Domain decomposition techniques for the nonlinear, steady state, finite element simulation of MEMS ultrasonic transducer arrays

Alexandre Halbach
University of Liège, Department of Electrical Engineering
and Computer Science, B-4000 Liège, Belgium

October 11, 2017

Thesis committee:

Prof. Eric Béchet, University of Liège, Président
Dr. Romain Boman, University of Liège
Prof. Christophe Geuzaine, University of Liège, Promoteur
Prof. Michael Kraft, University of Liège & KU Leuven
Dr. Patrice Lemoal, CNRS & FEMTO-ST Institute
Dr. Stéphane Paquay, Open Engineering S.A.

Abstract

With the ever increasing computation power provided by modern computing units comes an ever increasing demand for accurate numerical simulations. Micromachined electromechanical devices (MEMS) such as microscale gyroscopes and accelerometers, ubiquitous in modern smartphones, or capacitive micromachined ultrasonic transducers (CMUTs), the candidate for next generation ultrasound imaging technology, can benefit from ever more accurate simulations. In some circumstances this indeed translates into a more performant final device or simply enables to consider novel devices or working modes.

In the case of arrays of CMUTs, accurate simulations can help to understand the crosstalk phenomenon between individual cells that adds noise to the sensor output and leads to a decrease in the imaging quality. For that purpose large arrays of hundreds of CMUT cells can be simulated. Such simulations are typically carried out on arrays of reduced size. However, when such simulations are carried out on large arrays the problem to simulate becomes computationally untractable on classical workstations and additional assumptions are required. Typical assumptions are to suppose a linear behaviour and to neglect the influence on each other of cells too far apart in the array. While these assumptions provide accurate results for specific excitation settings the accuracy can be further improved when removing them or the constraints on the excitation settings can simply be relaxed. In any case simulating large CMUT arrays without these two assumptions has not been demonstrated yet.

In this thesis we propose to get rid of the two above assumptions: large MEMS and CMUT arrays will be simulated without neglecting the influence of distant membranes while at the same time the nonlinear electromechanical coupling, typical for MEMS devices, will be fully taken into account. Doing so will widen the spectrum of acceptable excitation settings and allow for example the simulation of nonlinearly excited, resonant CMUTs with a large and distant crosstalk. The domain decomposition method will be used to take advantage of parallel computation while the considered steady state time simulations will be carried out in Fourier space.

Acknowledgements

First and foremost I want to thank my advisor Christophe Geuzaine for giving me the chance to start this PhD. In addition to all I had the chance to learn during these four years it gave me the taste for research and a somewhat unexpected passion for the finite element method. Christophe, it has been a pleasure to be your PhD student and I particularly appreciated your open-mindedness and passion for your work.

I would also like to thank my thesis committee for the great advice they provided to improve the manuscript. My sincere thanks also to professor Xiao-Chuan Cai, for welcoming me at the University of Colorado, Boulder and for all his great advice. I also thank Patrick Dular and Romain Boman for their help at any time as well as Véronique Rochus for giving me the chance to present my work at the IMEC research center.

Without computing power there would have been no computations. For the computing clusters that I had the chance to use I would like to thank the Consortium des Equipements de Calcul Intensif (CECI) as well as the Wallonia-Brussels federation and the Walloon region. Many thanks to David Colignon for all the time spent on computer cluster issues.

This section can not be finished without thanking all those who helped me with technical answers and with their good mood. I would like to specially thank Maxime Spirlet for the hours spent on technical issues, Vanessa Mattesi for being the best colleague in my two-person office, Kevin Hogan, Jean-Francois Pirlet, Nicolas Marsic for his advice as well as (listed in random order) Alexandre, Fréderic, Yannick, Kevin, Véronique, Erin, Steven, Francois, Axel, Amaury, Jean de Dieu, Aliaksei. Finally I would like to thank my family for always being there for me.

Contents

Introduction

In this thesis we develop a method to simulate numerically large arrays of nonlinearly excited microelectromachined (MEMS) devices. Such a simulation leads to a finite element discretised algebraic system that is way too large to be solved on a typical workstation or with classical direct resolution techniques. The focus is on arrays of capacitive micromachined ultrasonic transducers (CMUTs), the candidate for next generation ultrasound imaging. When receiving the ultrasound waves reflected by the object to image, a human tissue for example, the individual CMUT cells vibrate with a magnitude and at a timing depending on the reflecting object so that an image of that object can be reconstructed. Unfortunately a vibration on a given cell also induces a parasitic acoustic pertubation on all other cells in the array via the so-called *crosstalk* phenomenon, deteriorating the imaging performance. Simulating that crosstalk on arrays of realistic sizes and with a good accuracy can lead to a better understanding of the origin of the problem and on ways to limit its negative impact.

Simulating such a problem however requires enormous computation power and memory, not available on classical workstations, and is thus in practice performed on arrays of very limited size. Doing so allows to capture only a limited part of the complex reality of the phenomenon. Simplifying assumptions have thus been devised to still be able to quickly perform simulations on realistic array sizes. By neglecting the crosstalk between distant membranes and considering a linear behaviour large arrays could be simulated in a reasonable time in [1, 2].

In this thesis we want to go further and propose a way to simulate in a reasonable time large arrays of CMUTs without neglecting the electromechanical nonlinearity, typical for MEMS devices. The crosstalk between distant CMUT cells will be taken into account so that in the end a nonlinearly excited, resonant CMUT with a large crosstalk between distant membranes can be simulated and the actual complexity of this phenomenon can be grasped. Since the large simulations that will be performed heavily rely on fast data transfers not available in the Matlab code used we choose not to show simulation timings as these might not reflect what can actually be obtained and could lead to wrong conclusions.

For the numerical discretisation of the problems the finite element method with hierarchical high order interpolations will be used. In order to have tens of millions degrees of freedom-strong simulations running in a reasonable time we will take advantage of the high computation power of a parallel computing architecture using a selection of domain decomposition methods whose performance will be compared. Our focus regarding the time dependency of the problem will be on steady state simulations which will be advantageously performed in the Fourier domain while still taking into account all nonlinear effects.

Outline

This thesis is divided into five chapters as follows.

Chapter 1 begins with a general introduction to the working principle of MEMS ultrasonic transducers with a focus on capacitive micromachined ultrasonic transducers (CMUTs). Afterwards a model for the three involved physics (electrostatics, elastodynamics and acoustics) and the underlying assumptions are detailed. The nonlinearity (and the associated pull-in phenomenon) arising from the coupling of the physics is then described. Finally the inter-subdomain crosstalk in CMUT arrays and its negative impact on the imaging quality are discussed and the device geometries used throughout this thesis are detailed.

Chapter 2 starts with an overview of the state of the art for the finite element simulation of large MEMS arrays in steady state as well as the associated intrinsic difficulties. Afterwards the mathematical formulations for each uncoupled physical model are recalled and the finite element discretisations are derived. Coupling schemes to combine the three physics are then detailed. The chapter ends with the selection of an appropriate finite element discretisation for the reference ultrasonic transducers presented in the previous chapter.

Chapter 3 describes an automatic multiharmonic resolution method to get the steady state solution in time and compares it to the classical Newmark time stepping method. After a review of the Newmark method the multiharmonic resolution method is first detailed on a 1D electrostatic problem on a vibrating mesh. Its application to the simulation of CMUTs is then described. Finally the suitability of both methods is assessed for the reference 2D CMUT model.

Chapter 4 describes domain decomposition methods (DDMs) with the goal of solving large nonlinear electroelastoacoustic problems on a parallel computing architecture. The classical DDMs for linear problems are first introduced and the impact of the choice of interface conditions on the convergence rate is discussed. Methods for the nonlinear electroelastoacoustic problem are then detailed and compared. Finally a coarse grid preconditionner is proposed to speed up convergence for large MEMS arrays.

Chapter 5 begins with a 2D and 3D verification of the multiharmonic solver. Simulations are then performed on large 3D CMUT arrays with up to 20 million degrees of freedom to simulate the crosstalk appearing through acoustic waves in the fluid. Finally the fluid is removed and the crosstalk via elastic waves in the bulk is simulated. In any case nonlinearity is taken into account.

Original contributions and communications

To the best of our knowledge the following contributions are original.

1. Providing the final 3D forms of all formulations required to implement the faster converging, strongly-coupled nonlinear electromechanical resolution method derived in [3]

2. Applying the multiharmonic method to the simulation of nonlinearly excited electromechanical devices vibrating in a fluid

3. Combining a multiharmonic resolution and domain decomposition methods for nonlinear electroelastoacoustic problems

4. Investigating the performance of various linear and intrinsically nonlinear domain decomposition techniques for the parallel computation of MEMS

5. Simulating large CMUT arrays without a linearity hypothesis and fully taking the crosstalk into account

It is worth mentioning that all simulations in this thesis have been implemented from scratch in a vectorised Matlab code.

The following publications have been made in journal papers:

1. A. Halbach, P. Dular and C. Geuzaine, "Comparison of nonlinear domain decomposition schemes for coupled electromechanical problems", IEEE Transactions on Magnetics 52 (3), 1-4, 2016

2. A. Halbach and C. Geuzaine, "Automatic derivation of multiharmonic formulations for nonlinear electromechanical problems with time dependent mesh deformation", 17th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), 1-7, 2016

3. A. Halbach and C. Geuzaine, "Steady-state, Nonlinear Analysis of Large Arrays of Electrically Actuated Micromembranes Vibrating in a Fluid", Engineering with Computers, in press

This work has also been presented at the following conferences: COMPUMAG 2015 (Canada), DD23 2015 (South Korea), EUROSIME 2016 (France), ICOSAHOM 2016 (Brazil), DD24 2017 (Norway).

Chapter 1

Physical modeling of micromachined ultrasonic transducers

This chapter begins with a general introduction to the working principle of MEMS ultrasonic transducers with a focus on capacitive micromachined ultrasonic transducers (CMUTs). Afterwards a model for the three involved physics (electrostatics, elastodynamics and acoustics) and the underlying assumptions are detailed. The nonlinearity (and the associated pull-in phenomenon) arising from the coupling of the physics is then described. Finally the inter-subdomain crosstalk in CMUT arrays and its negative impact on the imaging quality are discussed and the device geometries used throughout this thesis are detailed.

1.1 Micromachined ultrasonic transducers

Micromachined ultrasonic transducers are microscale devices able to emit or receive ultrasound through the vibration of a mechanical component. They are part of the wider microelectromechanical systems (MEMS) family. Capacitive micromachined ultrasonic transducers (CMUTs, displayed in figure 1.1) are one example of these devices. They are used in applications such as ultrasound imaging [4, 5, 6, 7] and nondestructive inspection [4, 5, 8, 9, 10, 11]. A 2D model of a modern CMUT with typical dimensions [12, 13] is illustrated in figure 1.2 and the associated material characteristics are listed in table 1.1: an electrical contact (the electrode) is metal-deposited on top of a polysilicon membrane which has a typical length of some tens of micrometers and a typical height of the order of the micrometer. It lies above a vacuum-sealed gap whose height typically is less than a micrometer. The membrane is supported by polysilicon pillars. The vacuum gap, the membrane and the support pillars stand on top of an electrical insulator (silicon dioxyde). This insulator prevents current from flowing from the electrode via the conducting polysilicon to the grounded electrical contact beneath. The whole CMUT finally stands on top of a large polysilicon substrate (also called *bulk*) mechanically clamped at its bottom. Applying an alternating electric voltage between the electrode and the ground creates, due to the tiny electrode-to-ground distance, a downward pointing electrostatic force on the membrane. Because the force magnitude varies in time the membrane vibrates. This vibration moves the fluid that lies on top of the device and this

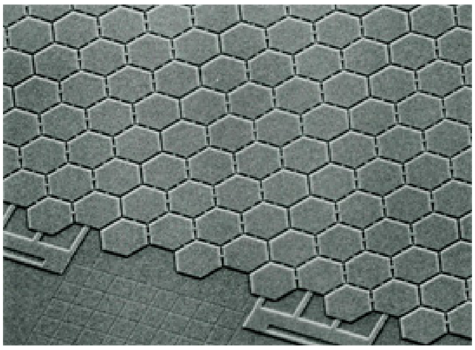


Figure 1.1: Illustration of a modern CMUT array. A single CMUT cell dimension is of the order of ten micrometers. (Source: Hitachi Corp.)

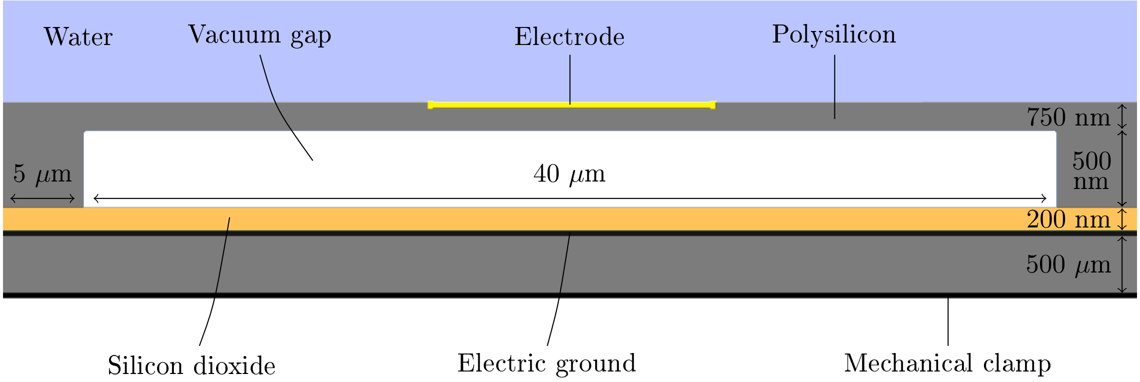


Figure 1.2: Two dimensional illustration of a capacitive micromachined ultrasonic transducer (CMUT) with typical dimensions (not to scale).

radiates ultrasound power. In the medical domain the ultrasound waves propagate through human tissues with a large proportion of water. As an approximation the fluid can thus be considered to be water for such applications.

In the configuration that has just been described the CMUT works as an actuator. When the CMUT is not used as an actuator but as an ultrasound sensor then external ultrasound waves (e.g. the reflected echo of the emitted waves) create a vibration of the membrane and the vibration amplitude and timing can be measured via a capacitance change between the electrode and the ground. With a large array of elementary CMUT cells measuring the magnitude and timing of the reflected waves one can build ultrasound images, e.g. in the medical domain or for nondestructive crack evaluation.

The electric voltage applied between the electrode and the ground typically consists in a large constant bias voltage plus, in case ultrasound is emitted, a tiny alternating voltage that creates a membrane vibration: $v(t) = V_{DC} + v_{AC}\ \sin(2\pi f t)$ V, typically with V_{DC} of the order of 100 V and v_{AC} of the order of 1 V. In a way similar to microphones this bias voltage increases the sensitivity of the sensor and, as will be seen later, leads to a quasi linear vibration around a constant deflection. The electrical excitation frequency f can be very high thanks to the microscale size of the vibrating membranes. For medical applications with a vibration in a water-like fluid, typical frequencies used

Table 1.1: Material properties in a typical CMUT for medical applications.

Material	**Quantity**	**Symbol**	**Typical value**
Polysilicon	Young's modulus	E_{Si}	$150 \cdot 10^9\,\mathrm{N/m}^2$
Polysilicon	Poisson's ratio	$\nu_{\mathbf{Si}}$	0.3
Vacuum	Electric permittivity	ϵ_V	$8.854 \cdot 10^{-12}$ F/m
Water	Volumic mass	ρ	$1000\,\mathrm{kg/m}^3$
Water	Sound propagation speed	c	1484 m/s
$\mathrm{SiO_2}$	Electric permittivity	$\epsilon_{\mathrm{SiO_2}}$	$3.9 \cdot 8.854 \cdot 10^{-12}$ F/m
$\mathrm{SiO_2}$	Young's modulus	$E_{\mathrm{SiO_2}}$	$70 \cdot 10^9\,\mathrm{N/m}^2$
$\mathrm{SiO_2}$	Poisson's ratio	$\nu_{\mathrm{SiO_2}}$	0.17

Table 1.2: Physical quantities involved in the electromagnetic model.

Quantity	**Name**	**Units**
$\boldsymbol{D}$	Electric displacement field	$\mathrm{C/m}^2$
$\boldsymbol{E}$	Electric field	V/m
$\boldsymbol{B}$	Magnetic flux density	T
$\boldsymbol{H}$	Magnetic field	N/(m·A)
$\boldsymbol{J}$	Current density	$\mathrm{A/m}^2$
ρ_v	Volume charge density	$\mathrm{C/m}^3$
ϵ	Electric permittivity	F/m
v	Electric potential	V

in recent devices are in the megahertz range [14, 15, 16]. The corresponding wavelength in water is of the order of the smallest features of the human body to image: a typical 1 MHz frequency indeed corresponds to a wavelength λ in water of $\lambda = \frac{\text{sound speed}}{\text{frequency}} = \frac{c}{f} = \frac{1484\text{ m/s}}{1\text{ MHz}} \approx 1.5$ mm.

1.2 Electromagnetic model: electrostatics

General electromagnetic problems are described by Maxwell's equations. Using the notations defined in table 1.2 they can be written as:

$$\begin{cases} \operatorname{div} \boldsymbol{D} = \rho_v, \\ \operatorname{div} \boldsymbol{B} = 0, \\ \operatorname{curl} \boldsymbol{E} = -\dfrac{\partial \boldsymbol{B}}{\partial t}, \\ \operatorname{curl} \boldsymbol{H} = \boldsymbol{J} + \dfrac{\partial \boldsymbol{D}}{\partial t}. \end{cases} \tag{1.1}$$

For the capacitive, electrostatically-actuated MEMS types under consideration the role of the electric current is solely to charge and discharge the electrode-ground capacitor. Even though the polysilicon in the membrane is an electrical conductor a layer of insulating silicon dioxyde ensures that a negligeable conduction current flows through the membrane. Additionally, for a typical capacitance value $C = 100$ fF [17, 18] under a typical alternating electrode to ground voltage of $1{\cdot}\sin(2\pi{\cdot}10^6 t)$ V the magnitude of the current flowing through the capacitor is $1{\cdot}2\pi{\cdot}10^6{\cdot}C \approx 0.6\ \mu$A. Such a small current, even at 1 MHz, can not create a magnetic field that significantly impacts the huge electric field between the electrode and the ground, and can thus not significantly impact the force applied to the membrane. The electric field is indeed very large: with the typical 1 V voltage drop across the 700 nm gap of figure 1.2 the electric field is $\frac{1}{700\cdot 10^{-9}} = 1.4 \cdot 10^6$ V/m. We will thus neglect all magnetic effects so that $\operatorname{curl} \boldsymbol{E} \approx 0$ and only two relations matter for the modeling:

$$\begin{cases} \operatorname{div} \boldsymbol{D} = \rho_v, \\ \operatorname{curl} \boldsymbol{E} = 0. \end{cases} \tag{1.2}$$

Assuming isotropic materials gives $\boldsymbol{D} = \epsilon \boldsymbol{E}$ so that

$$\begin{cases} \operatorname{div} \epsilon \boldsymbol{E} = \rho_v, \\ \operatorname{curl} \boldsymbol{E} = 0. \end{cases} \tag{1.3}$$

The second equality in (1.3) means that one can find a scalar potential field v (the electric potential field) such that $\boldsymbol{E} = -\nabla v$. With this relation the first equality finally gives the electrostatic equation

$$\operatorname{div}(\epsilon \nabla v) = -\rho_v. \tag{1.4}$$

All electrical problems in this work will be solved using equation (1.4).

1.3 Mechanical model: elastodynamics

Mechanical problems are ruled by a general equilibrium relation linking the stresses inside the material, the applied external forces and the inertia terms. With the notations of table 1.3 a solid volume is at equilibium if

$$\operatorname{div} \boldsymbol{\sigma} + \boldsymbol{f} = \rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \tag{1.5}$$

that is if the stresses on the volume boundaries and the external forces exactly compensate the inertia terms. Equation (1.5) alone does not allow to compute all the unknown fields. An additional relation linking the Cauchy stress tensor $\boldsymbol{\sigma}$ to the strain tensor $\boldsymbol{\varepsilon}$ as well as a way to compute the strains $\boldsymbol{\varepsilon}$ in terms of the displacement $\boldsymbol{u}$ are needed [19]. The two latter relations depend on the mechanical model we select: a better accuracy is obtained at the expense of an increased complexity of the equations. For the ultrasonic transducer devices considered nonlinear models have been analysed [20, 21]. However, because of the large aspect ratio (length to height) of CMUT membranes, even at the maximum membrane deflection the rotations and elongations in

Table 1.3: Physical quantities involved in the mechanical model.

Quantity	**Name**	**Units**
$\boldsymbol{\sigma}$	Cauchy stress tensor	N/m^2
$\boldsymbol{\varepsilon}$	Infinitesimal strain tensor	
C	Hooke's law stiffness tensor	Pa
$\boldsymbol{f}$	Body force per unit volume	N/m^3
$\boldsymbol{u}$	Mechanical displacement $[u_x\ u_y\ u_z]^T$	m
u_x	Displacement in the $\boldsymbol{e}_x$ direction	m
u_y	Displacement in the $\boldsymbol{e}_y$ direction	m
u_z	Displacement in the $\boldsymbol{e}_z$ direction	m
ρ	Volumic mass	kg/m^3

the polysilicon material are small. For the typical dimensions of figure 1.2 with a 40 μm vacuum length and 500 nm height the maximum central displacement does not exceed in practice 200 nm. Considering a linearly increasing membrane displacement from the vacuum sides to the center gives a rough approximation of the maximum membrane rotation for a first mode mechanical vibration: $\alpha = tan^{-1}\left(\frac{200 \text{ nm}}{20\ \mu\text{m}}\right) \approx 0.6°$. The half membrane length above the vacuum gap is then elongated from 20 μm to $\frac{20\ \mu\text{m}}{\cos(\alpha)} \approx 20\ \mu\text{m} + 1$ nm giving a tiny deformation of about $5 \cdot 10^{-5}$. With such small deformations and relatively small rotations a linear elasticity model is typically used [22, 1, 23]. Furthermore because the membrane thickness over length ratio is still rather large ($\frac{0.75}{40}$) the geometrical nonlinearity can be ignored. With the notations defined in table 1.3 the elasticity model can be written as follows:

$$\begin{cases} \operatorname{div}\boldsymbol{\sigma} + \boldsymbol{f} = \rho\,\dfrac{\partial^2 \boldsymbol{u}}{\partial t^2}, \\[2ex] \boldsymbol{\varepsilon} = \dfrac{1}{2}[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T], \\[2ex] \boldsymbol{\sigma} = \mathsf{C} : \boldsymbol{\varepsilon}, \end{cases} \tag{1.6}$$

with

$$\begin{aligned} \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^T &= \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix}, \\ \boldsymbol{\sigma} = \boldsymbol{\sigma}^T &= \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}, \end{aligned} \tag{1.7}$$

where $\mathsf{C} : \boldsymbol{\varepsilon} := \sum\limits_{kl} c_{ijkl}\varepsilon_{kl}$.

Table 1.4: Physical quantities involved in the fluid model.

Quantity	**Name**	**Units**
$\boldsymbol{v}$	Fluid velocity	m/s
f	Body force per unit volume	N/m^3
p	Pressure	Pa
ρ	Volumic mass	kg/m^3
μ	Dynamic viscosity	Pa·s

1.4 Fluid model: linear acoustics

The water fluid used in CMUTs designed for the medical domain is assumed to be Newtonian and it thus follows the Navier–Stokes equations for compressible Newtonian fluids. With the notations of table 1.4 they can be written as

$$\begin{cases} \rho\,(\dfrac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}) = -\nabla p + \operatorname{div}(\mu\,(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T)) + \nabla(-\dfrac{2}{3}\,\mu\operatorname{div}\boldsymbol{v}) + \boldsymbol{f}, \\ \dfrac{\partial \rho}{\partial t} + \operatorname{div}(\rho\boldsymbol{v}) = 0. \end{cases} \tag{1.8}$$

In the first equality one can find the inertia terms on the left followed by the pressure force, two viscous forces and the external forces. To (1.8) an extra relation linking the pressure and density has to be added. For an isentropic process, i.e. at constant entropy S, one gets

$$c = \sqrt{\frac{\partial p}{\partial \rho}}, \tag{1.9}$$

where c is the speed of sound in the fluid. For small density and pressure variations this equation holds.

The viscosity terms in (1.8) are neglected since only the region within a few acoustic wavelength of the source is of interest in this work. A very simple artificial viscous damping could however be incorporated in the model as done in [24] to take into account losses of all origins. For an actual viscoelastic model refer to e.g. [1]. Considering an inviscid fluid and considering no external forces (1.8) rewrites as

$$\begin{cases} \rho\,(\dfrac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v}) = -\nabla p, \\ \dfrac{\partial \rho}{\partial t} + \operatorname{div}(\rho\boldsymbol{v}) = 0, \end{cases} \tag{1.10}$$

which can, for tiny perturbations, be linearised around a mean value:

$$\begin{cases} p = \overline{p} + \delta p, \\ \rho = \overline{\rho} + \delta\rho, \\ \boldsymbol{v} = \overline{\boldsymbol{v}} + \boldsymbol{\delta v} = \boldsymbol{\delta v}, \end{cases} \tag{1.11}$$

where the overlined quantities are the mean values, constant in space and time and where the δ terms are tiny perturbations around the mean value of the pressure, the density and the fluid velocity respectively. Since the fluid is at rest $\overline{\boldsymbol{v}}$ equals zero. Injecting (1.11) into (1.10) and neglecting nonlinear perturbations gives

$$\begin{cases} \overline{\rho}\,\dfrac{\partial \boldsymbol{\delta v}}{\partial t} + \delta\rho\,\dfrac{\partial \boldsymbol{\delta v}}{\partial t} + \overline{\rho}\,(\boldsymbol{\delta v}\cdot\nabla)\boldsymbol{\delta v} = -\nabla\delta p, \\ \dfrac{\partial \delta\rho}{\partial t} + \overline{\rho}\,\mathrm{div}\,\boldsymbol{\delta v} = 0. \end{cases} \tag{1.12}$$

The second and third terms of the first relation in (1.12) cancel out. Indeed, using the derivative of a product rule and the second relation multiplied by $\boldsymbol{\delta v}$:

$$\begin{aligned} \overline{\rho}\,(\boldsymbol{\delta v}\cdot\nabla)\boldsymbol{\delta v} &= \overline{\rho}\sum_{j=1}^{3}\delta v_j\frac{\partial \delta v_i}{\partial x_j} \\ &= -\overline{\rho}\sum_{j=1}^{3}\delta v_i\frac{\partial \delta v_j}{\partial x_j} + \overline{\rho}\sum_{j=1}^{3}\frac{\partial(\delta v_j\delta v_i)}{\partial x_j} \\ &\approx -\overline{\rho}\sum_{j=1}^{3}\delta v_i\frac{\partial \delta v_j}{\partial x_j} = -\overline{\rho}\,\mathrm{div}\,\boldsymbol{\delta v}\,\boldsymbol{\delta v} \\ &= \frac{\partial\delta\rho}{\partial t}\,\boldsymbol{\delta v} = \frac{\partial(\delta\rho\,\boldsymbol{\delta v})}{\partial t} - \delta\rho\frac{\partial\boldsymbol{\delta v}}{\partial t} \\ &\approx -\delta\rho\frac{\partial\boldsymbol{\delta v}}{\partial t}, \end{aligned} \tag{1.13}$$

neglecting again second order perturbations. As can be seen with (1.13) the two middle terms in the first relation of (1.12) cancel out so that

$$\begin{cases} \overline{\rho}\,\dfrac{\partial \boldsymbol{\delta v}}{\partial t} + \nabla\delta p = 0, \\ \dfrac{\partial \delta\rho}{\partial t} + \overline{\rho}\,\mathrm{div}\,\boldsymbol{\delta v} = 0. \end{cases} \tag{1.14}$$

Taking the divergence of the first relation and the time derivative of the second one gives

$$\begin{cases} \overline{\rho}\;div\,\dfrac{\partial \boldsymbol{\delta v}}{\partial t} + \Delta\delta p = 0, \\ \dfrac{\partial^2 \delta\rho}{\partial t^2} + \overline{\rho}\,\mathrm{div}\,\dfrac{\partial \boldsymbol{\delta v}}{\partial t} = 0, \end{cases} \tag{1.15}$$

which can be combined into a single one

$$\frac{\partial^2\delta\rho}{\partial t^2} - \Delta\delta p = 0. \tag{1.16}$$

With the isentropic approximation (1.9) the acoustic wave equation can be written:

$$\frac{1}{c^2}\frac{\partial^2\delta p}{\partial t^2} - \Delta\delta p = 0, \tag{1.17}$$

with c the speed of sound in the fluid and δp the pressure variation around the mean pressure.

It is worth noting that alternative acoustic formulations have been proposed. A displacement-based acoustic formulation [25] uses displacement unknowns in the fluid, simplifying the fluid-structure coupling. This formulation however requires three times the unknowns of the scalar pressure-based formulation. Moreover spurious non-zero frequency circulation modes have been observed [26]. One can also use a scalar velocity potential-based formulation [27, 28, 26] which has the advantage, unlike the pressure formulation, of being symmetric. An additional irrotational flow hypothesis is however required.

1.5 Nonlinear coupling, pull-in phenomenon and crosstalk

The mathematical models selected for the electromagnetic, mechanical and fluid problems are all linear when uncoupled. Coupling the elastic and the electrostatic model together however brings in nonlinearity and the *pull-in* phenomenon, also called *pull-in* instability [29, 30]. For illustration purposes let us consider in figure 1.3 a very simple static 1D model of the CMUT described in figure 1.2 (a circular 3D membrane shape is used). The model consists of an infinitely thin electrode connected via a spring of stiffness k (N/m) to a clamped and electrically grounded bottom electrode. Applying a constant voltage V between the upper and lower electrode creates an electrostatic force (N) pushing the upper electrode downwards, closer to the bottom one. In case the voltage is set to 0 the equilibrium distance between both electrodes is u_0. The quantity u (m) measures the difference between this position and the actual electrode position. It approximates the peak deflection of the 3D membrane.

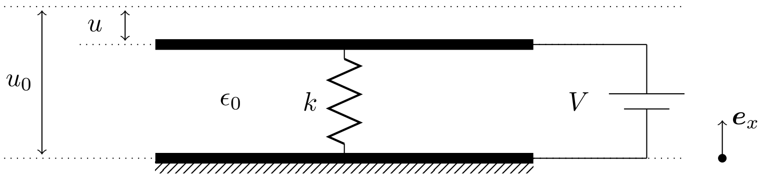


Figure 1.3: Lumped model of a membrane electrically actuated by a time-independent voltage V

Two forces are at play in figure 1.3 namely the electrostatic force and the spring restoring force. The spring force (N) simply equals

$$\boldsymbol{f}_{\text{spring}} = -k\,u\,\boldsymbol{e}_x \tag{1.18}$$

where the stiffness k can be computed using plate theory as done in [18]:

$$k = \frac{\boldsymbol{f}_{\text{spring}}}{u} = \frac{64\pi E_{\text{Si}} h^3}{12\,a^2(1-\nu_{\text{Si}}^2)} = 2912\,\text{N/m}, \tag{1.19}$$

where a is the membrane radius and h its height. In this simple one dimensional setting the electrostatic force between the electrodes can be computed as follows:

$$\boldsymbol{f}_{\text{electrostatic}} = \frac{1}{2} q\,\boldsymbol{E} = -\frac{1}{2}\left(\epsilon_0 \frac{V}{(u_0+u)}\right)\frac{V}{(u_0+u)} A\,\boldsymbol{e}_x = -\frac{1}{2}\epsilon_0 \frac{V^2}{(u_0+u)^2} A\,\boldsymbol{e}_x, \tag{1.20}$$

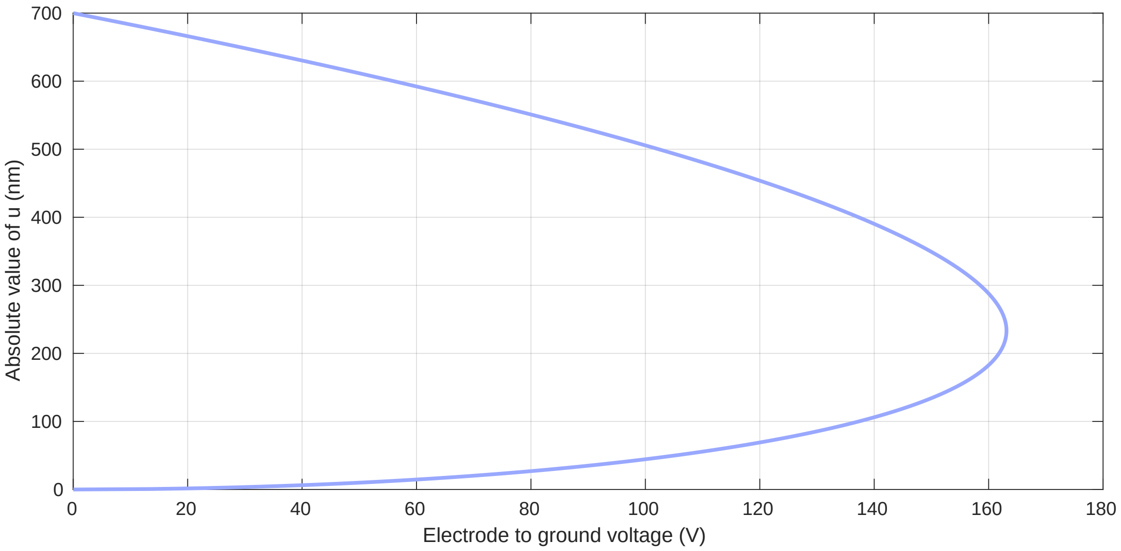


Figure 1.4: Deflection u (nm) versus electrode voltage (V) for the 1D electrically actuated membrane model. The largest deflection corresponds to $u = u_0$.

with $u_0 = 700$ nm (the membrane thickness is not taken into account since it is made of polysilicon, a conductor) and where the surface $A = 1.26 \cdot 10^{-9}$ m^2. In practical applications the computation is not straightforward and numerical methods will be used for the accurate computation of the electrostatic force on complex geometries [31, 32, 33, 34, 3, 35].

With (1.20) the static equilibrium relation writes

$$-\,k\,u - \frac{1}{2}\epsilon_0 \frac{V^2}{(u_0+u)^2} A = 0, \tag{1.21}$$

which is quadratic in V. This nonlinearity is clearly visible in figure 1.4 where relation (1.21) is plotted: for a sweep on the time-independent electrical excitation the associated displacement u is displayed. There are two equilibrium displacements for every voltage, one stable, on the side below the voltage maximum and one unstable, on the side above the maximum. The unstable equilibrium will end up either in the stable region or collapsed to the grounded electrical contact. Beyond a limit voltage of about 160 V there is no more solution. What happens is that as the voltage is increased slightly beyond a threshold the upper electrode suddenly collapses on the bottom electrode because the restoring force can not compensate the ever increasing electrostatic force anymore: this is called the *pull-in* phenomenon. As a rule of thumb pull-in typically occurs at a displacement of about a third of the vacuum height. CMUTs working in collapse mode, i.e. beyond pull-in, have been investigated [36, 37, 13]. For all typical CMUTs considered in this work however the collapse mode will be avoided.

Unlike the electro-mechanical interaction, the fluid-structure interaction is linear. It requires coupling terms at the fluid-solid interface [38, 39]:

$$\nabla \delta p \cdot \boldsymbol{n} \equiv \frac{\partial \delta p}{\partial \boldsymbol{n}} = -\rho_{\text{fluid}} \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{n} \quad \text{and} \quad \boldsymbol{f}_{\text{pressure}} = -\delta p\, \boldsymbol{n}, \tag{1.22}$$

where $\boldsymbol{n}$ is the normal to the membrane pointing towards the fluid. The first relation is Newton's law linking a pressure gradient to an acceleration and the second one links the pressure to a surface force on the membrane.

In an array of micromembranes vibrating in a fluid, particularly when close to membrane resonance, the vibration of one membrane can affect the vibration of another membrane in the array. This phenomenon is called crosstalk. It has a negative impact on the imaging quality in e.g. medical applications [40]. Analyses of the crosstalk have shown [41, 42, 43] that it mainly appears via surface waves at the fluid-structure interface and via Lamb waves propagating in the silicon substrate. For medical applications the coupling between membranes via the water fluid is high compared e.g. to air and the focus will be on this first form of crosstalk.

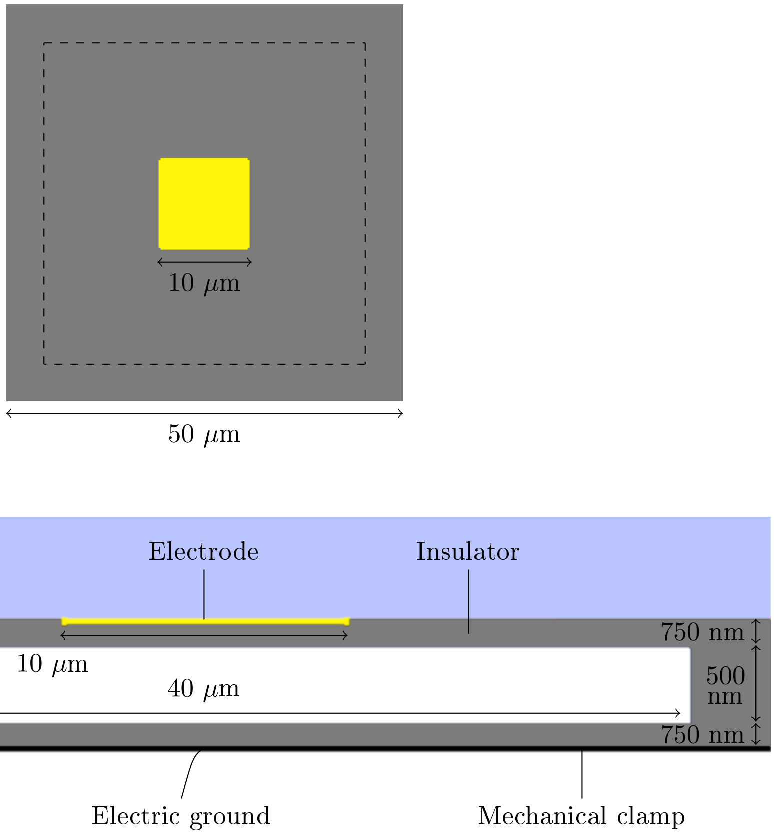


Figure 1.5: Illustration of the 3D test case used throughout this thesis. Top view (top) and side view (bottom). The 2D test case is the side view (bottom) of the 3D test case.

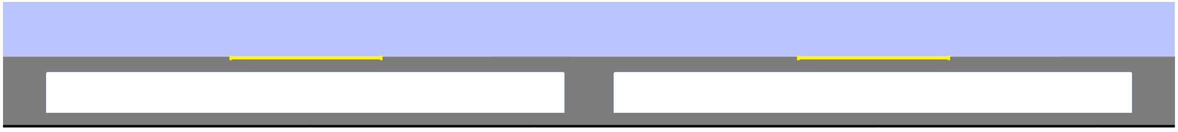


Figure 1.6: Illustration of a 2 by 1 array of 2D CMUTs.

1.6 Reference ultrasonic transducer geometries

In figure 1.5 we define a representative CMUT for medical applications in order to highlight the main properties of the methods used in this thesis and be able to compare them on identical geometries. The dimensions used correspond to the typical dimensions of a modern CMUT [12]. They are summarised along with the material properties in table 1.5 and table 1.6 respectively.

Table 1.5: Dimensions of the reference CMUT geometries.

Membrane thickness	750 nm
Vacuum height	500 nm
Vacuum length	40 μm
Electrode length	10 μm
Support pillars thickness	5 μm

Table 1.6: Material properties in the reference CMUT geometries.

Material	**Quantity**	**Symbol**	**Typical value**
Insulator	Young's modulus	E_I	$150 \cdot 10^9\,\mathrm{N/m^2}$
Insulator	Poisson's ratio	ν_I	0.3
Insulator	Electric permittivity	ϵ_I	$11.7 \cdot 8.854 \cdot 10^{-12}$ F/m
Insulator	Volumic mass	ρ_I	$2330\,\mathrm{kg/m^3}$
Vacuum	Electric permittivity	ϵ_V	$8.854 \cdot 10^{-12}$ F/m
Water	Volumic mass	ρ	$1000\,\mathrm{kg/m^3}$
Water	Sound propagation speed	c	1484 m/s

The 2D test geometry (i.e. the side view of the 3D geometry) will most of the time not be used alone but rather in a two-dimensional array with n elementary membranes. An array with two membranes is depicted in figure 1.6. This array is recurrent in this thesis and it thus deserves some preliminary analysis. The first important value is the pull-in voltage of an individual membrane, measured by increasing the time-independent voltage applied between the electrode and the ground. This voltage sweep is shown in figure 1.7. The pull-in voltage in the 2D test case settings is slightly above 110 V. It is worth noting that the maximum membrane deflection before pull-in is of the order of a third of the vacuum gap height, a general rule of thumb.

Another important characteristic is the first resonance frequency. To get it let us excite the 2×1 array with an electrical excitation leading to a linear vibration. The electrical excitation is set to a 10 V time-constant value on both membranes plus a 1 V sine excitation applied only to the left membrane. The absolute value of the maximum displacement is displayed in figure 1.8 for an electrical excitation frequency ranging from close to DC to the second resonance peak. The first resonance appears for an electrical excitation frequency slightly above 1 MHz. Although not

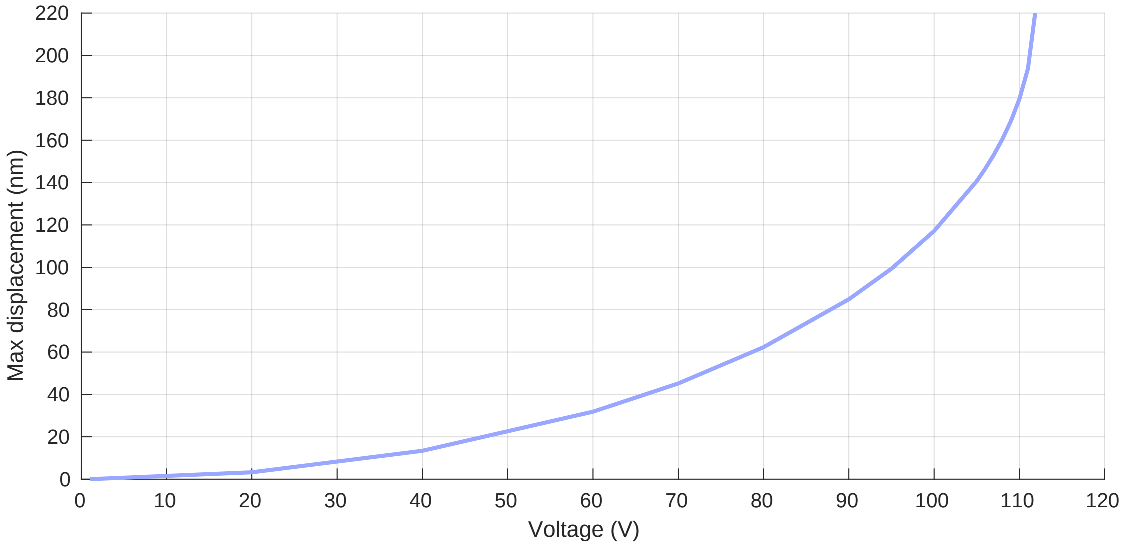


Figure 1.7: Maximum deflection for a range of time-independent electric excitation voltages on the 2D test case of figure 1.5.

exactly the same because of the different electrical excitation the left and the right membrane share a similar resonance frequency. The crosstalk is clearly visible.

Because of the selected electrical excitation the behaviour in the current test settings is linear everywhere but very close to resonance. Indeed the displacements are small compared to the pull-in displacement and the alternating voltage is much smaller than the constant term. Linearity implies that the mechanical displacement $\boldsymbol{u}$ can be written as

$$\boldsymbol{u}(x,t) = \boldsymbol{U}_0(x) + \boldsymbol{U}_{s1}(x)\sin(2\pi f_0 t) + \boldsymbol{U}_{c1}(x)\cos(2\pi f_0 t), \tag{1.23}$$

for the considered $v(x,t) = 10 + 1 \cdot \sin(2\pi f_0 t)$ V electrical excitation on the left membrane and $v(x,t) = 10$ V on the right one. Using this decomposition gives additional insights into the behaviour of the membranes. The constant deflection $\boldsymbol{U}_0(x)$, in-phase vibration $\boldsymbol{U}_{s1}(x)$ and quadrature vibration $\boldsymbol{U}_{c1}(x)$ versus frequency plot are shown in figure 1.9 for the left membrane and in figure 1.10 for the right membrane. It is worth noting that the expected 90° phase shift at resonance is correctly captured in the figures since at the resonance peak the quadrature component is dominant.

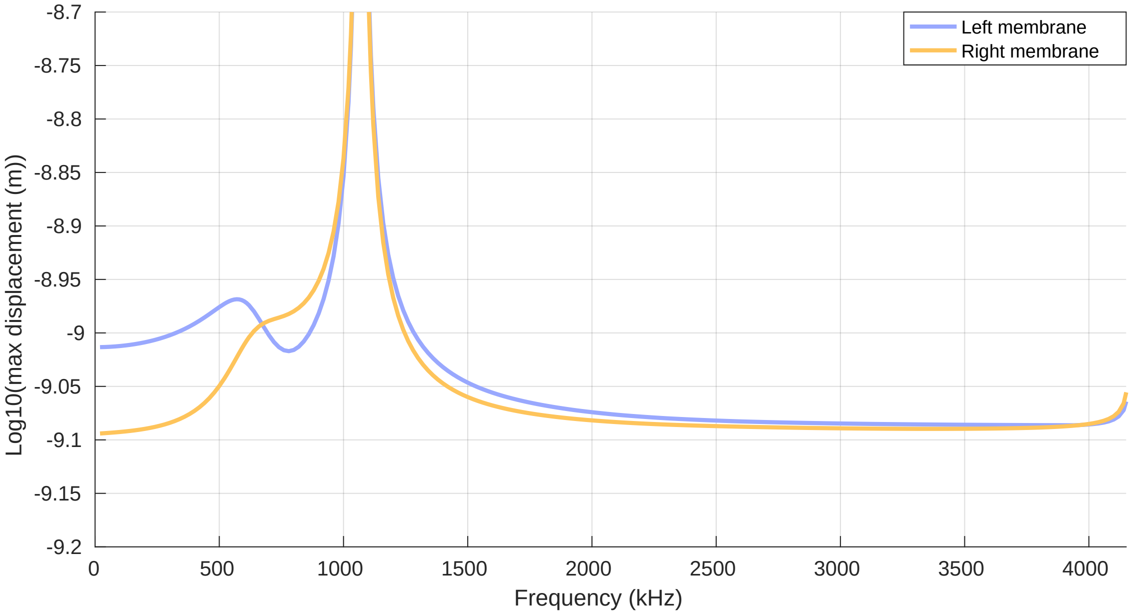


Figure 1.8: Absolute value of the maximum deflection on the left and right membrane of the 2×1 array test case of figure 1.6 as the electrical excitation frequency is swept from close to DC till the second resonance frequency. The electrical excitation consists in a 10 V constant excitation on both membranes plus a 1 V sine excitation applied only to the left membrane.

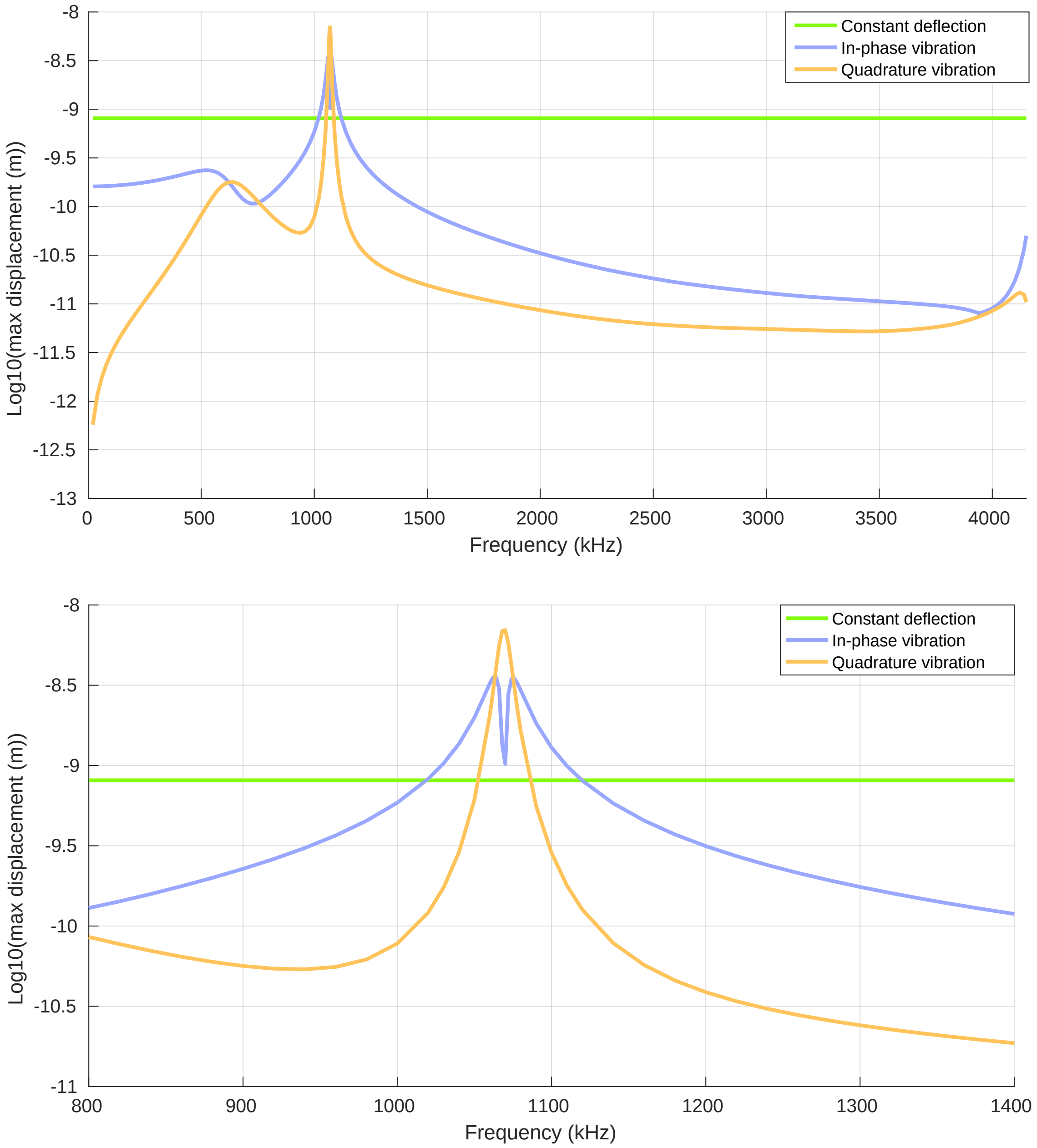


Figure 1.9: Maximum deflection of $\boldsymbol{U}_0(x)$, $\boldsymbol{U}_{s1}(x)$ and $\boldsymbol{U}_{c1}(x)$ on the left membrane of the 2×1 array test case of figure 1.6 versus electrical excitation frequency f. The electrical excitation is $v(t) = 10 + 1 \cdot \sin(2\pi f t)$ V and $v(t) = 10$ V on the left and right membranes respecitvely. The bottom plot is a zoom around the first resonance.

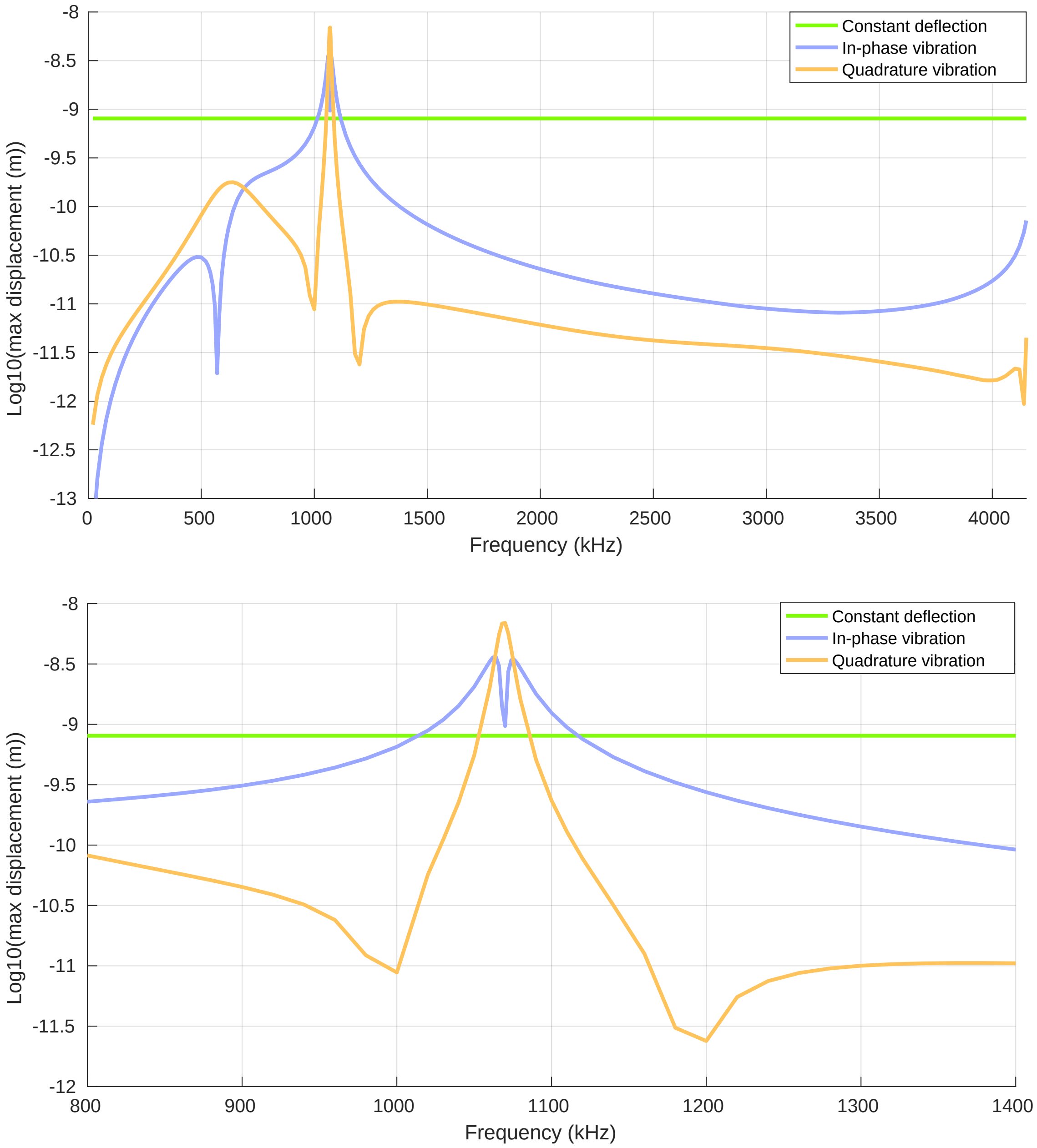


Figure 1.10: Maximum deflection of $\boldsymbol{U}_0(x)$, $\boldsymbol{U}_{s1}(x)$ and $\boldsymbol{U}_{c1}(x)$ on the right membrane of the 2×1 array test case of figure 1.6 versus electrical excitation frequency f. The electrical excitation is $v(t) = 10 + 1 \cdot \sin(2\pi f t)$ V and $v(t) = 10$ V on the left and right membranes respecitvely. The bottom plot is a zoom around the first resonance.

Chapter 2

Numerical simulation with finite elements

This chapter begins with an overview of the state of the art for the finite element simulation of large MEMS arrays in steady state as well as the associated intrinsic difficulties. Afterwards the mathematical formulations for each uncoupled physical model are recalled and the finite element discretisations are derived. Coupling schemes to combine the three physics are then detailed. The chapter ends with the selection of an appropriate finite element discretisation for the reference ultrasonic transducers presented in the previous chapter.

2.1 State of the art and challenges

MEMS devices are modeled with multiple techniques providing different degrees of accuracy at varying computational costs. Lumped models provide approximations of physical quantities at very low computational cost since they consist of only a few degrees of freedom. They however lack in general the accuracy of more advanced, but also computationally more intensive numerical methods. This section provides an overview of the main methods used to model MEMS and CMUT devices [22, 44, 45].

2.1.1 Simulation using lumped models

Lumped parameter models can be used to get a fast-to-solve approximation of a physical quantity of interest. Instead of solving a physical problem with advanced, computationally intensive numerical methods a lumped model only requires to solve an equivalent system with a few degrees of freedom. Getting the parameters of the lumped model can be done in several ways such as by physically measuring them, approximating them analytically or computing them with a numerical method at a given working point. As an illustration, an equivalent stiffness can be computed on the reference CMUT of figure 1.5 to correspond to the stiffness k in the single degree of freedom model of figure 1.3. Note that the equivalent stiffness k is only valid for the settings at which it was measured. With that stiffness determined, all parameters of this very simple lumped model are known and it can be used to very quickly compute an approximation of the pull-in displacement and voltage of the reference CMUT. Indeed using the notations of section 1.5 the electric potential is a function of the deflection u:

$$V = \sqrt{\frac{-2ku\left(u_0+u\right)^2}{\epsilon_0 A}} \tag{2.1}$$

and the pull-in displacement is such that $\frac{\partial V(u_{\text{pull-in}})}{\partial u} = 0$. The derivative $\frac{\partial V}{\partial u}$ is

$$\frac{\partial V}{\partial u} = -\frac{1}{2\sqrt{\frac{-2ku\left(u_0+u\right)^2}{\epsilon_0 A}}} \cdot \frac{2k}{\epsilon_0 A}\left[\left(u_0+u\right)^2 + 2u\left(u_0+u\right)\right] \tag{2.2}$$

which equals zero for the pull-in displacement $u = -\frac{1}{3}u_0$. The corresponding pull-in voltage is

$$V(-\frac{1}{3}u_0) = \sqrt{\frac{8ku_0^3}{27\epsilon_0 A}}. \tag{2.3}$$

The lumped model has the further advantage that the pull-in voltage and displacement can be immediately recomputed if for example the vacuum gap is replaced by a dielectric gas. Additionally, general trends can be immediately read from (2.3): the pull-in voltage strongly increases with the gap height but decreases for high permittivity materials.

Because of the few involved quantites, the low time-to-solution as well as the ability to read trends in the analytic equations, lumped models are widely used to model MEMS: the pull-in voltage and displacement was approximated with a formula similar to (2.3) for a resonant gate transistor in [30] and for fixed-fixed microbeams in [46], the deflection of a cantilever microswitch under various static electrical excitations was calculated in [47] while (weak) nonlinearity in MEMS devices was modeled in [48, 44]. An overview of lumped models for MEMS devices can be found in [49]. Modeling arrays of CMUT devices and the fluid coupling between individual CMUTs is typically done via the use of self and mutual *radiation impedances* [50, 51, 52, 53, 54, 55] linking the velocity of a pressure radiating membrane to the subsequent pressure forces applied to the membrane of another CMUT cell. Several commercial software tools (e.g. Coventor [56]) implement such lumped models.

Unfortunately the major advantage of lumped models, namely the few number of degrees of freedom, is also their main drawback: it only deals with a very simplified model of reality and of the actual MEMS geometry and can thus not handle details as well as an advanced numerical method can. As a result lumped models are only able to provide a (possibly rough) approximation over a limited range of working settings and MEMS geometries [53, 57, 58, 47]. A better representation of the actual physics can only be obtained at the expense of an ever increasing complexity and number of degrees of freedom in the model.

2.1.2 Simulation using the finite element method

Numerical simulations of MEMS with the finite element method have the ability to capture more details in the physics and in the geometries than lumped models [53, 57, 58, 47]. They are nowadays widely used for the simulation of MEMS and CMUT devices, often via the use of commercial software such as COMSOL, ANSYS or Abaqus [1, 59, 60, 61, 62]. Even though a 2D simulation leads to a reduced computational cost, 3D simulations often need to be carried out to accurately capture physical quantities when the 3D effects can not be neglected: in [62] a 2D simulation accurately captured the maximum deflection of a fixed-fixed microbeam while in [63, 64] it was observed that an accurate prediction of the pull-in voltage on the considered MEMS could neither be obtained with lumped models nor with a 2D finite element simulation so that a computationally demanding 3D simulation was performed to closely match experimental results.

2.1.3 Challenges for the finite element simulation of large CMUT arrays

Exciting CMUT arrays close to membrane resonance can dramatically strengthen the vibration and thus the emitted output pressure in emission and the sensitivity in reception. Unfortunately working close to resonance can lead to a large crosstalk in arrays made up of replicated identical cells. This crosstalk then heavily depends on the membrane boundary conditions and thus on the position in the array. To accurately capture the crosstalk the problem can therefore not be solved simply with periodicity conditions and the full array must be simulated.

Because of the high computational cost practical finite element simulations of CMUT arrays are limited to only a few cells [65, 21, 57, 59, 66, 67]. The computation indeed rapidly becomes untractable without simplification for large nonlinear arrays using standard direct solvers. For this reason a wide range of simplifying approximations are used in practice, such as periodicity conditions [68], supposing clamped-clamped membranes [69], using beam theory to compute the membrane deflection [70, 71], supposing flat CMUTs in order to use the Rayleigh integral to compute the pressure field [1], neglecting the fluid coupling between distant membranes [1] or considering a linear behaviour [1, 2]. While some approximations are fully valid for some CMUTs others like the fluid coupling restriction to the close neighbours or the linearity assumption can be easily violated, for example when close enough to resonance. Nevertheless, with simplifying approximations including at least linearity the resolution of large arrays of CMUTs in a reasonable amount of time has been demonstrated [1, 2]. Work has also been done to take into account nonlinearity on single electromechanical micromembranes (vibrating in vacuum) in steady state [65, 21] and for the transient simulation of arrays thereof [23].

To the best of our knowledge, getting in a reasonable amount of time the nonlinear solution of large arrays of electromechanical micromembranes vibrating in a fluid (e.g. CMUTs) has not been demonstrated, neither for the transient nonlinear solution, nor for the nonlinear steady state solution. Capturing the steady state in a nonlinear simulation of large CMUT arrays can be even more challenging than capturing a transient since by definition the steady state is obtained only after all transients are damped enough, which for CMUT with low damping and working close to resonance takes a long time compared to the electrical excitation period.

2.2 Finite element formulations

The partial differential equations (1.4), (1.6) and (1.17) described in the previous chapter model respectively the uncoupled electromagnetic, elastodynamic and acoustic physics involved in MEMS ultrasonic transducers. In this section they are reformulated for a numerical resolution using the

(Galerkin) finite element method [72] (refer to appendix A for an overview of the method and implementation details).

2.2.1 Electrostatics

The partial differential equation (1.4) describing mathematically the electrostatic problem is:

$$\mathrm{div}(\epsilon \nabla v) = -\rho_v. \tag{2.4}$$

For the reference CMUT of figure 1.5 the domain $\mathbb{R}^3$ is truncated (refer to section 2.2.4) so as to obtain a finite domain Ω. The domain Ω_e^* on which (2.4) is defined is equal to the truncated domain: it includes the fluid, the solid and the vacuum region in figure 1.5. However, since the electric potential v depends on how the electric domain is deformed by the mechanical displacement $\boldsymbol{u}$, Ω_e^* equals the truncated domain Ω deformed by the mechanical displacement $\boldsymbol{u}$. This is denoted by the star symbol.

Form (2.4) is called the *strong* form of the partial differential equation. In the (Galerkin) finite element method it is is not solved as such. It is instead solved in its so-called *weak* form: for a problem defined on domain Ω_e^* the strong form implies that

$$\int_{\Omega_e^*} \mathrm{div}(\epsilon \nabla v)\, v'\, d\Omega_e^* = \int_{\Omega_e^*} -\rho_v\, v'\, d\Omega_e^*, \tag{2.5}$$

holds for any appropriate function v' called *test function.* The corresponding boundary conditions are detailed in section 2.3. A rigorous mathematical framework can be found in [72, 73, 74, 75]. One can then apply the generalised integration by parts formula (i.e. Green's formula) to get the actual weak form for the electrostatic problem:

$$-\int_{\Omega_e^*} \epsilon \nabla v \cdot \nabla v' d\Omega_e^* + \int_{\partial\Omega_e^*} v'\, \epsilon\, \partial_{\boldsymbol{n}} v\, d\partial\Omega_e^* = \int_{\Omega_e^*} -\rho_v\, v'\, d\Omega_e^*, \tag{2.6}$$

where $\partial\Omega_e^*$ is the boundary of Ω_e^*, $\boldsymbol{n}$ is the outward-pointing normal to the boundary and $\partial_{\boldsymbol{n}} v = \nabla v \cdot \boldsymbol{n}$ is the normal derivative of v.

2.2.2 Elastodynamics

The system of partial differential equations for the elastodynamic model (1.6) is:

$$\begin{cases} \mathrm{div}\, \boldsymbol{\sigma} + \boldsymbol{f} = \rho \dfrac{\partial^2 \boldsymbol{u}}{\partial t^2}, \\ \boldsymbol{\varepsilon} = \dfrac{1}{2}[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T], \\ \boldsymbol{\sigma} = \mathsf{C} : \boldsymbol{\varepsilon}, \end{cases} \tag{2.7}$$

with

$$\begin{aligned}\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^T &= \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix}, \\ \boldsymbol{\sigma} = \boldsymbol{\sigma}^T &= \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix},\end{aligned} \tag{2.8}$$

and $\boldsymbol{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$. In the reference CMUT problem of figure 1.5 the volume forces in $\boldsymbol{f}$ only consist of electrostatic forces and pressure forces.

