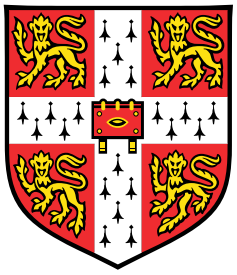
Shape Optimization of Annular Combustors

with an Open Source Parallelized Thermoacoustic Helmholtz Solver



Ekrem Ekici

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Wolfson College

June 2024

I would like to dedicate this thesis to my beautiful country ...

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures. As far as the spelling concerned, the author uses Oxford spelling, e.g. using 'parametrize' instead of 'parametrise'.

Ekrem Ekici
June 2024

Acknowledgements

First and foremost, I would like to thank my Allah, who is the most merciful and most generous. Without his help, I would not be able to continue my journey.

I would also like to acknowledge Prof. Matthew Juniper for supervising my PhD. His guidance was critical, especially during COVID.

I would also like to extend my thanks to Government of Türkiye for granting me a scholarship. Without their support and trust, these studies might not have been possible. I would also extend my thanks to Department of Engineering and Wolfson College for supporting my conference travels.

I would like to extend my gratitude to my lab mates Javier, Alexandros, Matthew and Jingquan for making such a sarcastic environment, to Stefano Falco for helping me to learn his Helmholtz solver in my first year, to Jørgen Dokken for providing an immense help to fix *FEniCSx* issues, to Dong Yang for having fruitful discussions on acoustic damping modelling, to Alberto Paganini for delivering a 2 hours online crash course on shape derivatives, to Peter Benie for being a wizard regarding *Linux* issues, to MDO lab for suggesting *pyGeo* and relevant resources for implementing FFD, to Simon Stow for suggesting the use of an industrial low order network code.

I would like to say my big appreciations for my dad, mum, brother and sister to complete my time in UK.

Last but not least, I would like to thank Allah again for granting me an 'Aşk'. She and Ceylan have completely changed my life.

Finally, learning how to think is a lifelong journey and this PhD can only be a part of it, not a result;

"So once you have fulfilled 'your duty', strive 'in devotion', turning to your Lord 'alone' with hope." [94:7-8]

Abstract

Thermoacoustic instability occurs in gas turbine combustors due to the dynamic coupling between heat release rate and acoustic oscillations. If the unsteady heat release rate is sufficiently in phase with the acoustic pressure, then the amplitude of the acoustic pressure intensifies. This then increases the fatigue of the components and can lead to engine failure. Frameworks that offer quick and accurate results to study thermoacoustic instability are desirable, since the system's stability is extremely sensitive to small changes. In this thesis, we offer a framework that simulates the thermoacoustic behaviour of increasingly complex geometries and suggests design changes through adjoint methods combined with geometry parametrization techniques.

We implement an inhomogeneous Helmholtz equation solver, *helmholtz-x*, written in an open-source framework. The mesh is generated with Gmsh and the solver uses DOLFINx and UFL from FEniCSx. The performance, validity, stability and extensibility of the solver are demonstrated through several examples of thermoacoustic instability, from the one-dimensional Rijke tube to the three-dimensional MICCA combustor. The implementation of Bloch-type boundary conditions is explained and tested.

We use *helmholtz-x* to perform shape optimization of thermoacoustic systems. We first propose a surface parametrization technique, NURBS, to study shape sensitivities of a 30kW laboratory-scale annular combustor (MICCA from EM2C). We parametrize the surfaces of the three-dimensional geometry with NURBS control points. We apply two different strategies, perpendicular boundary movements and control point perturbations, to implement shape changes proportional to these shape derivatives. With NURBS, we apply symmetry-preserving and symmetry-breaking geometry modifications for demonstration. For the plenum and the combustion chamber, we calculate the eigenvalue derivatives with respect to the positions of the NURBS control points. Following these derivatives, we impose displacements on the NURBS control points. We show that this geometry change reduces the growth rate of the unstable mode by increasing the phase shift between the pressure and the heat release rate oscillations.

In order to leverage the parametrization capability further, we then introduce the Free Form Deformation (FFD) technique to handle more complex thermoacoustic systems for

industrial applications. We study three geometries: Rijke tube and LPP combustor geometries, which are relatively simple demonstration cases, and an industrial aeroengine combustor geometry, which is more complicated. We use the same analysis as in the cases with NURBS parametrization, but for FFD control points. We modify the FFD control point positions in order to reduce the thermoacoustic growth rates of the unstable eigenmodes.

These findings show how, when combined with other constraints, the numerical framework we developed, *helmholtz-x*, could be used to study thermoacoustic oscillations and reduce their growth rates in geometrically parametrized industrial annular combustors through automated geometry changes.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

c	speed of sound
c_p	specific heat capacity at constant pressure
c_v	specific heat capacity at constant volume
d	diameter
f	body force
L	Length
M	Mach number
m	injected mass
$\mathbf{n}$	unit normal vector
p	pressure
q	heat release rate
R	reflection coefficient
r_{gas}	universal gas constant
T	temperature
t	time
u	velocity
$\mathbf{u}$	velocity vector

u_b bulk velocity

$\mathbf{x}$ position vector

Z specific acoustic impedance

ζ entropy

Greek Symbols

α standard deviation of Gaussian function

γ heat capacity ratio

i unit imaginary number $\sqrt{-1}$

κ curvature

ν specific density

ω angular frequency

ϕ_i trial function

ϕ_j test function

ρ density

τ time delay

Superscripts

$(\cdot)^*$ complex conjugate

$(\cdot)^\dagger$ adjoint operator

$\hat{(\cdot)}$ unsteady perturbed quantity (in frequency domain)

$(\cdot)'$ unsteady perturbed quantity (in time domain)

Subscripts

$(\cdot)_0$ mean quantity

$(\cdot)_1$ perturbed quantity

$(\cdot)_a$ ambient

$()_i$ the index of the control point in the radial direction

$()_j$ the index of the control point in the circumferential direction

$()_k$ the index of the control point in the axial direction

$(\cdot)_d$ downstream

$(\cdot)_f$ flame subdomain tag

$(\cdot)_{in}$ inlet

$(\cdot)_{out}$ outlet

$(\cdot)_r$ reference point

$(\cdot)_{sec}$ sector

$(\cdot)_{tot}$ total

$(\cdot)_u$ upstream

Other Symbols

$\mathscr{L}$ linear differential operator

∇ gradient operator

Acronyms / Abbreviations

API Application Programming Interface

CFD Computational Fluid Dynamics

FEM Finite Element Method

FFD Free Form Deformation

FTF Flame Transfer Function

LPP lean premixed prevaporized

MPI Message Passing Interface

NURBS Non-Uniform Rational B-Splines

PETSC Portable, Extensible Toolkit for Scientific Computation

SLEPC Scalable Library for Eigenvalue Problem Computations

Chapter 1

Introduction

Thermoacoustic oscillations endanger the safe operation of gas turbine engines. At certain operation regimes, unsteady feedback occurs between the heat release rate and the acoustic pressure. This leads to significant vibrations that influence the gas turbine engine components by reducing their structural and thermal performance. Despite the acoustic damping caused by the liner walls and dilution holes through vortex formation, oscillations can still persist and cause engine failure. One of the first practical outcomes of these are seen during the development of liquid rocket engines [1, 2]. Even after nearly a century of research, thermoacoustic oscillations continue to threaten the reliable operation of gas turbines, especially under the lean burn conditions driven by current greenhouse gas regulations [3, 4].

Even minor changes in thermoacoustic parameters and design can affect the stability of a system [5]. The influence of these changes needs to be determined during the design process. Adjoint methods are elegant, enabling cheap calculation of the gradients of the eigenvalue with respect to each parameter [6]. Two functions are required to find these: the direct and adjoint eigenfunction.

Helmholtz solvers are well-suited for analyzing the thermoacoustic behaviour of complex shaped, three-dimensional combustors. Here, we achieve this with the finite element method (FEM). The thermoacoustic Helmholtz equation is discretized with test and trial functions. We solve a discrete eigenvalue problem that returns an eigenvalue and an eigenvector of the thermoacoustic system. The sign of the imaginary part of the eigenvalue determines the system's stability. A positive sign indicates that the acoustic oscillations grow in time. In order to study a complex-shaped thermoacoustic system, we also implement the adjoint Helmholtz solver. The complex modeshapes and their stability can be determined, while simultaneously calculating the influence of the geometry parameters with adjoint methods.

Parametrization of complex geometries gives control over the design. This can be achieved with Berstein polynomials accompanied with control points. A first attempt to parametrize a thermoacoustic system was performed with B-Splines [7].

In this thesis, the influence of the shape of the combustor on the linear stability will be determined with more robust parametrization techniques combined with an adjoint Helmholtz solver. We build an open-source parallelized adjoint Helmholtz solver, which we call *helmholtz-x*. With this solver, we offer an extendable, reproducible and quick numerical framework to study thermoacoustic instabilities in complicated combustion systems. Having this tool, we further extend the B-Spline parametrization to Non-Uniform Rational B-Splines (NURBS) and Free Form Deformation (FFD) for shape optimization purposes. We demonstrate with examples that *helmholtz-x*, combined with robust shape parametrization tools, could be a useful design tool for gas turbine combustor manufacturers.

1.1 Thermoacoustic Mechanism

The thermoacoustic oscillation mechanism can be explained by an analogy between flame-acoustic waves and internal combustion engines where the piston is represented by acoustic waves, and the flame is represented by periodically combusted gas in the combustor [5]. When the acoustic waves encounter the flame, the flame becomes perturbed due to its interaction with the sound waves. Then these perturbations of the flame shape cause the heat release rate to fluctuate. If the moment of higher than average heat release rate coincides with moment of higher local acoustic pressure, the fluctuating portion of the heat release rate does work on the acoustic pressure waves. This work amplifies the acoustic pressures and, if it is not dissipated in the chamber, thermoacoustic instability arises. The mathematical representation of the thermoacoustic instability is given by the general form of Rayleigh's criterion [8]:

$$\int_{\Omega} \frac{(\gamma-1)\overline{p_1 q_1}}{\overline{\rho_0 c^2}} d\mathbf{x} > \int_{\partial\Omega} \overline{p_1 \mathbf{u}_1} dS \tag{1.1}$$

where overline denotes an average quantity over one period of the thermoacoustic oscillation, p_1 is the acoustic pressure, q_1 is the heat release rate perturbation and $\mathbf{u}_1$ is the acoustic velocity, γ is the ratio of specific heat capacities, c is the speed of sound and ρ_0 is the mean density. In Eq. 1.1, the left-hand side of the inequality expresses the amount of work done by the gas expansion due to the fluctuating heat release rate while the right-hand side term expresses the acoustic losses through the boundaries. If the volume integral term exceeds the surface integral term, then the acoustic oscillations grow in time. Therefore, in order to stabilize a thermoacoustic system, we need either to increase the phase difference between

the acoustic pressure and heat release rate fluctuation, or to increase the acoustic losses through the system boundaries.

1.2 Thermoacoustic Modelling Tools

In this section, we provide a brief overview of existing thermoacoustic modelling tools and their capabilities.

1.2.1 Low order network models

In low-order network models, the acoustics are modelled within connected modules with simple geometries. Each modules represent a particular component, for example, a duct, a burner, or a flame. These interconnected elements form a thermoacoustic network model for the corresponding system, together with acoustic boundary conditions. Starting with a model for relatively simple cases (the Rijke tube for example), these networks can be increasingly complex. For example they can model realistic thermoacoustic rigs [9, 10]. Frameworks such as LOTAN [11], OSCILOS[1] [12] and taX[2] [13] are the simplest tools for studying thermoacoustic instability. Low-order network models, however, can only model simple geometries, although they can account for mean flow effects such as entropy waves [14]. Despite being limited to simple geometries, longitudinal and circumferential modes can be captured by these models [15]. Thermoacoustic modelling of geometrically-complicated components and boundaries is mainly performed with scattering or transfer matrices [16]. Acoustic liners can be modelled with a state-space approach [17]. Despite their simplicity, network models perform well. However, tools with high resolution temperature fields such as Helmholtz solvers, may be required for more rigorous analysis.

1.2.2 Helmholtz solvers

Finite element models can model acoustics within complicated geometries containing spatially-varying temperature fields. These Helmholtz solvers assume that the mean flow Mach number is small and do not model entropy waves. For example, the package PyHoltz[3] is a Python-based Finite Element Method Helmholtz solver that calculates thermoacoustic eigenvalues and corresponding eigenvectors. It includes Bloch boundary conditions

[1] https://github.com/MorgansLab/OSCILOS_long
[2] https://gitlab.lrz.de/tfd/tax
[3] https://bitbucket.org/pyholtzdevelopers/public/src/master/

[18, 19], uncertainty quantification [20] and non-iterative solvers [21]. The authors of PyHoltz switched to the Julia programming language because it is a compiled language, which increases the speed of calculations. The new package is called WavesAndEigenvalues[4] and it implements nonlinear eigenproblem solvers with subspace [22] and iterative algorithms such as Banach' fixed point iteration and Householder's method [23]. Regarding the modelling of hydro-acoustic interactions to model dissipative effects, Helmholtz solvers are coupled with matrix boundary conditions [24, 25]. In this approach, the measured transfer matrix coming from experiments or LES acts as a 'transfer matrix' and builds an acoustic relationship between two separate Helmholtz domains.

Helmholtz solvers can be positioned between the low order network models and Large Eddy Simulations when studying thermoacoustics. This is because Helmholtz solvers are expensive to run compared to network models, but computationally cheaper than Large Eddy Simulations. Unlike LES, they can be combined with adjoint methods.

1.2.3 Large Eddy Simulation

Large Eddy Simulation (LES) resolves the reacting flow within combustion systems at high computation cost. It offers high fidelity information about the flow, which can be used to extract acoustics through, for example, dynamic mode decomposition (DMD) [26]. LES is used to investigate self-excited thermoacoustic instabilities for laboratory [27] and industrial [28] combustion systems. There are two main procedures to conduct LES to study thermoacoustic instabilities: forced response LES for system identification and brute-force LES to reveal self-excited modes [29]. LES outperforms other tools to capture vortex formation and turbulence-chemistry interaction, but is too expensive for extensive parametric studies and its results do not show how to control thermoacoustic instabilities [30].

1.3 Integration of Adjoint Methods

The stability of thermoacoustic systems is highly sensitive to small changes in many parameters [5]. Knowing the response of the system to these changes would help the design process. One way of determining the sensitivities is to perform extensive parametric and experimental studies. For example, the boundary conditions, mean temperature gradient, mean Mach number, time delay and interaction index of thermoacoustic systems are examined through several parametric studies [31–33]. Regarding geometrical sensitivities, a database and

[4] https://github.com/JulHoltzDevelopers/WavesAndEigenvalues.jl

design strategies for suppressing acoustic oscillations by changing geometrical parameters are experimentally examined in [34] and [35] by focusing on the burner shape.

In addition to these, relatively cheap and accurate tools are desirable to examine the influence of design parameters. Adjoint methods achieve this at a low computation cost. This was first presented in [6] in which stabilizing mechanisms for a hot wire Rijke tube were quickly determined. Adjoints were then applied with a wave-based approach [36]. They have been used in low-order network models to stabilize annular [37] and longitudinal [38] combustors by changing their shapes. For more complex geometries, adjoint Helmholtz solvers are required, which increases the computational cost.

One of the first applications of an adjoint Helmholtz solver aimed to perform uncertainty quantification of the growth rates during thermoacoustic stability analysis [39]. Compared to Monte-Carlo simulations, calculation of eigenvalue drifts became much quicker via adjoint methods. This motivated the use of adjoint Helmholtz solvers on other thermoacoustic problems, e.g. investigating symmetry breaking effects [40], calculating thermoacoustic maps [41] and shape optimization [42, 7].

A comprehensive assessment of adjoint thermoacoustic Helmholtz solvers is outlined in [43] for 1D solvers with MATLAB scripts. Effective use of adjoint methods for design of thermoacoustic systems can be found in [44].

1.4 Shape Parametrization Tools

Parametrization of complex shapes poses a significant technical challenge in shape optimization. As the geometric complexity increases, the number of parameters increases, but the computational expense of the adjoint calculation does not increase because the gradient with respect to all parameters is found with a single calculation.

For complex-shaped geometries, the shapes need to be parametrized with, for example, B-Splines [7] and NURBS [45]. B-Splines are limited to line/edge parametrization. The combination of B-Splines gives NURBS surfaces. The NURBS representation facilitates the design process because it allows the manipulation of the entire surface thorough a small number of control points [46]. The design process with NURBS parametrization can be accelerated with adjoint methods. This procedure has been applied to optimize wing shapes and turbine blade geometry [47, 48].

When dealing with realistic geometries, the tool needs to be more descriptive and needs to manage both local and global parametrization across several components. Using FFD allows us to handle any geometry and allows us to control the degree of local or global parametrization complexity. This method was first introduced in [49] to increase the mod-

elling capabilities and the representation accuracy of the surfaces of solid bodies in a free-form manner. With a free-form lattice formed by a few control points, the FFD technique offers the sensitivity to local and global deformations of the embedded geometry through control point displacements. In addition to computer graphics, the effectiveness of FFD at handling complex geometries has made it a strong candidate to improve structural designs in industry. One of the first examples of FFD is found in aerodynamic shape optimization problems in [50] and has recently been applied to airfoil geometries with adaptive parametrization techniques [51]. For turbomachinery applications, FFD has been combined with adjoint methods in order to optimize a jet-engine fan blade [52]. In gas turbine engines, the applications of FFD can be found for compressor [53] and turbine [54] designs. Recently, FFD has been applied in thermoacoustics, stabilizing the first longitudinal mode of the Rijke tube and reducing the growth rate of the circumferential mode of an academic combustor [55]. Considering more complicated geometries, FFD parametrizes an industrial aeroengine combustor geometry with a handful of control points and the growth rate of the unstable axial mode is reduced by adjoint based shape optimization [56].

1.5 Thesis scope and structure

In this thesis, we extend the existing analysis on of shape sensitivity of thermoacoustic systems in [57] to more complicated geometries. We parallelize and optimize the existing Helmholtz solver [57] for quicker calculations. We also propose several parametrization techniques when dealing with realistic combustor geometries.

In Chapter 2, we present the open-source parallel adjoint Helmholtz solver, *helmholtz-x*, developed during this project. We start by obtaining the weak form of the thermoacoustic Helmholtz equation, then discretize it using FEM. We then provide the implementation details of the thermoacoustic eigenvalue problem including the models for choked and Bloch boundary conditions. Next, we demonstrate the accuracy of *helmholtz-x* by comparing the results against other experimental and numerical findings. We further present the parallelization capability of the solver by including the total execution times for the solution of the thermoacoustic eigenvalue problem for MICCA combustor.

In Chapter 3, we propose a surface parametrization technique, NURBS, and show examples, starting from a unit circle. We then extend it to a cylinder parametrization and eventually to the MICCA combustor. We show that the academic annular combustor can be parametrized through just a few control points. Further, these can be used to identify the shape sensitivity of the eigenvalues. Thanks to adjoint methods, the derivatives of the eigenvalues with respect to all the control points are found with a single calculation. We

then present shape sensitivity analysis for reducing the growth rate of the azimuthal and mixed eigenmodes. Results show the entire MICCA geometry being deformed through the NURBS control points in the direction provided by the shape derivatives. We found that the unstable eigenmodes became less stable after a few iterations. We present further analysis of symmetry breaking effects of the movements of the NURBS control points on the degenerate eigenvalues.

In Chapter 4, we present the free form deformation technique and its application to three-dimensional thermoacoustic cases. Unlike NURBS, free form deformation is not limited to surfaces but can also parametrize volume meshes with a control lattice formed by a few control points. We start with an application of the FFD technique to shape sensitivity analysis of the Rijke tube and LPP academic combustor. We reduce the growth rate of their unstable eigenmodes. Furthermore, we deploy the FFD parametrization to an industrial gas turbine combustor. We consider an unstable axial eigenmode and calculate the stabilizing FFD control point movements informed by the shape derivatives. After applying mesh deformation to the entire combustor geometry, we reduce the growth rate of the axial mode.

In Chapter 5, we conclude the findings in this thesis and propose future work. Appendix A contains a detailed derivation of the thermoacoustic Helmholtz equation used in this thesis. Appendix B provides a detail for the parallel matrix assembly that we used during the flame matrix construction.

Chapter 2

Open-source parallelized adjoint Helmholtz solver

In this chapter, we present a detailed description of the parallelized adjoint thermoacoustic Helmholtz solver, *helmholtz-x*. We show the derivation and FEM discretization of the thermoacoustic Helmholtz equation with the corresponding boundary conditions. We also include code snippets and show how to calculate eigenmodes. At the end of this chapter, we provide some verification cases for various thermoacoustic problems found in the literature.

Part of the content of this chapter has been submitted to *Engineering with Computers* for journal publication [58]. The established framework in this chapter is used to conduct studies presented in Chapter 3 and Chapter 4. I acknowledge Stefano Falco for providing the non-parallelized and not-optimized version of the thermoacoustic Helmholtz solver written in DOLFIN.

2.1 helmholtz-x

2.1.1 FEniCSx, PETSc and SLEPc

helmholtz-x is built upon the open-source FEM framework, FEniCSx and numerical toolkits PETSc and SLEPc. The FEniCSx project [59] offers a framework for solving PDEs using FEM. It has efficient matrix assembly kernels for reducing the solution time. The software also offers a scalable framework for computationally demanding problems with OpenMPI [60] parallelization through its Python interface, mpi4py [61].

We define weak forms of the PDEs through a high-level Python interface with the Unified Form Language (UFL) package [62]. UFL weak forms are assembled with the FEniCSx Form Compiler, FFCx [63], which transforms high-level Python codes into low-

level high performance C codes. Subsequently, the UFL compiled forms are assembled as sparse matrices. Matrices are formatted with the Portable Extensible Toolkit for Scientific Computation (PETSc) [64] so as to be compatible with MPI and to use a Python binding, petsc4py [65]. It also uses the open-source scalable and flexible toolkit for the solution of eigenvalue problems (SLEPc) [66], which solves eigenvalue problems of PETSc matrices, returning eigenvalues and their corresponding PETSc eigenvectors. SLEPc also offers Python binding through slepc4py [67]. FEniCSx, PETSc and SLEPc all support complex numbers.

2.1.2 Thermoacoustic Helmholtz Equation

The derivation of the direct and adjoint thermoacoustic Helmholtz equations follows the methodology in [57]. The direct Helmholtz equation and momentum equation are

$$\nabla \cdot \left(c^2 \nabla \hat{p}_1\right) + \omega^2 \hat{p}_1 = i\omega(\gamma - 1)\hat{q}_1 + c^2 \nabla \cdot \hat{f}_1 + c^2 i\omega \hat{m}_1 \tag{2.1a}$$

$$-i\rho_0 \omega \hat{\mathbf{u}}_1 + \nabla \hat{p}_1 = \hat{f}_1, \tag{2.1b}$$

where c is the spatially-varying speed of sound, $\hat{p}_1$ is the direct acoustic pressure, $\hat{\mathbf{u}}_\mathbf{1}$ is the acoustic velocity, ω is the complex valued angular frequency, γ is the heat capacity ratio, $\hat{q}_1$ is any fluctuating heat release rate, $\hat{f}_1$ is any fluctuating body force, $\hat{m}_1$ is any fluctuating mass injection, and p_0 is the mean pressure. Equation (2.1) can be written as $\mathscr{L}(\omega)\hat{p}_1 = 0$, where $\mathscr{L}$ is a differential operator that is linear in $\hat{p}_1$ but potentially nonlinear in ω. The detailed derivation of Eq. (2.1) is given in Appendix A. The property

$$\langle \hat{p}_1^\dagger | \mathscr{L} \hat{p}_1 \rangle = \langle \mathscr{L}^\dagger \hat{p}_1^\dagger | \hat{p}_1 \rangle + \text{boundary terms} = 0 \quad [44]$$

defines the adjoint Helmholtz and momentum equations [57] as

$$\nabla \cdot \left(c^2 \nabla \hat{p}_1^\dagger\right) + \omega^{*2} \hat{p}_1^\dagger = i\omega^*(\gamma - 1)\hat{q}_1(\omega^*) + c^2 \nabla \cdot \hat{f}_1 + i\omega^* c^2 \hat{m}_1, \tag{2.2a}$$

$$-i\rho_0 \omega^* \hat{\mathbf{u}}_1 + \nabla \hat{p}_1^\dagger = \hat{f}_1. \tag{2.2b}$$

where $\hat{p_1}^\dagger$ is the adjoint acoustic pressure and w^* is the complex conjugate of the angular eigenfrequency.

2.1.3 Source terms in the Helmholtz equation

We assume that the local heat release rate perturbation, q_1, is proportional to the acoustic velocity at a measurement point:

$$\frac{q_1(\mathbf{x},t)}{q_0} = \mathrm{FTF}\frac{\mathbf{u}_1(x_r)\cdot\mathbf{n}_r}{\mathbf{u}_b} \tag{2.3}$$

where q_0 is the mean heat release rate, FTF is the complex-valued flame transfer function, which depends on ω, u_b is the mean velocity and $\mathbf{n}_r$ is the unit normal vector in the reference direction. The fluctuating heat release rate, $\hat{q}_1$, is often modelled with a local $n-\tau$ formulation [68]. In Eq. (2.4), as in [43]:

$$\frac{q_1(\mathbf{x},t)}{q_0} = \frac{nh(\mathbf{x})\int_\Omega w(\mathbf{x})\mathbf{u}_1(\mathbf{x},t-\tau(\mathbf{x}))\cdot\mathbf{n}_r\mathbf{dx}}{u_b} \tag{2.4}$$

where n is the interaction index, $\tau(\mathbf{x})$ is the time delay, $h(\mathbf{x})$ is the heat release rate distribution and $w(\mathbf{x})$ is the measurement field. In the frequency domain, we can write Eq. (2.4) as

$$\hat{q}_1 = ne^{i\omega\tau}\int\frac{q_0}{u_b}h(\mathbf{x})w(\mathbf{x})\hat{\mathbf{u}}_1(\mathbf{x})\cdot\mathbf{n_r}\mathbf{dx} \tag{2.5}$$

The fields $h(\mathbf{x})$ and $\tau(\mathbf{x})$ can be obtained from experiments or simulations [27]. If $\tau(\mathbf{x})$ is uniform, we replace $ne^{i\omega\tau}$ with a complex-valued FTF. Without a fluctuating body force, Eq. (2.1b) becomes $\nabla\hat{p}_1 = i\omega\rho_0\hat{\mathbf{u}}_1$, so Eq. (2.5) becomes:

$$\hat{q}_1 = \mathrm{FTF}\frac{q_0}{u_b}h(\mathbf{x})\int\frac{w(\mathbf{x})}{i\omega\rho_0}\nabla\hat{p}_1\cdot\mathbf{n_r}\,\mathbf{dx} \tag{2.6}$$

We then obtain the thermoacoustic Helmholtz equation with a distributed measurement function:

$$\nabla\cdot\left(c^2\nabla\hat{p}_1\right)+\omega^2\hat{p}_1 = \mathrm{FTF}(\gamma-1)\frac{q_0}{u_b}h(\mathbf{x})\int\frac{w(\mathbf{x})}{\rho_0(\mathbf{x})}\nabla\hat{p}_1\cdot\mathbf{n_r}\,\mathbf{dx} \tag{2.7}$$

If w is a Dirac delta function, δ_D, then Eq. (2.7) becomes:

$$\nabla\cdot\left(c^2\nabla\hat{p}_1\right)+\omega^2\hat{p}_1 = \mathrm{FTF}(\gamma-1)\frac{q_0}{u_b}h(\mathbf{x})\frac{\nabla\hat{p}_1(\mathbf{x_r})\cdot\mathbf{n_r}}{\rho_0(\mathbf{x_r})} \tag{2.8}$$

We label the case where the FTF (i.e. n) is zero as the *passive flame* and the others as the *active flame*.

2.1.4 Finite Element Formulation

Discretization

Within the finite element framework, we integrate the terms in (2.1a) over the domain and multiply by a test function v to obtain

$$\begin{aligned}\int_\Omega \nabla\cdot\left(c^2\nabla\hat{p}_1\right)\, v\,\mathbf{dx}+\int_\Omega \omega^2\hat{p}_1\, v\,\mathbf{dx} &= \int_\Omega i\omega(\gamma-1)\hat{q}_1\, v\,\mathbf{dx}\\ &+\int_\Omega c^2\nabla\cdot\hat{f}_1\, v\,\mathbf{dx}\\ &+\int_\Omega c^2 i\omega\hat{m}_1\, v\,\mathbf{dx}.\end{aligned} \tag{2.9}$$

Note that the final two terms in Eq. (2.9) are only integrated over the domains in which the fluctuating body force $\hat{f}_1$ and $\hat{m}_1$ act, which are assumed to be negligible in this thesis. We use trial functions ϕ_i such that $\hat{p}_1=\sum_i \phi_i\cdot p_{1,i}$ and test functions ϕ_j such that $v=\sum_j\phi_j$. Integrating the terms in Eq. (2.9) by parts gives

$$\begin{aligned}&\sum_i\Big(\sum_j\big(-\int_\Omega c^2\nabla\phi_i\cdot\nabla\phi_j\,\mathbf{dx}+\int_{\partial\Omega}c^2\nabla\phi_i\cdot\mathbf{n}\,\phi_j\,\mathbf{dS}+\int_\Omega\omega^2\phi_i\,\phi_j\,\mathbf{dx}\big)p_{1,i}\Big)=\\ &\sum_i\Big(\sum_j\big(\int_\Omega i\omega(\gamma-1)\hat{q}_1\phi_j\,\mathbf{dx}\big)\Big),\end{aligned} \tag{2.10}$$

where $\mathbf{n}$ is the normal vector of the relevant boundary. The specific acoustic impedance [69], Z, is defined as

$$Z=\frac{\hat{p}_1}{\rho_0 c\hat{\mathbf{u}}_1\cdot\mathbf{n}} \tag{2.11}$$

Using Eq. 2.11, we can transform the second integral in Eq. (2.10) into the Robin integral using Eq. (2.1b) by writing

$$\int_{\partial\Omega}c^2\left(\nabla\phi_i\cdot\mathbf{n}\right)\,\phi_j\,dS=\int_{\partial\Omega}c^2\Big(\frac{iw}{cZ}\phi_i\Big)\,\phi_j\,dS \tag{2.12}$$

Hence, the matrix form of Eq. (2.10) is

$$\left[\mathbf{A}-\mathbf{D}(\omega)+\omega\mathbf{B}+\omega^2\mathbf{C}\right]\mathbf{p}=0 \tag{2.13}$$

where

$$\mathbf{A} = -\int_{\Omega} c^2 \nabla\phi_i \cdot \nabla\phi_j \, \mathbf{dx}, \tag{2.14a}$$

$$\mathbf{B} = \int_{\partial\Omega} \frac{ic}{Z} \phi_i \, \phi_j \, \mathbf{dS}, \tag{2.14b}$$

$$\mathbf{C} = \int_{\Omega} \phi_i \, \phi_j \, \mathbf{dx}, \tag{2.14c}$$

$$\mathbf{D} = \text{FTF} \, (\gamma - 1) \frac{q_0}{u_b} \int_{\Omega} \phi_j \, h(\mathbf{x}) \, \mathbf{dx} \int_{\Omega} \frac{w(\mathbf{x})}{\rho_0} \, \nabla\phi_i \cdot \mathbf{n}_r \mathbf{dx} \tag{2.14d}$$

and $\mathbf{p}$ is the direct eigenvector. In Eq. (2.14d), there is an outer product between the left integral and the right integral. We denote matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ the *acoustic* matrices and matrix $\mathbf{D}$ as the *flame* matrix.

To derive the adjoints of Eq. (2.14) in matrix form, we take the conjugate transpose[1] (^{H}) of Eq. (2.14) and calculate the right eigenvector, which is the adjoint eigenvector:

$$\left[\mathbf{A}^H - (\mathbf{D}(\omega))^H + \omega^* \mathbf{B}^H + \omega^{*2} \mathbf{C}^H\right] \mathbf{p}^\dagger = 0 \tag{2.15}$$

where

$$\mathbf{A}^H = -\int_{\Omega} c^2 \nabla\phi_j \cdot \nabla\phi_i \, \mathbf{dx}, \tag{2.16a}$$

$$\mathbf{B}^H = \int_{\partial\Omega} \frac{ic}{Z^*} \phi_j \cdot \phi_i \, \mathbf{dS}, \tag{2.16b}$$

$$\mathbf{C}^H = \int_{\Omega} \phi_j \cdot \phi_i \, \mathbf{dx}, \tag{2.16c}$$

$$\mathbf{D}^H = \text{FTF}^* (\gamma - 1) \eta \int_{\Omega} \frac{w(\mathbf{x})}{\rho_0} \, \nabla\phi_j \cdot \mathbf{n}_r \mathbf{dx} \int_{\Omega} \phi_i \, h(\mathbf{x}) \mathbf{dx} \tag{2.16d}$$

and $\mathbf{p}^\dagger$ is the adjoint eigenvector. Matrices $\mathbf{A}$ and $\mathbf{C}$ are self-adjoint [2] but matrix $\mathbf{B}^H$ is not self-adjoint if the specific impedance Z has a complex component. Matrix $\mathbf{D}^H$ is calculated by swapping the left and right vectors of the outer product in Eq. (2.14d) and replacing the FTF with its conjugate, FTF*.

Typical boundary conditions

There are three typical boundary conditions in acoustics: Dirichlet, Neumann, and Robin, all of which can be expressed through Eq. (2.11):

[1] superscript ^{H} is called Hermitian operator

[2] $\mathbf{A}^H == \mathbf{A}$ and $\mathbf{C}^H == \mathbf{C}$

1. For open boundaries (Dirichlet), $Z \to 0$ and $\hat{p}_1 = 0$.
2. For closed boundaries (Neumann), $Z \to \infty$ because $\hat{\mathbf{u}}_1$ is zero. So $\nabla \hat{p}_1 \cdot \mathbf{n} = 0$.
3. For other boundaries (Robin), Z is a finite complex number that quantifies acoustic radiation and phase shift at the boundary.
4. For choked boundaries, we define a reflection coefficient either for inlet or outlet. Then we impose them as Robin boundary conditions.

In *helmholtz-x*, Neumann boundaries are imposed naturally through the FEM discretization. For Dirichlet boundaries, DOF indices of the nodes on those surfaces are collected as a list, which we use to modify **A** and **C**. For Robin and its special cases choked inlet and choked outlet, we define the weak forms for these boundaries and use them to build **B**. For example, the UFL implementation for a Dirichlet boundary condition is shown in Listing 2.1.

```
1 if 'Dirichlet' in boundary_conditions[boundary]:
2     u_bc = Function(self.V)
3     facets = np.array(self.facet_tags.indices[self.facet_tags.values
    == boundary])
4     dofs = locate_dofs_topological(self.V, self.fdim, facets)
5     bc = dirichletbc(u_bc, dofs)
6     self.bcs_Dirichlet.append(bc)
```

Listing 2.1 Implementation of the Dirichlet boundary condition by adding degrees of freedom (DOFs) to the list *bcs_Dirichlet* so that the boundaries are added to the matrices **A** and **C** during assembly.

Z can be imposed on Robin boundaries[3] through the reflection coefficients R with Eq. (2.14b). The UFL implementation of a Robin boundary condition is shown in Listing 2.2.

```
1 if 'Robin' in boundary_conditions[boundary]:
2     R = boundary_conditions[boundary]['Robin']
3     Z = (1+R)/(1-R)
4     integrals_Impedance = 1j * self.c / Z * inner(self.phi_i, self.
    phi_j) * self.ds(boundary)
5     self.integrals_R.append(integrals_Impedance)
```

Listing 2.2 Robin boundary condition implementation. The weak form in line 4 is identical to Eq. (2.14b). We add the contributions of the Robin boundaries to the list *integrals_R*.

