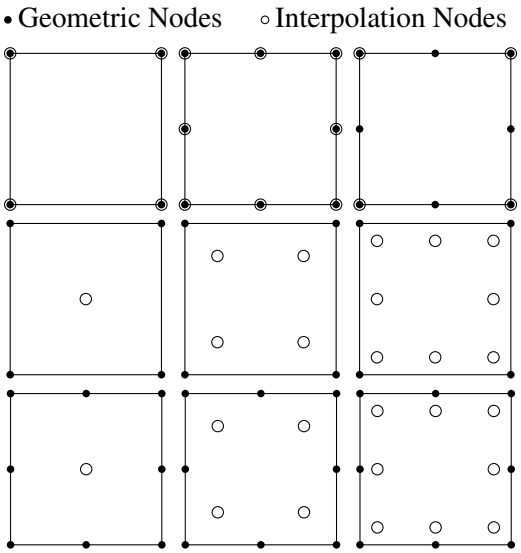


Figure 2. Top row: Linear and quadratic continuous interpolation. Middle row: Linear Geometry with discontinuous Constant, linear and quadratic interpolation. Bottom row: Quadratic Geometry with discontinuous constant, linear and quadratic interpolation.

3.3 Fast Operators

The implementation currently supports applying $\mathbf{G}$ and $\mathbf{F}$ using the Fast Multipole Method [1] (through the FMM3D library [9]) and the $\mathcal{H}$-matrix approach [5] (using the `HMatrices.jl` package [10]). Additionally there is an experimental implementation of the Interpolated Factored Green's Function approach [11] (using the `IFGF.jl` package [12]).

4. EXAMPLES

In the following examples, it was chosen to set the speed of sound as $c = 343\,\mathrm{m\,s^{-1}}$ and the medium density of air as $\rho = 1.21\,\mathrm{kg\,m^{-3}}$. In the case of viscothermal losses the ambient properties of air was used. A more in-depth descriptions of each the three examples, including the code, can be found in the online documentation of the software.

4.1 Rigid sphere with radius equal to 1m

The acoustic scattering of a rigid sphere coming from a plane wave traveling in the z-direction can be computed analytically as [13]

$$p^{\text{analytical}}(r,\theta) = P_0 \exp\left(\mathrm{i}kr\cos(\theta)\right) - P_0 \sum_n^{\infty} \mathrm{i}^n (2n+1) \frac{j_n'(ka)}{h_n'(ka)} P_n(\cos(\theta)) h_n(kr), \tag{30}$$

where r is the distance to the origin, θ is the colatitude angle, P_0 is the magnitude of the plane wave, j_n is the spherical Bessel function of the first kind, h_n is the spherical Hankel function of the first kind and a is the radius of the sphere.

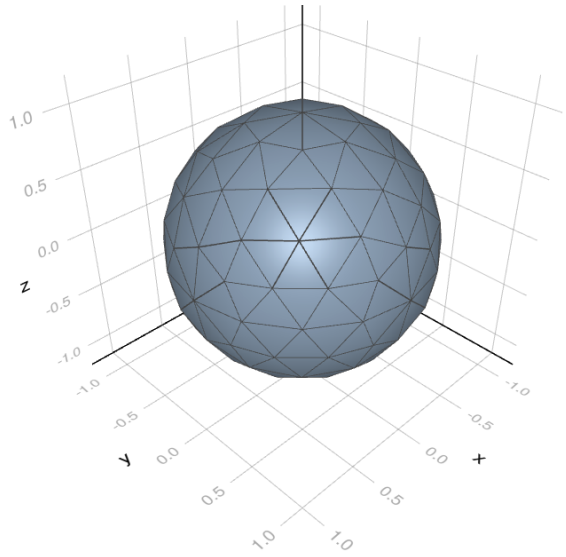


Figure 3. Visualization of the sphere mesh. Note that while the mesh is quadratic only flat triangles are plotted due to limitations in the used plotting library.