Using Green's formula the equation of motion in (2.7) leads to:

$$\begin{aligned}&\operatorname{div}\boldsymbol{\sigma} + \boldsymbol{f} = \rho\frac{\partial^2\boldsymbol{u}}{\partial t^2} \\ \Rightarrow &\int_{\Omega_m} \operatorname{div}\boldsymbol{\sigma}\cdot\boldsymbol{u}' d\Omega_m + \int_{\Omega_m} \boldsymbol{f}\cdot\boldsymbol{u}' d\Omega_m = \int_{\Omega_m} \rho\frac{\partial^2\boldsymbol{u}}{\partial t^2}\cdot\boldsymbol{u}' d\Omega_m \\ \Rightarrow -&\int_{\Omega_m} \nabla\boldsymbol{u}' : \boldsymbol{\sigma} d\Omega_m + \int_{\partial\Omega_m} \boldsymbol{n}\cdot(\boldsymbol{\sigma}\,\boldsymbol{u}')d\partial\Omega_m + \int_{\Omega_m} \boldsymbol{f}\cdot\boldsymbol{u}' d\Omega_m = \int_{\Omega_m} \rho\frac{\partial^2\boldsymbol{u}}{\partial t^2}\cdot\boldsymbol{u}' d\Omega_m,\end{aligned} \tag{2.9}$$

which holds for any appropriate test function $\boldsymbol{u}'$. Domain Ω_m is the solid region in the reference geometry of figure 1.5, $\partial\Omega_m$ is its boundary. Unlike for the electrostatic problem domain Ω_m is not deformed by $\boldsymbol{u}$ because of the linear elasticity approximation: all elasticity calculations are performed on the undeformed geometry. Furthermore

$$\nabla\boldsymbol{u}' = \begin{bmatrix} \partial_x u'_x & \partial_y u'_x & \partial_z u'_x \\ \partial_x u'_y & \partial_y u'_y & \partial_z u'_y \\ \partial_x u'_z & \partial_y u'_z & \partial_z u'_z \end{bmatrix}, \tag{2.10}$$

and : is the Frobenius product such that $\nabla\boldsymbol{u}' : \boldsymbol{\sigma} := \sum_{ij}(\nabla\boldsymbol{u}')_{i,j}\sigma_{i,j}$. Expanding $\nabla\boldsymbol{u}' : \boldsymbol{\sigma}$ and using the symmetry of $\boldsymbol{\sigma}$ gives:

$$\begin{aligned}\nabla\boldsymbol{u}' : \boldsymbol{\sigma} &= \partial_x u'_x\sigma_{xx} + \partial_y u'_x\sigma_{xy} + \partial_z u'_x\sigma_{xz} + \partial_x u'_y\sigma_{xy} + \partial_y u'_y\sigma_{yy} + \partial_z u'_y\sigma_{yz} \\ &\quad + \partial_x u'_z\sigma_{xz} + \partial_y u'_z\sigma_{yz} + \partial_z u'_z\sigma_{zz} \\ &= \partial_x u'_x\sigma_{xx} + \partial_y u'_y\sigma_{yy} + \partial_z u'_z\sigma_{zz} \\ &\quad + (\partial_x u'_y + \partial_y u'_x)\sigma_{xy} + (\partial_z u'_y + \partial_y u'_z)\sigma_{yz} + (\partial_x u'_z + \partial_z u'_x)\sigma_{xz} \\ &= \varepsilon'_{xx}\sigma_{xx} + \varepsilon'_{yy}\sigma_{yy} + \varepsilon'_{zz}\sigma_{zz} + 2\varepsilon'_{xy}\sigma_{xy} + 2\varepsilon'_{yz}\sigma_{yz} + 2\varepsilon'_{xz}\sigma_{xz} \\ &= \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix}^T \begin{bmatrix} \varepsilon'_{xx} \\ \varepsilon'_{yy} \\ \varepsilon'_{zz} \\ 2\varepsilon'_{xy} \\ 2\varepsilon'_{yz} \\ 2\varepsilon'_{xz} \end{bmatrix} \equiv \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix}^T \begin{bmatrix} \varepsilon'_{xx} \\ \varepsilon'_{yy} \\ \varepsilon'_{zz} \\ \gamma'_{xy} \\ \gamma'_{yz} \\ \gamma'_{xz} \end{bmatrix},\end{aligned} \tag{2.11}$$

where the relations $(\partial_x u'_y + \partial_y u'_x) = 2\varepsilon_{xy}$ and the like come from the strain-displacement relation in (2.7). The quantity γ is commonly used in mechanics to represent the shear strain.

Hooke's law for homogeneous isotropic materials is as follows:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \quad \begin{bmatrix} \boldsymbol{C_{11}} & \boldsymbol{C_{12}} \\ \boldsymbol{C_{21}} & \boldsymbol{C_{22}} \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix}, \tag{2.12}$$

where ν is Poisson's ratio, E Young's modulus,

$$\boldsymbol{C}_{11} = \begin{bmatrix} 1-\nu & \nu & \nu \\ \nu & 1-\nu & \nu \\ \nu & \nu & 1-\nu \end{bmatrix}, \quad \boldsymbol{C}_{22} = \begin{bmatrix} \frac{1-2\nu}{2} & 0 & 0 \\ 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}, \tag{2.13}$$

$$\boldsymbol{C}_{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{C}_{21} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.14}$$

Injecting (2.12) in (2.11) and putting everything in (2.9) finally gives the weak formulation for the elastodynamic problem:

$$\begin{aligned}
&-\int_{\Omega_m} \nabla \boldsymbol{u}' : \boldsymbol{\sigma} d\Omega_m + \int_{\partial\Omega_m} \boldsymbol{n} \cdot (\boldsymbol{\sigma u}')\, d\partial\Omega_m + \int_{\Omega_m} \boldsymbol{f} \cdot \boldsymbol{u}' d\Omega_m = \int_{\Omega_m} \rho \, \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{u}' d\Omega_m \\
&\Rightarrow -\int_{\Omega_m} [\begin{bmatrix} \varepsilon'_{xx} \\ \varepsilon'_{yy} \\ \varepsilon'_{zz} \end{bmatrix}^T \boldsymbol{C_{11}} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \end{bmatrix} + \begin{bmatrix} \varepsilon'_{xx} \\ \varepsilon'_{yy} \\ \varepsilon'_{zz} \end{bmatrix}^T \boldsymbol{C_{12}} \begin{bmatrix} \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} + \begin{bmatrix} \gamma'_{xy} \\ \gamma'_{yz} \\ \gamma'_{xz} \end{bmatrix}^T \boldsymbol{C_{21}} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \end{bmatrix} \\
&+ \begin{bmatrix} \gamma'_{xy} \\ \gamma'_{yz} \\ \gamma'_{xz} \end{bmatrix}^T \boldsymbol{C_{22}} \begin{bmatrix} \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix}] d\Omega_m + \int_{\partial\Omega_m} \boldsymbol{n} \cdot (\boldsymbol{\sigma u}')\, d\partial\Omega_m + \int_{\Omega_m} \boldsymbol{f} \cdot \boldsymbol{u}' d\Omega_m = \int_{\Omega_m} \rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{u}' d\Omega_m
\end{aligned} \tag{2.15}$$

or, in compact form,

$$\begin{aligned}
&-\int_{\Omega_m} [\mathcal{M}_n(\boldsymbol{u}')^T \boldsymbol{C_{11}} \mathcal{M}_n(\boldsymbol{u}) + \mathcal{M}_n(\boldsymbol{u}')^T \boldsymbol{C_{12}} \, \mathcal{M}_s(\boldsymbol{u}) + \mathcal{M}_s(\boldsymbol{u}')^T \boldsymbol{C_{21}} \mathcal{M}_n(\boldsymbol{u}) \\
&+ \mathcal{M}_s(\boldsymbol{u}')^T \boldsymbol{C_{22}} \mathcal{M}_s(\boldsymbol{u})] d\Omega_m + \int_{\partial\Omega_m} \boldsymbol{n} \cdot (\boldsymbol{\sigma u}')\, d\partial\Omega_m + \int_{\Omega_m} \boldsymbol{f} \cdot \boldsymbol{u}' d\Omega_m = \int_{\Omega_m} \rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{u}' d\Omega_m.
\end{aligned} \tag{2.16}$$

where

$$\mathcal{M}_s(\boldsymbol{u}) = [\ \tfrac{\partial u_y}{\partial x} + \tfrac{\partial u_x}{\partial y} \quad \tfrac{\partial u_y}{\partial z} + \tfrac{\partial u_z}{\partial y} \quad \tfrac{\partial u_z}{\partial x} + \tfrac{\partial u_x}{\partial z} \ ]^T, \tag{2.17}$$

is the 3D shear strain operator and

$$\mathcal{M}_n(\boldsymbol{u}) = [\ \tfrac{\partial u_x}{\partial x} \quad \tfrac{\partial u_y}{\partial y} \quad \tfrac{\partial u_z}{\partial z}\ ]^T, \tag{2.18}$$

is the 3D normal strain operator. The external force term consists of pressure forces applied by the fluid as well as the nonlinear electrostatic force derived in section 2.2.3 using the virtual work principle. It is worth noting that (2.16) is not the only form to represent elastodynamic problems: it could alternatively be written using e.g. Lamé coefficients [76].

2.2.3 Electrostatic force

Electrostatic forces appear in an uncountable number of MEMS applications. Several methods have been developed to compute them [77, 78, 79]. The most commonly used are the Maxwell stress tensor and the virtual work principle [31, 32, 33, 34, 3, 35, 80].

The principle of virtual work states that a system is at equilibrium if and only if any tiny perturbation in the external energy of that system is perfectly balanced by the variation of the internal energy. Mathematically:

$$\delta W_{\text{int}} = \delta W_{\text{ext}} \iff \text{equilibrium.} \tag{2.19}$$

The external work W_{ext} originates from external mechanical forces and electric potentials applied to the structure. The internal work W_{int} takes into account all work done inside the system in reaction to the external forces. It is proportional to the strains and stresses inside the solid. The electric potential is also a player in the internal energy as the accumulated internal energy increases when the electric field increases.

Based on Gibbs energy as done in [3] the internal work can be rewritten:

$$\delta W_{\text{int}} = \delta W_m - \delta W_e, \tag{2.20}$$

where W_m is the mechanical work and W_e the electrical work.

Perturbing the internal energy by a compatible virtual displacement yields the mechanical forces and perturbing the internal energy by a compatible electric potential perturbation yields the electric charge:

$$\begin{aligned} \boldsymbol{f_m} \cdot \boldsymbol{\delta u} &= W_{\text{int}}(\boldsymbol{\delta u}) \equiv \delta_u W_{\text{int}} = \delta_u W_m - \delta_u W_e, \\ q_e\, \delta v &= W_{\text{int}}(\delta v) \equiv \delta_v W_{\text{int}} = \cancelto{0}{\delta_v W_m} - \delta_v W_e, \end{aligned} \tag{2.21}$$

where the mechanical energy does not change when the electric potential is perturbed. Quantity $\boldsymbol{f_m}$ includes all forces applied to the system. That includes usual external loads as well as the electrostatic forces. In what follows only the latter are of interest.

The electrostatic forces appear only because there is a change in electrical energy when perturbing the system with a virtual displacement. If there is no change in electrical energy there can be no electrostatic force. This leads to the conclusion that

$$\boldsymbol{f}_{\text{electrostatic}} \cdot \boldsymbol{\delta u} = -\delta_u W_e. \tag{2.22}$$

In order to compute the electrostatic force the working domain Ω^* is perturbed by a compatible virtual displacement $\boldsymbol{\delta u}$. Let (x, y, z) be the coordinates in the unperturbed Ω^* region and (ξ, η, ϕ) the coordinates in the perturbed region Ω^{p*}. Equation (2.22) gives:

$$\int_{\Omega^*} \boldsymbol{f}_{\text{electrostatic}} \cdot \boldsymbol{\delta u}\, d\Omega^* = -(W_e^p - W_e), \tag{2.23}$$

where W_e and W_e^p are the unperturbed and perturbed electrical energy respectively:

$$W_e = \frac{1}{2}\int_{\Omega^*} \epsilon \nabla_{x,y,z} v \cdot \nabla_{x,y,z} v\, d\Omega^*, \ W_e^p = \frac{1}{2}\int_{\Omega^{p*}} \epsilon \nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi} v\, d\Omega^{p*}. \tag{2.24}$$

It is worth noting that in (2.23), integrating on the mechanical domain Ω_m would give the same result since the electrostatic force is zero everywhere but on Ω_m. In (2.24) however the energy can not be integrated only on Ω_m: it must be integrated on Ω^*.

The perturbed electrical energy is brought back on the unperturbed region Ω^* using the following coordinate change:

$$\begin{cases} \xi &= x + \delta u_x, \\ \eta &= y + \delta u_y, \\ \phi &= z + \delta u_z, \end{cases} \tag{2.25}$$

whose Jacobian matrix is:

$$\boldsymbol{J} = \frac{\partial(\xi, \eta, \phi)}{\partial(x, y, z)} = \begin{bmatrix} 1 + \delta_x u_x & \delta_x u_y & \delta_x u_z \\ \delta_y u_x & 1 + \delta_y u_y & \delta_y u_z \\ \delta_z u_x & \delta_z u_y & 1 + \delta_z u_z \end{bmatrix}, \tag{2.26}$$

with $\delta_x u_x$ standing for $\frac{\partial \delta u_x}{\partial x}$.

The Jacobian determinant $|\boldsymbol{J}|$ and its inverse matrix are needed. Since $\boldsymbol{\delta u}$ can be taken arbitrarily small only the first order $\boldsymbol{\delta u}$ terms must be considered and $|\boldsymbol{J}|$ has a simple form:

$$|\boldsymbol{J}| \approx 1 + \delta_x u_x + \delta_y u_y + \delta_z u_z + O(\delta \boldsymbol{u}^2), \tag{2.27}$$

and

$$\boldsymbol{J}^{-1} \approx \frac{1}{|\boldsymbol{J}|} \begin{bmatrix} 1 + \delta_y u_y + \delta_z u_z & -\delta_x u_y & -\delta_x u_z \\ -\delta_y u_x & 1 + \delta_x u_x + \delta_z u_z & -\delta_y u_z \\ -\delta_z u_x & -\delta_z u_y & 1 + \delta_x u_x + \delta_y u_y \end{bmatrix}.$$

The inverse Jacobian determinant, in the form $\frac{1}{1+x}$ can be approximated to the first order by $1 - x$ which leads to a simple form:

$$\boldsymbol{J}^{-1} \approx \begin{bmatrix} 1 - \delta_x u_x & -\delta_x u_y & -\delta_x u_z \\ -\delta_y u_x & 1 - \delta_y u_y & -\delta_y u_z \\ -\delta_z u_x & -\delta_z u_y & 1 - \delta_z u_z \end{bmatrix}. \tag{2.28}$$

One can take advantage of the particular form of the inverse Jacobian matrix and rewrite it in the simpler form $\boldsymbol{J}^{-1} \approx \boldsymbol{I} - \begin{bmatrix} \nabla\delta u_x & \nabla\delta u_y & \nabla\delta u_z \end{bmatrix}$, where $\boldsymbol{I}$ is 3 by 3 identity matrix and ∇ the gradient operator giving a column vector.

Using this simple form and dropping the subscripts on the undeformed gradient operator ($\nabla_{x,y,z} v \equiv \nabla v$) one gets:

$$\nabla_{\xi,\eta,\phi} v = \boldsymbol{J}^{-1} \nabla v = \nabla v - \nabla\delta u_x \frac{\partial v}{\partial x} - \nabla\delta u_y \frac{\partial v}{\partial y} - \nabla\delta u_z \frac{\partial v}{\partial z}. \tag{2.29}$$

Injecting (2.29) in (2.24) and multiplying by $|\boldsymbol{J}|$ for the coordinate change leads to:

$$\begin{aligned} W_e^p =& \frac{1}{2}\int_{\Omega^*} \epsilon \left[\nabla v \cdot \nabla v - 2\nabla v \cdot \nabla\delta u_x \frac{\partial v}{\partial x} - 2\nabla v \cdot \nabla\delta u_y \frac{\partial v}{\partial y} - 2\nabla v \cdot \nabla\delta u_z \frac{\partial v}{\partial z}\right] \\ & \cdot \left(1 + \delta_x u_x + \delta_y u_y + \delta_z u_z\right) d\Omega^*, \end{aligned}$$

which transforms (2.23) into:

$$\begin{aligned} \int_{\Omega^*} \boldsymbol{f}_{\text{electrostatic}} \cdot \boldsymbol{\delta u}\, d\Omega^* &= -(W_e^p - W_e) \\ &= -\frac{1}{2}\int_{\Omega^*} \epsilon \left[\nabla v \cdot \nabla v \left(\delta_x u_x + \delta_y u_y + \delta_z u_z\right) - 2\nabla v \cdot \nabla\delta u_x \frac{\partial v}{\partial x}\right. \\ &\quad \left. - 2\nabla v \cdot \nabla\delta u_y \frac{\partial v}{\partial y} - 2\nabla v \cdot \nabla\delta u_z \frac{\partial v}{\partial z}\right] d\Omega^*, \end{aligned} \tag{2.30}$$

neglecting again all higher order perturbations.

In order to remove $\boldsymbol{\delta u}$ from equation (2.30) let us extract the $\boldsymbol{\delta u}$ perturbation's gradient from the electrical energy difference. For that we first notice that

$$\nabla v\, \delta_x u_x = \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 \\ \frac{\partial v}{\partial y} & 0 & 0 \\ \frac{\partial v}{\partial z} & 0 & 0 \end{bmatrix} \nabla\delta u_x, \tag{2.31}$$

$$\nabla v\, \delta_y u_y = \begin{bmatrix} 0 & \frac{\partial v}{\partial x} & 0 \\ 0 & \frac{\partial v}{\partial y} & 0 \\ 0 & \frac{\partial v}{\partial z} & 0 \end{bmatrix} \nabla\delta u_y, \tag{2.32}$$

$$\nabla v\, \delta_z u_z = \begin{bmatrix} 0 & 0 & \frac{\partial v}{\partial x} \\ 0 & 0 & \frac{\partial v}{\partial y} \\ 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \nabla\delta u_z, \tag{2.33}$$

$$\nabla\delta u_x \frac{\partial v}{\partial x} = \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 \\ 0 & \frac{\partial v}{\partial x} & 0 \\ 0 & 0 & \frac{\partial v}{\partial x} \end{bmatrix} \nabla\delta u_x. \tag{2.34}$$

Equation (2.34) is not displayed for y and z because it has exactly the same structure. Using in (2.30) the previous relations one gets:

$$\begin{aligned}
&\int_{\Omega^*} \boldsymbol{f}_{\text{electrostatic}} \cdot \boldsymbol{\delta u}\, d\Omega^* = \\
&-\frac{1}{2}\int_{\Omega^*} \epsilon\, [(\nabla v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 & 0 & \frac{\partial v}{\partial x} & 0 & 0 & 0 & \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} & 0 & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & 0 & \frac{\partial v}{\partial y} \\ \frac{\partial v}{\partial z} & 0 & 0 & 0 & \frac{\partial v}{\partial z} & 0 & 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \begin{bmatrix} \nabla \delta u_x \\ \nabla \delta u_y \\ \nabla \delta u_z \end{bmatrix} \\
&-2\,(\nabla v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} & 0 & 0 \\ 0 & \frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} & 0 \\ 0 & 0 & \frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \begin{bmatrix} \nabla \delta u_x \\ \nabla \delta u_y \\ \nabla \delta u_z \end{bmatrix}] d\Omega^*,
\end{aligned} \tag{2.35}$$

which is identical to

$$\begin{aligned}
&\int_{\Omega^*} \boldsymbol{f}_{\text{electrostatic}} \cdot \boldsymbol{\delta u}\, d\Omega^* = \\
&\frac{1}{2}\int_{\Omega^*} \epsilon\, (\nabla v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 & 2\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial x} & 0 & 2\frac{\partial v}{\partial z} & 0 & -\frac{\partial v}{\partial x} \\ -\frac{\partial v}{\partial y} & 2\frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & 2\frac{\partial v}{\partial z} & -\frac{\partial v}{\partial y} \\ -\frac{\partial v}{\partial z} & 0 & 2\frac{\partial v}{\partial x} & 0 & -\frac{\partial v}{\partial z} & 2\frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \begin{bmatrix} \nabla \delta u_x \\ \nabla \delta u_y \\ \nabla \delta u_z \end{bmatrix} d\Omega^*,
\end{aligned} \tag{2.36}$$

the continuous weak formulation of the electrostatic force. It holds for any appropriate virtual displacement $\boldsymbol{\delta u}$.

2.2.4 Linear acoustics

Let δp be the acoustic pressure variation around the average pressure defined on domain Ω_a (the fluid domain on the reference CMUT of figure 1.5). The acoustic wave equation (1.17) is

$$\frac{1}{c^2}\frac{\partial^2 \delta p}{\partial t^2} - \Delta \delta p = 0. \tag{2.37}$$

Its weak form is derived in a way similar to the electrostatic weak form:

$$-\int_{\Omega_a} \nabla \delta p \cdot \nabla \delta p' d\Omega_a + \int_{\partial\Omega_a} \delta p'\, \partial_{\boldsymbol{n}}\, \delta p\, d\partial\Omega_a - \int_{\Omega_a} \frac{1}{c^2}\frac{\partial^2 \delta p}{\partial t^2} \delta p' d\Omega_a = 0, \tag{2.38}$$

where $\partial\Omega_a$ is the boundary of Ω_a and $\partial_{\boldsymbol{n}}\delta p = \nabla \delta p \cdot \boldsymbol{n}$ is the normal derivative outward to the fluid region. Taking into account the mechanical vibration as a pressure source is done with Newton's law [38] as introduced in (1.22):

$$\partial_{\boldsymbol{n}}\delta p = -\rho_{\text{fluid}}\frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{n}. \tag{2.39}$$

Because the fluid region is infinite in theory it has to be truncated for the finite element method and appropriate boundary conditions have to be imposed to avoid any wave reflection at the non-physical truncation surface. A classical method is to use Sommerfeld's radiation condition [81, 82]:

$$\partial_{\boldsymbol{n}}\delta p + \frac{1}{c}\frac{\partial \delta p}{\partial t} = 0. \tag{2.40}$$

The Sommerfeld condition (2.40) forces outgoing pressure waves at infinity: a pressure field of the form

$$\delta p(r,t) = \delta P \, \cos(\omega t - kr), \tag{2.41}$$

for a pressure value δP Pa, a pulsation ω rad/s, a wavenumber $k = \frac{\omega}{c}$ rad/m and a wave propagation direction $\boldsymbol{e}_r$ perpendicular to the truncation surface corresponds to an outgoing wave and indeed satisfies the Sommerfeld relation:

$$\partial_{\boldsymbol{n}} \delta p = \partial_{\boldsymbol{e}_r} \delta p = k \, \delta P \, \sin(\omega t - kr) = \frac{\omega}{c} \, \delta P \, \sin(\omega t - kr) = -\frac{1}{c} \frac{\partial \delta P \, \cos(\omega t - kr)}{\partial t}. \tag{2.42}$$

In practice outgoing waves are never exactly achieved in 2D or 3D but when far enough from the acoustic source the relative decrease of δP is so slow in the normal direction that reflexion is negligeable. A spherical truncation surface can be advantageously chosen to have waves propagating normally to the surface. Alternatively higher order absorbing boundary conditions [83, 84] or perfectly matched layers (PML) [85, 86] can be used to deal with the truncation.

2.2.5 Sensitivity matrix

Solving the nonlinear coupled electroelastoacoustic problem (2.6)-(2.16)-(2.38) with the techniques exposed in section 2.4 requires to compute a sensitivity matrix $\boldsymbol{S}$ telling how the mechanical energy, the electrical energy, the acoustic energy and the electrostatic forces change given a change in the electric potential v, mechanical displacement $\boldsymbol{u}$ and acoustic pressure δp. The method used below to derive it was proposed in [3].

Rewriting equation (2.21) gives:

$$\begin{cases} \boldsymbol{f}_m & = \frac{\partial W_m}{\partial \boldsymbol{u}} - \frac{\partial W_e}{\partial \boldsymbol{u}}, \\ q_e & = -\frac{\partial W_e}{\partial v}. \end{cases} \tag{2.43}$$

Linearising both relations at equilibrium leads to:

$$\begin{cases} \boldsymbol{f}_m \approx & \boldsymbol{f}_m^e & +(\dfrac{\partial^2 W_m}{\partial \boldsymbol{u}^2} - \dfrac{\partial^2 W_e}{\partial \boldsymbol{u}^2}) \, \boldsymbol{du} - \dfrac{\partial^2 W_e}{\partial v \partial \boldsymbol{u}} dv, \\ q_e \approx & q_e^e & -\dfrac{\partial^2 W_e}{\partial \boldsymbol{u} \partial v} \, \boldsymbol{du} - \dfrac{\partial^2 W_e}{\partial v^2} dv, \end{cases} \tag{2.44}$$

since the mechanical energy does not depend on the electric potential. This can be written in matrix form:

$$\begin{bmatrix} \boldsymbol{S}_{\boldsymbol{uu}} & \boldsymbol{S}_{\boldsymbol{u}v} \\ \boldsymbol{S}_{v\boldsymbol{u}} & \boldsymbol{S}_{vv} \end{bmatrix} \begin{bmatrix} \boldsymbol{du} \\ dv \end{bmatrix} = \begin{bmatrix} \boldsymbol{df_m} \\ dq_e \end{bmatrix}, \tag{2.45}$$

in which $\boldsymbol{S}$ can be identified as the sensitivity matrix of the electroelastic problem. Matrix $\boldsymbol{S}$ can be decomposed as follows:

$$\boldsymbol{S} = \boldsymbol{A} + \begin{bmatrix} -\frac{\partial^2 W_e}{\partial^2 \boldsymbol{u}} & -\frac{\partial^2 W_e}{\partial v \partial \boldsymbol{u}} \\ -\frac{\partial^2 W_e}{\partial \boldsymbol{u} \partial v} & 0 \end{bmatrix} := \boldsymbol{A} + \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{uu}} & \boldsymbol{K}_{\boldsymbol{u}v} \\ \boldsymbol{K}_{v\boldsymbol{u}} & \boldsymbol{0} \end{bmatrix}, \tag{2.46}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial W_m^2}{\partial \boldsymbol{u}^2} & 0 \\ 0 & \frac{\partial^2 W_e}{\partial v^2} \end{bmatrix}, \ \boldsymbol{K}_{\boldsymbol{uu}} = \begin{bmatrix} \boldsymbol{K}_{u_x u_x} & \boldsymbol{K}_{u_x u_y} & \boldsymbol{K}_{u_x u_z} \\ \boldsymbol{K}_{u_y u_x} & \boldsymbol{K}_{u_y u_y} & \boldsymbol{K}_{u_y u_z} \\ \boldsymbol{K}_{u_z u_x} & \boldsymbol{K}_{u_z u_y} & \boldsymbol{K}_{u_z u_z} \end{bmatrix}, \tag{2.47}$$

$$\boldsymbol{K}_{\boldsymbol{u}v} = \begin{bmatrix} \boldsymbol{K}_{u_x v} & \boldsymbol{K}_{u_y v} & \boldsymbol{K}_{u_z v} \end{bmatrix}^T, \ \boldsymbol{K}_{v\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{K}_{v u_x} & \boldsymbol{K}_{v u_y} & \boldsymbol{K}_{v u_z} \end{bmatrix}, \tag{2.48}$$

with $\frac{\partial W_m^2}{\partial \boldsymbol{u}^2}$ and $\frac{\partial^2 W_e}{\partial v^2}$ the usual uncoupled elastodynamic (2.16) and electrostatic (2.6) systems. The sensitivity matrix for the electroelastic problem is thus simply the sum of the uncoupled elastodynamic and electrostatic terms and 3 extra terms whose 3D expression is computed below. Because the acoustic formulation (2.38) and its coupling with the two other physics is linear the sensitivity matrix for the electroelastoacoustic problem is simply obtained by including the terms of the uncoupled acoustic formulation (2.38) and the corresponding coupling terms in matrix $\boldsymbol{A}$.

$\boldsymbol{K}_{\boldsymbol{u}v}$ matrix block:

Let us first calculate the expression of the $\boldsymbol{K}_{\boldsymbol{u}v}$ block, which corresponds to $\frac{\partial \boldsymbol{f}_{\text{elec}}}{\partial v}$. For that recall (2.30):

$$\begin{aligned} \int_{\Omega^*} \boldsymbol{f}_{\text{elec}} \cdot \boldsymbol{\delta u} \, d\Omega^* = -\frac{1}{2} \int_{\Omega^*} & \epsilon[\nabla v \cdot \nabla v \, (\delta_x u_x + \delta_y u_y + \delta_z u_z) - 2\nabla v \cdot \nabla \delta u_x \frac{\partial v}{\partial x} \\ & - 2\nabla v \cdot \nabla \delta u_y \frac{\partial v}{\partial y} - 2\nabla v \cdot \nabla \delta u_z \frac{\partial v}{\partial z}] d\Omega^*. \end{aligned} \tag{2.49}$$

Perturbing it by a compatible perturbation δv while keeping only linear perturbations of δv gives:

$$\begin{aligned} \int_{\Omega^*} \delta v \, \boldsymbol{K}_{\boldsymbol{u}v}^T \boldsymbol{\delta u} \, d\Omega^* = -\frac{1}{2} \int_{\Omega^*} & \epsilon \, [2\nabla \delta v \cdot \nabla v \, (\delta_x u_x + \delta_y u_y + \delta_z u_z) \\ & - 2\nabla \delta v \cdot \nabla \delta u_x \, \frac{\partial v}{\partial x} - 2\nabla v \cdot \nabla \delta u_x \delta_x v \\ & - 2\nabla v \cdot \nabla \delta u_y \, \frac{\partial v}{\partial y} - 2\nabla v \cdot \nabla \delta u_y \delta_y v \\ & - 2\nabla v \cdot \nabla \delta u_z \, \frac{\partial v}{\partial z} - 2\nabla v \cdot \nabla \delta u_z \delta_z v] d\Omega^*, \end{aligned} \tag{2.50}$$

where $\boldsymbol{K}_{\boldsymbol{u}v}$ is a 3 by 1 column vector. Considering equations (2.31) through (2.34) and the following variant thereof:

$$(\nabla v)^T \, \delta_x v = (\nabla \delta v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.51}$$

one can rewrite (2.50) as:

$$
\begin{aligned}
&\int_{\Omega^*} \delta v\, \boldsymbol{K}_{\boldsymbol{u}v}^T \boldsymbol{\delta u}\, d\Omega^* = \\
&- \int_{\Omega^*} \epsilon\, [(\nabla \delta v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 & 0 & \frac{\partial v}{\partial x} & 0 & 0 & 0 & \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} & 0 & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & 0 & \frac{\partial v}{\partial y} \\ \frac{\partial v}{\partial z} & 0 & 0 & 0 & \frac{\partial v}{\partial z} & 0 & 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \begin{bmatrix} \nabla \delta u_x \\ \nabla \delta u_y \\ \nabla \delta u_z \end{bmatrix} \\
&- (\nabla \delta v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} & 0 & 0 \\ 0 & \frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} & 0 \\ 0 & 0 & \frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \begin{bmatrix} \nabla \delta u_x \\ \nabla \delta u_y \\ \nabla \delta u_z \end{bmatrix} \qquad (2.52) \\
&- (\nabla \delta v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{bmatrix} \begin{bmatrix} \nabla \delta u_x \\ \nabla \delta u_y \\ \nabla \delta u_z \end{bmatrix}] d\Omega^*,
\end{aligned}
$$

which is identical to

$$
\begin{aligned}
&\int_{\Omega^*} \delta v\, \boldsymbol{K}_{\boldsymbol{u}v}^T \boldsymbol{\delta u}\, d\Omega^* = - \int_{\Omega^*} \epsilon\, (\nabla \delta v)^T \\
&\cdot \begin{bmatrix} -\frac{\partial v}{\partial x} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial z} & -\frac{\partial v}{\partial y} & \frac{\partial v}{\partial x} & 0 & -\frac{\partial v}{\partial z} & 0 & \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} & -\frac{\partial v}{\partial x} & 0 & -\frac{\partial v}{\partial x} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial z} & 0 & -\frac{\partial v}{\partial z} & \frac{\partial v}{\partial y} \\ \frac{\partial v}{\partial z} & 0 & -\frac{\partial v}{\partial x} & 0 & \frac{\partial v}{\partial z} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial x} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial z} \end{bmatrix} \begin{bmatrix} \nabla \delta u_x \\ \nabla \delta u_y \\ \nabla \delta u_z \end{bmatrix} d\Omega^*. \qquad (2.53)
\end{aligned}
$$

$\boldsymbol{K}_{v\boldsymbol{u}}$ matrix block:

The $\boldsymbol{K}_{v\boldsymbol{u}}$ block corresponds to $-\frac{\partial^2 W_e}{\partial \boldsymbol{u} \partial v}$. It is equal to the transpose of $\boldsymbol{K}_{\boldsymbol{u}v}$: $\boldsymbol{K}_{v\boldsymbol{u}} = \boldsymbol{K}_{\boldsymbol{u}v}^T$.

$\boldsymbol{K}_{\boldsymbol{uu}}$ matrix block:

The matrix block $\boldsymbol{K}_{\boldsymbol{uu}} = -\frac{\partial^2 W_e}{\partial^2 \boldsymbol{u}} = -\frac{\partial \boldsymbol{f}_{\text{elec}}}{\partial \boldsymbol{u}}$ is now derived. For that recall (2.30) with pertubation $\boldsymbol{\delta u}$ renamed as $\boldsymbol{\delta u^1}$ as there will be two perturbations here:

$$
\begin{aligned}
\int_{\Omega^*} \boldsymbol{f}_{\text{elec}} \cdot \boldsymbol{\delta u^1}\, d\Omega = -\frac{1}{2} \int_{\Omega^*} & \epsilon\, [\nabla v \cdot \nabla v\, (\delta_x u_x^1 + \delta_y u_y^1 + \delta_z u_z^1) \\
& - 2 \nabla v \cdot \nabla \delta u_x^1 \, \frac{\partial v}{\partial x} - 2 \nabla v \cdot \nabla \delta u_y^1 \, \frac{\partial v}{\partial y} \qquad (2.54) \\
& - 2 \nabla v \cdot \nabla \delta u_z^1 \, \frac{\partial v}{\partial z}]\, d\Omega^*.
\end{aligned}
$$

The term $\frac{\partial \boldsymbol{f}_{\text{elec}}}{\partial \boldsymbol{u}}$ is obtained by taking the difference of the electrostatic force computed on the original subdomain Ω described by (x, y, z) and the original subdomain pertubed by an infinitesimal perturbation $\boldsymbol{\delta u^2}$, called Ω^p (which is different from the Ω^p used before) and described by (ξ, η, ϕ). Unlike for $\boldsymbol{K}_{v\boldsymbol{u}}$, $\boldsymbol{K}_{\boldsymbol{uu}}$ can not be obtained by simply perturbing (2.54) since the perturbation of $\boldsymbol{\delta u^2}$ on v is not straightforward. The electrostatic force on the perturbed domain Ω^p is given by:

$$
\begin{aligned}
\int_{\Omega^{p*}} \boldsymbol{f}_{\text{elec}} \cdot \boldsymbol{\delta u^1}\, d\Omega^{p*} = -\frac{1}{2} \int_{\Omega^{p*}} \epsilon\, [&\nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi}\, v\, (\delta_\xi u_x^1 + \delta_\eta u_y^1 + \delta_\phi u_z^1) \\
&- 2\nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi} \delta u_x^1 \, \frac{\partial v}{\partial \xi} - 2\nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi} \delta u_y^1 \, \frac{\partial v}{\partial \eta} \\
&- 2\nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi} \delta u_z^1 \, \frac{\partial v}{\partial \xi}] d\Omega^{p*}.
\end{aligned}
\tag{2.55}
$$

In order to bring this equation back to the unperturbed domain Ω we introduce in a similar way to (2.25) the coordinate change:

$$
\left\{
\begin{array}{rcl}
\xi & = & x + \delta u_x^2, \\
\eta & = & y + \delta u_y^2, \\
\phi & = & z + \delta u_z^2,
\end{array}
\right.
\tag{2.56}
$$

whose inverse Jacobian matrix and Jacobian determinant are obtained in the same way:

$$
\boldsymbol{J}^{-1} = \frac{\partial(x,y,z)}{\partial(\xi,\eta,\phi)} \approx \left[\begin{array}{ccc} 1 - \delta_x u_x^2 & -\delta_x u_y^2 & -\delta_x u_z^2 \\ -\delta_y u_x^2 & 1 - \delta_y u_y^2 & -\delta_y u_z^2 \\ -\delta_z u_x^2 & -\delta_z u_y^2 & 1 - \delta_z u_z^2 \end{array} \right],
\tag{2.57}
$$

$$
|\boldsymbol{J}| \approx 1 + \delta_x u_x^2 + \delta_y u_y^2 + \delta_z u_z^2,
\tag{2.58}
$$

with $\delta_x u_x^2$ standing for $\frac{\partial \delta u_x^2}{\partial x}$. Now all terms in (2.55) need to be turned into computations on the undeformed mesh described by (x, y, z):

- $\delta_\xi u_x^1 + \delta_\eta u_y^1 + \delta_\phi u_z^1$: since δu_x^1 and the like are functions of (x, y, z) one has $\delta_\xi u_x^1 \equiv \frac{\partial \delta u_x^1}{\partial \xi} = \delta_x u_x^1 \frac{\partial x}{\partial \xi} + \delta_y u_x^1 \frac{\partial y}{\partial \xi} + \delta_z u_x^1 \frac{\partial z}{\partial \xi}$ which using $\boldsymbol{J}^{-1}$ enables to compute

$$
\begin{aligned}
\delta_\xi u_x^1 + \delta_\eta u_y^1 + \delta_\phi u_z^1 = (1 - \delta_x u_x^2)\delta_x u_x^1 &- \delta_x u_y^2\, \delta_y u_x^1 - \delta_x u_z^2\, \delta_z u_x^1 \\
- \delta_y u_x^2\, \delta_x u_y^1 &+ (1 - \delta_y u_y^2)\, \delta_y u_y^1 - \delta_y u_z^2\, \delta_z u_y^1 \\
- \delta_z u_x^2\, \delta_x u_z^1 &- \delta_z u_y^2\, \delta_y u_z^1 + (1 - \delta_z u_z^2)\, \delta_z u_z^1,
\end{aligned}
\tag{2.59}
$$

- $\frac{\partial v}{\partial \xi}$, $\frac{\partial v}{\partial \eta}$, $\frac{\partial v}{\partial \phi}$: similarly to the above term we get

$$
\begin{array}{rcl}
\dfrac{\partial v}{\partial \xi} & = & (1 - \delta_x u_x^2)\, \dfrac{\partial v}{\partial x} - \delta_x u_y^2\, \dfrac{\partial v}{\partial y} - \delta_x u_z^2\, \dfrac{\partial v}{\partial z}, \\
\dfrac{\partial v}{\partial \eta} & = & -\delta_y u_x^2\, \dfrac{\partial v}{\partial x} + (1 - \delta_y u_y^2)\, \dfrac{\partial v}{\partial y} - \delta_y u_z^2\, \dfrac{\partial v}{\partial z}, \\
\dfrac{\partial v}{\partial \xi} & = & -\delta_z u_x^2\, \dfrac{\partial v}{\partial x} - \delta_z u_y^2\, \dfrac{\partial v}{\partial y} + (1 - \delta_z u_z^2)\, \dfrac{\partial v}{\partial z},
\end{array}
\tag{2.60}
$$

- $\nabla_{\xi,\eta,\phi} \delta u_x^1$ in the same way as (2.29)

$$
\begin{aligned}
\nabla_{\xi,\eta,\phi} \delta u_x^1 = \boldsymbol{J}^{-1} \nabla \delta u_x^1 = \nabla \delta u_x^1 &- \nabla \delta u_x^2\, \frac{\partial \delta u_x^1}{\partial x} \\
- \nabla \delta u_y^2\, \frac{\partial \delta u_x^1}{\partial y} &- \nabla \delta u_z^2\, \frac{\partial \delta u_x^1}{\partial z},
\end{aligned}
\tag{2.61}
$$

- $\nabla_{\xi,\eta,\phi} v$ similarly

$$\begin{aligned} \nabla_{\xi,\eta,\phi} v = \boldsymbol{J}^{-1} \nabla v = \nabla v - \nabla \delta u_x^2 \, \frac{\partial v}{\partial x} \\ - \nabla \delta u_y^2 \, \frac{\partial v}{\partial y} - \nabla \delta u_z^2 \, \frac{\partial v}{\partial z}. \end{aligned} \tag{2.62}$$

Using the above relations one can replace the four terms in (2.55) by their expression on the unperturbed region Ω, not forgetting to multiply everything by $|\boldsymbol{J}|$ to change the integration domain:

- $\nabla^T_{\xi,\eta,\phi} v \nabla_{\xi,\eta,\phi} v \, (\delta_\xi u_x^1 + \delta_\eta u_y^1 + \delta_\phi u_z^1) \, |\boldsymbol{J}|$:

$$\begin{aligned} \nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi} v \, (\delta_\xi u_x^1 + \delta_\eta u_y^1 + \delta_\phi u_z^1) \, |\boldsymbol{J}| = {} & [\nabla v - \nabla \delta u_x^2 \, \frac{\partial v}{\partial x} - \nabla \delta u_y^2 \, \frac{\partial v}{\partial y} - \nabla \delta u_z^2 \, \frac{\partial v}{\partial z}] \\ & \cdot [\nabla v - \nabla \delta u_x^2 \, \frac{\partial v}{\partial x} - \nabla \delta u_y^2 \, \frac{\partial v}{\partial y} - \nabla \delta u_z^2 \, \frac{\partial v}{\partial z}] \\ & \cdot [(1 - \delta_x u_x^2) \, \delta_x u_x^1 - \delta_x u_y^2 \, \delta_y u_x^1 - \delta_x u_z^2 \, \delta_z u_x^1 \\ & - \delta_y u_x^2 \, \delta_x u_y^1 + (1 - \delta_y u_y^2) \, \delta_y u_y^1 - \delta_y u_z^2 \, \delta_z u_y^1 \\ & - \delta_z u_x^2 \, \delta_x u_z^1 - \delta_z u_y^2 \, \delta_y u_z^1 + (1 - \delta_z u_z^2) \, \delta_z u_z^1] \\ & \cdot (1 + \delta_x u_x^2 + \delta_y u_y^2 + \delta_z u_z^2), \end{aligned} \tag{2.63}$$

where all nonlinear perturbations can be neglected (but not the first and second perturbation product as this would result in a trivial $0 = 0$ equation). The expanded equation is straightforward and is not displayed here.

- $-2 \nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi} \delta u_x^1 \, \frac{\partial v}{\partial \xi} \, |\boldsymbol{J}|$:

$$\begin{aligned} -2 \, \nabla_{\xi,\eta,\phi} v \cdot \nabla_{\xi,\eta,\phi} \delta u_x^1 \, \frac{\partial v}{\partial \xi} \, |\boldsymbol{J}| = {} & -2 \, [\nabla v - \nabla \delta u_x^2 \, \frac{\partial v}{\partial x} - \nabla \delta u_y^2 \, \frac{\partial v}{\partial y} - \nabla \delta u_z^2 \, \frac{\partial v}{\partial z}] \\ & \cdot [\nabla \delta u_x^1 - \nabla \delta u_x^2 \, \delta_x u_x^1 - \nabla \delta u_y^2 \, \delta_y u_x^1 - \nabla \delta u_z^2 \, \delta_z u_x^1 \\ & \cdot [(1 - \delta_x u_x^2) \, \frac{\partial v}{\partial x} - \delta_x u_y^2 \, \frac{\partial v}{\partial y} - \delta_x u_z^2 \, \frac{\partial v}{\partial z}] \\ & \cdot (1 + \delta_x u_x^2 + \delta_y u_y^2 + \delta_z u_z^2). \end{aligned} \tag{2.64}$$

The last three terms are derived in the same way as the latter one.

Finally, to compute $\boldsymbol{K_{uu}}$ the difference between the force computed on the unpertubed domain Ω and on the domain Ω^p perturbed by $\boldsymbol{\delta u^2}$ has to be taken, as done for the electrostatic force computation. The final formulation is not displayed due to its size.

2.3 Spatial discretisation

In section 2.2 all required weak formulations have been derived at the continuous level. What is eventually solved in the finite element method however is an algebraic matrix obtained by discretising the weak formulation both in space and in time. The space discretised weak formulations are presented in this section while the time discretisation is adressed in chapter 3.

2.3.1 Electrostatics

The weak formulation of the electrostatic problem was obtained in (2.6):

$$-\int_{\Omega_e^*} \epsilon \nabla v \cdot \nabla v' d\Omega_e^* + \int_{\partial\Omega_e^*} v' \, \epsilon \, \partial_{\boldsymbol{n}} v \, d\partial\Omega_e^* = \int_{\Omega_e^*} - \rho_v \, v' \, d\Omega_e^*, \tag{2.65}$$

Discretising it requires first to discretise the domain Ω_e^* into a set of discrete elements approximating the geometry. The set of discrete elements obtained after this step is called the finite element *mesh*. It is obtained with meshing softwares as e.g. GMSH [87]. The second step is to discretise the unknown field v (the electric potential field) as follows:

$$v = \sum_{i=1}^{m} c_i \, N_i(\boldsymbol{x}) = \boldsymbol{N}^T \boldsymbol{c}, \tag{2.66}$$

i.e. as a sum of products of space and time independent coefficients c_i and time independent but space dependant shape functions $N_i(\boldsymbol{x})$. Because v' in (2.65) can be any appropriate function one can simply use $v'_i = N_i$ $(i = 1...m)$ so as to get m equations for the m unknown coefficients c_i. This leads to an algebraic system of the form $\boldsymbol{K} \, \boldsymbol{c} = \boldsymbol{b}$:

$$\begin{bmatrix} K_{11} & \cdots & K_{1m} \\ \vdots & \ddots & \vdots \\ K_{m1} & \cdots & K_{mm} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{2.67}$$

A well known example of an appropriate order one shape function $N_i(x)$ is a piecewise linear function equal to one on mesh node i and zero on all others. Figure 2.1 illustrates such a shape function on a 1D mesh. As can be seen the shape function is non zero on only two mesh elements so that matrix $\boldsymbol{K}$ in (2.67) has only few non-zero entries: it is *sparse*. High order interpolation shape functions N_i are also considered: for a same number of mesh elements, and thus degrees of freedom, they typically lead to a much more accurate solution so that the mesh can be coarser with fewer elements for an as accurate solution. For more details on the selected high order shape functions and their implementation refer to section A.2.

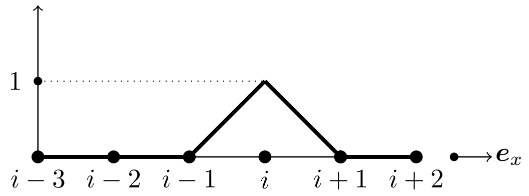


Figure 2.1: First order 1D Lagrange shape function for node i

The resolution of the algebraic system (2.67) provides the set of m coefficients required in (2.66) to compute the unknown field v. $\boldsymbol{K}$ is called *stiffness* matrix of the weak electrostatic formulation. The entry at the ith row and jth column in matrix $\boldsymbol{K}$ can be computed as follows:

$$\boldsymbol{K}_{ij} = -\int_{\Omega_e^*} \epsilon \nabla N_j \cdot \nabla N_i \, d\Omega_e^* + \int_{\partial\Omega_e^*} N_i \, \epsilon \, \partial_{\boldsymbol{n}} v \, d\partial\Omega_e^*, \tag{2.68}$$

while the ith entry in vector $\boldsymbol{b}$ is

$$\boldsymbol{b}_i = \int_{\Omega_e^*} - \rho_v \, N_i \, d\Omega_e^*. \tag{2.69}$$

In this work the discretised electrostatic weak formulation does not include the volumic charge term (2.69): the electric potential value is instead forced to the desired value on the electrode and to 0 on the ground of the reference test case in figure 1.5.

2.3.2 Elastodynamics

The weak elastodynamic formulation was derived in (2.16):

$$\begin{aligned} -\int_{\Omega_m} &[\mathcal{M}_n(\boldsymbol{u}')^T \boldsymbol{C_{11}} \mathcal{M}_n(\boldsymbol{u}) + \mathcal{M}_n(\boldsymbol{u}')^T \boldsymbol{C_{12}} \, \mathcal{M}_s(\boldsymbol{u}) + \mathcal{M}_s(\boldsymbol{u}')^T \boldsymbol{C_{21}} \mathcal{M}_n(\boldsymbol{u}) \\ &+ \mathcal{M}_s(\boldsymbol{u}')^T \boldsymbol{C_{22}} \mathcal{M}_s(\boldsymbol{u})] d\Omega_m + \int_{\partial\Omega_m} \boldsymbol{n} \cdot (\boldsymbol{\sigma} \boldsymbol{u}') \, d\partial\Omega_m + \int_{\Omega_m} \boldsymbol{f} \cdot \boldsymbol{u}' d\Omega_m = \int_{\Omega_m} \rho \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{u}' d\Omega_m. \end{aligned} \tag{2.70}$$

Every component of the mechanical displacement $\boldsymbol{u} = [u_x \; u_y \; u_z]^T$ can be discretised with (2.66):

$$u_x = \sum_{i=1}^{m} c_{xi}(t) \, N_i(\boldsymbol{x}), \tag{2.71}$$

$$u_y = \sum_{i=1}^{m} c_{yi}(t) \, N_i(\boldsymbol{x}), \tag{2.72}$$

$$u_z = \sum_{i=1}^{m} c_{zi}(t) \, N_i(\boldsymbol{x}), \tag{2.73}$$

so that

$$\boldsymbol{u} = \sum_{i=1}^{m} \left(c_{xi} \, [N_i \; 0 \; 0]^T + c_{yi} \, [0 \; N_i \; 0]^T + c_{zi} \, [0 \; 0 \; N_i]^T \right), \tag{2.74}$$

which can be rewritten as

$$\boldsymbol{u} = \sum_{i=1}^{m} \left(c_{xi} \, \boldsymbol{N}_{xi} + c_{yi} \, \boldsymbol{N}_{yi} + c_{zi} \, \boldsymbol{N}_{zi} \right), \tag{2.75}$$

where

$$\begin{aligned} \boldsymbol{N}_{xi} &= \begin{bmatrix} N_i & 0 & 0 \end{bmatrix}^T, \\ \boldsymbol{N}_{yi} &= \begin{bmatrix} 0 & N_i & 0 \end{bmatrix}^T, \\ \boldsymbol{N}_{zi} &= \begin{bmatrix} 0 & 0 & N_i \end{bmatrix}^T. \end{aligned} \tag{2.76}$$

Using (2.75) to discretise the weak elasticity formulation (2.70) leads in general to an algebraic system $\boldsymbol{M}\frac{\partial^2 \boldsymbol{c}}{\partial t^2} + \boldsymbol{C}\frac{\partial \boldsymbol{c}}{\partial t} + \boldsymbol{K}\boldsymbol{c} = \boldsymbol{b}$:

$$\begin{aligned} &\begin{bmatrix} M_{11} & \cdots & M_{1m} \\ \vdots & \ddots & \vdots \\ M_{m1} & \cdots & M_{mm} \end{bmatrix} \frac{\partial^2}{\partial t^2} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} + \begin{bmatrix} C_{11} & \cdots & C_{1m} \\ \vdots & \ddots & \vdots \\ C_{m1} & \cdots & C_{mm} \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} \\ &+ \begin{bmatrix} K_{11} & \cdots & K_{1m} \\ \vdots & \ddots & \vdots \\ K_{m1} & \cdots & K_{mm} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \end{aligned}, \tag{2.77}$$

whose resolution provides the set of m coefficients c_x, c_y and c_z required in (2.75) to compute the unknown displacement field $\boldsymbol{u}$. $\boldsymbol{K}$ is the *stiffness* matrix, $\boldsymbol{C}$ is the *damping* matrix and $\boldsymbol{M}$ is the *mass* matrix. In (2.70) matrix $\boldsymbol{C}$ is simply $\boldsymbol{0}$. Matrix $\boldsymbol{K}$ and $\boldsymbol{M}$ are made up of 9 blocks

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{xx} & \boldsymbol{K}_{xy} & \boldsymbol{K}_{xz} \\ \boldsymbol{K}_{yx} & \boldsymbol{K}_{yy} & \boldsymbol{K}_{yz} \\ \boldsymbol{K}_{zx} & \boldsymbol{K}_{zy} & \boldsymbol{K}_{zz} \end{bmatrix}, \tag{2.78}$$

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{xx} & \boldsymbol{M}_{xy} & \boldsymbol{M}_{xz} \\ \boldsymbol{M}_{yx} & \boldsymbol{M}_{yy} & \boldsymbol{M}_{yz} \\ \boldsymbol{M}_{zx} & \boldsymbol{M}_{zy} & \boldsymbol{M}_{zz} \end{bmatrix}, \tag{2.79}$$

where each row corresponds to a component of the test function $\boldsymbol{u}'$ and each column to a component of the unknown field $\boldsymbol{u}$. Vector $\boldsymbol{b}$ is only made up of 3 blocks:

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_x \\ \boldsymbol{b}_y \\ \boldsymbol{b}_z \end{bmatrix} \tag{2.80}$$

each of them corresponding to a component of the test function $\boldsymbol{u}'$.

With that definition the entry at the ith row and jth column in matrix $\boldsymbol{K}_{xz}$ for example (the remaining 8 submatrices are computed similarly) can be computed as follows:

$$\begin{aligned} (\boldsymbol{K}_{xz})_{ij} = -\int_{\Omega_m} & [\mathcal{M}_n(\boldsymbol{N}_{xi})^T \boldsymbol{C_{11}} \mathcal{M}_n(\boldsymbol{N}_{zj}) + \mathcal{M}_n(\boldsymbol{N}_{xi})^T \boldsymbol{C_{12}} \mathcal{M}_s(\boldsymbol{N}_{zj}) + \mathcal{M}_s(\boldsymbol{N}_{xi})^T \boldsymbol{C_{21}} \\ & \cdot \mathcal{M}_n(\boldsymbol{N}_{zj}) + \mathcal{M}_s(\boldsymbol{N}_{xi})^T \boldsymbol{C_{22}} \mathcal{M}_s(\boldsymbol{N}_{zj})] d\Omega_m + \textstyle\int_{\partial\Omega_m} \boldsymbol{n} \cdot (\boldsymbol{\sigma}(\boldsymbol{N}_{zj}) \boldsymbol{N}_{xi})\, d\partial\Omega_m, \end{aligned} \tag{2.81}$$

the entry at the ith row and jth column in matrix $\boldsymbol{M}_{zy}$ for example can be computed as

$$(\boldsymbol{M}_{zy})_{ij} = -\int_{\Omega_m} \rho \boldsymbol{N}_{yj} \cdot \boldsymbol{N}_{zi}\, d\Omega_m, \tag{2.82}$$

while the ith entry in vector $\boldsymbol{b}_x$ for example is

$$(\boldsymbol{b}_x)_i = \int_{\Omega_m} \boldsymbol{f}_{\text{electrostatic}} \cdot \boldsymbol{N}_{xi}\, d\Omega_m + \int_{\Omega_m \cap \Omega_a} \delta p\, \boldsymbol{n} \cdot \boldsymbol{N}_{xi}\, d\Omega. \tag{2.83}$$

i.e. the sum of the electrostatic force and the pressure force. Homogeneous Dirichlet boundary conditions (i.e. a clamp) are applied on the bulk clamp in the reference CMUT of figure 1.5.

2.3.3 Electrostatic force

In the continuous formulation (2.36) used to compute the electrostatic force the perturbation $\boldsymbol{\delta u}$ can be extracted from the gradient when considering the finite element approximation $\delta u_x = \boldsymbol{N}^T \boldsymbol{\delta U_x}$, $\delta u_y = \boldsymbol{N}^T \boldsymbol{\delta U_y}$ and $\delta u_z = \boldsymbol{N}^T \boldsymbol{\delta U_z}$ in which $\boldsymbol{N}$ is the column vector defined in (2.66). $\boldsymbol{U_x}$ is the column vector listing all shape function coefficients of the δu_x field's finite element approximation. Using this approximation one can write:

$$\begin{array}{rcl} \nabla \delta u_x &=& \nabla \boldsymbol{N}^T\, \boldsymbol{\delta U_x}, \\ \nabla \delta u_y &=& \nabla \boldsymbol{N}^T\, \boldsymbol{\delta U_y}, \\ \nabla \delta u_z &=& \nabla \boldsymbol{N}^T\, \boldsymbol{\delta U_z}, \end{array} \tag{2.84}$$

where $\nabla \boldsymbol{N}^T$ has a row for the x, y and z spatial derivatives. Injecting (2.84) in (2.36) gives:

$$\begin{gathered} \int_{\Omega^*} \begin{bmatrix} \boldsymbol{f}_x \\ \boldsymbol{f}_y \\ \boldsymbol{f}_z \end{bmatrix}^T \begin{bmatrix} \boldsymbol{N}^T & 0 & 0 \\ 0 & \boldsymbol{N}^T & 0 \\ 0 & 0 & \boldsymbol{N}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta U_x} \\ \boldsymbol{\delta U_y} \\ \boldsymbol{\delta U_z} \end{bmatrix} d\Omega^* = \\ \frac{1}{2} \int_{\Omega^*} \epsilon\, (\nabla v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 & 2\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial x} & 0 & 2\frac{\partial v}{\partial z} & 0 & -\frac{\partial v}{\partial x} \\ -\frac{\partial v}{\partial y} & 2\frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & 2\frac{\partial v}{\partial z} & -\frac{\partial v}{\partial y} \\ -\frac{\partial v}{\partial z} & 0 & 2\frac{\partial v}{\partial x} & 0 & -\frac{\partial v}{\partial z} & 2\frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \\ \cdot \begin{bmatrix} \nabla \boldsymbol{N}^T & 0 & 0 \\ 0 & \nabla \boldsymbol{N}^T & 0 \\ 0 & 0 & \nabla \boldsymbol{N}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta U_x} \\ \boldsymbol{\delta U_y} \\ \boldsymbol{\delta U_z} \end{bmatrix} d\Omega^*. \end{gathered} \tag{2.85}$$

Since (2.85) is valid for any compatbile perturbation they can simply be removed from the equation. Considering only the *ith* test function N_i we get:

$$\begin{gathered} \int_{\Omega^*} \begin{bmatrix} \boldsymbol{f}_x N_i & \boldsymbol{f}_y N_i & \boldsymbol{f}_z N_i \end{bmatrix} d\Omega^* = \\ \frac{1}{2} \int_{\Omega^*} \epsilon\, (\nabla v)^T \begin{bmatrix} \frac{\partial v}{\partial x} & 0 & 0 & 2\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial x} & 0 & 2\frac{\partial v}{\partial z} & 0 & -\frac{\partial v}{\partial x} \\ -\frac{\partial v}{\partial y} & 2\frac{\partial v}{\partial x} & 0 & 0 & \frac{\partial v}{\partial y} & 0 & 0 & 2\frac{\partial v}{\partial z} & -\frac{\partial v}{\partial y} \\ -\frac{\partial v}{\partial z} & 0 & 2\frac{\partial v}{\partial x} & 0 & -\frac{\partial v}{\partial z} & 2\frac{\partial v}{\partial y} & 0 & 0 & \frac{\partial v}{\partial z} \end{bmatrix} \\ \cdot \begin{bmatrix} \nabla N_i & 0 & 0 \\ 0 & \nabla N_i & 0 \\ 0 & 0 & \nabla N_i \end{bmatrix} d\Omega^*. \end{gathered} \tag{2.86}$$

Expanding (2.86) finally shows how the discretised components of the electrostatic force can be computed:

$$\boxed{\begin{array}{l} \displaystyle\int_{\Omega^*} \left[\begin{array}{ccc} \boldsymbol{f}_x N_i & \boldsymbol{f}_y N_i & \boldsymbol{f}_z N_i \end{array} \right]^T d\Omega^* = \\ \displaystyle\frac{1}{2}\int_{\Omega^*} \epsilon \left[\begin{array}{c} \frac{\partial v}{\partial x}^2 \frac{\partial N_i}{\partial x} - \frac{\partial v}{\partial y}^2 \frac{\partial N_i}{\partial x} - \frac{\partial v}{\partial z}^2 \frac{\partial N_i}{\partial x} + 2\frac{\partial v}{\partial x}\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial y} + 2\frac{\partial v}{\partial x}\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial z} \\ -\frac{\partial v}{\partial x}^2 \frac{\partial N_i}{\partial y} + \frac{\partial v}{\partial y}^2 \frac{\partial N_i}{\partial y} - \frac{\partial v}{\partial z}^2 \frac{\partial N_i}{\partial y} + 2\frac{\partial v}{\partial x}\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial x} + 2\frac{\partial v}{\partial y}\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial z} \\ -\frac{\partial v}{\partial x}^2 \frac{\partial N_i}{\partial z} - \frac{\partial v}{\partial y}^2 \frac{\partial N_i}{\partial z} + \frac{\partial v}{\partial z}^2 \frac{\partial N_i}{\partial z} + 2\frac{\partial v}{\partial x}\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial x} + 2\frac{\partial v}{\partial y}\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial y} \end{array} \right] d\Omega^*, \end{array}} \tag{2.87}$$

The equation system (2.87) can readily be used to compute the contribution of the electrostatic force to the right handside vector $\boldsymbol{b}$ in the discretised elastodynamic system (2.77). Indeed we have the following relations

$$\begin{array}{l} (\boldsymbol{b}_x)_i = \int_{\Omega^*} \boldsymbol{f}_x N_i d\Omega^* \\ (\boldsymbol{b}_y)_i = \int_{\Omega^*} \boldsymbol{f}_y N_i d\Omega^* \\ (\boldsymbol{b}_z)_i = \int_{\Omega^*} \boldsymbol{f}_z N_i d\Omega^* \end{array} \tag{2.88}$$

2.3.4 Linear acoustics

The weak acoustic formulation (2.38) is

$$-\int_{\Omega_a} \nabla\delta p \cdot \nabla\delta p' d\Omega - \int_{\partial\Omega_a \cap \partial\Omega_m} \delta p' \, \rho_{\text{fluid}} \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{n} \; d\partial\Omega - \int_{\partial\Omega_a \setminus \partial\Omega_m} \delta p' \, \frac{1}{c}\frac{\partial \delta p}{\partial t} \; d\partial\Omega - \int_{\Omega_a} \frac{1}{c^2}\frac{\partial^2 \delta p}{\partial t^2} \delta p' d\Omega = 0. \tag{2.89}$$

The acoustic pressure variation δp can be discretised using (2.66) as

$$\delta p = \sum_{i=1}^{m} c_i \, N_i = \boldsymbol{N}^T \boldsymbol{c}, \tag{2.90}$$

to give a system of the form $\boldsymbol{M}\frac{\partial^2 \boldsymbol{c}}{\partial t^2} + \boldsymbol{C}\frac{\partial \boldsymbol{c}}{\partial t} + \boldsymbol{K}\boldsymbol{c} = \boldsymbol{b}$ where the ith row, jth column of the stiffness matrix $\boldsymbol{K}$ is given by

$$\boldsymbol{K}_{ij} = -\int_{\Omega_a} \nabla N_j \cdot \nabla N_i d\Omega, \tag{2.91}$$

the ith row, jth column of the damping matrix $\boldsymbol{C}$ is

$$\boldsymbol{C}_{ij} = -\int_{\partial\Omega_a \setminus \partial\Omega_m} N_i \frac{1}{c} N_j \, d\partial\Omega, \tag{2.92}$$

the ith row, jth column of the mass matrix $\boldsymbol{M}$ is

$$\boldsymbol{M}_{ij} = -\int_{\Omega_a} \frac{1}{c^2} N_j N_i d\Omega_m, \tag{2.93}$$

and the ith entry of vector $\boldsymbol{b}$ is

$$\boldsymbol{b}_i = -\int_{\partial\Omega_a \cap \partial\Omega_m} N_i\, \rho_{\text{fluid}} \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{n}\, d\partial\Omega. \tag{2.94}$$

2.3.5 Sensitivity matrix

The sensitivity matrix derived in (2.46) has three nontrivial terms which will contribute to the discrete sensitivity matrix: $\boldsymbol{K}_{\boldsymbol{u}v}$, $\boldsymbol{K}_{v\boldsymbol{u}}$ and $\boldsymbol{K}_{\boldsymbol{u}\boldsymbol{u}}$.

$\boldsymbol{K}_{\boldsymbol{u}v}$ matrix block:

In the same way as for (2.84) one can use the finite element approximation for $\boldsymbol{\delta u}$ and δv and remove the perturbation shape function coefficient vectors $\boldsymbol{\delta U}$ and $\boldsymbol{\delta V}$ on both sides of equation (2.53):

$$\begin{aligned} &\int_{\Omega^*} \boldsymbol{\delta V}^T \boldsymbol{N} \boldsymbol{K}_{\boldsymbol{u}v}^T \begin{bmatrix} \boldsymbol{N}^T & 0 & 0 \\ 0 & \boldsymbol{N}^T & 0 \\ 0 & 0 & \boldsymbol{N}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta U_x} \\ \boldsymbol{\delta U_y} \\ \boldsymbol{\delta U_z} \end{bmatrix} d\Omega^* = -\int_{\Omega^*} \epsilon\, \boldsymbol{\delta V}^T (\nabla \boldsymbol{N}^T)^T \\ &\cdot \begin{bmatrix} -\frac{\partial v}{\partial x} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial z} & -\frac{\partial v}{\partial y} & \frac{\partial v}{\partial x} & 0 & -\frac{\partial v}{\partial z} & 0 & \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} & -\frac{\partial v}{\partial x} & 0 & -\frac{\partial v}{\partial x} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial z} & 0 & -\frac{\partial v}{\partial z} & \frac{\partial v}{\partial y} \\ \frac{\partial v}{\partial z} & 0 & -\frac{\partial v}{\partial x} & 0 & \frac{\partial v}{\partial z} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial x} & -\frac{\partial v}{\partial y} & -\frac{\partial v}{\partial z} \end{bmatrix} \\ &\cdot \begin{bmatrix} \nabla \boldsymbol{N}^T & 0 & 0 \\ 0 & \nabla \boldsymbol{N}^T & 0 \\ 0 & 0 & \nabla \boldsymbol{N}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta U_x} \\ \boldsymbol{\delta U_y} \\ \boldsymbol{\delta U_z} \end{bmatrix} d\Omega^*, \end{aligned} \tag{2.95}$$

where $(\nabla \boldsymbol{N}^T)^T$ has a column for the x, for the y and for the z spatial derivatives. Since this is valid for any $\boldsymbol{\delta u}$ and δv the perturbations their coefficient vectors can be removed. The algebraic matrix $\boldsymbol{K}_{\boldsymbol{u}v}$ obtained after discretisation has the form

$$\boldsymbol{K}_{\boldsymbol{u}v} = \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{u}_x v} \\ \boldsymbol{K}_{\boldsymbol{u}_y v} \\ \boldsymbol{K}_{\boldsymbol{u}_z v} \end{bmatrix}. \tag{2.96}$$

i.e. it has a block for every component of $\boldsymbol{u}$.