[3] $Z = (1+R)/(1-R)$

In thermoacoustics, most of the boundaries are assumed to be Neumann or choked boundary conditions. The reflection coefficient of the inlet choked boundary condition is [12]

$$R_{in} = \frac{1 - \gamma_{in} M_{in} / (1 + (\gamma_{in} - 1) M_{in}^2)}{1 + \gamma_i M_{in} / (1 + (\gamma_{in} - 1) M_{in}^2)}, \tag{2.17}$$

where γ_{in} is the heat capacity ratio on the inlet choked boundary and M_{in} is the Mach number near the downstream of the inlet choked boundary. The UFL implementation of the choked inlet boundary condition is shown in Listing 2.3.

```
1  if 'ChokedInlet' in boundary_conditions[boundary]:
2      A_inlet = MPI.COMM_WORLD.allreduce(assemble_scalar(form(self.
       AreaConstant * self.ds(boundary))), op=MPI.SUM)
3      gamma_inlet_form = form(self.gamma/A_inlet* self.ds(boundary))
4      gamma_inlet = MPI.COMM_WORLD.allreduce(assemble_scalar(
       gamma_inlet_form), op=MPI.SUM)
5
6      Mach = boundary_conditions[boundary]['ChokedInlet']
7      R = (1-gamma_inlet*Mach/(1+(gamma_inlet-1)*Mach**2))/(1+
       gamma_inlet*Mach/(1+(gamma_inlet-1)*Mach**2))
8      Z = (1+R)/(1-R)
9      integral_C_i = 1j * self.c / Z * inner(self.phi_i, self.phi_j) *
       self.ds(boundary)
10     self.integrals_R.append(integral_C_i)
```

Listing 2.3 Choked inlet boundary condition implementation. Lines 2 to 4 calculate the average γ on the choked boundary. Line 7 calculates the reflection coeffient for the choked inlet boundary using the near-upstream Mach number of the flow. Line 9 implements the Robin boundary condition with the specific impendance Z calculated in line 8. Finally the implemented Robin BC is added to the Robin integrals list in line 10.

Similarly, we write the choked outlet condition [12]

$$R_{out} = \frac{1 - (\gamma_{out} - 1) M_{out} / 2}{1 + (\gamma_{out} - 1) M_{out} / 2}, \tag{2.18}$$

where γ_{out} is the heat capacity ratio on the outlet choked boundary and M_{out} is the Mach number near the upstream of the outlet choked boundary. The UFL of the choked outlet boundary condition is implemented by changing R, γ and Ma in Listing 2.3

2.1.5 Implementation of the acoustic matrices

We generate UFL for the matrices using the relations in Eq. (2.14). The UFL code of the assembly process is shown in Listing 2.4.

```
1  # Matrix A
2  self.a_form = form(-self.c**2* inner(grad(self.phi_i), grad(self.
       phi_j))*self.dx)
3  A = assemble_matrix(self.a_form, bcs=self.bcs_Dirichlet)
4  A.assemble()
5  self._A = A
6
7  # Matrix B
8  if self.integrals_R:
9      self.b_form = form(sum(self.integrals_R))
10     B = assemble_matrix(self.b_form)
11     B.assemble()
12     B_adj = B.copy()
13     B_adj.transpose()
14     B_adj.conjugate()
15     self._B = B
16     self._B_adj = B_adj
17
18 # Matrix C
19 self.c_form = form(inner(self.phi_i , self.phi_j) * self.dx)
20 C = assemble_matrix(self.c_form, self.bcs_Dirichlet)
21 C.assemble()
22 self._C = C
```

Listing 2.4 UFL for construction of the acoustic matrices. Lines 2 and 19 represent Eqs. (2.14a) and (2.14c). In lines 3 and 20, the list bcs_Dirichlet in Listing 2.1 is used to impose Dirichlet boundary conditions. Lines 8 to 16 construct matrix **B** in Eq. (2.14) containing Robin boundary conditions using the list integrals_R.

2.1.6 Implementation of the flame matrix

The flame matrix, **D** contains an outer product between two sparse vectors, which requires careful implementation. The relation

$$(\gamma-1)\frac{q_0}{u_b}\int_\Omega \phi_j h(\mathbf{x})\,\mathbf{dx}\int_\Omega \frac{w(\mathbf{x})}{\rho_0}\nabla\phi_i\cdot\mathbf{n}_r\mathbf{dx} \tag{2.19}$$

is shared between Eq. (2.14d) and (2.16d). For computational efficiency, we first perform the calculation of this cross product and compute the direct and adjoint submatrices, $\mathbf{D}_{ij}$ and $\mathbf{D}_{ji}$. Then we multiply the submatrices with FTF or $(\text{FTF})^*$, to obtain the direct or adjoint $\mathbf{D}$. The left and right components of Eq. (2.19) are calculated separately during assembly:

$$\underbrace{(\gamma-1)\frac{q_0}{u_b}\int_\Omega \phi_j h(\mathbf{x})\,\mathbf{dx}}_{\text{left vector}}\underbrace{\int_\Omega \frac{w(\mathbf{x})}{\rho_0}\nabla\phi_i\cdot\mathbf{n}_r\mathbf{dx}}_{\text{right vector}} \tag{2.20}$$

The left and right vectors in Eq. (2.20) are swapped when generating the adjoint $\mathbf{D}$.

In *helmholtz-x*, two different flame matrices are implemented: one for a distributed measurement function $w(\mathbf{x})$ and the other for a pointwise measurement function $w(\mathbf{x}_r)$. The right vector of Eq. (2.20) is implemented differently for the pointwise flame matrix, as explained in Sec. 2.1.6. For the distributed flame matrix, Eq. (2.20) remains the same (Sec. 2.1.6). Appendix B explains how parallel matrix data is handled through custom MPI utility functions.

Distributed measurement function

The measurement region $w(\mathbf{x})$ can take any shape. We choose a truncated Gaussian distribution that integrates to 1 (Fig. 2.6). This distribution introduces more nonzero contributions to the flame matrix, so is less sparse than the pointwise flame matrix.

In *helmholtz-x*, we use distributed $w(\mathbf{x})$ for longitudinal combustors and pointwise $w(\mathbf{x})$ for annular combustors. Although distributed $w(\mathbf{x})$ can be used for annular combustors, it requires a lot of memory. The UFL forms of the left and right vectors are defined in Listing 2.5.

```
self.left_form = form((gamma - 1) * q_0 / u_b * self.phi_i * h *  dx)
self.right_form = form(inner(self.n_r,grad(self.phi_j)) / rho * w *
    dx)
```

Listing 2.5 UFL forms for left and right vectors shown in Eq. (2.20).

Then we generate sparse vectors and store their nonzero (index, value) pairs for both vectors in Listing 2.6.

```
1  def _assemble_vectors(self, problem_type='direct'):
2      left_vector = self.indices_and_values(self.left_form)
3      right_vector = self.indices_and_values(self.right_form)
4      if problem_type == 'direct':
5          left_vector = distribute_vector_as_chunks(left_vector)
6          right_vector = broadcast_vector(right_vector)
7      elif problem_type == 'adjoint':
8          right_vector = distribute_vector_as_chunks(right_vector)
9          left_vector = broadcast_vector(left_vector)
10     return left_vector, right_vector
```

Listing 2.6 Extracting nonzero values from the left (line 2) and right (line 3) vectors. If the problem is direct, the right vector is replicated over the processors and the left vector is distributed evenly. For the adjoint problem, the procedure is reversed.

Pointwise measurement function

The pointwise measurement function has a nonzero contribution only at the measurement point(s) ($\mathbf{x}_r$). We use this for annular combustors, where multiple pairs of $w(\mathbf{x}_r)$ and $h(\mathbf{x}_f)$ exist. We calculate the pointwise values of the gradient of the trial function $\nabla\phi_i$ near the points $\mathbf{x}_r$ such that Eq. (2.19) becomes

$$\underbrace{(\gamma-1)\frac{q_{0_f}}{u_b}\int_\Omega \phi_j\, h(\mathbf{x})\,\mathbf{dx}_f}_{\text{left vector}}\underbrace{\int_\Omega \frac{\nabla\phi_i(\mathbf{x}_{r_f})\cdot\mathbf{n}_r}{\rho_0(\mathbf{x}_{r_f})}\mathbf{dx}_f}_{\text{right vector}} \tag{2.21}$$

where subscript $_f$ represents the relevant flame index. If there are N discrete sectors in the annular combustor, there are N measurement points and heat release rate distributions. We find the contributions to $\mathbf{D}$ of the corresponding flame and its measurement point iteratively. We access the nonzero data through the subscript $_f$. In *helmholtz-x*, each individual $h(\mathbf{x})$ integrates to 1 over the domain[4]. In addition, the heat release rate volumes are tagged as separate subdomains starting from 0 to $N-1$. These tags are used in *helmholtz-x* during assembly (Fig. 2.1). The algorithm to obtain the nonzero data for the pointwise flame matrix is in Listing 2.7.

[4]The integral $\int_\Omega h(\mathbf{x})\mathbf{dx} = N$ over the domain and we input the heat release rate q_0 per single flame ($q_0 N = q_{total}$ where q_{total} is the total power of the annular combustor).

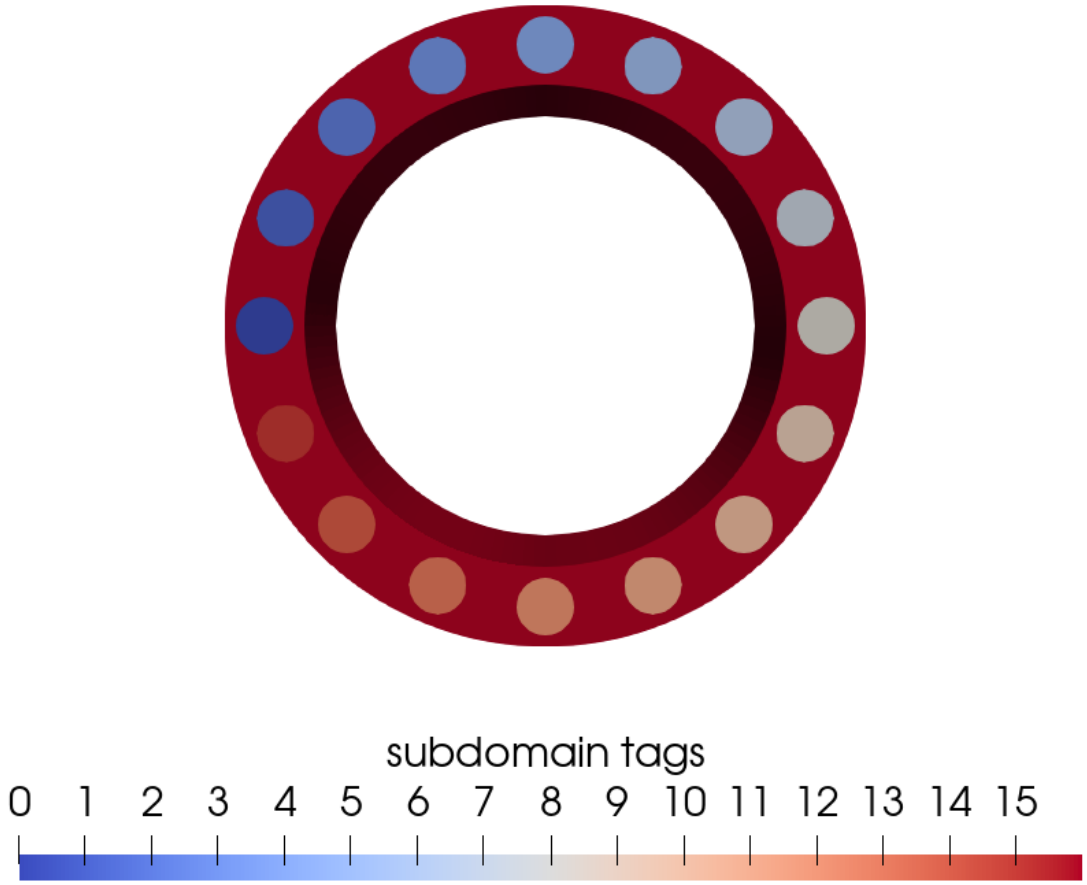


Fig. 2.1 Subdomain tagging for the annular combustor. The numbering runs from 0 to 15 because there are 16 flame volumes in total within the heat release rate function $h(\mathbf{x})$.

```
1  def _assemble_vectors(self, flame, point):
2      left_form = form((self.gamma - 1) * self.q_0 / self.u_b * inner(
       self.h, self.phi_j)*self.dx(flame))
3      left_vector = self.indices_and_values(left_form)
4      _, _, owning_points, cell = determine_point_ownership( self.mesh.
       _cpp_object, point, 1e-10)
5      right_vector = []
6      if len(cell) > 0: # Only add contribution if cell is owned
7          cell_geometry = self.mesh.geometry.x[self.mesh.geometry.
       dofmap[cell[0]], :self.gdim]
8          point_ref = self.mesh.geometry.cmaps[0].pull_back([point],
       cell_geometry)
9          right_form = Expression(inner(grad(TestFunction(self.V)),
       self.n_r), point_ref, comm=MPI.COMM_SELF)
10         dphij_x_rs = right_form.eval(self.mesh, cell)[0]
11         right_values = dphij_x_rs / self.rho_u
12         global_dofs = self.dofmaps.index_map.local_to_global(self.
       dofmaps.cell_dofs(cell[0]))
13         for global_dof, right_value in zip(global_dofs, right_values)
       :
14             right_vector.append([global_dof, right_value ])
15     right_vector = broadcast_vector(right_vector)
16     return left_vector, right_vector
```

Listing 2.7 Code to calculate the nonzero data for the pointwise flame matrix. The parameters *flame* and *point* represent the flame tag and its measurement point (line 1). Line 2 is identical to the left vector of Eq. (2.21). For the left vector data, we use only the corresponding flame subdomain ($\mathbf{dx}_f$) during integration and we calculate its nonzero data in line 3. Line 9 is identical to the right vector of Eq. (2.21). We calculate the value of the gradient of the test function at the measurement point (line 11) and find the DOFs of the cell that includes the measurement point (line 12). We store the global indices of the DOFs of the cell and construct the (col index, value) pairs of the right vector (lines 13 and 14). Finally, we copy the data of the right vector over the processors for parallel pointwise $\mathbf{D}$ generation (line 15).

2.1.7 Bloch boundary condition

If the computational domain has an N-fold discrete rotational symmetry, the circumfrential eigenmodes can be calculated by repeating a single geometry N times [70], first implemented in thermoacoustics by [18, 19]. For this, we apply Bloch-type boundary conditions to the relevant (master and slave) boundaries. This boundary condition reduces the computation load by 2^N times. *helmholtz-x* follows the methodology presented in [19]. According to

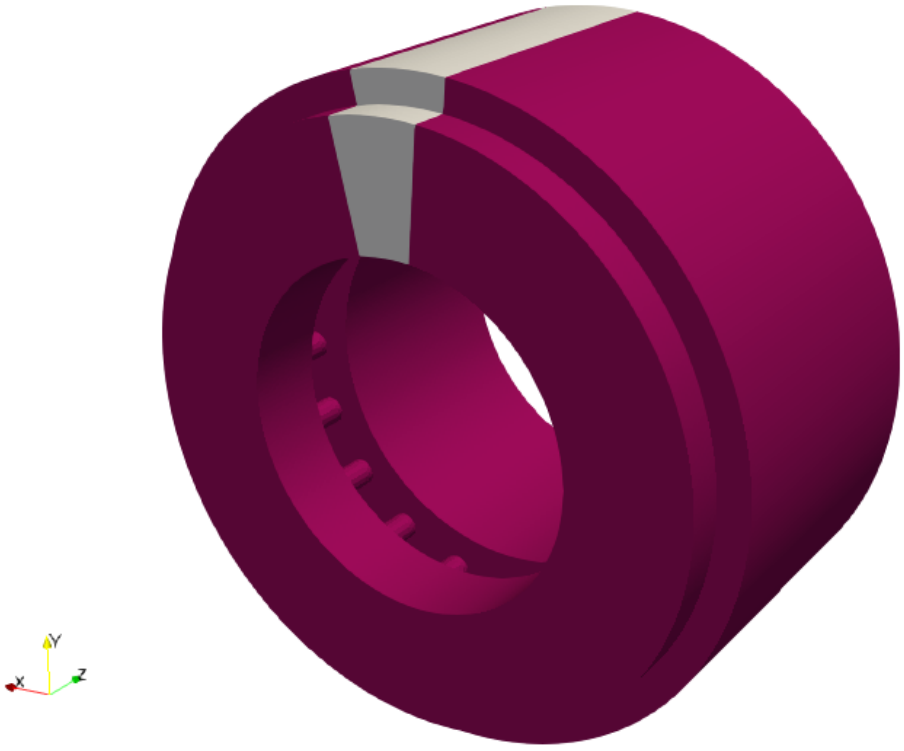


Fig. 2.2 Example annular combustor geometry. The gray section represents a single sector out of 20 identical sectors. With Bloch boundary condition, the azimuthal eigenmodes can be calculated by considering only the gray section.

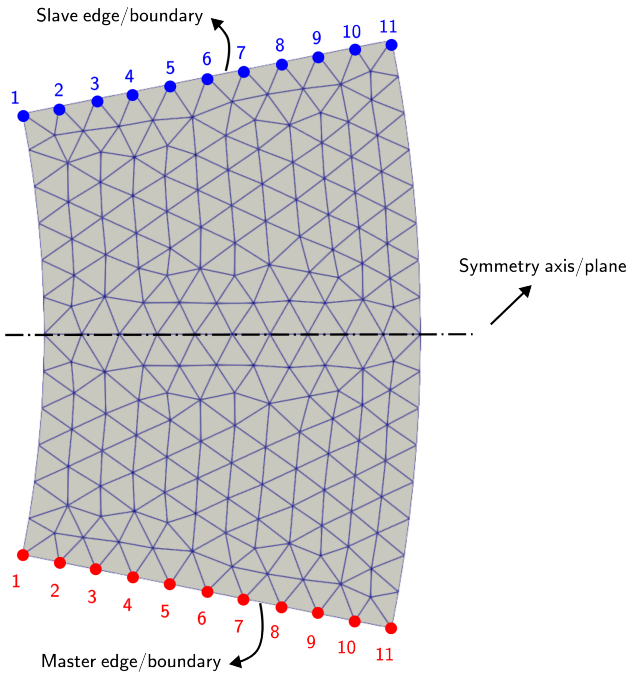


Fig. 2.3 Example mesh for Bloch BC application. The DOFs of the blue (slave) nodes should be paired with the DOFs of the red (master) nodes according to the numbers (from 1 to 11 in this example). The half-sector mesh is then reflected with respect to the symmetry axis/plane in order to guarantee one-to-one DOF mapping.

Bloch-wave theory, the acoustic wave can be expressed by:

$$\hat{p}_b(\phi,r,z) = \hat{p}_{b+N}(\phi,r,z)e^{ib\phi}(\phi,r,z) \tag{2.22}$$

where ϕ, r and z are angular coordinates and b is the Bloch wavenumber. In *helmholtz-x*, the Bloch boundary condition is implemented as a Dirichlet (essential) boundary condition [19]. We map the matching nodes between master and slave boundaries (Fig. 2.3) with Eq. (2.22). When Bloch boundary conditions are applied, the number of DOFs in the mesh reduces. This requires manipulation of the matrices in Eq. (2.14) such that the DOFs of the slave boundary and its entries are deleted, and a periodicity scalar $f_b = e^{ib2\pi/N}$ is imposed on the master facets. The eigenmode of the system is found with these matrices. Then the slave DOFs are added back to the eigenvector.

2.1.8 Fixed-point Iteration

The nonlinear eigenvalue problem consists in finding the eigenvalues ω in complex function space and the non-zero eigenvectors $\mathbf{p}$ in Eq. (2.13). We reformulate Eq. (2.13) as

$$\mathbf{L}(\omega)\mathbf{p} = 0 \tag{2.23}$$

in which linear operator $\mathbf{L}$ changes nonlinearly with ω because of $\mathbf{D}(\omega)$. Hence, it should be represented as a generalized nonlinear eigenvalue problem and solved iteratively;

$$\left[(\mathbf{A} - \mathbf{D}(\omega_{k-1})) + \omega_k \mathbf{B} + \omega_k^2 \mathbf{C}\right] \mathbf{p} = 0 \tag{2.24}$$

where k is the iteration number. Nicoud's fixed point iteration is well-suited to this nonlinear eigenvalue problem[71]. A short description of fixed point iteration is presented as pseudocode in Algorithm 1.

function `FixedPointIteration`(**A**, **B**, **C**, **D**, *tol*, *maxiter*)**:**
- $k \longleftarrow -1$;
- $\alpha^0 \longleftarrow 0.5$;
- $\omega^{|k|} \longleftarrow 0$;
- Determine $\omega^{|k+1|}$ using **A**, **B**, **C**;
- $\Delta\omega \longleftarrow 2 \times tol$;
- **while** $|\Delta\omega| > tol$ ***and*** $k < maxiter$ **do**
  - $k \longleftarrow k + 1$;
  - Assemble $\mathbf{D}(\omega^{|k|})$;
  - $\mathbf{D} \longleftarrow \mathbf{A} - \mathbf{D}(\omega^{|k|})$;
  - Determine $\omega^{|k+1|}$ using **D**, **B**, **C**;
  - **if** $k \neq 0$ **then**
    - $\alpha^k \longleftarrow \frac{1}{(1-(\omega^{|k|}-\omega^{|k+1|}))(\omega^{|k|}-\omega^{|k+1|})}$
  - **end**
  - $\omega^{|k+1|} = \alpha^k * \omega^{|k|} + (1 - \omega^{|k|}) \times \omega^{|k|}$;
  - $|\Delta\omega| \longleftarrow \left|\omega^{|k+1|} - \omega^{|k|}\right|$;
- **end**
- **return** $\omega^{[k+1]}$, **p**;

end

Algorithm 1: Fixed-point iteration

2.1.9 Software Structure

helmholtz-x follows the philosophies of scalability, readibility, reproducibility, evolvability and maintainability. *helmholtz-x* heavily exploits the principles of object oriented programming (OOP). The typical simulation flow is visualized in Fig. 2.4, in which the submodules of *helmholtz-x* and their functionalities are classified. The source code of *helmholtz-x* can be found in the `helmholtz_x` directory in the repository[5].

Pipeline of helmholtz-x

In this section, we describe the *helmholtz-x* utilities step by step.

Mesh, subdomains, and facets: We first need to generate the mesh, subdomains and facets. If flames are included, we need to define the flame volume subdomains during mesh generation. These subdomains are labeled from 0 to $N-1$. We also tag facets to impose boundary conditions.

For simple geometries, DOLFINX provides built-in functions for mesh, subdomains and facets. We import these using:

```
n_elem = 3000 # number of elements in 1D mesh
mesh, subdomains, facet_tags = OneDimensionalSetup(n_elem)
```

For complex geometries, we use the open source finite element mesh generator Gmsh [72]. Gmsh can generate grids for .step files through its Python API and OCC kernel. For Gmsh meshes, we transform the generated grids into the XDMF format for consistency with DOLFINx modules. The following lines read an XDMF mesh with its subdomains and facets tags:

```
geometry = XDMFReader("PathForMesh")
mesh, subdomains, facet_tags = geometry.getAll()
```

There are several examples with different grids in the */numerical_examples* folder.

Assembling the acoustic matrices: We define the parameters for acoustic matrices **A**, **B** and **C** with a standalone *params.py* file that is imported. First we define boundary conditions by specifying facet tags as a Python dictionary, for example:

```
boundary_conditions = {1:{"Dirichlet"},
                       3:{"ChokedInlet":params.M0},
                       8:{"ChokedOutlet":params.M1},
                       11:{"Robin":params.R}}
```

where the numbers represent the corresponding Gmsh tags of each boundary condition. The choked inlet and choked outlet boundaries take the Mach number near the boundaries. Robin

[5]https://github.com/ekremekc/helmholtz-x/tree/paper

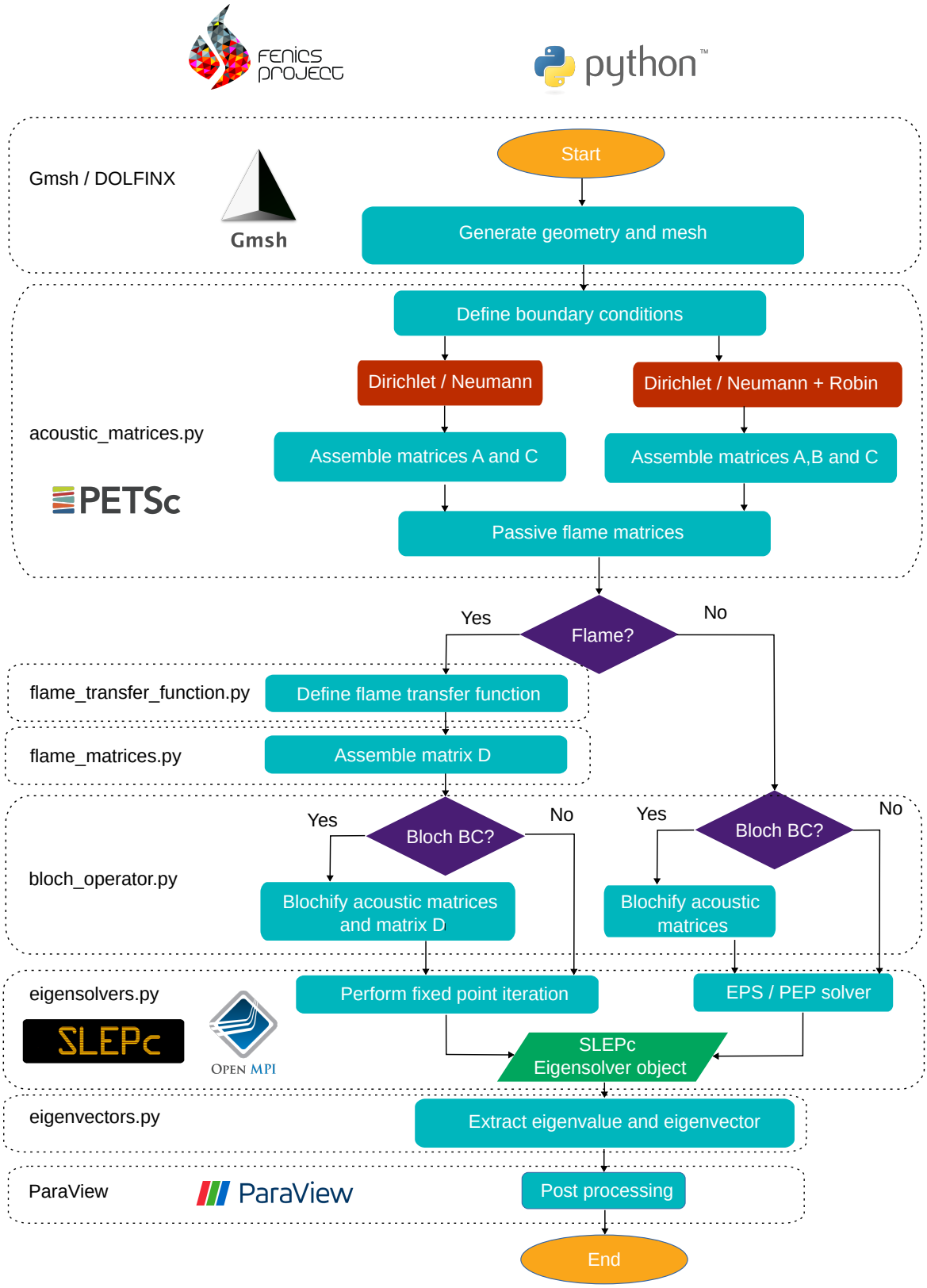


Fig. 2.4 The components of *helmholtz-x* and the flowchart for the solution of the inhomogeneous thermoacoustic Helmholtz equation.

boundaries are specified with their reflection coefficients. We input the speed of sound or temperature field to construct the acoustic matrices:

```
c = params.c(mesh)
matrices = AcousticMatrices(mesh, facet_tags, boundary_conditions, c,
    degree=degree)
```

where the *degree* represent the polynomial degree of basis functions on the finite element space. The parameter c is the speed of sound. The *AcousticMatrices* class can also take temperature as a parameter and convert it to the speed of sound [6].

Defining the flame transfer function: If we solve the inhomogeneous Helmholtz equation, matrix **D** needs to be implemented, which requires an FTF. *helmholtz-x* has two different FTFs: the $n-\tau$ formulation or the state space representation (from an experimental FTF). These can be defined by using *nTau* or *stateSpace* classes as

```
FTF = nTau(params.n, params.tau)
FTF = stateSpace(params.S1, params.s2, params.s3, params.s4)
```

by importing the necessary parameters from *params.py*.

Assembling the flame matrix: For distributed **D**, we define the parameters of Eq. (2.19) and input them to the *DistributedFlameMatrix* class with;

```
rho = rho_step(mesh, params.x_f, params.a_f, params.rho_d, params.
   rho_u)
w = gaussianFunction(mesh, params.x_r, params.a_r)
h = gaussianFunction(mesh, params.x_f, params.a_f)
FTF = nTau(params.n, params.tau)
D = DistributedFlameMatrix(mesh, w, h, rho, T, params.q_0, params.u_b
   , FTF, degree=degree)
D.assemble_submatrices()
```

where the function *D.assemble_submatrices()* takes two parameters, 'direct' (by default) or 'adjoint'. For implementing pointwise **D**, we import the necessary parameters of Eq. (2.21) and use them in the *PointwiseFlameMatrix* class by, for example:

```
h = Q_multiple(mesh, subdomains, params.N_sector)
D = PointwiseFlameMatrix(mesh, subdomains, params.x_r, h, params.
   rho_xr, params.q_0, params.u_b, FTF, degree=degree)
D.assemble_submatrices()
```

Imposing Bloch boundary conditions: If there are Bloch boundaries, we define them in the *boundary_conditions* dictionary. For this, we specify slave and master boundaries with their physical tags such as

[6] $c = \sqrt{\gamma r_{gas} T_0}$

```
boundary_conditions = {11: {'Robin':params.R_outlet},
                       12: 'Master',
                       13: 'Slave'}
```

Then we manipulate the matrices in the system with

```
bloch_matrices = Blochifier(geometry, boundary_conditions, N,
    acoustic_matrices)
D = PointwiseFlameMatrix(mesh, subdomains, params.x_r, h, params.
    rho_amb, params.q_0, params.u_b, FTF, degree=degree, bloch_object=
    bloch_matrices)
D.blochify()
```

where N is the Bloch number. The flame matrix classes take the post-Bloch matrices as a bloch_object parameter to create $\mathbf{D}$.

Solving the system: If the Helmholtz equation is homogeneous, we use the EPS solver such that

```
target = 200 * 2 * np.pi
E = eps_solver(matrices.A, matrices.C, target, nev=2, print_results=
    True)
```

or, if we have Robin boundaries, the PEP solver such that

```
target_dir = 262 * 2 * np.pi
E = pep_solver(matrices.A, matrices.B, matrices.C, target_dir, nev
    =10, print_results=True)
```

In *helmholtz-x*, the unit of the target eigenvalue is rad s^{-1}. We converge to the targeted angular eigenfrequency. If the problem is inhomogeneous, we have $\mathbf{D}$ and we use fixed point iteration (or a Newton solver) such that

```
target = 200 * 2 * np.pi
E = fixed_point_iteration(matrices, D, target, nev=2, i=0,
    print_results= False)
```

All these functions return a SLEPc object E, from which we extract eigenvalues and eigenvectors.

Extracting the eigenvalues and eigenvectors: In *helmholtz-x*, we normalize the eigenvectors such that $\int \hat{p}_1^2 = 1$. When the object E is computed, it has eigenvalue ω and eigenvectors $\mathbf{p}$ and $\mathbf{p}^\dagger$ as instances. We extract them with the *normalize_eigenvector* function with

```
omega, p = normalize_eigenvector(mesh, E, i=0, degree=degree, which='
    right')
```

where the parameter i is the index of the converged eigenvalue and the keyword *which* decides whether to return the right or left eigenvector.

Saving the eigenvector and eigenvalue: We save the eigenvector and its eigenvalue with *xdmf_writer* and *dict writer* functions

```
# Save eigenvectors
xdmf_writer("PathToWrite", mesh, p)
# Save eigenvalues
omega_dict = {'direct':omega}
dict_writer("PathToWrite", omega_dict)
```

We then visualize the resulting XDMF file with the open-source visualization toolkit ParaView [73].

Parallelization

Any calculation using *helmholtz-x* can be parallelized with the command *mpirun -np n_proc python3 -u file.py* where *n_proc* specifies the number of processors and *file.py* is the Python script to be parallelized. After writing the *params.py* and main scripts following the pipeline (Sec. 2.1.9), *helmholtz-x* handles the parallelization internally. All computations in this section are performed using hardware with Intel(R) Xeon(R) E5-2620 v4 2.10GHz x 16 processors and 32GB memory.

2.2 Solver verification

In this section, we present several test cases with *helmholtz-x* for longitudinal and annular geometries and compare them against other numerical tools in the literature.

2.2.1 Longitudinal thermoacoustic systems

We first verify the results of *helmholtz-x* in relatively simple cases: a hot wire Rijke tube, a longitudinal combustor, and an industrial network model.

Hot wire Rijke tube

In this section, 1D, 2D and 3D test cases of the hot wire Rijke tube are implemented. The code is presented in `numerical_examples/Longitudinal/NetworkCode/RijkeTube*` folders in the repository. The schematic representation of the hot wire Rijke tube case is shown in Fig. 2.5.

We define $w(\mathbf{x})$ (Fig. 2.6) and $h(\mathbf{x})$ as multi-dimensional Gaussian distributions (Eq (2.25)) in which n_{dim} is the spatial dimension and σ controls the width of the Gaussian

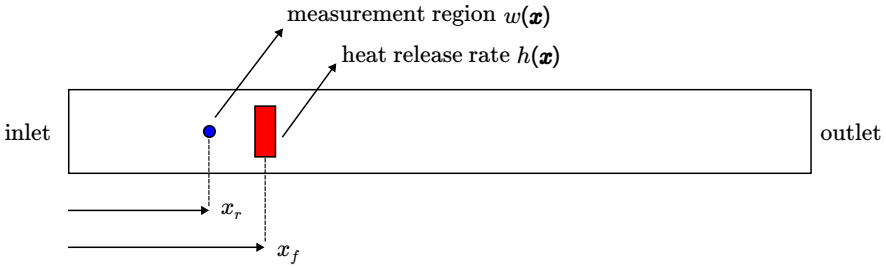


Fig. 2.5 Schematic representation of the Rijke tube. We implement $w(\mathbf{x})$ and $h(\mathbf{x})$ with Gaussian functions.

around its central point $P(x_0, y_0, z_0)$:

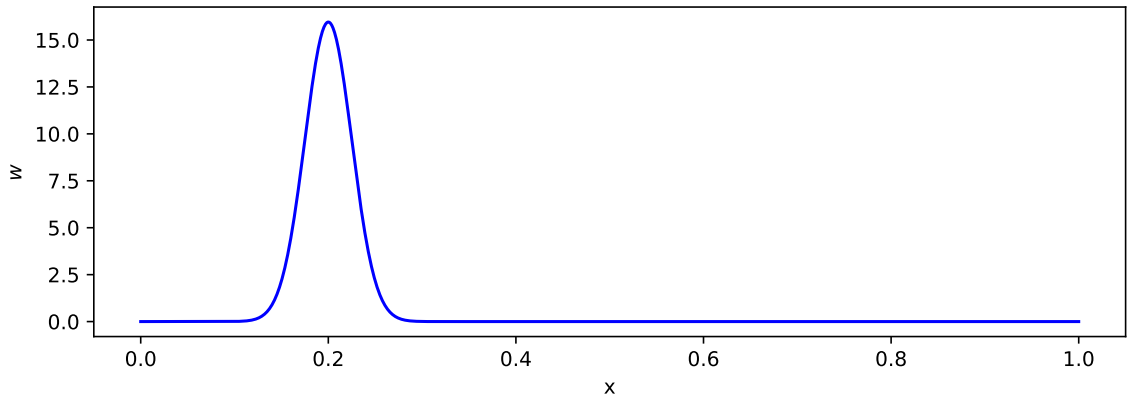
$$G(\mathbf{x}) = \frac{1}{\sigma^{n_{dim}}(2\pi)^{n_{dim}/2}} \exp\left(-\frac{(x-x_0)^2+(y-y_0)^2+(z-z_0)^2+...}{2\sigma^2}\right) \tag{2.25}$$

We use $w(\mathbf{x})$ and $h(\mathbf{x})$ fields *DistributedFlameMatrix* class during flame matrix construction. The parameters of the Helmholtz solver are tabulated in Table 2.1.

Table 2.1 Dimensional parameters of the hot wire Rijke tube.