In the simulation we set $P_0 = 1\,\text{Pa}$ and $f = 100\,\text{Hz}$. The geometry of the sphere was approximated using 246 quadratic triangular elements with the interpolation function set to be discontinuous linear (Figure 3). As a result the simulation had 738 Degrees of Freedom. Furthermore given that the sphere is rigid we set $\mathbf{v_n} = \mathbf{0}$ in (10). As such this simulation solves

$$\left(\alpha \text{diag}(\zeta) - \mathbf{F}\right)\mathbf{p} = \mathbf{p}^{\text{incident}}, \tag{31}$$

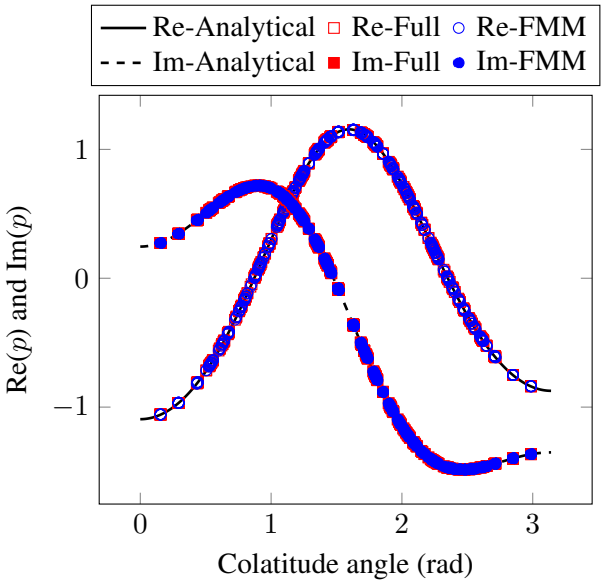


Figure 4. Real and imaginary part of the surface pressure as a function of the angle for a rigid sphere with radius $1\,\text{m}$ at $100\,\text{Hz}$.

where $\mathbf{p}^{\text{incident}}$ is the pressure of the plane wave at the collocation nodes. The results of the simulation can be seen in Figure 4.

4.2 Plane wave in cube with side lengths equal to 1m

The analytical description of the pressure from a plane wave is given by

$$p(\mathbf{x}) = P_0 \exp\left(-\mathrm{i}\mathbf{k}\cdot\mathbf{x}\right), \tag{32}$$

where P_0 is the magnitude of the plane wave, $\mathbf{k}$ is the wave vector and $\mathbf{x}$ is the position vector. If a plane wave is perfectly aligned with the sides of an open duct the pressure should behave the same as (32), i.e. a plane wave in the free field. In the simulation we align the wave with the x-direction and set $P_0 = 1\,\text{Pa}$ and $f = 54.59\,\text{Hz}$. Given the flat sides of the cube it was chosen to use 224 linear elements for the geometry while the chosen interpolation was set to be discontinuous quadratic resulting in 1344 Degrees of Freedom. For the simulation the side at $x = 0$ is applied a pressure condition equal to P_0 while the side at $x = 1$ is applied a ρc impedance condition. The remaining boundaries are set as rigid ($\mathbf{v}_n = \mathbf{0}$). The resulting linear system of equations becomes

$$\left(\mathbf{H}\text{diag}(\mathbf{p}_0^{\mathbb{C}}) - \alpha \mathrm{i}\rho c k \mathbf{G}\text{diag}(\mathbf{y})\right)\mathbf{z} = -\mathbf{H}\text{diag}(\mathbf{p}_0), \tag{33}$$

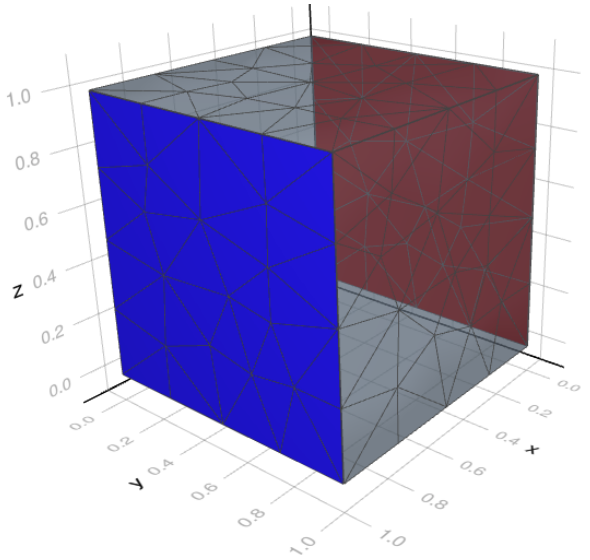


Figure 5. Visualization of the cube mesh. The red side ($x = 0$) is applied a pressure condition while the blue side ($x = 1$) is applied an impedance condition.