The ith row (corresponding to the displacement) and jth column (corresponding to the electric potential) of the sub-blocks $\boldsymbol{K}_{\boldsymbol{u}_x v}$, $\boldsymbol{K}_{\boldsymbol{u}_y v}$ and $\boldsymbol{K}_{\boldsymbol{u}_z v}$ in the discretised matrix $\boldsymbol{K}_{\boldsymbol{u}v}$ can be computed by using the following relations:

$$\boxed{\begin{aligned}&\int_{\Omega^*}\begin{bmatrix} N_j\,\boldsymbol{K}_{\boldsymbol{u}_x v}\,N_i \\ N_j\,\boldsymbol{K}_{\boldsymbol{u}_y v}\,N_i \\ N_j\,\boldsymbol{K}_{\boldsymbol{u}_z v}\,N_i \end{bmatrix} d\Omega^* = \\ &-\int_{\Omega^*}\epsilon\begin{bmatrix} \begin{array}{c} -\frac{\partial v}{\partial x}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x}-\frac{\partial v}{\partial y}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial y}-\frac{\partial v}{\partial z}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial z}+\frac{\partial v}{\partial y}\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial x} \\ -\frac{\partial v}{\partial x}\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial y}+\frac{\partial v}{\partial z}\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial x}-\frac{\partial v}{\partial x}\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial z} \end{array} \\ \\ \begin{array}{c} -\frac{\partial v}{\partial x}\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial x}-\frac{\partial v}{\partial y}\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial y}-\frac{\partial v}{\partial z}\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial z}-\frac{\partial v}{\partial y}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x} \\ +\frac{\partial v}{\partial x}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial y}+\frac{\partial v}{\partial z}\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial y}-\frac{\partial v}{\partial y}\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial z} \end{array} \\ \\ \begin{array}{c} -\frac{\partial v}{\partial x}\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial x}-\frac{\partial v}{\partial y}\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial y}-\frac{\partial v}{\partial z}\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial z}-\frac{\partial v}{\partial z}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x} \\ +\frac{\partial v}{\partial x}\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial z}-\frac{\partial v}{\partial z}\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial y}+\frac{\partial v}{\partial y}\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial z} \end{array} \\ \\ \end{bmatrix} d\Omega^*. \end{aligned}} \tag{2.97}$$

$\boldsymbol{K}_{v\boldsymbol{u}}$ matrix block:

The $\boldsymbol{K}_{v\boldsymbol{u}}$ block corresponds to $-\frac{\partial^2 W_e}{\partial \boldsymbol{u}\partial v}$. It is the transposition of $\boldsymbol{K}_{\boldsymbol{u}v}$ and thus has the form

$$\boldsymbol{K}_{v\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{K}_{v\boldsymbol{u}_x} & \boldsymbol{K}_{v\boldsymbol{u}_y} & \boldsymbol{K}_{v\boldsymbol{u}_z} \end{bmatrix} \tag{2.98}$$

i.e. it has a column for every component of $\boldsymbol{u}$. The terms for $\boldsymbol{K}_{v\boldsymbol{u}}$ are simply obtained by flipping i and j in (2.97):

$$\boxed{\begin{aligned}&\int_{\Omega^*}\begin{bmatrix} N_i\boldsymbol{K}_{v\boldsymbol{u}_x}N_j & N_i\boldsymbol{K}_{v\boldsymbol{u}_y}N_j & N_i\boldsymbol{K}_{v\boldsymbol{u}_z}N_j \end{bmatrix} d\Omega^* = \\ &-\int_{\Omega^*}\epsilon\begin{bmatrix} \begin{array}{c} -\frac{\partial v}{\partial x}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}-\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y}-\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial z}+\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x} \\ -\frac{\partial v}{\partial x}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}+\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x}-\frac{\partial v}{\partial x}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z} \end{array} \\ \\ \begin{array}{c} -\frac{\partial v}{\partial x}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x}-\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}-\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial z}-\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} \\ +\frac{\partial v}{\partial x}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y}+\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial y}-\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z} \end{array} \\ \\ \begin{array}{c} -\frac{\partial v}{\partial x}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x}-\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial y}-\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}-\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} \\ +\frac{\partial v}{\partial x}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial z}-\frac{\partial v}{\partial z}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}+\frac{\partial v}{\partial y}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial z} \end{array} \\ \\ \end{bmatrix}^T d\Omega^*, \end{aligned}} \tag{2.99}$$

$\boldsymbol{K}_{\boldsymbol{u}\boldsymbol{u}}$ matrix block:

$\boldsymbol{K}_{\boldsymbol{u}\boldsymbol{u}}$ has the form

$$\boldsymbol{K}_{\boldsymbol{u}\boldsymbol{u}} = \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{u}_x\boldsymbol{u}_x} & \boldsymbol{K}_{\boldsymbol{u}_x\boldsymbol{u}_y} & \boldsymbol{K}_{\boldsymbol{u}_x\boldsymbol{u}_z} \\ \boldsymbol{K}_{\boldsymbol{u}_y\boldsymbol{u}_x} & \boldsymbol{K}_{\boldsymbol{u}_y\boldsymbol{u}_y} & \boldsymbol{K}_{\boldsymbol{u}_y\boldsymbol{u}_z} \\ \boldsymbol{K}_{\boldsymbol{u}_z\boldsymbol{u}_x} & \boldsymbol{K}_{\boldsymbol{u}_z\boldsymbol{u}_y} & \boldsymbol{K}_{\boldsymbol{u}_z\boldsymbol{u}_z} \end{bmatrix} \tag{2.100}$$

i.e. it has a row and a column for every component of $\boldsymbol{u}$. Its discretisation is based on the calculations performed in section (2.2.5): taking the difference between the force computed on the unpertubed domain Ω and on the domain Ω^p perturbed by $\boldsymbol{\delta u^2}$ while replacing the perturbations by their finite element approximation $\delta u^1_x = \boldsymbol{N^1}^T\boldsymbol{U^1_x}$, $\delta u^2_x = \boldsymbol{N^2}^T\boldsymbol{U^2_x}$ as done for the electrostatic force computation one gets:

$$
\begin{aligned}
&\int_{\Omega^*}(\boldsymbol{\delta u^2})^T\frac{\partial \boldsymbol{f}_{\text{elec}}}{\partial \boldsymbol{u}}\boldsymbol{\delta u^1}d\Omega^* = \int_{\Omega^*}(\boldsymbol{\delta u^2})^T\boldsymbol{K_{uu}}\boldsymbol{\delta u^1}d\Omega^* = \\
&\int_{\Omega^*}\begin{bmatrix}\boldsymbol{\delta U^2_x}\\ \boldsymbol{\delta U^2_y}\\ \boldsymbol{\delta U^2_z}\end{bmatrix}^T\begin{bmatrix}\boldsymbol{N^2} & 0 & 0\\ 0 & \boldsymbol{N^2} & 0\\ 0 & 0 & \boldsymbol{N^2}\end{bmatrix}\boldsymbol{K_{uu}}\begin{bmatrix}\boldsymbol{N^1}^T & 0 & 0\\ 0 & \boldsymbol{N^1}^T & 0\\ 0 & 0 & \boldsymbol{N^1}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta U^1_x}\\ \boldsymbol{\delta U^1_y}\\ \boldsymbol{\delta U^1_z}\end{bmatrix}d\Omega^* = \\
&\int_{\Omega^*}-\frac{\epsilon}{2}\begin{bmatrix}\boldsymbol{\delta U^2_x}\\ \boldsymbol{\delta U^2_y}\\ \boldsymbol{\delta U^2_z}\end{bmatrix}^T\begin{bmatrix}\nabla\boldsymbol{N^2} & 0 & 0\\ 0 & \nabla\boldsymbol{N^2} & 0\\ 0 & 0 & \nabla\boldsymbol{N^2}\end{bmatrix}
\begin{bmatrix}
2E_x^2 & 0 & 0 & 2E_xE_y & -E_x^2-E_y^2+E_z^2 & -2E_yE_z & 2E_xE_z & -2E_yE_z & -E_x^2+E_y^2-E_z^2\\
0 & 2E_x^2 & 0 & E_x^2+E_y^2-E_z^2 & 2E_xE_y & 2E_xE_z & 2E_yE_z & 2E_xE_z & -2E_xE_y\\
0 & 0 & 2E_x^2 & 2E_yE_z & -2E_xE_z & 2E_xE_y & E_x^2-E_y^2+E_z^2 & 2E_xE_y & 2E_xE_z\\
2E_xE_y & E_x^2+E_y^2-E_z^2 & 2E_yE_z & 2E_y^2 & 0 & 0 & 2E_yE_z & 2E_xE_z & -2E_xE_y\\
-E_x^2-E_y^2+E_z^2 & 2E_xE_y & -2E_xE_z & 0 & 2E_y^2 & 0 & -2E_xE_z & 2E_yE_z & E_x^2-E_y^2-E_z^2\\
-2E_yE_z & 2E_xE_z & 2E_xE_y & 0 & 0 & 2E_y^2 & 2E_xE_y & -E_x^2+E_y^2+E_z^2 & 2E_yE_z\\
2E_xE_z & 2E_yE_z & E_x^2-E_y^2+E_z^2 & 2E_yE_z & -2E_xE_z & 2E_xE_y & 2E_z^2 & 0 & 0\\
-2E_yE_z & 2E_xE_z & 2E_xE_y & 2E_xE_z & 2E_yE_z & -E_x^2+E_y^2+E_z^2 & 0 & 2E_z^2 & 0\\
-E_x^2+E_y^2-E_z^2 & -2E_xE_y & 2E_xE_z & -2E_xE_y & E_x^2-E_y^2-E_z^2 & 2E_yE_z & 0 & 0 & 2E_z^2
\end{bmatrix} \\
&\qquad\begin{bmatrix}\nabla\boldsymbol{N^1}^T & 0 & 0\\ 0 & \nabla\boldsymbol{N^1}^T & 0\\ 0 & 0 & \nabla\boldsymbol{N^1}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\delta U^1_x}\\ \boldsymbol{\delta U^1_y}\\ \boldsymbol{\delta U^1_z}\end{bmatrix}d\Omega^*,
\end{aligned}
\tag{2.101}
$$

where $\begin{bmatrix} E_x & E_y & E_z \end{bmatrix} = \begin{bmatrix} \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \end{bmatrix}$ and (2.84) was used to express the finite element approximation of the perturbation's gradient. Again $\nabla\boldsymbol{N^2}^T$ is a matrix with row 1 giving the

x derivative of the row vector $\boldsymbol{N^2}^T$, row 2 the y derivative and row 3 the z derivative. Since (2.101) must hold for any $\boldsymbol{\delta u^1}$ and $\boldsymbol{\delta u^2}$ perturbation their coefficient vectors $\boldsymbol{\delta U^1}$ and $\boldsymbol{\delta U^2}$ can be removed. The ith row, jth column of the discretised submatrices in $\boldsymbol{K_{uu}}$ is given by:

$$
\boxed{
\begin{array}{l}
\displaystyle\int_{\Omega^*}\begin{bmatrix} N_j\boldsymbol{K}_{\boldsymbol{u}_x\boldsymbol{u}_x}N_i & N_j\boldsymbol{K}_{\boldsymbol{u}_x\boldsymbol{u}_y}N_{i,x} & N_j\boldsymbol{K}_{\boldsymbol{u}_x\boldsymbol{u}_z}N_i \\ N_j\boldsymbol{K}_{\boldsymbol{u}_y\boldsymbol{u}_x}N_i & N_j\boldsymbol{K}_{\boldsymbol{u}_y\boldsymbol{u}_y}N_i & N_j\boldsymbol{K}_{\boldsymbol{u}_y\boldsymbol{u}_z}N_i \\ N_j\boldsymbol{K}_{\boldsymbol{u}_z\boldsymbol{u}_x}N_i & N_j\boldsymbol{K}_{\boldsymbol{u}_z\boldsymbol{u}_y}N_i & N_j\boldsymbol{K}_{\boldsymbol{u}_z\boldsymbol{u}_z}N_i \end{bmatrix} d\Omega^* = \\[2em]
\displaystyle\int_{\Omega^*} -\frac{\epsilon}{2}\left[\begin{array}{c}
2E_x^2\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x} + 2E_x^2\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial y} + 2E_x^2\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial z} \\[2em]
[2E_xE_y\frac{\partial N_j}{\partial x} + (E_x^2 + E_y^2 - E_z^2)\frac{\partial N_j}{\partial y} + 2E_yE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial x} \\
+[(-E_x^2 - E_y^2 + E_z^2)\frac{\partial N_j}{\partial x} + 2E_xE_y\frac{\partial N_j}{\partial y} - 2E_xE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial y} \\
+[-2E_yE_z\frac{\partial N_j}{\partial x} + 2E_xE_z\frac{\partial N_j}{\partial y} + 2E_xE_y\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial z} \\[2em]
[2E_xE_z\frac{\partial N_j}{\partial x} + 2E_yE_z\frac{\partial N_j}{\partial y} + (E_x^2 - E_y^2 + E_z^2)\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial x} \\
+[-2E_yE_z\frac{\partial N_j}{\partial x} + 2E_xE_z\frac{\partial N_j}{\partial y} + 2E_xE_y\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial y} \\
+[(-E_x^2 + E_y^2 - E_z^2)\frac{\partial N_j}{\partial x} - 2E_xE_y\frac{\partial N_j}{\partial y} + 2E_xE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial z}
\end{array}\right. \qquad \cdots \\[3em]
\begin{array}{c}
[2E_xE_y\frac{\partial N_j}{\partial x} + (-E_x^2 - E_y^2 + E_z^2)\frac{\partial N_j}{\partial y} - 2E_yE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial x} \\
+[(E_x^2 + E_y^2 - E_z^2)\frac{\partial N_j}{\partial x} + 2E_xE_y\frac{\partial N_j}{\partial y} + 2E_xE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial y} \\
+[2E_yE_z\frac{\partial N_j}{\partial x} - 2E_xE_z\frac{\partial N_j}{\partial y} + 2E_xE_y\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial z} \\[2em]
\cdots \qquad 2E_y^2\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x} + 2E_y^2\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial y} + 2E_y^2\frac{\partial N_j}{\partial z}\frac{\partial N_i}{\partial z} \qquad \cdots \\[2em]
[2E_yE_z\frac{\partial N_j}{\partial x} - 2E_xE_z\frac{\partial N_j}{\partial y} + 2E_xE_y\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial x} \\
+[2E_xE_z\frac{\partial N_j}{\partial x} + 2E_yE_z\frac{\partial N_j}{\partial y} + (-E_x^2 + E_y^2 + E_z^2)\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial y} \\
+[-2E_xE_y\frac{\partial N_j}{\partial x} + (E_x^2 - E_y^2 - E_z^2)\frac{\partial N_j}{\partial y} + 2E_yE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial z}
\end{array} \\[3em]
\cdots \quad \left.\begin{array}{c}
[2E_xE_z\frac{\partial N_j}{\partial x} - 2E_yE_z\frac{\partial N_j}{\partial y} + (-E_x^2 + E_y^2 - E_z^2)\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial x} \\
+[2E_yE_z\frac{\partial N_j}{\partial x} + 2E_xE_z\frac{\partial N_j}{\partial y} - 2E_xE_y\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial y} \\
+[(E_x^2 - E_y^2 + E_z^2)\frac{\partial N_j}{\partial x} + 2E_xE_y\frac{\partial N_j}{\partial y} + 2E_xE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial z} \\[2em]
[-2E_yE_z\frac{\partial N_j}{\partial x} + 2E_xE_z\frac{\partial N_j}{\partial y} - 2E_xE_y\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial x} \\
+[-2E_xE_z\frac{\partial N_j}{\partial x} + 2E_yE_z\frac{\partial N_j}{\partial y} + (E_x^2 - E_y^2 - E_z^2)\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial y} \\
+[2E_xE_y\frac{\partial N_j}{\partial x} + (-E_x^2 + E_y^2 + E_z^2)\frac{\partial N_j}{\partial y} + 2E_yE_z\frac{\partial N_j}{\partial z}]\frac{\partial N_i}{\partial z} \\[2em]
2E_z^2\frac{\partial N_j}{\partial x}\frac{\partial N_i}{\partial x} + 2E_z^2\frac{\partial N_j}{\partial y}\frac{\partial N_i}{\partial y} + 2E_z^2\frac{\partial N_{j,z}}{\partial z}\frac{\partial N_i}{\partial z}
\end{array}\right] d\Omega^*,
\end{array}
}
\tag{2.102}
$$

where $N_i \equiv N_i^1$ and $N_j \equiv N_j^2$ to have a shorter formulation.

2.4 Resolution schemes for the nonlinear coupling

Coupling the electrostatic (2.6) and elastodynamic (2.16) formulations together brings in nonlinearity as illustrated in section 1.5. The coupling is nonlinear because the electrostatic force depends nonlinearly on the electric field which depends on the mechanical deformation which itself depends on the electrostatic force. In case of very small displacements one could linearize the problem by considering that the electrostatic force and electric field do not change with the tiny deformation. When close to the pull-in voltage however the mechanical displacement can not be neglected, even though the deformations are still small enough to use linear elasticity. This is due to the particularly long shape of the vibrating membrane.

Solving the electroelastoacoustic (2.6)-(2.16)-(2.38) coupled nonlinear problem can be achieved in two main ways:

- *Staggered/weak* resolution [88]: solve the formulations independently at every nonlinear iteration while using the fields of the previous solved formulation as input for the next one
- *Monolithic/strong* resolution [3]: Solving the whole coupled problem at every nonlinear iteration, using Newton's method for faster convergence. This uses the sensitivity matrix discretised in section 2.3.5

When getting closer to the pull-in voltage the nonlinearity gets stronger and the staggered approach requires much more nonlinear iterations than Newton's method as illustrated in figure 2.2. The figure shows the number of nonlinear iterations required for both nonlinear resolution methods as the time-independent electrode-to-ground excitation voltage is increased. The computation is performed on the reference CMUT geometry of figure 1.5 (no fluid considered). Fot the same test settings with 105 V electrode-to-ground electrical excitation the nonlinear relative residual history is displayed in figure 2.3. In either resolution method the nonlinear relative residual is computed on the coupled electrostatic-elastodynamic formulation (2.6)-(2.16) at nonlinear iteration k as

$$\|\frac{\boldsymbol{b}(\boldsymbol{x_k}) - \boldsymbol{A}(\boldsymbol{x_k})\,\boldsymbol{x_k}}{\boldsymbol{b}(\boldsymbol{x_0}) - \boldsymbol{A}(\boldsymbol{x_0})\boldsymbol{x_0}}\|_2, \tag{2.103}$$

where $\boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{b}(\boldsymbol{x})$ is the algebraic system corresponding to the finite element discretised electrostatic-elastodynamic problem and $\boldsymbol{x_0}$ is an initial guess described in section 2.4.3. Quantity $\boldsymbol{x}$ includes the coefficients c_i in the finite element discretisation of the electric potential field (2.66) and of the mechanical displacement field (2.75). For reasonably low voltages the staggered resolution approach is the best choice since every iteration is lightweight compared to a Newton iteration. Beyond a certain voltage however the overhead associated to Newton's method is overcompensated by the dramatic reduction of the iteration count.

2.4.1 Staggered resolution

The staggered approach to solve the nonlinear electroelastoacoustic coupling consists in solving alternatively the electrostatic formulation (2.6) and the coupled elastoacoustic formulation (2.16)-(2.38) . At the end of every nonlinear iteration, i.e. after having solved the elastoacoustic formulation the mesh has to be deformed by the mechanical displacement $\boldsymbol{u}$ so that the electrostatic

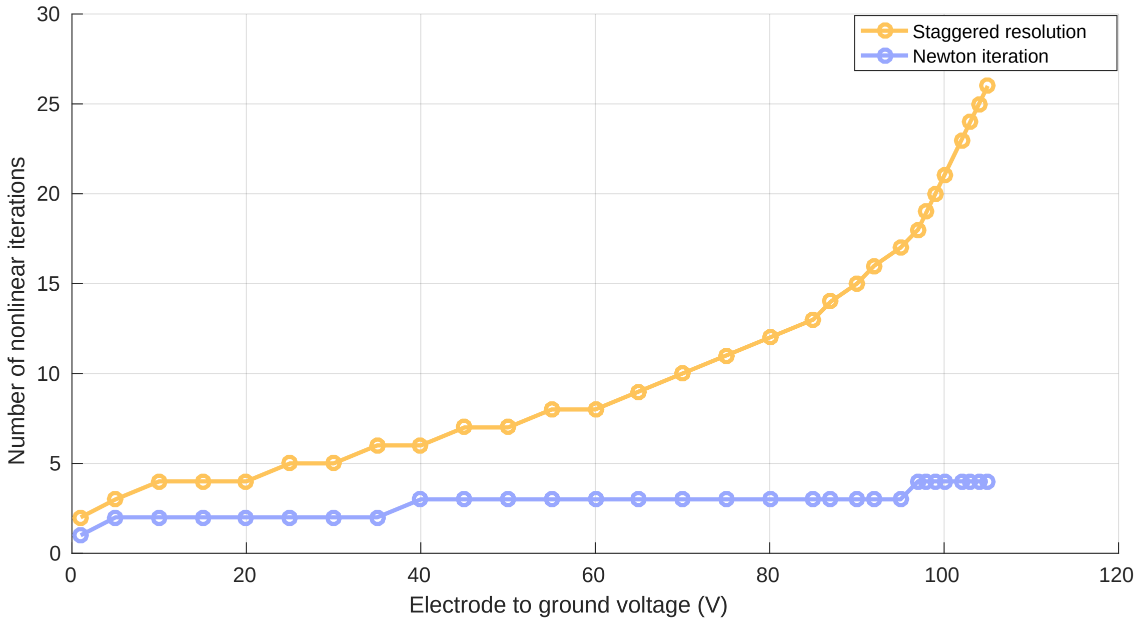


Figure 2.2: Number of nonlinear iterations to reach a 10^{-8} relative residual versus electrode to ground excitation voltage (V) for a staggered resolution scheme and a Newton iteration on the reference CMUT geometry of figure 1.5 (no fluid considered).

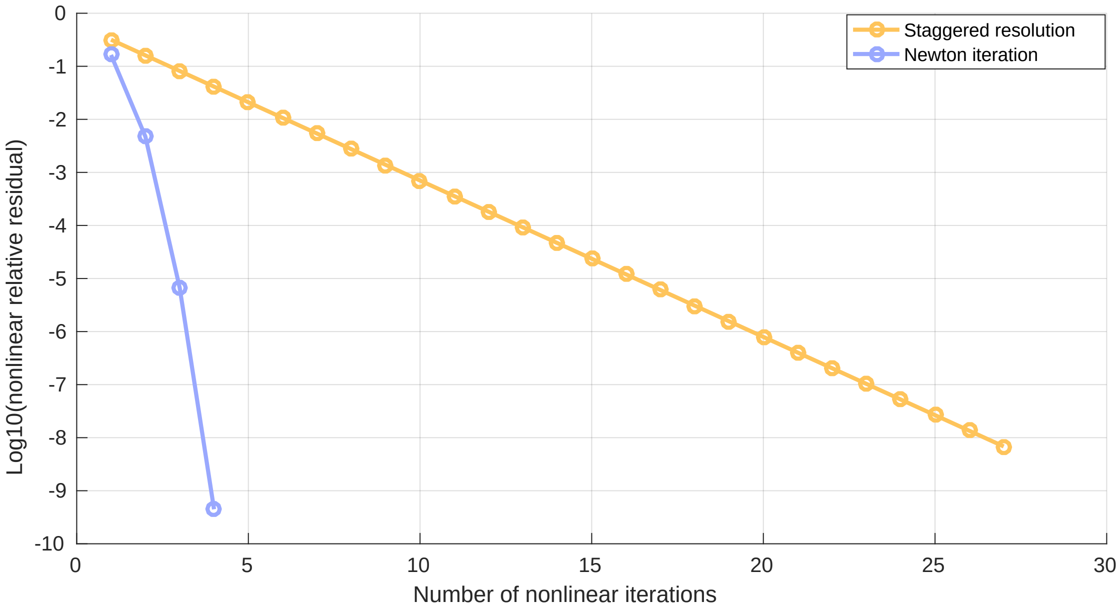


Figure 2.3: Nonlinear relative residual versus number of iterations for a staggered resolution scheme and Newton's iteration close to the pull-in voltage on the reference CMUT geometry of figure 1.5 (no fluid considered).

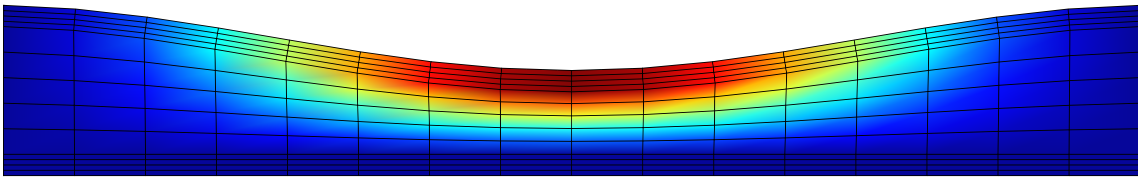


Figure 2.4: CMUT vacuum gap mesh smoothly deformed by a Laplacian resolution. Height is exagerated for illustration.

formulation and the electrostatic forces (2.87) are computed on the deformed mesh. Because no displacements are defined in the vacuum gap under the membrane (illustrated in figure 1.5) and because the membrane displacement can be a quite large portion of the gap height the gap mesh elements could be distorded. To fix this a Laplace equation is numerically solved for every space coordinate x, y and z in the vaccum gap Ω_v:

$$\begin{cases} \Delta u_{\text{mesh},x} = 0 \text{ on } \Omega_v, \\ \Delta u_{\text{mesh},y} = 0 \text{ on } \Omega_v, \\ \Delta u_{\text{mesh},z} = 0 \text{ on } \Omega_v, \\ \boldsymbol{u}_{\text{mesh}} = \boldsymbol{u} \text{ on } \Omega_m \cap \Omega_v, \end{cases} \tag{2.104}$$

with Dirichlet constraints on the solid-vacuum interface $\Omega_m \cap \Omega_v$. Figure 2.4 illustrates the smooth mesh that is obtained with (2.104).

At every outer staggered nonlinear iteration the elastoacoustic problem (2.16)-(2.38) itself can either be solved at once in a coupled fashion or iteratively until convergence. The latter resolution method requires the least amount of memory resources and has thus been investigated. Unfortunately the convergence of such weakly coupled structure-acoustic problems can be hard to achieve [39], in particular for stiff structures with a high acoustic coupling. In this thesis it has been observed that with air the problem converges when not too close to resonance while it diverges with water, even with relaxation methods. For that reason the strongly-coupled elastoacoustic resolution will be preferred. Even with this resolution method however the acoustic coupling with a very stiff structure still causes problems: the resulting algebraic system to solve has a bad conditionning [39] of up to 10^{30}. Fortunately this can be dramatically improved by rescaling the acoustic pressure force term in (1.22) whose stiffness terms are orders of magnitudes smaller than the other terms in the elastodynamic formulation (2.16). Using the same notation as in (1.22) the two coupling terms in the elastoacoustic formulation are changed as follows:

$$-\int_{\Omega_a \cap \Omega_m} \rho_{\text{fluid}} \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{n} \, \delta p' \, d\partial\Omega \rightarrow -\int_{\Omega_a \cap \Omega_m} \frac{1}{a} \rho_{\text{fluid}} \frac{\partial^2 \boldsymbol{u}}{\partial t^2} \cdot \boldsymbol{n} \, \delta p' \, d\partial\Omega, \tag{2.105}$$

$$\int_{\Omega_m \cap \Omega_a} \delta p \, \boldsymbol{n} \cdot \boldsymbol{u}' d\Omega \rightarrow \int_{\Omega_m \cap \Omega_a} a \, \delta p \, \boldsymbol{n} \cdot \boldsymbol{u}' d\Omega, \tag{2.106}$$

where a is a large constant scalar, e.g. 10^{10}. In other words the vibrating membrane pressure source is factor a weaker (2.105) and so is the acoustic pressure but the force applied on the mechanical membrane is a times stronger (2.106) to compensate. This increases the stiffnesses associated to term $\displaystyle\int_{\Omega_m \cap \Omega_a} a\,\delta p\,\boldsymbol{n} \cdot \boldsymbol{u}' d\Omega$ to something more comparable to the other mechanical stiffnesses. The actual pressure is then simply $a\,\delta p$.

2.4.2 Newton's method

Solving iteratively the nonlinear discretised electroelastoacoustic algebraic system $\boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{b}(\boldsymbol{x})$ with Newton's method is done by updating at every iteration k the unknown field $\boldsymbol{x}$ as follows:

$$\begin{aligned}\boldsymbol{J}(\boldsymbol{x}_k)\boldsymbol{dx} &= \boldsymbol{b}(\boldsymbol{x}_k) - \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{x}_k,\\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \boldsymbol{dx},\\ \boldsymbol{x}_{k=0} &= \boldsymbol{x}_0,\end{aligned} \tag{2.107}$$

where $\boldsymbol{J}$ is the iteration *Jacobian* matrix of Newton's iteration. Matrix $\boldsymbol{J}$ is equal to the sensitivity matrix discretised in section 2.3.5, obtained as a sum of $\boldsymbol{A}$ and extra terms coming from the linearisation around the kth iterate of the nonlinear terms. In the electroelastoacoustic problem considered $\boldsymbol{A}$ is obtained by generating the electrostatic (2.6), the elastodynamic (2.16) and the acoustic (2.38) systems in a same formulation using the finite element method. $\boldsymbol{J}$ is then obtained by adding to $\boldsymbol{A}$ the Jacobian terms $\boldsymbol{K_{uv}}$, $\boldsymbol{K_{vu}}$ and $\boldsymbol{K_{uu}}$ whose finite element discretised expression can be found in section 2.3.5.

Similarly to the staggered resolution method the mesh is smoothly deformed by solving (2.104) after (2.107) at every nonlinear iteration. Doing so however changes the node coordinates in the mesh without adjusting the associated electric potentials. This leads to a noisy $\boldsymbol{dx}$ update in (2.107) since the electric potentials computed at the previous nonlinear iteration were computed on the mesh as it was before smoothing. As a consequence convergence slows down as can be seen in figure 2.5. This problem is easily fixed by recomputing the electric potential on the electric mesh region where no mechanical displacements are defined $\Omega_e \backslash \Omega_m$:

$$\left\{\begin{array}{l}\text{Solve (2.6) on } \Omega_e \backslash \Omega_m,\\ v = v_{\Omega_m} \text{ on } \Omega_m.\end{array}\right. \tag{2.108}$$

2.4.3 Initial guess

The proposed iterative nonlinear resolution schemes require an initial guess $\boldsymbol{x_0} = \begin{bmatrix} \boldsymbol{u_0} & v_0 & \delta p_0 \end{bmatrix}^T$ for the mechanical displacement $\boldsymbol{u}$, the electric potential v and the acoustic pressure δp_0. An inapropriate choice can lead to the divergence of the algorithm. As an example the following initial guess on the reference geometry (figure 1.5)

$$\left\{\begin{array}{l}\boldsymbol{u}_0 = \boldsymbol{0} \text{ on } \Omega_m,\\ v_0 = 0 \text{ on } \Omega_e,\\ v_0 = 105\ V \text{ on } \Omega_{\text{electrode}},\\ \delta p_0 = 0 \text{ on } \Omega_a\end{array}\right. \tag{2.109}$$

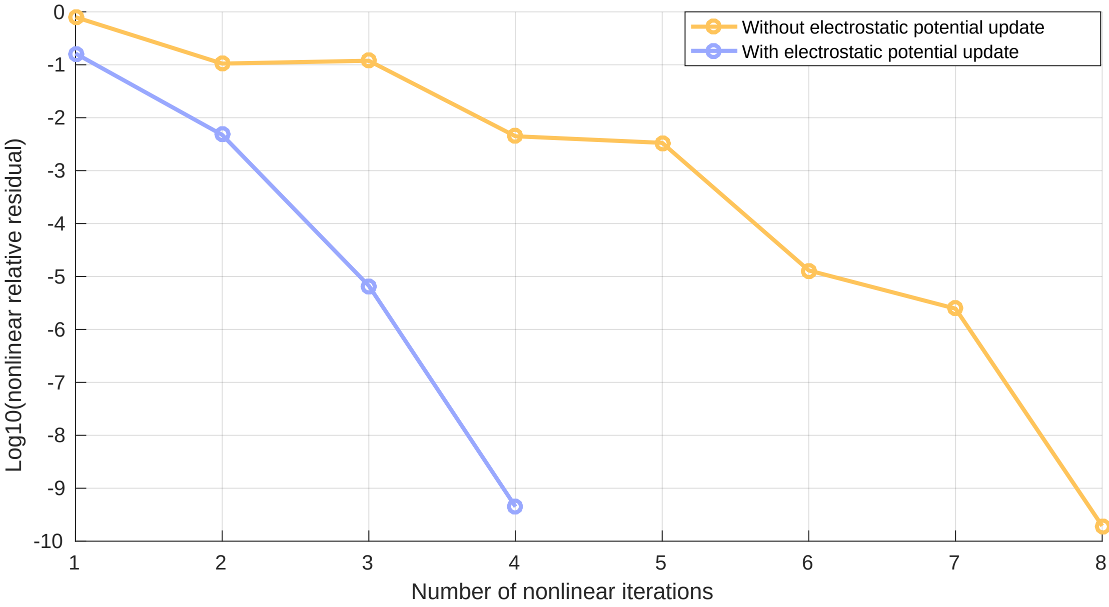


Figure 2.5: Nonlinear relative residual versus number of iterations for Newton's iteration close to the pull-in voltage when the mesh is smoothed at every nonlinear iteration with or without subsequent electrostatic field update.

brings the CMUT membrane beyond pull-in at the first iteration even though in the actual solution it is not. This happens because the electric field computed with the finite element method will be huge right below the electrode, leading to too high electric forces. The following initial guess is used throughout the thesis: $\boldsymbol{u_0} = \boldsymbol{0}$ on Ω_m and v_0 is the solution of (2.6) when the electric potential is fixed to the electrical excitation on the electrode and to 0 on the electric ground. The pressure δp_0 is initially set to zero on its domain. This leads to more realistic electric forces and to a smooth convergence.

2.4.4 Diagonal scaling

It is known [89] that the monolithic coupling of electromechanical formulations can have a bad conditionning. This is not hard to imagine given the huge difference between the electric permittivity of the order of 10^{-12} and Young's modulus here of the order 10^{10}. The eigenvalues of the electric system are much smaller than the mechanical ones as illustrated in figure 2.6 (two top plots). This leads to a very large condition number.

In order to improve the conditionning a diagonal scaling is used (in addition to (2.105) and (2.106) for electroelastoacoustic problems). The diagonal scaling for a general algebraic matrix $\boldsymbol{Ax} = \boldsymbol{b}$ is as follows:

Define a diagonal matrix $\boldsymbol{P}$ which is the inverse of the square root of the diagonal of $\boldsymbol{A}$:

$$P_{ij} = \frac{1}{\sqrt{|A_{ij}|}} \delta_{ij}, \tag{2.110}$$

where δ_{ij} is the Kronencker delta. Defining

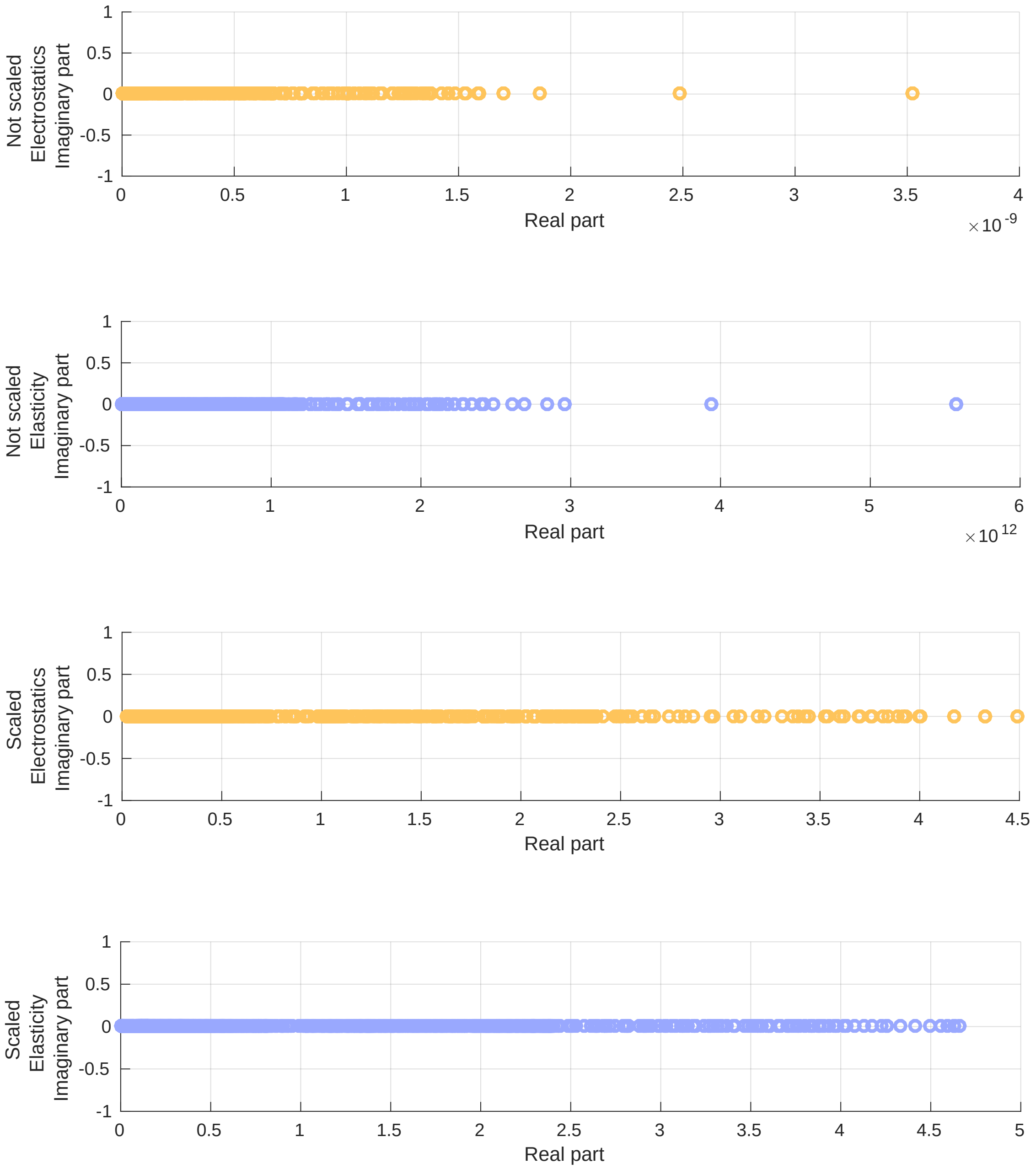


Figure 2.6: All eigenvalues for the elasticity (2.16) and electrostatic (2.6) formulations without (two top) and with (two bottom) diagonal scaling.

$$\begin{aligned} \tilde{\boldsymbol{A}} &= \boldsymbol{PAP}, \\ \tilde{\boldsymbol{b}} &= \boldsymbol{Pb}, \end{aligned} \tag{2.111}$$

one solves for $\tilde{\boldsymbol{x}}$: $\tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{b}}$ and the solution vector $\boldsymbol{x}$ is $\boldsymbol{x} = \boldsymbol{P}\tilde{\boldsymbol{x}}$. With the diagonal scaling the $\boldsymbol{A}$ matrix to invert is replaced by a better conditionned scaled matrix $\tilde{\boldsymbol{A}}$ whose elements are

$$\tilde{A}_{ij} = \frac{1}{\sqrt{|A_{ii}|}} A_{ij} \frac{1}{\sqrt{|A_{jj}|}}, \tag{2.112}$$

which brings the electrostatic and elastodynamic stiffnesses closer to each other. This results in more comparable eigenvalues as can be seen in figure 2.6 (two bottom plots).

2.5 Optimal mesh size and interpolation order

Creating an optimised mesh is fundamental in the finite element method to get an accurate solution at the lowest possible computational cost. As an example on the reference CMUT model in figure 1.5 it does not make sense to mesh the vacuum gap with a very fine mesh if the membrane is not finely meshed as well. Selecting the appropriate interpolation order in the finite element discretisation of the fields (refer to section 2.3.1) is also of crucial importance to keep the size of the algebraic systems to solve low while still getting an accurate solution. This section gives an idea of the best combination of the interpolation order and mesh size parameters for a realistic membrane deflection shape. Results are only shown for the electroelastic problem as the fluid region in this thesis is obtained for simplicity via an extrusion of the CMUT membrane and can thus hardly lead to an optimal mesh. Because the mesh used is a structured quadrangular mesh in 2D, structured hexahedral mesh in 3D, there are three parameters to tweak: the number of mesh layers vertically in the vacuum gap, vertically in the membrane and horizontally. The mesh of the reference CMUT in figure 1.5 is optimised for an electroelastic (2.6)-(2.16) problem with a time-independent electrical excitation voltage of 10 V. The Matlab code used imposes same interpolation orders for the mechanical displacement $\boldsymbol{u}$ and for the electric potential v.

Figure 2.7 shows the mechanical displacement $\boldsymbol{u}$ versus interpolation order (1 through 6) for a coarse horizontal mesh with only 10 element layers on the membrane (uniform divisions are used). As can be seen an order 1 interpolation leads to a very stiff membrane whose deformation is far below the actual deformation while order 2 provides a much better approximation. For order 3 and beyond the deformations are accurate and the deflections are visually the same. Table 2.1 shows the convergence to the actual maximum mechanical displacement (0.812 nm) as the interpolation order and the mesh density are increased. The top table shows a sweep of the number of vertical element layers in the vacuum, the middle table shows a sweep of the number of vertical element layers in the membrane while the bottom table shows a sweep of the number of horizontal element layers. In any case the sweep is on a single mesh parameter while the two other parameters are chosen fine enough to remove their influence.

As can be seen on table 2.1 a good compromise between accuracy and computational cost for any interpolation order seems to be to mesh the vacuum gap with a single vertical element layer. The membrane seems however to be best meshed with 2 or even 3 order 1 element layers vertically

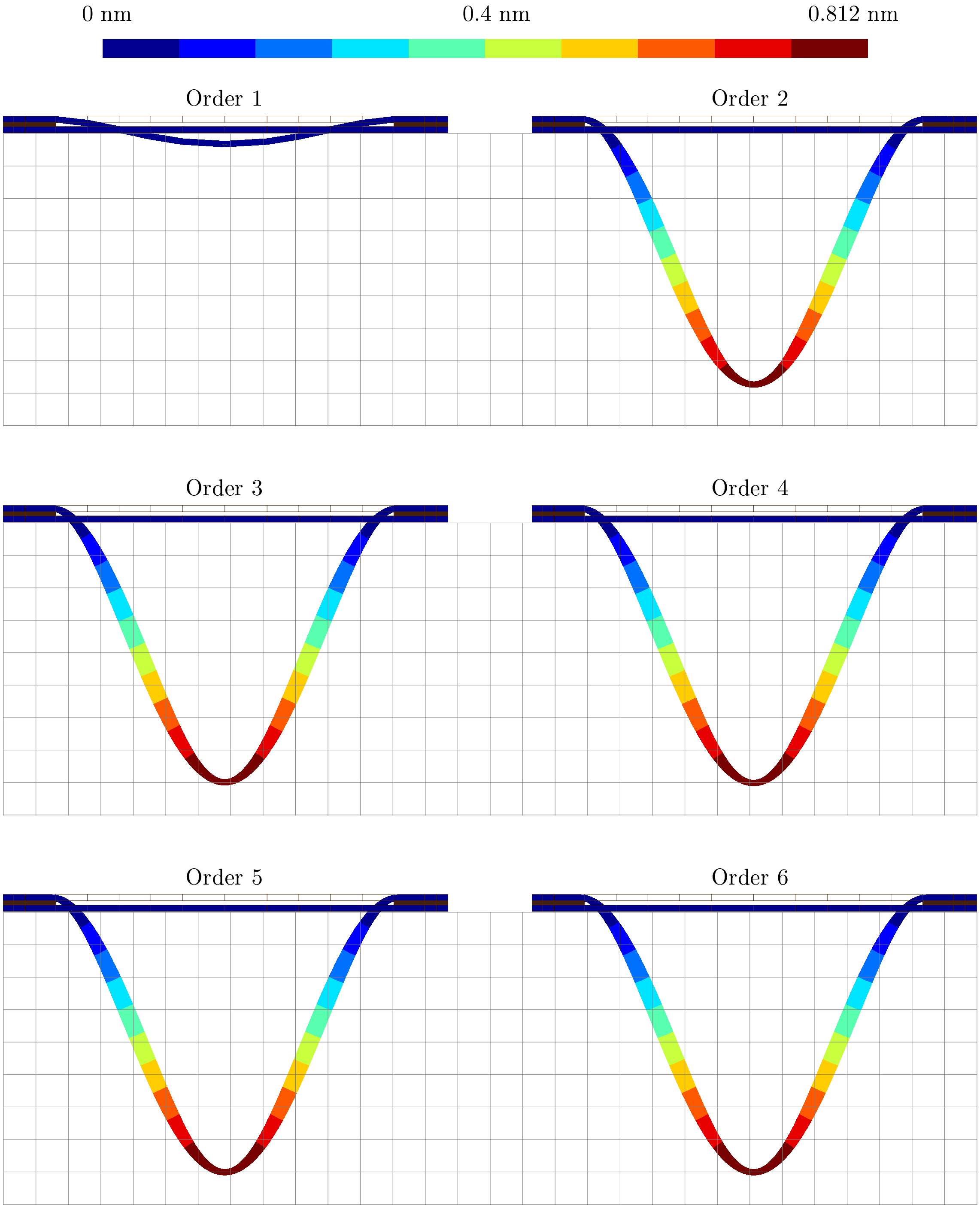


Figure 2.7: Exagerated membrane displacement for the reference CMUT illustrated in figure 1.5 versus interpolation order (1 through 6) for a coarse horizontal mesh (10 element layers horizontally on the membrane). The electroelastic problem (2.6) - (2.16) is solved for a time-independent electrode-to-ground electrical excitation.

Table 2.1: For the electroelastic problem (2.6)-(2.16) on the reference CMUT illustrated in figure 1.5 the tables show the maximum mechanical displacement $\boldsymbol{u}$ nm versus interpolation order and versus number of vertical element layers in the vacuum gap (top), number of vertical element layers in the membrane (middle), number of horizontal element layers (bottom). Actual maximum displacement is 0.812 nm.

# vertical vacuum layers	**1**	**2**	**3**
Order 1	0.805	0.809	0.810
Order 2	0.810	0.812	0.812
Order 3	0.811	0.812	0.812

# vertical membrane layers	**1**	**2**	**3**	**4**	**5**
Order 1	0.746	0.794	0.804	0.808	0.809
Order 2	0.811	0.812	0.812	0.812	
Order 3	0.812	0.812	0.812		

# horizontal layers	**3**	**10**	**42**	**63**	**120**	**240**	**480**	**960**
Order 1	0.021	0.073	0.502	0.636	0.762	0.798	0.807	0.810
Order 2	0.381	0.787	0.809	0.810				
Order 3	0.726	0.809	0.812					
Order 4	0.713	0.812						

while a single layer is here again enough for higher order elements. As for the best number of horizontal element layers on the membrane, as few as 3 elements with order 3 interpolation can be used if an up to 10% error is acceptable. A much better approximation can however be obtained with 10 order 3 element layers horizontally. For order 2 a similar accuracy is obtained with about 42 elements while up to a thousand are required for a first order interpolation: this amounts to roughly a factor 20 increase in the number of horizontal layers.

The analysis done in this section still makes sense for dynamical excitations as long as only the first membrane vibration mode is excited.

2.6 Mesh used for the reference geometry

For the reference geometry of figure 1.5 the 2D meshed fluid region on top of the membranes is box-shaped and illustrated (not to scale) on figure 2.8. It is truncated horizontally at 4 membrane length, i.e. 200 μm on the left and right side of the 2D membrane array and at 10 membrane length, i.e. 500 μm from the membrane top in the vertical direction. The mesh is structured, made of quadrangles. The mesh of the fluid region that lies above the membranes is extruded from the membrane mesh with 80 layers. The membrane, the vacuum gap and the bulk all have 2 vertical mesh element layers. The support pillars have 4 horizontal element layers while the membrane has

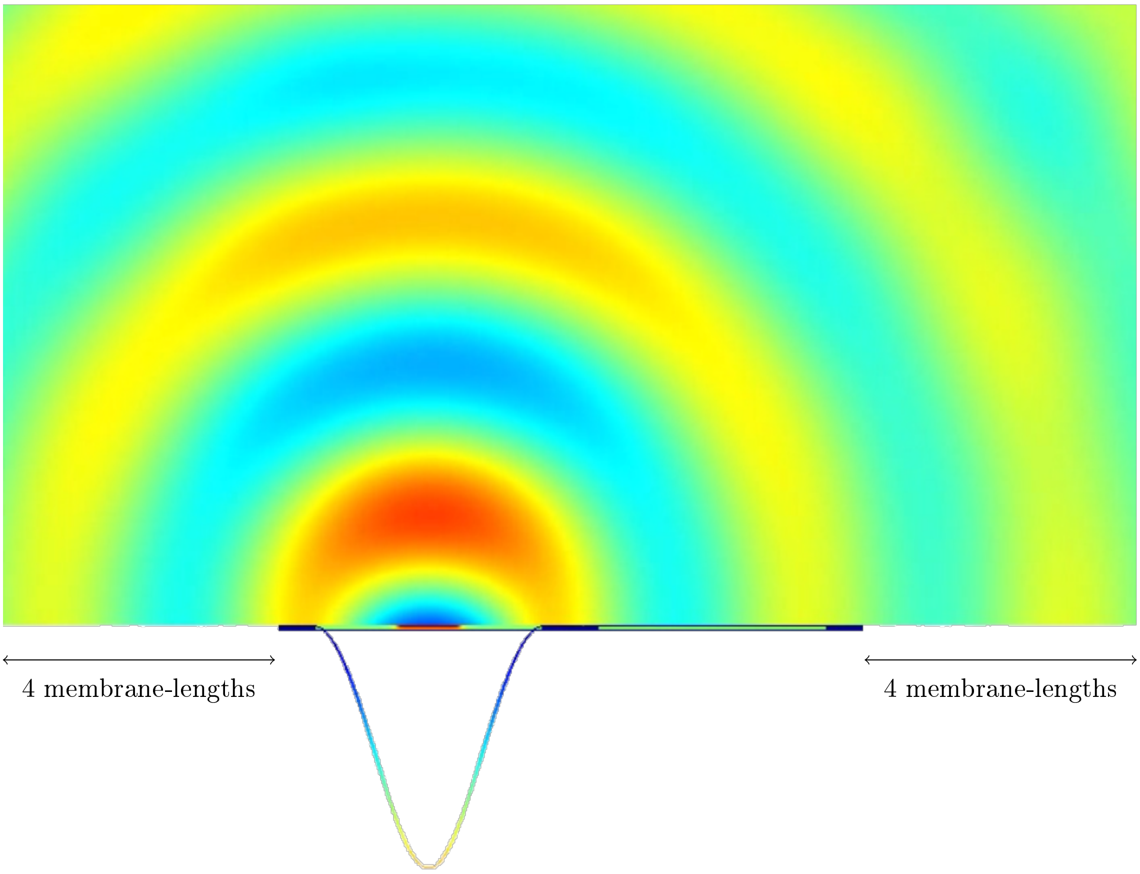


Figure 2.8: Illustration of the displacement field (exagerated), the corresponding acoustic pressure field in the fluid and the electric potential field for an array of two reference membranes detailed on figure 1.5. The fluid box is not to scale. Its height is 10 membrane-lengths.

23. The fluid boxes surrounding the membrane on the left and right side both have 65 horizontal element layers. The total number of quadrangles in the mesh is 12735.

When using order 3 interpolations on the proposed mesh to get the maximum mechanical displacement versus frequency plot in the settings used in figure 1.8 one gets an excellent accuracy both because the mesh is dense enough and because the fluid box size is large enough to decrease sufficiently the fluid truncation effects: for frequencies ranging from 100 kHz to 4 MHz the maximum displacement of the constant deflection, the in-phase vibration and the quadrature vibration all have an accuracy better than 1%. Furthermore the first resonance peak only moves by 2 kHz, i.e. 0.1% when doubling the fluid box size and the mesh density in each direction.

Chapter 3

Steady state time resolution

This chapter describes an automatic multiharmonic resolution method to get the steady state solution in time and compares it to the classical Newmark time stepping method. After a review of the Newmark method the multiharmonic resolution method is first detailed on a 1D electrostatic problem on a vibrating mesh. Its application to the simulation of CMUTs is then described. Finally the suitability of both methods is assessed for the reference 2D CMUT model.

3.1 Steady state analysis

To define and illustrate what steady state is consider the CMUT lumped model of figure 1.3. In order to predict the dynamic deflection the mass m (kg) of the membrane has to be taken into account. Additionally a damping term has to be added to take into account the power radiated by the emitted pressure waves and any other source of losses: for simplicity a damping proportional to the membrane speed is used in the following illustrations. As a result a damped spring-mass system is obtained (figure 3.1). The equivalent mass can be computed with [18]: $m = 1.84\,\pi\, a^2\, h\, \rho_{Si} = 4 \cdot 10^{-12}$ kg.

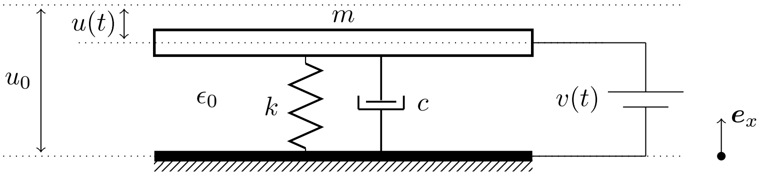


Figure 3.1: Lumped model of a membrane electrically actuated by a time-dependent voltage $v(t)$.

Adding the inertia forces $m\frac{\partial^2 u}{\partial t^2}$ and damping forces $c\frac{\partial u}{\partial t}$ to (1.21) gives the updated equilibrium equation

$$-\,k\,u - \frac{1}{2}\epsilon_0 \frac{v^2}{(u_0+u)^2} A - c\frac{\partial u}{\partial t} - m\frac{\partial^2 u}{\partial t^2} = 0, \tag{3.1}$$

where the damping coefficient $c = 6 \cdot 10^{-6}$ (Ns/m) has been exagerated for illustration.

When a periodic electrical excitation is applied to an initially unexcited CMUT membrane at

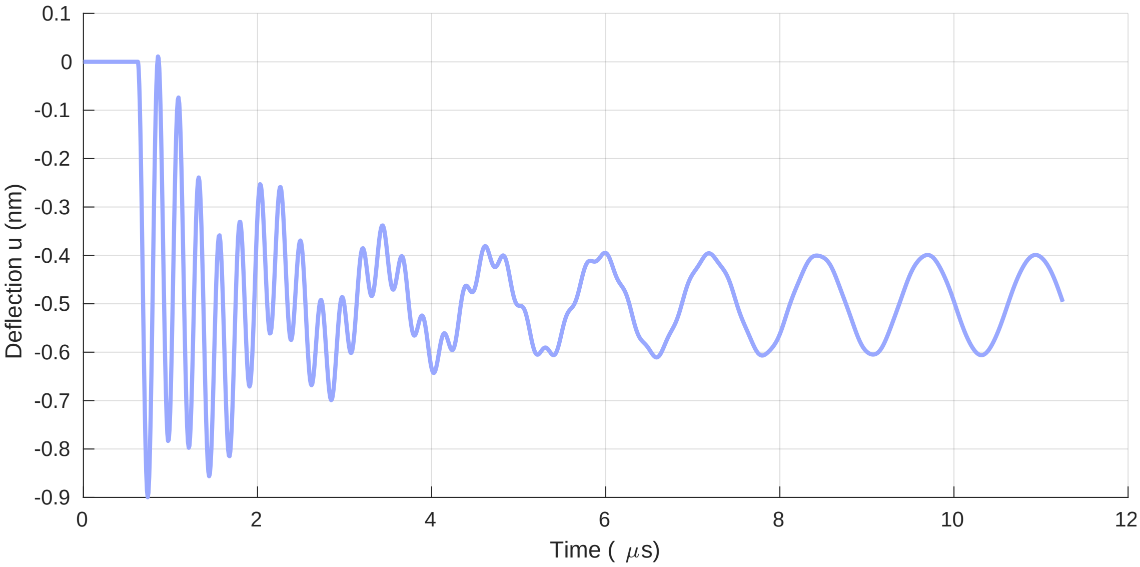


Figure 3.2: Membrane deflection $u(t)$ versus time for an electrical excitation $v(t) = 10 + 1 \cdot sin(2\pi \cdot 800000t)$ V applied to the system at rest.

rest (figure 3.2) the membrane first vibrates in an aperiodic fashion before the vibration is damped enough and settles to a periodic waveform: the *steady state*. The aperiodic signal is the *transient* state. For CMUTs it can last much longer than illustrated. As can be seen in the figure the transient state can overshoot, reaching deflections that are not reached in steady state.

In order to evaluate the crosstalk deteriorating the imaging performance of a CMUT array one may electrically excite a given membrane in the array and compute the perturbation induced on the other membranes. The electrical excitation can be a pulse with a high spectral content or can be made up of a single or a few harmonics to catch the behaviour at or around a specific frequency f_0. In the latter case, adopted in this work, the periodic steady state solution is of interest rather than the non-periodic transient solution.

The following two sections propose methods to get the steady state solution. The first method does it without taking advantage of the periodic behaviour while the second one does.

3.2 Newmark's time stepping method

In this work second order time dependant systems will be obtained:

$$\boldsymbol{M}\frac{\partial^2 \phi(\boldsymbol{x},t)}{\partial t^2} + \boldsymbol{C}\frac{\partial \phi(\boldsymbol{x},t)}{\partial t} + \boldsymbol{K}\phi(\boldsymbol{x},t) = f, \tag{3.2}$$

where $\boldsymbol{M}$ is the mass matrix containing the inertia terms, $\boldsymbol{C}$ the damping matrix, $\boldsymbol{K}$ the stiffness matrix and f the external sources. For a nonlinear problem $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ and f are functions of space and time. The mass matrix $\boldsymbol{M}$ for the time-dependent elastodynamic problem was obtained in (2.82), the stiffness matrix $\boldsymbol{K}$ was obtained in (2.81) and the forces in (2.83). For the time-dependent acoustic problem $\boldsymbol{M}$ is obtained from (2.93), $\boldsymbol{C}$ from(2.92) and $\boldsymbol{K}$ from (2.91).

Table 3.1: Typical parameter pairs (γ, β) for the Newmark algorithm

γ	β	**Algorithm name**	**Convergence order**	**Stability**
0	0	Explicit	2	unstable
1/2	0	Centered difference	2	cond. stable
1/2	1/12	Fox & Goodwin	3	cond. stable
1/2	1/6	Linear acceleration	2	cond. stable
1/2	1/4	Constant acceleration	2	uncond. stable

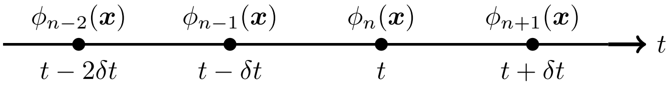

$\phi_{n-2}(\boldsymbol{x})$
$\phi_{n-1}(\boldsymbol{x})$
$\phi_{n}(\boldsymbol{x})$
$\phi_{n+1}(\boldsymbol{x})$
t
$t-2\delta t$
$t-\delta t$
t
$t+\delta t$

Figure 3.3: Solution field ϕ at every discrete time value.

System (3.2) can be solved very generally (even for transients) by advancing from the solution ϕ_n at one time step to the next one ϕ_{n+1} (as illustrated in figure 3.3) with a *time stepping* method such as Newmark's method [90, 91]:

$$\begin{aligned}(\boldsymbol{M} + \gamma\delta t\boldsymbol{C} + \beta\delta t^2\boldsymbol{K})\phi_{n+1}(\boldsymbol{x}) = {} & (2\boldsymbol{M} - (1-2\gamma)\,\delta t\boldsymbol{C} - (\frac{1}{2} + \gamma - 2\beta)\,\delta t^2\boldsymbol{K})\phi_n(\boldsymbol{x}) \\ & + (-\boldsymbol{M} - (\gamma - 1)\,\delta t\boldsymbol{C} - (\frac{1}{2} - \gamma + \beta)\,\delta t^2\boldsymbol{K})\phi_{n-1}(\boldsymbol{x}) \\ & + \delta t^2(\beta f_{n+1} + (\frac{1}{2} + \gamma - 2\beta)f_n + (\frac{1}{2} - \gamma + \beta)f_{n-1}),\end{aligned}$$

with all zero initial conditions used in this work. Variable δt (s) is the time step in the time-discretisation (supposed constant in this work): a solution is computed at every time step δt. For an accurate solution it should be small enough. Variables ϕ_{n+1}, ϕ_n and ϕ_{n-1} are the unknown fields at the next, the current and the previous time steps respectively. Parameters γ and β can be used to tune the properties of Newmark's method [92].

Table 3.1 lists typical choices of parameters (γ, β). Choosing $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ supposes that the acceleration is constant during each time step δt. This set of parameters leads to an unconditionally stable algorithm, i.e. the error at a given instant t on the solution ϕ will be damped in the next time steps, no matter how large the time step δt is [93]. For all other conditionally stable algorithms, stability is not unconditional, it is only obtained with conditions on δt. Furthermore it is worth noting that the choice $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ introduces in theory no error on the solution magnitude, only on its phase.

3.3 An automatic multiharmonic resolution

In section 3.2 a method was detailed to compute the solution in time of general problems, including their steady state solution. Unfortunately with this method the transient state has to be first time stepped through before the steady state is obtained, approximately. For linear problems with a single or a few excitation frequencies a harmonic approach can be preferred: because the steady state is periodic in time with a period $\frac{1}{f_0}$ all fields can be written in terms of their Fourier series

$$\phi(\boldsymbol{x},t) = \sum_{k=0}^{\infty} \phi_{sk}(\boldsymbol{x}) \, \sin(\omega_k t) + \phi_{ck}(\boldsymbol{x}) \, \cos(\omega_k t), \tag{3.3}$$

where $\omega_k = 2\pi k f_0$, or as used in practice in terms of complex valued phasors $\overline{\phi}(\boldsymbol{x})$

$$\phi(\boldsymbol{x},t) = \sum_{k=0}^{\infty} \Re\left(\overline{\phi}_k(\boldsymbol{x}) \, e^{i\,\omega_k t}\right), \tag{3.4}$$

where $\overline{\phi}_k(\boldsymbol{x}) = \phi_{sk}(\boldsymbol{x}) - i\,\phi_{ck}(\boldsymbol{x})$ and $i = \sqrt{-1}$. Due to linearity all frequencies are uncoupled so that phasor field $\overline{\phi}_k$ (or alternatively fields $\phi_{sk}(\boldsymbol{x})$ and $\phi_{ck}(\boldsymbol{x})$) can be computed for every index k separately by using the harmonic equivalent of (3.2)

$$-\,\omega_k^2 \, \boldsymbol{M} \, \overline{\phi}_k(\boldsymbol{x}) + i\omega_k \, \boldsymbol{C} \, \overline{\phi}_k(\boldsymbol{x}) + \boldsymbol{K} \, \overline{\phi}_k(\boldsymbol{x}) = f, \tag{3.5}$$

for which the time-dependency is removed. Phasor field $\overline{\phi}_k(\boldsymbol{x})$ is then obtained by solving

$$(-\omega_k^2 \, \boldsymbol{M} + i\omega_k \, \boldsymbol{C} + \boldsymbol{K}) \, \overline{\phi}_k(\boldsymbol{x}) = f. \tag{3.6}$$

The harmonic approach just described can be readily used to compute the vibration of the reference CMUT of figure 1.5 for an electrical excitation $v(t) = V + \delta v(t)$ (V) with $\delta v \lll V$ since this corresponds to a linear vibration around a static deflection. For illustration consider the lumped model in (3.1): the only nonlinear term $\frac{v^2}{(u_0+u)^2}$ can be linearised with an excitation $v(t) = V + \delta v(t)$ as

$$\frac{v^2}{(u_0+u)^2} = \frac{(V+\delta v)^2}{(u_0+U+\delta u)^2} \approx \frac{V^2 + 2V\,\delta v}{(u_0+U)^2} \tag{3.7}$$

if $\delta v \lll V$ and thus $\delta u \lll U$ and $\delta u \lll u_0$ as well.

The actual nonlinear equation (3.1) was solved for an electrical excitation $v(t) = 10 + 1 \cdot \sin(2\pi\,800000t)$ V. The steady state deflection u versus time is plotted in figure 3.4 (top) along with its Fourier series computed on a single period (bottom). As expected, for the tiny 1 V alternating voltage added to the 10 V bias only the excitation frequencies appear on the Fourier plot, the other ones are much smaller: the vibration is linear. It is worth noting that the constant harmonic is 5 times stronger than the vibration harmonic which is in good agreement with the linear approximation (3.7). When the excitation voltage is $v(t) = 10 + 10 \cdot \sin(2\pi\,800000t)$ or even higher with $v(t) = 60 + 60 \cdot \sin(2\pi\,800000t)$ V however the vibration clearly becomes nonlinear as shown in figure 3.5 and 3.6. Because of the nonlinearity the frequencies in the Fourier series of the mechanical displacement and electric potential are coupled and new harmonics appear.

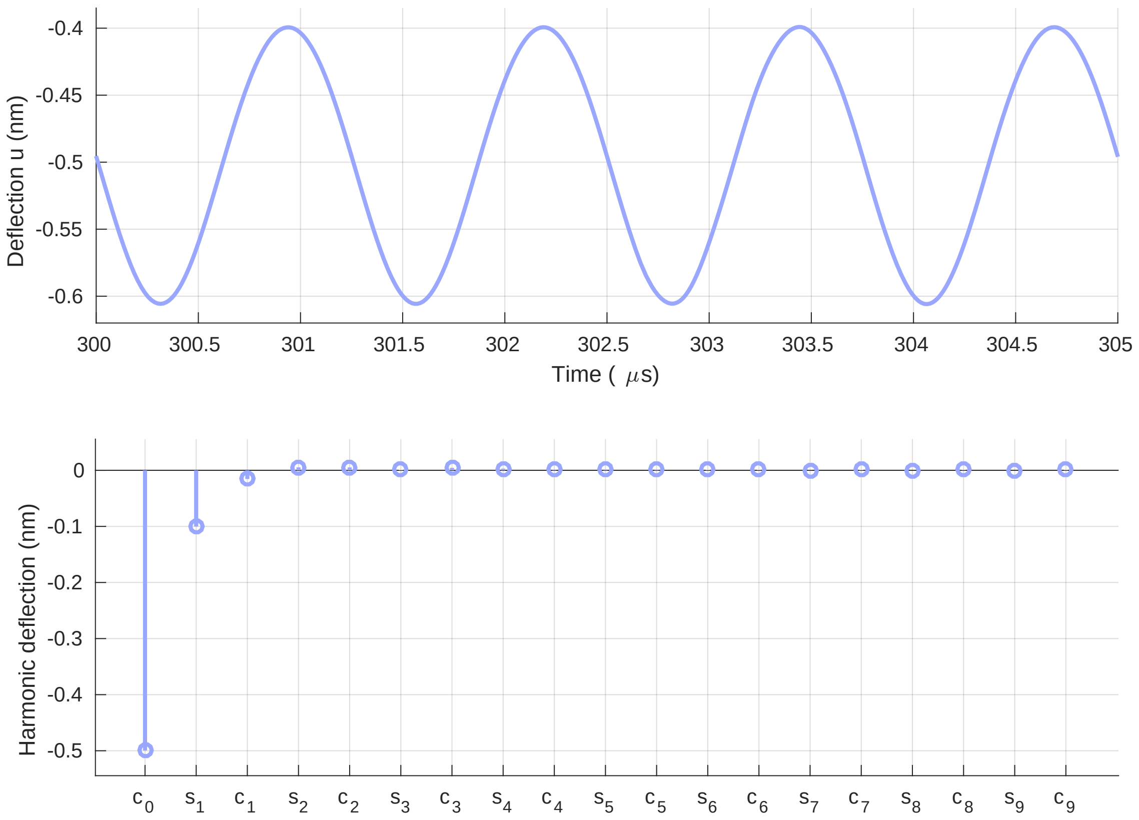


Figure 3.4: Deflection $u(t)$ versus time (top) and corresponding Fourier series (bottom) for an electrical excitation $v(t) = 10 + 1 \cdot \sin(2\pi \cdot 800000t)$ V in the lumped model (3.1).