Parameter	value	unit
L	1	m
d	0.047	m
r_{gas}	287.1	$\mathrm{Jkg^{-1}K^{-1}}$
p_0	101325	Pa
ρ_u	1.22	$\mathrm{kg\ m^{-3}}$
ρ_d	0.85	$\mathrm{kg\ m^{-3}}$
T_u	285.6	K
T_d	409.92	K
q_0	-27.0089	W
u_b	0.1006	$\mathrm{m\ s^{-1}}$
n	0.1	-
τ	0.0015	s
x_f	0.25	m
a_f	0.025	-
x_r	0.2	m
a_r	0.025	-

The speed of sound field is calculated from the temperature distribution. The interaction index n is scaled by dividing by the cross-sectional area of the tube ($\pi d^2/4$) for the 1D case and by $\pi d/4$ for the 2D case. The 3D case does not require scaling. For calculation of $\gamma =$



(a)

(b)

(c)

Fig. 2.6 1D (a), 2D (b) and 3D (c) examples of the measurement region $w(\mathbf{x})$ with $\sigma = 0.025$.

$c_p/c_v = c_p/(c_p - r_{gas})$, a linear temperature dependence of $c_p(T) = 973.60091 + 0.1333T$ is used for both the Helmholtz solver and the network model. For simplicity, all boundaries are assumed to be Neumann. Passive and active flame simulations are performed. Their eigenfrequencies are tabulated in Table. 2.2. The acoustic pressure and acoustic velocity eigenfunctions are shown in Fig. 2.7.

Table 2.2 Eigenfrequencies of the passive and active flame test cases for the Rijke tube. GR denotes the growth rate. The eigenfrequencies become closer as the grid resolutions of *helmholtz-x* increases.

Run	Passive		Active	
	f (1/s)	GR (rad/s)	f (1/s)	GR (rad/s)
Network code	169.178074	0.	197.784121	6.411332
1D *helmholtz-x*	169.377645	0.	197.699903	6.683160
2D *helmholtz-x*	169.377337	0.	197.762459	6.668631
3D *helmholtz-x*	169.410068	0.	198.577709	6.797977

We also present tests to check the adjoint capability of *helmholtz-x* by replicating the results in [43]. The *helmholtz-x* codes for these cases are in the `numerical_examples/Longitudinal/PRF` folder in the repository. The parameters are in Table 2.3.

We run test cases for multidimensional configurations (as in Sec. 2.2.1) with the nondimensionalized system. The inlet and outlet boundaries are Robin, with reflections coefficients tabulated in Table. 2.3. The resulting eigenvalues are in Table 2.4. We also show the direct and adjoint pressure eigenfunctions in Fig. 2.8.

Flame in a cylindrical duct with choked boundaries

In this case, we use a geometry that has area changes and choked boundaries at both ends. The code is implemented in `numerical_examples/Longitudinal/NetworkCode/FlamedDuct` folder in the repository. A Schematic representation of this test case is shown in Fig. 2.9.

This example is useful to check the Helmholtz solver's ability to capture the influence of the area change and acoustic energy losses through the choked boundaries. The parameters of this test are given in Table 2.5.

The density field ρ_0 is calculated from the ideal gas equation of state, $p_0 = \rho_0 r_{gas} T_0$ using the temperature field. The Mach number near the inlet is $M_{in} = 0.0092$ and near the outlet is $M_{out} = 0.011$. The heat release rate and measurement region fields for this case are shown in Fig. 2.10.

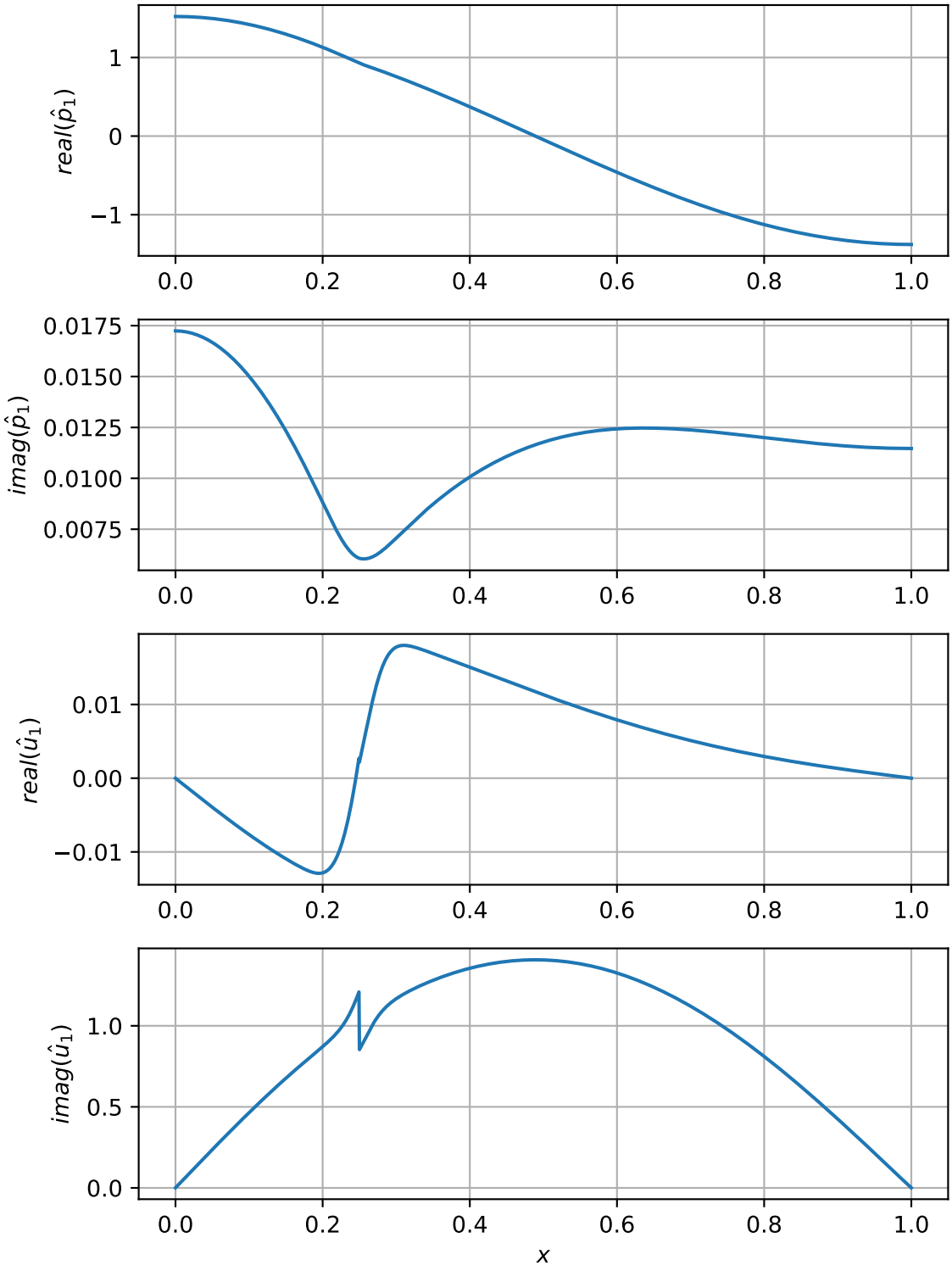


Fig. 2.7 1D acoustic pressure and acoustic velocity eigenmodes for the Rijke tube.

Table 2.3 Dimensional & non-dimensional parameters of the hot wire Rijke tube taken from [43]. The interaction index n changes for 1D and 2D for dimensional consistency.

Parameter	value	unit
L	1	m
d	0.047	m
r_{gas}	287.1	$\mathrm{Jkg^{-1}K^{-1}}$
p_0	100000	Pa
ρ_u	1.22	$\mathrm{kg\ m^{-3}}$
ρ_d	0.85	$\mathrm{kg\ m^{-3}}$
q_0	200	W
u_b	0.1	$\mathrm{m\ s^{-1}}$
n	1.4e-7	-
τ	0.0015	s
R_{in}	-0.975 -0.05i	-
R_{out}	-0.975 -0.05i	-
x_f	0.25	m
a_f	0.025	-
x_r	0.2	m
a_r	0.025	-

Table 2.4 Eigenfrequencies of the passive and active flame test cases for the Rijke tube, where GR denotes the growth rate. The grid resolutions of the *helmholtz-x* tests can be improved to obtain eigenfrequencies closer to the results in [43]

Run	Direct		Adjoint	
	f (1/s)	GR (rad/s)	f (1/s)	GR (rad/s)
[43]	3.425513	+0.001926	3.425514	-0.001904
1D-*helmholtz-x*	3.421902	+0.002225	3.421902	-0.002224
2D-*helmholtz-x*	3.422663	+0.002180	3.422663	-0.002180
3D-*helmholtz-x*	3.420690	+0.002666	3.420690	-0.002667

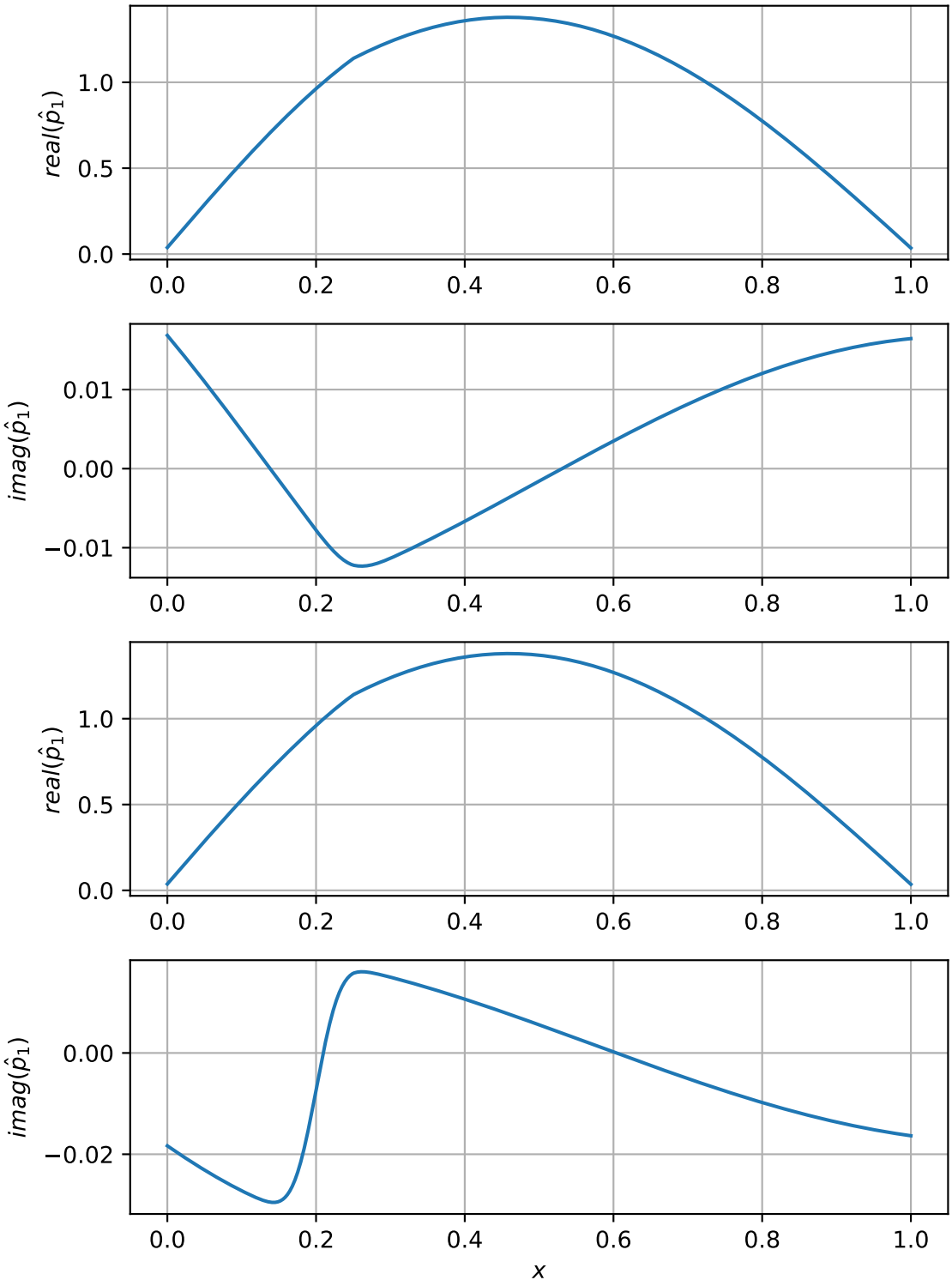


Fig. 2.8 1D direct and adjoint pressure eigenfunctions in the Rijke tube from *helmholtz-x*. The results agree with [43].

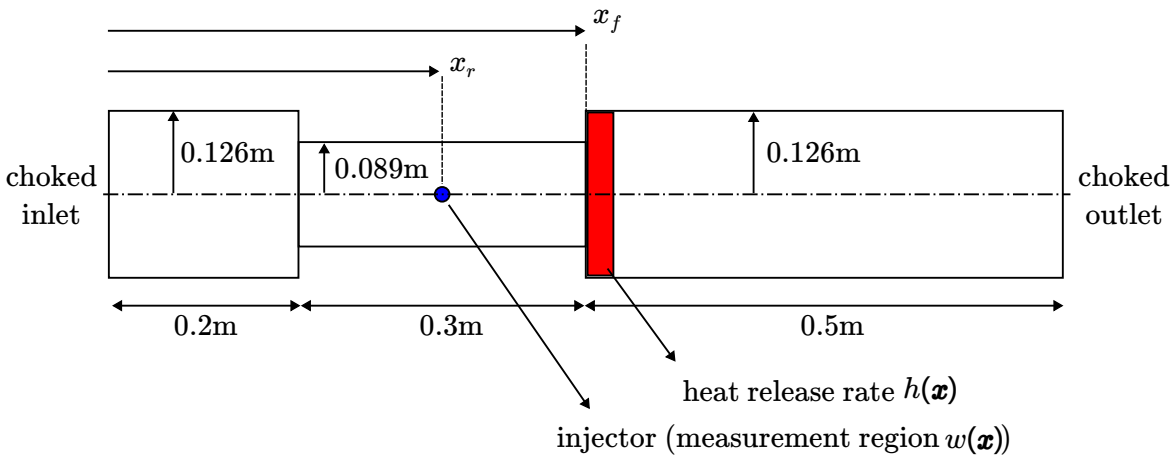
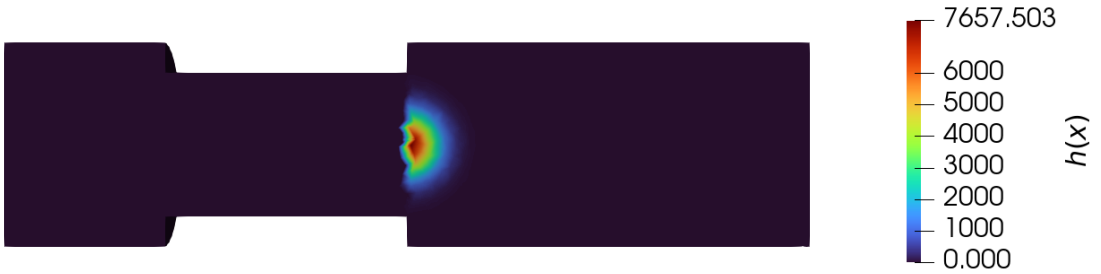


Fig. 2.9 Schematic representation of the flame in a cylindrical duct with choked boundary conditions. The red zone represents the heat release rate field and the blue zone shows the fuel injection point, which is at the centre of the 0.3m duct.

Table 2.5 Dimensional parameters of the flame in a cylindrical duct. T_u denotes the temperature before the flame and T_d denotes the temperature after the flame. γ linearly depends on the temperature as in Sec. 2.2.1

Parameter	value	unit
r_{gas}	287.1	$\mathrm{Jkg^{-1}K^{-1}}$
p_0	101325	Pa
T_u	1000	K
T_d	1500	K
q_0	-57015.232	W
u_b	11.4854	$\mathrm{m\ s^{-1}}$
n	1	-
τ	0.002	s
x_f	0.5	m
a_f	0.025	-
x_r	0.35	m
a_r	0.025	-



(a)

(b)

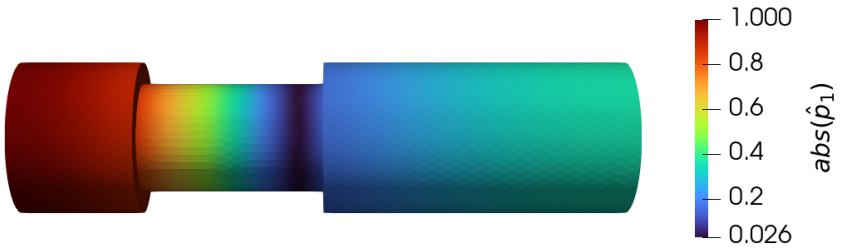
Fig. 2.10 (a) the heat release rate field, $h(\mathbf{x})$, (b) the measurement field, $w(\mathbf{x})$. In (a), the Gaussian function is halved and rescaled such that it integrates to 1.

Table 2.6 shows the eigenvalues for passive and active flame configurations, comparing *helmholtz-x* against the network code. For the passive flame, the thermoacoustic system loses energy through the choked boundaries as expected. For the active flame, the growth rate becomes more negative.

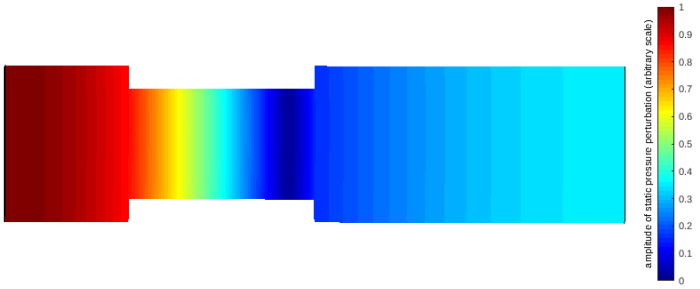
Table 2.6 Eigenfrequencies of the passive and active flame test cases for the flame in a duct. GR denotes the growth rate. 177,737 elements are used for the FEM simulation.

Run	Passive		Active	
	frequency (1/s)	GR (rad/s)	frequency (1/s)	GR (rad/s)
Network code	267.1030	-10.944425	267.307657	-43.4478
helmholtz-x	261.7945	-11.9214	262.559781	-43.2349

The normalized magnitudes of the eigenfunctions of *helmholtz-x* and the network model for the active flame case can be seen in Fig. 2.11. They look almost identical, although the grid of the network code is coarse.



(a) Helmholtz solver



(b) Network model

Fig. 2.11 The (a) and (b) shows the normalized amplitude of the direct eigenfunction $\hat{p}_1$

2.2.2 Annular Combustors

In this section, we demonstrate the capability of *helmholtz-x* to compute thermoacoustic eigenmodes for annular geometries. The *helmholtz-x* code is held in the `numerical_examples/AnnularCombustor/MICCA` folder in the repository. For this test case, we choose a laboratory-scale annular combustor, MICCA [74, 75]. Thermoacoustic limit cycles of standing, spinning, and slanting modes are observed at some operating conditions [76, 77]. The MICCA combustor is composed of an annular plenum, 16 injectors and an annular combustion chamber. Each injector has a burner and a perforated plate. Following [75], the perforated plate and the burner are represented by a cylindrical volume. Fig. 2.12 shows one sector of the MICCA combustor model.

For annular geometries, we use the *PointwiseFlameMatrix* class to implement **D** (see Sec. 2.1.6). We consider the same operating conditions as operating point B in [75]. A standing mode with a stable limit cycle at a frequency of 487 Hz is observed in the experiments. The total power of the flame for each burner is q_0 = 2080 W, and the bulk flow velocity is u_b = 0.66 m/s. The ratio of specific heats, $\gamma = 1.4$, is assumed to be independent of temperature. The mean temperature in the plenum and up to the combustion chamber is $\bar{T}$ = 300 K. In the

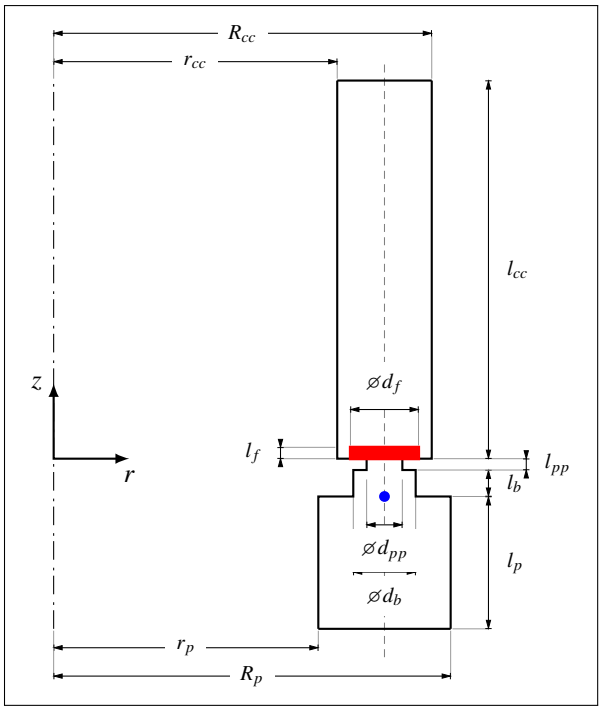


Fig. 2.12 Section of one sector of the MICCA combustor. The dash-dotted line is the axis of symmetry. The subscripts stand for: plenum (p), burner (b), perforated plate (pp), flame (f), combustion chamber (cc). r_p = 140 mm, R_p = 210 mm, l_p = 70 mm, d_b = 33 mm, l_b = 14 mm, d_{pp} = 18.9 mm, l_{pp} = 6 mm, d_f = 36 mm, l_f = 6 mm, r_{cc} = 150 mm, R_{cc} = 200 mm, l_{cc} = 200 mm. The vertical dashed axis represents the longitudinal axis of the burner. The red zone represents the cylindrical heat release rate domain and the blue circle represents the pointwise measurement function.

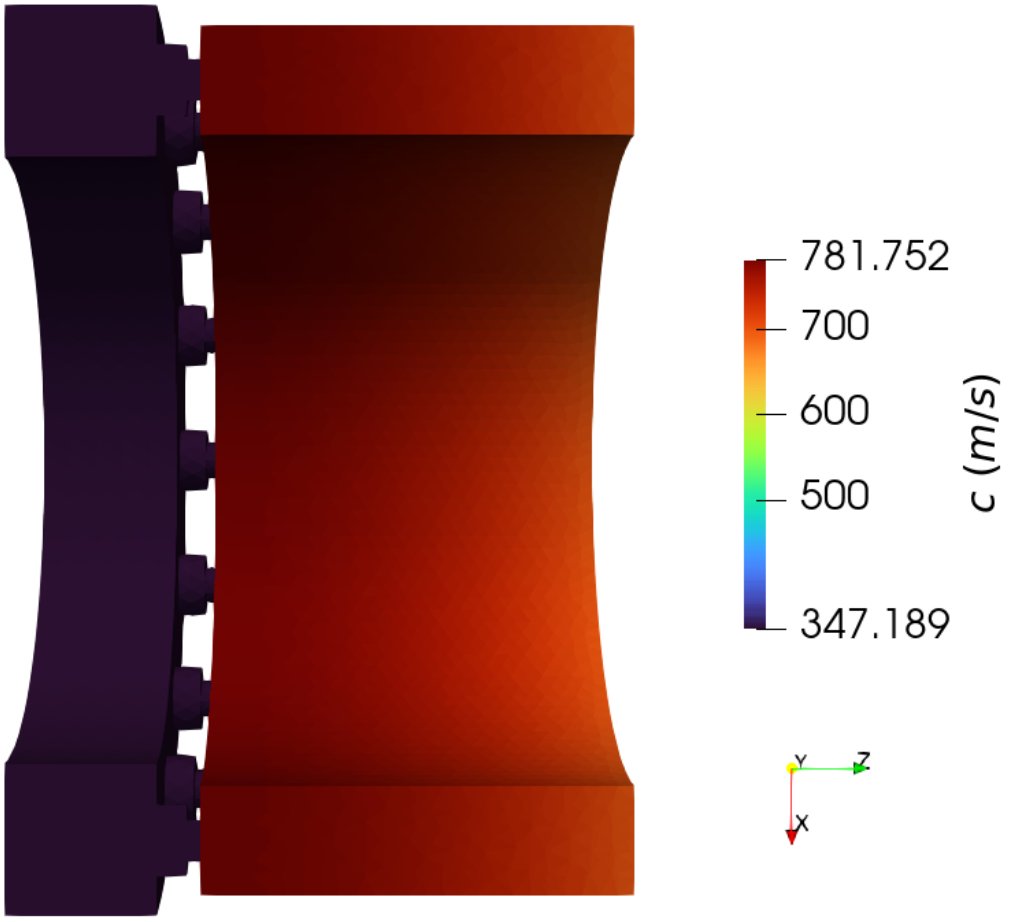


Fig. 2.13 Speed of sound distribution of MICCA combustor calculated using Eq. (2.26).

combustion chamber, the temperature profile is parabolic, gradually decreasing between the values at the flame positions $\mathbf{x}_f$ and the chamber outlet, given in Eq. (2.26). In *helmholtz-x*, Eq. (2.26) is implemented with built-in functions.

$$T(z) = \begin{cases} 300 & \text{if } z < z_f \\ (1200 - 1521)\left(\frac{z - z_f}{l_{cc}}\right)^2 + 1521, & \text{otherwise} \end{cases} \tag{2.26}$$

The corresponding speed of sound field for MICCA is shown in Fig. 2.13. The experimental flame transfer function (FTF) depends on the frequency of the excitation and on the ratio of the root mean square of the velocity fluctuation measured at the reference point, u_1, to the average flow velocity in the injector, u_b (Eq. (2.3)).

We apply Neumann boundary conditions at the combustor walls and a Robin boundary condition at the outlet surface. The reflection coefficient at the outlet boundary is $R_{outlet} =$

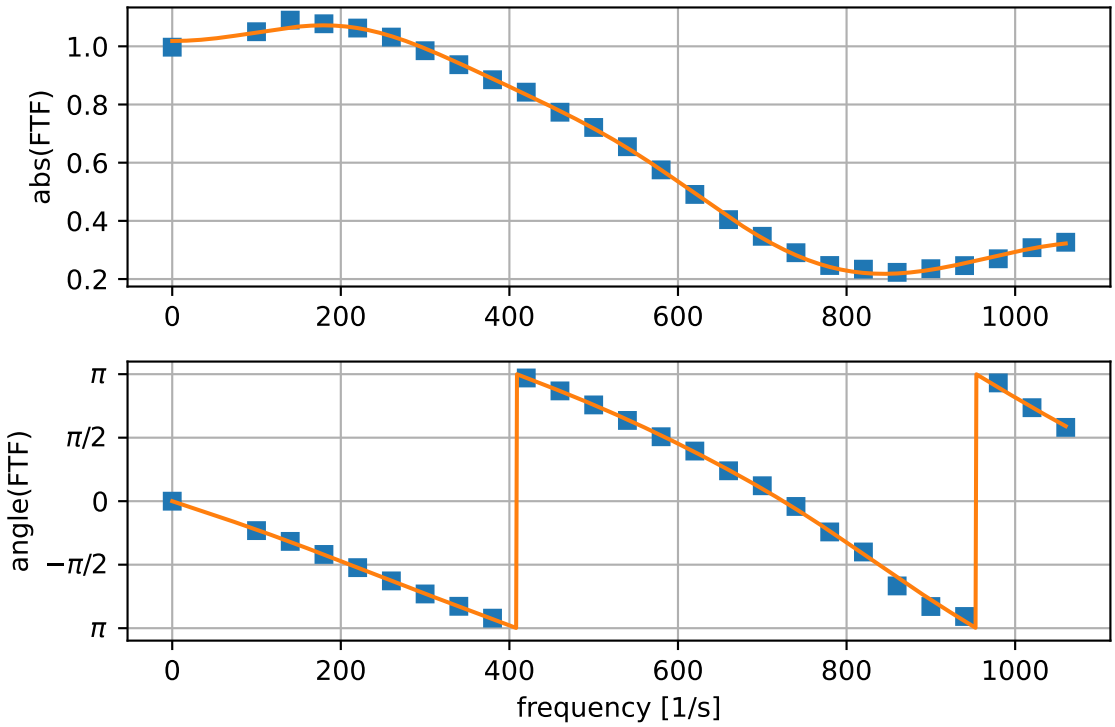


Fig. 2.14 Gain and phase of the flame transfer function ($|u'/\bar{u}| = 0.1$) as a function of the frequency. The squares are the values obtained from the experiments [75], and the solid line is the transfer function of the linear state-space model, evaluated at real values of the angular frequency ω. The *stateSpace* class in *helmholtz-x* is used to obtain an analytical function for FTF(ω).

$-0.875 - 0.2i$. We obtain the flame transfer function FTF, by considering a relatively small amplitude, $|u'/\bar{u}| = 0.1$. In order to calculate the first derivative of the linear operator $\mathscr{L}$ with respect to the eigenvalue ω without approximations, we need the FTF(ω) in equation (2.3) to be analytic in the complex plane [40]. We approximate the frequency response of the flame with a linear state-space model. The transfer function of the state-space model,

$$\text{FTF}(\omega) = \mathbf{s}_3^T \left(i\omega\mathbf{I} - \mathbf{S}_1\right)^{-1} \mathbf{s}_2 + s_4 \tag{2.27}$$

will correspond to the FTF. In order to obtain an analytic transfer function, we apply the Vector Fitting algorithm [78, 40]. The experimental FTF and the transfer function of the state-space model are shown in Fig. 2.14.

Eigenmodes

helmholtz-x can capture numerous eigenmodes by specifying the nearest target to the corresponding eigenvalue. Computations for different eigenfunctions are shown in Fig. 2.15 and their eigenfrequencies are presented in Table 2.7.

Table 2.7 Eigenfrequencies of the active flame test cases for the MICCA combustor using *helmholtz-x*. The corresponding modeshapes are shown in Fig. 2.15. The growth rates of Fig. 2.15c and Fig. 2.15d become closer when the numerical grid is refined.

Mode	Active	
	frequency (1/s)	GR (rad/s)
Fig. 2.15a	149.151	-534.155
Fig. 2.15b	289.976	-629.029
Fig. 2.15c	517.364	+465.643
Fig. 2.15d	517.355	+435.378
Fig. 2.15e	721.206	+3.871
Fig. 2.15f	1314.411	-5.202
Fig. 2.15g	1617.749	-22.147
Fig. 2.15h	1721.129	+333.431

Bloch boundary conditions

In this section, we check the Bloch boundary condition implementation within *helmholtz-x*. We use a single sector of MICCA and the same parameters as in Sec. 2.2.2. As explained in Sec. 2.1.9, we impose slave and master boundaries of Bloch boundaries with their physical tags. We calculate the Bloch form of the acoustic and flame matrices for MICCA. For this comparison, we only consider the plenum-dominant azimuthal mode (Fig. 2.15c) and we verify the results against [18]. The results for different case studies, including with parallelization, are tabulated in Table 2.8.

2.3 Conclusions

This chapter presents an open-source parallelized finite element framework, *helmholtz-x*, which solves the thermoacoustic Helmholtz equation, and present increasingly elaborate examples. In Sec. 2.1, we explain the FEM discretization and implementation details for *helmholtz-x*. In Sec. 2.2.1, we investigate axial eigenmodes in longitudinal combustors. We begin with a relatively simple example, the Rijke tube with Neumann boundary conditions.

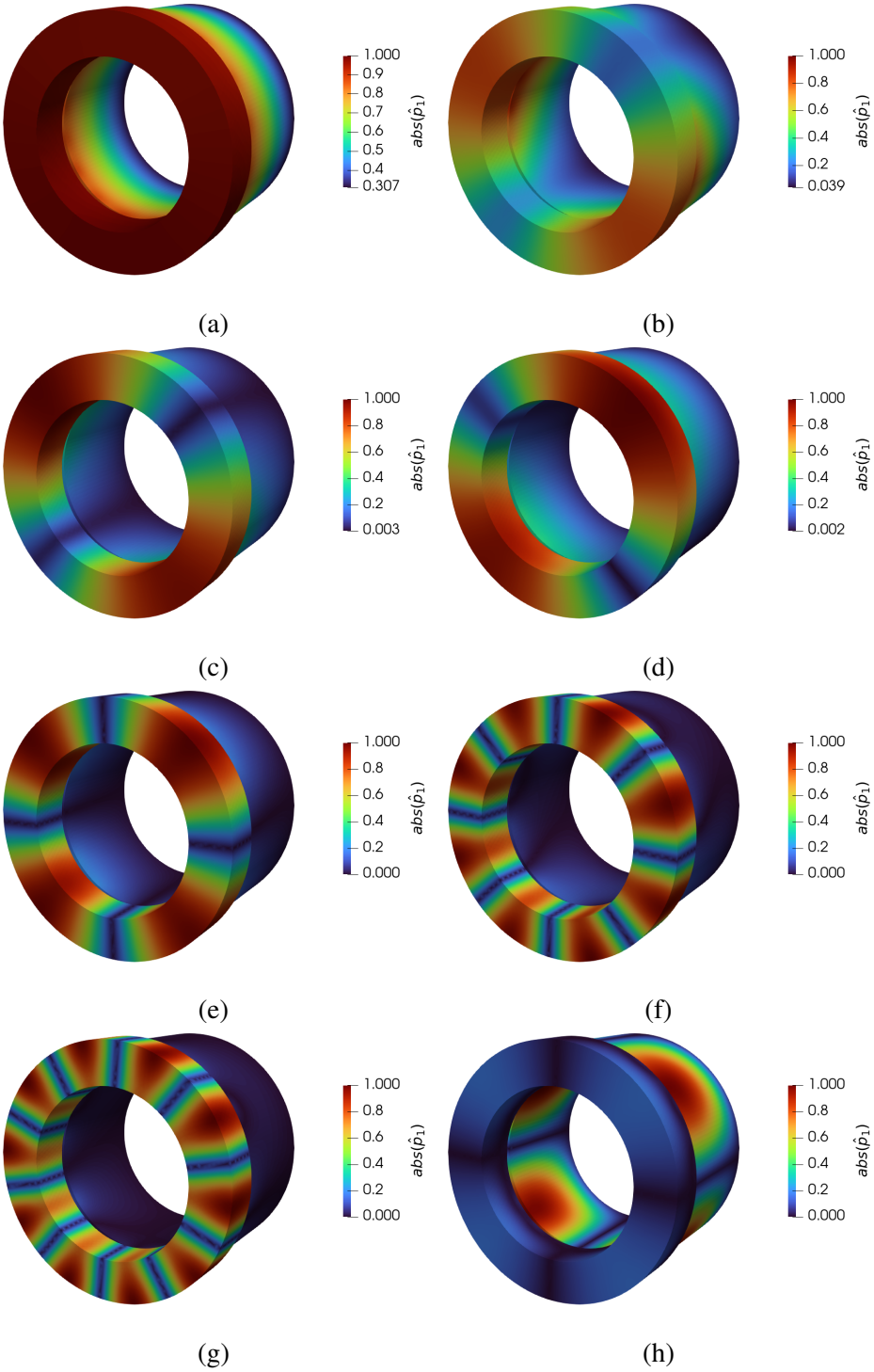


Fig. 2.15 Computed eigenmodes for the MICCA combustor with *helmholtz-x*. The corresponding eigenvalues are given in Table 2.7.

Table 2.8 Eigenfrequencies of the active flame test cases for the MICCA combustor calculated with fixed point iteration.

Case	Tool	Number of processors	Number of cells	Eigenfrequency (1/s)	Computation time (s)
Experiment	[75]	-	-	487	-
Full Annulus	[19]	1	10,528	511.4+79.4j	4627.55
Bloch	[19]	1	658	511.4+79.4j	82.88
Full Annulus	*helmholtz-x*	1	163,165	517.3+74.1j	122.47
Full Annulus	*helmholtz-x*	8	167,401	517.3+74.1j	14.01
Bloch	*helmholtz-x*	1	47,672	513.3+75.6j	15.70

Then we propose a more detailed longitudinal example with area changes in the axial direction and choked boundary conditions at the inlet and outlet boundaries. We find that eigenmode computations of *helmholtz-x* in different configurations agree well with those of a network model for passive and active flame cases (Table 2.2 and Table 2.6). In Sec. 2.2.2, we present a numerical example of a laboratory 3D annular combustor, MICCA. We implement a 3D parabolic temperature field. We also present a state space representation of the experimental data of the flame transfer function and obtain its analytical expression. Then we present different possible eigenmodes of the MICCA combustor. We visualize the corresponding eigenvectors in Fig. 2.15, in which *helmholtz-x* manages to capture axial (Fig. 2.15a), circumferential (e.g. Figs. 2.15c, 2.15d) and mixed modes (Fig. 2.15h). For the efficient calculation of circumferential modes, we also introduce Bloch boundary condition to MICCA in Sec. 2.2.2. When combined with parallel computing, the circumferential eigenmode computations are much quicker with *helmholtz-x* than with existing 3D FEM tools in the literature.