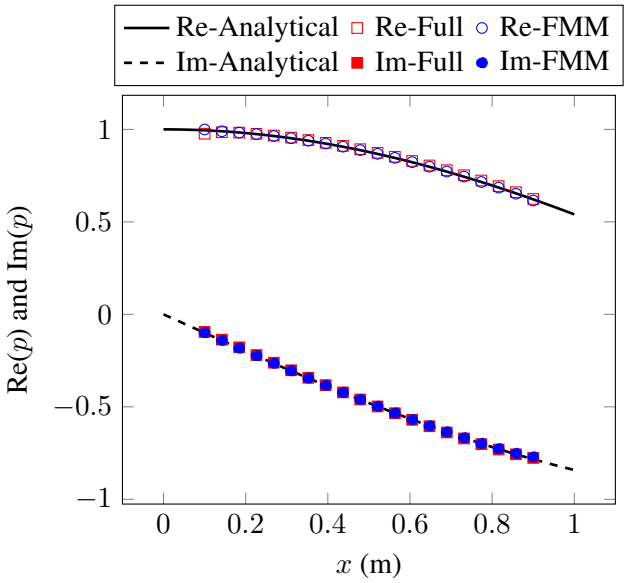


Figure 6. Real and imaginary part of the pressure on the line $(x, 0.5, 0.5)$ inside of the duct of side lengths 1 m at 54.59 Hz.

where $\mathbf{H} = (\alpha \text{diag}(\zeta) - \mathbf{F})$, $\mathbf{p}_0$ is a vector containing P_0 at the collocation points positioned at $x = 0$, $\mathbf{p}_0^{\mathbf{C}} = \mathbf{1} - \mathbf{p}_0/P_0$ and $\mathbf{y}$ contains the prescribed admittance at the collocation points positioned at $x = 1$. Finally, (33) is solved iteratively without ever forming the matrix-matrix products. From the solution ($\mathbf{z}$) the unknown surface pressures and normal velocities are extracted. Using this the pressure inside the cube is evaluated using (1) directly. The results can be seen in Figure 6.

4.3 Oscillating sphere with radius equal to 1m including viscothermal losses

The analytical solution of a sphere oscillating in the z-direction in a viscous fluid is described in section 6.9 of [14]. While the solution is stated as a infinite series it is dominated by the first order terms. As such the analytical solution for the acoustical pressure, normal velocity and tangential velocity can be approximated accurately as

$$\begin{aligned} p_a^{\text{analytical}}(r,\theta) &\approx -3\rho c A_1 h_1^{(1)}(kr)\cos(\theta) \\ |\mathbf{v_n}|^{\text{analytical}}(r,\theta) &\approx -\frac{6ikB_1}{k_v r} h_1(k_v r)\cos(\theta) \\ |\mathbf{v_t}|^{\text{analytical}}(r,\theta) &\approx -\frac{3iB_1\left(k_v r h_1'(k_v r) + h_1(k_v r)\right)}{r}\sin(\theta), \end{aligned} \tag{34}$$

where r is the distance to the origin, θ is the colatitude angle, h_1 is the spherical Hankel function of the first kind, k_v is the viscous wavenumber, and the coefficients A_1 and B_1 is computed as the solution to

$$\begin{bmatrix} v_z/(3ik) \\ av_z/(3i) \end{bmatrix} = \begin{bmatrix} h_1'(ka) & -2h_1(k_v a)/(ka) \\ h_1(ka) & -\left(k_v a h_1'(k_v a) + h_1(k_v a)\right) \end{bmatrix} \begin{bmatrix} A_1 \\ B_1 \end{bmatrix}, \tag{35}$$

where a is the radius of the sphere and v_z is the speed in the z-direction.

In the simulation we set $f = 100\,\text{Hz}$ and $v_z = 0.01\,\text{m}\,\text{s}^{-1}$. The mesh used was the same as for the scattering case (Figure 3). This time, however, utilizing continuous quadratic interpolation functions resulting in 494 Degrees of Freedom. Given the velocity in the z-direction the following surface velocity vector is created

$$\mathbf{v}_s = \begin{bmatrix} \mathbf{0} & \mathbf{0} & v_z\mathbf{1} \end{bmatrix}^\top \in \mathbb{R}^{3n}, \tag{36}$$

where $\mathbf{0}$ denotes a vector filled with zeros of length n and $\mathbf{1}$ denotes a vector filled with ones of length n. In order to compute the acoustical, $\mathbf{p}_a$, we solve (27) directly while $\mathbf{v}$ is computed afterwards using (29). The results of the simulation can be seen in Figure 7.