As has been observed in figure 3.6 the harmonic approach described above can not be used for nonlinear problems anymore: instead the more general *multiharmonic* or *harmonic balance* method can be used. In this method the fields are still decomposed with a Fourier series as done in (3.3) but the harmonics considered are not anymore only those of the excitation field since new harmonics can appear. A field is instead approximated by a truncated Fourier series including as many as possible of the dominant harmonics:

$$\phi(\boldsymbol{x}, t) = \sum_{k=0}^{N} \phi_{sk}(\boldsymbol{x}) \, \sin(\omega_k t) + \phi_{ck}(\boldsymbol{x}) \, \cos(\omega_k t). \tag{3.8}$$

For the lumped model (3.1) with $v(t) = 10 + 10 \cdot \sin(2\pi \, 800000t)$ V it can be seen from figure 3.5 that a good approximation of the deflection u can be obtained with

$$u(t) = u_{c0} + u_{s1} \, \sin(\omega_1 t) + u_{c2} \, \cos(\omega_2 t). \tag{3.9}$$

where $\omega_1 = 2\pi \cdot 800000t$ and $\omega_2 = 2 \cdot 2\pi \cdot 800000t$. Moreover since the largest u harmonic deflection is 1 nm, much smaller than u_0, relation (3.1) can be simplified to

$$-\, k \, u - \frac{1}{2} \epsilon_0 \frac{v^2}{u_0^2} A - c \frac{\partial u}{\partial t} - m \frac{\partial^2 u}{\partial t^2} \approx 0, \tag{3.10}$$

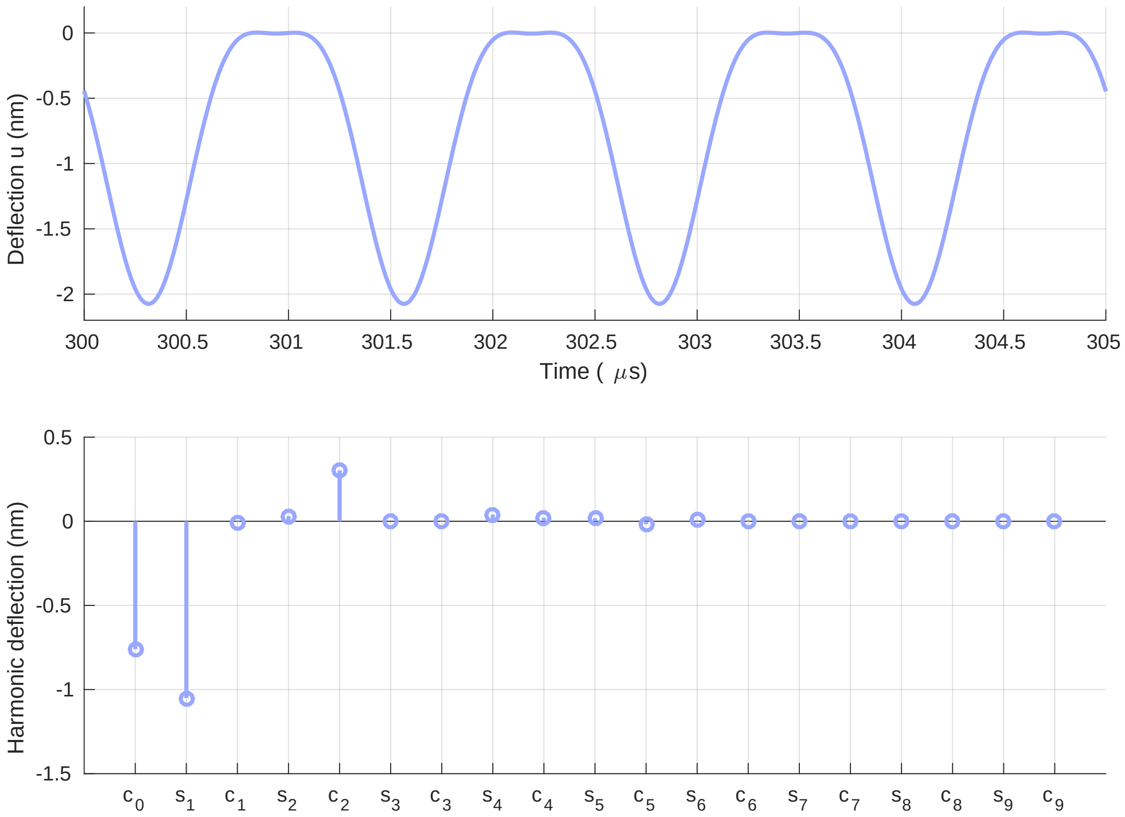


Figure 3.5: Deflection $u(t)$ versus time (top) and corresponding Fourier series (bottom) for an electrical excitation $v(t) = 10 + 10 \cdot \sin(2\pi \cdot 800000t)$ V in the lumped model (3.1).

In the multiharmonic method, to obtain the three unknown harmonic coefficients u_{c0}, u_{s1} and u_{c2} the field u in (3.10) is replaced by its Fourier truncation:

$$\begin{aligned} &-k\,u_{c0} - k\,u_{s1}\,\sin(\omega_1 t) - k\,u_{c2}\,\cos(\omega_2 t) - \tfrac{100}{2}\,\epsilon_0\,\tfrac{1+2\sin(\omega_1 t)+\sin^2(\omega_1 t)}{u_0^2}\,A \\ &-c\,\omega_1\,u_{s1}\,\cos(\omega_1 t) + c\,\omega_2\,u_{c2}\,\sin(\omega_2 t) + m\,\omega_1^2\,u_{s1}\,\sin(\omega_1 t) + m\,\omega_2^2\,u_{c2}\,\cos(\omega_2 t) = 0, \end{aligned} \tag{3.11}$$

a simple form that could only been obtained because in (3.10) the impact of u on the electrode to ground distance is neglected. For higher excitation voltages this does not hold anymore and section 3.3.1 will present a technique to deal with the full $\frac{1}{(u_0+u)^2}$ term.

Using the trigonometry identity $\sin^2(\alpha) = \frac{1-\cos(2\alpha)}{2}$ to rewrite (3.11) gives:

$$\begin{aligned} &-k\,u_{c0} - k\,u_{s1}\,\sin(\omega_1 t) - k\,u_{c2}\,\cos(\omega_2 t) - 50\,\epsilon_0\,\tfrac{1}{u_0^2}A - 50\,\epsilon_0\,\tfrac{2\sin(\omega_1 t)}{u_0^2}A - 50\,\epsilon_0\,\tfrac{1-cos(\omega_2 t)}{2u_0^2}A \\ &-c\,\omega_1\,u_{s1}\,\cos(\omega_1 t) + c\,\omega_2\,u_{c2}\,\sin(\omega_2 t) + m\,\omega_1^2\,u_{s1}\,\sin(\omega_1 t) + m\,\omega_2^2\,u_{c2}\,\cos(\omega_2 t) = 0. \end{aligned} \tag{3.12}$$

Since (3.12) is valid for any time t one can extract an equation for every sine and cosine term.

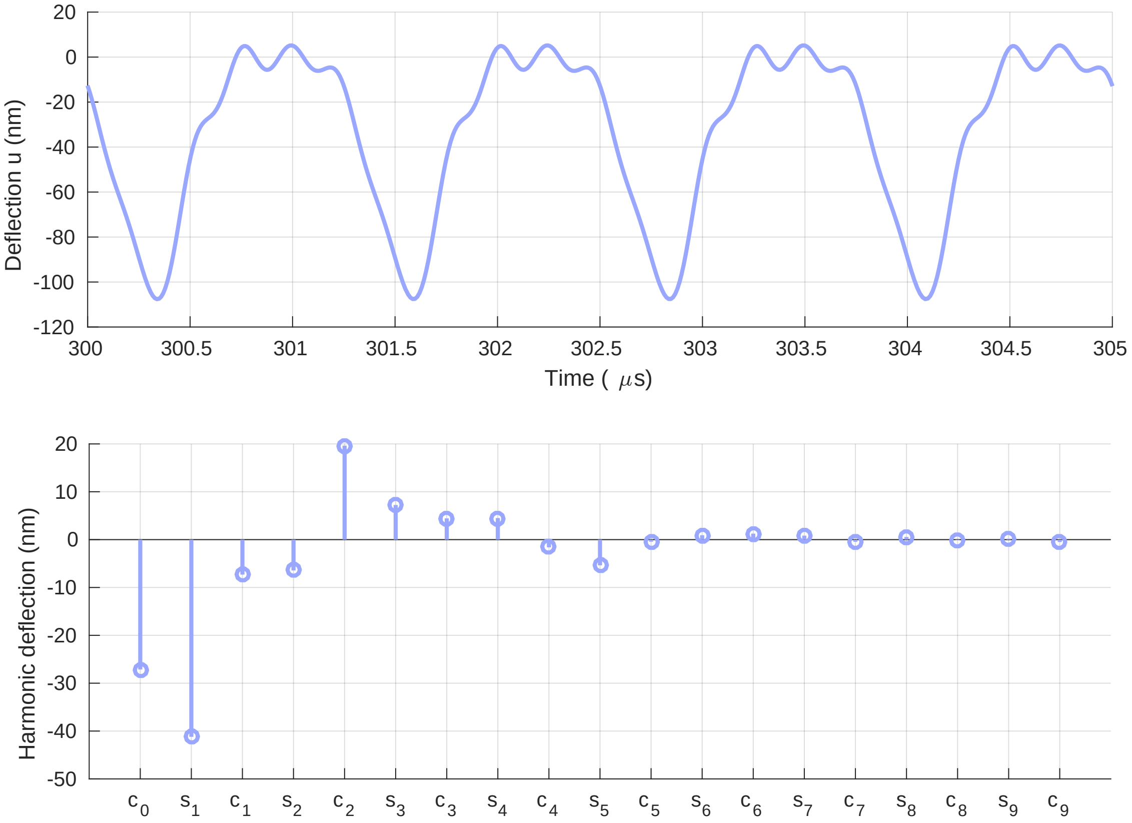


Figure 3.6: Deflection $u(t)$ versus time (top) and corresponding Fourier series (bottom) for an electrical excitation $v(t) = 60 + 60 \cdot \sin(2\pi \cdot 800000t)$ V in the lumped model (3.1).

Keeping only the equations corresponding to the Fourier truncation of u, that is the equation for the constant terms, the $\sin(\omega_1 t)$ terms and the $\cos(\omega_2 t)$ terms gives

$$\begin{cases} -\,k\,u_{c0} - 50\,\epsilon_0 \dfrac{1}{u_0^2} A - 50\epsilon_0 \dfrac{1}{2u_0^2} A = 0, \\ -\,k\,u_{s1} - 50\,\epsilon_0 \dfrac{2}{u_0^2} A + m\,\omega_1^2\,u_{s1} = 0 \\ -\,k\,u_{c2} + 50\,\epsilon_0 \dfrac{1}{2u_0^2} A + m\,\omega_2^2\,u_{c2} = 0, \end{cases} \tag{3.13}$$

i.e. three equations for the three unknown Fourier coefficients. The coefficients can finally be computed:

$$\begin{cases} u_{c0} = -0.74 \text{ nm}, \\ u_{s1} = -1.03 \text{ nm} \\ u_{c2} = \ \ 0.29 \text{ nm}, \end{cases} \tag{3.14}$$

which matches closely the coefficients obtained in figure 3.5.

In the hypothesis of $u \lll u_0$, i.e. for low excitation voltages the equations (3.13) giving the Fourier coefficients of the deflection u do not exhibit coupled harmonics. Once the hypothesis is removed (as will be seen in section 3.3.1) however all harmonics become coupled. For $2N+1$ harmonics the system to solve is thus $2N+1$ times larger.

The multiharmonic method has already been investigated in several fields [94, 95, 96, 97, 98, 99, 100]. Its effective use on large scale applications is however impeded by two main factors. On the one hand the derivation of the equation terms in the multiharmonic formulation (even for a number of harmonics considered $2N+1$ equal 3) can become extremely tedious when done manually. On the other hand the size of the nonlinear system is multiplied by $2N+1$ compared to the time-domain approach. The first issue is adressed by using an automatic implementation of [101, 102] using symbolic computation. Concerning the second issue it has been shown in [103] that the convergence of the Fourier approximation is generally of order N^{-1} but can be much faster for simple harmonic excitations [94, 104].

3.3.1 Application to 1D electrostatics on a vibrating mesh

Electrostatic formulation on the undeformed mesh

Let us consider a 1D electrostatic problem solved in terms of the electrostatic potential v on a multiharmonically vibrating mesh, deformed by the mechanical displacement $\boldsymbol{u}$. The system is excited on an electrode via a time-harmonic Dirichlet boundary condition on the electric potential: $\overline{v} = V_1 \sin(2\pi f_0 t)$. A reference potential of 0 is imposed on the ground. Let us assume that the electric potential solution $v(\boldsymbol{x}, t)$ and the mechanical displacement $\boldsymbol{u}(\boldsymbol{x}.t)$ can be approximated by the truncated Fourier series $v(\boldsymbol{x}, t) = V_{s1}(\boldsymbol{x}) \sin(2\pi f_0 t) + V_{s3}(\boldsymbol{x}) \sin(3 \cdot 2\pi f_0 t)$ and $\boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{U}_{c0}(\boldsymbol{x}) + \boldsymbol{U}_{c2}(\boldsymbol{x}) \cos(2 \cdot 2\pi f_0 t)$, i.e. considering the second and sixth harmonic for the electric potential and the constant term and fifth harmonic for the displacement. While this limited expansion is chosen for the simplicity of the following analytic calculations, it already leads in practice to very accurate numerical results. The goal of the multiharmonic resolution is to find the Fourier coefficients $V_{s1}(\boldsymbol{x})$, $V_{s3}(\boldsymbol{x})$, $\boldsymbol{U}_{c0}(\boldsymbol{x})$ and $\boldsymbol{U}_{c2}(\boldsymbol{x})$.

Since the mesh deformation $\boldsymbol{u}$ is decomposed as a sum of harmonics, integration on the mesh deformed by $\boldsymbol{u}$ must be handled carefully. All the quantities are brought back to the undeformed mesh [105], by introducing the change of variables for the 1D x coordinate $x^* = x + u_x$ with Jacobian $\boldsymbol{J}(\boldsymbol{x}, t)$. Denoting by Ω the undeformed configuration and by Ω^* the deformed one and using relations $d\Omega^* = |\boldsymbol{J}| d\Omega$ and $\nabla^* = \boldsymbol{J}^{-1} \nabla$ leads to the following weak formulation of the electrostatic problem: Find v in an appropriate function space such that

$$- \int_{\Omega^*} \epsilon \, (\nabla^* v)^T \nabla^* v' d\Omega^* = 0, \tag{3.15}$$

holds for all test functions v' with $v = \overline{v}$ on the electrode and 0 on the ground. On the undeformed mesh the formulation becomes

$$- \int_{\Omega} \epsilon \, (\nabla v)^T \boldsymbol{J}^{-T} \boldsymbol{J}^{-1} \nabla v' |\boldsymbol{J}| d\Omega, \tag{3.16}$$

with

$$\boldsymbol{J} = \frac{\partial x^*}{\partial x} = \left[1 + \frac{\partial u_x}{\partial x} \right], \quad \boldsymbol{J}^{-1} = \frac{\partial x}{\partial x^*} = \frac{1}{|\boldsymbol{J}|} \cdot [1],$$

and $|\boldsymbol{J}| = 1 + \frac{\partial u_x}{\partial x}$ in this one dimensional setting. Equation (3.16) can be rewritten as:

$$-\int_{\Omega} \epsilon \frac{\partial v}{\partial x} \frac{\partial v'}{\partial x} \frac{1}{|\boldsymbol{J}|} d\Omega = 0. \tag{3.17}$$

It should be noted that in 2D and 3D extra Jacobian terms appear. Some of those can be neglected in small displacement applications but not in the kind of vibrating micromembrane test cases we consider, where the displacements can be large compared to the overall geometrical dimension of the problem. Also note that because of the abrupt change of $\boldsymbol{J}$ and $|\boldsymbol{J}|$ between a solid material and a non-solid material like e.g. air, one should avoid any kind of averaging for the Jacobian at these interfaces.

Multiharmonic expansion

In order to obtain the final multiharmonic formulation, the non-polynomial factor $\boldsymbol{G}(\boldsymbol{x}, t) := \frac{1}{|\boldsymbol{J}|}$ is first computed in weak form: Find $\boldsymbol{G}$ such that

$$\int_{\Omega_e} \boldsymbol{G}|\boldsymbol{J}|\boldsymbol{G}' d\Omega = \int_{\Omega_e} \boldsymbol{G}' d\Omega, \tag{3.18}$$

holds for appropriate test functions $\boldsymbol{G}'$. $\boldsymbol{G}$ itself is computed using a multiharmonic resolution. In practice it is well approximated with the same Fourier coefficients as for the mechanical displacement: $\boldsymbol{G}(\boldsymbol{x}, t) = \boldsymbol{G}_{c0}(\boldsymbol{x}) + \boldsymbol{G}_{c2}(\boldsymbol{x}) \cos(2 \cdot 2\pi f_0 t)$.

The next step could then simply be, as done in section 3.3, to symbolically replace v and $\frac{1}{|\boldsymbol{J}|}$ (and $\boldsymbol{u}$ in 2D and 3D) by their truncated Fourier expansion and then expand the whole formulation and multiply the sines and cosines together using recursively the following four identities to leave only sines and cosines of degree one but at higher frequencies:

$$\begin{aligned}
\cos(a) \cdot \cos(b) &= \frac{\cos(a+b)}{2} + \frac{\cos(a-b)}{2}, \\
\sin(a) \cdot \sin(b) &= \frac{\cos(a-b)}{2} - \frac{\cos(a+b)}{2}, \\
\sin(a) \cdot \cos(b) &= \frac{\sin(a+b)}{2} + \frac{\sin(a-b)}{2}, \\
\cos(a) \cdot \sin(b) &= \frac{\sin(a+b)}{2} - \frac{\sin(a-b)}{2}.
\end{aligned} \tag{3.19}$$

Doing so at this step would however make the number of expanded symbolic terms increase dramatically. In 1D it can be shown that it increases as N^2 because of the product between G and $\frac{\partial v}{\partial x}$, while in 2D it increases as N^4. To limit the explosion of the number of terms one can multiharmonically precompute all products of terms that are known, i.e. all terms but the unknown and the test function. Alternatively the formulation can be rewritten as a sum of products of a coefficient multiplying the unknown term and the shape function term and the coefficient can be computed via an FFT with a tunable accuracy, i.e. with a tunable number of harmonics considered. This has the advantage to be systematic and general. Furthermore it removes the need to perform step (3.18). The Matlab `fft` function or the FFTW library [106] can be called for that purpose.

In the 1D case there is only the $\frac{1}{|J|}$ known term so that the number of symbolic terms appearing after expansion is limited. One can immediately move to the next steps:

- replace the multiplied known term and the unknown by their truncated Fourier series
- expand the formulation
- apply time derivatives to the sines and cosines
- use (3.19) to transform the sines and cosines powers and products into sums of higher frequency sines and cosines

This amounts to transforming (3.17) into:

$$\begin{aligned}
&-\int_\Omega \epsilon \frac{\partial v}{\partial x}\frac{\partial v'}{\partial x}\frac{1}{|J|}d\Omega = 0\\
\Leftrightarrow &-\int_\Omega \epsilon \frac{\partial(V_{s1}\sin(2\pi f_0 t) + V_{s3}\sin(3\cdot 2\pi f_0 t))}{\partial x}\frac{\partial v'}{\partial x}(G_{c0} + G_{c2}\ \cos(2\cdot 2\pi f_0 t))d\Omega = 0\\
\Leftrightarrow &-\int_\Omega \epsilon\,[G_{c0}\frac{\partial V_{s1}}{\partial x}\sin(2\pi f_0 t) + G_{c0}\frac{\partial V_{s3}}{\partial x}\sin(3\cdot 2\pi f_0 t) + G_{c2}\ \cos(2\cdot 2\pi f_0 t)\frac{\partial V_{s1}}{\partial x}\sin(2\pi f_0 t)\\
&+ G_{c2}\ \cos(2\cdot 2\pi f_0 t)\frac{\partial V_{s3}}{\partial x}\sin(3\cdot 2\pi f_0 t)]\frac{\partial v'}{\partial x}d\Omega = 0\\
\Leftrightarrow &-\int_\Omega \epsilon\,[(G_{c0}\frac{\partial V_{s1}}{\partial x} - \frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x} + \frac{1}{2}G_{c2}\frac{\partial V_{s3}}{\partial x})\ \sin(2\pi f_0 t) + (G_{c0}\frac{\partial V_{s3}}{\partial x} + \frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x})\ \sin(3\cdot 2\pi f_0 t)\\
&+ (\frac{1}{2}G_{c2}\frac{\partial V_{s3}}{\partial x})\ \sin(5\cdot 2\pi f_0 t)]\frac{\partial v'}{\partial x}d\Omega = 0,
\end{aligned} \tag{3.20}$$

which is valid for any time t and can thus be split into three independent equations, with the sine terms removed. Taking the equations corresponding to the Fourier expansion of v, i.e. the terms multipled by $\sin(2\pi f_0 t)$ and the ones multiplied by $\sin(3\cdot 2\pi f_0 t)$ gives an excellent approximation of the actual electrostatic formulation and leads to the final multiharmonic formulation: Find V_{s1} and V_{s3} such that

$$\begin{cases}
\displaystyle\int_\Omega \epsilon\,(G_{c0}\frac{\partial V_{s1}}{\partial x} - \frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x} + \frac{1}{2}G_{c2}\frac{\partial V_{s3}}{\partial x})\frac{\partial v'}{\partial x}d\Omega = 0,\\[2ex]
\displaystyle\int_\Omega \epsilon\,(G_{c0}\frac{\partial V_{s3}}{\partial x} + \frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x})\frac{\partial v'}{\partial x}d\Omega = 0,
\end{cases} \tag{3.21}$$

holds for appropriate test functions v'. This system can be rewritten in matrix form:

$$\begin{bmatrix} K_{V_{s1}V_{s1}} & K_{V_{s1}V_{s3}} \\ K_{V_{s3}V_{s1}} & K_{V_{s3}V_{s3}} \end{bmatrix}\begin{bmatrix} V_{s1}(\boldsymbol{x}) \\ V_{s3}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{3.22}$$

where each of the four blocks can be generated using a usual monoharmonic finite element assembler and where the right-hand-side incorporates the contribution of the nonhomogeneous Dirichlet boundary condition.

As can be seen, even for linear electrostatic problems the harmonics V_{s1} and V_{s3} can be coupled if the mesh is deformed. To understand that this makes sense simply consider a mechanical membrane vibrating harmonically as a sine wave and a constant applied electrostatic voltage between two electrodes. Even though the electrostatic voltage on the electrode is constant the voltage inside the membrane will vary with time and thus the overall voltage will have a constant component plus a harmonic component. In case the membrane displacement is a constant or simply zero then the electric potential harmonics are uncoupled and the off-diagonal blocks $K_{V_{s1}V_{s3}}$and $K_{V_{s3}V_{s1}}$ are zero as can be seen in (3.21) when G_{c2} is set to zero.

Implementation aspects

When implementing the multiharmonic method in an already existing finite element assembler software the focus should be on reusing as much as possible what already exists and has been validated, and modifying as little as possible of the software. With that in mind, using the excplicit symbolic computation as throughout section 3.3.1 is desirable since it can be implemented as a top layer orchestrating the functions available in the software API. Solving the 1D electrostatic problem detailed above with the multiharmonic method could then be done with a pseudocode similar to algorithm 3.1.

Algorithm 3.1 Multiharmonic resolution

```
// ------------------------ Define and initialise fields -----------------------
field v,u, vs1,vs3,uc0,uc2;
v.sin1 = vs1; v.sin3 = vs3;
u.cos0 = uc0; u.cos2 = uc2;

// ------------- Compute multiharmonically G as the inverse of |J| -------------
field G,detJ, Gc0,Gc2,detJc0,detJc2;
G.cos0 = Gc0; G.cos2 = Gc2;
detJ.cos0 = 1+dUc0/dx
detJ.cos2 = dUc2/dx
// Define the invjac formulation (unknown is G, test function G'):
formulation invjac;

invjac.definemultiharmonic( $\int_\Omega G(\boldsymbol{x},t)|J(\boldsymbol{x},t)|G'(\boldsymbol{x})\,d\Omega = \int_\Omega G'(\boldsymbol{x})\,d\Omega$ );

// Assemble the algebraic system of the invjac formulation:
invjac.assemblemultiharmonic();
// Solve:
G = invjac.solve();

// --- Define the electrostatic formulation (unknown is v, test function v') ---
formulation electrostatics;

electrostatics.definemultiharmonic( $\int_\Omega -\epsilon\frac{\partial v(\boldsymbol{x},t)}{\partial x}\frac{\partial v'(\boldsymbol{x})}{\partial x}\,G(\boldsymbol{x},t) = 0$ );

// Assemble the algebraic system of the electrostatic formulation:
electrostatics.assemblemultiharmonic();
// Solve and save:
v = electrostatics.solve();
vs1.save(); vs3.save();
```

Function `.definemultiharmonic(string)` is detailed in algorithm 3.2 along with the output of the work done by every line when the 1D electrostatic problem is considered. The function is written for general polynomial formulations. It fully *automatically* derives the multiharmonic formulations for the required number of harmonics in every field. At the same time it also gives all the required directives to the algebraic matrix assembly process so that the multiharmonic generation is also fully automated and transparent for the user.

Algorithm 3.2 .definemultiharmonic(string)

```
string.replacefieldsbytruncations();

>>  $-\epsilon\,\frac{\partial(V_{s1}\sin(2\pi f_0 t)+V_{s3}\sin(3\cdot 2\pi f_0 t))}{\partial x}\frac{\partial v'}{\partial x}\,(G_{c0}+G_{c2}\ \cos(2\cdot 2\pi f_0 t))$

string.expand();

>>  $-\epsilon\,[(G_{c0}\frac{\partial V_{s1}}{\partial x}-\frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x}+\frac{1}{2}G_{c2}\frac{\partial V_{s3}}{\partial x})\ \sin(2\pi f_0 t)+(G_{c0}\frac{\partial V_{s3}}{\partial x}+\frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x})\ \sin(3\cdot 2\pi f_0 t)$
$+\ (\frac{1}{2}G_{c2}\frac{\partial V_{s3}}{\partial x})\ \sin(5\cdot 2\pi f_0 t)]\,\frac{\partial v'}{\partial x}$

string.extractharmonicequations();

>>  $\int_\Omega \epsilon\,(G_{c0}\frac{\partial V_{s1}}{\partial x}-\frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x}+\frac{1}{2}G_{c2}\frac{\partial V_{s3}}{\partial x})\frac{\partial v'}{\partial x}d\Omega=0$

>>  $\int_\Omega \epsilon\,(G_{c0}\frac{\partial V_{s3}}{\partial x}+\frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x})\frac{\partial v'}{\partial x}d\Omega=0$

string.defineharmonicblocks();

>> Defining block 1, i.e. $K_{V_{s1}V_{s1}}$:
>> .define( $\int_\Omega \epsilon\,(G_{c0}\frac{\partial V_{s1}}{\partial x}-\frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x})\frac{\partial v'}{\partial x}d\Omega=0$ ) for unknown $V_{s1}$, test function $V_{s1}$'
>> Defining block 2, i.e. $K_{V_{s3}V_{s1}}$:
>> .define( $\int_\Omega \epsilon\,\frac{1}{2}G_{c2}\frac{\partial V_{s3}}{\partial x}\frac{\partial v'}{\partial x}d\Omega=0$ ) for unknown $V_{s3}$, test function $V_{s1}$'
>> Defining block 3, i.e. $K_{V_{s1}V_{s3}}$:
>> .define( $\int_\Omega \epsilon\,\frac{1}{2}G_{c2}\frac{\partial V_{s1}}{\partial x}\frac{\partial v'}{\partial x}d\Omega=0$ ) for unknown $V_{s1}$, test function $V_{s3}$'
>> Defining block 4, i.e. $K_{V_{s3}V_{s3}}$:
>> .define( $\int_\Omega \epsilon\,G_{c0}\frac{\partial V_{s3}}{\partial x}\frac{\partial v'}{\partial x}d\Omega=0$ ) for unknown $V_{s3}$, test function $V_{s3}$'
```

Algorithm 3.3 .assemblemultiharmonic

```
// Call the monoharmonic assembler on the four (unknown-test function)
// blocks defined in .definemultiharmonic:

// Assemble block  $K_{V_{s1}V_{s1}}$:
 assemble(block 1), unknown is $V_{s1}$, test function $V_{s1}$'
// Assemble block  $K_{V_{s3}V_{s1}}$:
 assemble(block 2), unknown is $V_{s3}$, test function $V_{s1}$'
// Assemble block  $K_{V_{s1}V_{s3}}$:
 assemble(block 3), unknown is $V_{s1}$, test function $V_{s3}$'
// Assemble block  $K_{V_{s3}V_{s3}}$:
 assemble(block 4), unknown is $V_{s3}$, test function $V_{s3}$'
```

In algorithm 3.2 `.define` is the usual monoharmonic finite element definer available in the software API: it processes a user entered formulation to a form understandable by the assembler. After the `.definemultiharmonic` step an algebraic matrix of the form (3.22) is simply obtained by calling `.assemblemultiharmonic`, detailed in algorithm 3.3, in which `.assemble` is the classical monoharmonic finite element assembler already available in the software API. The `.solve` step then solves the algebraic problem (3.22). By defining a new unknown field for each harmonic of the electric potential the already available finite element assembler should be able without modification to assemble at the right place in matrix $\boldsymbol{K}$ all monoharmonic blocks: the existing software does not require major modifications, if any at all. All what is required is to be able to perform simple symbolic operations on strings (or equivalent techniques), which is at least partially available in symbolic processing libraries (e.g. with the `expand` function in Matlab).

Timing tests have been performed on a homemade Matlab code implementing the described explicit-symbolic multiharmonic method. For a varying number of harmonics considered in all fields, figure 3.7 shows the symbolic computation time (top), matrix generation time (center) as well as the LU decomposition time of the generated matrix (bottom) for an electrostatic problem as well as a Newton iteration to solve the electroelastoacoutic problem on the 2D reference CMUT model with about 12000 mesh quadrangles and a third order finite element interpolation. Timings for the reference 3D CMUT are provided in section 5.2. It can be seen in figure 3.7 that the symbolic computation time can be rather large compared to the other timings. Fortunately the symbolic computations must only be computed once for a given Fourier truncation and do not depend on the number of elements in the mesh. As expected the timings for the sensitivity matrix in Newton's method are the largest, since the corresponding formulations include a large number of terms. In any case however the operations are performed in a matter of seconds in 2D since in practice less than 6 harmonics are required for an accurate solution. In 3D the symbolic computation and generation times are a matter of minutes. It is worth noting in figure 3.8 how independent the condition number of the algebraic matrix obtained from the discretisation of the electrostatic problem and from the sensitivity matrix is (the condition number shown is the condition number of the diagonally scaled matrices [89]). This is a desirable property since it means that a higher number of harmonics does not lead to an ill-conditionned algebraic matrix and thus a larger number of harmonics can be considered to accurately simulate the very nonlinear, close to pull-in vibration.

3.3.2 Application to CMUT models

A staggered algorithm to solve the nonlinear coupled electroelastoacoustic (2.65)-(2.70)-(2.89) problem for the simulation of CMUTs was presented in section 2.4.1. Combining this algorithm with the multiharmonic method is straightforward once the electrostatic field can be computed on a geometry deformed by the mechanical displacement, as detailed in section 3.3.1. Updating the mechanical displacement field $\boldsymbol{u}$, the acoustic pressure field δp and the electric potential v from the kth iteration ($\boldsymbol{u}_k$, δp_k, v_k) to the $(k+1)$th iteration ($\boldsymbol{u}_{k+1}$, δp_{k+1}, v_{k+1}) of the multiharmonic staggered algorithm is done in three steps.

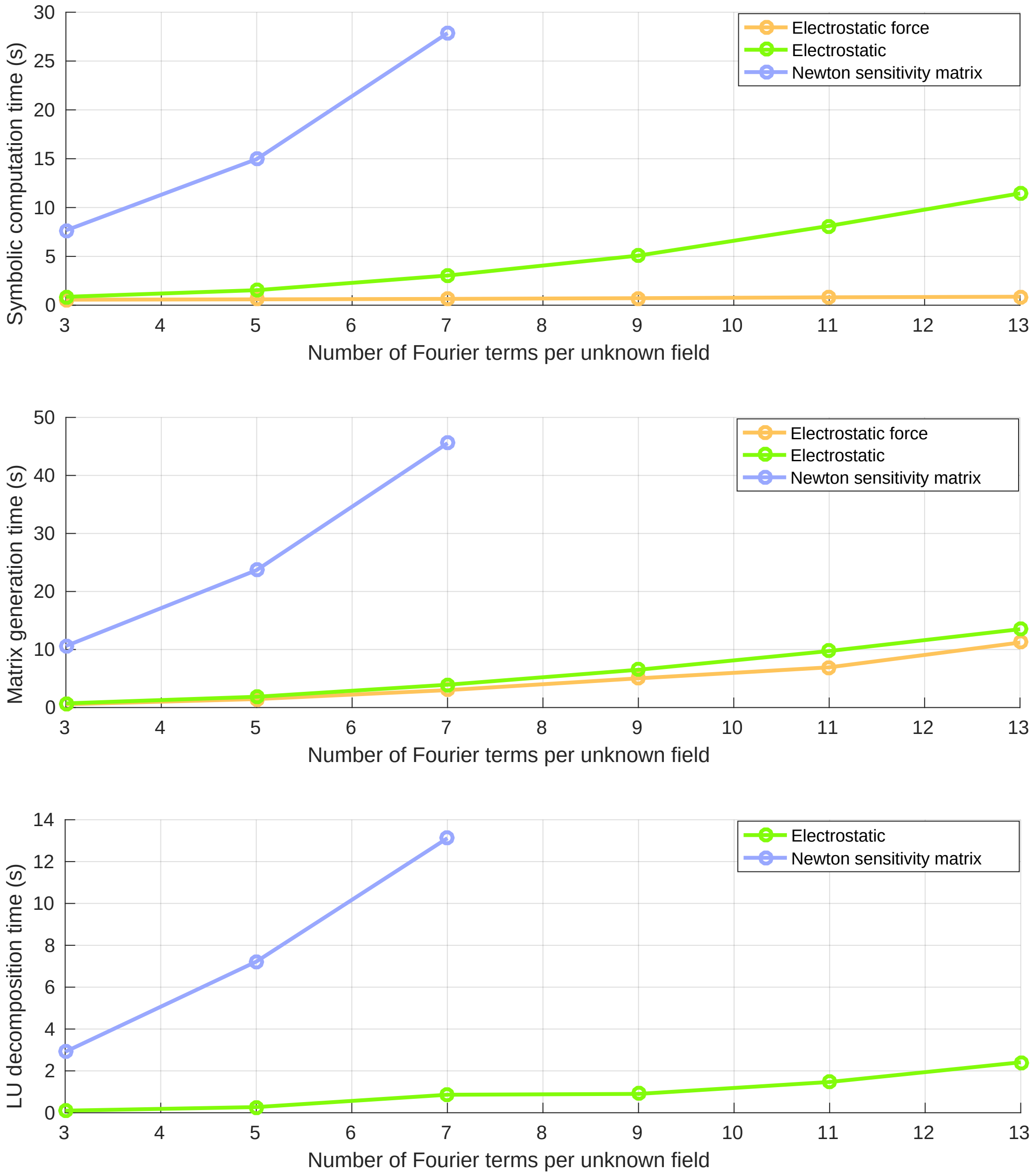


Figure 3.7: Time (s) for the symbolic multiharmonic computation (top), the matrix generation (center) and its LU decomposition (bottom) for the electrostatic force (2.87), the electrostatic problem (2.65) and a the sensitivity matrix (section 2.3.5) in the Newton iteration versus number of terms in the Fourier truncation of every field.

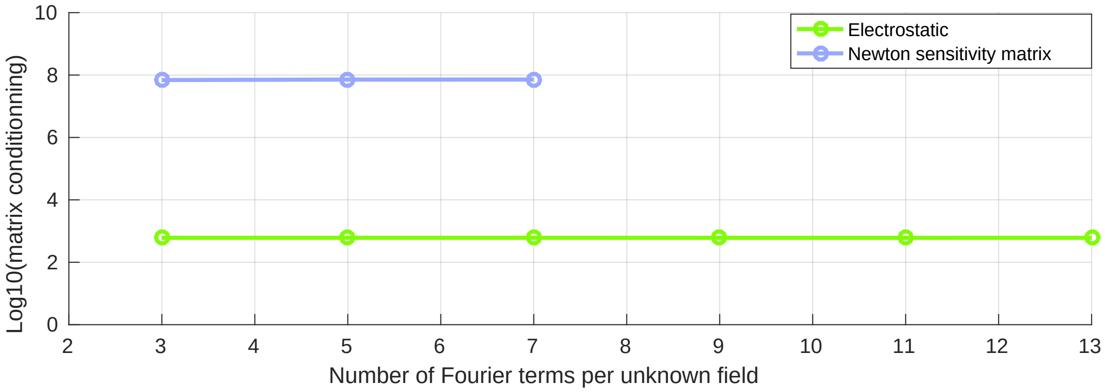


Figure 3.8: Condition number of the diagonally scaled finite element discretised algebraic matrices versus number of terms in the Fourier truncation of every field.

1. Update the electric potential field v_{k+1} (2.65) on the geometry deformed by $\boldsymbol{u}_k$ as detailed in section 3.3.1
2. Update the electrostatic forces f_{k+1} (2.87) (using the computed field v_{k+1}) on the geometry deformed by $\boldsymbol{u}_k$ in the same way as detailed in section 3.3.1
3. Update the mechanical displacement $\boldsymbol{u}_{k+1}$ and acoustic pressure δp_{k+1} (2.70) for every frequency independently with the linear harmonic resolution described in section 3.3 since the formulation is linear. Use f_{k+1} as the electrostatic force

Figure 3.9 (left) shows the sparsity pattern of the discretised algebraic system of the multiharmonic electrostatic problem to solve at step 1. The electric potential harmonics are clearly coupled together, as already observed in (3.22). The sparsity pattern on the right would be obtained if the elastoacoustic problem to solve at step 3 was solved with the multiharmonic resolution. The decoupling between harmonics at different frequencies can be observed: for every frequency kf_0 and hf_0 with $k \neq h$ the harmonic set $\boldsymbol{u}_{sk}$, $\boldsymbol{u}_{ck}$, δp_{sk} and δp_{ck} can be solved independently from the set $\boldsymbol{u}_{sh}$, $\boldsymbol{u}_{ch}$, δp_{sh} and δp_{ch}. This is a key point of the multiharmonic staggered resolution for CMUTs since in practice the elastoacoustic problem is the computationally intensive and memory demanding step of a staggered iteration.

Solving the nonlinear coupled electroelastoacoustic problem with Newton's method described in section 2.4.2 is a faster converging yet computationally expensive alternative to the staggered resolution. Once the staggered multiharmonic implementation is understood, applying the multiharmonic method to Newton's algorithm is straightforward with an automatic multiharmonic assembler. Unsurprisingly the finite element assembly is much more demanding in terms of computational power than for the staggered multiharmonic method since the extra sensitivity matrix (2.3.5) to generate is large and less sparse. All $\boldsymbol{u}$, δp and v harmonics in the coupled electroelastoacoustic (2.65)-(2.70)-(2.89) formulation at a given Newton iteration will be coupled so that unlike for the staggered resolution method the frequencies can not be solved independently: the size of the problem to solve at every Newton iteration is proportional to the total sum of the number of harmonics in every field.

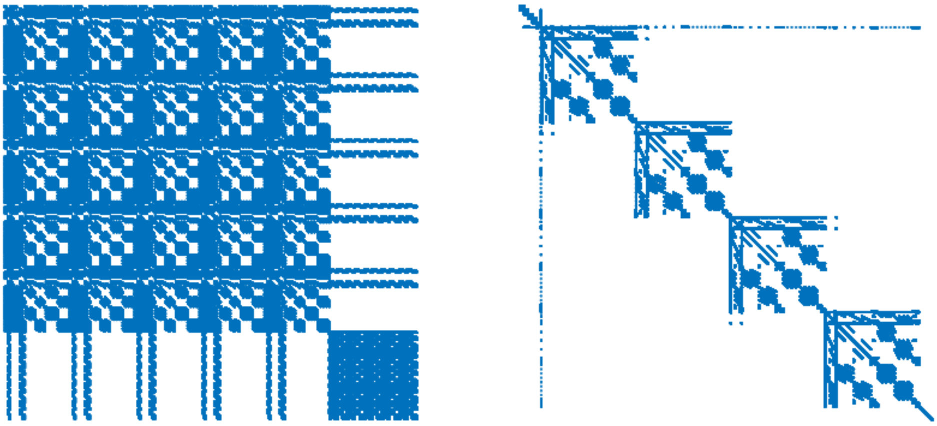


Figure 3.9: Sparsity pattern of the multiharmonic electrostatic stiffness matrix (left) and the multiharmonic elastoacoustic stiffness matrix (right) for 5 harmonics per unknown field (4 for the pressure field).

For both the staggered algorithm and Newton's method it is interesting to investigate the dependency of the nonlinear convergence rate on the number of harmonics considered in the electric potential v, in the mechanical displacement $\boldsymbol{u}$ and in the pressure field δp. Figure 3.10 shows the convergence history for an increasing number of harmonics in every field when the staggered multiharmonic algorithm is used to simulate the reference 2D CMUT of figure 1.5 with an electrical excitation voltage $v(t) = 40 + 40 \cdot \sin(2\pi \cdot 800000t)$ V on the left membrane and $v(t) = 40$ V on the right one. A same number of harmonics is used for fields v, $\boldsymbol{u}$ and δp. It appears that the nonlinear convergence rate of the staggered method is unaffected by an increase in the number of harmonics considered. With the same settings, figure 3.11 however shows a different behaviour for Newton's method: the convergence rate improves as the number of harmonics considered increases. When only the number of harmonics for v increases (bottom) the same strong improvement is observed: for only two electric potential harmonics the convergence is not quadratic but for 5 harmonics it becomes quadratic. This effect can be attributed to the way the multiharmonic Newton Jacobian matrix is computed: since it contains nonlinear terms it can not be computed exactly for a limited number of harmonics. The stiffness matrices of the electrostatic and elastoacoustic formulations however can be exactly computed since the formulations are linear.

3.3.3 Illustration on the reference nonlinear 2D CMUT model

We consider the reference 2D CMUT model of figure 1.5. In practical CMUT applications a close-to-linear vibration is achieved by adding a tiny alternating voltage (e.g. 1% of the pull-in voltage) to a big constant excitation voltage (e.g. 90% of the pull-in voltage). Here however the nonlinear behaviour is put to the fore by applying a larger than usual alternating voltage. In the current test case an electrode-to-ground excitation of $v(t) = 40 + 40 \sin(2\pi \cdot 800000\, t)$ V with a strong alternating voltage is applied to the left membrane and $v(t) = 40$ V is applied on the right one. The pull-in voltage is of about 110 V and the first mode resonance at about 1 MHz: the behaviour should thus be clearly nonlinear and close to resonant. Similarly to (3.8) the mechanical

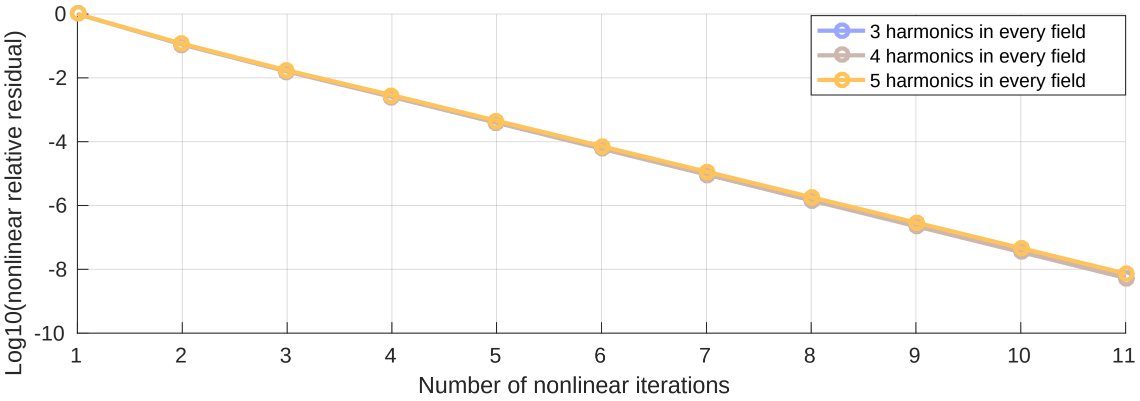


Figure 3.10: Convergence history when the staggered algorithm is used to solve the nonlinear electroelastoacoutic problem of the reference CMUT in figure 1.5.

displacement field $\boldsymbol{u}$, the pressure field δp and the electric potential field v are decomposed as

$$\boldsymbol{u}(x,t) = \boldsymbol{U}_{c0}(x)+\boldsymbol{U}_{s1}(x)\sin(2\pi f_0 t)+\boldsymbol{U}_{c1}(x)\cos(2\pi f_0 t)+\boldsymbol{U}_{s2}(x)\sin(4\pi f_0 t)+\boldsymbol{U}_{c2}(x)\cos(4\pi f_0 t)+..., \tag{3.23}$$

for the mechanical displacement field $\boldsymbol{u}$,

$$\delta p(x,t) = \delta P_{c0}(x)+\delta P_{s1}(x)\sin(2\pi f_0 t)+\delta P_{c1}(x)\cos(2\pi f_0 t)+\delta P_{s2}(x)\sin(4\pi f_0 t)+\delta P_{c2}(x)\cos(4\pi f_0 t)+..., \tag{3.24}$$

for the acoustic pressure field δp and

$$v(x,t) = V_{c0}(x)+V_{s1}(x)\sin(2\pi f_0 t)+V_{c1}(x)\cos(2\pi f_0 t)+V_{s2}(x)\sin(4\pi f_0 t)+V_{c2}(x)\cos(4\pi f_0 t)+..., \tag{3.25}$$

for the electric potential field v. The shape of the first 9 vibration harmonics is displayed in figure 3.12 for a single and for two membranes and in figure 3.13 for three membranes. In any case the 40 V bias excitation is applied to all membranes and the 40 V alternating excitation is only applied to the leftmost membrane. Comparing these two figures clearly shows the impact of the array size on the crosstalk and how complex it can be to predict the vibration harmonics.

The absolute magnitude of the displacement, the pressure and the electric potential harmonics with two membranes are shown in figure 3.14. For the more typical electrical excitation $v(t) = 95+V_{AC}\ \sin(2\pi\cdot 800000\,t)$ V (with V_{AC} ranging from 1 to 7 V) leading to a close-to-linear vibration around a static deflection, the maximum harmonic values is displayed in figure 3.15. From figure 3.14 it can be seen that for the nonlinear CMUT excitation a good approximation is obtained by considering five harmonics for the mechanical displacement $\boldsymbol{u}$, two for the electric potential v and four for the acoustic pressure δp.

The crosstalk and the nonlinear behaviour are clearly captured in figure 3.12 by the multiharmonic resolution. As can be seen additional mechanical vibration harmonics appear because of

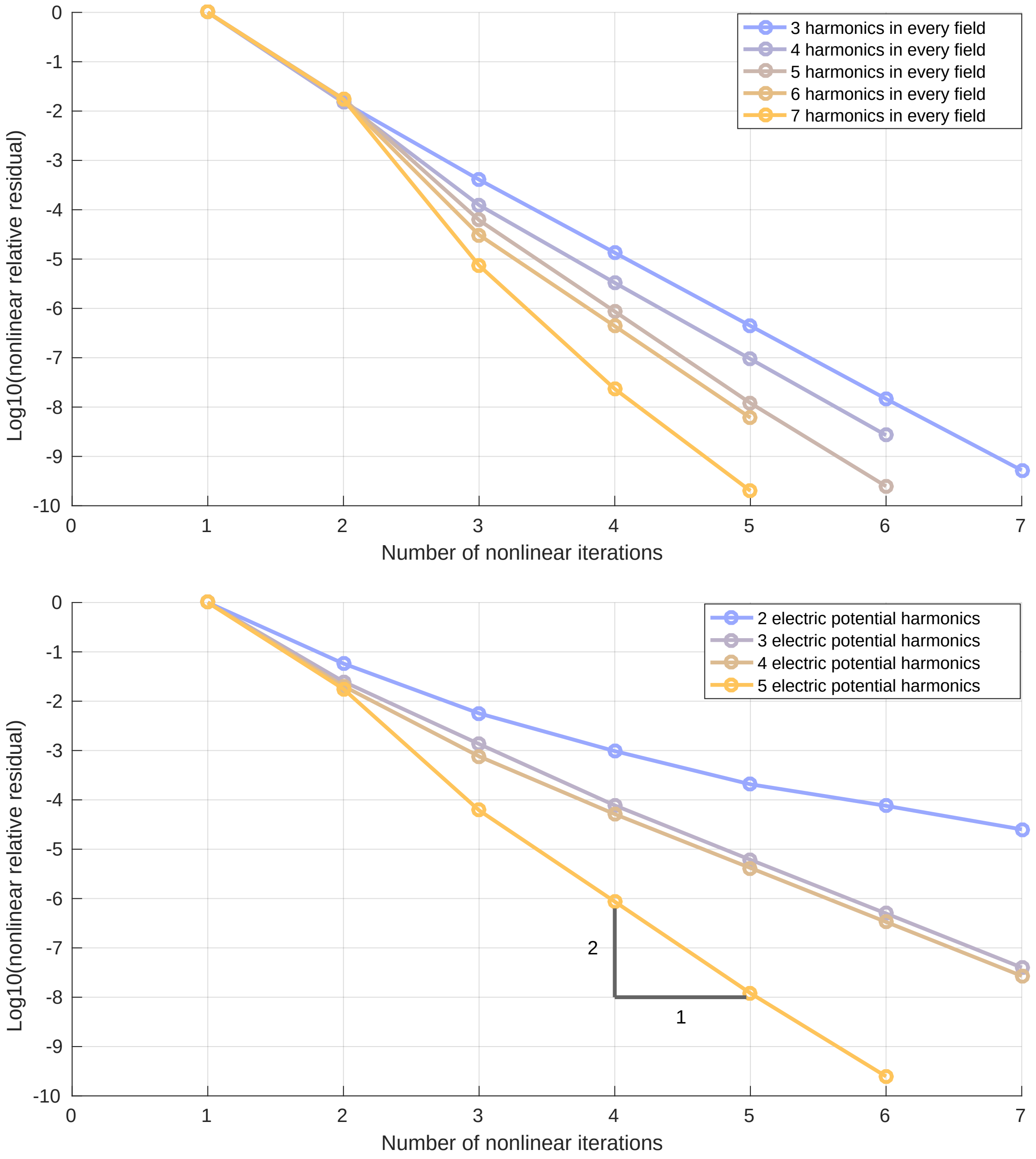


Figure 3.11: Convergence history when Newton's method is used to solve nonlinear electroelastoacoutic on the reference CMUT of figure 1.5. On the top figure the number of harmonics is increased at the same time for the electric potential v, the mechanical displacement $\boldsymbol{u}$ and the acoustic pressure δp. On the bottom figure it increases only for v while a constant 5 harmonics are considered for $\boldsymbol{u}$ and δp.

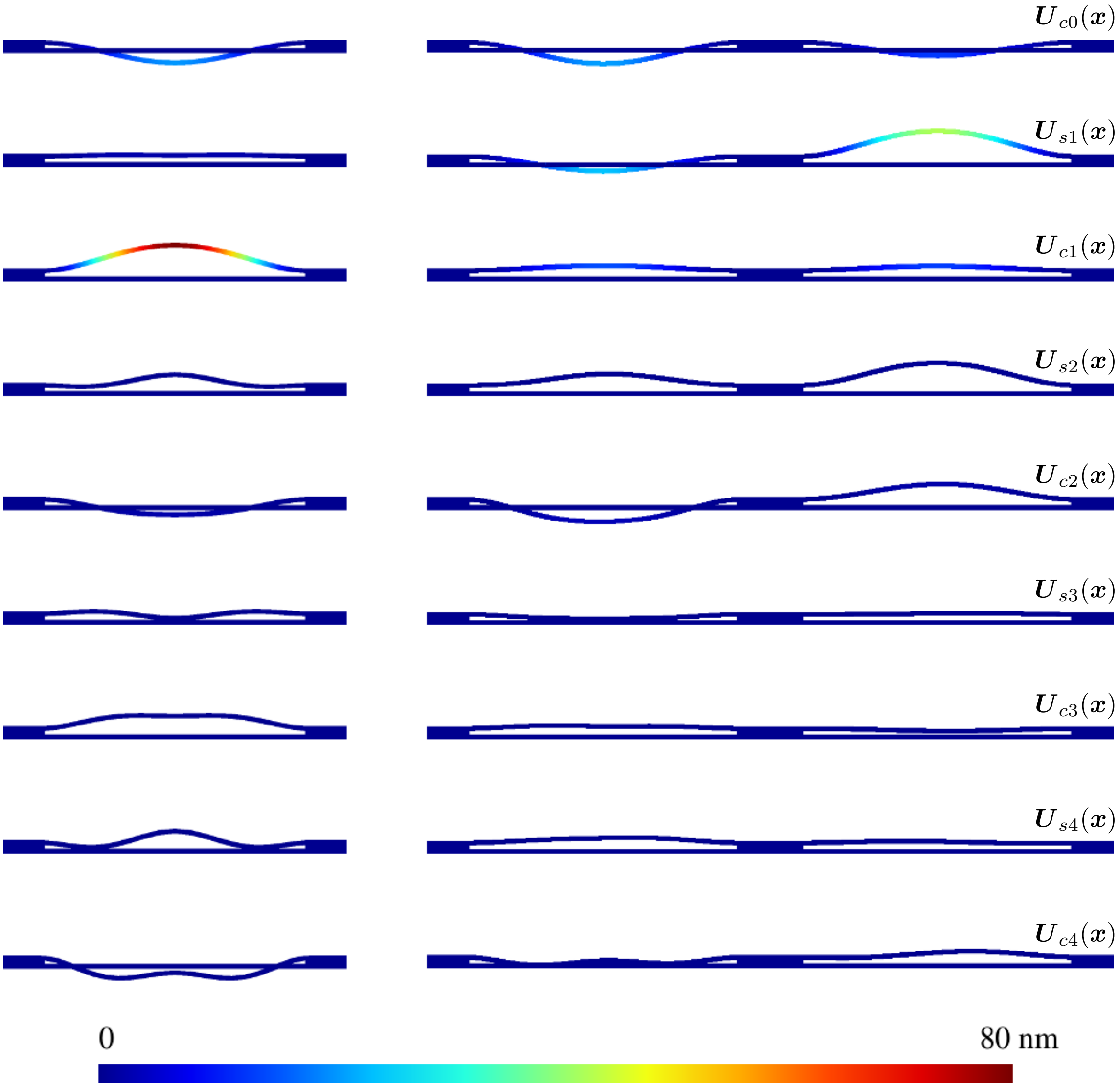


Figure 3.12: Shape of the displacement harmonic $\boldsymbol{U}_{c0}$, $\boldsymbol{U}_{s1}$, $\boldsymbol{U}_{c1}$, $\boldsymbol{U}_{s2}$, $\boldsymbol{U}_{c2}$, $\boldsymbol{U}_{s3}$, $\boldsymbol{U}_{c3}$, $\boldsymbol{U}_{s4}$, $\boldsymbol{U}_{c4}$ (from north to south) for a single and for two membranes. In the latter case an alternating excitation is added to the bias only on the left membrane. Displacements (m) are exagerated.

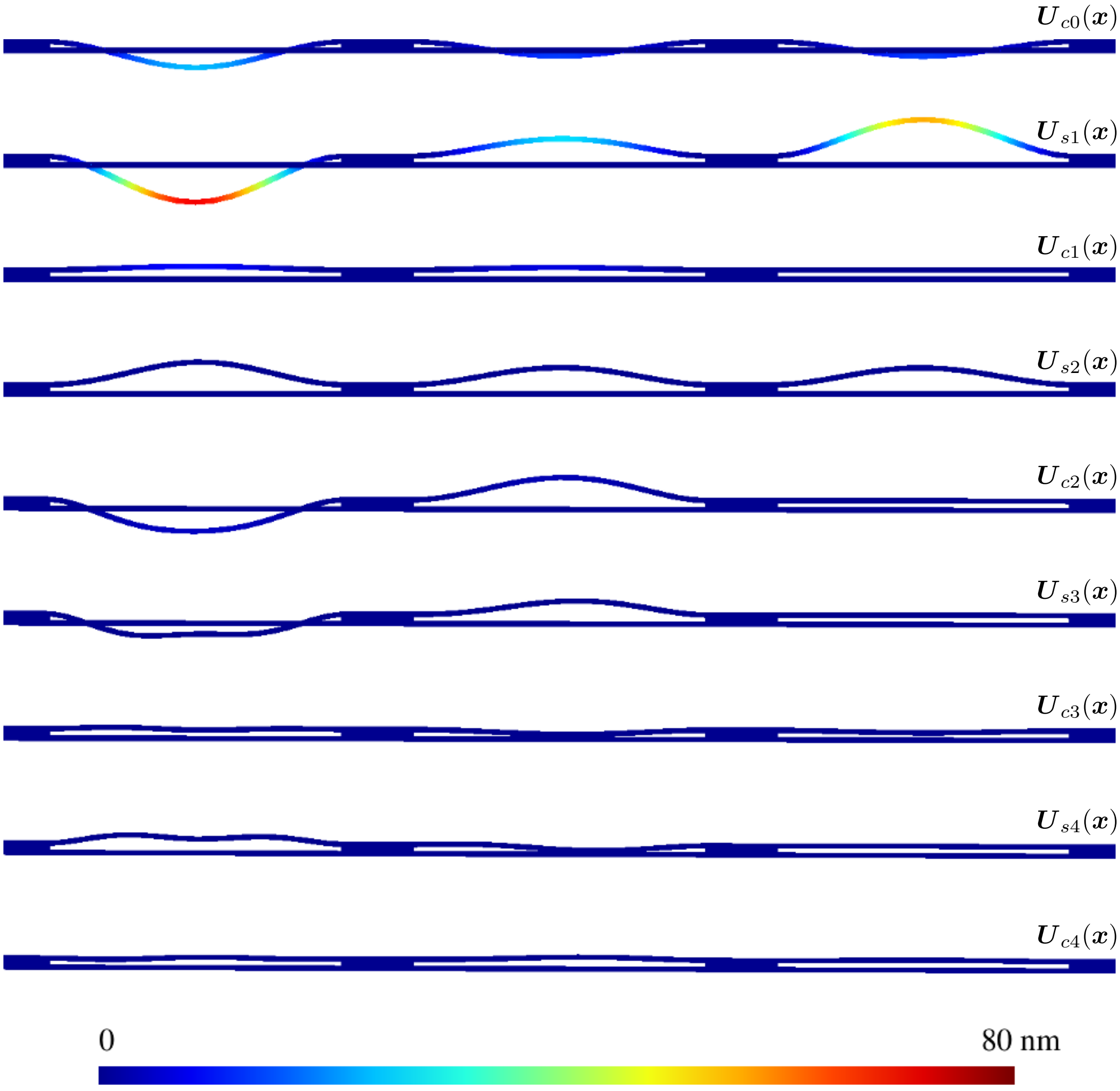


Figure 3.13: Shape of the displacement harmonic $\boldsymbol{U}_{c0}$, $\boldsymbol{U}_{s1}$, $\boldsymbol{U}_{c1}$, $\boldsymbol{U}_{s2}$, $\boldsymbol{U}_{c2}$, $\boldsymbol{U}_{s3}$, $\boldsymbol{U}_{c3}$, $\boldsymbol{U}_{s4}$, $\boldsymbol{U}_{c4}$ (from north to south). An alternating excitation is added to the bias only on the left membrane. Displacements (m) are exagerated.

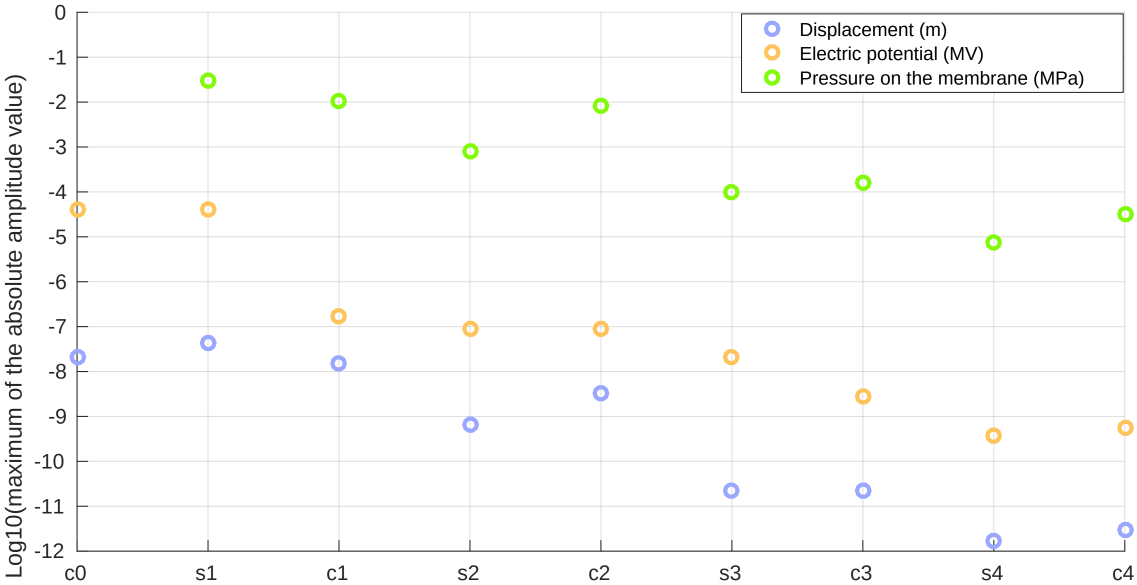


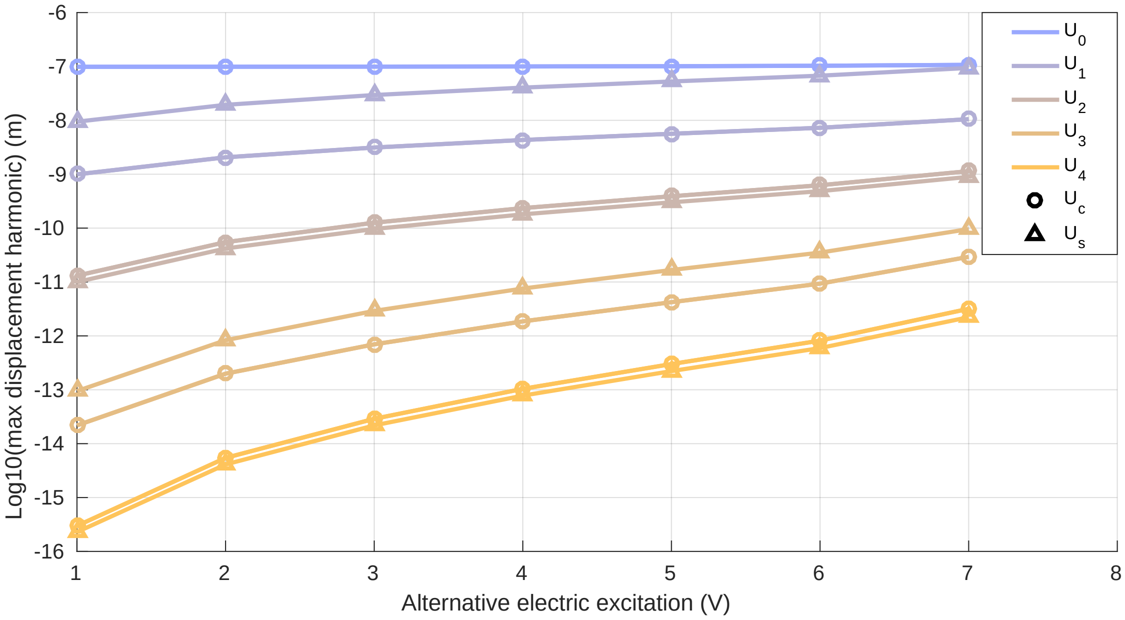
Figure 3.14: Maximum absolute value displacement (m), pressure (MPA, $10^6 \cdot \mathrm{Pa}$) and electric potential (MV, $10^6 \cdot V$) for every of the 9 first harmonics. Pressure is measured at the membrane top.

the nonlinearity. With a tiny alternating electrical excitation component (figure 3.15) the Fourier terms $\boldsymbol{U}_{c0}$, $\boldsymbol{U}_{s1}$ and $\boldsymbol{U}_{c1}$ are dominant and the vibration is close to linear. The terms $\boldsymbol{U}_{s2}$ and $\boldsymbol{U}_{c2}$ grow bigger as the alternating excitation component is increased and the constant component is decreased as can be seen in figure 3.14 and 3.15. In the limit case of a 0 V constant electrical excitation component all what is left is $\boldsymbol{U}_{c0}$, $\boldsymbol{U}_{s2}$, $\boldsymbol{U}_{c2}$ and higher harmonics. This makes physical sense since the electric force acting on the membrane will have the same direction no matter the sign of the alternating electric potential: the vibration frequency is doubled. Another impact of the nonlinear behaviour is the appearance of new resonance peaks between 0 Hz and the first mode linear resonance frequency. As an example the Fourier term $\boldsymbol{U}_{c2}$ vibrates at twice the $\boldsymbol{U}_{c1}$ vibration frequency and resonates thus at about half of its electrical excitation resonance frequency. The bottom harmonics visible in figure 3.12 are clearly vibrating beyond their first resonance mode. An additional nonlinear behaviour visible in the figure is the different time-constant deformation (top of the figure) on the left and right membranes, even though both have a same constant excitation of 40 V. This comes from the coupling between harmonics when nonlinearity is considered.

3.4 Comparison of the multiharmonic resolution and Newmark's method

For the lumped CMUT model of figure 3.1 the deflection $u(t)$ was plotted in figure 3.2 for the first couple of excitation voltage periods. The transient state was clearly visible for at least the five first excitation periods before settling down to reach the steady state reponse.