Given its applicability to the examples shown in this chapter, *helmholtz-x* could be a useful numerical tool to study and passively control thermoacoustic instabilities of complex shaped real-world combustors. The adjoint and parallel capabilities of *helmholtz-x* quickly calculate design changes that stabilize thermoacoustic systems. These can be combined with other constraints and entered into an optimization algorithm.

Chapter 3

Geometry parametrization with NURBS

Non-Uniform Rational B-Spline (NURBS) curves and surfaces are effective tools to represent complex 3D geometries. Defining the geometry using parametric functions enables gentle control over the surfaces and curves. Calculation of the derivative of the curve/surface with respect to the control point is easy. In this chapter, we define geometries using NURBS, calculate their displacement fields, and find shape derivatives of the control point. We start with a simple 2D circle and 3D cylinder. Finally, we represent the MICCA combustor using NURBS.

Part of the content of this chapter is published in *Computer Methods in Applied Mechanics and Engineering* [45]. The area derivatives for the azimuthal mode is calculated in [57] and the explanation of these written by Stefano Falco. In this chapter, we extend his anaysis further with NURBS parametrization and also analyze a mixed eigenmode of MICCA.

3.1 NURBS curves and surfaces

We give a brief introduction to NURBS curves and surfaces with a unit circle example. The main reference for this section is chapter 7 in [79]. A NURBS curve with degree p can be obtained from

$$C(k_1) = \frac{\sum_{i=0}^{n} N_{i,p}(k_1) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(k_1) w_i} \tag{3.1}$$

where $P_i(x_i, y_i, z_i)$ are the control points, w_i are the weights, $N_{i,p}$ are the pth degree B-spline basis functions and k_1 is the knot. Composition of two NURBS curves with different directions gives a NURBS surface. The mathematical representation of the NURBS surface

is similar to (3.1):

$$S(k_1,k_2) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} N_{i,p}(k_1)N_{j,q}(k_2)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{n} N_{i,p}(k_1)N_{j,q}(k_2)w_{i,j}} \tag{3.2}$$

where q is the degree and the k_2 is the knot of the second curve.

For shape derivative calculations, we need displacement fields from NURBS. We can obtain these by differentiating the NURBS geometry with respect to the control point, giving:

$$V_i = \frac{\partial C(k_1)}{\partial P_i} = \frac{\sum_{i=0}^{n} N_{i,p}(k_1)w_i}{\sum_{i=0}^{n} N_{i,p}(k_1)w_i} \tag{3.3}$$

for the curve and

$$V_{i,j} = \frac{\partial S(k_1,k_2)}{\partial P_{i,j}} = \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} N_{i,p}(k_1)N_{j,q}(k_2)w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{n} N_{i,p}(k_1)N_{j,q}(k_2)w_{i,j}} \tag{3.4}$$

for the surface to compute the corresponding deformation fields for node i in the circumferential direction and node j in the axial direction. Note that Eq. (3.4) implies the cross product of the basis functions of the two different NURBS curves. We use these formulae to calculate the displacement field in the shape gradient formula.

3.2 Shape Derivatives

We obtain the shape derivative for the Robin boundary conditions from [57]:

$$\omega' = \int_{\Gamma_1} C\left(\hat{p}^{\dagger *}\bar{c}^2\frac{\partial^2\hat{p}}{\partial n^2} - \kappa\hat{p}^{\dagger *}\bar{c}^2\frac{\partial\hat{p}}{\partial n} + \operatorname{div}_\Gamma\left(\hat{p}^{\dagger *}\bar{c}^2\nabla\hat{p}\right) - \frac{\partial\hat{p}^{\dagger *}}{\partial n}\bar{c}^2\frac{\partial\hat{p}}{\partial n}\right)\mathrm{d}S \tag{3.5}$$

where κ is the curvature. The shape derivative becomes

$$\omega' = -\int_{\Gamma_1} C\left(\bar{c}^2\frac{\partial\hat{p}^{\dagger *}}{\partial n}\frac{\partial\hat{p}}{\partial n}\right)\mathrm{d}S \tag{3.6}$$

for Dirichlet boundary conditions and

$$\omega' = \int_{\Gamma_1} C\Big(\nabla \cdot \big(\hat{p}^{\dagger^*} \bar{c}^2 \nabla \hat{p}\big)\Big)\, \mathrm{d}S \tag{3.7}$$

for Neumann boundary conditions.

3.3 NURBS parametrization of a unit circle

As a demonstration, we define a unit circle using NURBS. The parameters of the NURBS control points are tabulated in Table 3.1. Using Gmsh Python API, we define the control

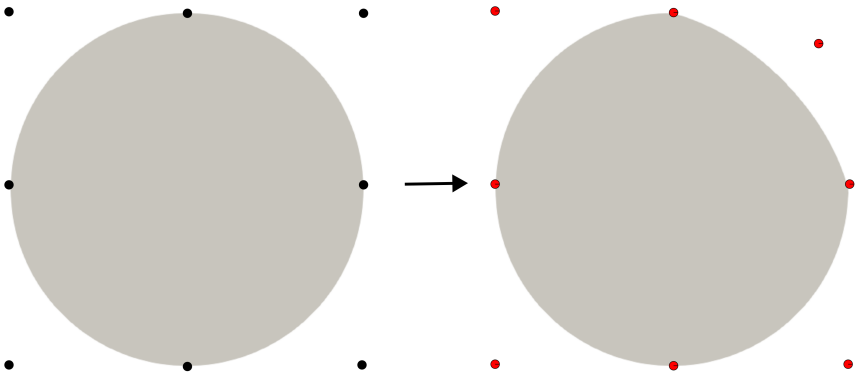
Table 3.1 Control points and their weights of the NURBS for the unit circle ($r = 1$) with degree 2. The knot vector is $k_1 = (0,0,0,0.25,0.25,0.5,0.5,0.5,0.75,0.75,1,1,1)$.

i	x_i	y_i	z_i	w_i
1	1.0	0.0	0.0	1.0
2	1.0	1.0	0.0	$\sqrt{2}/2$
3	0.0	1.0	0.0	1.0
4	-1.0	1.0	0.0	$\sqrt{2}/2$
5	-1.0	0.0	0.0	1
6	-1.0	-1.0	0.0	$\sqrt{2}/2$
7	0.0	-1.0	0.0	1.0
8	1.0	-1.0	0.0	$\sqrt{2}/2$
9	1.0	0.0	0.0	1.0

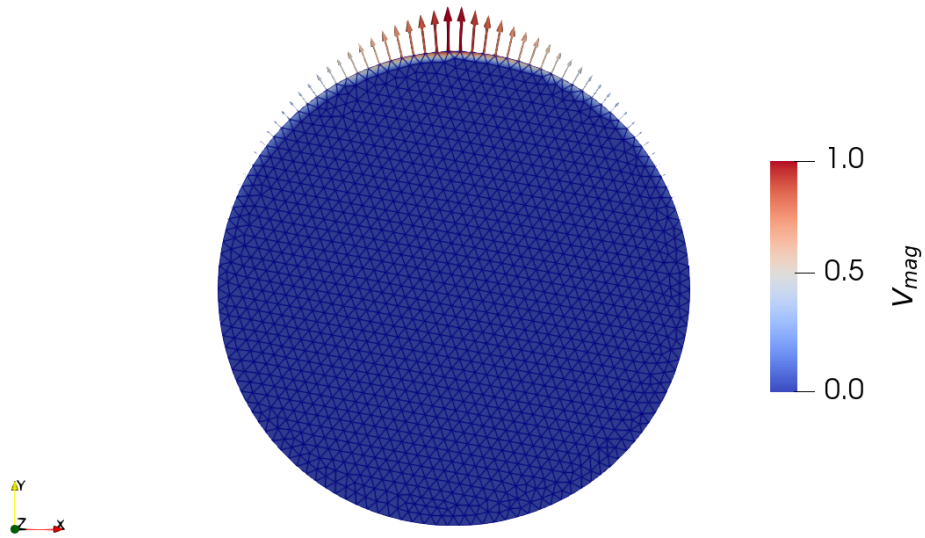
points in Table 3.1 in terms of the weights of the control points and the knot vector with multiplicities as well as characteristic mesh size of 0.04. We then generate the degree 2 closed NURBS curve and transform it into the NURBS surface to obtain the NURBS unit circle. If we deform point 2 in the direction towards the centre of the circle, we obtain the deformed circle shown in Fig. 3.1a.

The parametrization utility of the Gmsh model is used to parametrize the boundary curve. We use these parameters to compute the displacement field of the control points using Eq. (3.3) to compute the pointwise shape derivative in the outward normal direction. The parametrization utility of the Gmsh model is used to parametrize the boundary curve. We use these parameters to compute the displacement field of the 3rd control point using Eq. (3.3) (Fig. 3.1b).

We use 4760 P2 Galerkin finite elements to simulate the acoustics on the unit circle. We model the edge of the circle as a Dirichlet boundary. We set the speed of sound to 343



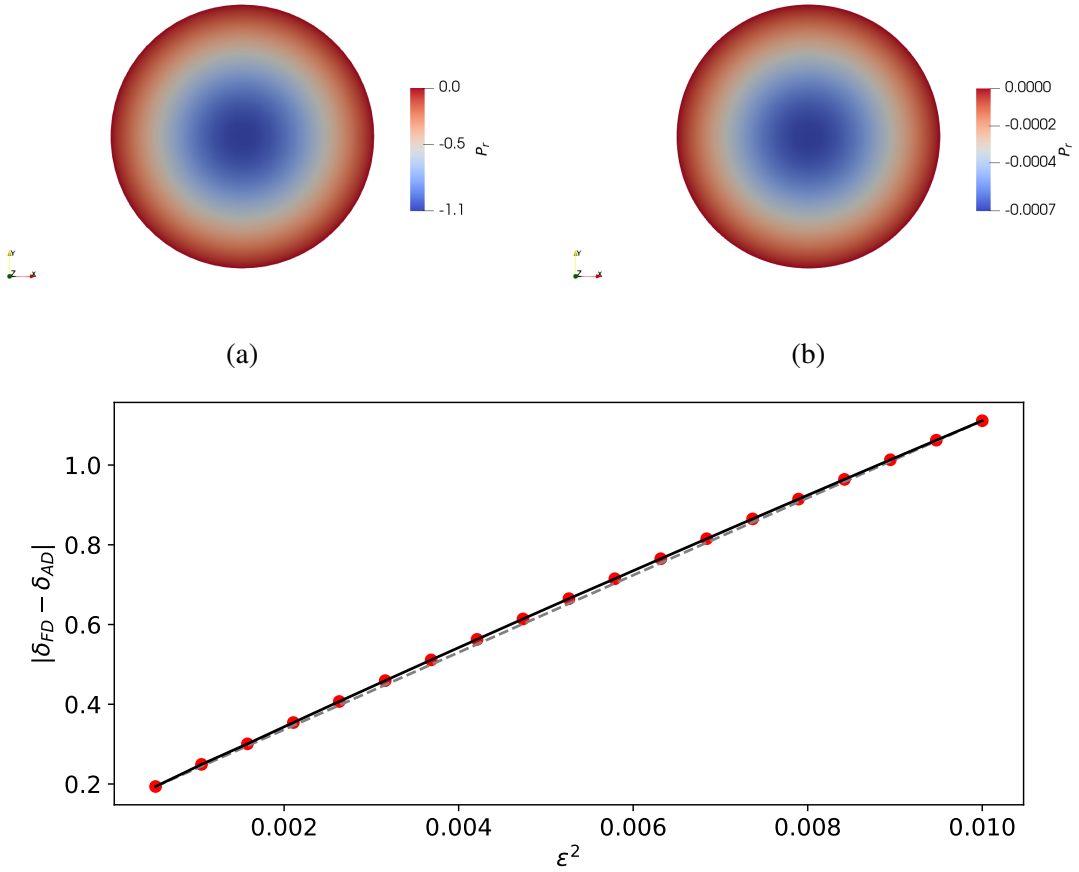
(a)



(b)

Fig. 3.1 (a) The unit circle before (left) and after (right) deformation. (b) Displacement field of the 3rd control point for the unit circle. As the vectors get closer to the 3rd control point, the magnitude of the normal vectors converges to 1.0.

m/s. We compute the direct and adjoint eigenmodes and find the eigenfrequency of the first radial mode to be 131.307Hz. Note that the adjoint eigenfunction needs to be normalized to compute the shape gradient correctly. The direct and adjoint eigenfunctions are visualized in Fig. 3.2.



(a) (b)

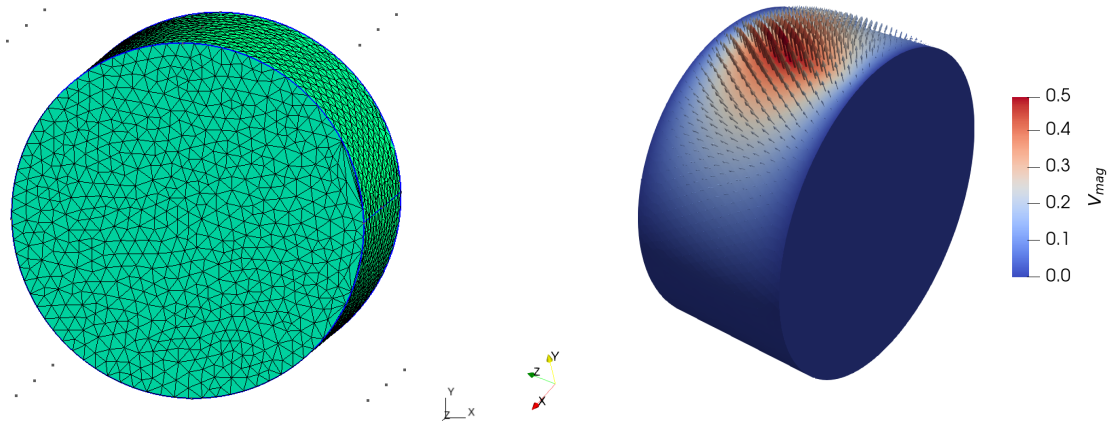
(c)

Fig. 3.2 The first radial (a) direct and (b) adjoint eigenfunctions of the circle. (c) Successful Taylor test for the unit circle.

Using the direct and adjoint eigenfunctions, the speed of sound, and the displacement field with corresponding facet normal, we compute the Dirichlet shape gradient for the 3rd control point to be $\omega_r' = -136.84 + 0j$. To check the validity of this shape gradient, we perform a Taylor test to check that the discrepancy with respect to the finite difference calculation scales with ε^2. We move the 3rd control point in the $+y$ direction with ε from 10^{-6} to 10^{-2}. The results are shown in Fig. 3.2c.

3.4 NURBS parametrization of a cylinder

In this section, we extend the analysis of the shape gradients to the 3D cylinder. We parametrize the cylinder with 27 control points. We extend the circle by repeating it twice in the z direction. The cylinder has radius 0.25m and height 0.25m. We specify a shorter height to reduce the computational effort during the Taylor test. We have 2 different knot vectors, one for the u direction and the other for the v direction. The degree of the curves is 2 for both directions. The knot vector in the u direction is the same as that for the circle. The knot vector for the v direction is $V = (0,0,0,1,1,1)$. The cylinder geometry with control points is visualized in Fig. 3.3a.



(a) (b)

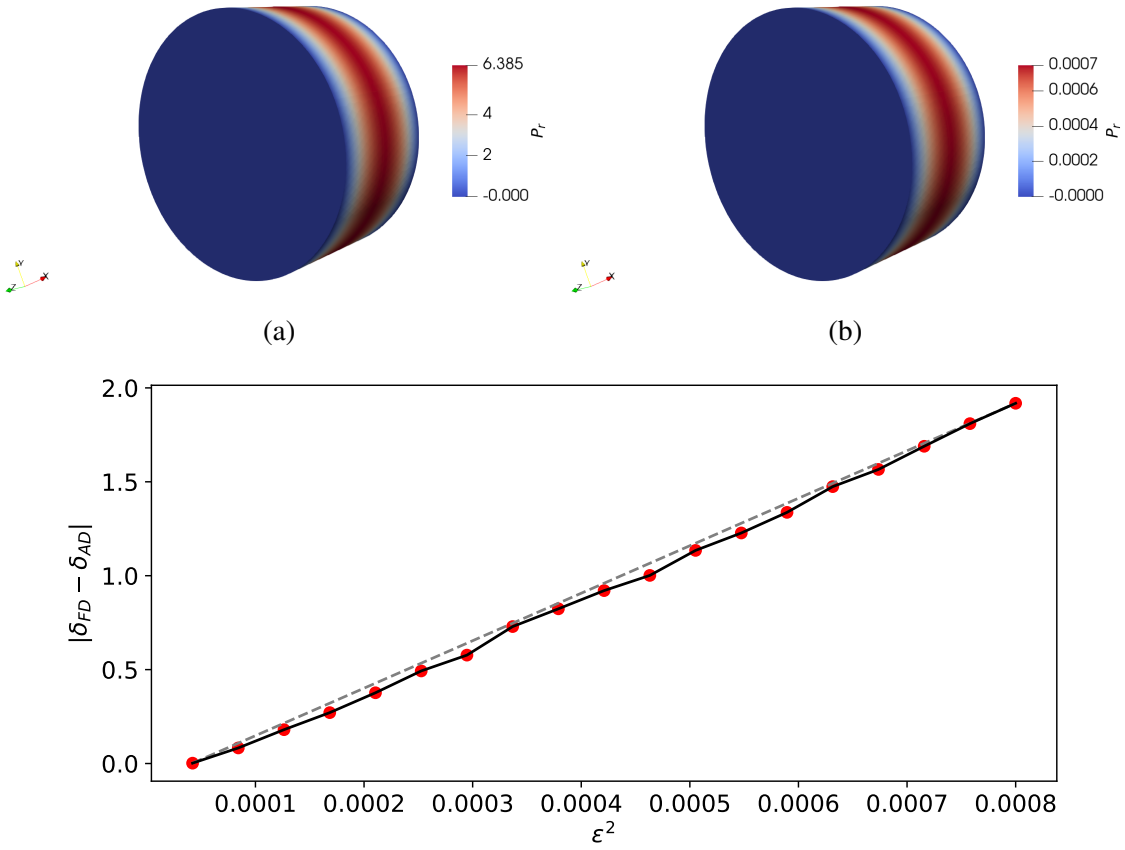
Fig. 3.3 (a) Computational grid of the cylinder with 27 control points. (b) The displacement field of the 3rd control point on the lateral surface of the cylinder. The magnitude of the normal vectors is maximal at the 3rd control point.

We use Eq. (3.4) to compute the displacement field for the control point on the $+y$ axis, which is the 3rd point among the control points at $z = 0.125$ (mid-height of the cylinder). Again, we exploit the utilities of Gmsh to extract the (u,v) parameters of the vertex coordinates on the lateral surface to calculate the basis functions. The 3D displacement field vectors are presented in Fig. 3.3b.

We use 14326 P2 Galerkin finite elements to perform acoustic simulations of the cylinder. We impose Dirichlet boundaries for the inlet and outlet, and a Neumann boundary for the lateral surface. We compute the longitudinal modes of the cylinder. The eigenfrequency of the first longitudinal mode is found to be 686.02Hz. The direct and normalized adjoint eigenfunctions are shown in Fig. 3.4a and Fig. 3.4b.

The shape gradient of the 3rd control point is $\omega_r' = -573.16 + 0j$. We perform a Taylor test to check the adjoint code for the Neumann boundary. The 3rd control point on the

lateral surface is moved in the $+y$ direction by 20 equispaced epsilons ranging from 10^{-6} to 8×10^{-4}. The Taylor test results are shown in Fig. 3.4c. We conclude that the adjoint code is valid to compute the shape gradients on the Neumann boundary.



(a) (b)

(c)

Fig. 3.4 The first longitudinal (a) direct and (b) adjoint eigenfunctions of the cylinder. (c) Taylor test for the 3rd control point on the lateral surface of the cylinder.

3.5 NURBS parametrization of MICCA

We are ready to parametrize a more complex 3D geometry. We consider the MICCA geometry presented in Sec. 2.2.2. Before generating the mesh, we decompose the geometry into the plenum, burner, perforated plate and combustion chamber. We start by defining the control points to generate the NURBS surfaces.

We define 9 control points in the circumferential parametric direction, k_1, for each individual component. For the parametric radial direction, k_2, we use 5 control points for the

plenum and combustion chamber and 3 control points for the burner and perforated plate. We use degree 2 B-Spline functions for the biparametric k_1 and k_2 directions. Then we generate inlet and outlet circles and a lateral boundary and fuse them to obtain a parametrized cylinder [79]. We iterate this process for each component. To generate the plenum and combustion chamber, we generate an inner cylinder and subtract it from the outer cylinder. To generate the burner and perforated plate for each sector, we copy the parametrized cylinders and rotate them 15 times. Finally, we merge the decomposed components to obtain the full geometry. The NURBS control points of the MICCA are shown in Fig. 3.5.

3.5.1 Numerical grid & operating conditions

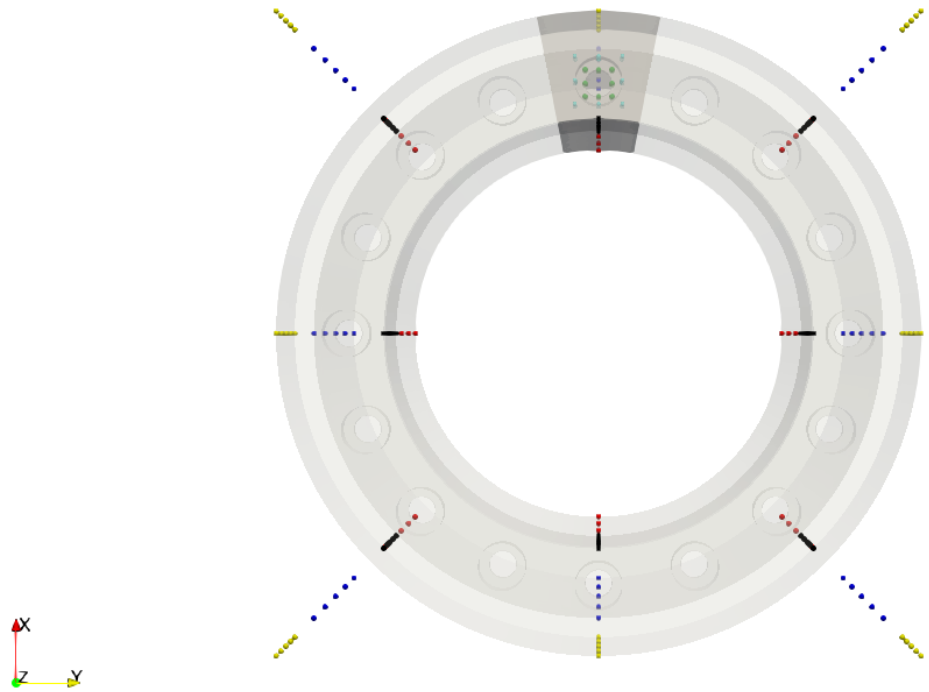
The parametrized geometry is then used to generate the unstructured mesh using the open-source automated 3D finite element mesh generator, Gmsh [72] with a Delaunay-triangulation method. We perform local mesh refinement near the reference points. We then optimize the quality of the tetrahedral elements using the Netgen optimizer. The unstructured mesh and the slice view of the sector mesh are shown in Fig. 3.6. We consider the same operating conditions as in Sec. 2.2.2.

3.5.2 Shape sensitivity of degenerate modes

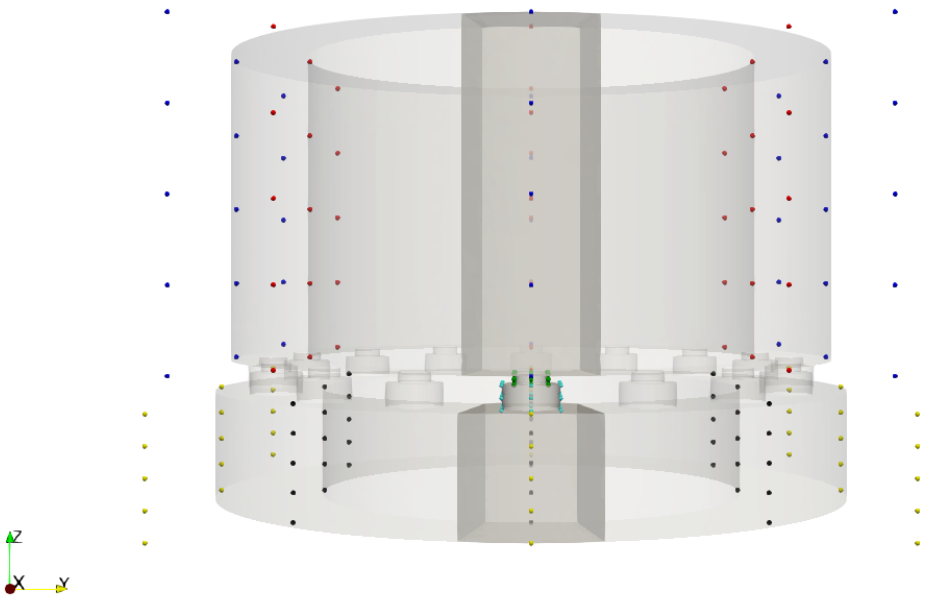
Because of the discrete rotational symmetry of the annular combustor, the modes are degenerate with algebraic and geometric multiplicity of 2. Consequently, these eigenvalues are referred to as semi-simple [80]. If this symmetry is broken, the eigenvalues split, and each can then be considered as a simple eigenvalue.

Shape derivatives for simple eigenvalues

In the case of simple eigenvalues, the shape gradient is the eigenvalue sensitivity to boundary perturbations. The shape derivative for a geometry perturbation proportional to the shape gradient constitutes an upper bound for the shape derivative itself. We obtain an upper bound for the angular frequency drift from the real part of G and for the growth rate drift from the imaginary part of G. In other words, shape changes proportional to the real part of the shape gradient have the maximum effect on the angular frequency, while shape changes proportional to the imaginary part of the shape gradient have the maximum effect on the growth rate.

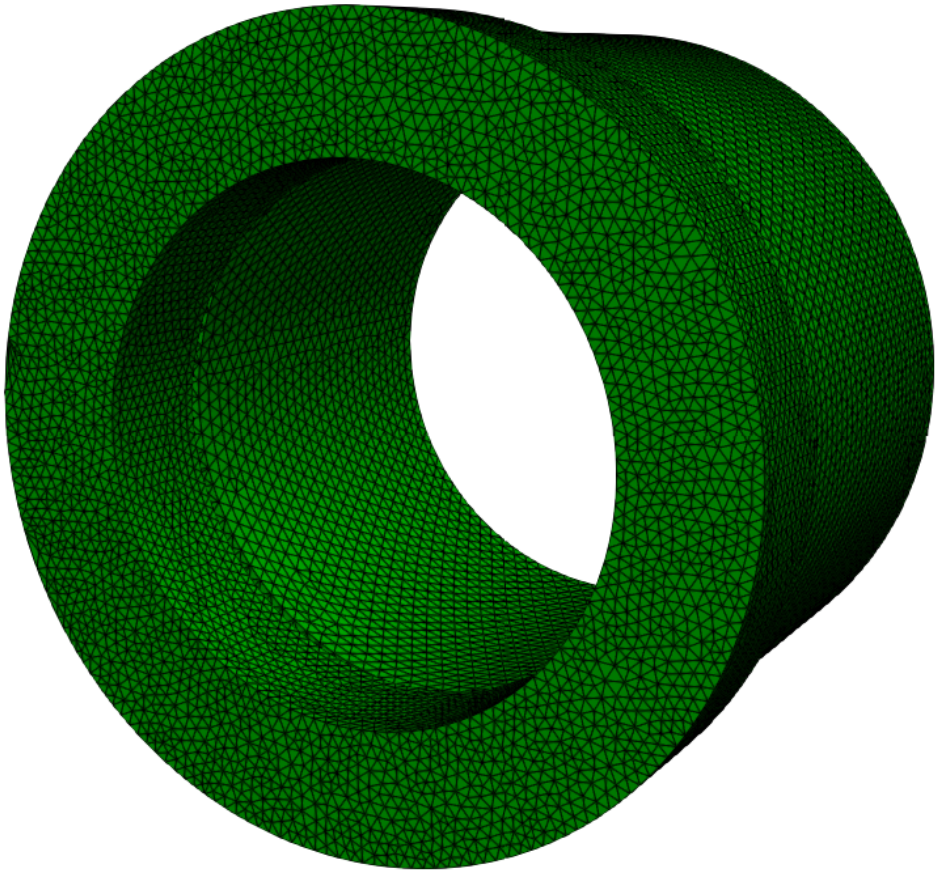


(a)

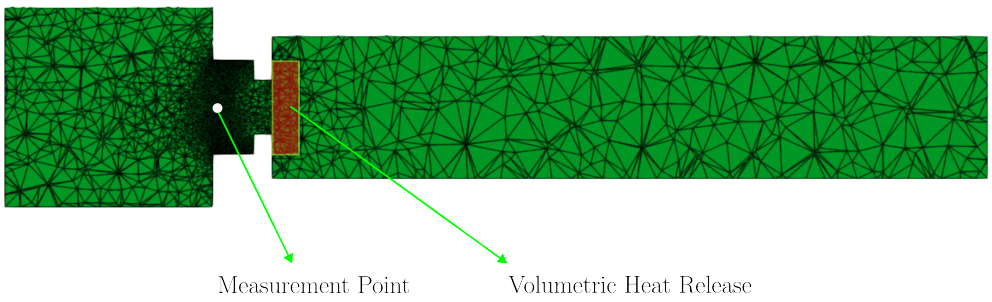


(b)

Fig. 3.5 NURBS geometry viewed in the (a) XY-plane and (b) ZY-plane . The control points for each component are shown in different colors. The yellow and black control points control the outer and inner surfaces of the plenum, respectively. The green control points represent the burner, the cyan points indicate the perforated plate, and the blue and red points correspond to the outer and inner surfaces of the combustion chamber.



(a)



(b)

Fig. 3.6 Finite element mesh of the MICCA combustor (a) external view and (b) slice through a burner. The grid consists of 5.4 million cells. Local refinement is applied at the measurement point to capture the gradient of the acoustic pressure there more accurately. The red zone shows the volumetric cylindrical heat release rate field.

Shape derivatives for semi-simple eigenvalues

In the case of semi-simple two-fold degenerate eigenvalues, the shape gradient, G_{ij}, has four entries and the shape derivatives are the eigenvalues of the matrix $\int CG_{ij}\,\mathrm{d}S$, where C is the perpendicular boundary perturbation or control point perturbation.

G_{11} gives the upper bound for ω_1 and G_{22} gives the upper bound for ω_2. For geometry perturbations proportional to G_{12} and G_{21}, either their real or imaginary parts, the diagonal elements of the matrix $\int CG_{ij}\,\mathrm{d}S$ are equal to zero. Therefore, the shape derivatives take the form $\omega' = \pm(a+bi)$. The shape derivative is the same for the two eigenvalues, ω_1 and ω_2, but with opposite sign. In other words, at first order the eigenvalues split in opposite directions. The off-diagonal elements also have the property that:

$$\int G_{ij}\,\mathrm{d}S = 0 \quad \text{if } i \neq j \tag{3.8}$$

which means that the volume of the combustor does not change when boundary perturbations proportional to them are applied. The resulting shape gradient would be

$$\left(\int_{\Gamma_1} C\,G\left(\hat{p}_i^{\dagger}, \hat{p}_j\right)\mathrm{d}S - \omega'\delta_{ij}\right)\alpha_j = 0. \tag{3.9}$$

Interestingly, this is conceptually similar to what was observed by Mensah [81] for the same annular combustor, when the burners/injectors are perturbed such that the FTF perturbations have the same phase and their average is zero. The eigenvalue perturbation for the two modes is the same but with opposite sign.

Shape differentiability

In the simple case, the eigenvalues are shape differentiable; i.e. the derivative $\mathrm{d}\omega(\Omega)$ exists for all directions V and the mapping $V \mapsto \mathrm{d}\omega(\Omega)$ is linear and continuous. In the semi-simple degenerate case, the eigenvalues are not differentiable at first order because each eigenvalue splits in two. This can cause a problem in gradient-based optimization because the gradients are in general discontinuous and the cost functional is therefore non-convex. This problem can be avoided by applying symmetry-preserving geometry changes. In this case, the two repeated eigenvalues do not split and the two shape derivatives have the same value. Sections 3.6.1 and 3.6.2 concern this type of geometry change. When applying symmetry-breaking changes, this problem can be worked around by noticing when an eigenvalue has split and subsequently calculating the shape derivatives of each split eigenvalue separately. Section 3.7 concerns this type of geometry change. In all cases, the real part of the shape derivative

represents changes that reduce the angular frequency and the imaginary part represents changes that reduce the growth rate.

3.5.3 Shape modification

For MICCA with NURBS, we consider changes to the plenum and combustion chamber. We can also compute shape derivatives for changes to the diameter of the burner and to the total flow passage area of the perforated plate. Such changes would, however, alter the flame's transfer function. We have used an experimentally-derived flame transfer function and we do not know how it would change due to these changes, so we do not consider these changes further. For an estimate of flame transfer function changes in a different burner, the reader is referred to [82].

We propose two different shape perturbation methods: perpendicular boundary perturbations and NURBS control point perturbations. In the former, the displacements along the boundary's normal direction are uniform. In the latter, the displacements along the boundary's normal direction are non-uniform due to the independent movements of individual control points. We apply both of these methods and demonstrate the shape changes for two different degenerate modes in Sec. 3.6.1 and Sec. 3.6.2. In Sec. 3.7, we consider symmetry-breaking changes for the mixed mode using the shape gradients calculated at the NURBS control points.

Perpendicular boundary perturbations

When using a non-parametric approach, the boundary displacement is proportional to the shape gradient. We divide the geometry into patches and, for each patch we use Eq. (3.9) to compute the shape derivatives for a unitary deformation field.

$$C = \begin{cases} 1 & \text{on boundaries} \\ 0 & \text{elsewhere} \end{cases} \tag{3.10}$$

Then we divide the shape derivatives by the surface area of the patch. In this way, we obtain a local average of the derivatives, which does not depend on the area of the patch. An example perpendicular deformation field, C, is shown in Fig. 3.7 for the outer surface of the combustion chamber.

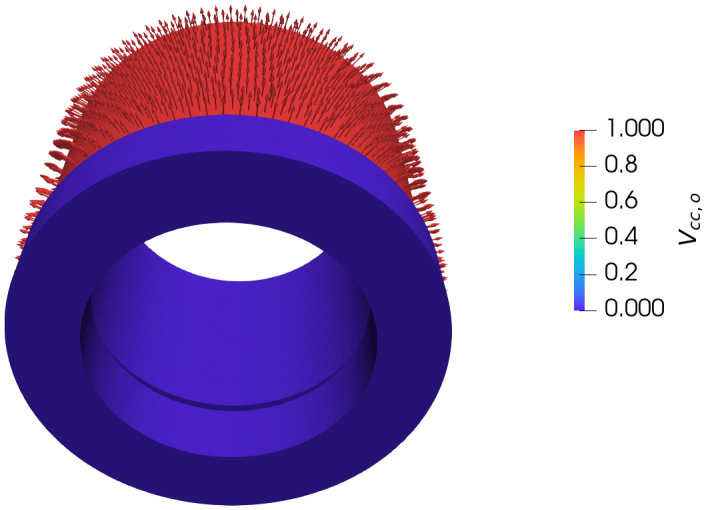


Fig. 3.7 Perpendicular deformation field (V) for the outer combustion chamber surface of the MICCA combustor. By construction, the deformation vectors have the same magnitude at each node of the surface mesh.

Control point perturbations

When using a parametric approach, as in the case of B-spline or NURBS surfaces, the set of admissible boundary displacements and shapes is expanded. We point out that more elaborate shape changes could be considered with this approach, because we obtained the NURBS parametrization of the shape. However, for this approach, we do not change the length of the plenum and the combustion chamber. The control point perturbation, $V_{i,j}$, is the derivative of the NURBS surface with respect to the position of the control point $P_{i,j}$. In Eq. (3.4), N denotes the B-Spline basis function, k_1 and k_2 denote the circumferential and axial parameters of the surface and $w_{i,j}$ denotes the weight of the control point $P_{i,j}$. We show the boundary perturbation of the control point on the plenum surface in Fig. 3.8.