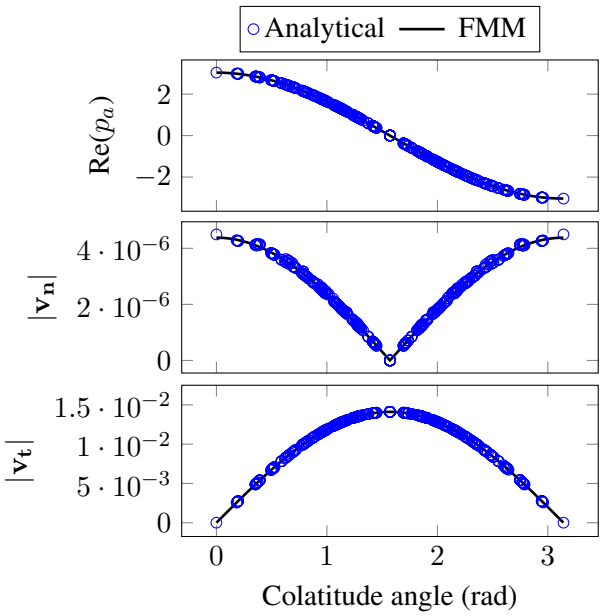


Figure 7. Real part of the acoustical pressure as well as absolute value of the normal and tangential viscous velocity for a sphere with 1 m radius in at 100 Hz.

5. CONCLUSIONS

In this paper some of the current features and implementation details of the open-source software `BoundaryIntegralEquations.jl` were explained. The package was validated on three different examples where a good compliance between the analytical solution and the simulation results. While the features are not expected to change we suggest that the reader look online for the most recent version of both the software and documentation.

6. ACKNOWLEDGMENTS

Many thanks to Simone Preuss (Lehrstuhl für Akustik mobiler Systeme, TUM) for many good discussions on BEM and the inclusion of viscothermal losses. It is always a pleasure.

7. REFERENCES

[1] L. Yijun, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*. Cambridge University Press, 2009.

[2] V. C. Henríquez and P. M. Juhl, "OpenBEM - an open source Boundary Element Method software in Acoustics," *Proceedings of Internoise 2010*, vol. 7, pp. 5796–5805, 2010.

[3] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.

[4] S. Marburg, "The Burton and Miller Method: Unlocking Another Mystery of Its Coupling Parameter," *Journal of Computational Acoustics*, vol. 24, no. 1, p. 1550016, 2016.

[5] M. Bebendorf, *Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems*. Springer, 2008.

[6] V. Cutanda Henríquez and P. R. Andersen, "A Three-Dimensional Acoustic Boundary Element Method Formulation with Viscous and Thermal Losses Based on Shape Function Derivatives," *Journal of Computational Acoustics*, vol. 26, no. 3, 2018.

[7] M. Brunaeu, P. Herzog, J. Kergomard, and J. Polack, "General formulation of the dispersion equation in bounded visco-thermal fluid, and application to some simple geometries," *Wave Motion*, vol. 11, no. 5, pp. 441–451, 1989.

[8] COMSOL AB, Stockholm, Sweden., "COMSOL Multiphysics®," 2021.

[9] T. Askham, Z. Gimbutas, L. Greengard, L. Lu, J. Magland, D. Malhotra, M. O'Neil, M. Rachh, and V. Rohklin, "FMM3D." `https://github.com/flatironinstitute/FMM3D`, 2023.

[10] L. M. Faria, "HMatrices.jl." `https://github.com/WaveProp/HMatrices.jl`, 2023.

[11] C. Bauinger and O. P. Bruno, ""Interpolated Factored Green Function" Method for accelerated solution of Scattering Problems," *Journal of Computational Physics*, vol. 430, p. 110095, 2021.

[12] L. M. Faria, "IFGF.jl." `https://github.com/WaveProp/IFGF.jl`, 2023.

[13] F. Ihlenburg, *Finite Element Analysis of Acoustic Scattering*. Springer, 1998.

[14] S. Temkin, *Elements of Acoustics*. Acoustical Society of America, 2001.

$\sqrt{17}$
$\Omega \int \delta e^{i\pi} = -1$
{2.7182818284}
ϑφερτυθιοπσδφγηξκλ
χ^2
$\sum$