For the 2D reference CMUT of figure 1.5 with a $v(t) = 40 + 40 \cdot \sin(2\pi \cdot 800000t)$ V electrical



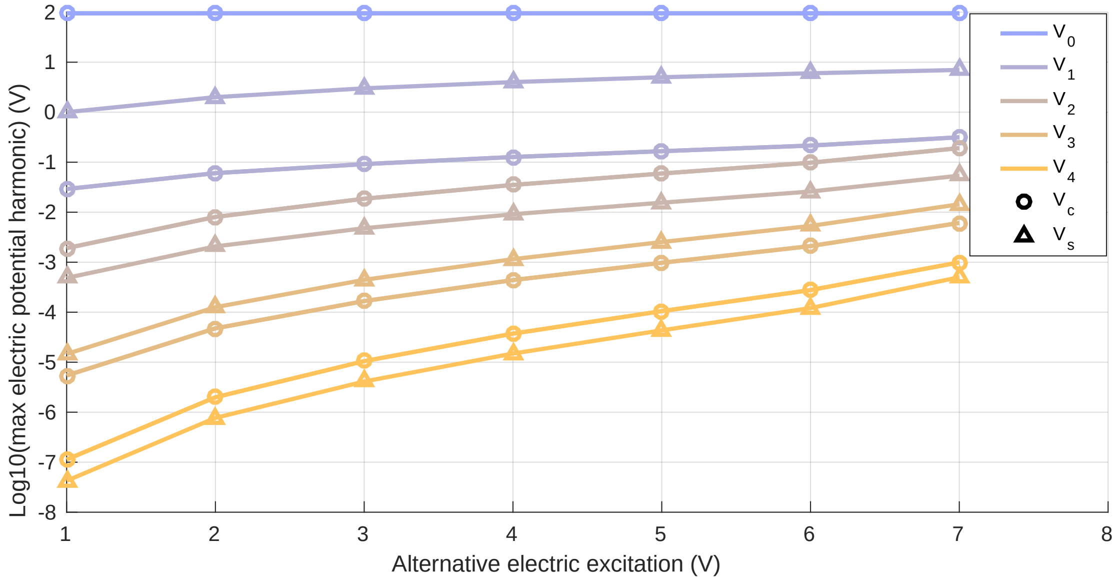


Figure 3.15: Maximum absolute value of the 9 first mechanical displacement (m) harmonics ($\boldsymbol{U}_{c0}$, $\boldsymbol{U}_{s1}$, $\boldsymbol{U}_{c1}$, $\boldsymbol{U}_{s2}$, $\boldsymbol{U}_{c2}$, $\boldsymbol{U}_{s3}$, $\boldsymbol{U}_{c3}$, $\boldsymbol{U}_{s4}$, $\boldsymbol{U}_{c4}$) (top) and electric potential (V) harmonics (V_{c0}, V_{s1}, V_{c1}, V_{s2}, V_{c2}, V_{s3}, V_{c3}, V_{s4}, V_{c4}) (bottom) versus alternative electrical excitation voltage V_{AC} (V) for a $95 + V_{AC}\,\sin(2\pi \cdot 800000\,t)$ volts electrode to ground electrical excitation.

excitation voltage on the left membrane and $v(t) = 40$ V on the right one a close to resonance, nonlinear vibration is obtained. Figure 3.16 shows the beginning of the transient response when the excitation is suddenly applied to the membranes at rest while figure 3.17 shows the steady state response on both the left and the right membranes. Unlike for the lumped model, after 10 μs all harmonics in the transient response have not been damped enough to be in steady state. This is because the damping coefficient used in the lumped model was exagerated for illustration purposes. For medical CMUT applications with a water fluid a large damping source comes from the power radiated by the emitted acoustic pressure waves. This damping is much smaller than the damping in the lumped model so that a long transient state has to be time stepped through with Newmark's method before reaching the steady state that is of interest: figure 3.18 illustrates that on the reference 2D CMUT. The figure shows horizontal lines as well as curves vibrating around those horizontal lines. The horizontal lines represent the steady state harmonic magnitudes computed with the multiharmonic method, 5 harmonics are considered for every field. The curves vibrating around come from a Fourier transform performed on the time-solution obtained with Newmark's method: at every electrical excitation period a Fourier transform is computed on the maximum membrane deflection and thus the period-by-period evolution of the harmonics in the time solution is obtained. Parameters $\gamma = 0.5$ and $\beta = 0.25$ were used in Newmark's method with $\delta t = 8.33$ ns (i.e. 150 time steps per period). The linear acceleration and Fox & Goodwin algorithm described in table 3.1 have been tried as well but more than 100000 time steps per period were required to have a stable integration scheme.

Figure 3.18 shows the large number of electrical excitation periods required to reach an approximation of the steady state: for a correct approximation at least 100 periods are required. While this only corresponds to a very short timespan (125 μs) the computational cost associated is large: 150 time steps were used in each period to have an accurate time resolution so that steady state was reached after more than 15000 time steps. While it is true that every time step corresponds to a smaller problem than in the multiharmonic resolution, the large number of time steps required combined with the relatively lost cost of the staggered multiharmonic resolution (where the large elastoacoustic problem can be solved for every frequency independently) make for an orders of magnitudes faster multiharmonic resolution. Additionally the multiharmonic algorithm has proven more robust since it can deliver the solution for electrical excitations even closer to resonance while Newmark's method can not without additional effort because the transient overshoots and brings the membrane beyond pull-in.

3.5 Conclusion

The multiharmonic method was introduced on a 1D electrostatic problem computed on a vibrating mesh. The difficulties arising form the nonlinearity and the vibrating mesh have been discussed and a solution has been proposed. A possible implementation of the multiharmonic method has been drafted with pseudo-code. It has demonstrated the potential of the method to be fully automated with e.g. symbolic computation while it has also shown that only few changes, if any, are required for an implementation in an existing finite element assembler software. Timings for the simulation on a 2D CMUT application, implemented in a non-optimised Matlab code, have shown that the

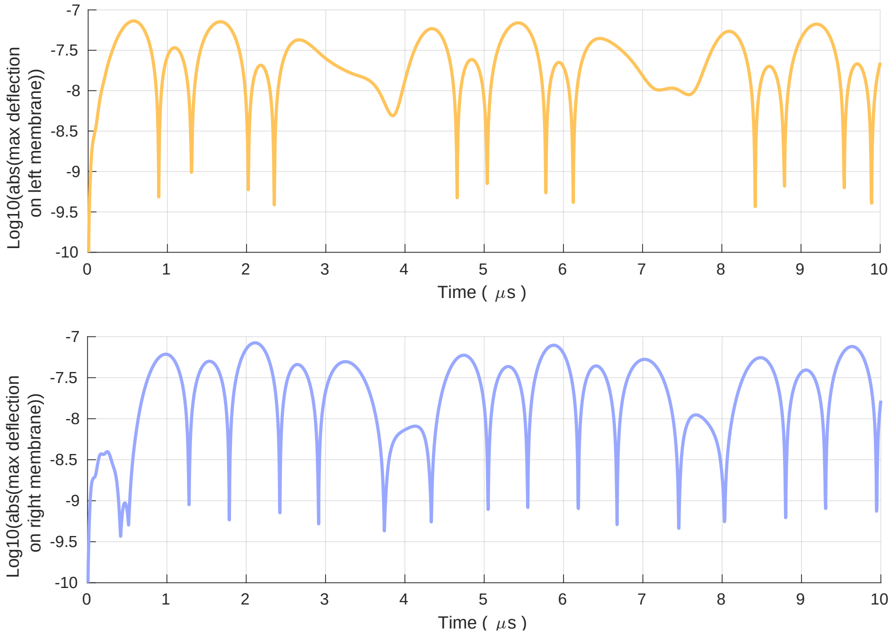


Figure 3.16: Absolute value of the maximum deflection on the left and right membrane of the reference CMUT of figure 1.5 when a $v(t) = 40 + 40 \cdot \sin(2\pi \cdot 800000t)$ V electrical excitation is suddenly applied to the left membrane and $v(t) = 40$ V to the right one. Initial state is at rest.

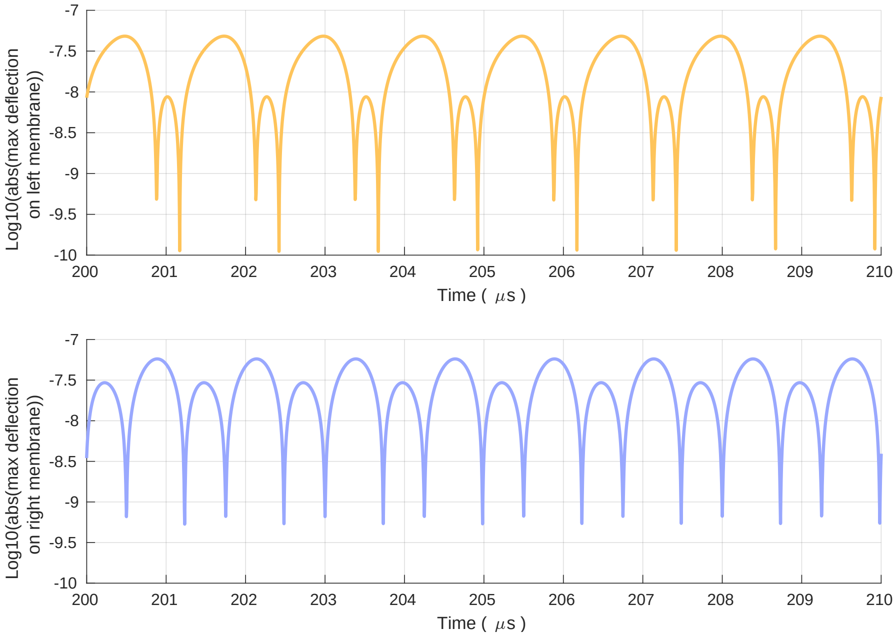


Figure 3.17: Absolute value of the maximum deflection on the left and right membrane of the reference CMUT of figure 1.5 when a $v(t) = 40 + 40 \cdot \sin(2\pi \cdot 800000t)$ V electrical excitation is applied to the left membrane and $v(t) = 40$ V to the right one. Steady state is reached.

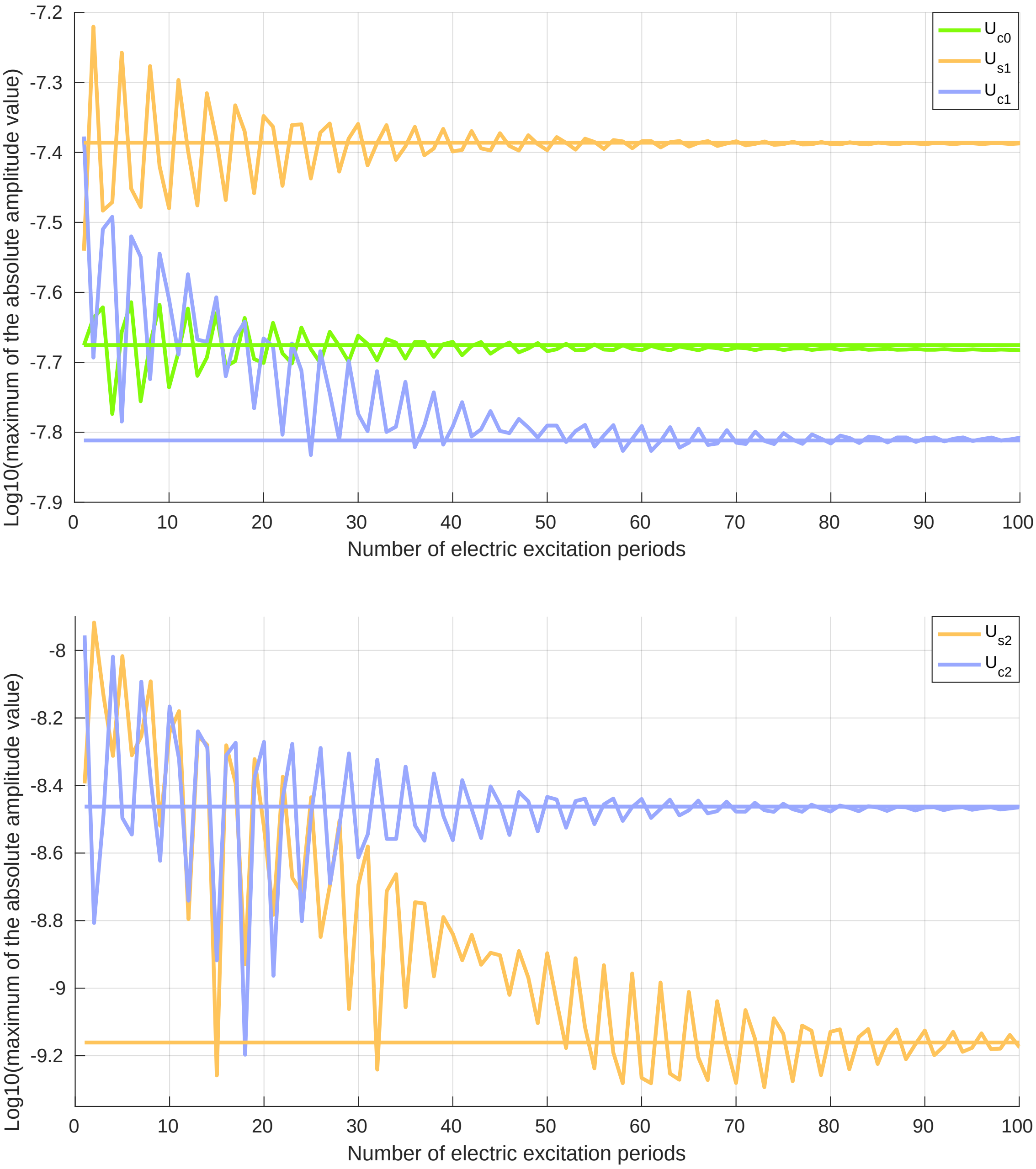


Figure 3.18: Maximum of the multiharmonically computed displacement (m) harmonics (horizontal lines) and of the harmonics computed via a Fourier transform at every electrical excitation period in the Newmark time resolution. Time increases in the right direction.

extra cost associated to the symbolic processing is not problematic. Finally the advantage of using the multiharmonic method over Newmark's method to simulate the steady state of CMUTs has been demonstrated on a realistic CMUT application with a close to resonance, nonlinear electrical excitation: the multiharmonic method is orders of magnitudes faster as well as more robust.

Chapter 4

Domain decomposition methods

This chapter describes domain decomposition methods (DDMs) with the goal of solving large nonlinear electroelastoacoustic problems on a parallel computing architecture. The classical DDMs for linear problems are first introduced and the impact of the choice of interface conditions on the convergence rate is discussed. Methods for the nonlinear electroelastoacoustic problem are then detailed and compared. Finally a coarse grid preconditionner is proposed to speed up convergence for large MEMS arrays.

4.1 Introduction

Typical capacitive micromachined ultrasonic transducers (CMUTs) used for ultrasound imaging consist of large arrays of replicated elementary 3D cells (a 2D cell is illustrated in figure 1.2). Getting the numerical solution to this problem is a computationally challenging task involving in 3D tens of millions degrees of freedom ruled by nonlinear equations. Such a resolution cannot rely solely on classical direct solver technology. In this thesis the large computation is made possible by a DDM taking advantage of parallel computation. The fundamental idea of DDM is to split the computational domain Ω into n smaller subdomains Ω_i with or without overlap and solve at every iteration of an iterative solver the problem independently on every subdomain, preferably in parallel. In order to end up with the correct solution on the whole domain one has to exchange interface data between the subdomains at every iteration. At convergence the union of the solution on all subdomains is close enough to the actual solution on Ω. Only overlapping DDMs are considered in this thesis and for simplicity a subdomain corresponds to a single CMUT cell.

4.2 Linear problems

This section introduces the fundamental DDM algorithms for linear problems. Most of the proposed DDM algorithms for nonlinear problems in section 4.3 are based on their linear counterparts exposed in this section.

4.2.1 Linear alternating Schwarz algorithm

One of the simplest DDM algorithms is the classical alternating Schwarz algorithm [107, 108]. As an illustration consider a 1D electrostatic problem (for a uniform electric permittivity) $\Delta v = 0 \equiv \frac{\partial^2 v}{\partial x^2} = 0$ on domain Ω with imposed electric potential v on the domain boundaries. Split Ω in two overlapping domains Ω_1 (with coordinates ranging from $x = 0$ to 6) and Ω_2 (ranging from $x = 4$ to 10) as shown in figure 4.1. Solving the problem for an initial all zero guess with the classical alternating Schwarz algorithm consists in performing the following updates at the kth iteration:

$$\begin{cases} \dfrac{\partial^2 v_1^{k+1}}{\partial x^2} = & 0 \text{ on } & \Omega_1, \\ v_1^{k+1} = & v_2^k \text{ on } & \partial\Omega_1, \\ v_1^{k+1} = & 3 \text{ on } & \Gamma_{D1}, \\ v_1^0 = & 0 \text{ on } & \Omega_1, \end{cases} \tag{4.1}$$

for v_{Ω_1} on Ω_1 and

$$\begin{cases} \dfrac{\partial v_2^{2k+1}}{\partial x^2} = & 0 \text{ on } & \Omega_2, \\ v_2^{k+1} = & v_1^k \text{ on } & \partial\Omega_2, \\ v_2^{k+1} = & 2 \text{ on } & \Gamma_{D2}, \\ v_2^0 = & 0 \text{ on } & \Omega_2, \end{cases} \tag{4.2}$$

for v_{Ω_2} on Ω_2. It is worth noting that this algorithm is well suited for parallelisation as the communication cost between processing units is limited to exchanging field values at the interfaces. Figure 4.1 illustrates the iterates of the algorithm. As can be seen after enough iterations the actual solution (a linear decrease from 3 V to 2 V) is reached. As can be easily understood from figure 4.1 for a zero overlap the method stagnates. For a non zero overlap the convergence is accelerated as the overlap length increases. Unfortunately even for large overlaps the Schwarz alternating algorithm exhibits a rather slow convergence as illustrated in figure 4.2 where the algorithm needs a hundred iterations to reach machine precision for the large 20% overlap.

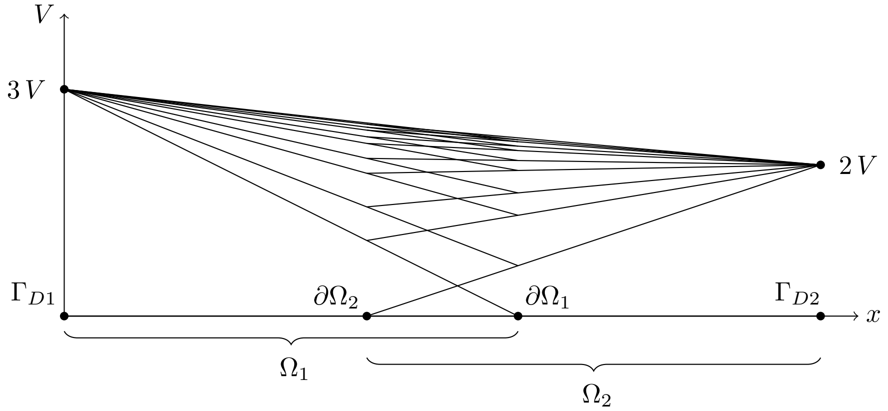


Figure 4.1: Iterations of the Schwarz alternating method for a 1D electrostatic problem with two subdomains and an all 0 initial guess.

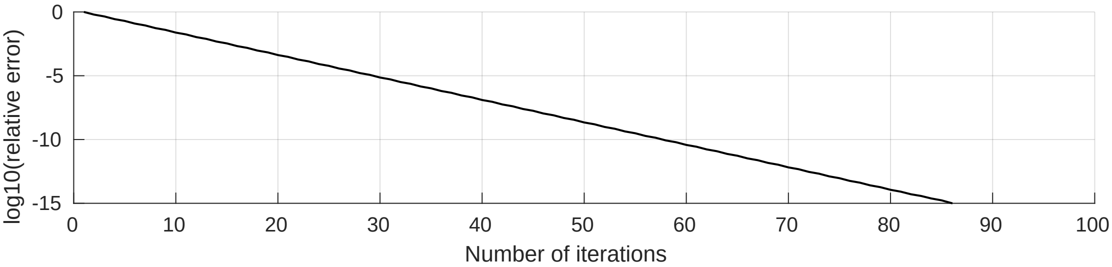


Figure 4.2: Relative error at coordinate $x = 4$ versus number of iterative Schwarz iterations for the 1D test case of figure 4.1.

For the reference 2D CMUT of figure 1.5 with 2 membranes and excitation voltage settings leading to a quasi-linear behaviour ($v(t) = 10 + 1 \cdot \sin(2\pi f_0 t)$ V on the electrode of the left membrane and 10 V on the right one) the resolution of the elastoacoutic formulation (2.16)-(2.38) with the linear alternating Schwarz method leads to the convergence curves illustrated in figure 4.3 for the 0 Hz harmonics of the problem and figure 4.4 for the fundamental frequency harmonics (the frequencies can be solved uncoupled as detailed in section 2.4). In both cases a 5% overlap is used. In the 0 Hz problem the 2 membranes in the CMUT array are not coupled by the fluid since the pressure field is zero everywhere and they thus barely influence each other. As a consequence the alternating Schwarz algorithm performs well. For the fundamental frequency problem the method seems to stagnate well below the first resonance frequency (about 1 MHz). As the electrical excitation frequency increases to get closer to resonance however the method diverges faster and faster. Once the first resonance frequency is passed figure 4.4 (bottom) clearly shows the convergence of the method. In any case however several hundreds of iterations are required to reach the lowest achievable overall relative residual. The overall relative residual is computed as $\frac{||\boldsymbol{b}-\boldsymbol{A}\boldsymbol{x}||_2}{||\boldsymbol{b}||_2}$ for a finite element discretised algebraic problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. $\boldsymbol{A}$ and $\boldsymbol{b}$ are the algebraic matrix/vector corresponding to the discretised elastoacoustic problem (as detailed in sections 2.3.2 and 2.3.4) on the whole domain Ω while the discretised acoustic pressure and mechanical displacement fields in the $\boldsymbol{x}$ vector are the 0-overlap disjointly-restricted sum of the fields on all subdomains, i.e. the value of a given field on a given subdomain is first zeroed everywhere but on the 0-overlap restricted subdomain before it is added to the field value on all other subdomains.

4.2.2 Krylov-Schwarz with interface unknowns

The Krylov-Schwarz algorithm with interface unknowns in the DDM iteration [109] (identified in the following by acronym S) improves the convergence rate of the linear alternating Schwarz algorithm while keeping its low communication cost advantage for parallelisation.

Consider a linear continuous problem $F(u) = f$ on the whole region Ω of figure 4.5 subject to applied loads f and Dirichlet constraints u_D on Γ_D. Ω is split into n overlapping subdomains Ω_i for $i = 1, ..., n$. Call Σ_{ji} the ith subdomain's outer interface shared with subdomain j . When subscript ij is used in the following then i represents the origin (i.e. on which subdomain the data is computed) and j the destination (i.e. for which subdomain the data is computed). Considering one harmonic frequency at a time the problem to solve is:

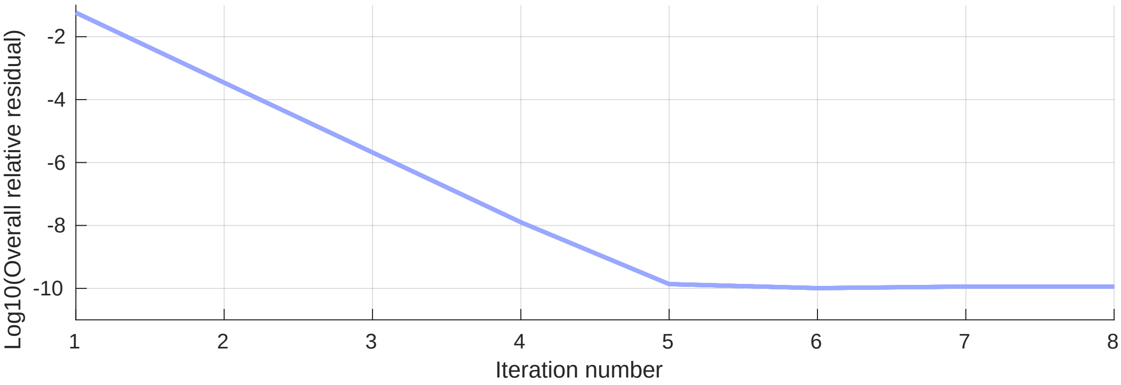


Figure 4.3: Overall relative residual versus iteration number for the linear iterative Schwarz algorithm applied to the 0 Hz part of the elastoacoutic formulation.

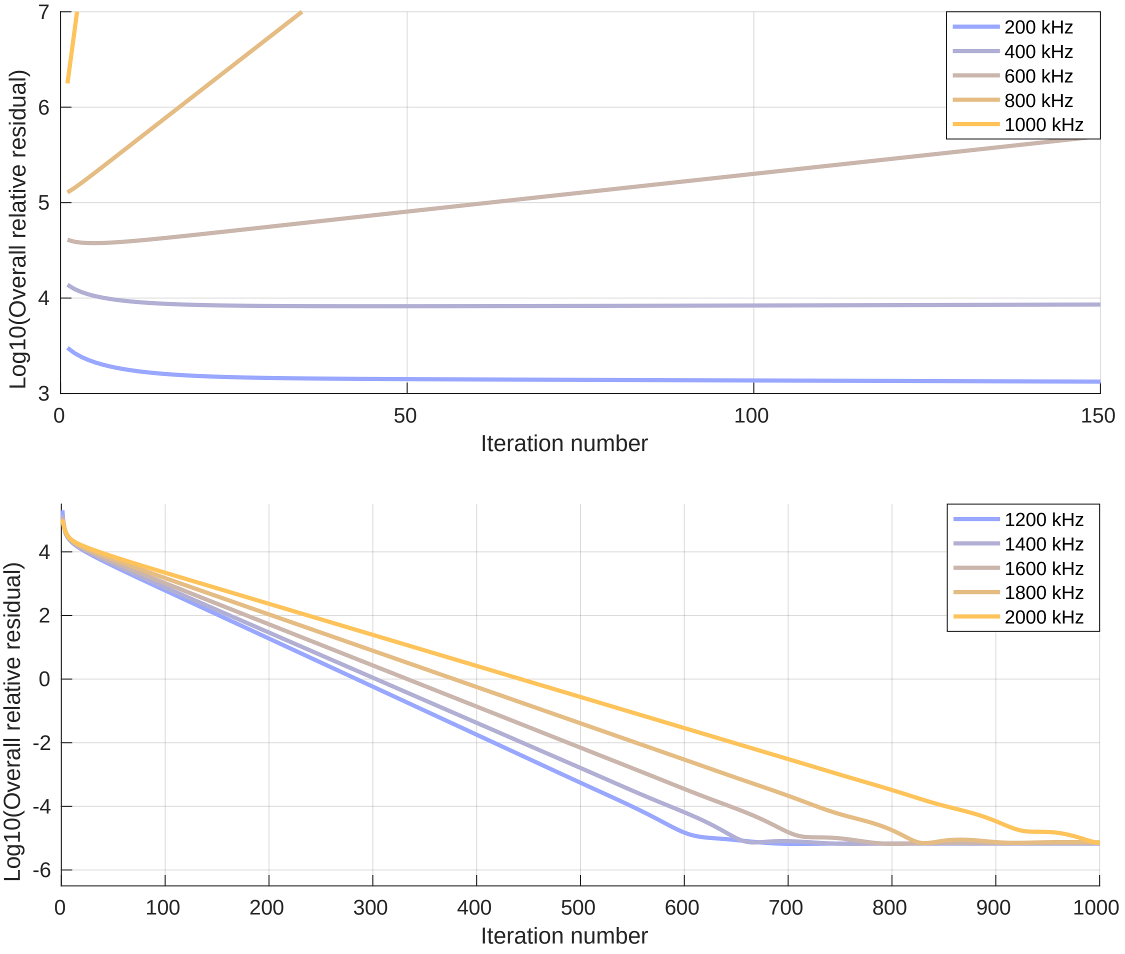


Figure 4.4: Overall relative residual versus iteration number for the linear iterative Schwarz algorithm applied to the fundamental frequency part of the elastoacoutic formulation.

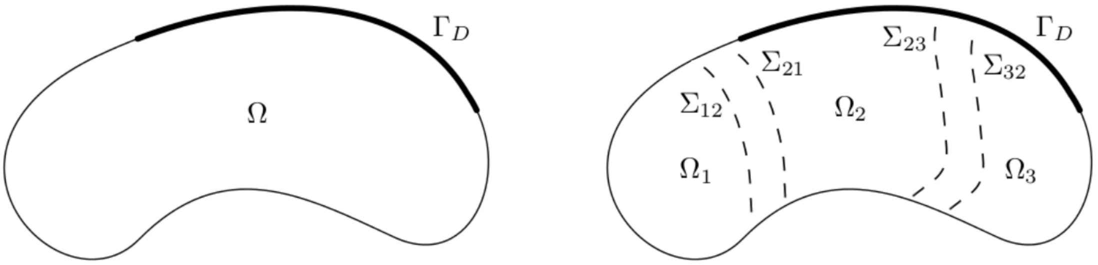


Figure 4.5: Illustration of a domain decomposed into three overlapping subdomains

$$\begin{cases} F(u) = f & \text{on } \Omega, \\ u = u_D & \text{on } \Gamma_D. \end{cases} \tag{4.3}$$

Generalising the linear alternating Schwarz algorithm 4.2.1 to more elaborate interface conditions gives the following update equation from iteration step k to $k+1$:

$$\begin{cases} F(u_i^{k+1}) = f_i & \text{on } \Omega_i, \\ u_i^{k+1} = u_{Di} & \text{on } \Gamma_{Di}, \\ (\mu\partial_{\boldsymbol{n}} + S)u_i^{k+1} = g_{ji}^k = (-\mu\partial_{\boldsymbol{n}} + S)u_j^k & \text{on } \Sigma_{ji}, \end{cases} \tag{4.4}$$

$\forall i = 1, ..., n$, where μ is a constant number, $\partial_{\boldsymbol{n}}$ is the normal derivative operator, g_{ji} is the interface data computed on Ω_j and given to Ω_i and S is an operator detailed below. The third equation guarantees that if the method converges then the overlall solution is equal to the union of the solutions on all subdomains. For μ set to zero and S to unity the third relation corresponds to imposing Dirichlet boundary conditions at the interface and (4.4) is identical to linear alternating Schwarz. In the following equations μ is set to one, leading to so-called Robin or mixed-type interface conditions. With such a choice the algorithm (4.4) is usually referred to as *optimised Schwarz* (*OS*). The optimal S operator can be easily deduced. Indeed, when considering only linear operators S, the interface data update relation at iteration $k+1$ for subdomain i

$$(\partial_{\boldsymbol{n}} + S)u_i^{k+1} = (-\partial_{\boldsymbol{n}} + S)u_j^k, \tag{4.5}$$

can be rewritten

$$(\partial_{\boldsymbol{n}} + S)(u_i^{k+1} - u_i^\infty) = (-\partial_{\boldsymbol{n}} + S)(u_j^k - u_j^\infty), \tag{4.6}$$

where u_i^∞ and u_j^∞ are the exact solutions at the Σ_{ji} interface and thus satisfy the interface data update relation as well. Renaming $u_i^{k+1} - u_i^\infty$ as e_i^{k+1}, the error on the u_i field at interface Σ_{ji} at iteration $k+1$ one gets

$$(\partial_{\boldsymbol{n}} + S)e_i^{k+1} = (-\partial_{\boldsymbol{n}} + S)e_j^k. \tag{4.7}$$

The optimal operator S is such that convergence is achieved after exactly one iteration. This implies that

$$e_i^1 = 0 \Rightarrow (-\partial_{\boldsymbol{n}} + S)e_j^0 = 0, \tag{4.8}$$

hence

$$(-\partial_{\boldsymbol{n}} + S)e_j^0 = 0 \Rightarrow \partial_{\boldsymbol{n}} e_j^0 = S\, e_j^0. \tag{4.9}$$

The optimal S operator for the considered form of interface conditions is thus the Dirichlet to Neuman operator (DtN). Because the DtN operator is nonlocal, a variety of local approximations have been derived such as Sommerfeld, complexified Sommerfeld, optimised order 0 and 2 conditions or a square root operator [110, 83]. In our case Dirichlet conditions will always be used for the electrostatic potential and mechanical displacement fields while both Dirichlet and Sommerfeld conditions will be compared for the propagating acoustic pressure waves, the only really coupled problem. The Sommerfeld condition corresponds to an S operator equal to $S = -\frac{1}{c}\frac{\partial}{\partial t}$ with c (m/s) the propagation speed in the fluid.

The Krylov-Schwarz algorithm with interface unknowns is not solved in the form (4.4) in practice, as this would lead to the same slow convergence as was observed for linear alternating Schwarz. Instead it takes advantage of the linearity of the operator F to rewrite the problem in a way such that fast linear iterative solvers can be used. Indeed due to linearity the solution u_i on every subdomain i can be decomposed as $u_i = v_i + w_i$, where v_i is the contribution of the artificial interfaces Σ_{ij} to the solution and w_i the contribution of the physical sources (f and u_D). Since from an iteration k to $k+1$ only the artificial interface data changes, w_i can be computed once and for all by solving:

$$\begin{cases} F(w_i) = f_i & \text{on } \Omega_i, \\ w_i = u_{Di} & \text{on } \Gamma_{Di}, \\ (\partial_{\boldsymbol{n}} + S)w_i = 0 & \text{on } \Sigma_{ji}. \end{cases} \tag{4.10}$$

Field v_i however does change from an iteration step to the other. Setting all physical sources to zero gives the update for v_i:

$$\begin{cases} F(v_i^{k+1}) = 0 & \text{on } \Omega_i, \\ v_i^{k+1} = 0 & \text{on } \Gamma_{Di}, \\ (\partial_{\boldsymbol{n}} + S)v_i^{k+1} = g_{ji}^k & \text{on } \Sigma_{ji}. \end{cases} \tag{4.11}$$

Let us now rewrite the interface data g update equation:

$$\begin{aligned} g_{ji}^{k+1} &= (-\partial_{\boldsymbol{n}} + S)u_j^{k+1} \\ &= (-\partial_{\boldsymbol{n}} + S)v_j^{k+1} + (-\partial_{\boldsymbol{n}} + S)w_j^{k+1} \\ &= (-\partial_{\boldsymbol{n}} + S)v_j^{k+1} + (-\partial_{\boldsymbol{n}} + S)w_j \\ &= (-\partial_{\boldsymbol{n}} + S)v_j^{k+1} + b_{ji}, \end{aligned} \tag{4.12}$$

where b_{ji} is the physical sources contribution for Ω_i computed on Ω_j. Since v_j^{k+1} is a function of g one can write the relation for the full interface data unknown vector:

$$\boldsymbol{g}^{k+1} = \mathcal{A}\boldsymbol{g}^{k} + \boldsymbol{b}, \tag{4.13}$$

where the operator $\mathcal{A}$ includes an iteration of the linear alternating Schwarz algorithm with physical sources set to 0 and Robin-type interface conditions as can be seen in (4.12). Vector $\boldsymbol{b}$ is the vector of the physical sources contribution and can be computed on every subdomain using relation $b_{ij} = (-\partial_{\boldsymbol{n}} + S)w_i$. At convergence $\boldsymbol{g}$ must be such that

$$\mathcal{F}\boldsymbol{g} \equiv (\boldsymbol{I} - \mathcal{A})\boldsymbol{g} = \boldsymbol{b}, \tag{4.14}$$

which can be solved using any linear solver, in particular a Krylov method like GMRES which requires only the matrix-free evaluation of $\mathcal{F}\boldsymbol{g}$.

Let us apply the algorithm on the reference 2D CMUT of figure 1.5 with the same quasi-linear settings as used for the linear alternating Schwarz algorithm. The overall relative residual is computed in the same way with the same initial residual to be able to compare both algorithms. Figure 4.6 shows the number of iterations required to reach a 10^{-3} overlall relative residual for both Dirichlet and Sommerfeld interface conditions. The fundamental-frequency harmonic subproblem of the elastoacoutic (2.16)-(2.38) formulation is solved. Here again a 5% overlap is used. The convergence with the current algorithm is dramatically faster and more robust compared to the linear alternating Schwarz algorithm. This holds for the whole range of frequencies of interest. With Dirichlet interface conditions the number of iterations does not significantly change with the frequency, even close to membrane resonance. A slight increase can however be observed as the frequency increases. For Sommerfeld interface conditions the convergence is faster than with Dirichlet except close to resonance. Unlike for Dirichlet the convergence speeds up as the frequency increases. In any case the lowest achievable residual is close to the one that could be achieved with the linear alternating Schwarz algorithm, except with Sommerfeld conditions close to resonance as shown in figure 4.6 (bottom). Figure 4.7 shows the convergence history for several overlap sizes with an excitation frequency of 800 kHz. For the top figure Dirichlet interface conditions are used while for the bottom one Sommerfeld conditions are used. In both cases the convergence slows down as the overlap is decreased. This behaviour is expected since the algorithm does not converge without overlap. In any case however the Sommerfeld interface conditions lead to a faster convergence. Figure 4.8 shows the convergence history for several electrical excitation frequencies with a 5% overlap. For the top figure Dirichlet interface conditions are used while for the bottom one Sommerfeld conditions are used. Unsurprisingly with Dirichlet interface conditions the convergence slows down as the frequency of the acoustic waves increases. It was already observed in [110] that Dirichlet interface conditions are not adapted for high frequency problems with wave propagation. Sommerfeld interface conditions force outgoing pressure waves and have been shown to be more adapted for wave propagation [110, 83]: the trend in figure 4.8 is opposite to what was observed for Dirichlet interface conditions. For low frequency acoustic waves the algorithm converges slower than at higher frequency. This behaviour should not surprise since below resonance the frequencies are low (below 1 MHz) and thus the wavelength in water (more than 1500 μm) is much longer than the CMUT dimensions so that the problem can be considered a low frequency problem. For the excitation frequency of 25600 kHz in the figure however the corresponding wavelength in water is 58 μm, similar to the CMUT dimension so that the problem can be considered a high frequency problem and Sommerfeld conditions are then much better adapted.

4.2.3 Krylov-Schwarz with volume unknowns

In order to solve the linear problem (4.3) for unknown u the linear alternating Schwarz algorithm introduced in section 4.2.1 uses DDM as an iterative solver with a fixed-point method, leading to slow convergence. In section 4.2.2 the linear system is rewritten to allow the usage of Krylov subspace methods such as GMRES. This leads to a dramatically faster convergence. In this section DDM is used as an algebraic preconditionner $\boldsymbol{M}$ to solve the finite-element discretised algebraic problem $\boldsymbol{Ax} = \boldsymbol{b}$, corresponding to the continuous problem in (4.3), in its preconditionned form $\boldsymbol{M}^{-1}\boldsymbol{Ax} = \boldsymbol{M}^{-1}\boldsymbol{b}$ with an efficient iterative solver. The Krylov-Schwarz algorithm with volume unknowns in the DDM iteration, usually referred to as *additive Schwarz* is a classical algorithm for linear problems.

Consider a domain decomposed into n overlapping subdomains as in figure 4.5. Let $\boldsymbol{R}_i$ be the restriction matrix for the *ith* subdomain. Applying $\boldsymbol{R}_i$ to a vector with elements on the whole domain gives a vector $\boldsymbol{x}_i$ of smaller size with only the elements of the vector that are on subdomain i; applying its transpose (called extension matrix) to a vector with elements only on subdomain i gives a vector with zero elements on the whole domain except on subdomain i. As an example for a vector $\boldsymbol{x}$ defined on the whole domain with five entries x_1, x_2, x_3, x_4, x_5 the restriction to a subdomain including only x_1, x_3 and x_4 is done with the restriction matrix detailed in (4.15): $\boldsymbol{R}_i$ is a matrix with a number of rows equal to the number of subdomain unknowns and made up of only zeros and ones. Applying the extension matrix (i.e. the transpose of the restriction matrix) on the subvector as shown in (4.16) returns a vector defined on the whole domain but with only the contribution of the subdomain.

$$\begin{bmatrix} x_1 \\ x_3 \\ x_4 \end{bmatrix} = \boldsymbol{R}_{134}\boldsymbol{x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} \tag{4.15}$$

$$\begin{bmatrix} x_1 \\ 0 \\ x_3 \\ x_4 \\ 0 \end{bmatrix} = \boldsymbol{R}_{134}^T\boldsymbol{R}_{134}\boldsymbol{x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ x_4 \end{bmatrix} \tag{4.16}$$

The additive Schwarz method consists in solving the original (linear) system $\boldsymbol{Ax} = \boldsymbol{b}$ preconditonned by $\boldsymbol{M}_{AS}^{-1}$ with e.g. GMRES, a Krylov subspace method:

$$\boldsymbol{M}_{AS}^{-1}\boldsymbol{Ax} \equiv \sum_{i=1}^{N} \boldsymbol{R}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i \boldsymbol{Ax} = \sum_{i=1}^{N} \boldsymbol{R}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i \boldsymbol{b} \equiv \boldsymbol{M}_{AS}^{-1}\boldsymbol{b}, \tag{4.17}$$

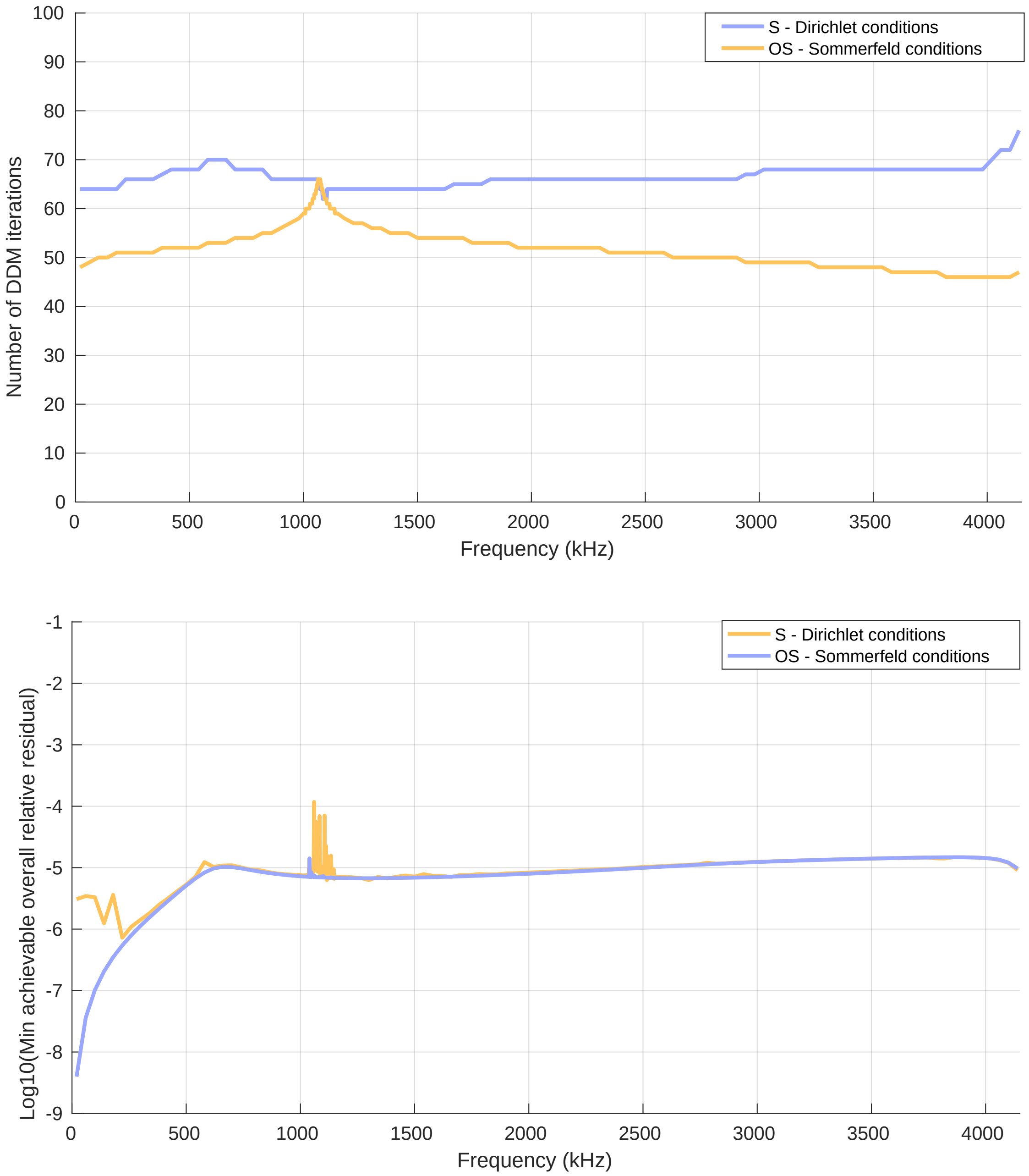


Figure 4.6: Number of GMRES iterations to reach a 10^{-3} overall relative residual (top) and lowest achievable overall relative residual (bottom) versus frequency for the fundamental-frequency harmonic subproblem of the elastoacoutic formulation in case of Dirichlet and Sommerfeld interface conditions.

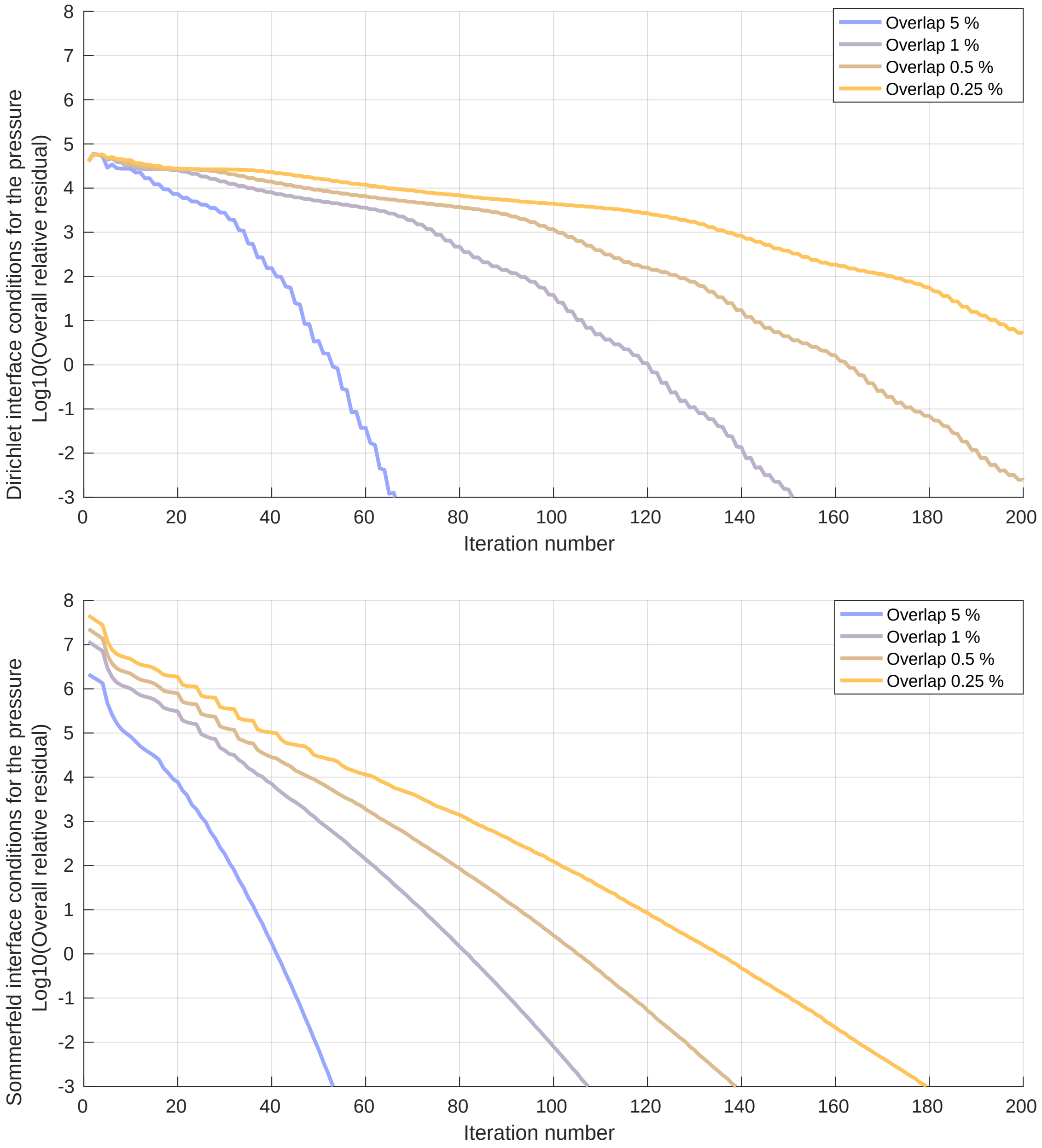


Figure 4.7: Convergence history for the fundamental frequency subproblem of the elastoacoutic formulation for a varying subdomain overlap. Interface conditions are Dirichlet (top) or Sommerfled (bottom). The algorithm used is Krylov-Schwarz with interface unknowns.

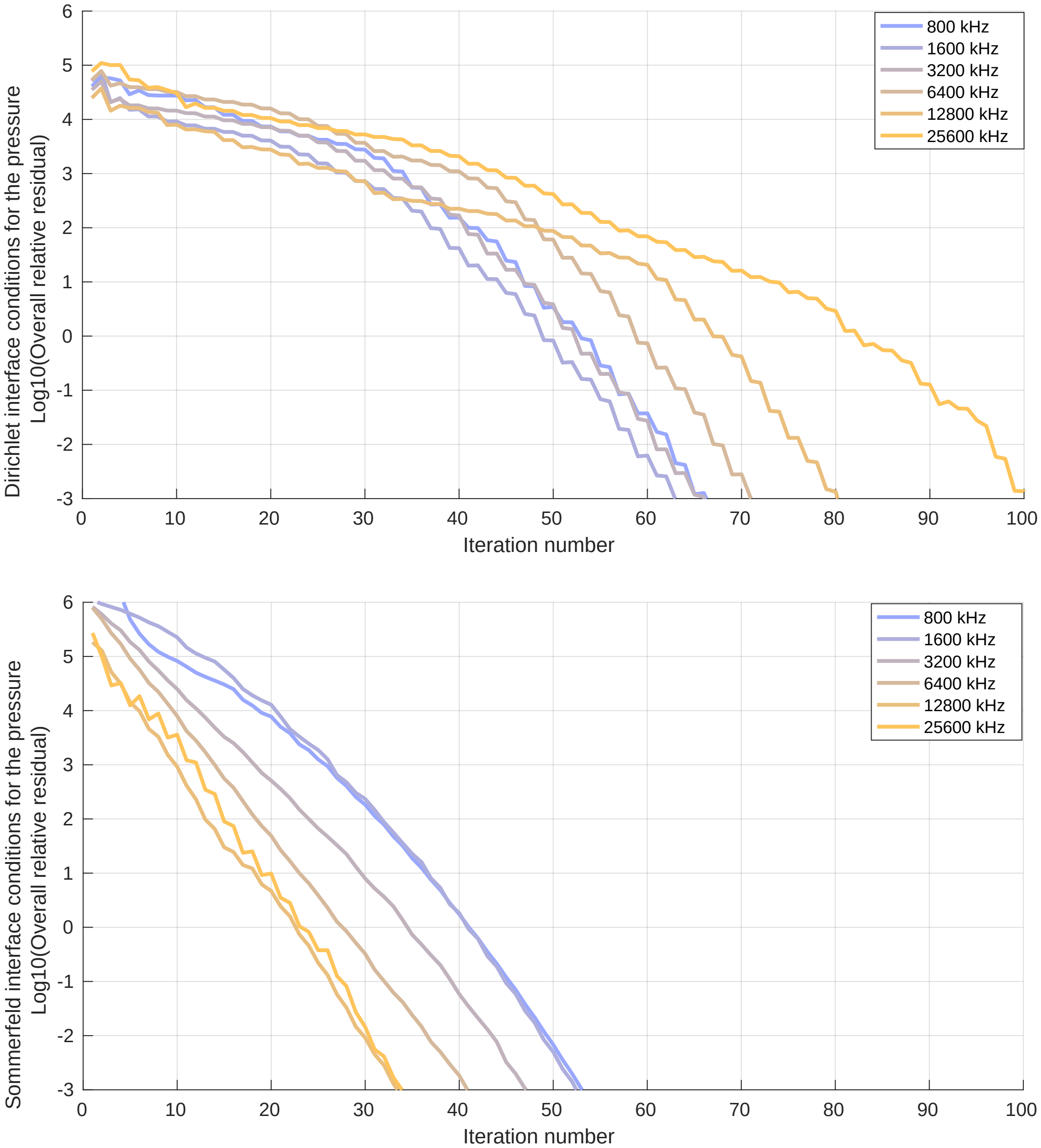


Figure 4.8: Convergence history for the fundamental frequency subproblem of the elastoacoutic formulation for a varying electrical excitation frequency. Interface conditions are Dirichlet (top) or Sommerfled (bottom). The algorithm used is Krylov-Schwarz with interface unknowns.

for N subdomains, where $\boldsymbol{A}_i$ is the part of the global matrix $\boldsymbol{A}$ corresponding to subdomain i, i.e. $\boldsymbol{A}_i = \boldsymbol{R}_i \boldsymbol{A} \boldsymbol{R}_i^T$. As can be seen, the unknown vector $\boldsymbol{x}$ contains volumic unknowns, unlike in the algorithm in section 4.2.2 where it only contained the (fewer) unknowns at the subdomain interfaces.

Applying $\boldsymbol{M}_{AS}^{-1}$ to a vector in the original additive Schwarz algorithm consists in applying $\boldsymbol{R}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i$ to the vector subdomain per subdomain and summing all subdomain contributions on the overlaps. It has been shown [111] that a faster convergence can be achieved using the restricted additive Schwarz variant introduced in [112] (reffered to by the acronym *RAS*). In this variant the contributions are not simply added together but are instead first restricted in a 0-overlap disjoint fashion (i.e. everything outside of the restricted disjoint subdomain is zeroed) before being summed together. The preconditionner then becomes:

$$\boldsymbol{M}_{RAS}^{-1} = \sum_{i=1}^{N} \tilde{\boldsymbol{R}}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i, \tag{4.18}$$

where $\tilde{\boldsymbol{R}}_i$ is the same as $\boldsymbol{R}_i$ except that it corresponds to a disjoint partitionning of the subdomains with fewer rows. The restricted variant will be used throughout this thesis.

Similarly as in section 4.2.2 more elaborate interface conditions can be used for a faster DDM convergence [113]. To use the same Sommerfeld interface conditions as in section 4.2.2 the only thing to change in the restricted additive Schwarz algorithm is the $\boldsymbol{A}_i^{-1}$ solve step: $\boldsymbol{A}_i^{-1}$ is no more equal to $\boldsymbol{A}_i = \boldsymbol{R}_i \boldsymbol{A} \boldsymbol{R}_i^T$. Applying $\boldsymbol{A}_i^{-1}$ to a vector $\boldsymbol{r}_i$ (corresponding to r at the continuous level) on subdomain i now corresponds to solving the following problem [114]:

$$\begin{cases} F(u_i) = r_i & \text{on } \Omega_i, \\ u_i = u_{Di} & \text{on } \Gamma_{Di}, \\ \partial_{\boldsymbol{n}} u_i + S(u_i) = 0 & \text{on } \Sigma_{ji}, \end{cases} \tag{4.19}$$

where S is the Sommerfeld operator used in section 4.2.2, that is $S(u) = -\frac{1}{c}\frac{\partial u}{\partial t}$ with c (m/s) the propagation speed in the fluid. Unlike in the previous algorithm the $\partial_{\boldsymbol{n}} u + S(u)$ term computed on the neighbouring subdomains is not required here which makes for a simpler implementation. With such interface conditions the algorithm is usually called *optimised restricted additive Schwarz* (*ORAS*).

Let us test the performance of the algorithm on the reference 2D CMUT of figure 1.5 with the same linear settings as in the previous section and with the same definition of the overall relative residual and same initial residual. Figure 4.9 shows the number of iterations required to reach a 10^{-3} overall relative residual for the fundamental-frequency harmonic subproblem of the elastoacoutic (2.16)-(2.38) formulation when considering RAS as well as ORAS with Sommerfeld interface conditions. A 5% overlap is used. The convergence is slightly faster than for the Krylov-Schwarz algorithm with interface DDM unknowns, in particular without Sommerfeld conditions as can be seen when comparing to figure 4.6. Furthermore RAS converges faster closer to resonance than ORAS but its iteration count increases as the frequency increases while it is the opposite for ORAS. Unlike the previous algorithm the current algorithm can not achieve an as small overall relative residual close to resonance as shown in figure 4.9 (bottom). This will have an impact when considering Newton's method to solve the nonlinear electroelastoacoustic problem since it tends to be poorly conditionned. Figure 4.10 shows the convergence history for several overlap sizes with an excitation frequency of 800 kHz. For the top figure the classical RAS algorithm is used while for the bottom one ORAS with Sommerfeld conditions is used. In the second case the convergence clearly slows down as the overlap is decreased. The ORAS algorithm with Sommerfeld conditions does not converge without overlap. For the classical RAS algorithm however the top figure shows a very limited increase in the iteration count as the overlap size decreases. For the CMUT application considered RAS converges even for minimal overlap which is an interesting property since decreasing the overlap size can decrease the number of unknowns in the subdomains. Figure 4.11 shows the convergence history for several electrical excitation frequencies with a 5% overlap. Unsurprisingly the behaviour is similar as for the previous algorithm in section 4.2.2: with Dirichlet interface conditions the convergence slows down as the frequency of the acoustic waves increases. For the Sommerfeld interface conditions, the opposite happens. For low frequency acoustic waves the algorithm converges slower than at higher frequency.

4.2.4 Choice of the linear domain decomposition method

In this section three main DDMs were detailed to efficiently solve a linear problem in parallel. It was observed that the alternating Schwarz algorithm leads to a slow convergence. This is mainly due to the fact that a fixed point resolution is used to solve the problem iteratively. For the Krylov-Schwarz algorithm with interface and with volume unknowns in the DDM vectors however, a linear problem was obtained and solved with the Krylov subspace method GMRES. This has dramatically improved convergence and alternating Schwarz will thus be disregarded as a building block for nonlinear DDM algorithms in the next section.

An analysis of the electrical excitation frequency impact on the convergence rate has shown that the optimised algorithms with the Sommerfeld interface conditions systematically exhibit a faster convergence rate than their non-optimised counterparts, except close to resonance. For lower-than-resonance electrical excitation frequencies the difference is limited while it becomes increasingly visible for higher frequencies. An analysis of the overlap size has shown a slowdown in the convergence rate for all algorihms but RAS. Unsurprisingly for the latter algorithm fast convergence could still be achieved for a small overlap [115].

Because the frequencies of interest are around the first resonance frequency (about 1 MHz) the convergence rate is either not improved or only slightly improved by the use of optimised interface conditions. Taking into account the difficulty to implement them on a 3D CMUT array it can be a resonable choice to use only Dirichlet interface conditions.

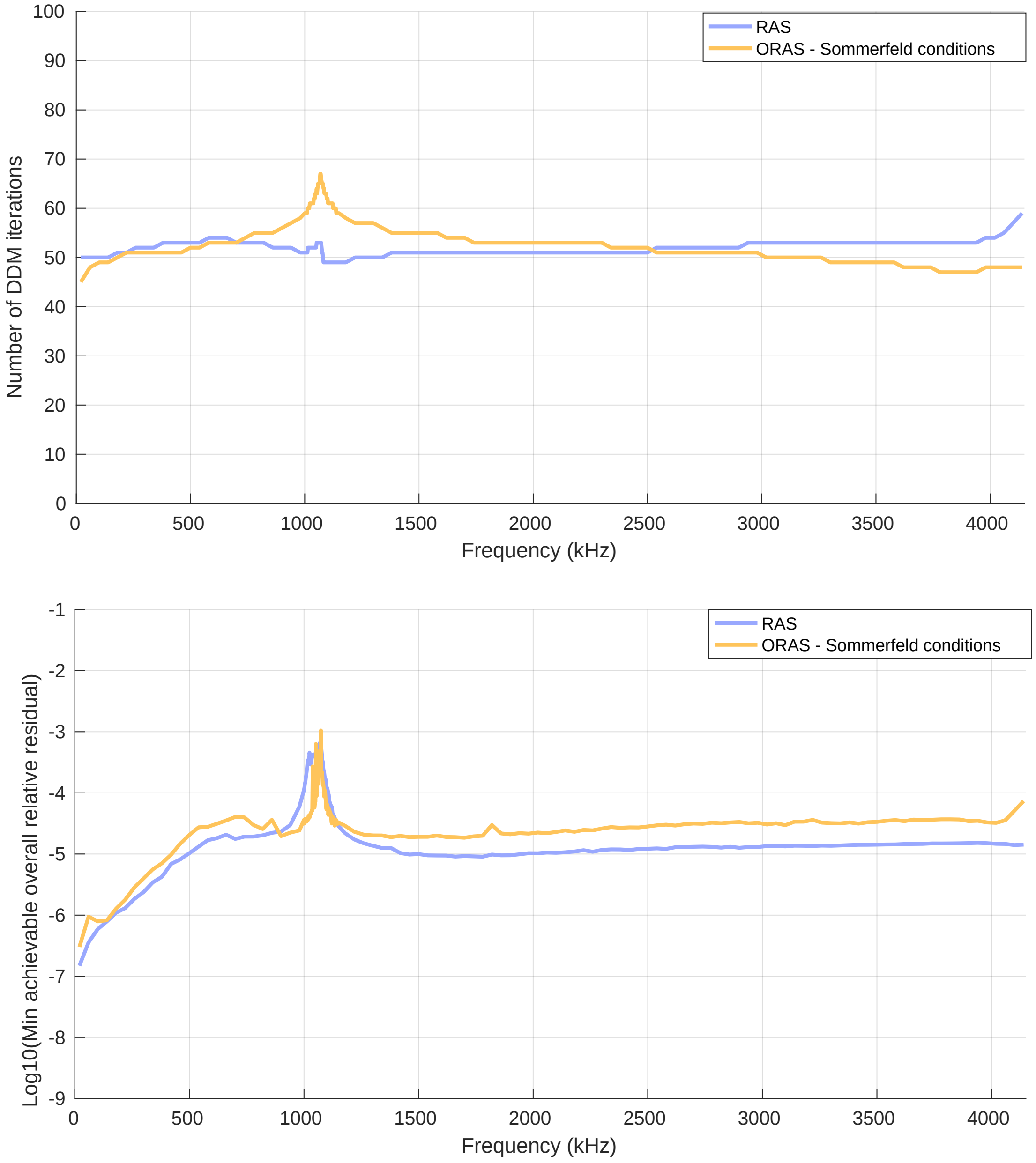


Figure 4.9: Number of GMRES iterations to reach a 10^{-3} overall relative residual (top) and lowest achievable overall relative residual (bottom) versus frequency for the fundamental-frequency harmonic subproblem of the elastoacoutic formulation with algorithm RAS and ORAS.

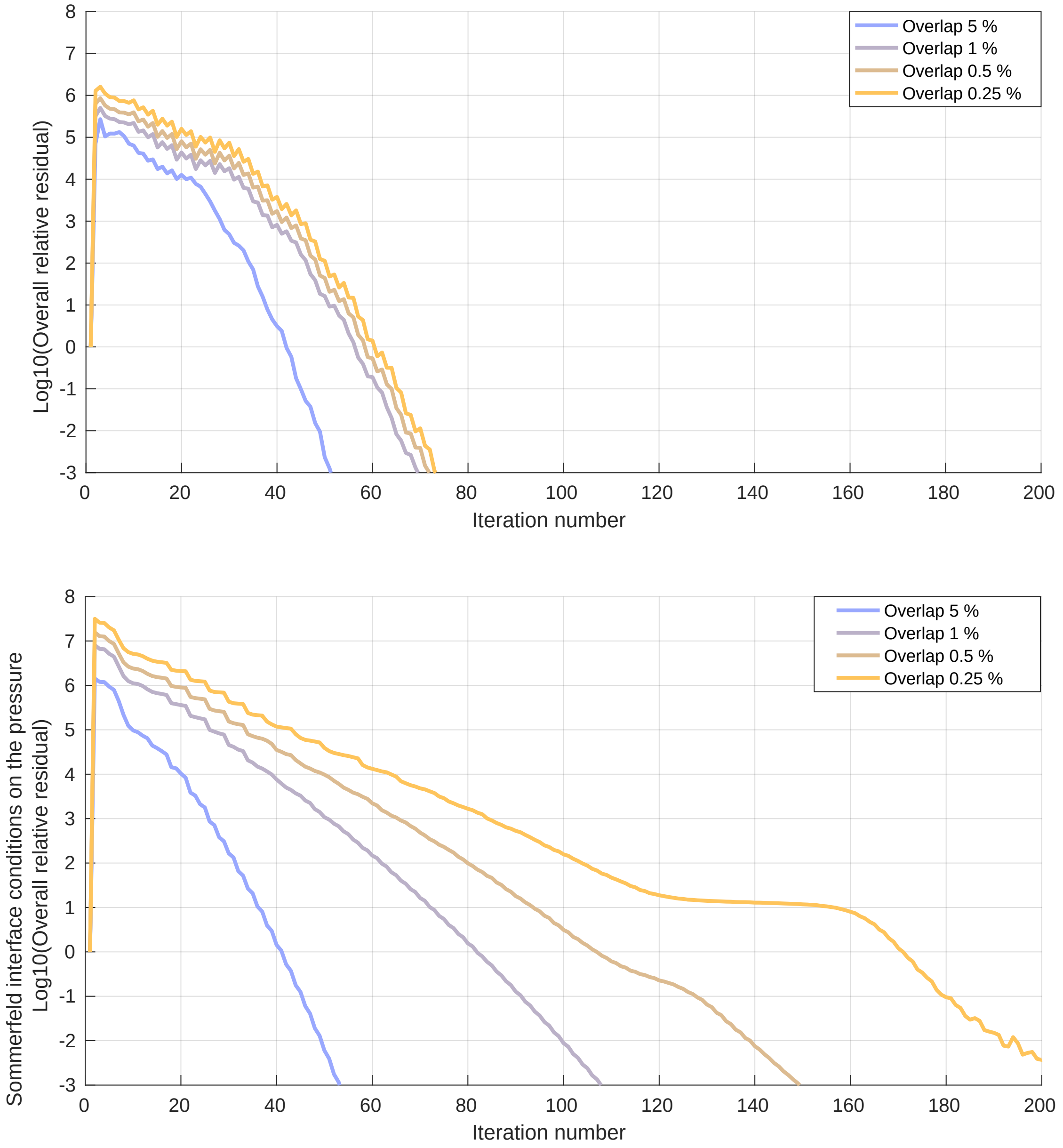
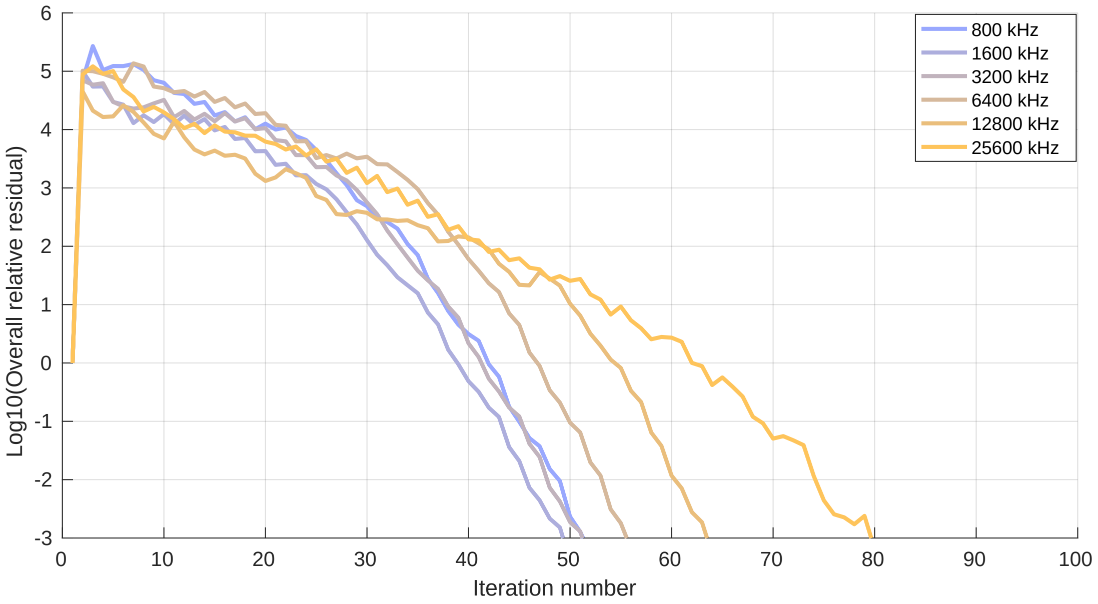


Figure 4.10: Convergence history for the fundamental frequency subproblem of the elastoacoutic formulation for a varying subdomain overlap. Interface conditions are Dirichlet (top) or Sommerfled (bottom). The algorithm used is Krylov-Schwarz with volume unknowns (RAS and ORAS).