As we only perform changes on the plenum and combustion chamber, we use Eq. (3.7) to compute the shape gradients of the control points on the Neumann boundaries. We apply two different geometry modifications using NURBS control points: symmetry-preserving (Sec. 3.6) and symmetry-breaking (Sec. 3.7).

3.5.4 Optimization procedure

When we modify the shape, we fix the geometry of the burner and allow changes only in the plenum and in the combustion chamber. For symmetry-preserving changes, the combustor

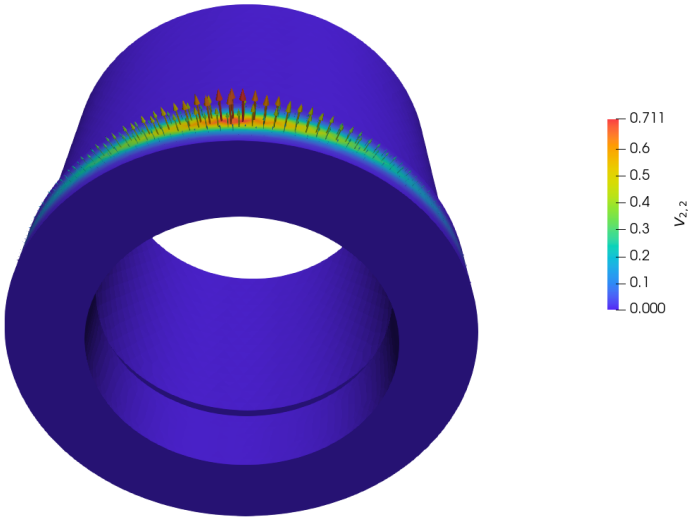


Fig. 3.8 Deformation field (V) due to changes in the middle node on the top of the lateral plenum surface of the MICCA combustor.

axis in the longitudinal direction is taken as the symmetry axis. Our goal is to reduce the eigenvalue growth rate by following the steps below.

(i) We use P2 elements to compute the shape derivatives for each boundary or control point.

(ii) We take the imaginary part and normalize such that the maximum absolute value among all the shape derivatives is 1.

(iii) We subtract from the geometric parameters the imaginary part of the shape derivatives multiplied by the step size, ε, which can take a range of values. Here, ε is 0.025 for symmetry-preserving changes and between 0.01 and 0.07 for symmetry-breaking changes.

(iv) We generate a refined finite element mesh with re-calculated geometric parameters.

(v) We simulate the optimized geometry with P1 elements and observe the changes in the growth rate of the new eigenvalue of the new eigenmode.

3.6 Symmetry-preserving Changes

In this section, we change geometries of the plenum and the combustion chamber with two methods: perpendicular boundary movements and control point displacements with symmetry-preserving changes.

3.6.1 Azimuthal Mode

Fig. 3.9 shows the normalized magnitude of the eigenvector, $\hat{p}$ of the first azimuthal mode. Figure 3.10 shows the real and the imaginary part of the shape derivatives for the plenum

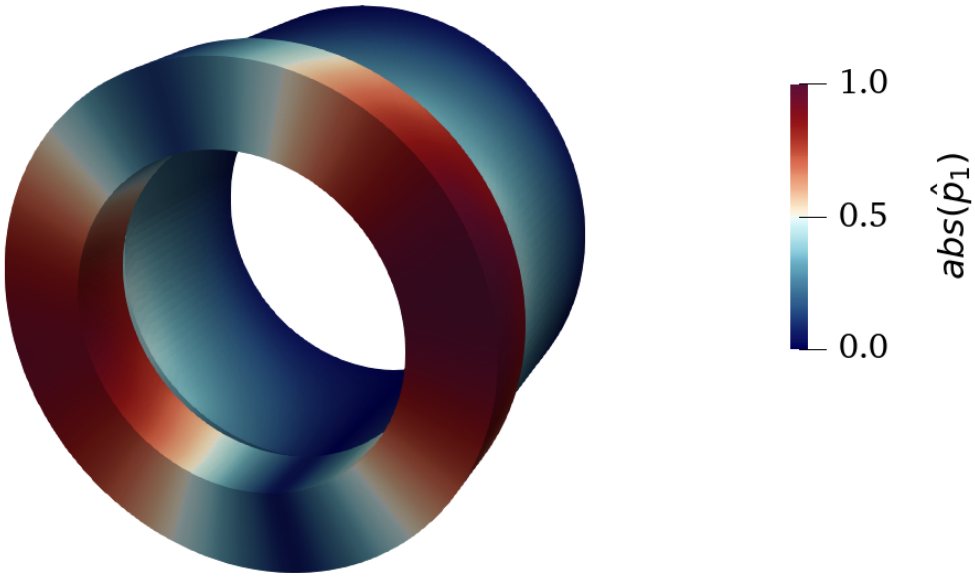


Fig. 3.9 Normalized absolute value of the eigenvector, $\hat{p}$, of the first azimuthal mode of the MICCA combustor (in arbitrary units). This is a weakly coupled plenum mode. The associated eigenvalue is $\omega = 3222 + 517i$ rad s^{-1}.

and the combustion chamber. These are shown as perpendicular boundary displacements proportional to the shape derivatives of the surfaces of the combustor. In order to investigate the effects of the control points on the lateral boundaries of the plenum and combustion chamber, the shape derivatives of the control points on the $+yz$ plane ($k_2 = 2$) are computed. Figure 3.11 shows the real and the imaginary part of the shape derivatives for the first azimuthal mode. These are shown as control point displacements proportional to the shape derivatives in the normal direction of the relevant surface. Fig. 3.12 (top) shows the initial and the final shapes of a sector of the annular combustor. The growth rate has reduced by 20% after 4 iterations. Although the process can be continued, the salient points are most

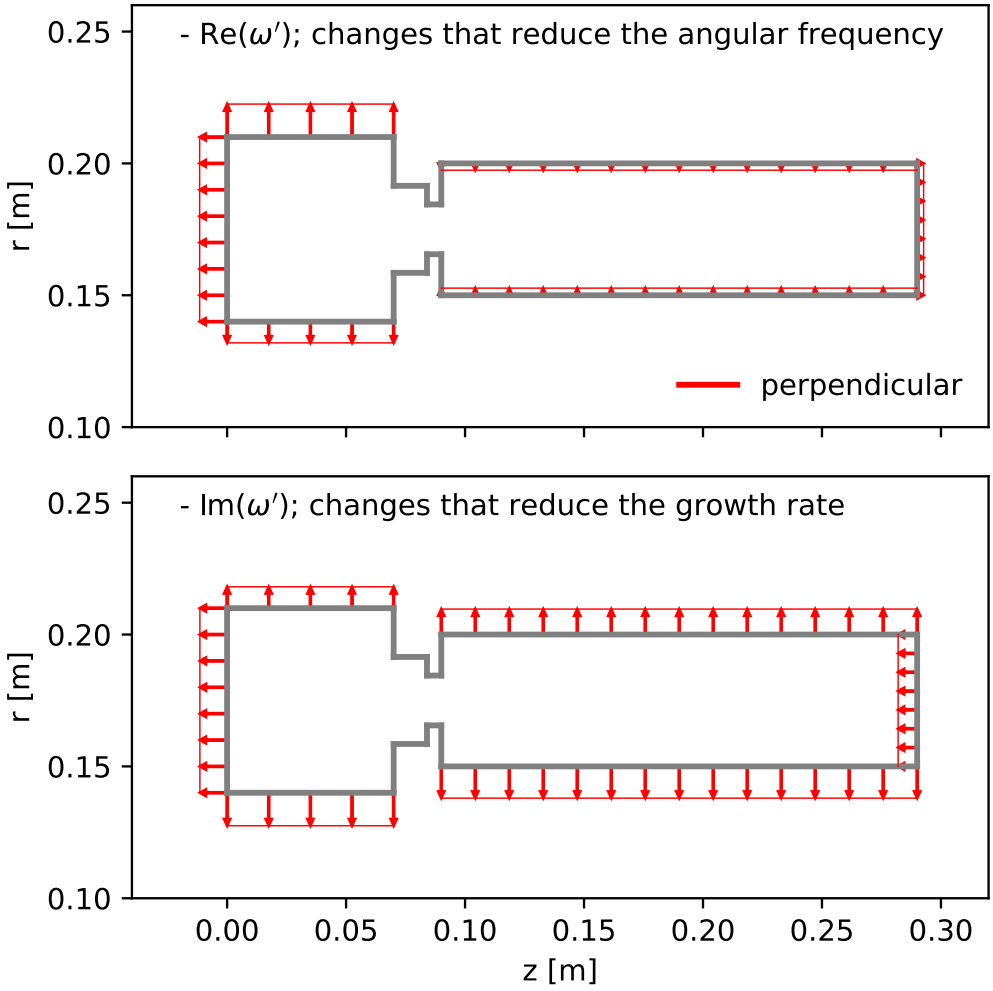


Fig. 3.10 Real (top) and imaginary (bottom) components of the eigenvalue shape derivatives for changes to the length and the radii of the plenum and the combustion chamber for the first azimuthal mode. The real part gives the influence on the angular frequency and the imaginary part gives the influence on the growth rate. This plot is recalculated with *helmholtz-x* and the results match the results in Sec 6.3 of [57].

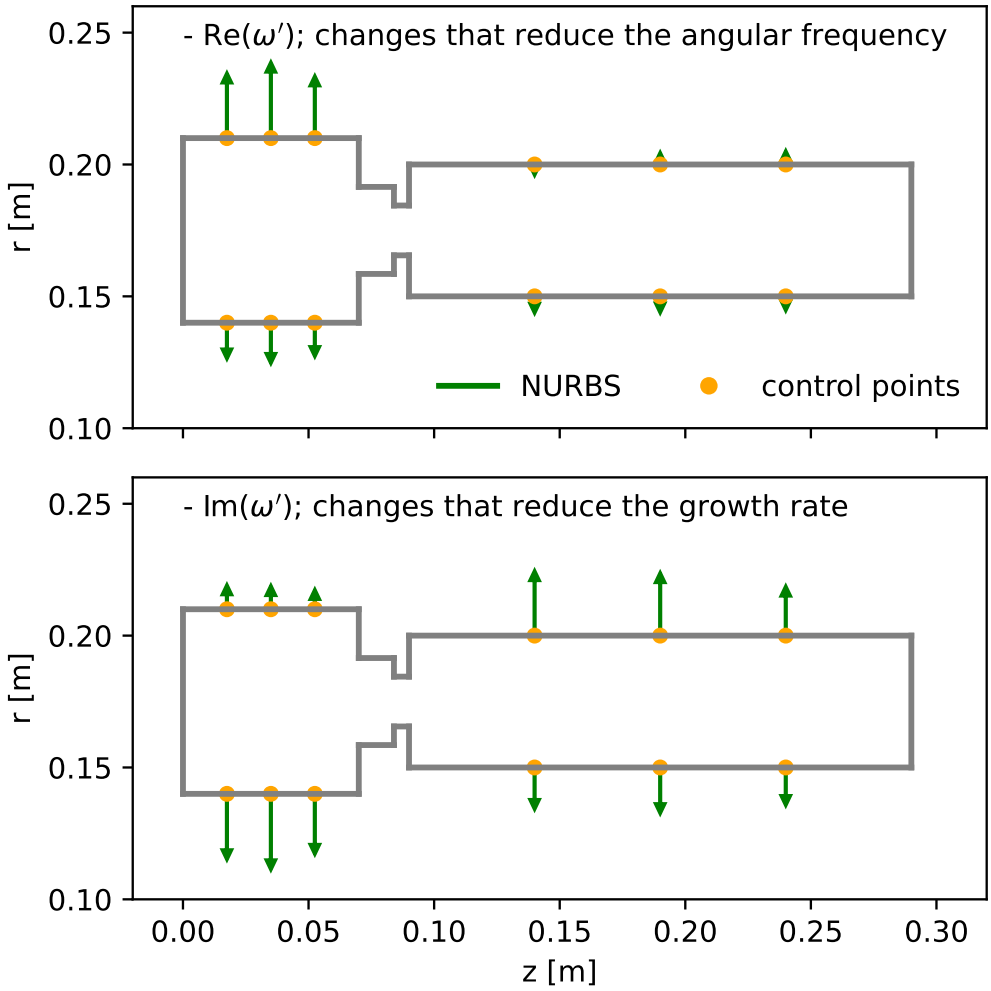


Fig. 3.11 As for Fig. 3.10 but for changes parametrized by the NURBS points.

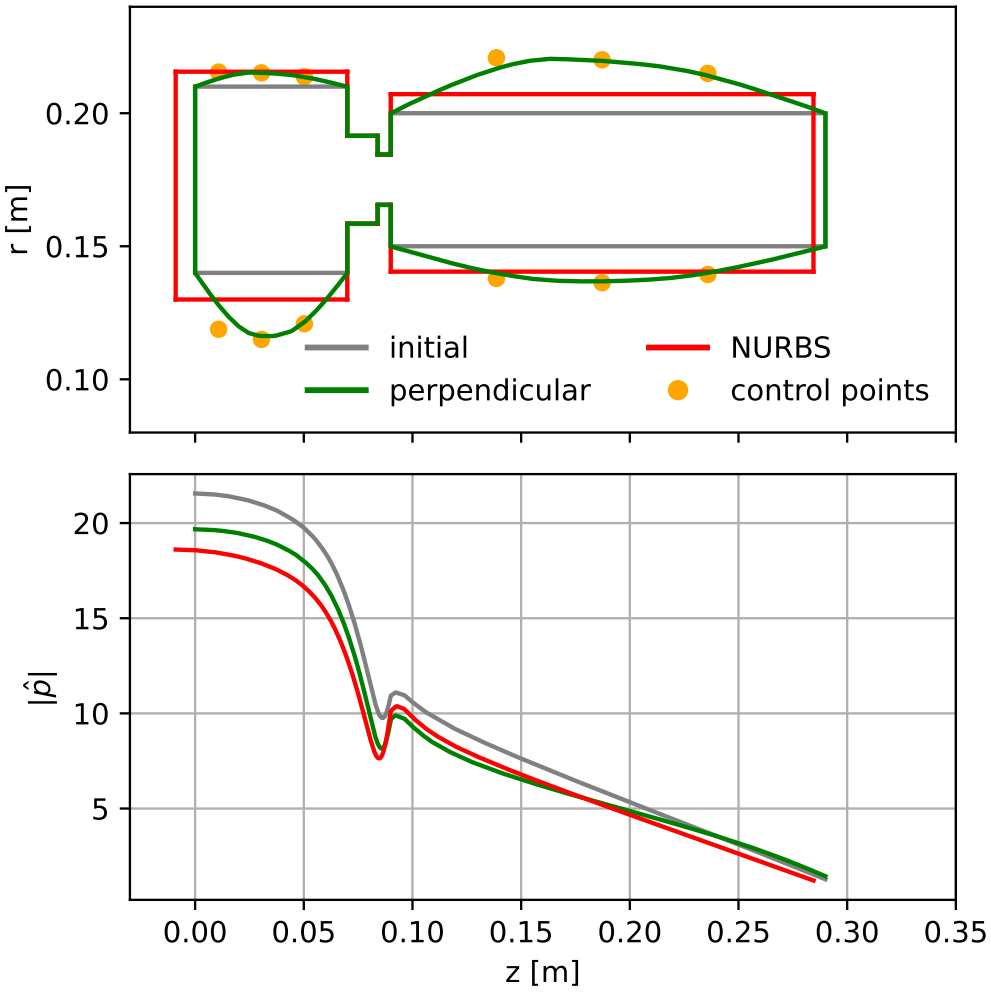


Fig. 3.12 (Top) Section view of one sector of the MICCA combustor for the initial (black), perpendicular displacement (red) and the NURBS displacement (green) after applying the shape changes to the plenum and the combustion chamber to reduce the growth rate. The gray dashed line represents the longitudinal axis of the burner as shown in Fig. 2.12. (Bottom) Absolute value of the eigenvector of the first azimuthal mode of the MICCA combustor, along the dotted line in the top figure, before (gray) and after (red and green) the shape changes. The sector is that in which the pressure is maximum. The results of perpendicular displacements in this plot are recalculated with *helmholtz-x* and the results match the results in Sec 6.4 of [57].

easily demonstrated before the geometry has changed significantly. Table 3.2 tabulates the associated eigenvalues for the initial geometry and the two final geometries. The size of the plenum has increased for both approaches. The only boundary that has moved inwards is the combustion chamber outlet. Fig. 3.12 (bottom) shows the modulus of the corresponding

Table 3.2 Degenerate eigenfrequencies of the MICCA combustor for the initial and the two optimized designs. The units of the eigenfrequencies are in rad s^{-1}. The eigenvalues for each case get closer as number of cells increase (not shown here).

Case	$\omega_{1,r}$	$\omega_{1,i}$	$\omega_{2,r}$	$\omega_{2,i}$	$\angle\hat{q}$ - $\angle\hat{p}$ [deg]
initial	3222.77	517.51	3222.50	518.66	56.83
perpendicular	3085.27	402.92	3084.97	401.03	65.37
NURBS	3152.32	413.42	3152.19	414.30	62.10

pressure eigenvectors along the dotted line at $r = 0.175$ m, where the absolute value of the eigenvector is maximal.

We observe that changes applied to the shape of the combustion chamber have little or no effect on the angular frequency, which is expected because this mode is a plenum mode. We also observe that the sensitivity of the eigenvalue growth rate in the plenum is higher than in the combustion chamber. This can be explained by the fact that the gas is cooler, so the local wavelength is shorter and therefore geometry modifications of a given size have more influence.

3.6.2 Mixed Mode

In this subsection, we find an unstable mixed tangential–longitudinal thermoacoustic eigenmode of the MICCA. We perform shape sensitivity analysis for this eigenmode and we reveal the optimized designs using perpendicular surface changes and control point movements with symmetry-preserving changes. Fig. 3.13 shows the absolute value of the eigenvector of the mixed mode. The corresponding eigenfrequency is 1663.60 Hz with 377.19 rad s^{-1} growth rate.

Figure 3.14 shows the real and the imaginary part of the shape derivatives for the plenum and the combustion chamber for the mixed mode. These are shown as perpendicular boundary movements proportional to the shape derivatives of the surfaces of the combustor. Figure 3.15 shows the real and the imaginary part of the shape derivatives for the unstable mixed mode. The shape derivatives of the control points on the $+xz$ plane ($k_2 = 0$) are computed. These are depicted as control point displacements proportional to the shape derivatives in the outward normal direction of the combustor boundary. Fig. 3.16 (top) shows the initial and the final shapes of a sector of the annular combustor. The most influential region on the growth rate is found to be the combustion chamber for the mixed mode. This is because the circumferential and axial modes combine in the combustion chamber. The contribution of the axial component of the mixed mode causes higher sensitivity on the outlet surface because

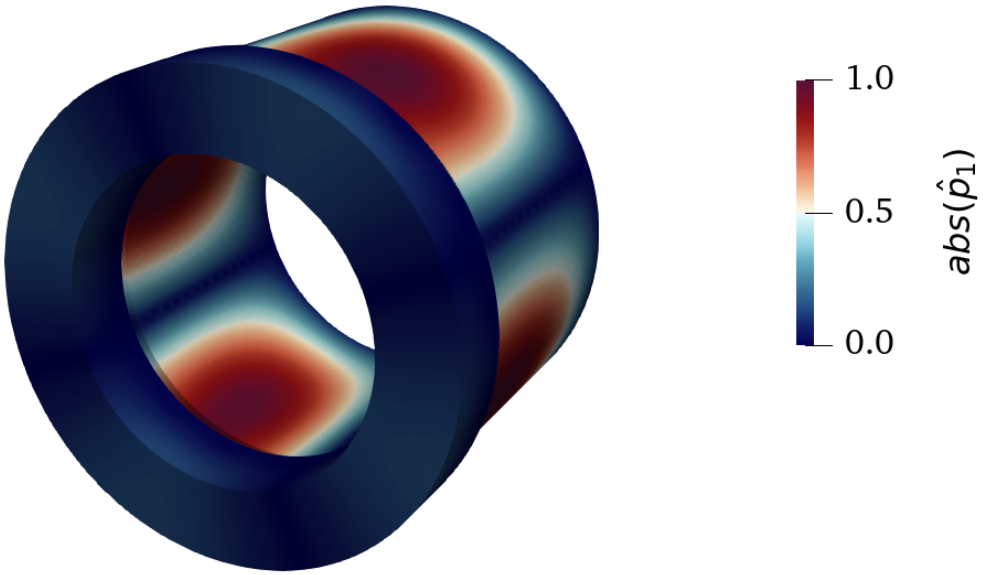


Fig. 3.13 Normalized magnitude of the eigenvector of the mixed mode. The associate eigenvalue is $\omega = 10830.42 + 401.16i$ rad s^{-1}.

it depends on the length of the combustion chamber. Table 3.3 tabulates the associated eigenvalues for the initial geometry and the optimized geometries. The same step size, 0.025 is chosen to perform shape changes with control points or boundary displacements. After 4 iterations, the growth rate has reduced by 32% with perpendicular boundary changes and 47% with control point displacements. The NURBS approach provides more stabilization for this unstable mixed mode considering same step size.

Table 3.3 Degenerate eigenfrequencies of the mixed mode for the initial and the two optimized designs. The units of the eigenfrequencies are in rad s^{-1}. N_c denotes the number of finite element cells for each case. The eigenvalues for each case get closer as N_c increases (not shown here).

Case	$\omega_{1,r}$	$\omega_{1,i}$	$\omega_{2,r}$	$\omega_{2,i}$	N_c	$\angle\hat{q}$ - $\angle\hat{p}$ [deg]
initial	10830.42	401.16	10831.18	402.10	13.60M	-15.94
perpendicular	10552.63	274.20	10552.97	274.83	11.78M	-19.70
NURBS	10322.15	211.83	10322.04	211.76	11.74M	-23.73

3.6.3 Dephasing mechanism

We can physically explain the dephasing mechanism using the (generalized) Rayleigh criterion. This says that a thermoacoustic system is unstable if the average of the product

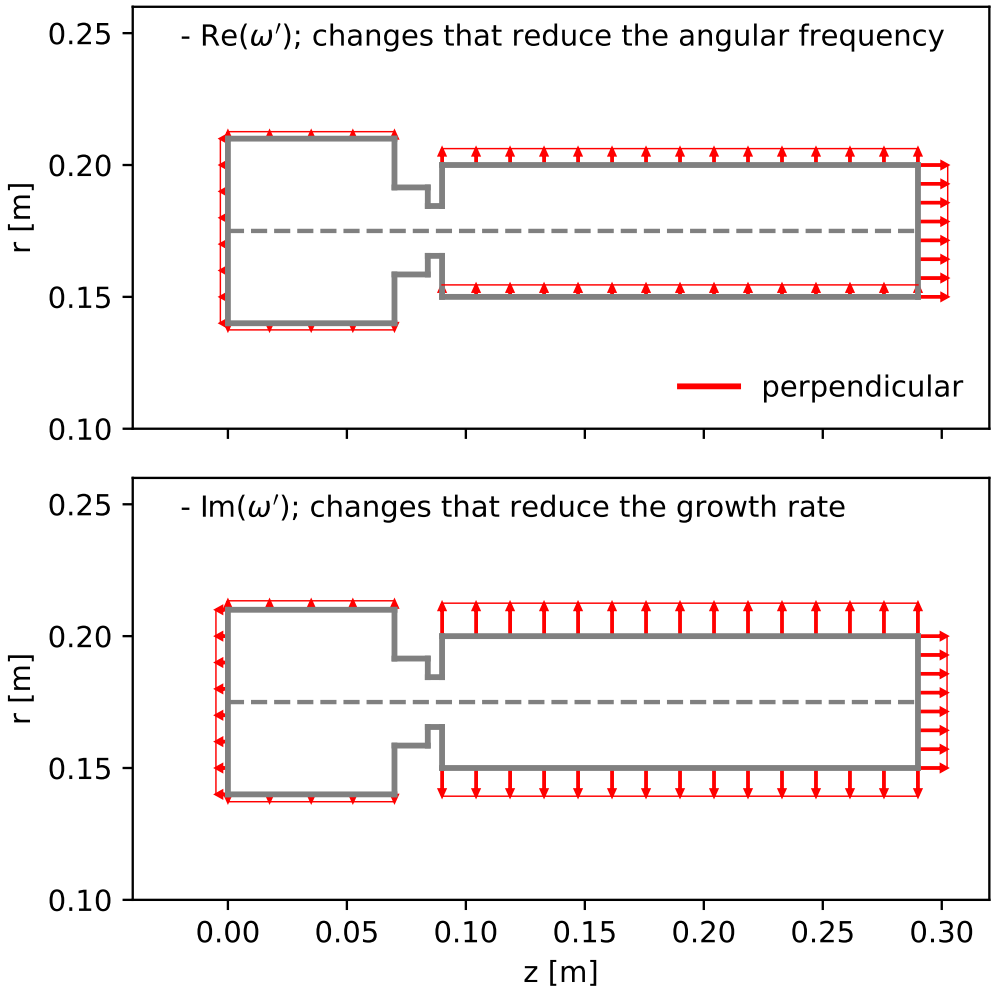


Fig. 3.14 As for Fig. 3.10 but for the mixed mode.

$p'\dot{q}'$ over the volume and over one period of oscillation is greater than the average of the losses due to acoustic dissipation from the system boundaries (Eq. (1.1)). We only focus on shape sensitivity of eigenvalues with Neumann and Robin boundary conditions. Internal acoustic losses could be included, but this would not change the analysis regarding shape sensitivity of eigenvalues. If we expand the Rayleigh index, $p'\dot{q}'$, we see that it depends on the modulus of the pressure in the flame volume, $|\hat{p}|$, and of the component of the velocity at the reference point in the reference direction, $|\hat{u}(x_r) \cdot n_r|$. Hence it depends on the phase angle difference between the heat release rate perturbation and the pressure perturbation as well as on the amplitude of the FTF. If we expand the acoustic energy flux, $p'u'$, we see that this only depends on the amplitude of the acoustic pressure at the boundary. On the other hand, the last columns of Table. 3.2 and Table 3.3 show the averaged phase difference

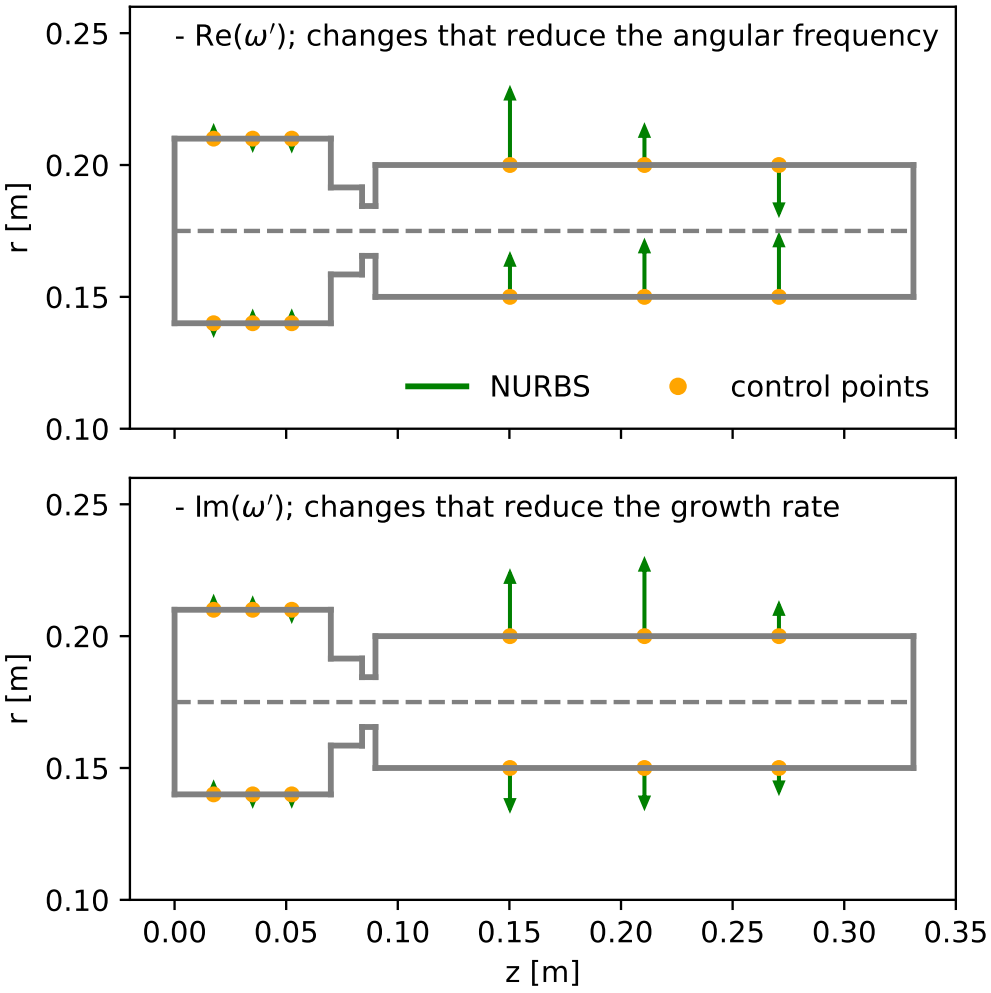


Fig. 3.15 As for Fig. 3.11 but for the mixed mode.

between the fluctuating heat release rate and the pressure perturbation for all sixteen burners for the azimuthal mode and the mixed mode, respectively. The phase of the acoustic pressure can be directly obtained from the calculated eigenvector, $\hat{p}$. For each burner we pick the centre point of the heat release zone (Fig. 3.6b) and calculate the phase angles of $\hat{q}$ and $\hat{p}$ at that point, then take the difference to obtain the phase difference. We see from Table 3.2 and Table 3.3 that, by changing the geometry, the algorithm increases the phase difference between the pressure, $\hat{p}$, and the heat release rate, $\hat{q}$. This reduces the heat converted to work each cycle and therefore reduces the growth rate of the system. We do not change the acoustic pressure within the flame volume and the acoustic velocity along the reference direction at the reference point. They would change, however, if we were to change the shape of the injector. The most sensitive component is the burner. As seen in [82], the changes

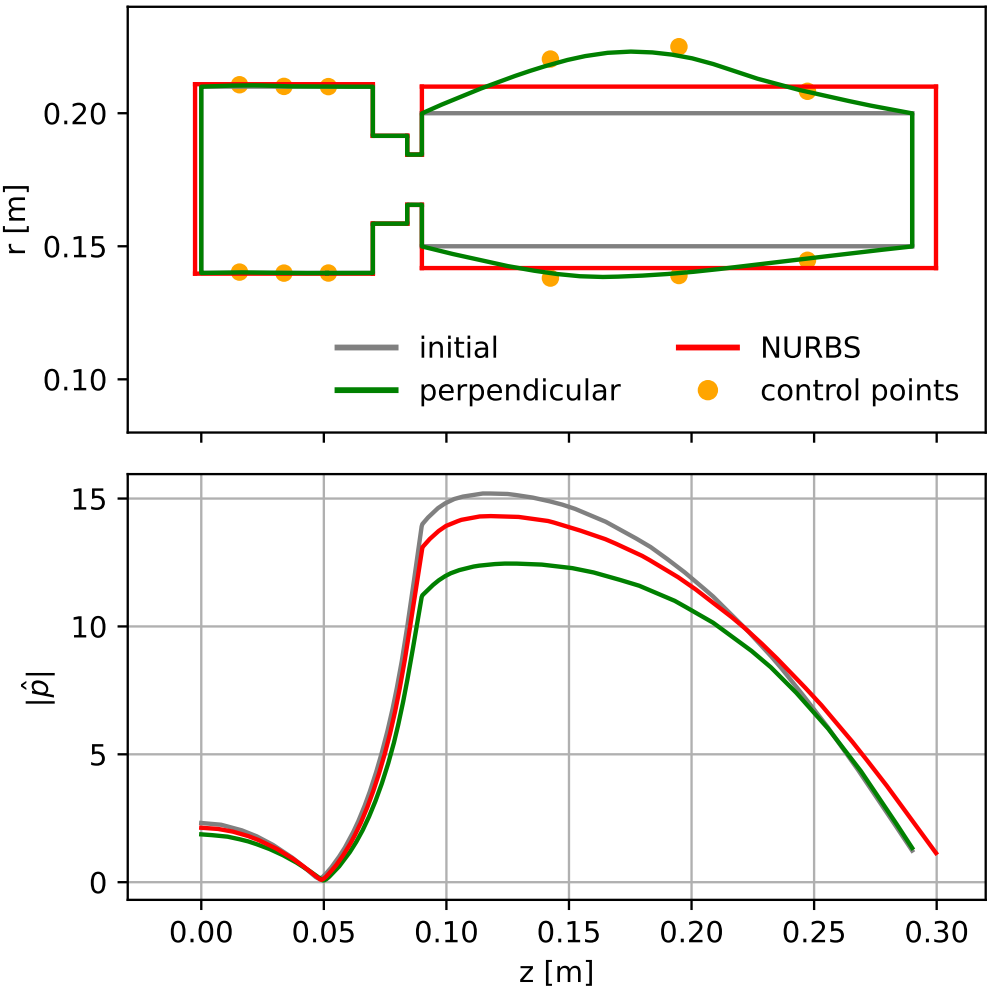


Fig. 3.16 As for Fig. 3.12 but for the mixed mode.

in bluff body geometry are taken into account, showing that even small changes influence the flame transfer function. Similarly, the algorithm would try to decouple the plenum and the combustion chamber by expanding the diameter of the burner, d_b, and contracting the diameter of the perforated plate, d_{pp}. Consequently it would reduce the amplitude of the acoustic velocity at the reference point and of the acoustic pressure within the flame volume.

This analysis shows how to significantly reduce the growth rate of thermoacoustic oscillations by modifying the geometry of a combustion chamber and plenum. In most practical devices, it is possible to alter the geometry of the plenum without extensive redesign. On the other hand, altering the geometry of the combustion chamber is less feasible due to constraints such as cooling requirements and high-altitude relight.

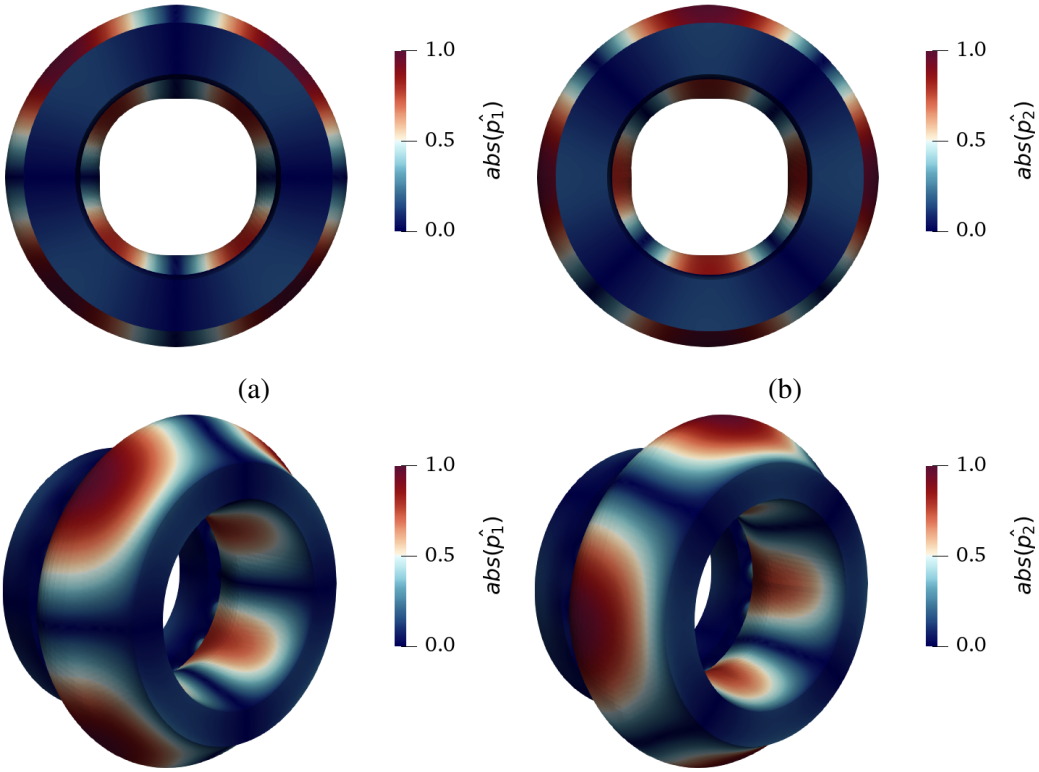


Fig. 3.17 Normalized magnitude of the mixed mode eigenvectors of the non-axisymmetrically deformed MICCA for $\varepsilon = 0.07$. The associate eigenvalues are $\omega_1 = 9709.58 + 100.72i$ rad s^{-1} (left column) and $\omega_2 = 9465.52 + 101.27$ rad s^{-1} (right column).