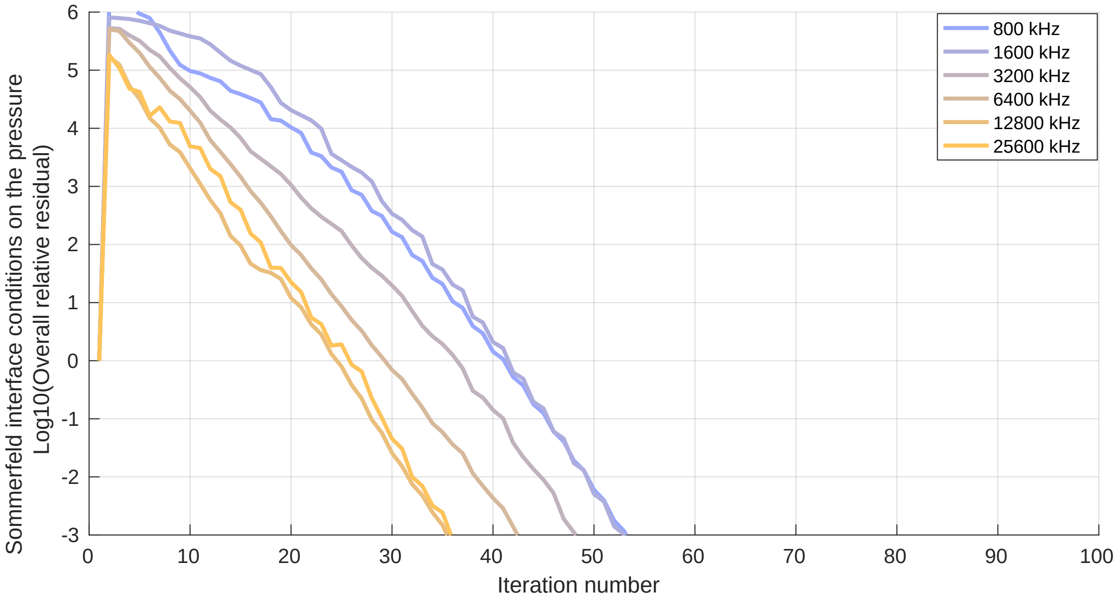


Figure 4.11: Convergence history for the fundamental frequency subproblem of the elastoacoutic formulation for a varying electrical excitation frequency. Interface conditions are Dirichlet (top) or Sommerfled (bottom). The algorithm used is Krylov-Schwarz with volume unknowns (RAS and ORAS).

4.3 Nonlinear problems

Using DDM to solve the nonlinearly coupled electroelastoacoutic formulation (2.6)-(2.16)-(2.38) corresponding to the nonlinear problem

$$F(u) = 0 \quad \text{on } \Omega \tag{4.20}$$

can be achieved with a combination of the Newton, Krylov and Schwarz methods. The nonlinear DDMs can be classified into two categories. The first category includes the methods that result in the direct application of the algorithms presented in section 4.2.2 and 4.2.3 to the linear system obtained at every staggered or Newton iteration of the nonlinear problem. The ***staggered*** and *Newton-Krylov-Schwarz* DDM algorithms listed in figure 4.12 are in the first category. The second category includes all other algorithms in the figure. They share the property of being nonlinearly preconditionned: a full nonlinear resolution step is added at every iteration to include as soon as possible the nonlinearity. For a rather extensive discussion of nonlinear DDMs refer to [116, 117].

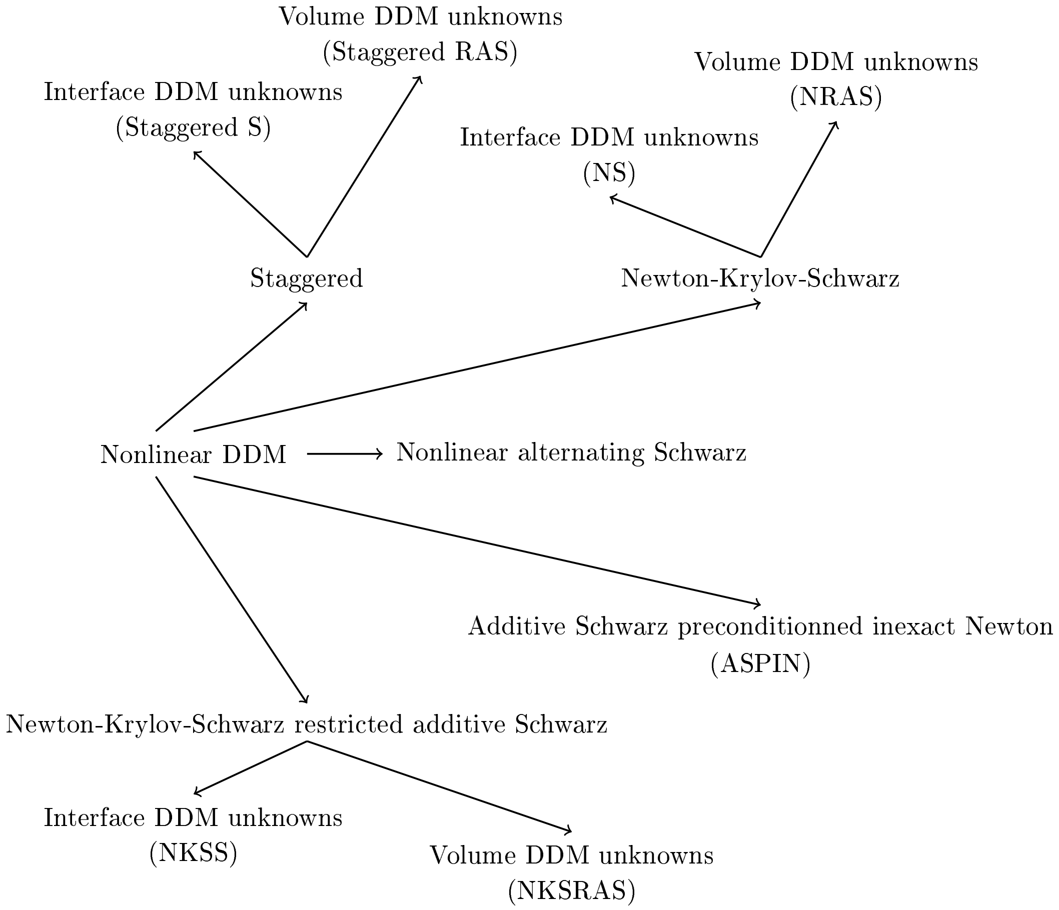


Figure 4.12: Nonlinear domain decomposition algorithms considered along with their acronyms. The algorithms are considered with domain decomposition unknowns taken either on the subdomain interfaces or on their volume (if applicable).

4.3.1 Staggered Krylov-Schwarz (*Staggered S* and *Staggered RAS*)

The staggered Krylov-Schwarz algorithm [118] is a DDM that can be used when the nonlinearity originates from the combination of multiple linear physics as in the electroelastoacoutic case in this paper. It is based on the Krylov-Schwarz algorithm described previously for linear formulations, with DDM unknowns taken either on the subdomain interfaces as in section 4.2.2 or on their volumes as in section 4.2.3. In the first case the algorithm will be referred to by *staggered S* while in the second case it will be referred to by *staggered RAS*. Their optimised counterparts are then named *staggered OS* and *staggered ORAS* respectively.

In the electroelastoacoutic case the *staggered S* (*staggered RAS*) algorithm simply consists in using the linear Krylov-Schwarz algorithm in alternance on the uncoupled linear electrostatic (2.6) and linear elastoacoutic formulations (2.16)-(2.38) until convergence. It shares the advantages of the staggered resolution scheme described in 2.4.1, that is all frequencies in the elastoacoustic formulation can be solved for independently. Furthermore it is lighter and faster to solve when the nonlinearity is not too strong compared to the Newton algorithm used in the coming sections. When close to pull-in the overhead of the Newton resolution is overcompensated by the decrease in the iteration count. Figure 4.13 shows the number of nonlinear iterations required to reach a 10^{-4} relative nonlinear residual versus frequency for a $40{+}40{\cdot}sin(2\pi f_0 t)$ V electrical excitation on the left membrane of the 2 by 1 2D CMUT array of figure 1.5 and 40 V on the right one. Results are shown for both the *staggered S* and *staggered RAS* algorithms as well as their optimised counterparts with Sommerfeld interface conditions. The figure clearly shows the large number of nonlinear iterations required by the staggered DDM algorithm when the nonlinearity becomes stronger closer to the resonance frequency (about 1 MHz). When too close to resonance the membrane deflects beyond pull-in and there is no more physical solution to the problem.

In combination with figure 4.15 (showing the minimum achievable overall relative DDM residual versus nonlinear iteration for a 800 kHz electrical excitation) it appears that in the vicinity of the resonance peak the number of nonlinear iterations to reach the 10^{-4} relative nonlinear residual is impacted by the accuracy to which the DDM solution can be computed. In the staggered case however low enough DDM residuals are reached and this effect has only a very limited impact. Finally figure 4.14 shows the number of DDM iterations required to reach a 10^{-3} overall relative DDM residual versus the staggered nonlinear iteration number. An electrical excitation frequency of 800 kHz is used. The 4 algorithms compare to each other similarly to what has been observed in sections 4.2.2 and 4.2.3, that is Krylov-Schwarz with volume unknowns (i.e. RAS) converges faster than with interface unknowns and OS and ORAS converge slightly faster than their nonoptimised equivalent. The new information of the figure is how the number of DDM iterations is impacted by the nonlinearity: no impact is observed, even for this close-to-resonance, nonlinear test case.

4.3.2 Newton-Krylov-Schwarz (*NS* and *NRAS*)

The Newton-Krylov-Schwarz algorithm [119, 120, 121] applies the Krylov-Schwarz algorithm with interface or volume unknowns described previously to the Newton algorithm described in 2.4.2. In case the Krylov-Schwarz algorithm with interface unknowns is used the algorihm will be referred to by the acronym *NS* while it be *NRAS* in case of volume unknowns. NS or NRAS can be preferred

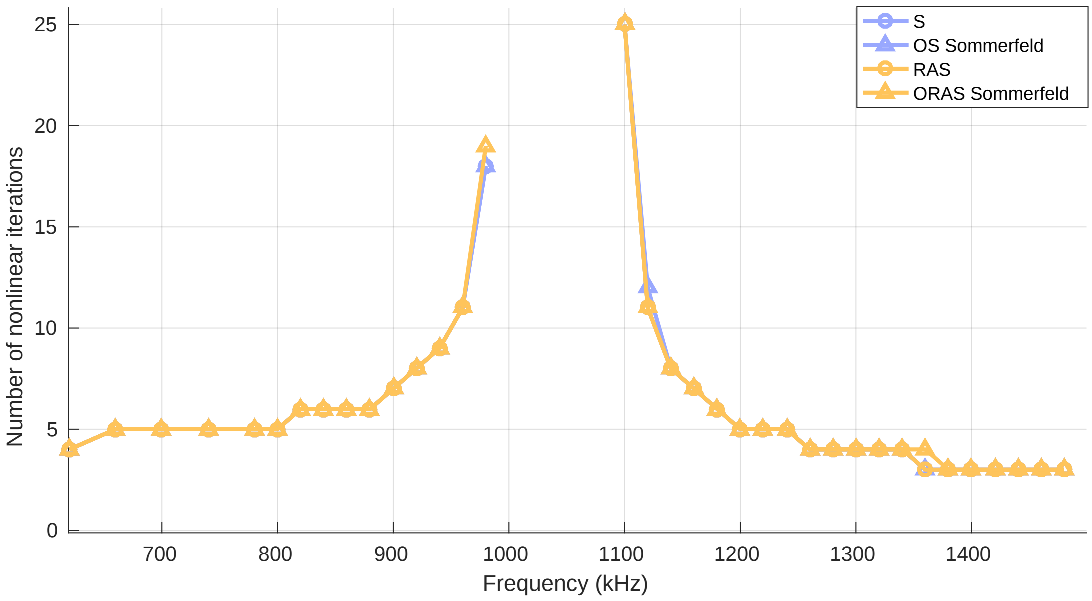


Figure 4.13: Number of nonlinear iterations to reach a 10^{-4} overall nonlinear relative residual versus frequency for the electroelastoacoutic problem solved with KS, RAS and their optimised variant at every nonlinear staggered iteration.

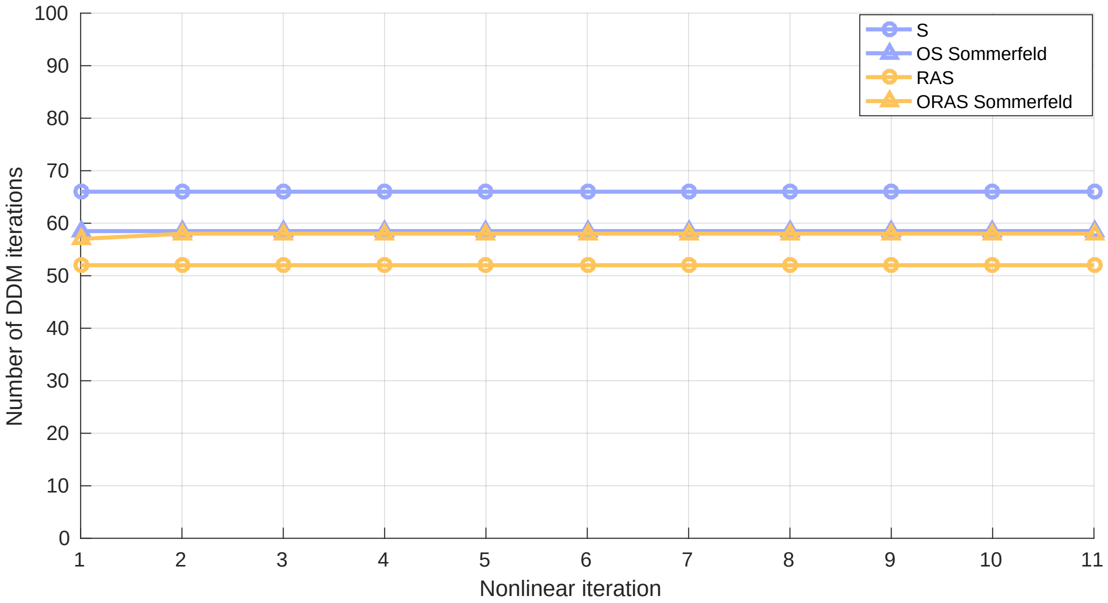


Figure 4.14: Number of domain decomposition iterations to reach a 10^{-3} overall relative residual for the fundamental-frequency harmonic subproblem of the elastoacoustic problem versus staggered nonlinear iteration.

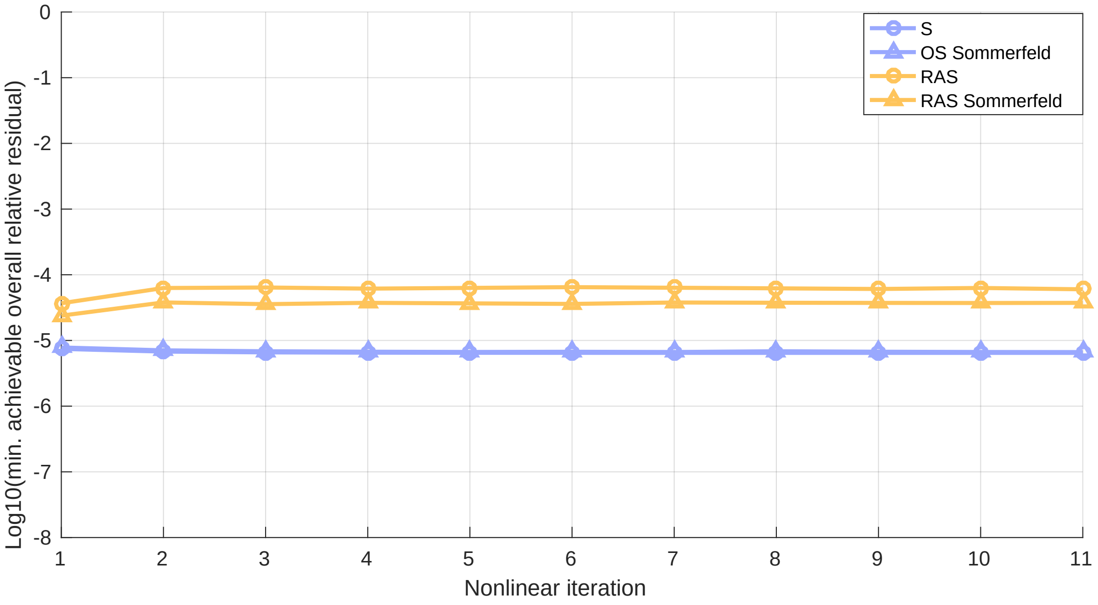


Figure 4.15: Minimum achievable overall relative residual for the fundamental-frequency harmonic subproblem of the elastoacoustic problem versus staggered nonlinear iteration.

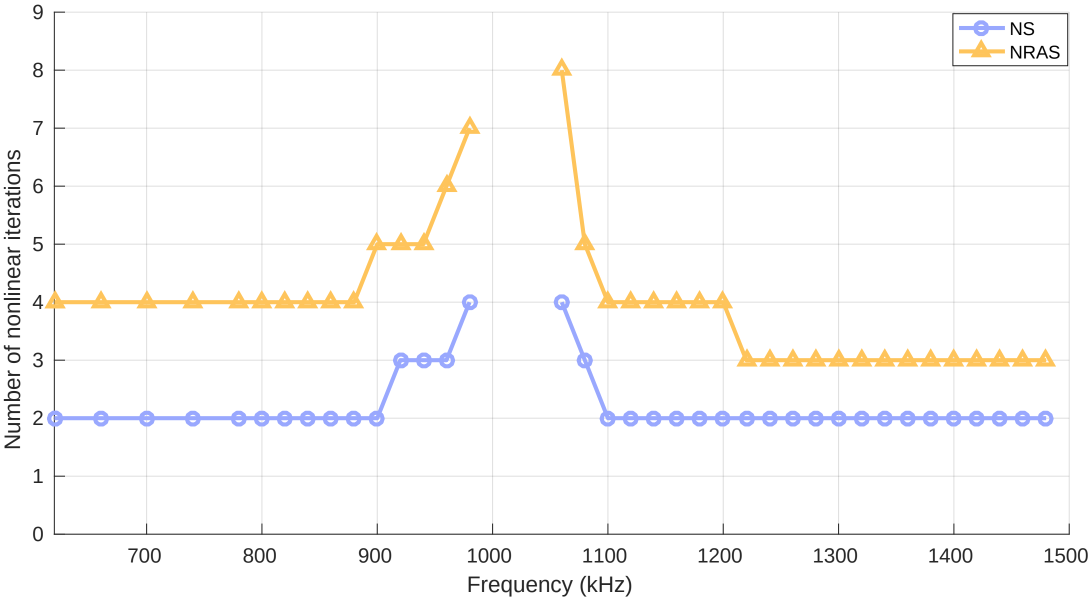


Figure 4.16: Number of nonlinear iterations to reach a 10^{-4} overall nonlinear relative residual versus frequency for the electroelastoacoutic formulation solved with algorithm NS and NRAS.

to the staggered Krylov-Schwarz algorithm when close to pull-in. In that case the associated Newton iteration overhead can be compensated by a dramatic decrease in the iteration count as already pointed out in figure 2.2. Unlike the staggered algorithm this algorithm does not solve the electrostatic (2.6) and elastoacoutic (2.16)-(2.38) formulations in alternance. It solves instead the three physics at once with a Newton iteration as described in section 2.4.2.

Consider the nonlinear finite element discretised problem $\boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} = \boldsymbol{b}(\boldsymbol{x})$. At every nonlinear iteration of Newton's method the following linear formulation has to be solved:

$$\boldsymbol{J}\boldsymbol{dx} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}, \tag{4.21}$$

where $\boldsymbol{J}$ is the Jacobian matrix for Newton's method (detailed in section 2.2.5 for the electroelastoacoustic problem) and $\boldsymbol{dx}$ the correction on the unknown displacement, pressure and electric field. Because this is a linear problem one can use the linear Krylov-Schwarz method, with interface unknowns as in section 4.2.2 or with volumic unknowns as in section 4.2.3. The implementation with volumic unknowns is straightforward. Care has to be taken however for the interface unknown variant when deciding what the physical sources are. Rewriting Newton's iteration as

$$\begin{aligned} & \boldsymbol{J}\boldsymbol{dx} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \\ \Longleftrightarrow\ & \boldsymbol{J}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^k \\ \Longleftrightarrow\ & \boldsymbol{J}\boldsymbol{x}^{k+1} = \boldsymbol{b} + (\boldsymbol{J} - \boldsymbol{A})\boldsymbol{x}^k, \end{aligned} \tag{4.22}$$

makes clear that an extra $(\boldsymbol{J} - \boldsymbol{A})\boldsymbol{x}^k$ physical source adds to the other physical sources. Note that setting $(\boldsymbol{J} - \boldsymbol{A})\boldsymbol{x}^k$ to zero when setting the physical sources to zero does not mean all computations must be performed on the undeformed mesh.

With the same settings as in section 4.3.1 figure 4.16 shows the number of nonlinear iterations required to reach a 10^{-4} nonlinear relative residual versus frequency for NS and NRAS. The figure clearly shows the dramatic nonlinear convergence speedup versus the staggered algorithm of section 4.3.1: for the considered test frequencies up to 10 times fewer nonlinear iterations are now required. Unlike for the staggered algorithm, the number of nonlinear iterations exhibits a strong dependence on the DDM algorithm used because the DDM can not solve as accurately the linear system (4.21) as it could solve the better conditionned problems in the staggered DDM algorithm of section 4.3.1. As a matter of facts the nonlinear convergence is faster for NS than for NRAS. This can be related to the fact that Krylov-Schwarz with interface unknown can provide a more accurate solution than with volume unknowns as was observed in figures 4.6 and 4.9.

4.3.3 Nonlinear alternating Schwarz algorithm

The nonlinear alternating Schwarz algorithm is the full-nonlinear extension of the linear alternating Schwarz algorithm of section 4.2.1. It consists in an outer and an inner iteration loop. In the outer iteration the field value on the neighbours of subdomain i is set as Dirichlet condition on the outer interfaces of subdomain i for every subdomain $i = 1...n$. The Dirichlet conditions are then kept constant in the inner iteration which consists in solving the full nonlinear problem $F(u) = 0$ on every subdomain Ω_i independently as follows

$$\begin{cases} F(u_i^{k+1}) = & 0 \text{ on } \quad \Omega_i, \\ \quad u_i^{k+1} = & u_j^k \text{ on } \quad \Sigma_{ji} \ \forall \text{neighbour } j, \end{cases} \tag{4.23}$$

for the kth outer iteration. In [116] the slow DDM convergence of this method was pointed out. This is confirmed in figure 4.17 for the 2D reference CMUT of figure 1.5 with same settings as in the previous nonlinear DDM section. This was expected because similarly to the linear alternating Schwarz algorithm the DDM is used as an iterator, not as a preconditionner. Additionally the figure and a previous study [122] underline convergence issues with this method close to resonance, a working mode often used in MEMS. Another downside of this algorithm is that the LU decomposition of the algebraic problem resulting from the finite element approximation of the continuous problem solved at every inner Newton iteration can not be reused, unlike in the staggered Krylov-Schwarz and Newton-Krylov-Schwarz algorithm families.

This algorithm has however the advantage of requiring a number of inner nonlinear iterations on every subdomain that adapts to the degree of nonlinearity. In case of an array of cross-talking MEMS cells with a single cell excited it might well be that only a single subdomain has strong nonlinearity, thus only this cell would get many inner nonlinear iterations. This is an interesting property, yet taking advantage of it might not be straightforward as it can be difficult to exactly predict where nonlinearity will be strong in case of resonance.

4.3.4 Newton-Krylov-Schwarz restricted additive Schwarz (*NKSS* and *NKSRAS*)

Newton-Krylov-Schwarz restricted additive Schwarz (called NKS-RAS in the literature) adds to the NKS or NRAS algorithm of section 4.3.2 a full-nonlinear resolution step at every nonlinear iteration. Working details are provided in algorithm 4.1 for a nonlinear problem to solve $F(u) = 0$, but one iteration can be summarised as:

- solve the nonlinear subproblem on every subdomain with the fields at the outer subdomain interfaces constrained to the field value on the neighbour subdomains

- take the union of the 0-overlap disjoint restriction of the nonlinear solution on every subdomain

- perform a single NKS or NRAS iteration while considering the previously computed field union on all subdomains as the initial guess

In case NKS is used in the last step the NKSRAS algorithm will be called *NKSS* while it will be called *NKSRAS* in case NRAS is considered.

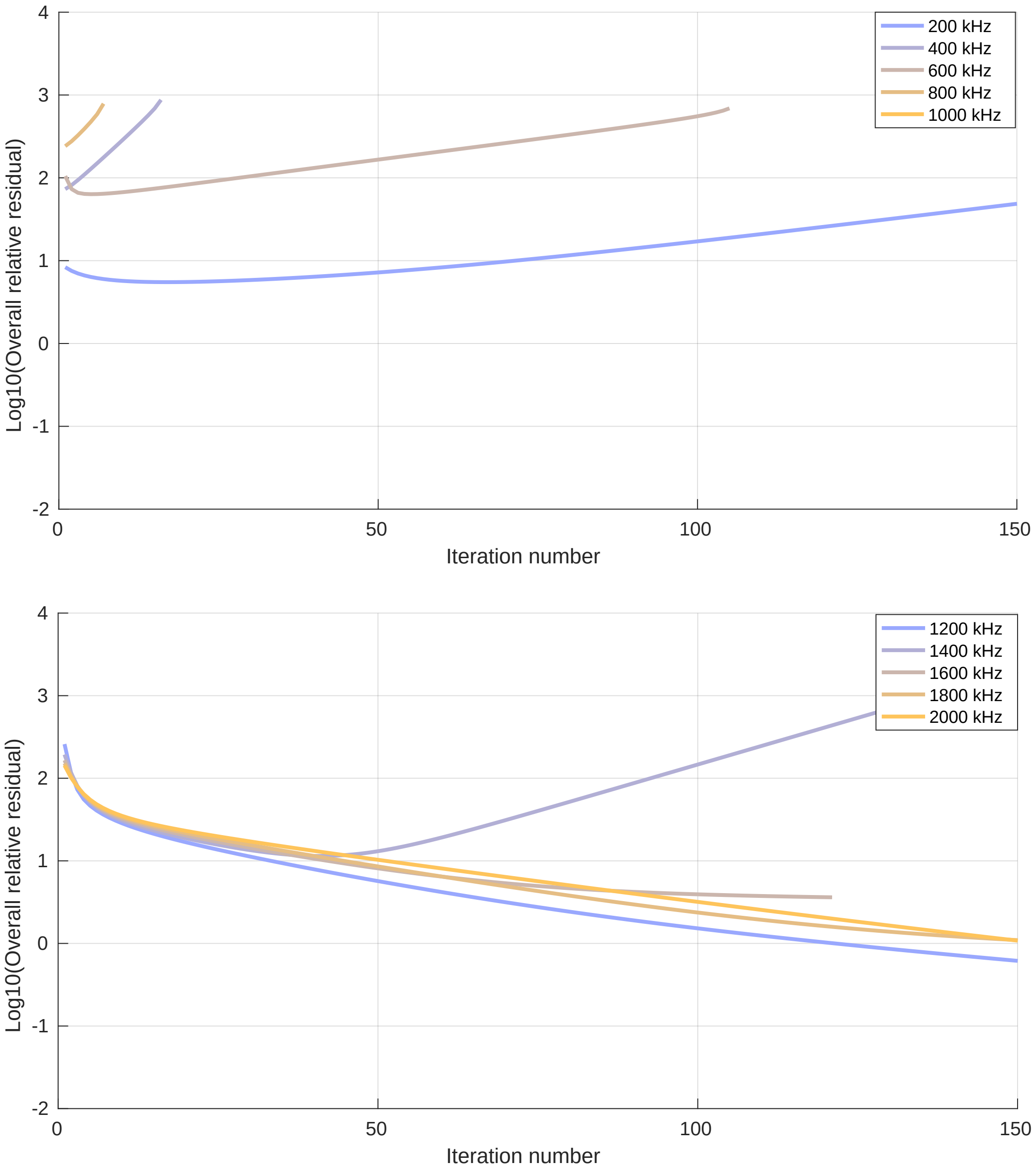


Figure 4.17: Overall relative residual versus iteration number for the nonlinear iterative Schwarz algorithm applied to the electroelastoacoutic formulation.

Algorithm 4.1 NKS-RAS algorithm

```
$u^0 = 0$;
while stopping criterion on the overall nonlinear relative residual not reached do
    for i=1...n do
        Find $u_i^k$ such that
        $$\begin{cases} F_{\Omega_i}(u_i^k) = & 0 \text{ on } & \Omega_i \\ u_i^k = & u_j^{k-1} \text{ on } & \Sigma_{ji} \ \forall \text{ neighbour } j \end{cases}$$
        using Newton's method
    end for
    $u^k := \sum_{i=1}^{n}$0-Overlap disjoint restriction($u_i^k$)
    Update $u^k$ with one step of the NKS or NRAS algorithm with initial guess $u^k$
end while
```

In this algorithm the LU decomposition can be reused during the NKS or NRAS iteration, unlike for the nonlinear classical Schwarz algorithm, which dramatically speeds up every iteration. The preliminary nonlinear resolution has the ability to reduce the number of outer iterations required compared to a standard NKS or NRAS method. This comes at the expense of an increased computational cost for every outer iteration, which however does not require more communication between processing units. In practice the algorithm will be of interest if it can achieve a low enough number of outer iterations.

Figure 4.18 shows the number of outer iterations of the NKSS and NKSRAS algorithms versus frequency around the first resonance while figure 4.19 shows for the same frequency sweep how the number of inner nonlinear iterations change on the left and right subdomains. As before the left and right subdomains have a constant 40 V electrical excitation but only the left subdomain has an extra 40 V alternating voltage. Two main observations can be made. Firstly in this CMUT problem NKSS and NKSRAS are not converging faster from a nonlinear residual point of view than their NKS and NRAS counterparts. This is essentially due to the added residual coming from the jump introduced at the fluid interface by the 0-overlap restriction step. The second observation is that right after the region around resonance (in the vicinity of the resonance peak there is no physical solution since the membrane is collapsed to the ground electrode) the algorithm does not converge, even though NKS and NRAS were converging. To recover convergence one has to move a bit further away from resonance: the convergence range around resonance seems to be more limited than it was for NKS or NRAS. The cause of this effect can be seen in figure 4.19. When decreasing the frequency down to the resonance the number of inner non linear iterations explodes well before the actual resonance frequency. This is because the full nonlinear problem is solved at the first outer iteration with 0 Dirichlet boundary conditions, which does not correspond to the actual solution. The inner nonlinear problem to solve, with 0 Dirichlet boundary condition, has a different resonance frequency than the actual problem. In the NKS or NRAS algorithms this was not a problem since the nonlinearity was incorporated more progressively in the DDM solution.

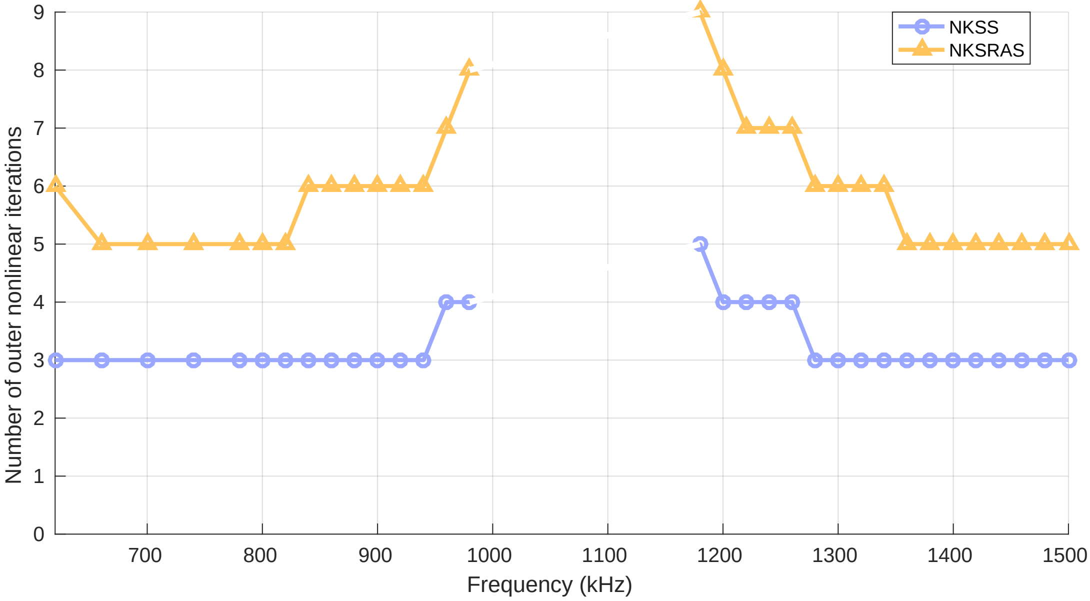


Figure 4.18: Number of outer nonlinear iterations to reach a 10^{-4} overall nonlinear relative residual versus frequency for the electroelastoacoutic formulation solved with algorithm NKSS and NKSRAS.

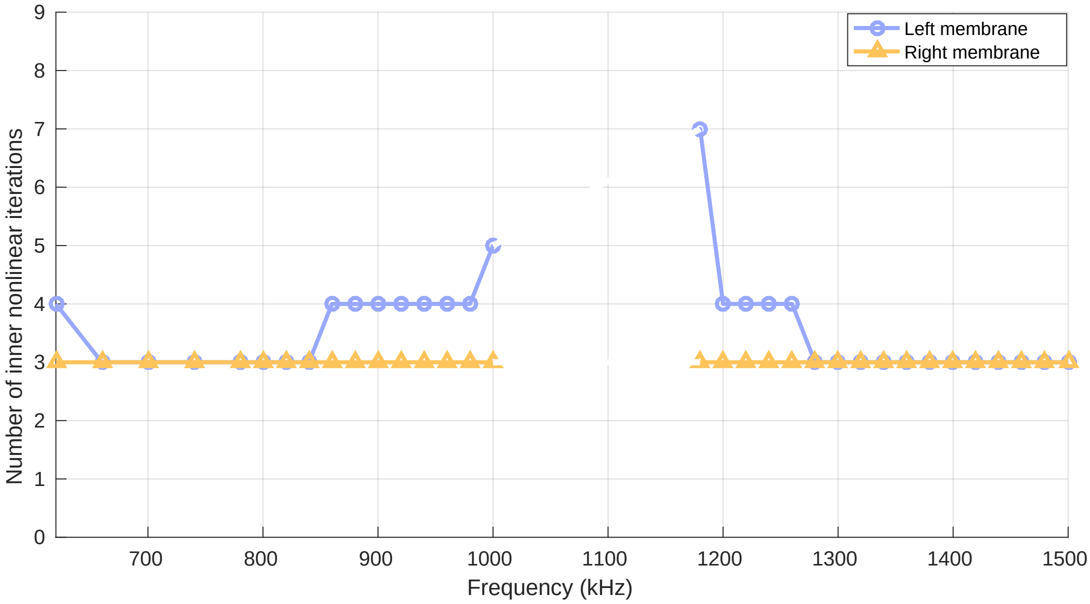


Figure 4.19: Number of inner nonlinear iterations on the left and right membrane to reach a 10^{-4} overall nonlinear relative residual versus frequency for the electroelastoacoutic formulation solved with algorithm NKSRAS.

4.3.5 Nonlinear additive Schwarz preconditioned inexact Newton (*ASPIN*)

Consider a nonlinear problem $F(u) = 0$ to be solved on domain Ω. Split Ω into n overlapping subdomains Ω_i as illustrated in figure 4.5. Instead of solving the original nonlinear problem $F(u) = 0$ the additive schwarz preconditioned inexact Newton algorithm (ASPIN) solves an equivalent nonlinear system

$$\mathcal{F}(u) = 0, \tag{4.24}$$

using an inexact Newton method. $\mathcal{F}$ is called the nonlinearly preconditionned original system F. For the $\mathcal{F}$ chosen below one can show that solving the original and the nonlinearly preconditionned nonlinear systems leads to the same solution [123].

Defining the subdomain projection $T_i(u)$ $\forall i = 1...n$ with support on the ith overlapping subdomain ($T_i = 0$ on the exterior boundaries of the overlapping subdomain) as the solution of the nonlinear system

$$F_{\Omega_i}(u - T_i(u)), \tag{4.25}$$

with F_{Ω_i} being the nonlinear function F restricted to the ith subdomain, one can write the $\mathcal{F}$ operator applied to unknown u as

$$\mathcal{F}(u) = \sum_{i=1}^{N} T_i(u). \tag{4.26}$$

The T_i projection can be interpretated as the nonlinear correction δu on subdomain Ω_i [124], in our electroelastoacoutics case of the displacement $\boldsymbol{u}$, acoustic pressure δp and electric potential v.

The problem $\mathcal{F}(\boldsymbol{u}) = 0$ is solved in ASPIN using an inexact Newton solver which requires the iteration Jacobian matrix. An approximation $\tilde{\boldsymbol{J}}$ to the heavy Jacobian matrix has been deduced in [123]:

$$\tilde{\boldsymbol{J}} = \sum_{i=1}^{N} \boldsymbol{J}_{\Omega_i}^{-1} \boldsymbol{J}, \tag{4.27}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the original system F on the whole domain and $\boldsymbol{J}_{\Omega_i}$ its restriction on subdomain Ω_i.

Rewriting the Newton iteration using (4.27) gives the following system to solve at every outer ASPIN iteration:

$$\sum_{i=1}^{N} \boldsymbol{J}_{\Omega_i}^{-1} \boldsymbol{J} du = \mathcal{F}(u) = \delta u. \tag{4.28}$$

Equation (4.28) resembles the additive Schwarz preconditionned linear Jacobian system of the original nonlinear system F. This can be solved using Krylov-Schwarz with volumic DDM unknowns as in section 4.2.3 in combination with a classical GMRES function but with the preconditionner only applied to the left hand side operator. Algorithm 4.2 describes the steps to perform in ASPIN.

Algorithm 4.2 ASPIN algorithm

```
$u^0 = 0$;
while stopping criterion on $\delta u$ correction not reached do
    for i=1...n do
        Find $\delta u_i$ such that
            $\begin{cases} F_{\Omega_i}(u_i^k - \delta u_i) = & 0 \text{ on } & \Omega_i \\ \delta u_i = & 0 \text{ on } & \Sigma_{ji} \ \forall \text{ neighbour } j \end{cases}$
        using Newton's method
    end for
    $\delta u := \sum_{i=1}^{n} \delta u_i$
    Solve (4.28), compute the Jacobian matrices for $u^k$
    $u^{k+1} := u^k + du$
end while
```

As can be seen in algorithm 4.2 every outer ASPIN iteration requires to solve a nonlinear problem on every subdomain as well as solve a linear problem using e.g. GMRES. Even though the nonlinear resolution need not be solved very accurately an ASPIN iteration might still be quite heavy. The performance of the algorithm will strongly depend on the number of outer iterations.

ASPIN has been shown to have a fast convergence in the case of incompressible Navier-Stokes equations [119] even for high Reynolds numbers where a classical Newton-Krylov-Schwarz did not. In our test case however quadratic convergence could not be obtained. The nonlinear residual would mainly remain at the interface between subdomains. This could be linked to the conditionning issues observed in section 4.3.2.

4.3.6 Choice of the nonlinear domain decomposition method

In this section the major nonlinear DDM algorithms have been detailed and compared. They can be classified into two broad categories. The first category contains the staggered Krylov-Schwarz and Newton-Krylov-Schwarz algorithms. Both exclusively rely on linear DDM algorithm: at every nonlinear iteration a linear problem is solved. For those two algorithms a single LU decomposition is required at every nonlinear iteration and can be reused throughout the linear DDM resolution step. They furthermore incorporate rather progressively the nonlinearity into the solution so that for the CMUT application convergence is obtained even close to resonance and pull-in. A limitation however is that no matter how localised the nonlinearity is the two algorithms will require a same amount of nonlinear iterations on every subdomain: the number of nonlinear iterations does not adapt to the degree of nonlinearity on every subsomain.

In the second category reside the nonlinear alternating Schwarz algorithm as well as the NKS-RAS and ASPIN algorithms. This category differs from the first one by the extra full nonlinear resolution performed at each outer nonlinear iteration. Doing so leads to a number of nonlinear iterations that adapts to the degree of nonlinearity on every subdomain. Taking advantage of it can however be a challenge since it might not be known a priori where nonlinearity will be strong, in particular for a close to resonance vibration.

For the nonlinear alternating Schwarz algorithm convergence was slow, if achievable at all. Additionally no LU decomposition could be reused at any time, leading to a poorly converging, computationally intensive algorithm. In the NKS-RAS algorithm the nonlinear convergence of NKS, which was already good, could not be improved further. NKS-RAS in fact provided a slower nonlinear convergence: most of the nonlinear residual would accumulate at the fluid subdomain interface because of the pressure field restriction step. Furthermore because the nonlinearity is rather brutally incorporated the problem solved at the very first step has a resonance frequency too much shifted compared to the actual problem so that divergence was observed for frequencies for which NKS converged smoothly.

Only the staggered and NKS algorithms will be considered in what follows.

4.4 Coarse grid

All simulations in this chapter have so far been performed considering in the reference 2D CMUT model of figure 1.5 only 2 vibrating membranes. Increasing the number of subdomains unfortunately leads to an increased number of DDM iterations required to reach a given tolerance. This can be understood by looking at the interface-data update equation (4.14) for algorithm Krylov-Schwarz with interface unknowns in section 4.2.2: recalling that the $\mathcal{A}$ operator consists in solving a linear problem on every subdomain with interface data coming from the fields of the neighbour at the previous DDM iteration should make clear that at every DDM iteration information can only flow from a subdomain to its direct neighbour. Thus, if the 2D model has n membranes then at least n iterations are required to exchange information between the two exterior membranes of the array. In practice this tends to increase the number of DDM iterations as illustrated on the top of figures 4.20 and 4.21. The figures show the convergence history for an increasing number of subdomains on the fundamental frequency harmonic subproblem of the elastoacoutic formulation (2.16)-(2.38). Both the Krylov-Schwarz algorithm with interface and with volume unknowns tend to converge slower as the number of subdomains is increased.

To make the number of iterations less dependent on the number of subdomains a *coarse grid correction* or a *coarse grid preconditionner* can be used. It enables direct information exchange between any two subdomains by solving a problem defined on the whole domain but which has much less degrees of freedom than the actual problem and is thus fast to solve.

Let us illustrate the fundamental idea of a coarse grid correction. For that consider an algebraic problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ coming from the finite element discretisation of a linear problem on a fine mesh. Consider also the algebraic problem $\boldsymbol{A}_c\boldsymbol{x}_c = \boldsymbol{b}_c$ coming from the finite element discretisation of the same problem but on the coarsened fine mesh: the coarse mesh. Adding at iteration k of an iterative solver the coarse grid correction to the current approximate solution $\boldsymbol{x}^k$ can be done in four steps.

1. Compute the residual $\boldsymbol{r}^k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^k$

2. Interpolate the residual $\boldsymbol{r}^k$ on the coarse mesh to get $\boldsymbol{r}_c^k$

3. Compute the coarse error $\boldsymbol{e}_c^k = \boldsymbol{A}_c^{-1}\boldsymbol{r}_c^k$

4. Interpolate the coarse error on the fine mesh and add it to $\boldsymbol{x}^k$

Quantity $\boldsymbol{e}_c^k$ is indeed an error since $\boldsymbol{e}_c^k = \boldsymbol{A}_c^{-1}\boldsymbol{r}_c^k = \boldsymbol{A}_c^{-1}(\boldsymbol{b}_C - \boldsymbol{A}_c\boldsymbol{x}_c^k) = \boldsymbol{A}_c^{-1}\boldsymbol{b}_C - \boldsymbol{x}_c^k$. It is worth noting that in case the coarse mesh and fine mesh coincide we have $\boldsymbol{x}^k + \boldsymbol{e}_c^k = \boldsymbol{x}^k + \boldsymbol{A}_c^{-1}\boldsymbol{b}_C - \boldsymbol{x}_c^k = \boldsymbol{x}^k + \boldsymbol{A}^{-1}\boldsymbol{b} - \boldsymbol{x}^k = \boldsymbol{A}^{-1}\boldsymbol{b}$ and $\boldsymbol{x}^k$ immediately becomes the solution of the problem on the fine mesh without further iterations.

In our case the fine mesh and the coarse mesh are identical. The only difference between the fine and the coarse problem is that order 3 finite element interpolations are used on the fine problem while order 1 is used on the coarse problem, leading to a lighter problem with much less degrees of freedom. Doing so with the selected hierarchical shape functions (see section A.2) has the advantage that fine-coarse and coarse-fine interpolations are straightforward. Indeed interpolating the finite element discretised vector $\boldsymbol{x}$ from the fine mesh to $\boldsymbol{x}_c$ on the coarse mesh is as simple as $\boldsymbol{x}_c = \boldsymbol{R}_{\text{COARSE}}\boldsymbol{x}$, where the restriction matrix $\mathbf{R}_{\text{COARSE}}$ selects only the entries of $\boldsymbol{x}$ that correspond to order 1 interpolation degrees of freedom. $\mathbf{R}_{\text{COARSE}}$ is only made of zeros and ones and has a number of rows equal to the number of order 1 degrees of freedom. Interpolating back $\boldsymbol{x}_c$ to $\boldsymbol{x}$ is done with the extension matrix $\boldsymbol{R}_{\text{COARSE}}^T$ as $\boldsymbol{x} = \boldsymbol{R}_{\text{COARSE}}^T\boldsymbol{x}_c$.

For the Krylov-Schwarz algorithm with interface unknowns of section 4.2.2 problem-specific coarse preconditionners have been proposed [125, 126]. Here a coarse grid correction is used. The interface data update equation (4.13)

$$\boldsymbol{g}^{k+1} = \mathcal{A}\boldsymbol{g}^k + \boldsymbol{b} \tag{4.29}$$

is changed as follows:

$$\boldsymbol{g}^{k+1} = \mathcal{C}\mathcal{A}\boldsymbol{g}^k + \boldsymbol{b}, \tag{4.30}$$

where the operator $\mathcal{C}$ adds a coarse correction to $\mathcal{A}\boldsymbol{g}^k$. The application of $\mathcal{C}$ to the interface data vector $\boldsymbol{y} = \mathcal{A}\boldsymbol{g}^k$ is detailed in algorithm 4.3. In the algorithm the problem (4.3) is solved and the notations of section 4.2.2 are used.

For the RAS algorithm of section 4.2.3 a coarse grid preconditionner is used. Designing it can be done in multiple ways [112, 127, 128]. The coarse algebraic preconditionner considered here changes the restricted additive Schwarz system (4.17) to:

$$(\boldsymbol{M}_{\text{RAS}}^{-1} + \boldsymbol{M}_{\text{COARSE}}^{-1})\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{M}_{\text{RAS}}^{-1} + \boldsymbol{M}_{\text{COARSE}}^{-1})\,\boldsymbol{b}, \tag{4.31}$$

where $\boldsymbol{M}_{\text{COARSE}}^{-1} = \mathbf{R}_{\text{COARSE}}^T\,\mathbf{A}_{\text{COARSE}}^{-1}\,\mathbf{R}_{\text{COARSE}}$ and $\boldsymbol{A}_{\text{COARSE}} = \mathbf{R}_{\text{COARSE}}\,\mathbf{A}\,\mathbf{R}_{\text{COARSE}}^T$.

Algorithm 4.3 Application of the coarse operator $\mathcal{C}$ to an interface data vector $\boldsymbol{y}$

```
for i=1...n do
    —— Find the solution to the artificial sources problem:
    Find $v_i$ such that
    $\begin{cases} F(v_i) = 0 & \text{on } \Omega_i \\ v_i = 0 & \text{on } \Gamma_{Di} \\ (\partial_{\boldsymbol{n}} + S)v_i = y_{ji} & \text{on } \Sigma_{ji} \end{cases}$
    —— Add the solution to the physical sources:
    $u_i = v_i + w_i$;
end for
—— Compute the overall residual:
$r = f - F(u)$;
—— Compute the coarse grid correction $e_c$:
Find the best order 1 approximation $e_c$ such that
$\begin{cases} F(e_c) = r_c & \text{on } \Omega \\ e_c = 0 & \text{on } \Gamma_D \end{cases}$
for i=1...n do
    —— Add the artificial sources contribution in $e_{ci}$ to $v_i$:
    $v_i = e_{ci} - w_{ci} + v_i$;
end for
Compute the updated interface data vector $\boldsymbol{y}$ based on subdomain fields $v_i$;
```

Computations have been performed on the reference 2D CMUT of figure 1.5 with a varying number of membranes. The DDM subdomains are the individual membranes. The nonlinear electroelastoacoustic problem is solved with a staggered resolution and the convergence behaviour of the DDM methods is displayed at the second nonlinear iteration for the elastoacoutic formulation (2.16)-(2.38). The membrane at the extreme left and right is fully clamped and has a 0 V electrical excitation. On all other membranes a 40 V constant voltage is set. Only on the second membrane (looking from the left of the array) an alternating voltage of 40 V at 800 kHz is added to the DC bias.

Figure 4.20 and 4.21 show the effect on the Krylov-Schwarz algorithm with interface and with volume unknowns with and without coarse grids of an increasing number of subdomains. Results for the fundamental frequency-harmonic subproblem of the elastoacoustic formulation are displayed. In any case adding a coarse grid speeds up convergence by at least a factor 3. The coarse grid used for Krylov-Schwarz with interface unknowns seems to converge faster than with volume unknowns (i.e. for RAS). Note the difference in the x-axis scale for the top and bottom figure. Unfortunately with interface unknowns it cannot achieve a really accurate solution and using it e.g. in NS (section 4.3.2) is problematic.

Figure 4.22 and 4.23 show results for the vibrations at the electrical excitation frequency f_0 as well as $2f_0$ and $3f_0$. Figure 4.22 is there to illustrate the in-phase and quadrature membrane vibration associated to the results of figure 4.23. Unsurprisingly the top harmonics, vibrating at f_0, require the largest number of DDM iterations to reach a 10^{-3} overall relative residual. This can be related to the large crosstalk they exhibit.

It is finally worth comparing the sizes of the algebraic matrix associated to the fine problem to solve on every subdomain with the coarse problem solved on the whole domain. The fine

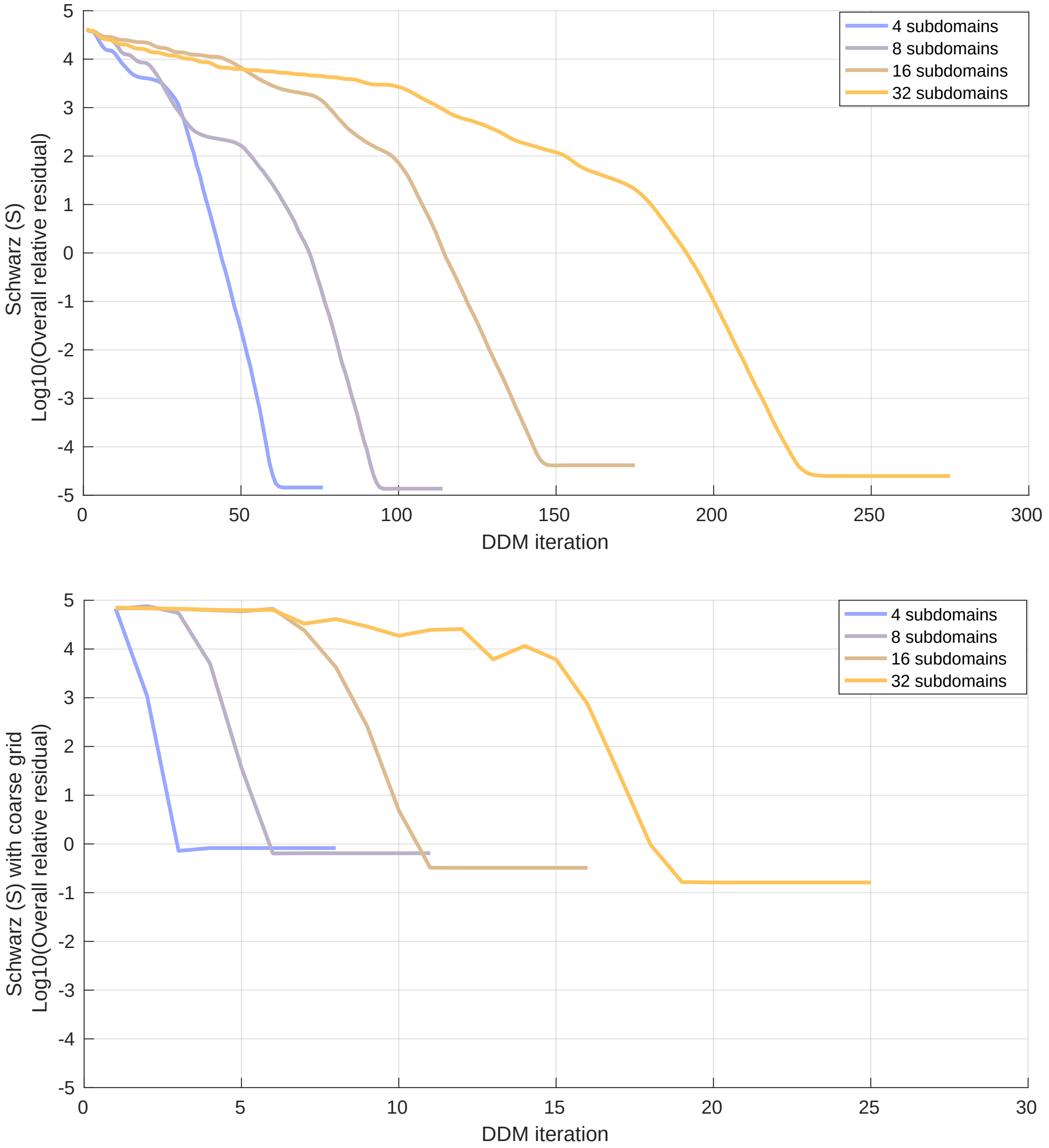


Figure 4.20: Overall relative residual versus domain decomposition iteration for the Krylov-Schwarz algorithm with interface unknowns without (top) and with (bottom) coarse grid as the number of subdomains is increased. The elastoacoutic problem is solved for the fundamental frequency harmonics.

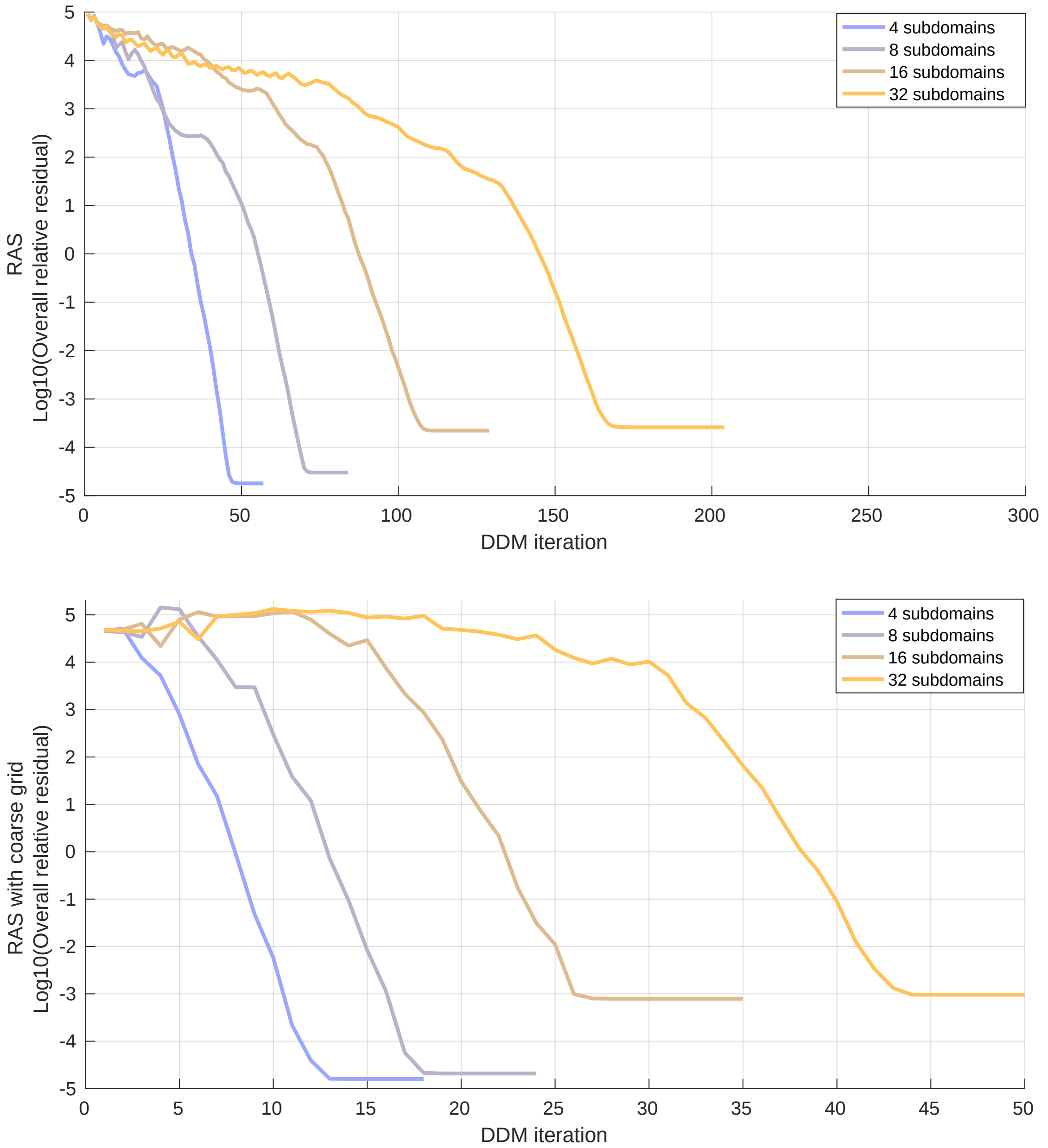


Figure 4.21: Overall relative residual versus domain decomposition iteration for the RAS algorithm without (top) and with (bottom) coarse grid as the number of subdomains is increased. The elastoacoutic problem is solved for the fundamental frequency harmonics.

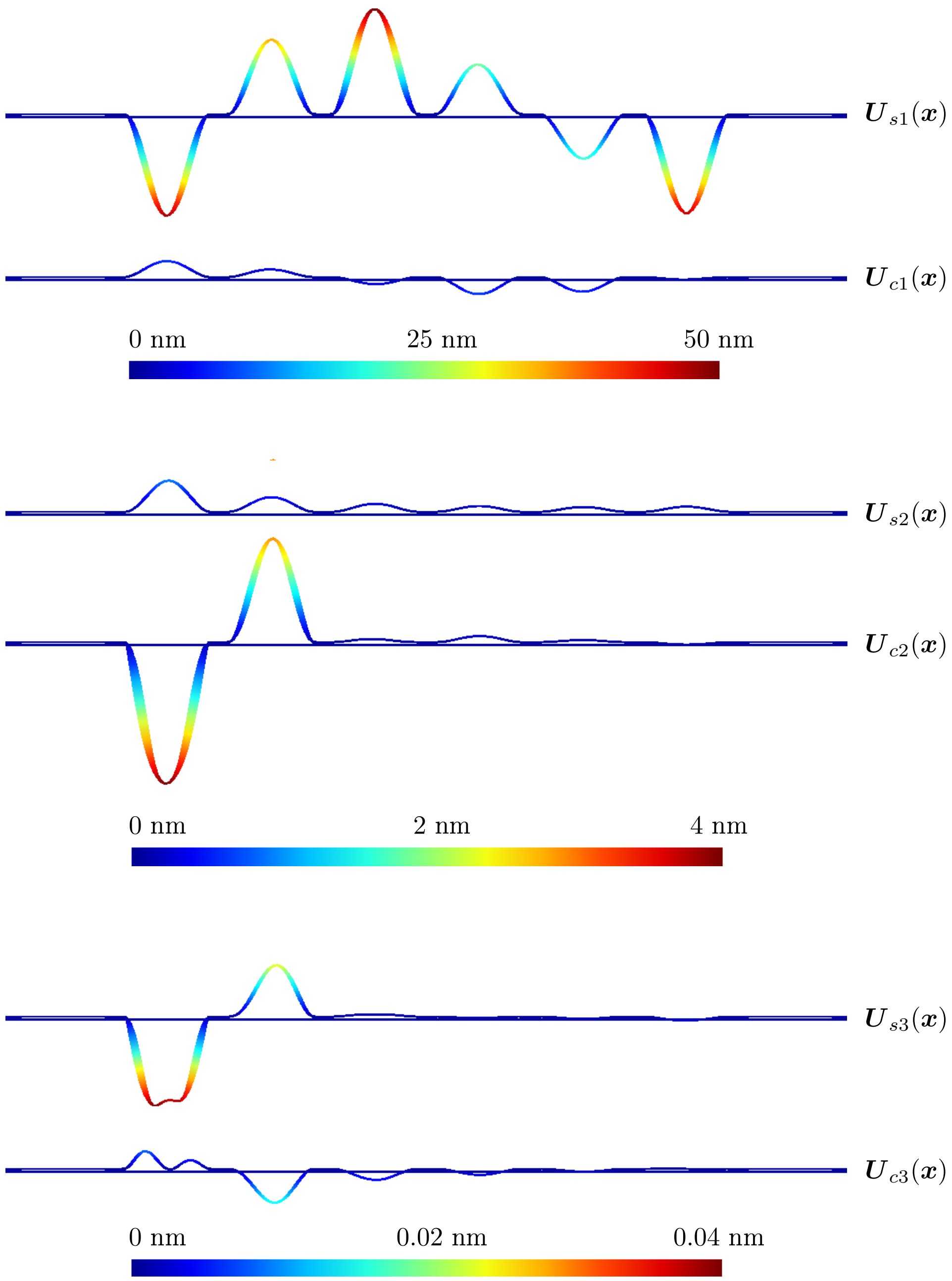


Figure 4.22: In-phase (subtop) and quadrature (subbottom) membrane vibration harmonics on the reference CMUT with 6 membranes for a vibration frequency equal to the electrical excitation frequency (top), twice as large (middle) and 3 times as large (bottom).

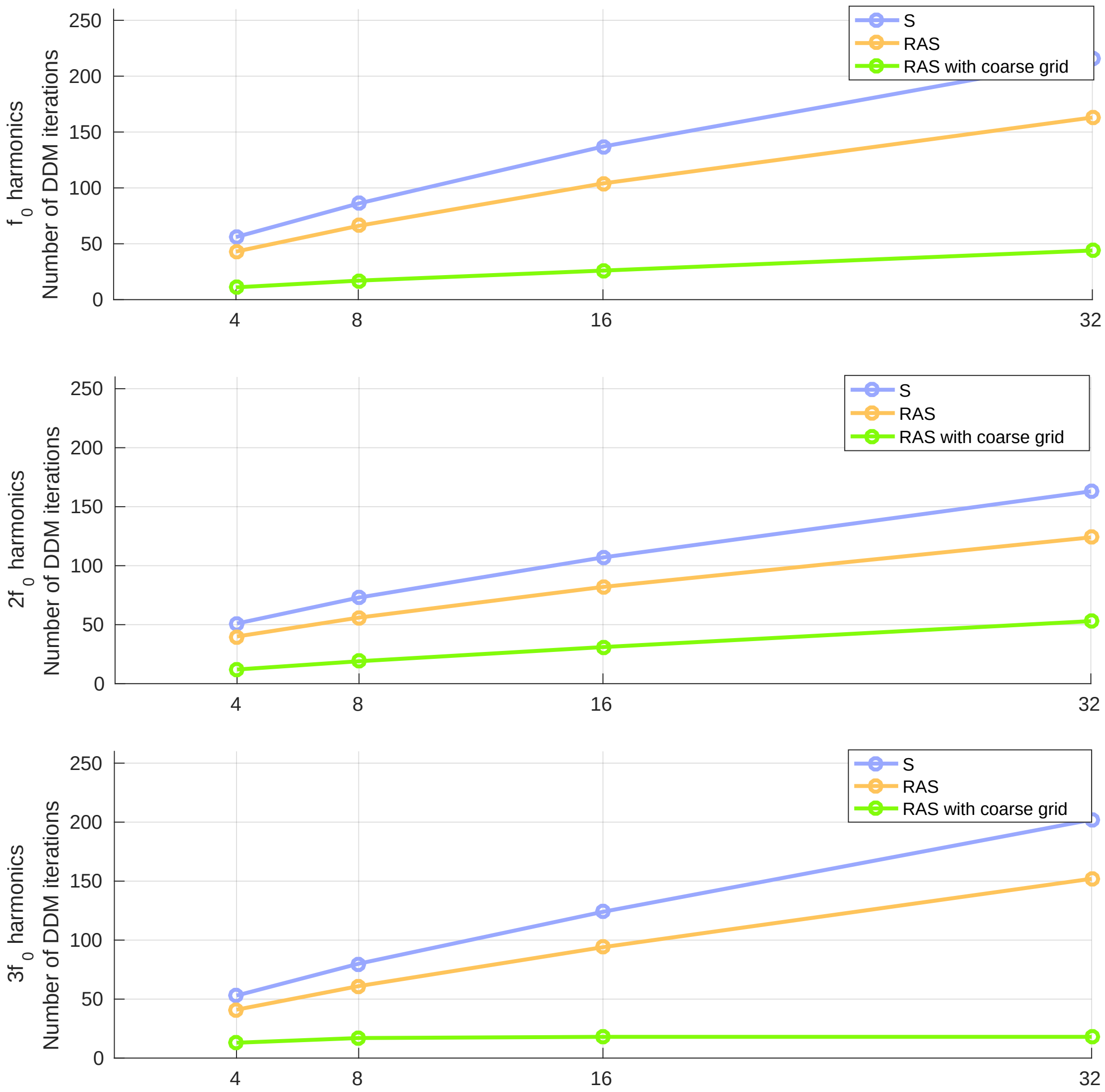


Figure 4.23: Number of domain decomposition iterations to reach a 10^{-3} overall relative residual as the number of subdomains is increased for the Krylov-Schwarz algorithms with interface and volume unknowns with coarse grid. The elastoacoutic problem is solved for the fundamental frequency f_0, $2f_0$ as well as $3f_0$ harmonics.

subdomain problems are computed in parallel while the coarse problem is computed on a single processor and should thus be small enough compared to the fine subdomain problem. Since for the fine problem an order 3 interpolation is used on a structured quadrangular mesh there are exactly 16 degrees of freedom per quadrangle. For the coarse problem order 1 is used and there are thus 4 times less degrees of freedom per quadrangle. At the same time the coarse problem only has a single degree of freedom on each side of the overlap, further reducing the size. One thus obtains a coarse problem for N subdomains that has an overall size of less than $\frac{N}{4} \cdot n$, where n is the size of the fine problem on a single subdomain. While this can seem to be a rather fine coarse mesh a higher fine interpolation order can be used with less elements in the mesh to further coarsen the coarse grid. A typical way to obtain a much coarser mesh however is to really consider different coarse and fine meshes but this requires the ability to interpolate between two different meshes, which has not been implemented.

Chapter 5

Application to the simulation of crosstalk in MEMS

This chapter begins with a 2D and 3D verification of the multiharmonic solver. Simulations are then performed on large 3D CMUT arrays with up to 20 million degrees of freedom to simulate the crosstalk appearing through acoustic waves in the fluid. Finally the fluid is removed and the crosstalk via elastic waves in the bulk is simulated. The nonlinearity introduced by the electromechanical coupling is taken into account in all cases.

5.1 Software verification

5.1.1 One-dimensional CMUT

In this section the multiharmonic solver will be verified for a nonlinear multiharmonic resolution on a 2D mesh against a 1D model of a CMUT vibrating in water. For that purpose, the pressure forces applied by the fluid on the membrane are added to the model of figure 1.3: the resulting model is displayed in figure 5.1.

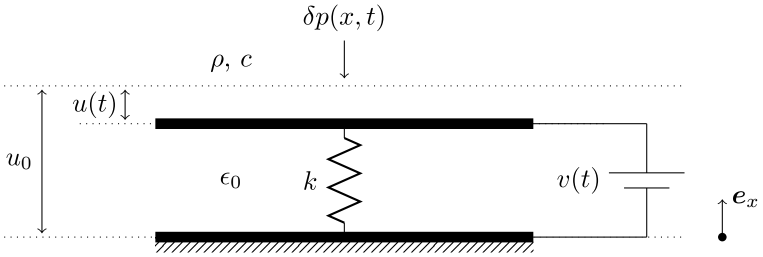


Figure 5.1: Lumped model of a membrane electrically actuated by a time-dependent voltage $v(t)$ and vibrating in a fluid.

With the pressure force applied by the fluid on the membrane the equilibrium equation (1.21) becomes

$$-\,k\,u(t) - \frac{1}{2}\epsilon_0 \frac{v(t)^2}{(u_0 + u(t))^2} A - \delta p(u_0 + u(t), t) A = 0. \tag{5.1}$$

Newton's law gives the relation at the electrode between the pressure gradient and the mechanical acceleration:

$$-\frac{d\delta p(u_0 + u(t), t)}{dx} = \rho \frac{d^2 u(t)}{dt^2}. \tag{5.2}$$

Finally the wave equation defines the acoustic behaviour in the fluid:

$$\frac{d^2\delta p(x,t)}{dx^2} - \frac{1}{c^2}\frac{d^2\delta p(x,t)}{dt^2} = 0, \tag{5.3}$$

and the Sommerfeld radiation condition at infinity

$$\frac{d\delta p}{dx} + \frac{1}{c}\frac{d\delta p}{dt} = 0 \tag{5.4}$$

leads to outgoing pressure waves.

Equation (5.1) can be rewritten as:

$$-\,k\,u(t)(u(t) + u_o)^2 - \frac{1}{2}\,\epsilon_0\, v(t)^2 A - \delta p(u_0 + u(t), t)(u(t) + u_0)^2 A = 0 \tag{5.5}$$

which is a cubic equation in $u(t)$ possibly leading to multiple real solutions. Only a single solution however is not beyond pull-in.

Considering an electrical excitation $v(t) = 300\sin(2\pi f_0 t)$ V (f_0 is set to 1 MHz) leads to a nonlinear periodic vibration $u(t)$ that can be approximated by its Fourier truncation similarly to what has been done in chapter 3:

$$u(t) = U_{c0} + U_{s1} sin(2\pi f_0 t) + U_{c1} cos(2\pi f_0 t) + U_{s2} sin(2 \cdot 2\pi f_0 t) + U_{c2} cos(2 \cdot 2\pi f_0 t) + ... \tag{5.6}$$

Because the pressure waves emitted by the vibrating electrode are outgoing one gets

$$\begin{aligned}\delta p(x,t) = \delta P_{s1}\sin(2\pi f_0 t - k_1 x) + \delta P_{c1}\cos(2\pi f_0 - k_1 x)\\ +\delta P_{s2}\sin(2 \cdot 2\pi f_0 t - k_2 x) + \delta P_{c2}\cos(2 \cdot 2\pi f_0 - k_2 x) + ...\end{aligned} \tag{5.7}$$

where k_i is the wavenumber for frequency if_0, that is $k_i = \frac{2\pi}{\lambda_i} = \frac{2\pi i f_0}{c}$. In (5.7) the time-independent term is dropped since a constant displacement creates no pressure variation. The pressure form (5.7) automatically satisfies Sommerfeld's condition (5.4). In order to obtain all unknown Fourier coefficients $U_{c0},\ U_{s1},\ U_{c1},\ U_{s2},\ U_{c2}, ...$ and $\delta P_{s1},\ \delta P_{c1},\ \delta P_{s2},\ \delta P_{c2}, ...$ equations (5.1), (5.2) and (5.3) can be used. A symbolic resolution has been implemented in Matlab to get the coefficients corresponding to the first 8 harmonic frequencies, i.e. the first 17 Fourier coefficients in (5.6) and the first 16 terms in (5.7). Table 5.1 shows their value while figure 5.2 shows the electrode acceleration versus time for a half excitation period (bottom) and the pressure versus distance to electrode for several time instants (top).

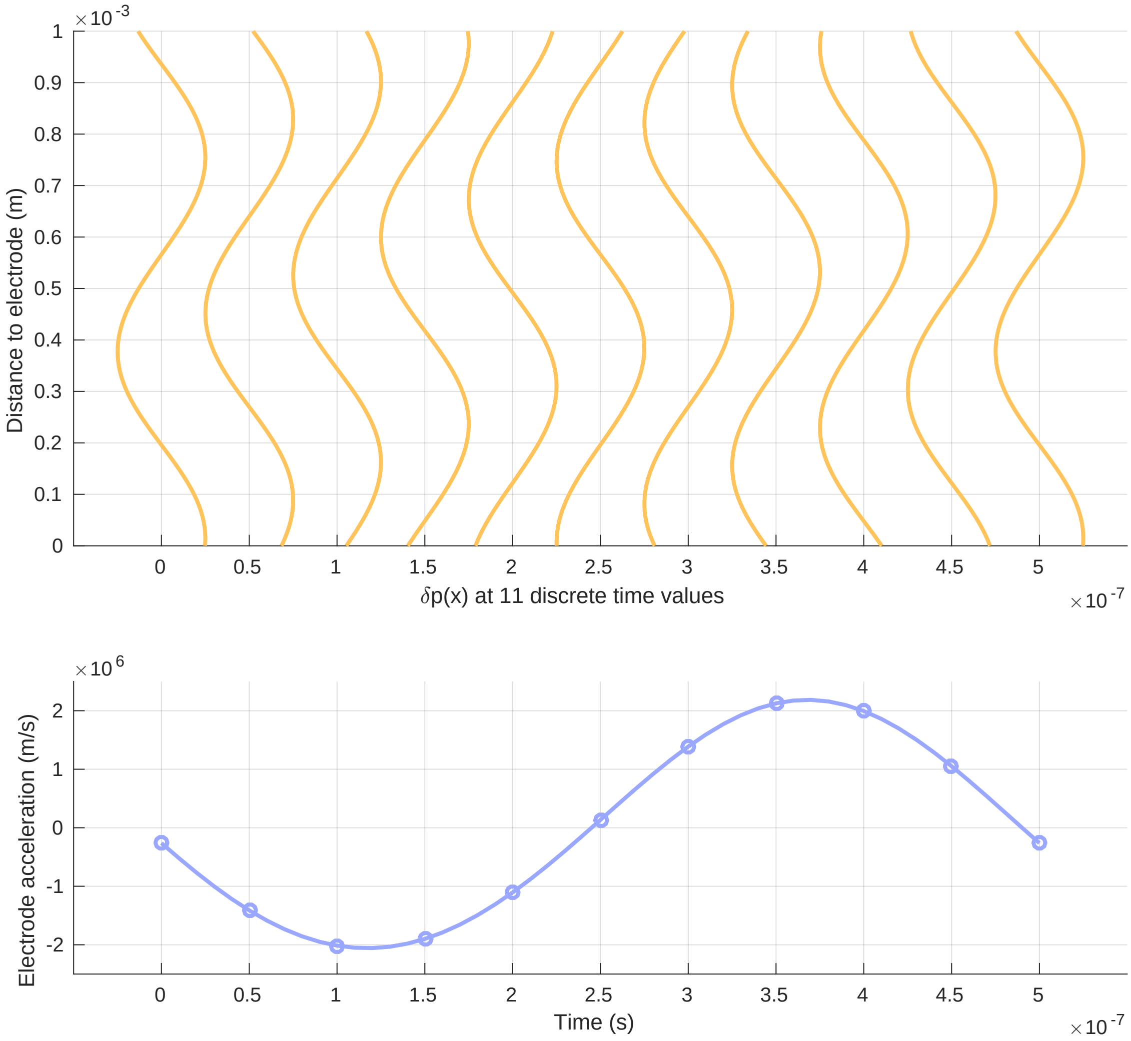


Figure 5.2: Electrode acceleration versus time (bottom) and pressure versus distance x to electrode at 11 different time steps.

A multiharmonic finite element simulation is performed for verification on a 2D mesh with a fluid truncated 400 μm above the membrane top. The mesh is structured. It is made up of 20000 rectangular elements with a first order finite element interpolation. There is a single element in the x direction. Because the finite element simulation of this 1D problem is performed on a 2D mesh one has to make sure to remove the dependency on the y coordinate. This is automatically achieved for the electrostatic and acoustic problems by imposing natural boundary conditions on the left and right boundaries of the vacuum and fluid regions. Imposing natural boundary conditions on the elasticity problem however does not exclude 2D effects as such conditions correspond to an interface that is free to move. The y-component of the displacement is thus forced to zero on the boundaries.

As can be seen on table 5.1 the Fourier coefficients of the membrane displacement and emitted pressure are very close for the analytic resolution and the finite element resolution and the multiharmonic solver can thus be considered verified in 2D. The fact that the results do not exactly match is linked to the finite number of elements in the mesh.

Table 5.1: Value of the Fourier coefficients of the displacement $u(t)$ and pressure $\delta p(t)$ on the 1D CMUT model of figure 5.1.

Harmonic	$u(t)$ analytic	$u(t)$ FEM	$\delta p(t)$ analytic	$\delta p(t)$ FEM
$c0$	$-1.08694 \cdot 10^{-7}$	$-1.08694 \cdot 10^{-7}$		
$s1$	0	0	0	0
$c1$	0	0	0	0
$s2$	$1.33661 \cdot 10^{-8}$	$1.33661 \cdot 10^{-8}$	$-2.33543 \cdot 10^{4}$	$-2.33543 \cdot 10^{4}$
$c2$	$1.25572 \cdot 10^{-9}$	$1.25571 \cdot 10^{-9}$	$2.48585 \cdot 10^{5}$	$2.48585 \cdot 10^{5}$
$s3$	0	0	0	0
$c3$	0	0	0	0
$s4$	$-1.19833 \cdot 10^{-11}$	$-1.19830 \cdot 10^{-11}$	$-3.75599 \cdot 10^{3}$	$-3.75599 \cdot 10^{3}$
$c4$	$1.00977 \cdot 10^{-10}$	$1.00984 \cdot 10^{-10}$	$-4.45741 \cdot 10^{2}$	$-4.45739 \cdot 10^{2}$
$s5$	0	0	0	0
$c5$	0	0	0	0
$s6$	$-1.26351 \cdot 10^{-12}$	$-1.26378 \cdot 10^{-12}$	$1.14365 \cdot 10$	$1.14363 \cdot 10$
$c6$	$-2.04971 \cdot 10^{-13}$	$-2.08111 \cdot 10^{-13}$	$-7.04958 \cdot 10$	$-7.04971 \cdot 10$
$s7$	0	0	0	0
$c7$	0	0	0	0
$s8$	$4.01561 \cdot 10^{-15}$	$4.37047 \cdot 10^{-15}$	1.40129	1.40611
$c8$	$-1.89011 \cdot 10^{-14}$	$-1.81790 \cdot 10^{-14}$	$3.16625 \cdot 10^{-1}$	$2.98733 \cdot 10^{-1}$

5.1.2 Three-dimensional CMUT

In this section the Matlab code is verified against a 3D finite element simulation performed in [1] on a circular CMUT cell surrounded by six neighbouring cells. Water couples all the cells together so that when the central cell is electrically excited the six surrounding ones vibrate due to crosstalk: the problem involves a coupling between the electric potential field, the mechanical displacement and the acoustic pressure. The electrostatic (2.6), elasticity (2.16) and acoustic (2.38) formulations are used for the simulations. The dimensions of a single cell are identical to those used in [1] and are provided in figure 5.3.

A linear vibration mode is considered in the same way as in [1]: a large bias voltage of 95 V (about 90% of the pull-in voltage) is applied between the electrode and the ground on (only) the central membrane. To the bias voltage a small 1 V alternating voltage is added, with a frequency ranging from close to zero to 40 MHz. This small alternating voltage leads to a linear vibration

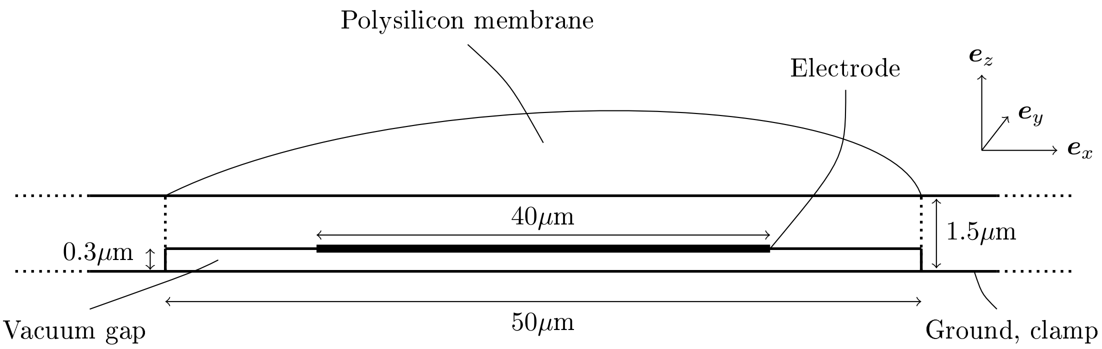


Figure 5.3: Model of a circular CMUT.

around the static deflection and thus allows to compute the constant deflection independently from the harmonic vibration. Moreover because of linearity the vibration only involves the in-phase and the quadrature vibration at the electrical excitation frequency. Close to membrane resonance however this might not hold anymore and extra harmonics can appear due to nonlinearity, but as done in [1] the vibration will still be supposed linear.

A staggered resolution scheme is used to solve the nonlinear electroelastoacoutic problem. For convenience the electrostatic as well as the elastoacoustic problem are solved with the automatic multiharmonic resolution detailed in section 3, even though the vibration is linear and a classical harmonic resolution method could be used once the static deflection is known. The multiharmonic resolution has indeed the advantage to automatically compute the correct electrostatic forces acting on the membrane. Indeed, as was the case in (3.7), the constant and alternating components of the electrostatic force cannot be computed by considering independently the constant electric field and the alternating electric field since the force depends nonlinearly on the electric field.

Unlike in [1] no extra mechanical damping will be added to the model since it is not straightforward to find the equivalent damping term to add to the elasticity formulation. The mesh used for the finite element computation is shown in figure 5.4: the mesh is extruded with a single layer in the vacuum gap and on the membrane while 5 layers are used in the water on top of the membranes. A second order finite element interpolation is used for the mechanical displacement, for the acoustic pressure and for the electric potential field. The truncated fluid region has a height and a radius of 200 μm.

In a first verification the six outer membranes are clamped. In any case water is considered for the fluid and only the inner membrane is electrically excited. Figure 5.5 is obtained when the maximum vibration magnitude is computed for a frequency sweep. The computed curve matches closely the curve obtained in [1]. Apart from the different mesh, interpolation order and fluid truncation considered, the main visible difference is the larger vibration close to resonance. This should come as no surprise as no mechanical damping has been considered. The only damping in the model comes from the radiated acoustic power.

In a second verification the outer membranes are free to move and figure 5.6 is obtained. Compared to figure 5.5 where the outer membranes are clamped new peaks appear. The curve obtained clearly shows the impact on the inner membrane of the crosstalk between the inner and the outer membranes. The results here again closely match those of [1].

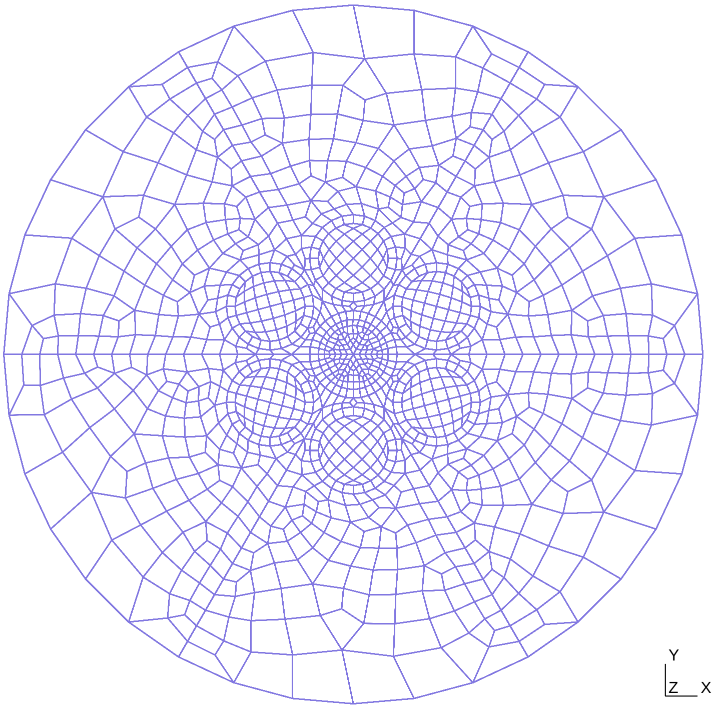


Figure 5.4: Top view of the meshed circular membrane geometry.

Table 5.2: First five resonance frequencies computed in [1].

f_{R1}	f_{R2}	f_{R3}	f_{R4}	f_{R5}
3.86 MHz	11 MHz	19.9 MHz	23.9 MHz	32.8 MHz

To quantify how close both simulations are one can compare the first five resonance frequencies (f_{R1}, f_{R2}, f_{R3}, f_{R4}, f_{R5}). The frequencies obtained in [1] are listed in table 5.2 while the resonance frequencies obtained with the multiharmonic solver are such that

1. 3.5 MHz $< f_{\mathrm{R1}} <$ 4 MHz

2. 11.5 MHz $< f_{\mathrm{R2}} <$ 12 MHz

3. 20.5 MHz $< f_{\mathrm{R3}} <$ 21 MHz

4. 23.5 MHz $< f_{\mathrm{R4}} <$ 24 MHz

5. 32.5 MHz $< f_{\mathrm{R5}} <$ 33 MHz

i.e. very close to the ones obtained in [1], especially at higher frequency. The associated resonance modes are displayed in figure 5.7 and correspond to the modes in the reference. The software can thus be considered validated in this 3D electroelastoacoustic test case as well.

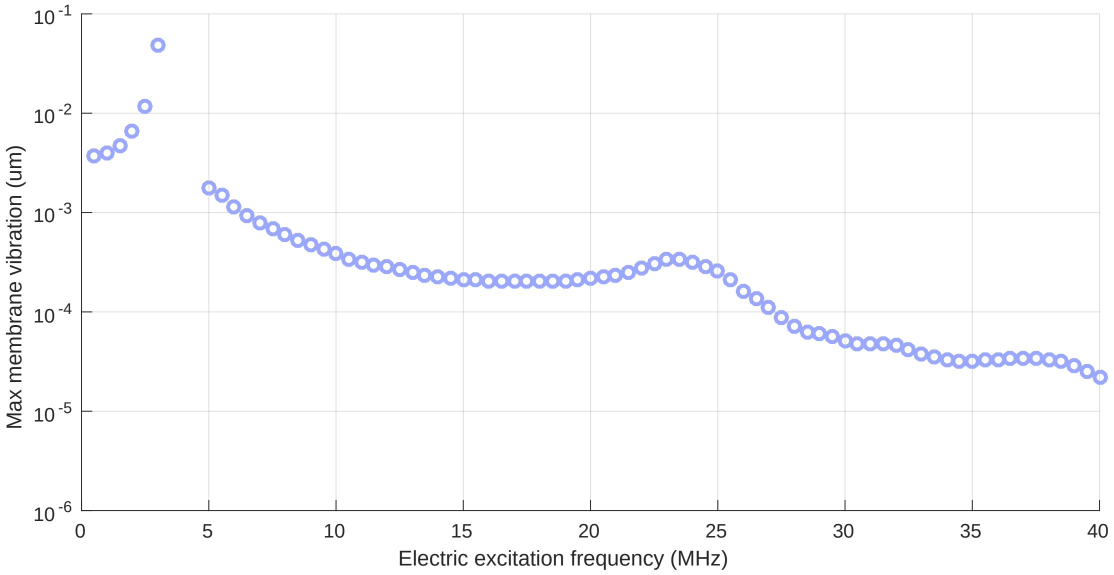


Figure 5.5: Maximum membrane deflection versus frequency when the central membrane is electrically excited and the outer membranes are clamped. The constant deflection is removed.

5.2 Crosstalk through acoustic waves

This section demonstrates that crosstalk through acoustic waves propagating in water can be simulated in steady state for large CMUT arrays with the techniques presented. The nonlinearity arising from the electromechanical coupling is in any case fully taken into account. For that, arrays of CMUTs are considered. As an illustration figure 5.8 shows a 10 by 10 array. The elementary cells are the reference 3D CMUT of figure 1.5, with water on top of the membranes and a clamped bulk region in order to focus on the fluid crosstalk.

The nonlinearly coupled electric-mechanic-acoustic problem is simulated with the staggered resolution method described in section 2.4.1. The electroelastoacoustic formulation (2.6)-(2.16)-(2.38) is solved with the multiharmonic framework.

Because obtaining the finite element discretised algebraic system with high-order finite element interpolations along with the multiharmonic framework can take up a large part of the total simulation time it is worth analysing the fraction of the time that is spent on the symbolic processing, the matrix assembly and the actual resolution of the algebraic system. The times are displayed in figure 5.9 for 1024 hexahedra in the mesh of every single CMUT cell and a second order finite element interpolation. They extend in 3D the analysis performed for figure 3.7.

For the staggered resolution the only problem that becomes larger with an increasing number of harmonics is the electrostatic problem. It has indeed been shown that the elastoacoustic problem can be solved for each harmonic frequency independently. For a Newton iteration however the sensitivity matrix (derived in section 2.3.5) must be considered for all electric field, mechanical displacement and acoustic pressure harmonics together, leading to a very large algebraic system to solve. A Newton iteration could thus not be performed for more than two harmonics with reasonable memory resources.

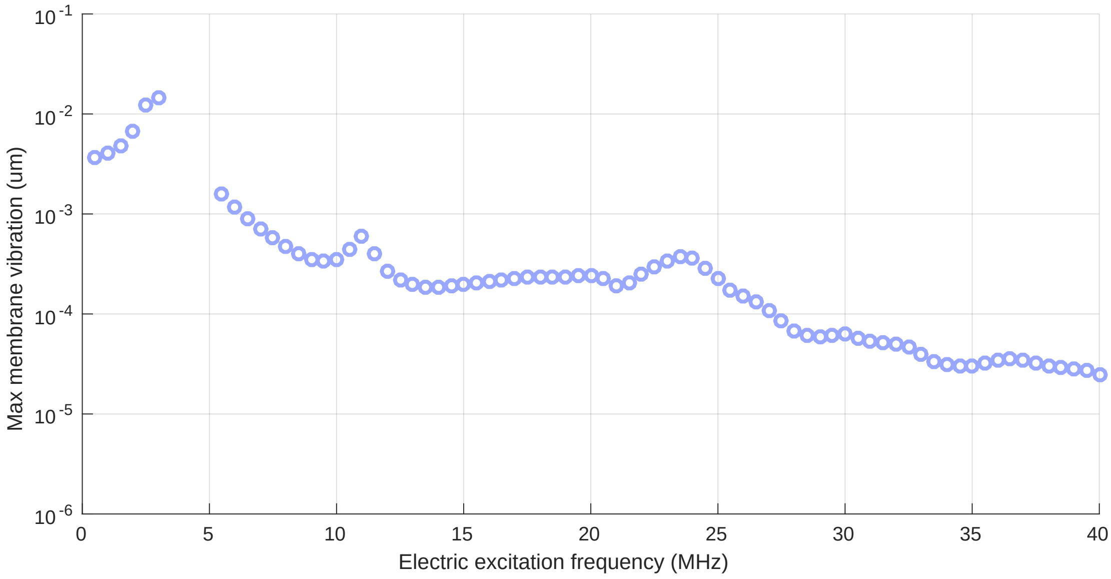


Figure 5.6: Maximum membrane deflection versus frequency when the central membrane is electrically excited and the outer membranes are free to vibrate. The constant deflection is removed.

Observations made in 2D for figure 3.7 are still valid here: the symbolic computations take a relatively large time but can be computed once and for all and are thus not problematic while the matrix generation time is dominant in the total simulation time. In any case however figure 5.9 shows that a large number of harmonics (more than 10) could be treated in a very reasonable amount of time in a 3D setting. The vast majority of excitation settings are thus within reach, considering that in chapter 3 only 5 harmonics were required for a very accurate simulation of a close-to-resonant vibrating CMUT with a large $v(t) = 40 + 40\sin(2\pi f_0 t)$ V excitation voltage (pull-in voltage was 110 V).

Simulating large CMUT arrays is also within reach with the domain decomposition methods presented in chapter 4, as long as the number of iterations required remains reasonable. Array sizes ranging from 2×2 to 7×7 have been simulated using the staggered RAS algorithm (detailed in section 4.3.1) and the number of domain decomposition iterations recorded. The seven first harmonics have been considered for every field. Figure 5.10 shows the number of iterations required to reach a 10^{-9} overall relative residual versus number of cells in the array. A plot is shown for every harmonic frequency. As can be seen for the constant frequency the number of iterations remains stable. This is because there is no fluid crosstalk for the 0 Hz frequency. For all other frequencies the number of iterations required increases with the number of subdomains but still remains very reasonable even for a 7×7 array. Furthermore by adding the coarse grid detailed in section 4.4 the number of iterations can be reduced by up to a factor 3. The detailed convergence history can be found in figure 5.11 (no coarse grid considered) and in figure 5.12 (coarse grid considered).

The influence of the electrical excitation frequency is also investigated: in figures 5.13 and 5.14 the excitation frequency is increased from 1 MHz to 8 MHz by factors of 2. The first figure shows the convergence history for the RAS algorithm without coarse grid correction while in the second

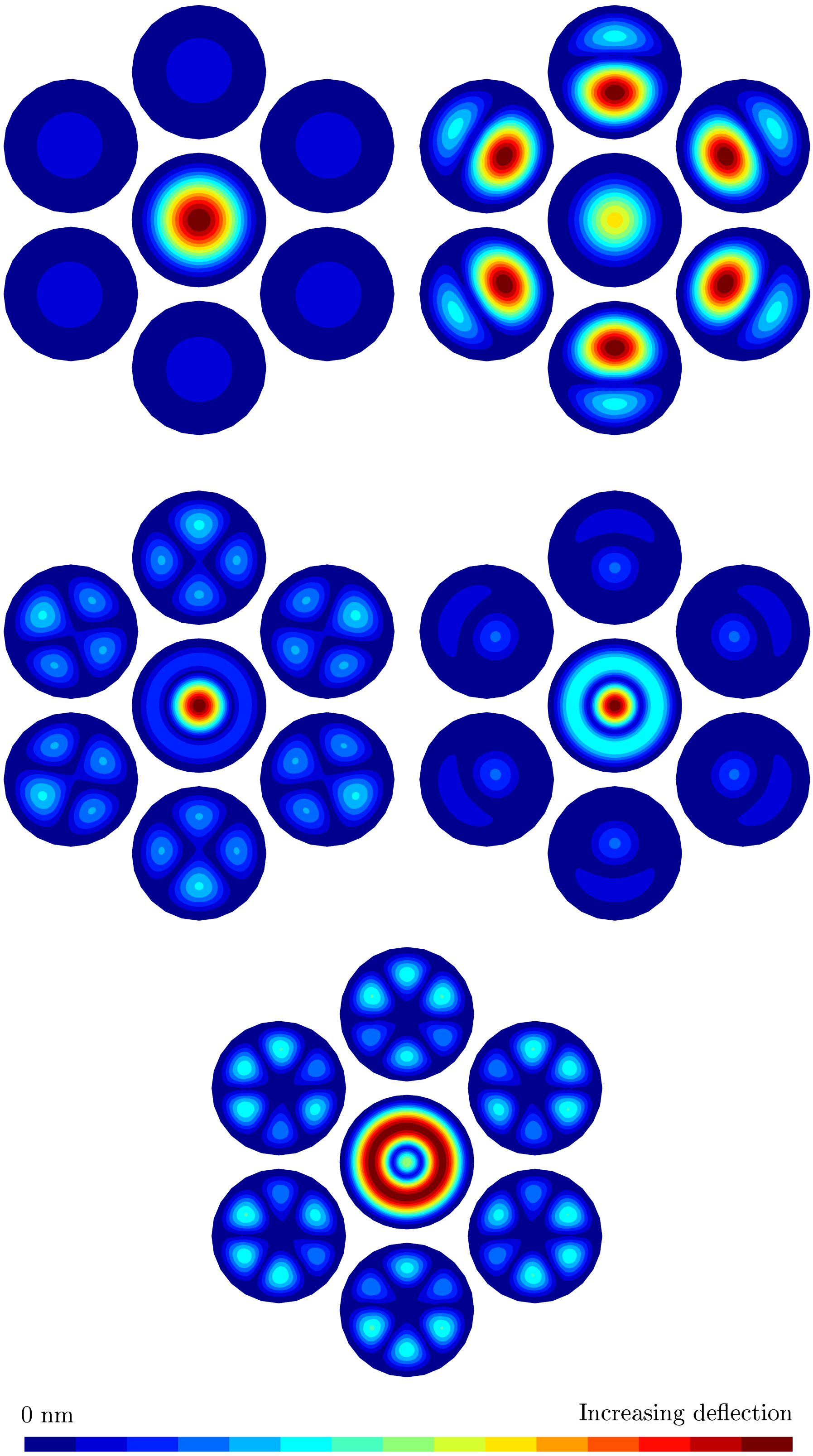


Figure 5.7: Shape of the first five vibration modes. The vibration component in quadrature is displayed.

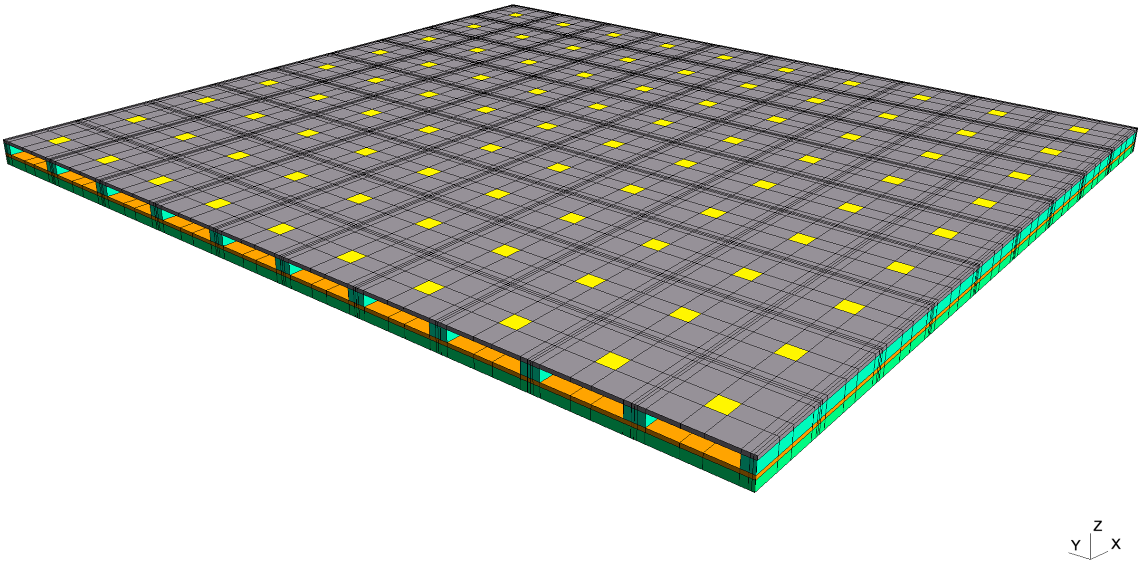


Figure 5.8: Illustration of a 10x10 CMUT array (not to scale). Water on top not displayed.

figure a coarse grid has been used. It is visible with the 4 and 8 MHz excitation frequencies that an increased frequency leads to a rather large convergence slowdown (the other 2 harmonics are heavily impacted by the resonance peak between the 1 and 2 MHz excitation frequency). This slowdown with an increasing frequency is expected and was already observed on the 2D reference geometry before, where it was improved with Sommerfeld interface conditions. The same trend is observed when a coarse grid is used, though the coarse grid still provides a noticeable factor 2 convergence speedup at 8 MHz.

In order to demonstrate the parallel computation capabilities of the algorithm a simulation of a 12×12 CMUT array surrounded by a fluid layer has been performed in parallel on 196 processing units. For high accuracy seven harmonics have been used in the Fourier truncation of the displacement, pressure and electric potential fields: out of the 7 the last 2 are negligeable while the first and the fifth are the largest. After discretisation there are $2,206,526$ degrees of freedom in the domain decomposition unknown vector - $\boldsymbol{x}$ in (4.17) - of the electrostatic problem and $18,305,463$ in total for all frequencies of the elastoacoustic problem. The solution time on 196 processing units (2 GHz Intel cores with 4 GB RAM) was about 8 hours per staggered nonlinear iteration with the multiharmonic solver (3 iterations accurately solve the nonlinearity) - in our quite basic Matlab implementation most of the time was spent in communication between processing units. In the membrane plane, structured mesh layers made up of 16×16 order 2 hexahedra were used to accurately capture the first mode membrane vibration (as shown in section 2.5). The fluid was 2 membrane-length (100 μm) high and meshed with 2 layers. Allowing more than 4 GB memory per processing unit would provide a more accurate simulation of the fluid. Fig. 5.16 shows a 3D view of the $\mathbf{U}_{c2}$ and δP_{c2} displacement and pressure harmonics in case of a 70 volts electrode to ground DC bias applied on all cells with an additional 70 $\sin(2\pi \cdot 10^6\, t)$ volts applied only on two cells. Figures 5.15, 5.17 and 5.18 show the top view of the membrane displacement and pressure at the membrane top for all harmonics considered. Because the pull-in voltage is around

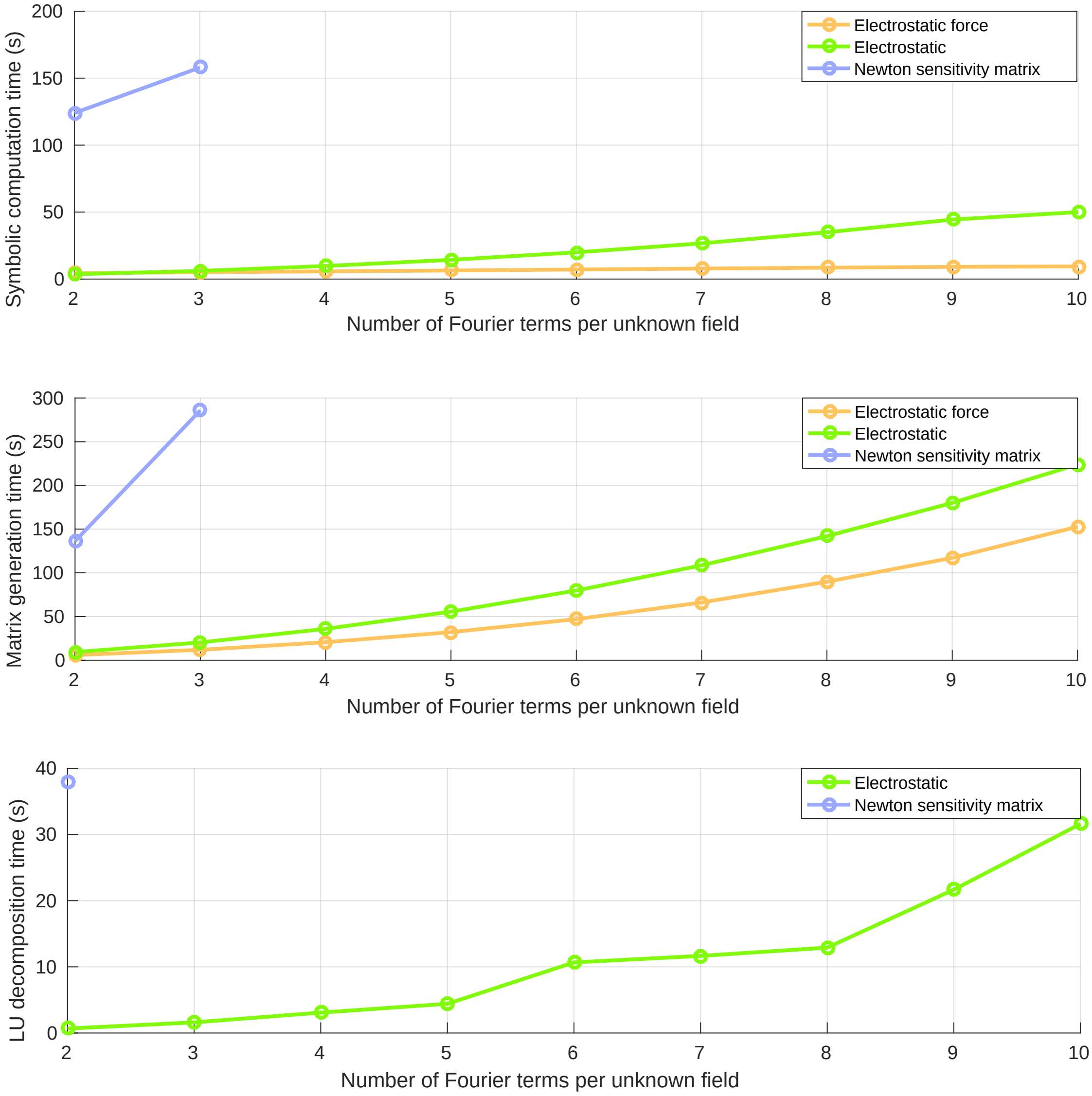


Figure 5.9: Time (s) for the symbolic multiharmonic computation (top), the matrix generation (center) and its LU decomposition (bottom) for the electrostatic force (2.87), the electrostatic problem (2.65) and a the sensitivity matrix (section 2.3.5) in the Newton iteration versus number of terms in the Fourier truncation of every field.

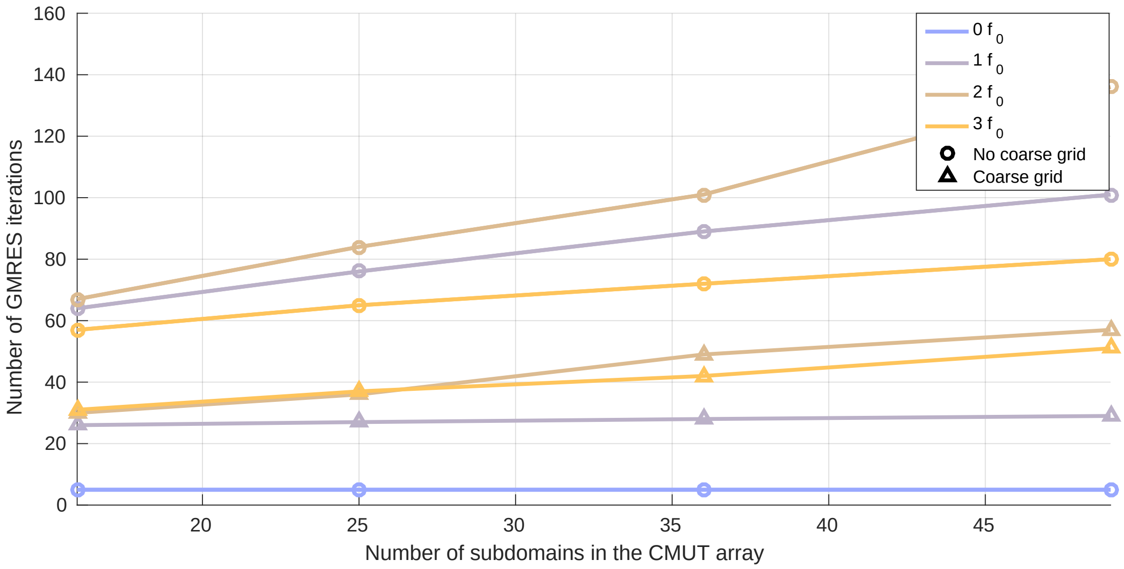


Figure 5.10: Number of DDM iterations to reach a 10^{-9} overall relative residual versus number of cells in the CMUT array.

200 volts and the membrane vibrates close to resonance the displacements changed by 30% from the first to the second staggered nonlinear iterations and nonlinearity had thus clearly to be taken into account. An example of nonlinear effect is visible on the top view of the constant deflection in figure 5.15: even though the constant bias voltage is the same for all cells the two cells excited with an additional alternating voltage have a larger constant deflection than the others.

5.3 Crosstalk through elastic waves

In this section the reference CMUT geometry of figure 1.5 is considered without the fluid layer on top, so that a MEMS device with an electromechanical coupling is obtained. With these settings one can simulate the crosstalk between CMUT cells that originates from bulk elastic waves. The electrostatic formulation used in the staggered nonlinear resolution is described in 2.3.1 while the elastodynamic formulation is described in 2.3.2.

For the finite element simulation the bulk region must be truncated and interface conditions must be set at the truncation boundaries. In the hypothesis of perfect reflection at the exterior bulk boundaries Dirichlet conditions can be used to clamp the bulk. In the hypothesis of radiated elastic waves appropriate conditions must be applied. For acoustic pressure waves this was achieved by using Sommerfeld interface conditions (2.40). In an elastic medium however all waves do not propagate at the same speed: compression waves typically have a higher velocity than shear waves. The Sommerfeld condition can thus not be used as such, it must be replaced by Kupradze interface conditions [129]. For a bulk truncation boundary parallel to the bottom of the membrane (i.e. in the $x - y$ plane perpendicular to the z axis) and located far enough from the membranes one can assume that the elastic waves are traveling downwards (in the $-z$ direction) and hit the truncation boundary perpendicularly. In this case the Kupradze conditions to impose on the truncation

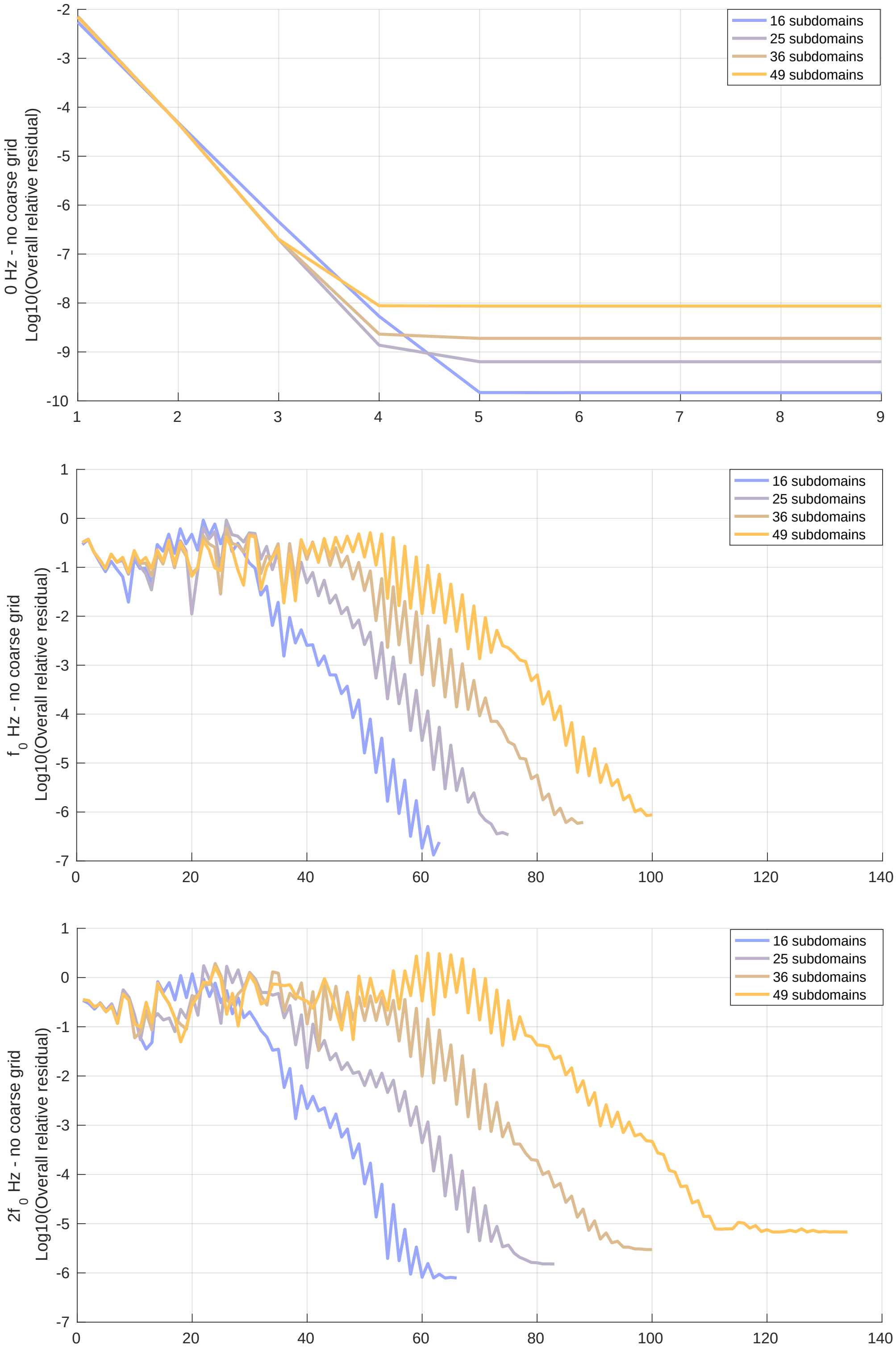


Figure 5.11: Influence of the number of subdomains on the convergence history for the 3D CMUT array with water. The top figure is for the constant harmonic, the middle one for the fundamental frequency harmonics and the bottom one for twice that frequency. The RAS DDM algorithm is used without coarse grid.

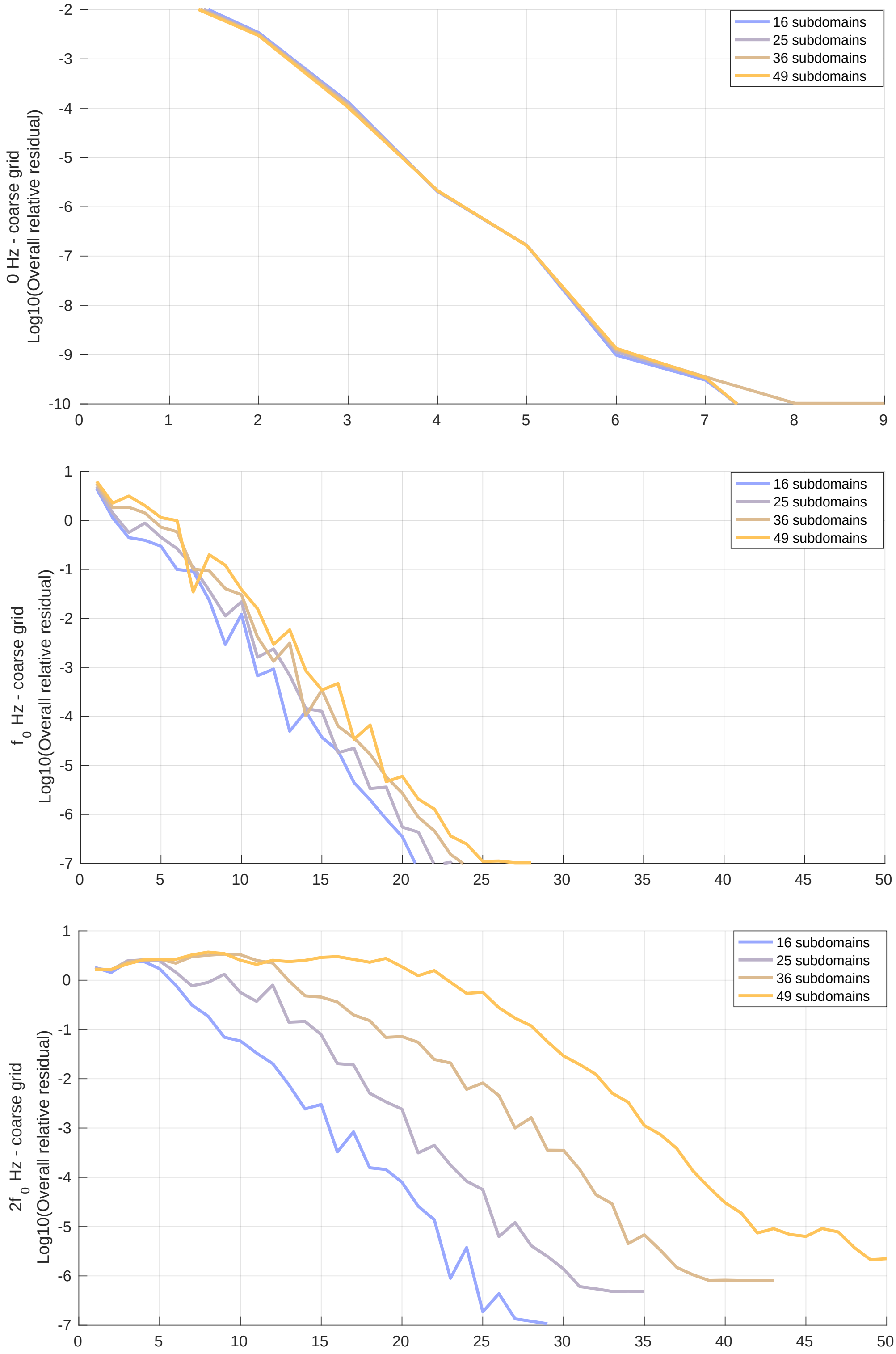


Figure 5.12: Influence of the number of subdomains on the convergence history for the 3D CMUT array with water. The top figure is for the constant harmonic, the middle one for the fundamental frequency harmonics and the bottom one for twice that frequency. The RAS DDM algorithm is used with a coarse grid.

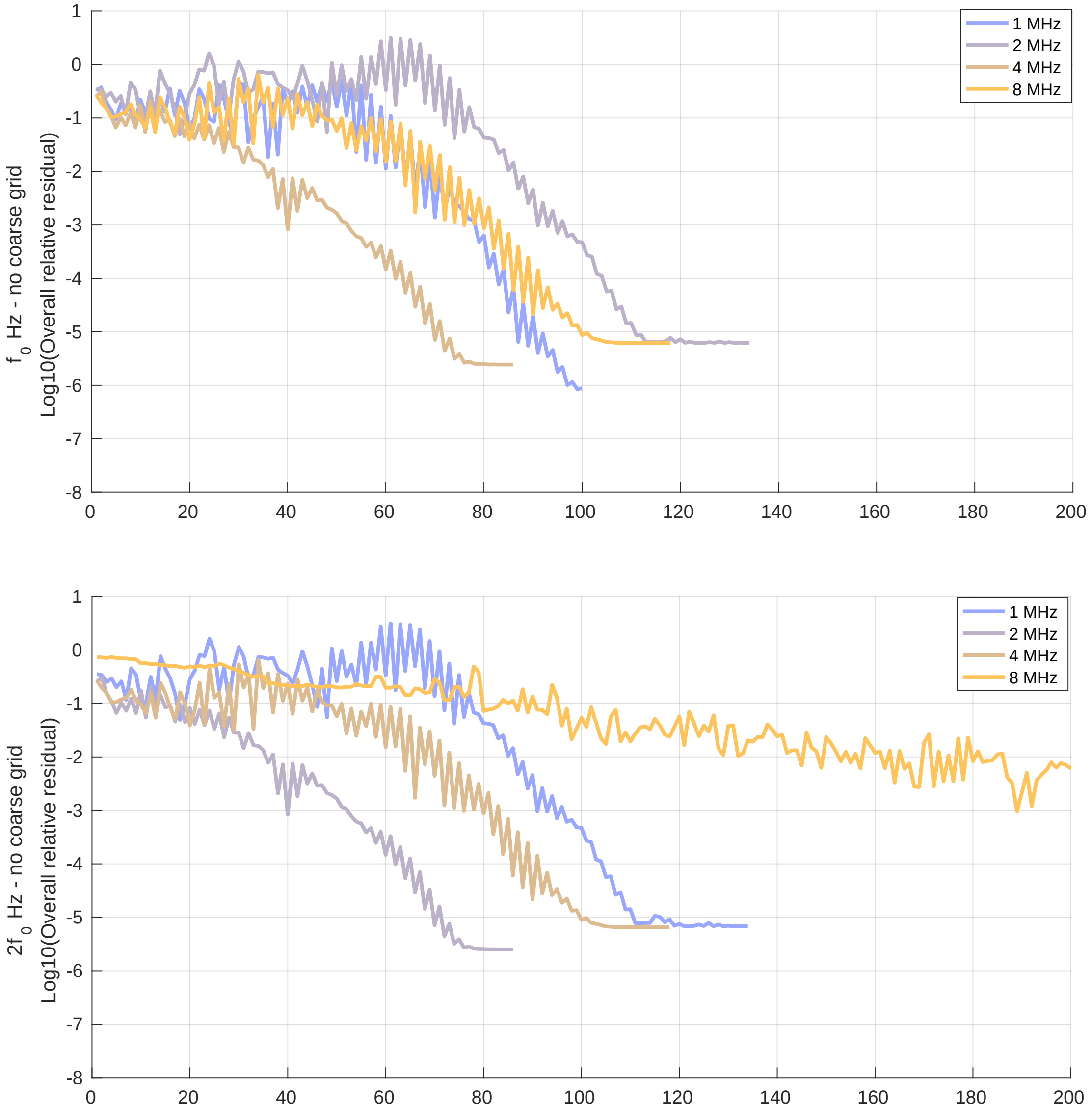


Figure 5.13: Influence of the excitation frequency on the convergence history for the 3D CMUT array with water. The top figure is for the fundamental frequency harmonics and the bottom one for twice that frequency. The RAS DDM algorithm is used without coarse grid.

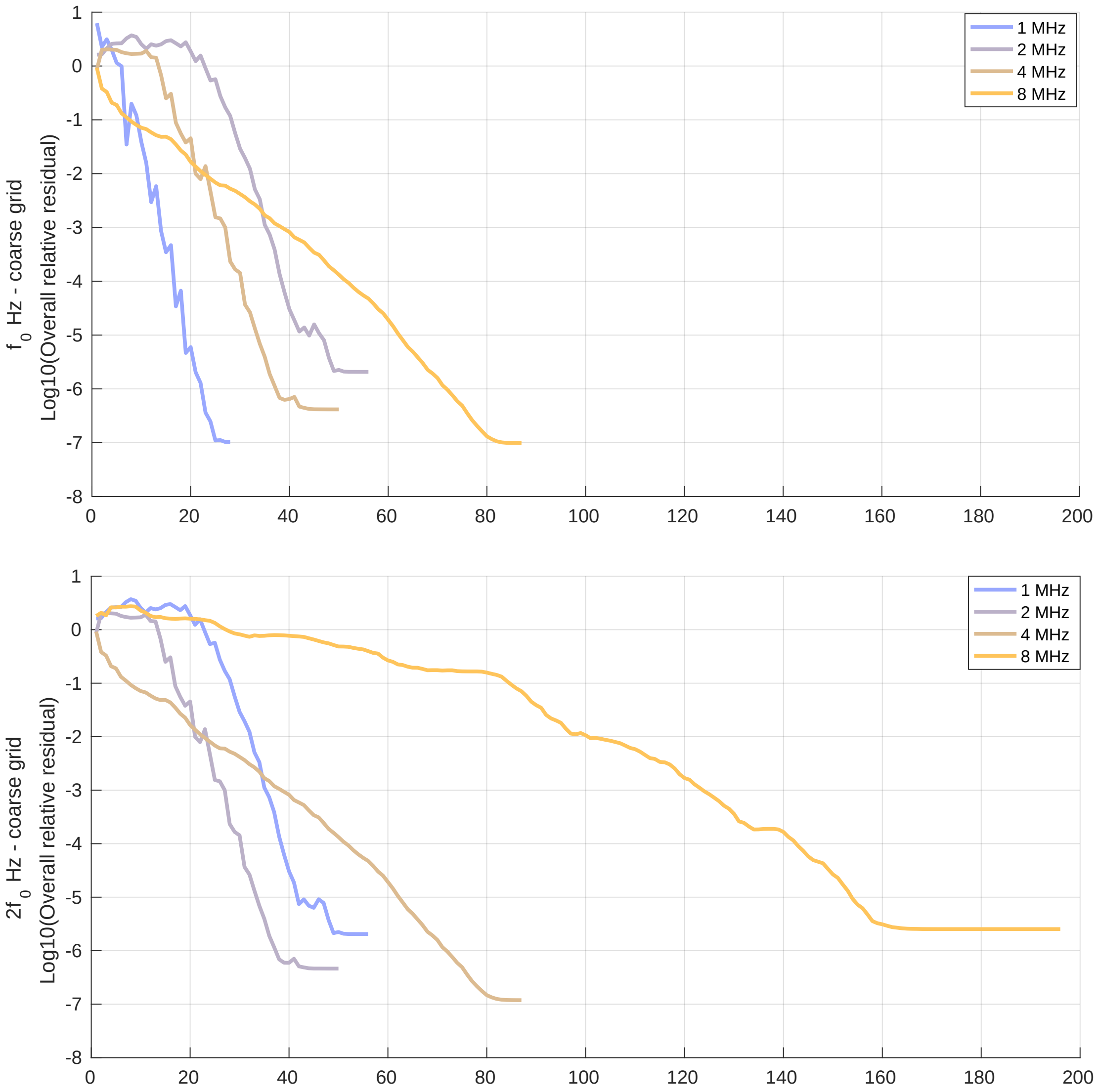


Figure 5.14: Influence of the excitation frequency on the convergence history for the 3D CMUT array with water. The top figure is for the fundamental frequency harmonics and the bottom one for twice that frequency. The RAS DDM algorithm is used with a coarse grid.

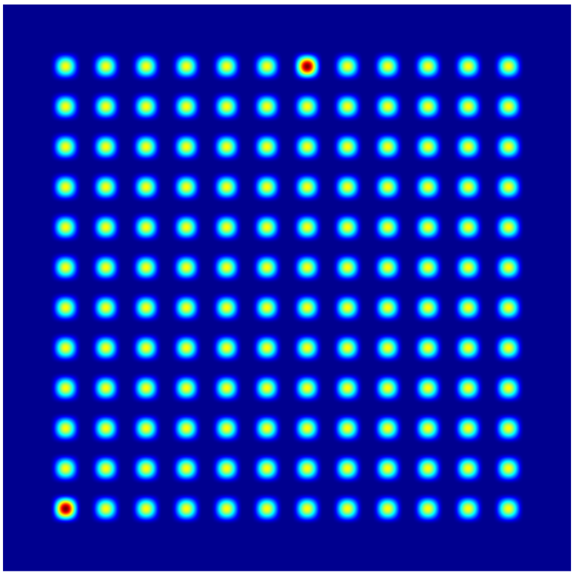


Figure 5.15: Top view of the constant deflection in the simulation of the $12x12$ CMUT array. Dark blue is 0 nm, dark red is 16 nm.

boundary are

$$\begin{cases} \partial_z u_x + \frac{1}{c_s}\frac{\partial u_x}{\partial t} = 0 \\ \partial_z u_y + \frac{1}{c_s}\frac{\partial u_y}{\partial t} = 0 \\ \partial_z u_z + \frac{1}{c_p}\frac{\partial u_z}{\partial t} = 0, \end{cases} \tag{5.8}$$

where c_p and c_s are respectively the propagation speed of the compression waves and of the shear waves. The speeds can be computed using the Lamé parameters

$$\begin{aligned} \lambda &= \frac{E\,\nu}{(1+\nu)(1-2\,\nu)} \\ \mu &= \frac{E}{2(1+\nu)} \end{aligned} \tag{5.9}$$

as follows:

$$\begin{aligned} c_p &= \sqrt{\frac{\mu}{\rho}} \\ c_s &= \sqrt{\frac{\lambda+2\mu}{\rho}}, \end{aligned} \tag{5.10}$$

so that for silicon $c_p = 9310$ m/s and $c_s = 4980$ m/s.

Because of the time derivative in the Kupraze interface conditions, supposing radiated pressure waves will add harmonics to the Fourier series of the displacement field and lead to a heavier problem to solve. While doing so is computationally tractable it will be avoided in the simulations of this section since the computing cluster used is designed for relatively low (4 GB) RAM per computing unit. Perfect radiation conditions are thus considered, leading to a mechanical displacement that can be written as

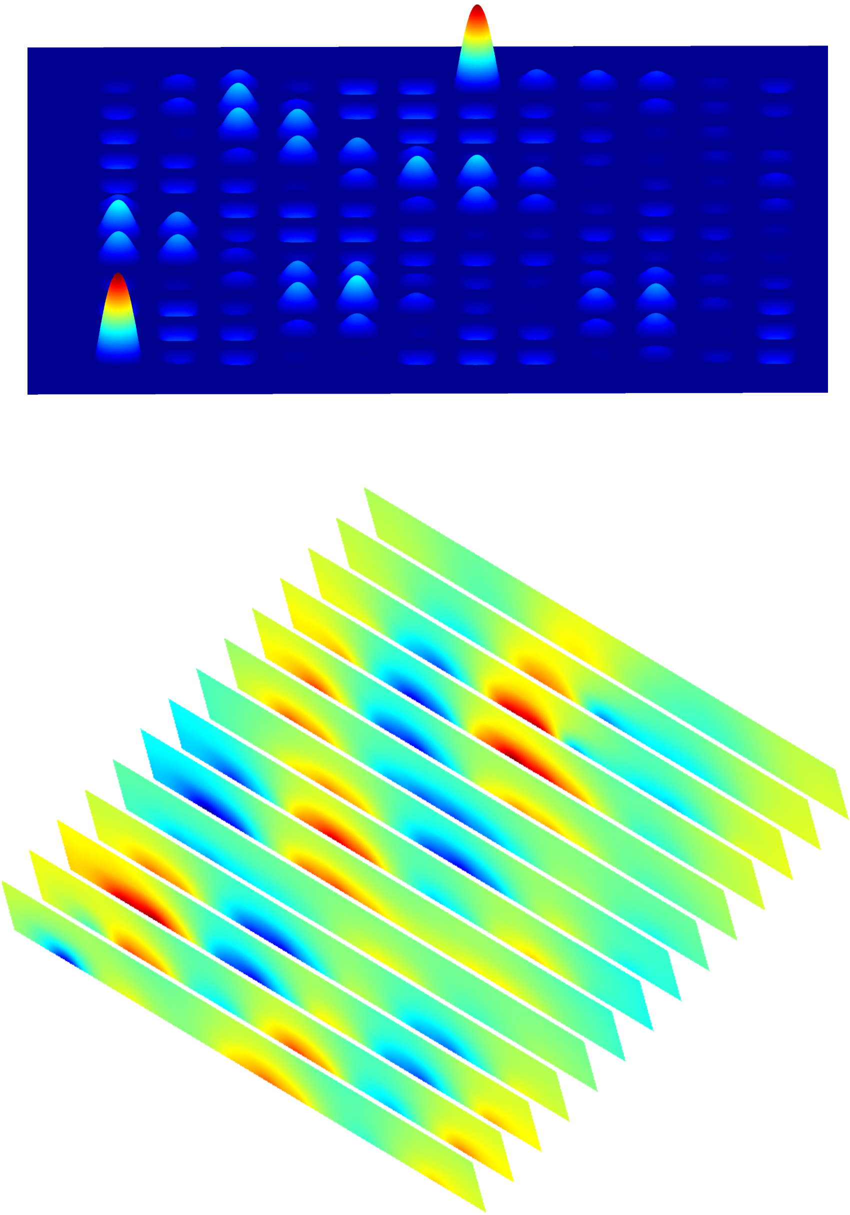
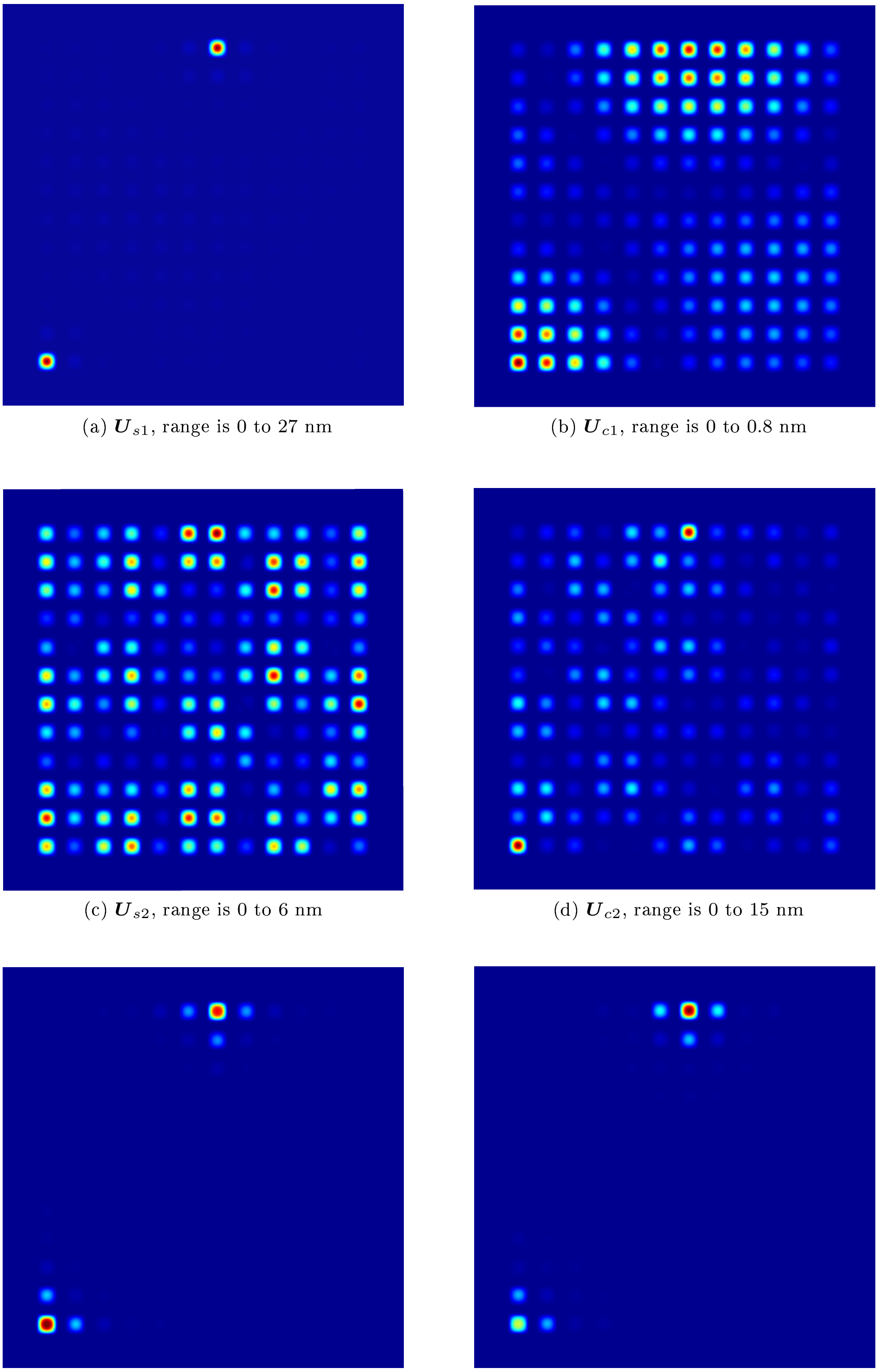


Figure 5.16: Side view of the $\boldsymbol{U}_{c2}$ displacement harmonic and δP_{c2} pressure harmonic in the simulation of the 12x12 CMUT array.