3.7 Symmetry-breaking Changes

In this section, we consider non-axisymmetic changes in the MICCA geometry for the unstable mixed mode. Compared to the axisymmetric changes in Sec. 3.6, we enlarge the deformation step size, ε, to better observe the effects of non-axisymmetry. Each control point of the NURBS geometry is moved in the direction provided by its shape derivative. The resulting eigenvectors can be seen in Fig. 3.17. We observe that the NURBS geometry is extended more in the outward normal direction of the control points on the pressure node of the first mixed eigenmode, $\hat{p}_1$. Due to the non-axisymmetric changes in the NURBS geometry, eigenvalue degeneracy is lost during symmetry-breaking changes.

From Fig 3.18, eigenvalue splitting is observed between two unstable mixed modes for different deformations. The changes in the eigenvalues follow two different branches. Starting from the degenerate case where $\varepsilon = 0$, the non-axisymmetric changes split both angular frequencies and growth rates. For each deformation case, the magnitude of the

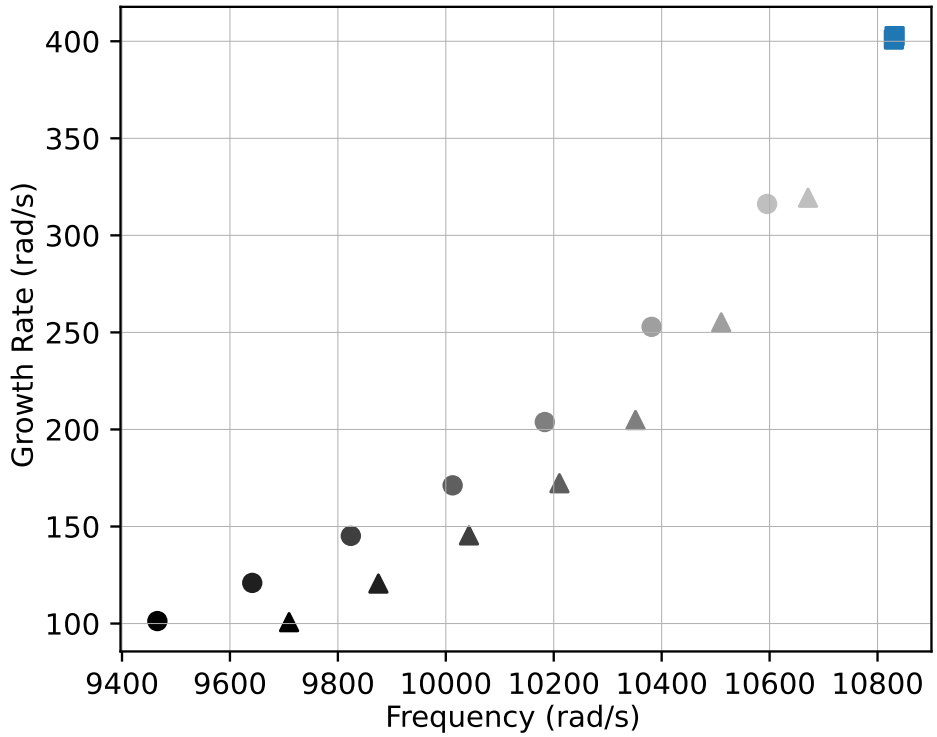


Fig. 3.18 Splitted eigenvalues of the $\hat{p}_1$ (triangles) and $\hat{p}_2$(circles) for different deformation sizes (ε). Blue square marker represents the degenerate configuration before perturbation. The lightest grey corresponds to $\varepsilon_0 = 0.01$ and the darkest to $\varepsilon_7 = 0.07$. Same triangle-circle couples correspond same deformation size.

frequency splits are much greater than those of the growth rate. For the most deformed case ($\varepsilon = 0.07$), the frequency difference between the two unstable modes reaches 244.06 rad s^{-1} whereas the growth rate varies by 0.55 rad s^{-1}.

As we follow the deformation directions provided by the shape derivatives, the splitted modes become more stable compared to the initial geometry, as expected. When compared to the axisymmetric deformations with the same deformation size ε, one mode becomes more stable and other mode becomes less stable for symmetry-breaking modifications in the geometry.

3.8 Conclusions

In this chapter, we introduce NURBS as a shape parametrization tool for thermoacoustic systems. We first show NURBS parametrization of simple geometries: a circle and a cylinder.

We perform a Taylor test to compare the shape gradients with adjoint methods to the ones calculated with finite differences. We then combine *helmholtz-x* with an adjoint-based shape optimization to reduce the growth rate of the linearly unstable thermoacoustic modes in a symmetric academic annular combustor. We parametrize the entire combustor geometry with NURBS control points. The modes are azimuthal and mixed, and therefore two-fold degenerate. We obtain the Hadamard form of the shape derivative of the eigenvalue. We apply both symmetry-preserving and symmetry-breaking changes to the MICCA geometry with NURBS and show that these can be used to reduce the thermoacoustic growth rate efficiently. This process could be continued to zero growth rate, but large geometry changes would be required. After modifying the shape, we use this analysis to reveal the physical mechanism that causes the growth rate's reduction. The shape modifications increase the phase difference between the pressure and the heat release rate fluctuations. The most influential component on thermoacoustic stability, however, is the burner. We use an experimentally-determined flame transfer function, which is fixed because we do not know how it would change with the burner geometry.

To summarize, this chapter demonstrates how to calculate the shape derivatives and iteratively modify the shape of a thermoacoustic system with NURBS parametrization using *helmholtz-x*. We explain the effect on the eigenvalues of both symmetry-breaking and symmetry-preserving shape changes and how the eigenvalue growth rate can be reduced. The shape optimization procedure in this chapter can now be extended to more complex geometries such as those on aircraft gas turbines.

Chapter 4

Geometry parametrization with FFD

The geometry of the thermoacoustic system was parametrized with NURBS surfaces in chapter 3. However, manipulation of more complex geometries requires extra care and we would like the parametrization approach to be applicable to any combustor shape. For this reason, we investigate the free form deformation technique to control the combustor geometries. In this chapter, we present the established free form deformation technique [49].

Sec. 4.3 and Sec. 4.4 of this chapter is accepted for publication at the conference *ASME Turbo Expo 2024* [55]. The content in Sec. 4.5 is to be submitted for journal publication in *Journal of Engineering for Gas Turbines and Power* [56].

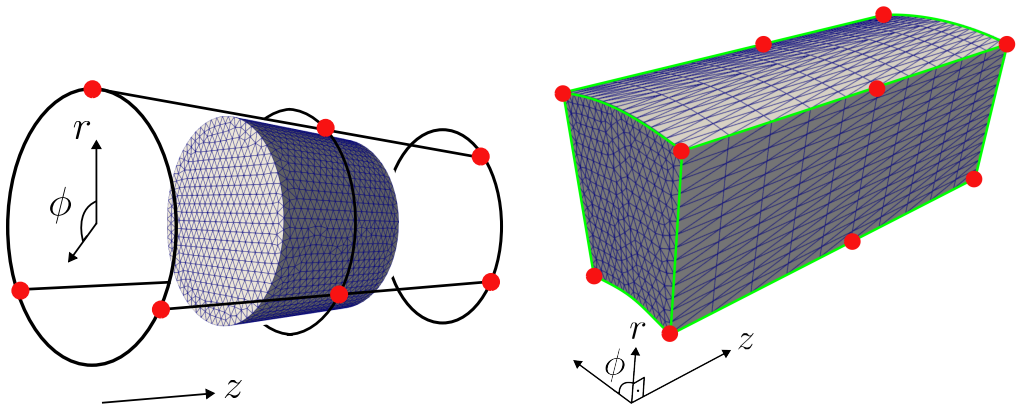
4.1 Free Form Deformation

Free form deformation creates a parametrization link between the mesh nodes and some control points. These control points form the control lattice (Fig. 4.1) and that lattice can take any geometric shape. In general, cylindrical or cube-shaped lattices are preferred in order to manipulate the control points more conveniently.

Any point within the control lattice can be represented by parametric coordinates (s,t,u) as in Eq. (4.1), where $\mathbf{X}_0$ denotes the center of the FFD lattice and $\mathbf{S}$, $\mathbf{T}$ and $\mathbf{U}$ are the parametric unit vectors in the the radial, circumferential and axial directions, respectively.

$$\mathbf{X} = \mathbf{X}_0 + s\mathbf{S} + t\mathbf{T} + u\mathbf{U} \tag{4.1}$$

Considering the lattice in Fig. 4.1, the mesh nodes are initially transformed into cylindrical coordinates and then their parametric coordinates are calculated with Eq. (4.1). The range of the parametric coordinates is between 0 and 1 for the radial (r) and axial (z) directions and between 0 and 2π for the azimuthal direction (ϕ).



(a) (b)

Fig. 4.1 Free form deformation (FFD) configurations with (a) cylindrical and (b) angular control lattices with control points (red dots). r, ϕ and z denotes the radial, circumferential and axial directions. (a) Blue and (b) green lines visualize the connections between the control points.

The position of the FFD control points can be arbitrarily defined according to the application. We usually specify the positions of the control points in a equispaced pattern within a control lattice using Eq. (4.2).

$$\mathbf{P}_{ijk} = \mathbf{X}_0 + \frac{i}{l}\mathbf{S} + \frac{j}{m}\mathbf{T} + \frac{k}{n}\mathbf{U} \tag{4.2}$$

where l, m, n specify the number of control points in the r, ϕ, z axes. The locations and number of FFD control points are crucial because they form the control lattice and determine the allowable deformation magnitudes and directions. Therefore, the control points should be numbered and positioned to prevent or minimize potential overlapping deformations resulting from control point displacements. For simple or symmetric geometries, equispaced control points can manage the deformations. However, for complex shapes, an irregular pattern for the placements might work better depending on the aim of the application. For instance, control points forming a cylindrical lattice could handle cylinder-like geometries better, whereas cornered geometries might benefit from box-like lattices with control points positioned at the corners.

After specifying the positions of the FFD control points in the lattice and calculating the parametric coordinates of the mesh nodes, we are then ready to deform the geometry. We first displace the positions of the FFD control points and deform the mesh nodes individually with trivariate Bernstein basis polynomials, as shown in Eq. (4.3).

$$\begin{aligned}\mathbf{X}_{FFD} = &\Bigg(\sum_{i=0}^{l} \binom{l}{i} (1-s)^{l-i} s^i \\ &\Big(\sum_{j=0}^{m} \binom{m}{j} (1-t)^{m-j} t^j \\ &\Big(\sum_{k=0}^{n} \binom{n}{k} (1-u)^{n-k} u^k \mathbf{P}_{ijk} \Big) \Big) \Bigg)\end{aligned} \tag{4.3}$$

In shape derivative calculations, the displacement field $\mathbf{V}_{ijk}$ is required for the control point $\mathbf{P}_{ijk}$. Taking the derivative of the mesh nodes with respect to the control point gives the displacement field, as shown in Eq. (4.4).

$$\begin{aligned}\frac{\partial}{\partial \mathbf{P}_{ijk}} \Big(\mathbf{X}_{ffd} \Big) = \mathbf{V}_{ijk} = &\Bigg(\sum_{i=0}^{l} \binom{l}{i} (1-s)^{l-i} s^i \\ &\Big(\sum_{j=0}^{m} \binom{m}{j} (1-t)^{m-j} t^j \\ &\Big(\sum_{k=0}^{n} \binom{n}{k} (1-u)^{n-k} u^k \Big) \Big) \Bigg)\end{aligned} \tag{4.4}$$

The displacement field $\mathbf{V}_{ijk}$ can then be used in Eq. (4.5) for calculating the shape derivative of the control point $\mathbf{P}_{ijk}$.

4.2 Shape Derivatives

With direct and adjoint eigenvectors, we can compute the shape derivative of the FFD control points in Hadamard-form. The most general expression for the shape derivative is that using Robin boundary conditions:

$$\begin{aligned}\omega'_{ijk} = \int_{\Gamma} \mathbf{V}_{ijk} \cdot \mathbf{n}_{ijk} \Bigg(&- \hat{p}_1^{\dagger *} \Big(\kappa c^2 \frac{\partial c}{\partial n} \Big) \frac{\partial \hat{p}_1}{\partial n} + \nabla \cdot \Big(\hat{p}_1^{\dagger *} c^2 \nabla \hat{p}_1 \Big) \\ &- 2 \frac{\partial \hat{p}_1^{\dagger *}}{\partial n} c^2 \frac{\partial \hat{p}_1}{\partial n} \Bigg) dS\end{aligned} \tag{4.5}$$

where ω'_{ijk} is the complex-numbered shape derivative for the FFD control point $\mathbf{P}_{ijk}$ and $\mathbf{n}_{ijk}$ is its outward normal vector. When applying Neumann boundaries, we impose $\partial \hat{p}_1 / \partial n = 0$

and $\partial \hat{p}_1^{\dagger}/\partial n = 0$. For degenerate cases, we use Eq. (3.9), where we impose $C = \mathbf{V}_{ijk} \cdot \mathbf{n}_{ijk}$. The shape derivative of any control point is

$$\omega'_{ijk} = \int_{\partial\Omega} \mathbf{V}_{ijk} \cdot \mathbf{n}_{ijk} \left(\nabla \cdot \left(\hat{p}_1^{\dagger *} c^2 \nabla \hat{p}_1 \right) \right) dS \tag{4.6}$$

for Neumann boundary conditions. The *helmholtz-x* implementation of Eq. (4.6) is shown in Listing 4.1.

```
1  def shapeDerivativesFFD(geometry, lattice, physical_facet_tag,
       omega_dir, p_dir, p_adj, c, acousticMatrices, FlameMatrix):
2      normal = FacetNormal(geometry.mesh)
3      ds = Measure('ds', domain = geometry.mesh, subdomain_data =
       geometry.facet_tags)
4      p_adj_norm = normalize_adjoint(omega_dir, p_dir, p_adj,
       acousticMatrices, FlameMatrix)
5      p_adj_conj = conjugate_function(p_adj_norm)
6      G_neu = div(p_adj_conj * c**2 * grad(p_dir))
7      derivatives = {}
8      i = lattice.l-1
9      for zeta in range(0,lattice.n):
10         derivatives[zeta] = {}
11         for phi in range(0,lattice.m):
12             V_ffd = ffd_displacement_vector(geometry, lattice,
       physical_facet_tag, i, phi, zeta, deg=1)
13             shape_derivative_form = form(inner(V_ffd, normal) * G_neu
        * ds(physical_facet_tag))
14             eig = assemble_scalar(shape_derivative_form)
15             derivatives[zeta][phi] = eig
16     return derivatives
```

Listing 4.1 *helmholtz-x* code for computing shape derivatives of FFD control points. Line 6 represents the UFL form of Eq. (4.6). In the radial direction, we only compute the shape derivatives of the control points on the lateral surface. Between lines 9 and 15, we loop over the control points in the azimuthal (ϕ) and axial (z) directions, respectively.

We calculate the shape gradient aligned with the outward normal vector of the relevant FFD control point. The physical interpretation of the complex-valued FFD shape derivatives are shown in Fig. 4.2 with example design changes to reduce the growth rate of the eigenvalue. In summary, the main steps of the adjoint based shape optimization method with FFD control points are similar to those in NURBS:

- the three dimensional numerical grid is generated;

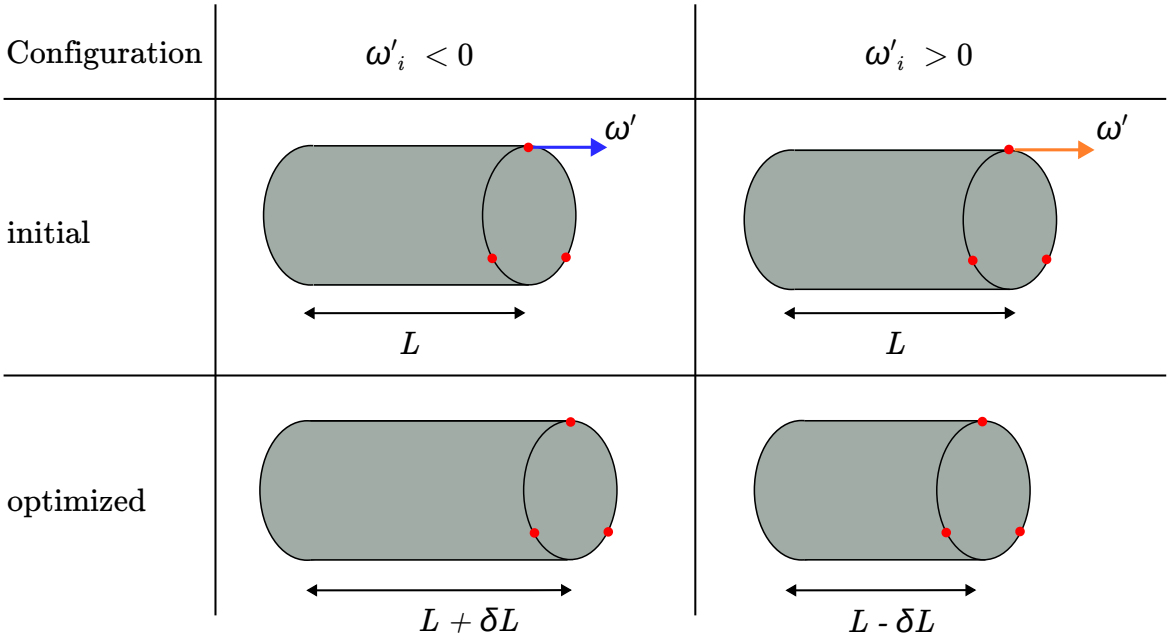


Fig. 4.2 Design changes that improve stability of the thermoacoustic system. ω' is the complex-valued shape derivative of the corresponding FFD control point. In scenarios in which the mode is unstable and the imaginary portion of the shape derivative at the control point has a negative sign, moving the control point along the outward normal vector direction improves system stability.

- the FFD lattice and its control points are defined after calculating the parametric coordinates of the nodes in the grid;
- direct and adjoint eigenmodes are calculated with P2 (degree 2) finite elements;
- the shape derivatives of the FFD control points are calculated and normalized;
- the shape is deformed in line with the direction provided by the normalized shape derivatives, with a certain step size.

4.3 Rijke Tube

4.3.1 Geometry and FFD Setup

We start with the canonical example of thermoacoustic instability, the Rijke tube. We consider a cylinder with length 1.0m and diameter 0.047m, as in [43]. The three-dimensional mesh is generated with $25,246$ elements using Delaunay-triangulation by Gmsh [72]. The global FFD setup for the Rijke tube can be seen in Fig. 4.3. We place more control points in the axial direction in order to increase our control over the tube geometry.

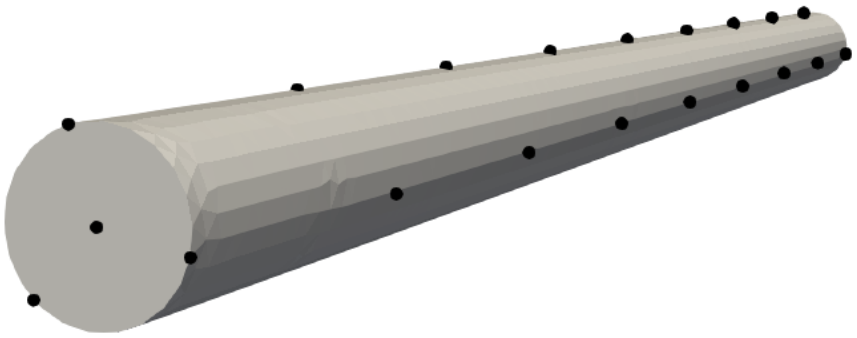


Fig. 4.3 Control points (black dots) and external surface (grey) of the Rijke tube. There are 2, 3 and 9 control points in the radial, circumferential and axial directions respectively.

4.3.2 Eigenmode

Based on the dimensional parameters in Table 2.3, the first axial unstable eigenmode of the Rijke tube is computed. The corresponding eigenfunction is shown in Fig.4.4 with the eigenfrequency of $184.101 + 0.137i\ \mathrm{s}^{-1}$, which agrees with [43].

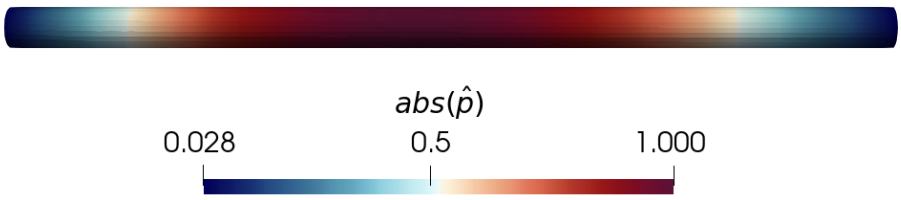


Fig. 4.4 Normalized amplitude of the first axial mode of the Rijke tube. The corresponding eigenfrequency is $\omega/2\pi = f = 184.101 + 0.137i\ \mathrm{s}^{-1}$

4.3.3 Shape Modification

After obtaining the direct and adjoint eigenmodes of the Rijke tube, we calculate the shape derivatives of the control points on the lateral (Neumann) surface using Eq. (4.5). We prohibit axial shape changes and only allow radial displacements of the control points. As an example, the radial displacement field for the control control point $\mathbf{P}[2,2,5]$ is shown in Fig. 4.5.

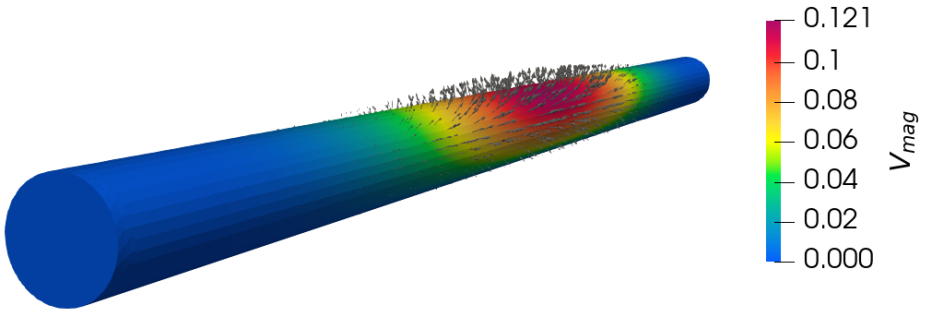


Fig. 4.5 Deformation field of the Rijke tube for the control point $\mathbf{P}[i,j,k]$ where $i = 2$, $j = 2$, $k = 5$. The colormap shows the magnitude of the displacement field. Some of the magnitude-scaled outward normal vectors of the mesh nodes are visualized on the lateral surface.

We iterate over the control points on the lateral boundary and move them individually in the direction indicated by the shape gradients. The resulting deformed geometry of the Rijke tube is shown in Fig. 4.6. The growth rate of the deformed design become negative after a few deformations. The trend of the growth rate due to deformation is similar to that in [7]. The example in this thesis, however, allows radii changes for the inlet and outlet circular surfaces.

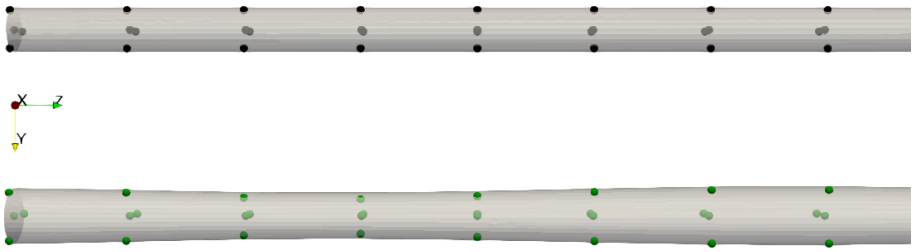


Fig. 4.6 optimized geometry of the Rijke tube. The eigenfrequency after free form deformation is $\omega/2\pi = f = 198.437 - 0.431\ \mathrm{s}^{-1}$. The black (top) and green (bottom) dots are the initial and final positions of the FFD control points for the initial (top) and final (bottom) geometries after few deformations.

4.4 Academic LPP Combustor

4.4.1 Geometry and FFD Setup

For this subsection, we extend the shape optimization procedure to a more complex combustor geometry. We consider the lean premixed prevaporized (LPP) aeroengine combustor in [83, 37]. The sector view of this combustor is shown in Fig. 4.7 with dimensions of the annular geometry. It has a plenum, 20 premixed ducts and a combustion chamber, as well as cylindrical flame volumes.

The cylindrical FFD setup for the LPP is visualized in Fig. 4.8. For this annular geometry we generate two local cylindrical lattices, to increase the control over the plenum and combustion chamber geometries separately. The numbers of localized FFD control points are tabulated in Table 4.1. We place more control points over the combustion chamber because it is longer than the plenum. Similar to [45], we generate the annulus mesh for the plenum and combustion chamber geometries first, and then define a premix duct. Next, we copy and rotate that premix duct geometry 20 times and fuse the geometries. Lastly, we generate a three-dimensional unstructured mesh with 137,060 elements and optimize it using the Netgen optimizer [72].

Table 4.1 FFD parameters of the control points for the annular combustor case

Volume	l	m	n
plenum	3	3	3
combustion chamber	3	3	6

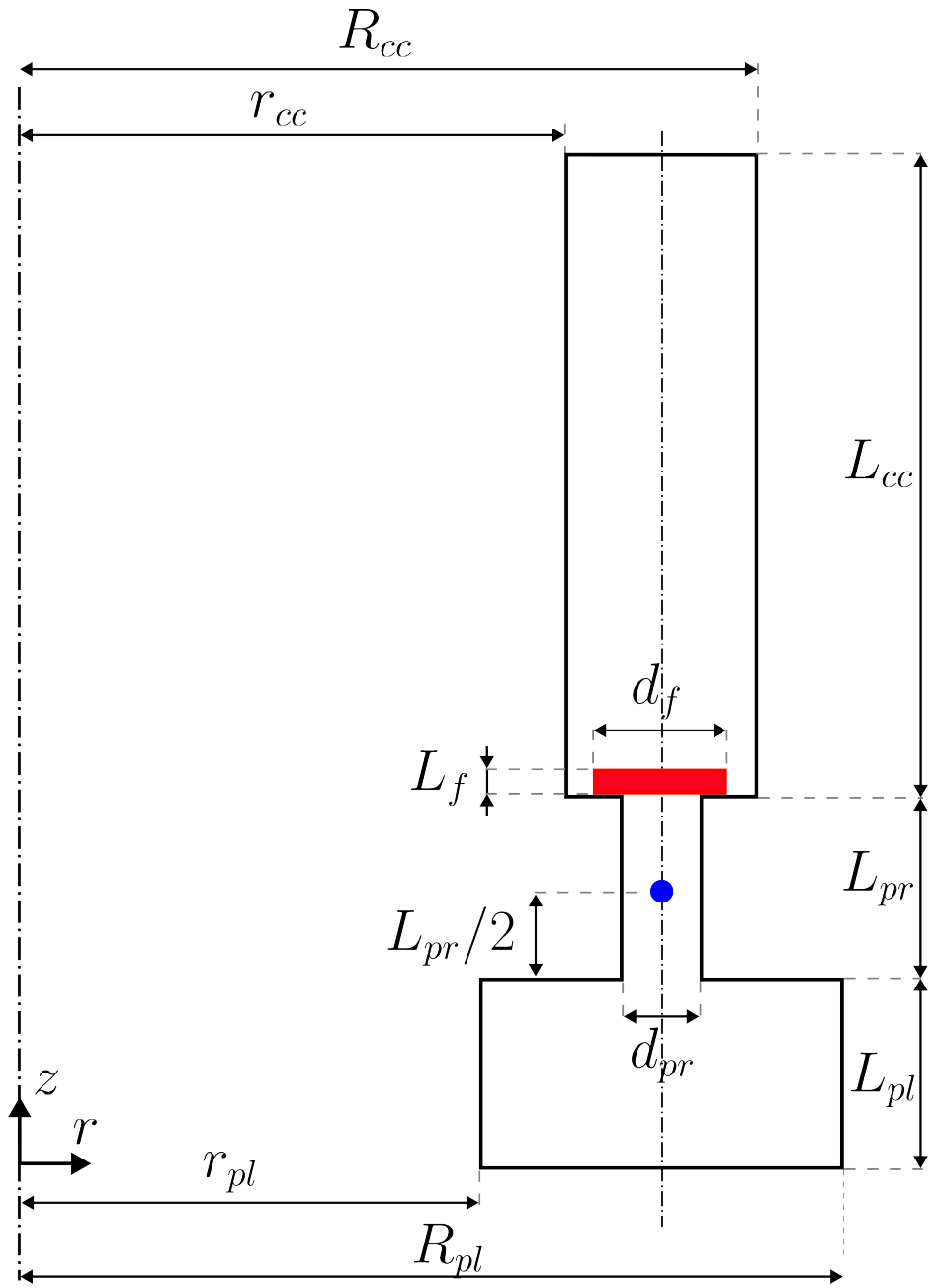


Fig. 4.7 A section of one sector of the LPP combustor with r_{pl} = 0.22 m, R_{pl} = 0.38 m, L_{pl} = 0.1 m, d_{pr} = 0.03568 m, L_{pr} = 0.1 m, d_f = 0.072 m, L_f = 0.012 m, r_{cc} = 0.25 m, R_{cc} = 0.35 m, L_{cc} = 0.3 m. The red zone represents the cylindrical flame volume and the blue circle denotes the position of the Dirac delta measurement function. The vertical dashed axis represents the longitudinal axis of the burner.

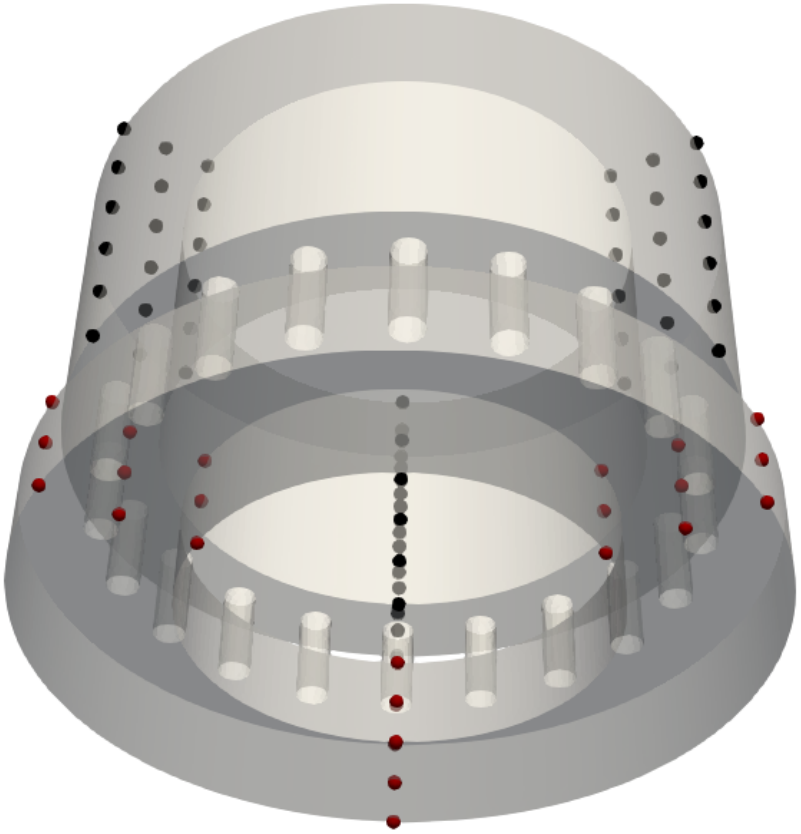


Fig. 4.8 Free form deformation configuration for the LPP combustor. The red and black dots represent the control points for the plenum and combustion chamber.

4.4.2 Parameters

The parameters for the LPP combustor are taken from Refs. [83, 37] and listed in Table 4.2. The mean temperature in the plenum and premix ducts is constant at $\bar{T} = 1000\text{K}$. In the combustion chamber the temperature profile is parabolically decreasing between the values at the flame position and the chamber outlet, as shown in (4.7).

$$\bar{T}(z) = \begin{cases} 1000, & \text{if } z < z_f \\ (1000 - 2500)\left(\frac{z - z_f}{L_{cc}}\right)^2 + 2500, & \text{otherwise} \end{cases} \tag{4.7}$$

The speed of sound field is computed from the temperature distribution. We set a volumetric heat release rate within the red volume sketched in Fig. 4.7, in which each of the burners has an equal power ($q_0/20$). The heat release rate model is a $n - \tau$ model [71].

The inlet surface of the plenum and outlet surface of the combustion chamber are modelled as choked boundaries. The corresponding Mach numbers near the downstream and upstream

Table 4.2 Dimensional parameters of the annular combustor case.

Parameter	value	unit
r_{gas}	287.1	$\mathrm{Jkg^{-1}K^{-1}}$
$\bar{p}_{gas}$	50e5	Pa
$\bar{q}_0$	151.1	MW
$\bar{u}_b$	287.13	$\mathrm{m\ s^{-1}}$
n	4.0	-
τ	0.0015	s
γ	1.4	-
$\bar{M}_{in}$	0.03	-
$\bar{M}_{out}$	0.07	-

ends of these boundaries are listed in Table 4.2. The reflection coefficients are calculated using Eq. (2.17) and Eq. (2.18).

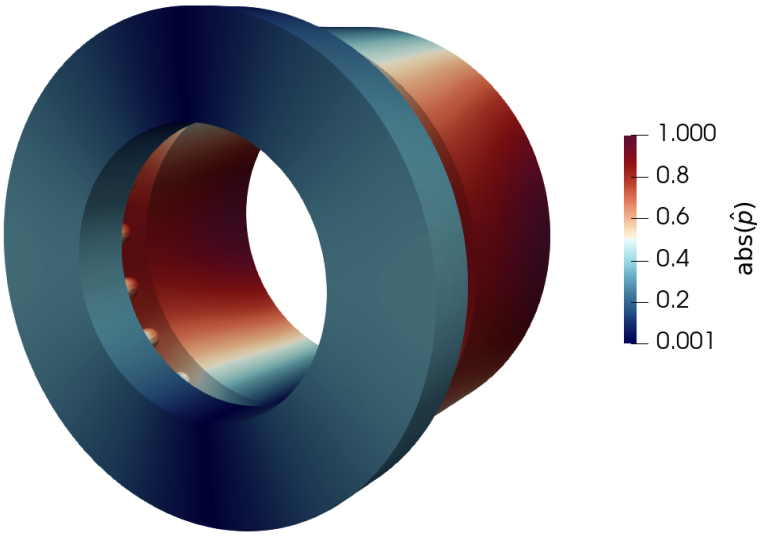
4.4.3 Eigenmode

We are interested in the dominant azimuthal thermoacoustic mode of the LPP combustion chamber. The eigenfrequency of this configuration is found to be 524.688 Hz, which is close to that found with the low order network code in [83], at 520Hz. The calculated mode shape shown in Fig. 4.9a is also very similar to that in [83]. However, the growth rate found with our Helmholtz solver is different from than found in [83, 37] because the heat release rate is not pointwise, as it is in the low order network models.

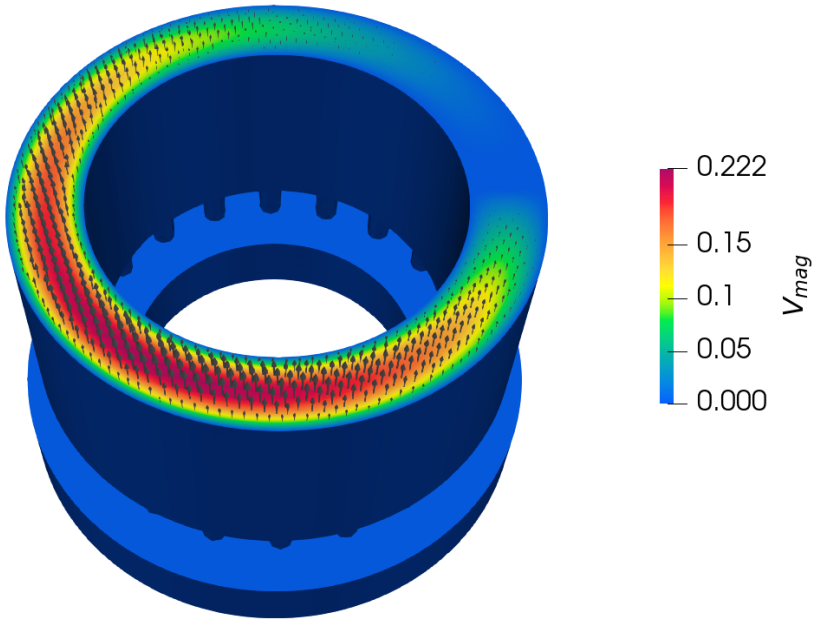
4.4.4 Shape Modification

We obtain the direct and adjoint degenerate eigenmodes of the LPP combustor. For the FFD case of the LPP geometry, we consider radius changes for the lateral surfaces and axial changes for the inlet and outlet choked boundaries. Symmetry-breaking changes are allowed. Our FFD framework does not take into account how geometry changes affect the flame behaviour. We therefore fix the premix duct geometry. We use Eq. (4.4) to compute the displacement field of the FFD control points. The example field representing the axial deformation of the FFD control point on the combustor outlet boundary is shown in Fig. 4.9b.

These shape derivatives inform the changes made to the FFD points to reduce the growth rate. We deform the geometry iteratively following these shape derivatives. The resulting 3D geometry is shown in Fig. 4.10a. Although the deformed geometry looks



(a)



(b)

Fig. 4.9 (a) Normalized amplitude of the chamber-dominant azimuthal mode. The corresponding eigenfrequency is $\omega = 3296.713 + 533.272i$ rad s^{-1}. (b) Deformation field of the LPP combustor for the control point $\mathbf{P}[i, j, k]$ where $i = 2$, $j = 2$, $k = 6$.

symmetrical, the rotational symmetry is slightly broken due to the differences between the shape derivatives along the circumferential direction. The eigenvalues of the deformed LPP geometry become $\omega_1 = 3251.7 + 405.2i$ and $\omega_2 = 3267.7 + 412.6i$ rad s^{-1}. The control points near the combustion chamber inlet are found to have more influence on the growth rate so the deformation magnitudes are large in that region. The smaller plenum together with bigger combustion chamber is more thermoacoustically stable than the original geometry.

The findings in this example agree with [37] for the changes in the combustion chamber geometry but not the plenum geometry. This might be due to the inclusion of mean flow effects in [37], which are not included in this example, or due to the compact flame assumption.

The control points lying on the burner axis in Fig. 4.10b are fixed, apart from the points on the choked boundaries. All the other control points are allowed to move. For the final geometry, the plenum volume becomes smaller while the combustion chamber volume is increased after FFD deformation. The thermoacoustic eigenvalues of the deformed LPP geometry have 24% lower growth rate than the initial case. This reduction could be increased by moving the points further.

4.5 Industrial Aeroengine Combustor

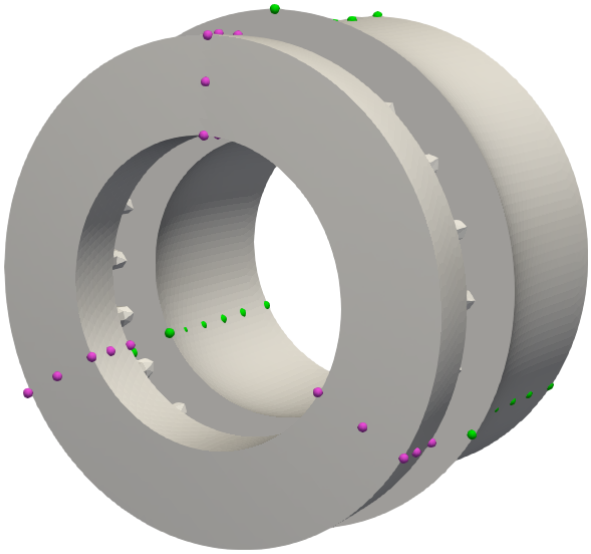
4.5.1 Geometry and FFD Setup

We start by simplifying the CFD geometry of the industrial annular combustor by removing cooling holes on the liner walls. In this example, we only deal with the axial eigenmodes. For that reason, we only consider a single sector of the full annulus. We generate 384,785 finite elements using Delaunay-triangulation. The global FFD setup for the combustor geometry can be seen in Fig. 4.11. We place 2, 3 and 4 FFD control points in the radial, circumferential and axial directions, respectively.

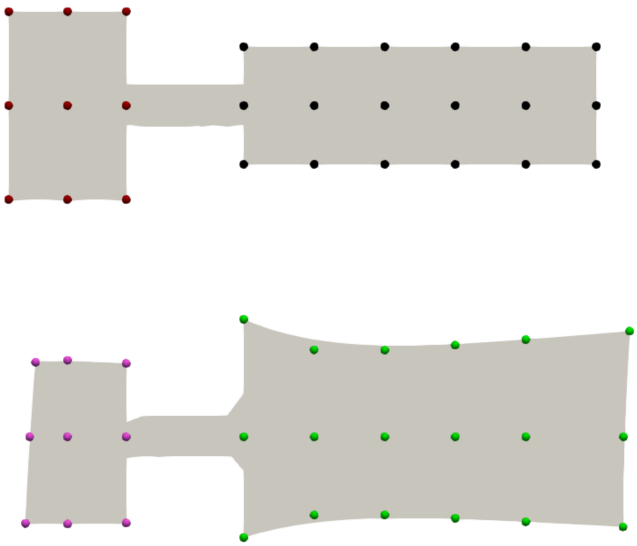
4.5.2 Parameters

The parameters of the thermoacoustic problem are tabulated in Table 4.3. The imposed volumetric heat release rate field is shown as in Fig. 4.11. This field integrates to 1 over the torus domain.

The measurement function distribution (Fig. 4.12a) is implemented by means of a three dimensional Gaussian function using Eq. (2.25). The relationship between the heat release rate and acoustic velocity at the measurement point is defined through an $n-\tau$ model. The temperature data of the combustor is taken from the network model and extrapolated onto



(a)



(b)

Fig. 4.10 The (a) 3D deformed geometry for the LPP combustor with its control points. The purple and green dots denote the positions of the FFD points after mesh deformation. The corresponding eigenvalues for deformed LPP geometry are $\omega_1 = 3251.7 + 405.2i$ and $\omega_2 = 3267.7 + 412.6i$ rad s^{-1}. The (b) sector slice of the initial (top) and deformed (bottom) geometry for the LPP combustor with its control points. The red and black dots denote the initial (top) positions of the FFD. The positions of the control points on the burner axis are kept unchanged, except the ones on the choked boundaries.

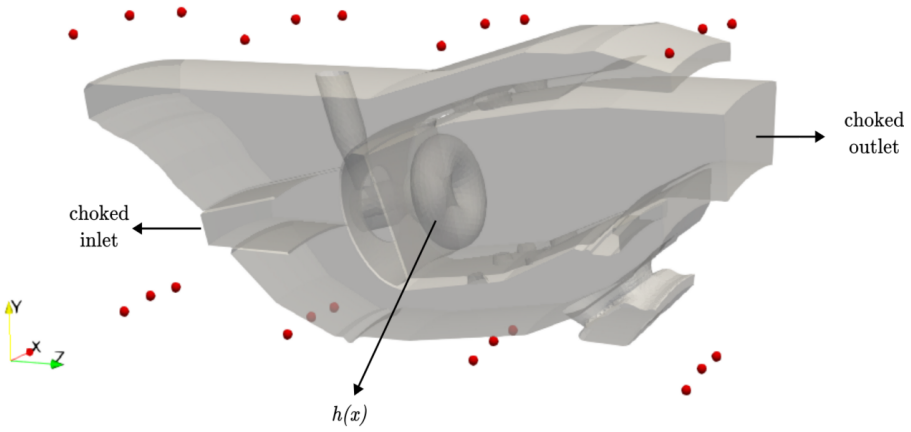


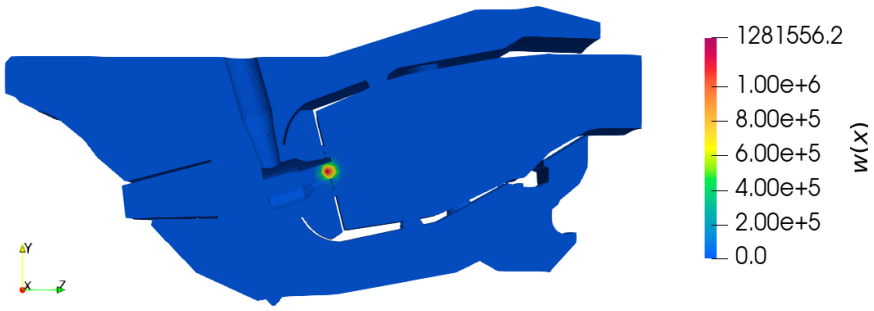
Fig. 4.11 FFD configuration of the simplified aeroengine combustor geometry. The red dots show the locations of the FFD control points of the angular lattice. The torus-like geometry in the combustion chamber represents the volumetric heat release rate field, $h(\mathbf{x})$. The choked boundaries are also shown.

the 3D numerical grid. The speed of sound field changes in the axial direction within the combustion chamber. Using the temperature field shown in Fig. 4.12b, the distribution of the speed of sound can be determined through the equation $c = \sqrt{\gamma r_{gas} T}$.

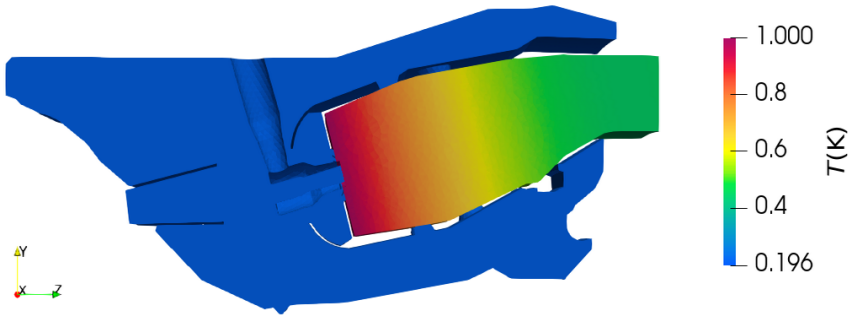
The specific heat capacity is assumed to change linearly with the temperature, $c_p(T) = 973.60 + 0.133T$. We compute the mean density field with the ideal gas model $p_{gas} = \rho_0 r_{gas} T$. We only consider one sector, so the heat release for this sector is $\bar{q}_{sec} = \bar{q}_{tot}/20$. Choked boundary conditions are imposed through the reflection coefficients for the inlet and outlet boundaries using Eqs. (2.17) and (2.18). The other boundaries are assumed to be perfectly reflecting (Neumann) surfaces.

4.5.3 Eigenmodes

Using this geometry, we show two unstable and one stable axial eigenmodes in Fig. 4.13. In this study, we only calculate shape derivatives of the most unstable eigenmode, shown in Fig. 4.13a. We first compute the eigenfrequency in Fig 4.13a without the flame response and call it f_{ref}. We then use this to divide the eigenvalues.

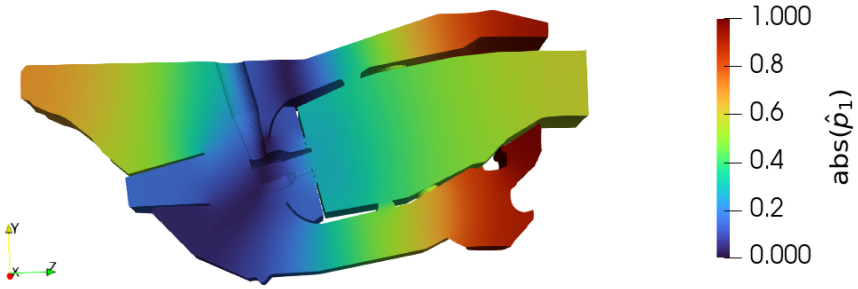


(a) Measurement function field.

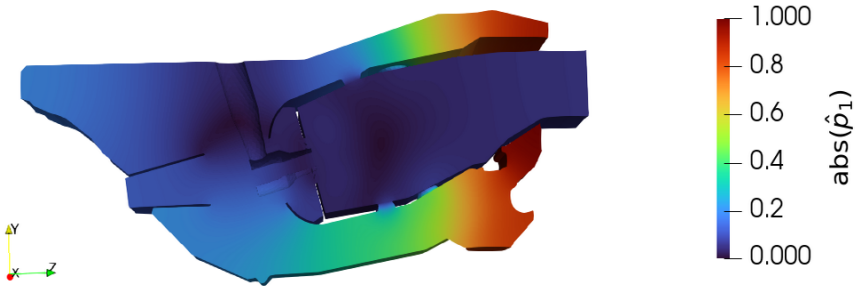


(b) Non-dimensional temperature field.

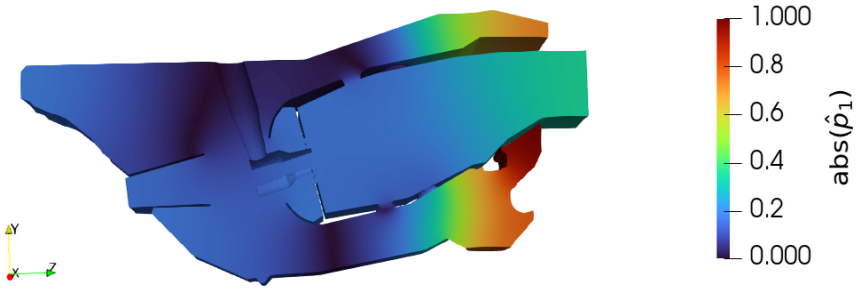
Fig. 4.12 The (a) measurement function field and (b) non-dimensional temperature field of the combustor. The temperature field is obtained from a low order network code.



(a) $f = 1.2056 + 0.0321i$



(b) $f = 1.5189 + 0.0006i$



(c) $f = 1.8014 - 0.0014i$

Fig. 4.13 Various axial eigenmodes of the combustor with the corresponding eigenfrequencies scaled with f_{ref}.

Table 4.3 Parameters of the thermoacoustic problem for the annular combustor.

Parameter	value	unit
r_{gas}	287.0	$\mathrm{Jkg^{-1}K^{-1}}$
$\bar{p}_{gas}$	334.809	kPa
$\bar{q}_{tot}$	-8.19568E6	W
$\bar{u}_b$	90.98	$\mathrm{m\ s^{-1}}$
n	-6.912	-
τ	0.0039	s
α	0.004	-
M_{in}	0.03074	-
M_{out}	0.0702	-

4.5.4 Surface Sensitivities

We start by obtaining the shape sensitivities of the boundaries for the axial mode in Fig. 4.13a. For this, we divide the shape derivative by the surface area of the boundary to obtain the local average as explained in Sec. 3.5.3. We modify Eq.(4.5) to obtain

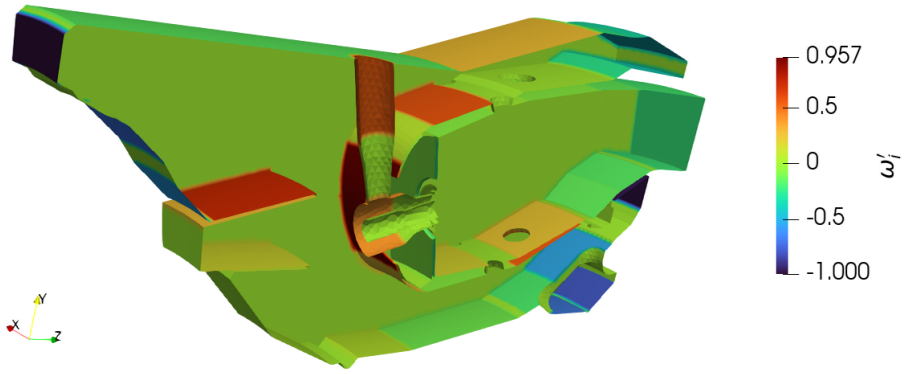
$$\omega_s' = \int_\Gamma \frac{1}{A_s} \Bigg(- \hat{p_1}^{\dagger^*} \left(\kappa c^2 \frac{\partial c}{\partial n} \right) \frac{\partial \hat{p_1}}{\partial n} + \nabla \cdot \left(\hat{p_1}^{\dagger^*} c^2 \nabla \hat{p_1} \right) - 2 \frac{\partial \hat{p_1}^{\dagger^*}}{\partial n} c^2 \frac{\partial \hat{p_1}}{\partial n} \Bigg) dS_s \tag{4.8}$$

where subscript $_s$ denotes the corresponding surface. We use Eq. (4.8) to calculate the average shape derivatives of the combustor surfaces. We then normalize them by the surface with the largest imaginary part. The imaginary parts of the computed surface sensitivities are visualized in Fig. 4.14. We do not show the surface sensitivities for surfaces with areas less than 1.3e-3 m^2 because the sensitivity per unit area is too large to show with this scale.

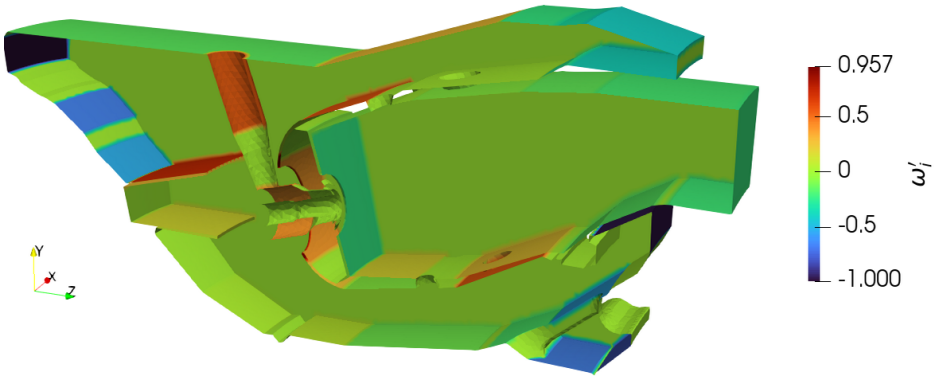
Inspecting Fig. 4.14, the surfaces near the injector are found to be the most influential surfaces for control of the thermoacoustic growth rate. In addition, the inlet and outlet boundaries should be moved in the outward normal direction to achieve a more stable axial eigenmode.

4.5.5 Control Point Sensitivities

As we consider Neumann boundary conditions for FFD control points, Eq. (4.5) simplifies to



(a)



(b)

Fig. 4.14 Surface sensitivities for the unstable eigenmode from (a) front and (b) rear views.

$$\omega'_{ijk} = \int_{\Gamma} (\mathbf{V}_{ijk} \cdot \mathbf{e}_{ijk}) \nabla \cdot \left(\hat{p_1}^{\dagger^*} c^2 \nabla \hat{p_1} \right) dS \tag{4.9}$$

where $\mathbf{e}_{ijk}$ represents the unit vector. We present an example displacement field for the control point $\mathbf{P}_{1,1,0}$ in Fig. 4.15. We use Eq. (4.9) to calculate the FFD control points' shape

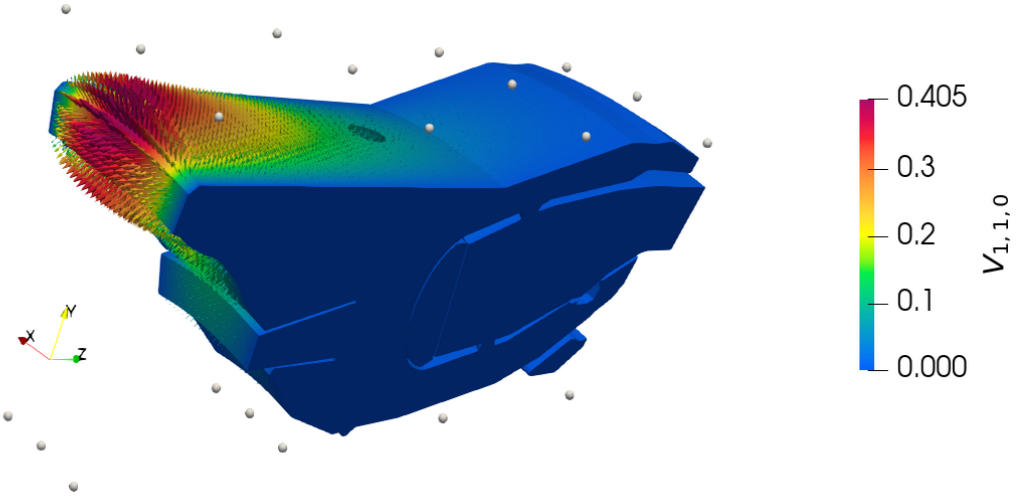


Fig. 4.15 An example displacement field for the FFD control point $\mathbf{P}_{1,1,0}$. We use Eq. (4.4) to compute this.

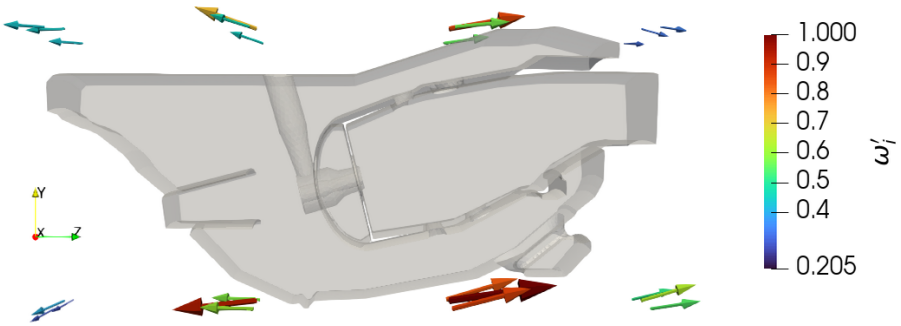
sensitivities. We then normalize them as in section 4.5.4. The imaginary components of the normalized shape derivatives are shown in Fig. 4.16.

We see that the FFD shape derivatives show that the thermoacoustic growth rate can be reduced by enlarging the combustor geometry in the axial direction. This is consistent with the area derivatives presented in Fig. 4.14.

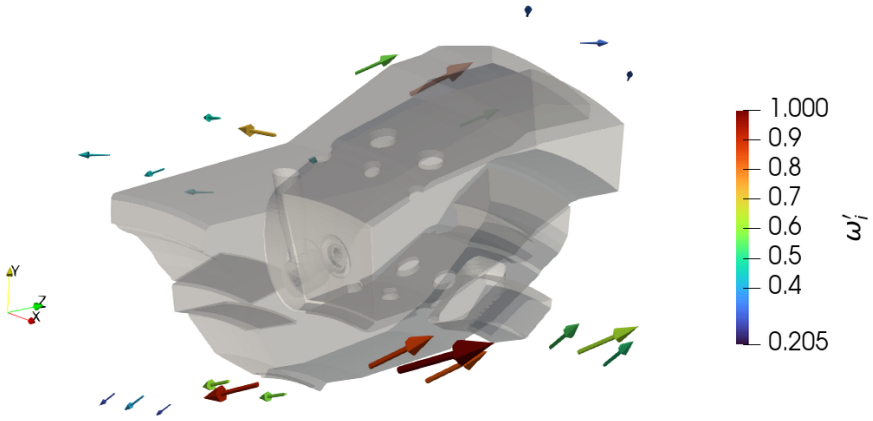
4.5.6 Mesh Deformation

We now apply free form deformation to the entire geometry. We update the positions of the FFD control points in the directions provided by the imaginary part of the shape derivatives. We deform the mesh by a certain step size equal to $1/40^{th}$ of the combustor length. We then update the location of the measurement point x_r to be positioned at a given distance from the injector end. We also adapt the temperature field such that it fits inside the combustion chamber after deformation.

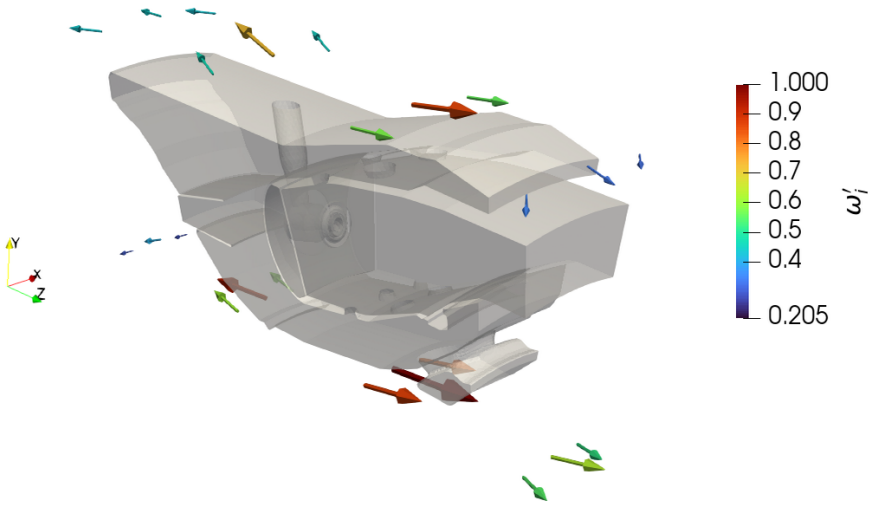
For the deformed geometry, we obtain the eigenfrequency of $f = 1.1728 - 0.0023i$ for the same axial eigenmode. This shows that this free form deformation has reduced the growth



(a)



(b)



(c)

Fig. 4.16 Imaginary parts of the FFD shape derivatives for the eigenfrequency $1.2056 + 0.0321i$. The direction of the derivatives show the direction of changes for stabilizing the system.

rate and thereby stabilized the unstable axial eigenmode. The frequency of the deformed system is reduced because the combustor length is increased in the axial direction. The slice view for the optimized geometry is presented in Fig. 4.17.

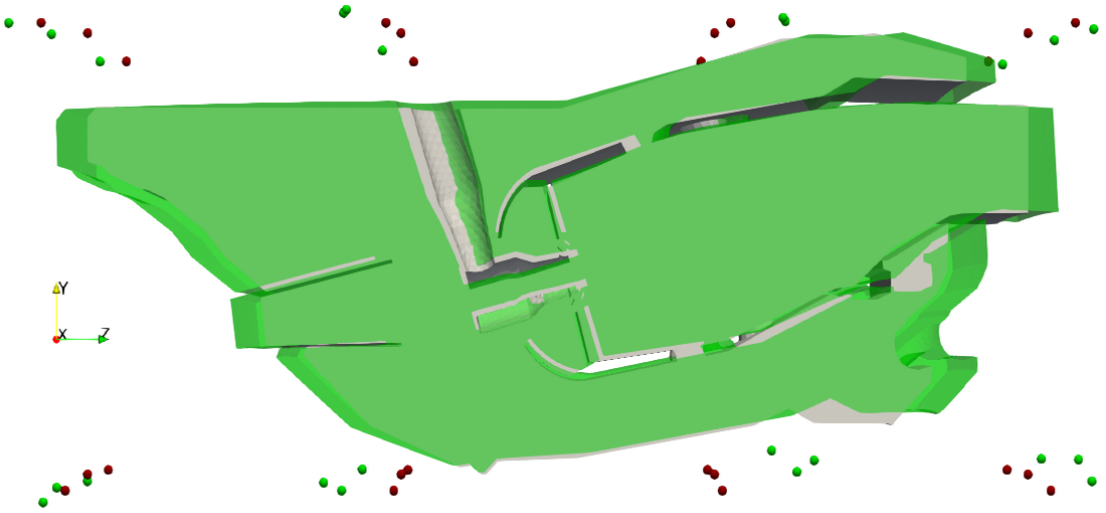


Fig. 4.17 The initial (gray) and deformed (green) geometries with initial (red dots) and final (green dots) positions of the FFD control points. The eigenfrequency of the system changed from $f = 1.2056 + 0.0321i$ (gray) to $f = 1.1728 - 0.0023$ (green).

4.6 Conclusions

In this chapter, we introduce free form deformation as a technique to reduce the instability of thermoacoustic systems. We find the shape derivatives of the eigenvalue with respect to the FFD control points using *helmholtz-x* with only two calculations: the direct and adjoint eigenmodes with degree 2 finite elements. Then we impose mesh morphing with control point displacements by following the directions provided by the shape gradients at these points.

We first show the capability of this approach on the Rijke tube, by stabilizing the first axial eigenmode through FFD. We then test the method to reduce the growth rate of the circumferential thermoacoustic mode of an academic LPP aeroengine combustor. Using separate control lattices for the plenum and combustion chamber of the LPP, we managed to reduce the acoustic oscillations of the degenerate eigenmodes.

Lastly, we have proposed a shape optimization procedure to reduce thermoacoustic instability in an industrial gas turbine combustor geometry. We use the mean flow data from an industrial low order network code. We initially calculate the influence of surface

deformations on the unstable eigenmode and calculate the averaged shape derivatives of the boundaries. These derivatives provide some insight as to the deformation directions that will stabilize the thermoacoustic eigenmodes. We then calculate the derivative of the unstable eigenvalue with respect to the positions of FFD control points. We use the parallelization capability of *helmholtz-x* to calculate the FFD shape derivatives for this industrial geometry, as it requires more finite elements compared to the previous examples. Following these gradients, we deform the shape using the FFD control points to reduce the growth rate of the specific axial eigenmode.

The results in the chapter show that adjoint-based shape sensitivity can be combined with free-form deformation to stabilize thermoacoustic modes in an industrial combustion chamber geometries.

Chapter 5

Conclusion

This thesis proposes a shape optimization procedure for reducing the growth rates of thermoacoustic instabilities in annular combustors with an open-source parallelized adjoint Helmholtz solver. Two different shape parametrization techniques are applied: NURBS and FFD. The thermoacoustic eigenvalue problem is formulated with the inhomogeneous Helmholtz equation. The resulting direct and adjoint equations and boundary conditions are discretized with the finite element method with degree 2 finite elements. The direct and adjoint thermoacoustic eigenmodes are determined with an adjoint Helmholtz solver. The solver is called *helmholtz-x*, which uses MPI for parallelization, DOLFINX and UFL for finite element discretization and PETSC and SLEPC packages for matrix construction and solution of the thermoacoustic Helmholtz equation. With the direct and adjoint eigenfunctions for the unstable eigenmode, *helmholtz-x* computes the shape derivatives of the unstable eigenvalues with respect to the NURBS/FFD control points. The design changes are then applied by moving the control points to stabilize the annular combustors. In order to compute the shape gradients for each control point, the adjoint feature of *helmholtz-x* greatly reduces the number of calculations required, compared with finite differences. This shape optimization procedure is first applied to a laboratory combustor, MICCA with NURBS to reduce the growth rate of the circumferential and mixed eigenmodes. Next, it is demonstrated on an industrial gas turbine combustor with FFD to suppress the thermoacoustic oscillations for the axial eigenmode.

In chapter 2, we present the FEM discretization and implementation details of *helmholtz-x* with increasingly elaborate examples. We first investigate axial eigenmodes in longitudinal combustors. We begin with a relatively simple example, the Rijke tube with Neumann boundary conditions. Then we propose a more detailed longitudinal example with area changes in the axial direction and choked boundary conditions at the inlet and outlet boundaries. We find that eigenmode computations of *helmholtz-x* in different configurations agree well with those

of a network model for passive and active flame cases. We then present a numerical example of a laboratory 3D annular combustor, MICCA. We implement a 3D parabolic temperature field. Then we present different possible eigenmodes of the MICCA combustor such as axial, circumferential and mixed eigenmodes. For the efficient calculation of circumferential modes, we also introduce Bloch boundary conditions to MICCA. When combined with parallel computing, the circumferential eigenmode computations are much quicker with *helmholtz-x* than with existing 3D FEM tools in the literature.

In chapter 3, we show thermoacoustic shape optimization studies with NURBS parametrization. We study a circle, cylinder and eventually MICCA combustor. We focus on the shape optimization of MICCA and parametrize the entire combustor geometry with NURBS control points. The modes are azimuthal and mixed, and therefore two-fold degenerate. We calculate the shape derivative of the eigenvalue. This method is applied to the plenum, the burner, and the combustion chamber. The most influential component on thermoacoustic behaviour is the burner shape. We use an experimentally-derived flame transfer function. Hence we fixed the shape of the burner because we do not know how it would change the flame transfer function. We apply both symmetry-preserving and symmetry-breaking changes to the MICCA geometry with NURBS and show that these can be used to reduce the thermoacoustic growth rate efficiently. After modifying the shape, we use the analysis to reveal the physical mechanism that causes the growth rate's reduction. The shape changes increase the phase difference between the pressure and the heat release rate fluctuations. This process could be continued to zero growth rate, but large geometry changes would be required. We also do not include acoustic dissipation from the lateral boundaries, which would also reduce the growth rate further.

In chapter 4, we present a free form deformation technique to deal with more complicated geometries through parametrization. We specify FFD control points around the geometry and calculate their shape derivatives using direct and adjoint eigenfunctions. Then we impose mesh morphing with control point displacements by following the directions provided by the shape gradients at FFD points. We apply the FFD procedure to three different thermoacoustic systems with an $n-\tau$ formulation for modelling the flame response. We first parametrize the 3D Rijke tube geometry. We specify control points around the tube and calculate their shape derivatives. Following the shape gradients, we stabilize the Rijke tube after FFD. We then apply this method to a lean premixed prevaporized (LPP) aeroengine combustor. We parametrize the plenum and combustion chamber of the LPP using local control lattices. After calculating the shape derivatives of the FFD control points, we modify the local control lattices accordingly. The deformed LPP geometry has a lower growth rate for the circumferential mode, as expected. Lastly, we have proposed a shape optimization procedure

for an industrial combustor geometry to reduce thermoacoustic instability. We input the mean flow and FTF data from an industrial low order network code. We initially calculate the influence of surface deformations on the unstable eigenmode and calculate the averaged shape derivatives of the boundaries. These derivatives provide some insight as to the deformation directions that will stabilize the thermoacoustic eigenmodes. We then calculate the derivative of the unstable eigenvalue with respect to the positions of FFD control points. Following these gradients, we deform the shape using the FFD control points to reduce the growth rate of the unstable eigenmode. With these deformations, we suppress the thermoacoustic instability of this axial mode. The proposed shape changes through the FFD control points agree with the averaged shape derivatives.

Given its applicability to the examples shown in this thesis, *helmholtz-x* could be a useful numerical tool to study and passively control thermoacoustic instabilities of complex shaped real-world combustors.

The next steps are to improve the accuracy of the components of the Helmholtz solver. Further acoustic or thermoacoustic test cases could be implemented, along with experimental or analytical analysis. More realistic geometries could be studied with experimentally-derived flame-transfer function data. More robust interpolation schemes for 3D temperature fields could be implemented to incorporate the temperature distributions from experiments or LES. In this thesis, we neglect the acoustic liners, which could be accounted for, e.g. via Rayleigh conductivity. We also removed the swirler geometry within the injector in the industrial annular combustor. These influence the thermoacoustic behaviour and have been modelled by another Helmholtz solver [24, 25]. We will implement these within our finite element framework. Regarding shape optimization, the FFD procedure could be advanced by including more complicated geometries with other engineering constraints. In practice, combustion chambers and plenums have other geometrical constraints that are not considered in this thesis. If adjoint methods are to be used for combustor design, then these methods must include the influence of the burner shape on the flow behaviour, as in [84], hence on the flame. Using the linearized Navier-Stokes equations as in [85] with resolvent analysis would provide a better approximation to the response of the flame and enable a deeper exploration of the influence of burner geometry on thermoacoustic oscillations.

References

[1] Luigi Crocco. Aspects of combustion stability in liquid propellant rocket motors part i: fundamentals. low frequency instability with monopropellants. *Journal of the American Rocket Society*, 21(6):163–178, 1951. (cited on page 1).

[2] Luigi Crocco and Sin-I Cheng. *Theory of combustion instability in liquid propellant rocket motors*, volume 8. on behalf of the Advisory Group for Aeronautical Research and Development . . . , 1956. (cited on page 1).

[3] Hukam C Mongia, TJ Held, GC Hsiao, and RP Pandalai. Challenges and progress in controlling dynamics in gas turbine combustors. *Journal of Propulsion and Power*, 19(5):822–829, 2003. (cited on page 1).

[4] Tim C Lieuwen. *Unsteady combustor physics*. Cambridge University Press, 2021. (cited on page 1).

[5] Matthew P Juniper and Raman I Sujith. Sensitivity and nonlinearity of thermoacoustic oscillations. *Annual Review of Fluid Mechanics*, 50:661–689, 2018. (cited on pages 1, 2, and 4).

[6] Luca Magri and Matthew P Juniper. Sensitivity analysis of a time-delayed thermo-acoustic system via an adjoint-based approach. *Journal of Fluid Mechanics*, 719:183–202, 2013. (cited on pages 1 and 5).

[7] Stefano Falco and Matthew P Juniper. Shape optimization of thermoacoustic systems using a two-dimensional adjoint helmholtz solver. *Journal of Engineering for Gas Turbines and Power*, 143(7):071025, 2021. (cited on pages 2, 5, and 77).

[8] Boa-Teh Chu. *Stability of systems containing a heat source-the Rayleigh criterion*. National Advisory Committee for Aeronautics, 1956. (cited on page 2).

[9] Owen S Graham and Ann P Dowling. A low-order modelling of ducted flames with temporally varying equivalence ratio in realistic geometries. In *Turbo Expo: Power for Land, Sea, and Air*, volume 54624, pages 277–288, 2011. (cited on page 3).

[10] Matthew Juniper, Jingquan Zheng, Matthew Yoko, Fischer Andre, and Lahiri Claus. Bayesian data assimilation in cold flow experiments on an industrial thermoacoustic rig. In *Turbo Expo*, page . Apollo - University of Cambridge Repository, 2024. (cited on page 3).

[11] Ann P Dowling and Simon R Stow. Acoustic analysis of gas turbine combustors. *Journal of propulsion and power*, 19(5):751–764, 2003. (cited on page 3).

[12] Jingxuan Li, Dong Yang, Charles Luzzato, and Aimee S Morgans. Open source combustion instability low order simulator (oscilos-long) technical report. *Imperial College London, London*, pages 1–48, 2015. (cited on pages 3 and 15).

[13] Robert Leandro, Andreas Huber, and Wolfgang Polifke. tax-a low-order modeling tool for thermo-and aero-acoustic instabilities. Technical report, Professur für Thermofluid-dynamik, 2010. (cited on page 3).

[14] Yasser Mahmoudi, Andrea Giusti, Epaminondas Mastorakos, and Ann P Dowling. Low-order modeling of combustion noise in an aero-engine: the effect of entropy dispersion. *Journal of Engineering for Gas Turbines and Power*, 140(1):011502, 2018. (cited on page 3).

[15] Dong Yang and Aimee S Morgans. Low-order network modeling for annular combustors exhibiting longitudinal and circumferential modes. In *Turbo Expo: Power for Land, Sea, and Air*, volume 51067, page V04BT04A026. American Society of Mechanical Engineers, 2018. (cited on page 3).

[16] A Fischer, C Hirsch, and T Sattelmayer. Comparison of multi-microphone transfer matrix measurements with acoustic network models of swirl burners. *Journal of Sound and Vibration*, 298(1-2):73–83, 2006. (cited on page 3).

[17] Charlelie Laurent, A Badhe, and Franck Nicoud. Representing the geometrical complexity of liners and boundaries in low-order modeling for thermoacoustic instabilities. *Journal of Computational Physics*, 428:110077, 2021. (cited on page 3).

[18] Georg A Mensah, Giovanni Campa, and Jonas P Moeck. Efficient computation of thermoacoustic modes in industrial annular combustion chambers based on bloch-wave

theory. *Journal of Engineering for Gas Turbines and Power*, 138(8):081502, 2016. (cited on pages 4, 20, and 41).

[19] Georg Atta Mensah. *Efficient computation of thermoacoustic modes*. PhD thesis, Technische Universität Berlin, 2019. (cited on pages 4, 20, 22, and 43).

[20] Georg A Mensah, Luca Magri, and Jonas P Moeck. Methods for the calculation of thermoacoustic stability boundaries and monte carlo-free uncertainty quantification. *Journal of Engineering for Gas Turbines and Power*, 140(6):061501, 2018. (cited on page 4).

[21] Philip E Buschmann, Georg A Mensah, Franck Nicoud, and Jonas P Moeck. Solution of thermoacoustic eigenvalue problems with a non-iterative method. In *Turbo Expo: Power for Land, Sea, and Air*, volume 58615, page V04AT04A051. American Society of Mechanical Engineers, 2019. (cited on page 4).

[22] Georg A Mensah, Alessandro Orchini, Philip E Buschmann, and Luka Grubišić. A subspace-accelerated method for solving nonlinear thermoacoustic eigenvalue problems. *Journal of sound and vibration*, 520:116553, 2022. (cited on page 4).

[23] Georg A Mensah, Philip E Buschmann, and Alessandro Orchini. Iterative solvers for the thermoacoustic nonlinear eigenvalue problem and their convergence properties. *International Journal of Spray and Combustion Dynamics*, 14(1-2):30–41, 2022. (cited on page 4).

[24] Franchine Ni, Maxence Miguel-Brebion, Franck Nicoud, and Thierry Poinsot. Accounting for acoustic damping in a helmholtz solver. *AIAA Journal*, 55(4):1205–1220, 2017. (cited on pages 4 and 97).

[25] F Ni, Franck Nicoud, Y Méry, and G Staffelbach. Including flow–acoustic interactions in the helmholtz computations of industrial combustors. *AIAA Journal*, 56(12):4815–4829, 2018. (cited on pages 4 and 97).

[26] Emmanuel Motheau, Franck Nicoud, and Thierry Poinsot. Mixed acoustic–entropy combustion instabilities in gas turbines. *Journal of Fluid Mechanics*, 749:542–576, 2014. (cited on page 4).

[27] Charles E Martin, Laurent Benoit, Yannick Sommerer, Franck Nicoud, and Thierry Poinsot. Large-eddy simulation and acoustic analysis of a swirled staged turbulent combustor. *AIAA journal*, 44(4):741–750, 2006. (cited on pages 4 and 11).

[28] G Boudier, Nicolas Lamarque, Gabriel Staffelbach, LYM Gicquel, and Thierry Poinsot. Thermo-acoustic stability of a helicopter gas turbine combustor using large eddy simulation. *International journal of Aeroacoustics*, 8(1):69–93, 2009. (cited on page 4).

[29] Thierry Poinsot and Denis Veynante. *Theoretical and numerical combustion*. RT Edwards, Inc., 2005. (cited on page 4).

[30] Thierry Poinsot. Prediction and control of combustion instabilities in real engines. *Proceedings of the Combustion Institute*, 36(1):1–28, 2017. (cited on page 4).

[31] Xinyu Zhao, Dan Zhao, Li Cheng, Cody M Shelton, and Joseph Majdalani. Predicting thermoacoustic stability characteristics of longitudinal combustors using different endpoint conditions with a low mach number flow. *Physics of Fluids*, 35(9):–, 2023. (cited on page 4).

[32] Nejat Olgac, Rudy Cepeda-Gomez, Umut Zalluhoglu, and Ayhan S Kammer. Parametric investigation of thermoacoustic instability (tai) in a rijke tube: a time-delay perspective. *International Journal of Spray and Combustion Dynamics*, 7(1):39–68, 2015. No citations.

[33] Seonyeong Kim and Daesik Kim. Analytical modeling of thermoacoustic instability influences in gas turbine combustors: A detailed parameter sensitivity analysis. *Case Studies in Thermal Engineering*, page 104595, 2024. (cited on page 4).

[34] Stefanie Bade, Michael Wagner, Christoph Hirsch, Thomas Sattelmayer, and Bruno Schuermans. Design for thermo-acoustic stability: modeling of burner and flame dynamics. *Journal of engineering for gas turbines and power*, 135(11):111502, 2013. (cited on page 5).

[35] Stefanie Bade, Michael Wagner, Christoph Hirsch, Thomas Sattelmayer, and Bruno Schuermans. Design for thermo-acoustic stability: Procedure and database. *Journal of engineering for gas turbines and power*, 135(12):121507, 2013. (cited on page 5).

[36] José G Aguilar, Luca Magri, and Matthew P Juniper. Adjoint-based sensitivity analysis of low-order thermoacoustic networks using a wave-based approach. *Journal of Computational Physics*, 341:163–181, 2017. (cited on page 5).

[37] José G Aguilar and Matthew P Juniper. Adjoint methods for elimination of thermoacoustic oscillations in a model annular combustor via small geometry modifications.

In *Turbo Expo: Power for Land, Sea, and Air*, volume 51050, page V04AT04A054. American Society of Mechanical Engineers, 2018. (cited on pages 5, 78, 80, 81, and 83).

[38] José G Aguilar and Matthew P Juniper. Thermoacoustic stabilization of a longitudinal combustor using adjoint methods. *Physical Review Fluids*, 5(8):083902, 2020. (cited on page 5).

[39] Camilo F Silva, Luca Magri, Thomas Runte, and Wolfgang Polifke. Uncertainty quantification of growth rates of thermoacoustic instability by an adjoint helmholtz solver. *Journal of Engineering for Gas Turbines and Power*, 139(1):011901, 2017. (cited on page 5).

[40] Georg A Mensah, Luca Magri, Alessandro Orchini, and Jonas P Moeck. Effects of asymmetry on thermoacoustic modes in annular combustors: a higher-order perturbation study. *Journal of Engineering for Gas Turbines and Power*, 141(4):041030, 2019. (cited on pages 5 and 40).

[41] Camilo F Silva, Laura Prieto, Maximiliano Ancharek, Pablo Marigliani, and Georg A Mensah. Adjoint-based calculation of parametric thermoacoustic maps of an industrial combustion chamber. *Journal of Engineering for Gas Turbines and Power*, 143(1):011003, 2021. (cited on page 5).

[42] Georg A Mensah, Alessandro Orchini, and Jonas P Moeck. *Adjoint-based computation of shape sensitivity in a Rijke-tube*. Universitätsbibliothek der RWTH Aachen, 2019. (cited on page 5).

[43] Matthew P Juniper. Sensitivity analysis of thermoacoustic instability with adjoint helmholtz solvers. *Physical Review Fluids*, 3(11):110509, 2018. (cited on pages xv, xxi, 5, 11, 31, 33, 34, and 76).

[44] Luca Magri. Adjoint methods as design tools in thermoacoustics. *Applied Mechanics Reviews*, 71(2):020801, 2019. (cited on pages 5 and 10).

[45] Ekrem Ekici, Stefano Falco, and Matthew P Juniper. Shape sensitivity of thermoacoustic oscillations in an annular combustor with a 3d adjoint helmholtz solver. *Computer Methods in Applied Mechanics and Engineering*, 418:116572, 2024. (cited on pages 5, 45, and 78).

[46] Shenren Xu, Wolfram Jahn, and Jens-Dominik Müller. Cad-based shape optimisation with cfd using a discrete adjoint. *International Journal for Numerical Methods in Fluids*, 74(3):153–168, 2014. (cited on page 5).

[47] DN Srinath and Sanjay Mittal. An adjoint method for shape optimization in unsteady viscous flows. *Journal of Computational Physics*, 229(6):1994–2008, 2010. (cited on page 5).

[48] Matteo Pini, Giacomo Persico, David Pasquale, and Stefano Rebay. Adjoint method for shape optimization in real-gas flow applications. *Journal of Engineering for Gas Turbines and Power*, 137(3):032604, 2015. (cited on page 5).

[49] Thomas W Sederberg and Scott R Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986. (cited on pages 5 and 71).

[50] Jamshid Samareh. Aerodynamic shape optimization based on free-form deformation. In *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 4630, 2004. (cited on page 6).

[51] Xiaolong He, Jichao Li, Charles A Mader, Anil Yildirim, and Joaquim RRA Martins. Robust aerodynamic shape optimization—from a circle to an airfoil. *Aerospace Science and Technology*, 87:48–61, 2019. (cited on page 6).

[52] Andrea Giugno, Shahrokh Shahpar, and Alberto Traverso. Adjoint-based optimization of a modern jet-engine fan blade. In *Turbo Expo: Power for Land, Sea, and Air*, volume 84096, page V02DT38A026. American Society of Mechanical Engineers, 2020. (cited on page 6).

[53] Alistair John, Shahrokh Shahpar, and Ning Qin. Novel compressor blade shaping through a free-form method. *Journal of Turbomachinery*, 139(8):081002, 2017. (cited on page 6).

[54] Lei Li, Tianyu Yuan, Yuan Li, Weizhu Yang, and Jialei Kang. Multidisciplinary design optimization based on parameterized free-form deformation for single turbine. *AIAA Journal*, 57(5):2075–2087, 2019. (cited on page 6).

[55] Ekrem Ekici and Matthew Juniper. Adjoint based shape optimization for thermoacoustic stability of combustors using free form deformation. In *Turbo Expo*, page . Apollo - University of Cambridge Repository, 2024. (cited on pages 6 and 71).

[56] Ekrem Ekici and Matthew P. Juniper. Shape optimization of an industrial aeroengine combustor to reduce thermoacoustic instability (to be submitted). *Journal of Engineering for Gas Turbines and Power*, x(x):x, 2024. (cited on pages 6 and 71).

[57] Stefano Falco. *Shape optimization for thermoacoustic instability with an adjoint Helmholtz solver*. PhD thesis, University of Cambridge, 2022. (cited on pages xvii, 6, 10, 45, 46, 60, and 62).

[58] Ekrem Ekici, Stefano Falco, and Matthew P. Juniper. helmholtz-x : Parallelized adjoint open source solver for the thermoacoustic helmholtz equation (submitted). *Engineering with Computers*, page x, 2024. (cited on page 9).

[59] Igor A Barrata, Joseph P Dean, Jørgen S Dokken, Michal HABERA, Jack HALE, Chris Richardson, Marie E Rognes, Matthew W Scroggs, Nathan Sime, and Garth N Wells. Dolfinx: The next generation fenics problem solving environment. *https://docs.fenicsproject.org/dolfinx/main/python/*, 2023. (cited on page 9).

[60] David W Walker and Jack J Dongarra. Mpi: a standard message passing interface. *Supercomputer*, 12:56–68, 1996. (cited on pages 9 and 115).

[61] Lisandro Dalcín, Rodrigo Paz, and Mario Storti. Mpi for python. *Journal of Parallel and Distributed Computing*, 65(9):1108–1115, 2005. (cited on page 9).

[62] Martin Sandve Alnæs. Ufl: a finite element form language. In *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, pages 303–338. Springer, 2012. (cited on page 9).

[63] Anders Logg, Kristian B Ølgaard, Marie E Rognes, and Garth N Wells. Ffc: the fenics form compiler. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, pages 227–238, 2012. (cited on page 9).

[64] Satish Balay, Shrirang Abhyankar, Mark Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Alp Dener, Victor Eijkhout, W Gropp, et al. Petsc users manual. *Astrophysics Source Code Library*, 2019. (cited on page 10).

[65] Lisandro Dalcin. Petsc for python, 2010. (cited on page 10).

[66] Vicente Hernandez, Jose E Roman, and Vicente Vidal. Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):351–362, 2005. (cited on page 10).

[67] Lisandro Dalcin. Slepc for python. *Trans. Math. Softw*, 31(3):351–362, 2005. (cited on page 10).

[68] Luigi Crocco, Jerry Grey, and David T Harrje. Theory of liquid propeliant rocket combustion instability and its experimental verification. *ARS Journal*, 30(2):159–168, 1960. (cited on page 11).

[69] S.W. Rienstra and A. Hirschberg. *An Introduction to Acoustics*. Eindhoven University of Technology, 2004. (cited on page 12).

[70] Felix Bloch. Über die quantenmechanik der elektronen in kristallgittern. *Zeitschrift für physik*, 52(7-8):555–600, 1929. (cited on page 20).

[71] F. Nicoud, L. Benoit, C. Sensiau, and T. Poinsot. Acoustic Modes in Combustors with Complex Impedances and Multidimensional Active Flames. *AIAA Journal*, 45(2):426–441, 2007. (cited on pages 23, 80, 109, and 110).

[72] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009. (cited on pages 24, 52, 76, and 78).

[73] Utkarsh Ayachit. *The paraview guide: a parallel visualization application*. Kitware, Inc., 2015. (cited on page 28).

[74] Jean-François Bourgouin, Daniel Durox, Jonas P Moeck, Thierry Schuller, and Sébastien Candel. Characterization and modeling of a spinning thermoacoustic instability in an annular combustor equipped with multiple matrix injectors. *Journal of Engineering for Gas Turbines and Power*, 137(2):021503, 2015. (cited on page 37).

[75] Davide Laera, Thierry Schuller, Kevin Prieur, Daniel Durox, Sergio M Camporeale, and Sébastien Candel. Flame describing function analysis of spinning and standing modes in an annular combustor and comparison with experiments. *Combustion and Flame*, 184:136–152, 2017. (cited on pages xvi, 37, 40, and 43).

[76] Daniel Durox, Jean-François Bourgouin, Jonas P Moeck, M Philip, Thierry Schuller, and Sébastien Candel. Nonlinear interactions in combustion instabilities coupled by azimuthal acoustic modes. In *n3l-Int'l Summer School and Workshop on Non-Normal and Nonlinear Effects In Aero-and Thermoacoustics*, page 14, 2013. (cited on page 37).

[77] Jean-François Bourgouin, Daniel Durox, JP Moeck, Thierry Schuller, and Sébastien Candel. A new pattern of instability observed in an annular combustor: The slanted mode. *Proceedings of the Combustion Institute*, 35(3):3237–3244, 2015. (cited on page 37).

[78] Bjorn Gustavsen and Adam Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Transactions on power delivery*, 14(3):1052–1061, 1999. (cited on page 40).

[79] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 1996. (cited on pages 45 and 52).

[80] Alessandro Orchini, Camilo F Silva, Georg A Mensah, and Jonas P Moeck. Thermoacoustic modes of intrinsic and acoustic origin and their interplay with exceptional points. *Combustion and Flame*, 211:83–95, 2020. (cited on page 52).

[81] Georg A. Mensah, Luca Magri, Alessandro Orchini, and Jonas P. Moeck. Effects of Asymmetry on Thermoacoustic Modes in Annular Combustors: A Higher-Order Perturbation Study. *J. Eng. Gas Turbines Power*, 141(4):041030, apr 2019. (cited on page 55).

[82] José G. Aguilar and Matthew P. Juniper. Thermoacoustic stabilization of a longitudinal combustor using adjoint methods. *Phys. Rev. Fluids*, 5(8):083902, 2020. (cited on pages 56 and 66).

[83] Aimee S Morgans and Simon R Stow. Model-based control of combustion instabilities in annular combustors. *Combustion and flame*, 150(4):380–399, 2007. (cited on pages 78, 80, and 81).

[84] O. Tammisola and M. P. Juniper. Coherent structures in a swirl injector at Re = 4800 by nonlinear simulations and linear global modes. *J. Fluid Mech.*, 792:620–657, apr 2016. (cited on page 97).

[85] Thomas L Kaiser, Gregoire Varillon, Wolfgang Polifke, Feichi Zhang, Thorsten Zirwes, Henning Bockhorn, and Kilian Oberleithner. Modelling the response of a turbulent jet flame to acoustic forcing in a linearized framework using an active flame approach. *Combustion and Flame*, 253:112778, 2023. (cited on page 97).

Appendix A

Thermoacoustic Helmholtz Equation

The thermoacoustic governing equations are based on the conservation laws of mass, momentum and energy for an ideal gas. We follow the methodology of [71]. Balance equations for mass, momentum and entropy can be written in inviscid and conservative form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{A.1a}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla p = 0 \tag{A.1b}$$

$$\frac{\partial \zeta}{\partial t} + \mathbf{u} \cdot \nabla \zeta = \frac{\dot{q}}{\rho T} \tag{A.1c}$$

where t represents time, $\mathbf{u}$ is the velocity vector, p is the pressure, ζ is entropy, T is temperature and $\dot{q}$ is the heat release rate per unit volume. These transport equations represent the time dependent/independent variations of flow variables within a physical domain.

The equations in (A.1) can be linearized through decomposition of flow variables. Each variable can be decomposed into two distinct parts: a mean part(order $\mathscr{O}(1)$, subscripted by $_0$) and a fluctuating part (order $\mathscr{O}(\varepsilon)$, subscripted by $_1$):

$$p(\mathbf{x},t) = p_0(\mathbf{x}) + p_1(\mathbf{x},t) \tag{A.2a}$$
$$\rho(\mathbf{x},t) = \rho_0(\mathbf{x}) + \rho_1(\mathbf{x},t) \tag{A.2b}$$
$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}_0(\mathbf{x}) + \mathbf{u}_1(\mathbf{x},t) \tag{A.2c}$$
$$T(\mathbf{x},t) = T_0(\mathbf{x}) + T_1(\mathbf{x},t) \tag{A.2d}$$
$$\dot{q}(\mathbf{x},t) = q_0(\mathbf{x}) + q_1(\mathbf{x},t) \tag{A.2e}$$
$$\zeta(\mathbf{x},t) = \zeta_0(\mathbf{x}) + \zeta_1(\mathbf{x},t) \tag{A.2f}$$

These perturbations are applied to equation (A.1) by making the following assumptions:

- The gas is assumed to be a perfect gas ($p_0 = \rho_0 r_{gas} T_0$, with c_p, c_v, γ constant).
- Viscous terms and volume forces are neglected.
- The zero Mach number mean flow assumption ($\mathbf{u}_0(\mathbf{x}) \approx 0$) is imposed for further simplification of Eq. (A.1)(for details, see Appendix A of [71]).
- Only first order perturbations terms are retained. Second and higher order terms are neglected.

If $\mathbf{u}_0(\mathbf{x}) \approx 0$, then Eq. (A.1) requires that the gradient of the mean pressure (∇p_0) is zero. The linearized governing equations for perturbed quantities then become:

$$\frac{\partial \rho_1}{\partial t} + \nabla \cdot (\rho_0 \mathbf{u_1}) = m_1 \tag{A.3a}$$
$$\rho_0 \frac{\partial \mathbf{u_1}}{\partial t} + \nabla p_1 = f_1 \tag{A.3b}$$
$$\frac{\partial \zeta_1}{\partial t} + \mathbf{u_1} \cdot \nabla \zeta_0 = \frac{q_1}{\rho_0 T_0} \tag{A.3c}$$

where m_1 is the fluctuating mass injection per unit volume per unit time and f_1 is the fluctuating body force per unit volume. These three balance equations in (A.3) can be combined. The ideal gas law $p_0 = \rho_0 r_{gas} T_0$ is written in total differential form:

$$dp = d\rho r_{gas} T + \rho r_{gas} dT \implies \frac{dp}{T} = \rho r_{gas} \frac{dT}{T} + r_{gas} d\rho \implies \frac{dp}{P} = \frac{dT}{T} + \frac{d\rho}{\rho} \tag{A.4}$$

The perturbations in (A.2) are applied to (A.4). By keeping all the first order terms(ε), this leads to

$$\frac{p_1}{p_0} = \frac{T_1}{T_0} + \frac{\rho_1}{\rho_0} \tag{A.5}$$

Combining $Tds = dh - vdp$ with the ideal gas law $p = \rho r_{gas} T$ gives[1]:

$$d\zeta = c_v \frac{dp}{p} - c_p \frac{d\rho}{\rho} \tag{A.6}$$

For a perfect gas changing from state 0 to a general state, the change in entropy is given by:

$$\zeta - \zeta_0 = c_v \ln\left(\frac{p}{p_0}\right) - c_p \ln\left(\frac{\rho}{\rho_0}\right) \implies \frac{p}{p_0} = \exp\left(\frac{\zeta - \zeta_0}{c_v}\right)\left(\frac{\rho}{\rho_0}\right)^{\gamma} \tag{A.7}$$

Applying Taylor series expansion while keeping only first order terms results in:

$$p(\rho_0 + \rho_1, \zeta_0 + \zeta_1) = p(\rho_0, \zeta_0) + \frac{\partial p}{\partial \rho}\Big|_{\rho_0,\zeta_0}(\rho - \rho_0) + \frac{\partial p}{\partial \zeta}\Big|_{\rho_0,\zeta_0}(\zeta - \zeta_0) \tag{A.8}$$

where $\rho = \rho_0 + \rho_1$. The partial derivatives of p with respect to density and entropy are:

$$\frac{\partial p}{\partial \rho}\Big|_{\rho_0,\zeta_0} = \gamma \frac{p_0}{\rho_0} \quad , \quad \frac{\partial p}{\partial \zeta}\Big|_{\rho_0,\zeta_0} = \frac{p_0}{c_v} \tag{A.9}$$

Substitution of terms in equation (A.9) into equation (A.8) yields:

$$p_1 + p_0 = p_0 + \gamma \frac{p_0}{\rho_0}\Big((\rho_0 + \rho_1) - \rho_0\Big) + \frac{p_0}{c_v}\Big((\zeta_0 + \zeta_1) - \zeta_0\Big) \tag{A.10}$$

Dividing both side by p_0 gives the relation for ζ_1:

$$\frac{p_1}{p_0} = \gamma \frac{\rho_1}{\rho_0} + \frac{\zeta_1}{c_v} \tag{A.11}$$

If there is no mean flow then $\nabla p_0 = 0$. Considering spatial variations around the unperturbed state, the entropy gradient is simply a function of the mean density gradient:

[1]

$Td\zeta = dh - vdp$

$d\zeta = \frac{c_p}{T}dT - \frac{1}{\rho T}dp = \frac{dT}{T}c_p - r_{gas}\frac{dp}{p} = c_p(\frac{dp}{p} - \frac{d\rho}{\rho} - r_{gas}\frac{dp}{p}) = (c_p - r_{gas})\frac{dp}{p} - c_p\frac{d\rho}{\rho} = c_v\frac{dp}{p} - c_p\frac{d\rho}{\rho}$

$$\nabla\zeta_0 = \frac{c_v}{p_0}\nabla p_0 - \frac{c_p}{\rho}\nabla\rho_0 \approx -\frac{c_p}{\rho}\nabla\rho_0 \tag{A.12}$$

Substituting equation (A.12) into equation (A.3c) gives:

$$\frac{\partial\zeta_1}{\partial t} - \mathbf{u}_1\frac{c_p}{\rho_0}\nabla\rho_0 = \frac{r_{gas}q_1}{p_0} \implies \frac{1}{c_p}\frac{\partial\zeta_1}{\partial t} - \frac{\mathbf{u}_1}{\rho_0}\nabla\rho_0 = \frac{r_{gas}q_1}{c_p p_0} \tag{A.13}$$

From relation (A.11), ρ_1 can be obtained and the partial derivative with respect to time is:

$$\rho_1 = \frac{\rho_0}{\gamma}\left(\frac{p_1}{p_0} - \frac{\zeta_1}{c_v}\right) \implies \frac{\partial\rho_1}{\partial t} = \frac{\rho_0}{\gamma p_0}\frac{\partial p_1}{\partial t} - \frac{\rho_0}{c_p}\frac{\partial\zeta_1}{\partial t} \tag{A.14}$$

Inserting equation (A.14) into (A.3a) and using (A.13) yields:

$$\frac{1}{\gamma p_0}\frac{\partial p_1}{\partial t} + \nabla\mathbf{u}_1 = \frac{r_{gas}q_1}{c_p p_0} + \frac{m_1}{\rho_0} \tag{A.15}$$

where $r_{gas} = c_p - c_v$ and c_p/c_v gives the heat capacity ratio γ. Taking the time derivative of (A.15) and dividing equation (A.3b) by ρ_0 leads to:

$$\frac{1}{\gamma p_0}\frac{\partial p_1}{\partial t^2} - \nabla\cdot\left(\frac{1}{\rho_0}\nabla p_1 - \frac{f_1}{\rho_0}\right) = \frac{\gamma-1}{\gamma p_0}\frac{\partial q_1}{\partial t} + \frac{1}{\rho_0}\frac{\partial m_1}{\partial t} \tag{A.16}$$

Substituting $c = \sqrt{\gamma p_0/\rho_0}$, into Eq. (A.16) gives:

$$\frac{\partial p_1}{\partial t^2} - \nabla\cdot\left(c^2\nabla p_1\right) = (\gamma-1)\frac{\partial q_1}{\partial t} - c^2\nabla\cdot f_1 + c^2\frac{\partial m_1}{\partial t} \tag{A.17}$$

Eq. (A.17) is easier to solve by transforming the variables from the time domain to the frequency domain:

$$p_1(\mathbf{x},t) = Re\left\{\hat{p_1}(\mathbf{x})e^{-i\omega t}\right\} \tag{A.18a}$$
$$\mathbf{u}_1(\mathbf{x},t) = Re\left\{\hat{\mathbf{u}}(\mathbf{x})e^{-i\omega t}\right\} \tag{A.18b}$$
$$q_1(\mathbf{x},t) = Re\left\{\hat{q_1}(\mathbf{x})e^{-i\omega t}\right\} \tag{A.18c}$$
$$m_1(\mathbf{x},t) = Re\left\{\hat{m_1}(\mathbf{x})e^{-i\omega t}\right\} \tag{A.18d}$$
$$f_1(\mathbf{x},t) = Re\left\{\hat{f_1}(\mathbf{x})e^{-i\omega t}\right\} \tag{A.18e}$$

where $\omega = \omega_r + \omega_i i$ denotes the complex angular frequency of the acoustic wave, its real part ω_r is the angular frequency and its imaginary part ω_i is the growth rate of the acoustic wave.

This gives the inhomogeneous acoustic wave equation in the frequency domain:

$$\nabla \cdot \left(c^2 \nabla \hat{p_1}\right) + \omega^2 \hat{p_1} = i\omega(\gamma - 1)\hat{q_1} + c^2 \nabla \cdot \hat{f_1} + c^2 i\omega \hat{m_1} \tag{A.19}$$

Eq. (A.19) is called the inhomogeneous Helmholtz equation. If its eigenvalue ω has positive growth rate $\omega_i > 0$, then the oscillating term ($e^{-i\omega t}$) in (A.18) grows exponentially.

Appendix B

MPI utility functions for handling parallel data

We present the MPI utility functions of *helmholtz-x* for constructing the sparse matrix $\mathbf{D}$ and its adjoint, $\mathbf{D}^H$. FEniCSx uses MPI for handling the parallelization [60]. When we parallelize the calculation using n_{proc} processes, FEniCSx partitions the mesh into n_{proc} pieces. Each piece has different nonzero entries for the left and right vectors that we use to construct $\mathbf{D}$ (Sec. 2.1.6). We need the positions and entries of those nonzero contributions for both vectors. We obtain these as (indices, nonzero values) pairs. For $\mathbf{D}$, we obtain row indices from the left vector and column indices from the right vector. We calculate the nonzero entries by multiplying the nonzero values of the left and right vectors within the same process. The nonzero data for the right vector may, however, be 'None', while the contribution for the left vector may be nonzero. Without modification, the multiplication of nonzero entries in that process would give 'None', meaning that the unmodified algorithm would fail to insert some nonzero contributions during assembly. To address this, we implement a broadcasting function that copies the right vector data to each process, as shown in Listing B.1.

```
1 def broadcast_vector(vector):
2     vector = MPI.COMM_WORLD.gather(vector, root=0)
3     if vector:
4         vector = [j for i in vector for j in i]
5     else:
6         vector=[]
7     vector = MPI.COMM_WORLD.bcast(vector,root=0)
8     return vector
```

Listing B.1 Broadcasting function to gather the right vector indices and values from the processors to process 0 (line 2) and broadcast the nonzero contributions back to the processors (line 7) during a parallel run.

To improve the share of computational load of each process, we also distribute the left vector data evenly over the processors. The algorithm for this is in Listing B.2.

```
1 def distribute_vector_as_chunks(vector):
2     vector = MPI.COMM_WORLD.gather(vector, root=0)
3     if MPI.COMM_WORLD.Get_rank() == 0:
4         vector = [j for i in vector for j in i]
5         chunks = [[] for _ in range(MPI.COMM_WORLD.Get_size())]
6         for i, chunk in enumerate(vector):
7             chunks[i % MPI.COMM_WORLD.Get_size()].append(chunk)
8     else:
9         vector = None
10        chunks = None
11    vector = MPI.COMM_WORLD.scatter(chunks, root=0)
12    return vector
```

Listing B.2 Function used to distribute the left vector indices and values over the processors during parallel runs. The algorithm initially gathers all nonzero data at the root (line 2). It then distributes the data across the processors as evenly as possible.

These operations are demonstrated in Fig. B.1.

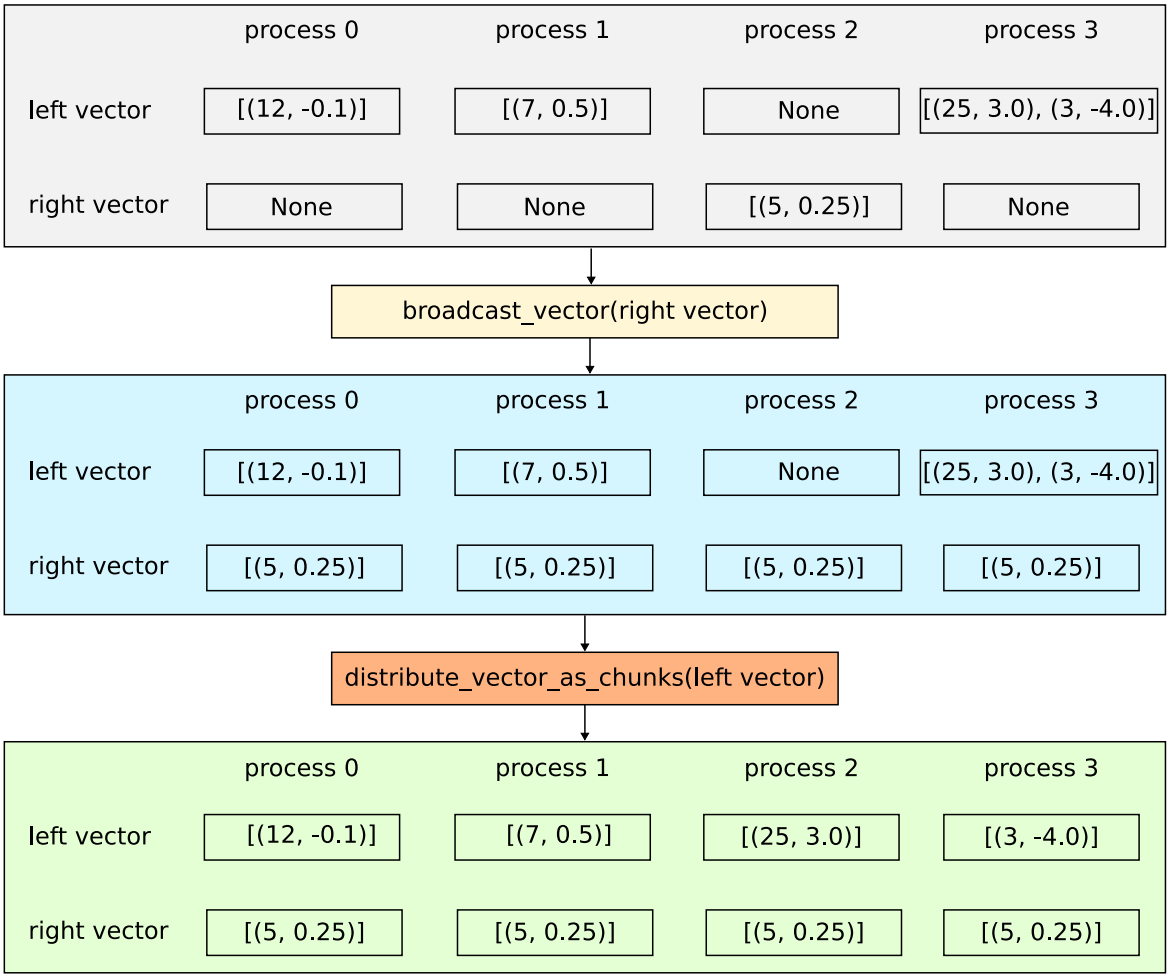


Fig. B.1 Example of the nonzero data handling for left and right vectors using four processors. The left vector's data consists of nonzero row pairs (row indices, row values), while the right vector's data consists of nonzero column pairs (column indices, column values). When the left and right vectors contain non-zero data, the cross multiplication of the values of the vectors can give 'None', so nonzero contributions are lost (gray case). To prevent this, we copy the data of the right vector to each process (blue case) using the algorithm in Listing B.1. We then evenly distribute the data of the left vector using the algorithm described in Listing B.2. This shares the workload among processors (green case). Finally, the left and right vector data become ready for matrix construction.