(a) $\boldsymbol{U}_{s1}$, range is 0 to 27 nm

(b) $\boldsymbol{U}_{c1}$, range is 0 to 0.8 nm

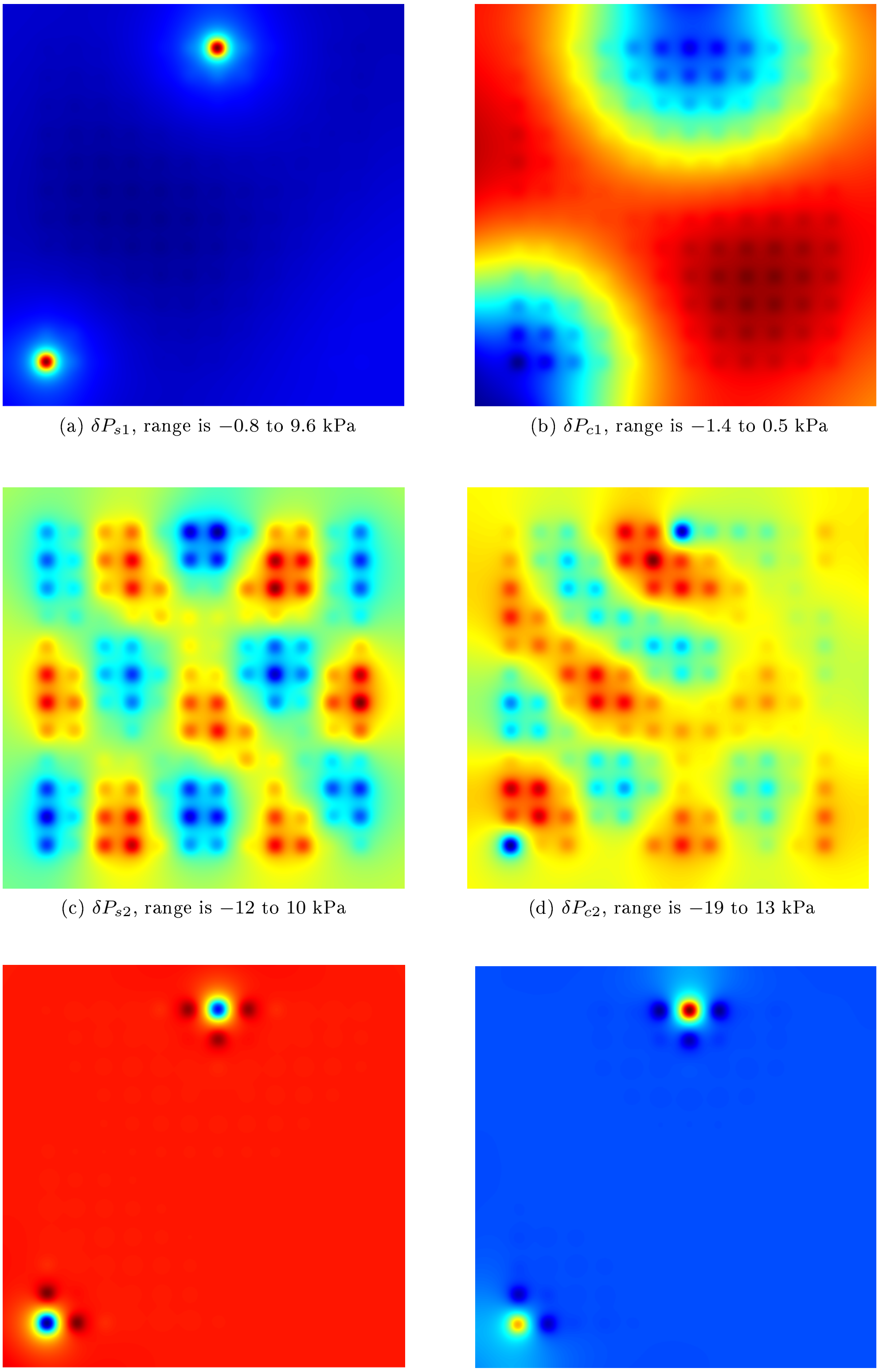
(c) $\boldsymbol{U}_{s2}$, range is 0 to 6 nm

(d) $\boldsymbol{U}_{c2}$, range is 0 to 15 nm

(e) $\boldsymbol{U}_{s3}$, range is 0 to 0.005 nm

(f) $\boldsymbol{U}_{c3}$, range is 0 to 0.004 nm

Figure 5.17: Top view of the displacement harmonics considered in the simulation of the 12x12 CMUT array. The constant deflection is not shown.



(a) δP_{s1}, range is -0.8 to 9.6 kPa

(b) δP_{c1}, range is -1.4 to 0.5 kPa

(c) δP_{s2}, range is -12 to 10 kPa

(d) δP_{c2}, range is -19 to 13 kPa

(e) δP_{s3}, range is -0.018 to 0.003 kPa

(f) δP_{c3}, range is -0.003 to 0.015 kPa

Figure 5.18: Top view of the 6 pressure harmonics considered in the simulation of the $12x12$ CMUT array. The pressure is shown at the membrane top.

$$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{U}_{c0}(\boldsymbol{x}) + \boldsymbol{U}_{s1}(\boldsymbol{x})\ \sin(2\pi f_0) + \boldsymbol{U}_{c2}(\boldsymbol{x})\ \cos(2\cdot 2\pi f_0) + ... \tag{5.11}$$

for the considered 70 V constant electric bias on all cells in the CMUT array and an additional 70 $\sin(2\pi f_0)$ V excitation on a single cell.

Figure 5.20 shows the influence of the number of subdomains in the array on the convergence of the RAS algorithm solving the elastoacoustic problem at the first staggered iteration. Since all harmonic frequencies are decoupled for the elastoacoustic problem at a given staggered iteration the convergence history is shown for the constant deflection $\boldsymbol{U}_{c0}$, the vibration at the excitation frequency $\boldsymbol{U}_{s1}$ and at twice that frequency $\boldsymbol{U}_{c2}$. The first three harmonics have been considered in (5.11) while f_0 is set to 1 MHz. To have an idea of the displacement field corresponding to the considered settings the $\boldsymbol{U}_{c2}$ harmonic is illustrated in figure 5.19.

Even though the crosstalk through elastic waves is much smaller than what was obtained through fluid coupling the required number of iterations is still relatively high without coarse grid correction. The general trend when the number of subdomains is increased is unsuprisingly a convergence slowdown. Furthermore the higher order harmonics, vibrating faster, tend to have a slower convergence, similarly to what was observed in 2D in figure 4.11.

When the coarse grid is considered figure 5.20 is obtained: the coarse grid works well, reducing the iteration count by a factor 3 and making the convergence independent of the number of subdomains. This could have been expected since the coarse grid, even though only using order one interpolations, is rather fine compared to the wavelength in the bulk domain and thus produces a rather accurate correction.

As a conclusion, crosstalk through elastic waves could also be simulated for large arrays. The crosstalk that could be obtained in the considered simulations was much weaker than via fluid coupling.

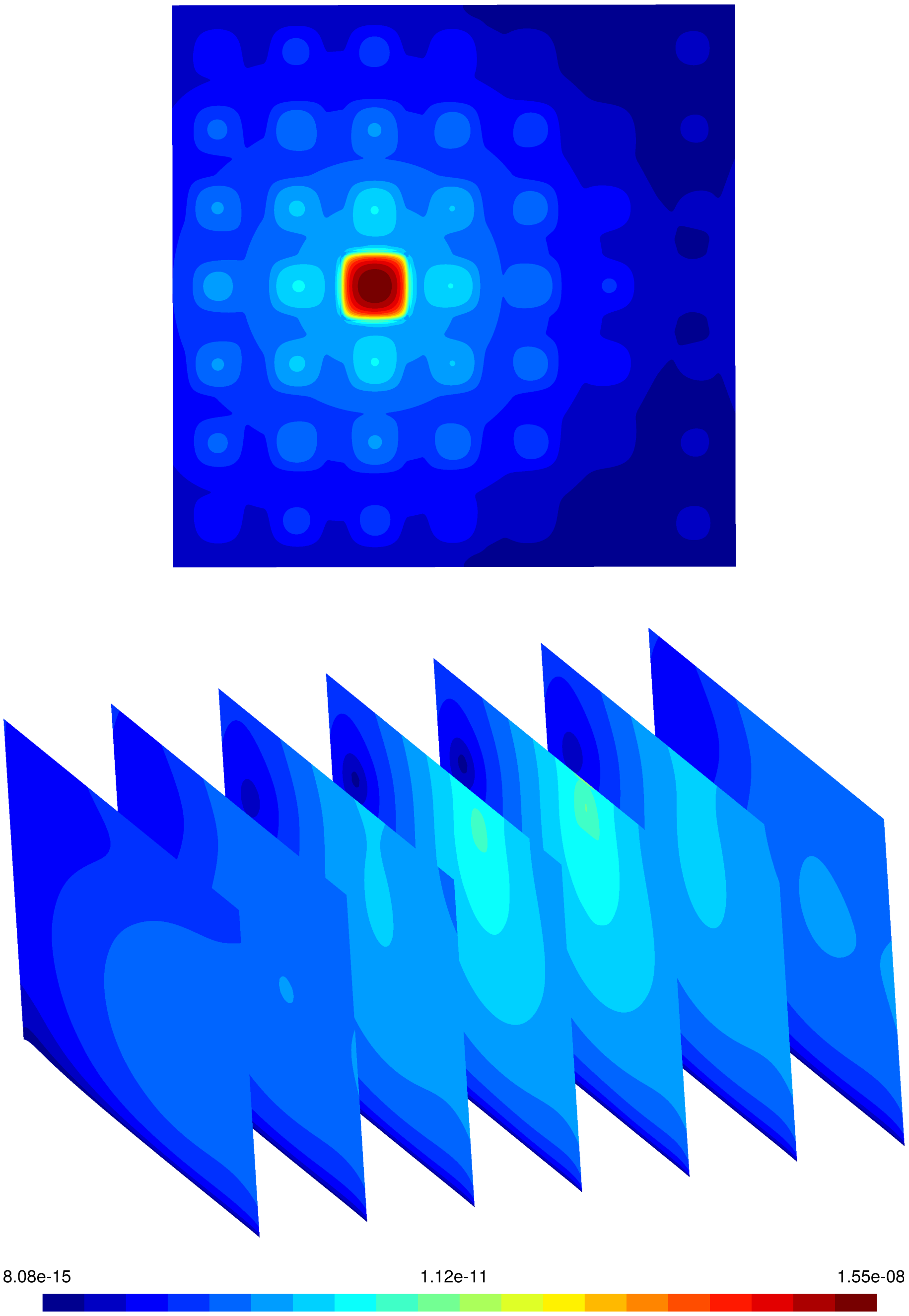


Figure 5.19: Displacement harmonic $\boldsymbol{U}_{c2}$ at the top of the membrane (top) and 3D view on bulk slices (bottom). Only a single cell is electrically excited at 2.5 MHz.

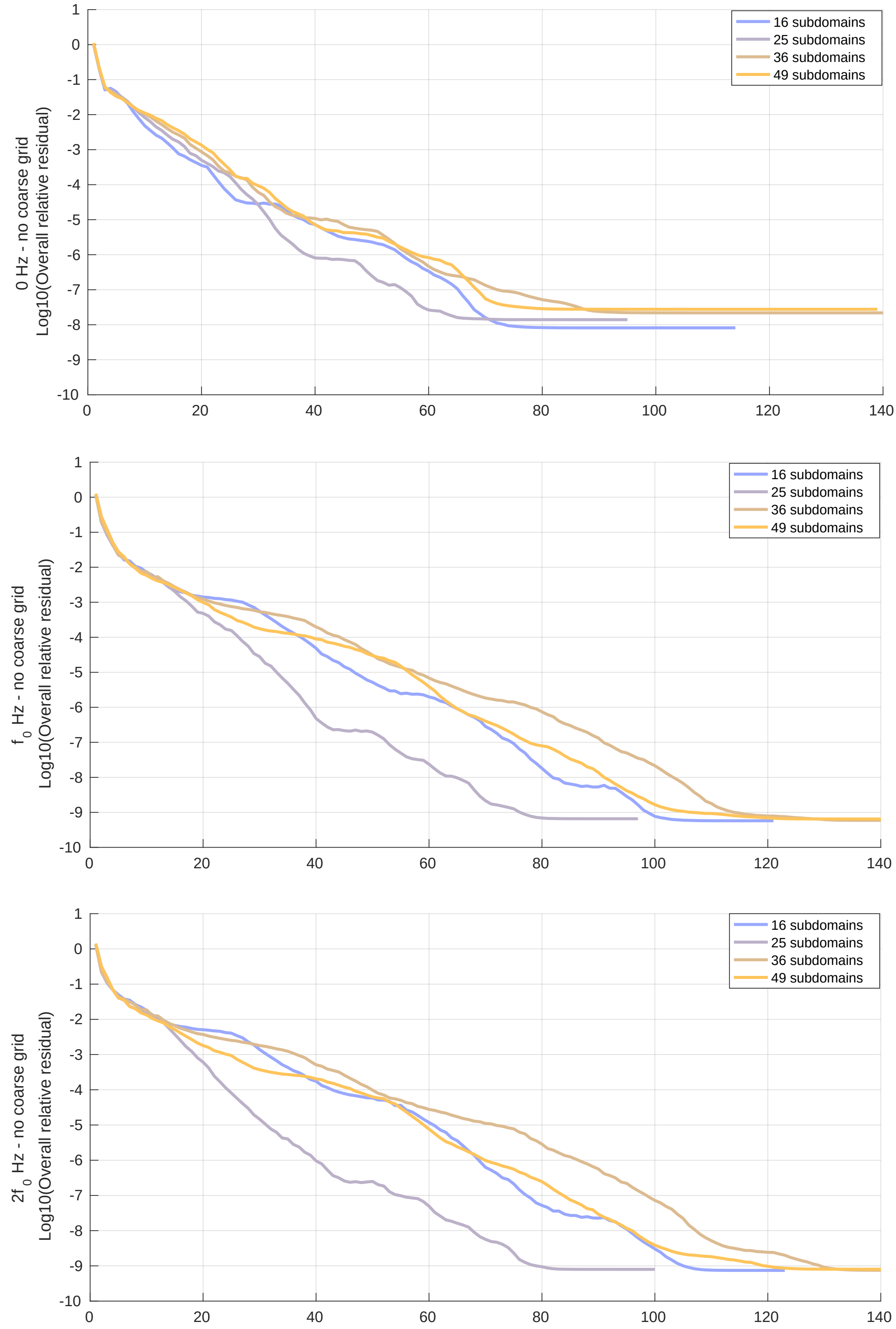


Figure 5.20: Influence of the number of subdomains on the convergence history for the 3D CMUT array without fluid. The top figure is for the $\boldsymbol{U}_{c0}$ harmonic, the middle one for $\boldsymbol{U}_{s1}$ and the bottom one for $\boldsymbol{U}_{c2}$. The RAS DDM algorithm is used without coarse grid.

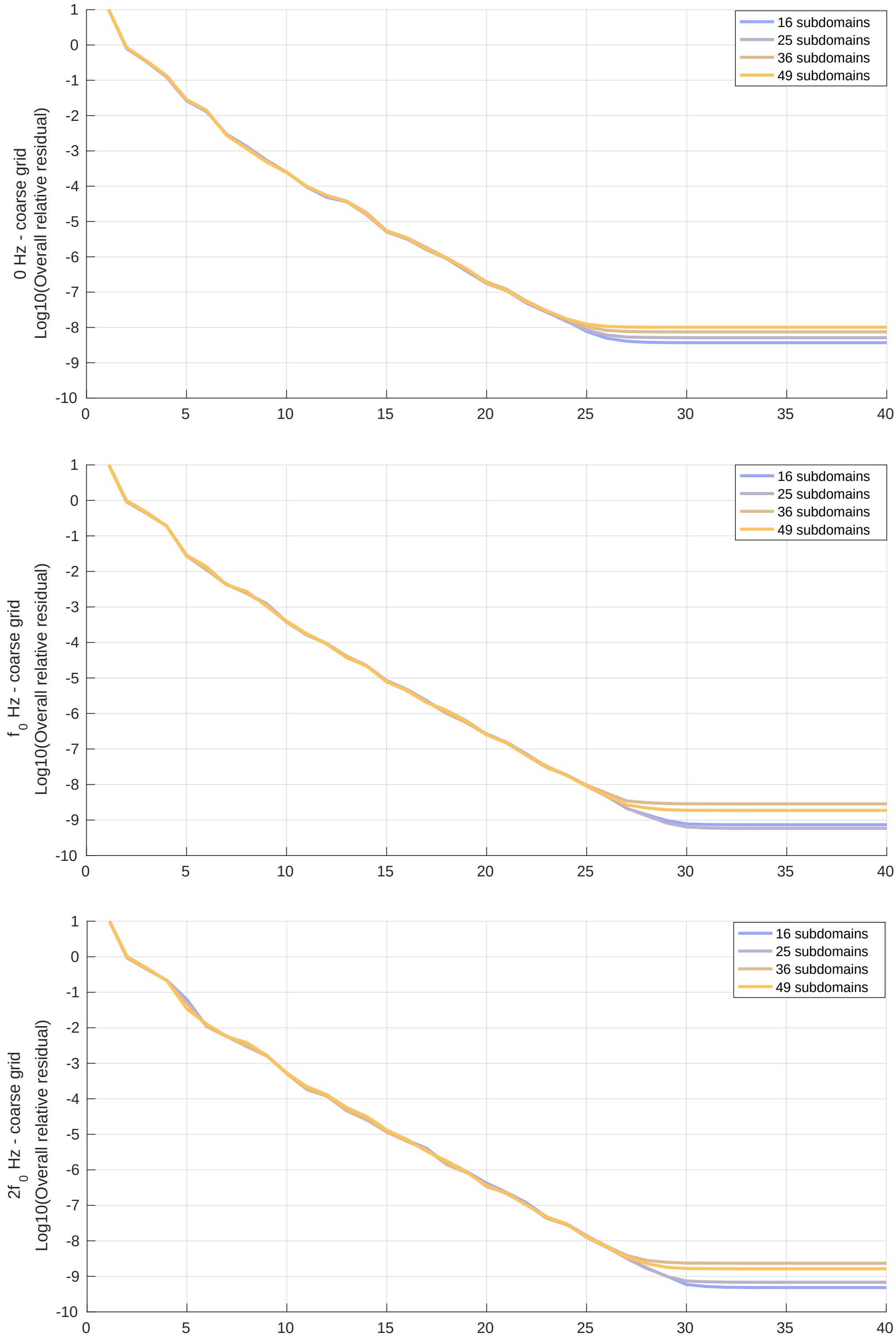


Figure 5.21: Influence of the number of subdomains on the convergence history for the 3D CMUT array without fluid. The top figure is for the $\boldsymbol{U}_{c0}$ harmonic, the middle one for $\boldsymbol{U}_{s1}$ and the bottom one for $\boldsymbol{U}_{c2}$. The RAS DDM algorithm is used with a coarse grid.

Conclusion

In this thesis a method to simulate in steady state large arrays of microelectromechanical devices (MEMS) vibrating in a fluid (e.g. CMUTs) has been proposed and its practical viability has been demonstrated. The method combines multiple ingredients which together make it possible to tackle the challenging, tens of millions degrees of freedom large, multiharmonic, multiphysics, nonlinear problem considered. The finite element method has been used in all simulations.

The first ingredient is the *domain decomposition method* (DDM), enabling the parallel computation on a large number of computing units. The major linear as well as intrinsically nonlinear DDM methods have been detailed and compared in terms of convergence rate, memory requirement and reusability of already available algebraic matrix decompositions. The linear DDM methods have been preferred for the rather smooth nonlinearity arising in microelectromechanical devices. Both the weakly coupled *staggered* resolution and the strongly-coupled *monolithic* resolution with Newton's method have been investigated in the DDM algorithms. The Newton iteration was preferred for large membrane deflections, very close to the pull-in instability thanks to its quadratic convergence for the nonlinear problem while in all other simulations the less memory demanding and lighter staggered resolution was preferred.

The second ingredient is the *multiharmonic resolution method* to accurately simulate the steady state vibration without the need of a time stepping method. A classical harmonic resolution could not be performed since the nonlinearity created new vibration harmonics that were not part of the electrical excitation signal and lead to a coupling of the harmonics corresponding to different frequencies. The multiharmonic method was implemented in an automatic way, computing the required formulations without user effort. To simulate the steady state of the considered devices it was observed that the multiharmonic resolution method is more systematic and requires much less computation power than a classical Newmark time stepping method. It was furthermore observed that the staggered nonlinear resolution approach enables to solve at every iteration the elastoacoutic problem, the heaviest problem at a given staggered iteration, for every harmonic frequency independently. Unlike with Newton's method this enabled to consider a large number of harmonics for a very good accuracy without requiring more memory and with a computation time proportional to the number of harmonic frequencies considered.

The third ingredient is to use the high-order finite element method to get accurate simulations with a dramatically reduced number of degrees of freedom. Even though the algebraic system to solve becomes denser it was observed that for the resolution of the mechanical problem using an order two interpolation instead of order one lead to a tenfold reduction of the number of degrees of freedom for a same accuracy. Higher orders, up to order six, have been considered but order two and three have been preferred.

As a proof of concept simulations have been performed in a couple of hours on large arrays of up to 196 elementary MEMS cells vibrating in water (CMUTs), with a 3D model of every cell leading to a system with about 20 million degrees of freedom. The simulation was performed for a large electrical excitation leading to a clearly nonlinear behaviour. With seven harmonics considered the steady state could be accurately simulated. The fluid crosstalk between individual cells as well as the nonlinear coupling between vibration harmonics could be clearly observed.

Perspectives

The simulation method proposed in this thesis has been applied to vibrating micromembranes and to CMUTs. This should not be seen as a limitation of the method to these two devices but rather as a guideline indicating to which devices the method is best suited. The DDM can be applied as described to general micro devices as long as the wavelength of the propagating waves is not much shorter than the size of an individual DDM subdomain. For such high frequency problems the boundary conditions detailed are not competitive anymore and more appropriate conditions [110, 83] should be used. The multiharmonic resolution method can be applied to a broad class of devices as long as the fields can be approximated with a reasonable number of harmonics. As a general rule the multiharmonic method should be competitive compared to a classical time resolution for low damping applications.

The improvements and extensions of this work can be classified in two categories. The first category concerns the numerical aspects and deals with speeding up the claculations. The second category deals with the physical modeling of the devices to obtain simulation results that closely match actual devices.

From a numerical point of view a major improvement would be to reduce the computation time required to simulate large arrays of devices while implementing everything in a proven finite element software. The major sources of speedup are the following:

- *Reduce the large communication cost by calling an efficient message passing library.*

  In the simulations all communication was done by writing and reading files on disk, leading to a large share of the computation time lost in communication. Calling a message passing library such as MPI can be the solution.

- *Using an optimised hp-fem code as well as an optimised mesh.*

  The mesh used in the 3D simulations of section 5.2 was obtained by extruding a 2D mesh and can thus be optimised to get the same accuracy at a lower computational cost. Using different interpolation orders for the mechanical displacement, the electric potential and the acoustic pressure field would provide an additional speedup.

- *Implementing in an efficient and general way the multiharmonic method.*

  As mentionned in section 3.3 the multiharmonic resolution method can be implemented for general nonlinear problems using external FFT libraries. Combined with a more efficient implementation avoiding the recomputation of terms that can be reused, the multiharmonic resolution can be accelerated.

From a physical modeling point of view a more accurate representation of actual MEMS devices can be obtained by taking into account:

- *Mechanical and acoustic damping sources.*

  The only damping source considered in this work comes from the radiated pressure waves. In practical devices the losses in the fluid and the membrane vibration provide extra damping, in particular for close to resonance excitations.

- *Squeeze film effect.*

  In practical CMUTs the vacuum gap below the membrane might be filled with some gas. As the membrane vibrates the gas alternatively expands and contracts. Taking into account this effect and the associated damping source will not only provide more accurate simulations but also allow the simulation of a wider class of MEMS.

- *Residual stresses.*

  The micromachining technologies used to produce the MEMS devices can create residual stresses in the device that influence its characteristics.

- *The geometrical nonlinearity in the membrane deformation.*

  Thin membranes exhibit in general a strong geometrical nonlinearity since their bending stiffness is tiny in the undeformed configuration and dramatically increases with the bending. Even though the membrane thickness considered in the simulations performed in this work is relatively large, a closer match with reality will be obtained by taking into account the geometrical nonlinearity.

Appendix A

Overview of the high order finite element method

A.1 Weak formulation and discretisation

Solving numerically a partial differential equation with the finite element method [72] always requires a discretisation of the problem. As an example when we are to solve the electrostatic model (1.4) for the electric potential v

$$\begin{cases} \mathrm{div}(\epsilon\,\nabla v) = -\rho_v & \text{on } \Omega, \\ v = 0 & \text{on } \Gamma, \end{cases} \tag{A.1}$$

using the Galerkin method the strong form (A.1) is not solved as such. It is instead solved in weak form: the first relation implies that

$$\int_\Omega \mathrm{div}(\epsilon\,\nabla v)\, v'\, d\Omega = \int_\Omega -\,\rho_v\, v'\, d\Omega, \tag{A.2}$$

holds for any appropriate function v', called *test function*. In (A.2) the unknown field v is then discretised as follows

$$v = \sum_{i=1}^{m} c_i\, N_i, \tag{A.3}$$

as a sum of products of constant coefficients c_i and space dependant shape functions $N_i(x)$. A well known example of an appropriate order one nodal shape function N_i is a piecewise linear function equal to one on a mesh node i and zero on all others. Figure A.1 illustrates such a shape function in 1D. As can be seen the shape function is non zero on only two mesh elements.

Because v' in (A.2) can be any appropriate function one can simply use $v'_i = N_i$ $(i = 1...m)$ so as to get m equations for the m unknown coefficients c_i:

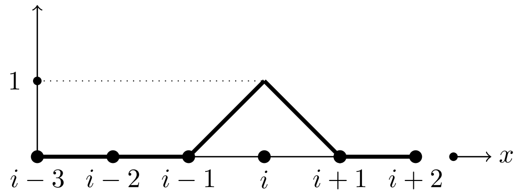


Figure A.1: First order 1D Lagrange shape function for node i

$$\begin{bmatrix} K_{1,1} & \cdots & K_{1,m} \\ \vdots & \ddots & \vdots \\ K_{m,1} & \cdots & K_{m,m} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}. \tag{A.4}$$

It is worth noting that since the shape functions in the finite element method are chosen with a compact support the matrix $\boldsymbol{K}$ corresponding to the system of m equations will be sparse.

As can be seen in (A.2), for the piecewise linear shape functions described the left side of the equation equals zero because of the second order space derivative. Fortunately one can apply Green's formula and get:

$$-\int_{\Omega} \epsilon \nabla v \cdot \nabla v' d\Omega + \int_{\partial\Omega} \epsilon\, v'\, \partial_{\boldsymbol{n}} v\, d\partial\Omega = \int_{\Omega} -\rho_v\, v'\, d\Omega, \tag{A.5}$$

where $\partial\Omega$ is the boundary of Ω and $\partial_{\boldsymbol{n}} v = \boldsymbol{n} \cdot \nabla v$ is the normal derivative of v. This form not only enables the use of linear shape functions but also gives easy access to the normal derivative: it will be used throughout the thesis.

Form (A.5) with the piecewise linear shape function can readily be assembled on all mesh elements to give a system of the form (A.4). As an example the left term assembled on a single line mesh element $\Omega_e \in \Omega$

$$-\int_{\Omega_e} \epsilon \nabla v \cdot \nabla v' d\Omega = -\int_{\Omega_e} \epsilon \nabla (\sum_{i=1}^{2} c_i\, N_i) \cdot \nabla v' d\Omega, \tag{A.6}$$

leads to an elementary matrix $\boldsymbol{K}_e$ equal to

$$-\int_{\Omega_e} \epsilon \begin{bmatrix} \nabla N_1 \cdot \nabla N_1 & \nabla N_2 \cdot \nabla N_1 \\ \nabla N_1 \cdot \nabla N_2 & \nabla N_2 \cdot \nabla N_2 \end{bmatrix} d\Omega, \tag{A.7}$$

where the first (second) row corresponds to the first (second) test function $v' = N_1$ ($v' = N_2$) while the first (second) column corresponds to the first (second) interpolation function $v = N_1$ ($v = N_2$). Adding together all elementary matrices at the right indexes leads to the full matrix $\boldsymbol{K}$.

A.2 High order discretisation

The finite element-computed approximation on a given mesh converges asymptotically to the actual solution as h^{p+1} where h quantifies the mesh size and decreases as the mesh is refined and p is the polynomial interpolation order of the fields on a mesh element. As a consequence for a given

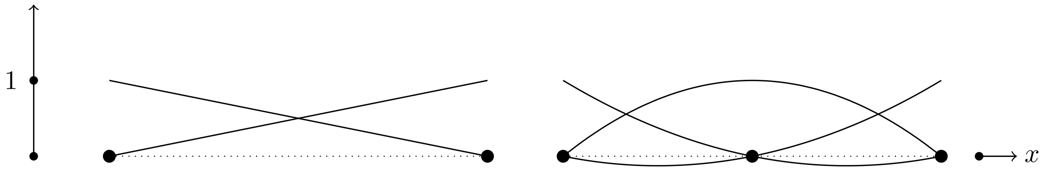


Figure A.2: Order 1 (left) and 2 (right) Lagrange shape functions on a 1D (line) element

mesh size a much more accurate approximation can be obtained by increasing the order p of the element shape functions. A classical high order shape function choice is *Lagrange* polynomials as already introduced in the previous section. The Lagrange shape function for node i on an n node 1D element is

$$N_i = \prod_{j=1, j \neq i}^{n} \frac{x - x_j}{x_i - x_j}, \tag{A.8}$$

where x is the 1D coordinate. In other words the shape function equals zero on all nodes but the *ith* on which it equals one. Figure A.2 illustrates the Lagrange shape functions for orders 1 and 2 on a 1D (line) element. As can be seen the shape functions associated to the external nodes in the first order and second order line element are not the same. In higher dimension elements this leads to shape function discontinuities at the interface of adjacent elements of different orders: Lagrange shape functions are not *hierarchical* and different interpolation orders cannot be combined in the same mesh.

Work [130] conveniently provides high order hierarchical shape functions for all classical elements but the pyramid and will be used throughout this thesis. In this case the set of shape functions for an order $p + 1$ element includes the shape functions for the order p element which makes them hierarchical. The shape functions can be classified into 4 categories: those associated to

- **nodes** (N_n): a shape function associated to node i is non zero on node i and zero on all other nodes in the element
- **edges** (N_e): a shape function associated to edge i is non zero on edge i and zero on all other edges in the element
- **surfaces** (N_s): a shape function associated to surface i is non zero on surface i and zero on all other surfaces in the element
- **volumes** (N_v): a shape function associated to volume i is non zero on volume i and zero on all other volumes in the element

An unknown field v can be spatially interpolated as

$$v = \sum_{i=1}^{\#\text{nodes}} c_{ni} \, N_{ni} + \sum_{i=1}^{\#\text{edges}} c_{ei} \, N_{ei} + \sum_{i=1}^{\#\text{faces}} c_{si} \, N_{si} + \sum_{i=1}^{\#\text{volumes}} c_{vi} \, N_{vi}. \tag{A.9}$$

The scalar nodal shape functions for a second order quadrilateral element are illustrated in figure A.3 (associated to a node, edge and surface from left to right). The number of node, edge, surface

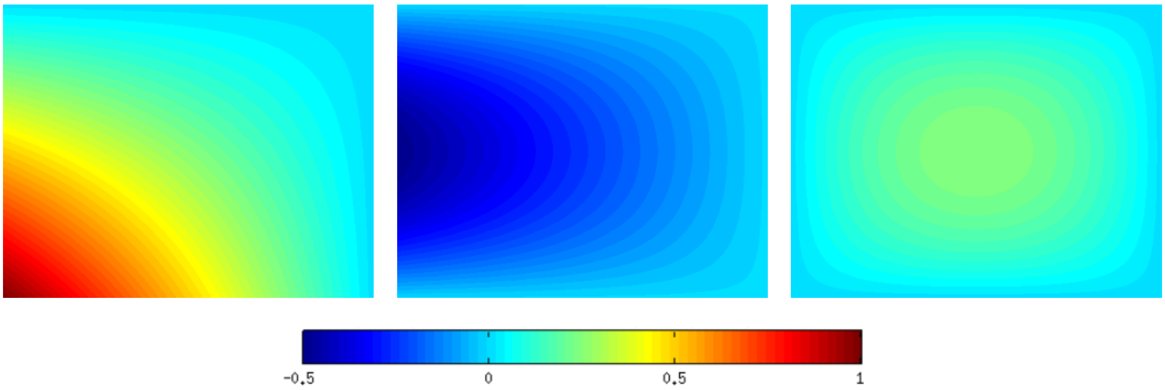


Figure A.3: Illustration of quadrilateral order two hierarchical shape functions associated to a node, edge and surface (from left to right)

Table A.1: Number of node, edge, surface and volume shape functions for order 1 through 5 quadrangle and hexahedron element

Quad. order	**1**	**2**	**3**	**4**	**5**
Node	4	4	4	4	4
Edge	0	4	8	12	16
Surface	0	1	4	9	16
Volume	0	0	0	0	0
Sum	4	9	16	25	36

Hex. order	**1**	**2**	**3**	**4**	**5**
Node	8	8	8	8	8
Edge	0	12	24	36	48
Surface	0	6	24	54	96
Volume	0	1	8	27	64
Sum	8	27	64	125	216

and volume shape functions versus element order is shown in figure A.1 for the quadrangle and hexahedron element.

A.3 Orientation

The edge and surface associated shape functions defined in [130] depend on the edge and surface orientation. Orienting the edges (and surfaces in 3D) in a same way on all adjacent elements is of crucial importance to guarantee shape function continuity at the element interfaces. For the quadrangle in figure A.4 for example the order 3 left edge associated shape function can have two forms depicted in figure A.5 depending on the left edge orientation 1 → 2 or 2 → 1: any orientation is valid but the same orientation must be chosen for the adjacent edge of the neighbouring element. In case of surface associated shape functions there are $2^4 = 8$ possible orientations in theory. In 2D however any choice of orientation is appropriate since the surface shape function equals zero at the element interface. Similarly in 3D any orientation of volume associated shape functions is valid.

The following procedure to orient edges (and surfaces in 3D), based on the element node numbering given by the meshing software, is based on [130]:

- edges defined by node with numbers a and b are oriented $a \rightarrow b$ with $a > b$

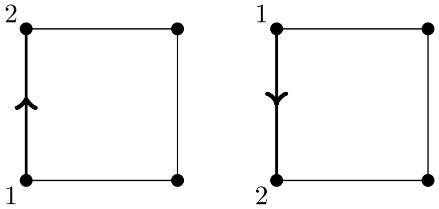


Figure A.4: Quadrangle with two possible edge orientations

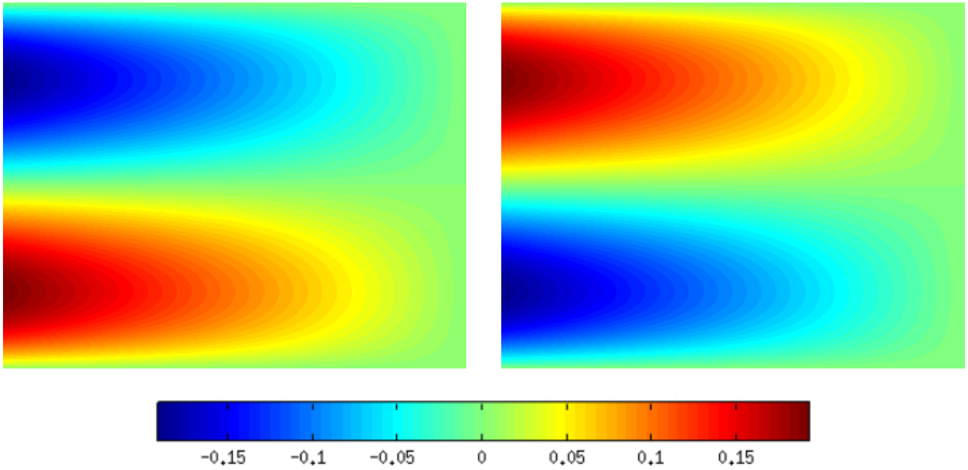


Figure A.5: Order 3 shape function for two opposite edge orientations

- triangular surfaces defined by nodes with numbers a, b and c are oriented $a \rightarrow b \rightarrow c$ with $a > b > c$

- quadrangular surfaces defined by nodes with numbers a, b, c and d are oriented as $a \rightarrow b \rightarrow c \rightarrow d$ with $a = \max(a, b, c, d)$, a and c correspond to opposite corners and $b > d$

Because mesh node numbers are shared by adjacent elements the above procedure guarantees shape function continuity at the interface.

In this thesis computations will have to be performed on multiple overlapping meshes with same elements and node coordinates on the overlap but different node numbering. In such a configuration the above procedure cannot be used as such and an extra step is first performed:

1. Remove the node coordinates roundoff noise

2. Sort the nodes according to their coordinates with coordinate priority $x > y > z$

3. Renumber nodes accordingly

Figure A.6 illustrates this step for two overlapping meshes with 2 quadrangles in each. The renumbering guarantees same $>$ and $<$ relations between node numbers on the overlap for the different meshes. Edge and surface shape functions are thus guaranteed to be identical. The renumbering and orientation steps are performed once and for all when the mesh is loaded.

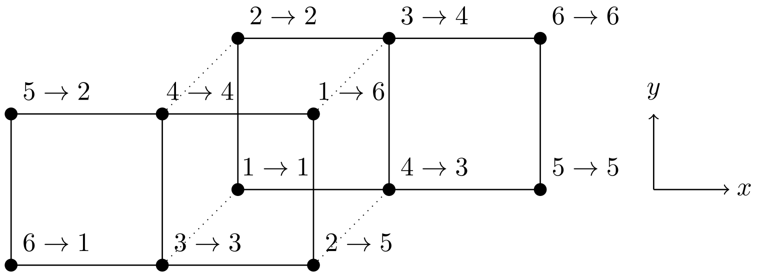


Figure A.6: Coordinate-wise node renumbering procedure for an overlapping mesh

A.4 Mapping on the reference element

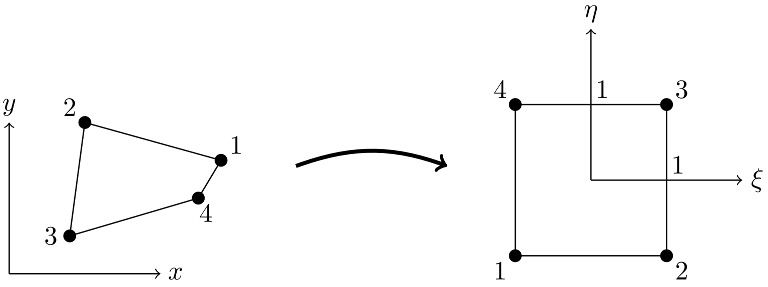


Figure A.7: Physical quadrangle brought back to the reference quadrangle

In the finite element method the actual computation is performed on standard *reference* elements. Figure A.7 illustrates a real quadrangle in the mesh brought back to the reference quadrangle element. Doing so simplifies the spatial derivation and numerical integration as the reference is the same for any quadrangle in the mesh. The formulations have however to be adapted by introducing a variable change Jacobian matrix. Considering only straight finite elements one can use the already introduced order 1 Lagrange shape functions N to write the coordinate change relation:

$$\begin{cases} x = \sum_{i=1}^{4} x_i N_i(\xi,\eta), \\ y = \sum_{i=1}^{4} y_i N_i(\xi,\eta), \end{cases} \tag{A.10}$$

so that the Jacobian is

$$\boldsymbol{J} = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \sum_{i=1}^{4} \begin{bmatrix} x_i \frac{\partial N_i(\xi,\eta)}{\partial \xi} & y_i \frac{\partial N_i(\xi,\eta)}{\partial \xi} \\ x_i \frac{\partial N_i(\xi,\eta)}{\partial \eta} & y_i \frac{\partial N_i(\xi,\eta)}{\partial \eta} \end{bmatrix}, \tag{A.11}$$

where $\frac{\partial N_i(\xi,\eta)}{\partial \xi}$ and $\frac{\partial N_i(\xi,\eta)}{\partial \eta}$ are the Lagrange shape function derivatives in the reference element. The two quantities can be easily derived analytically since the shape functions are simple polynomials in ξ and η.

A spatial derivative in the formulation can be rewritten using reference derivatives and elements of the inverse Jacobian matrix since $\boldsymbol{J}^{-1} = \frac{\partial(\xi,\eta)}{\partial(x,y)}$. As an example the left term in (A.5) computed on a quadrangle element Ω_e can be computed on the reference element Ω_{ref} as follows:

$$- \int_{\Omega_e} \epsilon \nabla v \cdot \nabla v' d\Omega = - \int_{\Omega_{ref}} \epsilon \, \boldsymbol{J}^{-1} \nabla v \cdot \boldsymbol{J}^{-1} \nabla v' |\boldsymbol{J}| d\Omega_{ref}, \tag{A.12}$$

where $|\boldsymbol{J}|$ is the Jacobian determinant taking into account the local surface change.

In this thesis the mesh will be deformed by a known mechanical displacement field $\boldsymbol{u}$ defined in the already mentionned high order hierarchical basis ($\boldsymbol{u}$ does not distort the elements). In order to take the induced curvature on the element into account the new Jacobian, called $\boldsymbol{J}'$, becomes

$$\boldsymbol{J}' = \frac{\partial(x + u_x, y + u_y)}{\partial(\xi,\eta)} = \boldsymbol{J} + \frac{\partial(u_x, u_y)}{\partial(\xi,\eta)} = \boldsymbol{J} + \boldsymbol{J}_u, \tag{A.13}$$

where $\boldsymbol{J}_u$ can be easily evaluated pointwise while fully taking into account the curvature in the high order interpolation of $\boldsymbol{u}$.

A.5 Numerical integration

Numerical integration in finite elements is typically performed on the reference element using a Gaussian quadrature method. For the reference quadrangle in figure A.7 it consists in computing

$$\int_{-1}^{1} \int_{-1}^{1} f(\xi,\eta) \, d\xi d\eta = \sum_{i=1}^{n} w_i f(\xi_i, \eta_i), \tag{A.14}$$

only requiring n pointwise evaluations of f at the Gauss points of coordinates (ξ_i, η_i). Quantity w_i gives a weight to every Gauss point.

For a high order element the function f is a high order polynomial and a high number of Gauss points are required for an exact integration. For a linear f however a single Gauss point is required at $(\xi,\eta) = (0,0)$ with a weight $w = 4$ for the reference quadrangle in figure A.7. Figure A.8 illustrates the Gauss points positions for order 0 through 5 (less nodes can sometimes be used with an optimised choice of point coordinates).

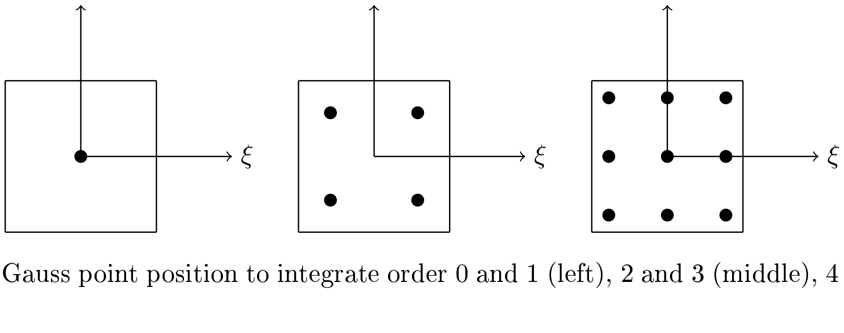


Figure A.8: Gauss point position to integrate order 0 and 1 (left), 2 and 3 (middle), 4 and 5 (right)

In this thesis some high order finite element formulations will be underintegrated [131, 132], i.e. the number of Gauss points will be lower than for an exact integration. This allows to speed up the finite element matrix generation. The integration degree will however be chosen high enough so that the effect on the simulation is negligeable.

A.6 Matrix assembly

In order to get the finite element approximate solution of a problem a system in the form (A.4) in which the stiffness matrix $\boldsymbol{K}$ contains all equations ruling the set of unknown shape function coefficients has to be assembled and solved. $\boldsymbol{K}$ is the sum (at the right indexes) of elementary matrices $\boldsymbol{K}_e$ assembled at the element level. The elementary matrix of an element with n shape functions is a dense n by n size matrix as illustrated in A.7. For high order elements the number of shape functions dramatically increases as reported in table A.1 while the number of Gauss points follows the same trend. As a result care has to be taken when writing a high order finite element assembler otherwise the matrix generation time might well exceed the system solve time. Because in this thesis a Matlab code is used any `for` loop at this step has to be avoided. The assembling step is fully vectorised for an orders of magnitude faster assembly.

Assembling any elementary matrix contribution (previously scalarised, with or without spatial unknown or test function derivatives)

$$\int_{\Omega_e} c\, v\, v'\, d\Omega_e, \tag{A.15}$$

on element Ω_e where c includes all terms but the unknown v and the test function v' is done at once for all elements of the same type in a single Matlab line using the speed optimised binary singleton expansion function `bsxfun`:

```
a=sum(bsxfun(@times,bsxfun(@times,coefmatrix,testfunmatrix),
permute(interpolfunmatrix,[1 2 4 3])),2)
a=permute(a, [1 3 4 2]);
```

where

- coefmatrix(i,j) equals c evaluated at element i, Gauss point j. It includes the Gauss point weights and all coordinate change Jacobian terms
- interpolfunmatrix(i,j,k) is the kth shape function (or its derivative on the reference element) on element i, computed at Gauss point j and associated to the unknown v
- testfunmatrix(i,j,k) is the kth shape function (or its derivative on the reference element) on element i, computed at Gauss point j and associated to the test function v'

After this call $a(i,j,k)$ gives the element corresponding to the jth test function and kth interpolation function in the elementary matrix of element i. Retrieving the interpolation and test function adresses in the full matrix is fully vectorised as well. Getting the sparse stiffness matrix uses the `sparse` Matlab function whose main inputs are the values in all elementary matrices and their corresponding row and column indexes in the global assembled matrix. Figure A.10 shows assembly times (computation of the stiffness terms and `sparse` call) for the left term in the electrostatic formulation (A.5) for several triangle/tetrahedron orders and mesh densities in 2D and 3D. Table A.2 shows the number of elements assembled per second. Dirichlet boundary conditions are taken into account. The number of Gauss points is such that a polynomial of degree twice the element

Order	**1**	**2**	**3**	**4**	**5**
2D triangles	287000	77000	25000	11000	5000
3D tetrahedra	140000	21000	4000	1000	400

Table A.2: Number of triangles and tetrahedra assembled per second for the 2D and 3D electrostatic formulation (2.66) (without volumic charges but with Dirichlet conditions instead) versus element order. The number of Gauss integration points is such that twice the element order can be exactly integrated.

order can be exactly integrated. The mesh is unstructured and the symmetry property of the generated matrix is not exploited. Computations are performed on a modern laptop with 16 GB RAM.

A.7 High order element validation

High order quadrangles and hexahedra are used throughout the thesis. They have been validated in the homemade Matlab code via the drop rate of the error (L_2 norm) between the finite element approximation v

$$\int_\Omega v\, v'\, d\Omega = \int_\Omega \sin(10x)\, \cos(10y)\, v'\, d\Omega, \tag{A.16}$$

and the actual solution $\sin(10x)\,\cos(10y)$ on a square/cubic geometry with a structured mesh. The error should asymptotically decrease at a rate h^{p+1} where p is the element order and h the number of elements in each space direction x, y (and z). The convergence curve is plotted in figure A.9 for the quadrangle element as an illustration. The slopes for the highest h are -2.003, -3.005, -4.016, -5.029, -6.065 for order 1 through 5 respectively, as expected.

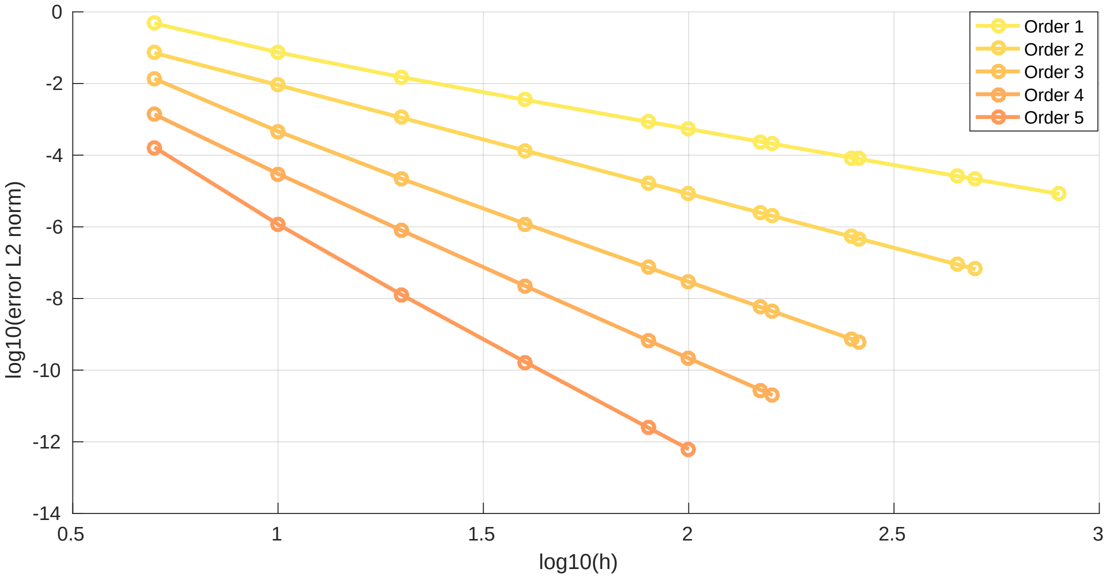


Figure A.9: Convergence curve for a high order quadrangle

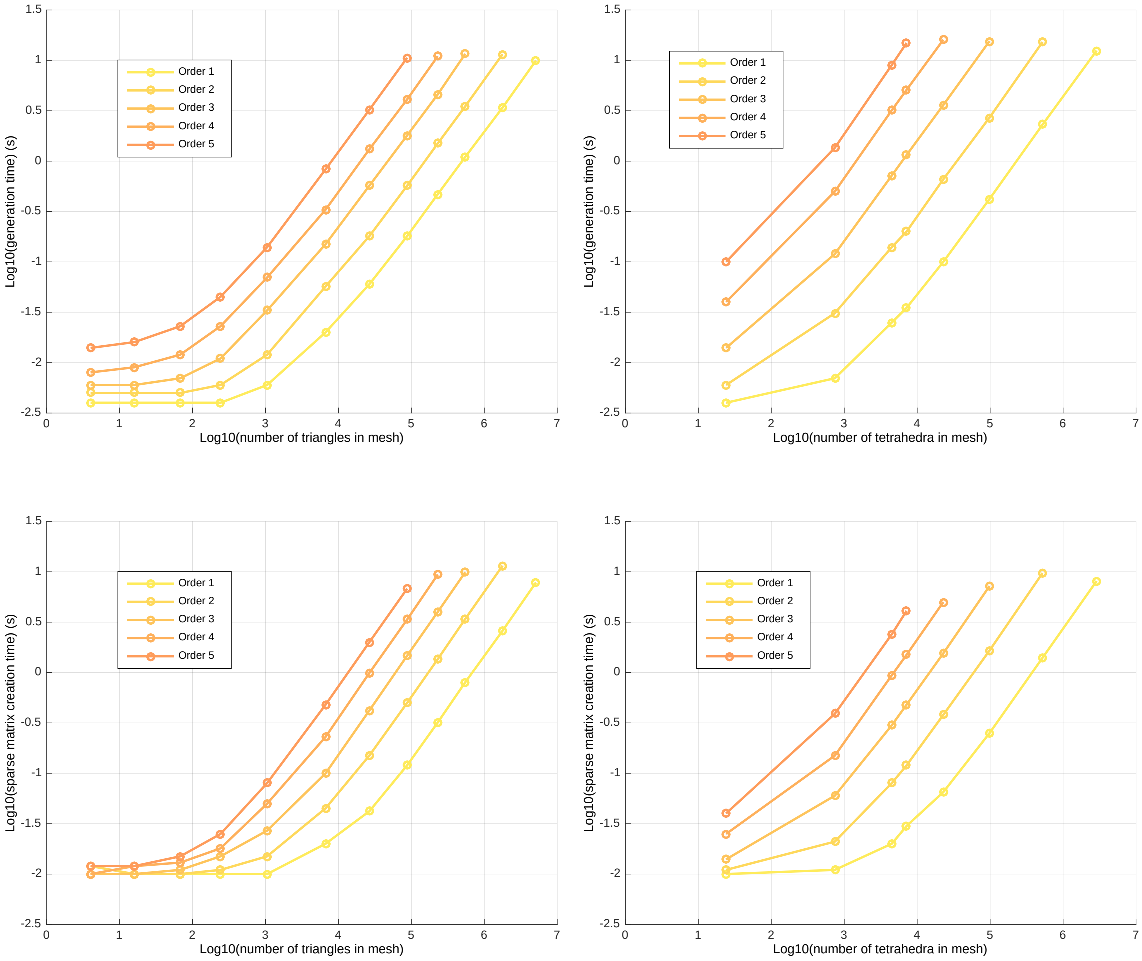


Figure A.10: Time to compute the terms of the electrostatic stiffness matrix (top) and to build the sparse matrix (bottom) on a 2D triangular (left) and 3D tetrahedral (right) mesh with increasing element order. Total generation time is the sum of top and bottom.

Bibliography

[1] M. Berthillier, P. Le Moal, and J. Lardies. Dynamic and acoustic modeling of capacitive micromachined ultrasonic transducers. In *2011 IEEE International Ultrasonics Symposium*, pages 608–611. IEEE, 2011.

[2] M. Berthillier, P. Le Moal, and J. Lardies. Comparison of various models to compute the vibro-acoustic response of large CMUT arrays. In *Acoustics 2012*, 2012.

[3] V. Rochus, D. J. Rixen, and J.-C. Golinval. Monolithical modeling of electro-mechanical coupling in micro-structures. *International Journal for Numerical Methods in Engineering*, 65(4):474–482, 2006.

[4] A. Caronti, G. Caliano, R. Carotenuto, A. Savoia, M. Pappalardo, E. Cianci, and V. Foglietti. Capacitive micromachined ultrasonic transducer (CMUT) arrays for medical imaging. *Microelectronics Journal*, 37(8):770–777, 2006.

[5] J. Song, S. Jung, Y. Kim, K. Cho, B. Kim, S. Lee, J. Na, I. Yang, O.-K. Kwon, and D. Kim. Reconfigurable 2D CMUT-asic arrays for 3d ultrasound image. In *SPIE Medical Imaging*, pages 83201A–83201A. International Society for Optics and Photonics, 2012.

[6] D. M. Mills. Medical imaging with capacitive micromachined ultrasound transducer (CMUT) arrays. In *Ultrasonics Symposium, 2004 IEEE*, volume 1, pages 384–390. IEEE, 2004.

[7] W. R. Dreschel, T. Kling, J. W. Sliwa, S. L. Carp, and S. Ayter. Diagnostic medical ultrasound systems and transducers utilizing micro-mechanical components, November 11 2003. US Patent 6,645,145.

[8] D. M. Mills and L. S. Smith. Real-time in-vivo imaging with capacitive micromachined ultrasound transducer (CMUT) linear arrays. In *Ultrasonics, 2003 IEEE Symposium on*, volume 1, pages 568–571. IEEE, 2003.

[9] X. Wang, Y. Fan, W.-C. Tian, H.-J. Kwon, S. Kennerly, G Claydon, and A. May. An air-coupled capacitive micromachined ultrasound transducer for noncontact nondestructive evaluation. In *Sensors, 2007 IEEE*, pages 1464–1467. IEEE, 2007.

[10] X. Wang, Y. Fan, W.-C. Tian, H.-J. Kwon, S. Kennerly, G. Claydon, and A. May. Development of air-coupled ultrasound transducers for nondestructive evaluation. In *Micro Electro Mechanical Systems, 2008. MEMS 2008. IEEE 21st International Conference on*, pages 932–935. IEEE, 2008.

[11] M. C. Remillieux, B. E. Anderson, T. J. Ulrich, P.-Y. Le Bas, M. R. Haberman, and J. Zhu. Review of air-coupled transduction for nondestructive testing and evaluation. *Acoust. Today*, 10(3):36–45, 2014.

[12] R. Manwar and S. Chowdhury. Experimental analysis of bisbenzocyclobutene bonded capacitive micromachined ultrasonic transducers. *Sensors*, 16(7):959, 2016.

[13] D. T. Yeh, O. Oralkan, I. O. Wygant, M. O'Donnell, and B. T. Khuri-Yakub. 3-D ultrasound imaging using a forward-looking CMUT ring array for intravascular/intracardiac applications. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 53(6):1202–1211, 2006.

[14] K. Chen, H.-S. Lee, A. P. Chandrakasan, and C. G. Sodini. Ultrasonic imaging transceiver design for CMUT: A three-level 30-vpp pulse-shaping pulser with improved efficiency and a noise-optimized receiver. *IEEE Journal of Solid-State Circuits*, 48(11):2734–2745, 2013.

[15] A. Nikoozadeh, O. Oralkan, M. Gencel, J.-W. Choe, D. N. Stephens, A. De La Rama, P. Chen, F. Lin, A. Dentinger, D. Wildes, et al. Forward-looking intracardiac imaging catheters using fully integrated CMUT arrays. In *2010 IEEE International Ultrasonics Symposium*, pages 770–773. IEEE, 2010.

[16] I. O. Wygant, X. Zhuang, D. T. Yeh, O. Oralkan, A. S. Ergun, M. Karaman, and B. T. Khuri-Yakub. Integration of 2D CMUT arrays with front-end electronics for volumetric ultrasound imaging. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 55(2):327–342, 2008.

[17] M. Rahman and S. Chowdhury. Square diaphragm CMUT capacitance calculation using a new deflection shape function. *Journal of sensors*, 2011, 2011.

[18] I. O. Wygant, M. Kupnik, and B. T. Khuri-Yakub. Analytically calculating membrane displacement and the equivalent circuit model of a circular CMUT cell. In *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, pages 2111–2114. IEEE, 2008.

[19] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary. *Nonlinear finite elements for continua and structures*. John Wiley & sons, 2013.

[20] M. I. Younis. *MEMS linear and nonlinear statics and dynamics*, volume 20. Springer Science & Business Media, 2011.

[21] N. Kacem, A. Jallouli, V. Walter, G. Bourbon, P. Lemoal, and J. Lardies. Nonlinear dynamics of circular capacitive micromachined ultrasonic transducers. In *SENSORS, 2015 IEEE*, pages 1–4. IEEE, 2015.

[22] R. C. Batra, M. Porfiri, and D. Spinello. Review of modeling electrostatically actuated microelectromechanical systems. *Smart Materials and Structures*, 16(6):R23, 2007.

[23] A. Corigliano, M. Dossi, and S. Mariani. Domain decomposition and model order reduction methods applied to the simulation of multi-physics problems in MEMS. *Computers & Structures*, 122:113–127, 2013.

[24] B. Lebental and F. Bourquin. Visco-acoustic modelling of a vibrating plate interacting with water confined in a domain of micrometric size. *Journal of Sound and Vibration*, 331(8):1870–1886, 2012.

[25] M. A. Hamdi, Y. Ousset, and G. Verchery. A displacement method for the analysis of vibrations of coupled fluid-structure systems. *International Journal for Numerical Methods in Engineering*, 13(1):139–150, 1978.

[26] X. Wang and K.-J. Bathe. Displacement/pressure based mixed finite element formulations for acoustic fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 40(11):2001–2017, 1997.

[27] G. C. Everstine. A symmetric potential formulation for fluid-structure interaction. *Journal of Sound and Vibration*, 79(1):157–160, 1981.

[28] L. G. Olson and K.-J. Bathe. Analysis of fluid-structure interactions. a direct symmetric coupled formulation based on the fluid velocity potential. *Computers & Structures*, 21(1):21–32, 1985.

[29] G. Taylor. The coalescence of closely spaced drops when they are at different electric potentials. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 306, pages 423–434. The Royal Society, 1968.

[30] H. C. Nathanson, W. E. Newell, R. A. Wickstrom, and J. R. Davis. The resonant gate transistor. *IEEE Transactions on Electron Devices*, 14(3):117–133, 1967.

[31] J. L. Coulomb. A methodology for the determination of global electromechanical quantities from a finite element analysis and its application to the evaluation of magnetic forces, torques and stiffness. *IEEE Transactions on Magnetics*, 19(6):2514–2519, 1983.

[32] T. Tärnhuvud and K. Reichert. Accuracy problems of force and torque calculation in fe-systems. *Magnetics, IEEE Transactions on*, 24(1):443–446, 1988.

[33] F. Henrotte and K. Hameyer. Computation of electromagnetic force densities: Maxwell stress tensor vs. virtual work principle. *Journal of Computational and Applied Mathematics*, 168(1):235–243, 2004.

[34] G.-H. Kim and K. C. Park. A continuum-based modeling of MEMS devices for estimating their resonant frequencies. *Computer Methods in Applied Mechanics and Engineering*, 198(2):234–244, 2008.

[35] V. Rochus and C. Geuzaine. A primal/dual approach for the accurate evaluation of the electromechanical coupling in MEMS. *Finite Elements in Analysis and Design*, 49(1):19–27, 2012.

[36] K.-K. Park, O. Oralkan, and B. T. Khuri-Yakub. A comparison between conventional and collapse-mode capacitive micromachined ultrasonic transducers in 10-MHz 1-D arrays. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 60(6):1245–1255, 2013.

[37] O. Oralkan, B. Bayram, G. Yaralioglu, A. S. Ergun, M. Kupnik, et al. Experimental characterization of collapse-mode CMUT operation. *ieee transactions on ultrasonics, ferroelectrics, and frequency control*, 53(8):1513–1523, 2006.

[38] G. C. Everstine. Finite element formulatons of structural acoustics problems. *Computers & Structures*, 65(3):307–321, 1997.

[39] S. Marburg and B. Nolte. *Computational acoustics of noise propagation in fluids: finite and boundary element methods*, volume 578. Springer, 2008.

[40] M. Thränhardt, P.-C. Eccardt, H. Mooshofer, and P. Hauptmann. A resonant CMUT-based fluid sensor: Modeling and simulation. *Sensors and Actuators A: Physical*, 156(1):191–195, 2009.

[41] X. Jin, O. Oralkan, F. L. Degertekin, and B. T. Khuri-Yakub. Characterization of one-dimensional capacitive micromachined ultrasonic immersion transducer arrays. *ieee transactions on ultrasonics, ferroelectrics, and frequency control*, 48(3):750–760, 2001.

[42] P.-C. Eccardt, A. Lohfink, and H.-G. Garssen. Analysis of crosstalk between fluid coupled CMUT membranes. In *IEEE Ultrasonics Symposium, 2005.*, volume 1, pages 593–596. IEEE, 2005.

[43] B. T. Khuri-Yakub, G. G. Yaralioglu, B. Bayram, and M. Kupnik. 5f-4 acoustic crosstalk reduction method for CMUT arrays. In *Ultrasonics Symposium, 2006. IEEE*, pages 590–593. IEEE, 2006.

[44] M. Younis, E. M. Abdel-Rahman, and A. Nayfeh. A reduced-order model for electrically actuated microbeam-based MEMS. *Journal of Microelectromechanical systems*, 12(5):672–680, 2003.

[45] N. Sénégond, A. Boulmé, C. Plag, F. Teston, and D. Certon. Fast time-domain modeling of fluid-coupled CMUT cells: from the single cell to the 1-D linear array element. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 60(7):1505–1518, 2013.

[46] S. Pamidighantam, R. Puers, K. Baert, and H. A. C. Tilmans. Pull-in voltage analysis of electrostatically actuated beam structures with fixed–fixed and fixed–free end conditions. *Journal of Micromechanics and Microengineering*, 12(4):458, 2002.

[47] P. M. Zavracky, S. Majumder, and N. E. McGruer. Micromechanical switches fabricated using nickel surface micromachining. *Journal of Microelectromechanical Systems*, 6(1):3–9, 1997.

[48] J. Chen, S.-M. Kang, J. Zou, C. Liu, and J. Schutt-Ainé. Reduced-order modeling of weakly nonlinear MEMS devices with Taylor-series expansion and arnoldi approach. *Journal of Microelectromechanical systems*, 13(3):441–451, 2004.

[49] P.-C. Eccardt, P. Wagner, and S. Hansen. 5f-1 analytical models for micromachined transducers-an overview. In *Ultrasonics Symposium, 2006. IEEE*, pages 572–581. IEEE, 2006.

[50] D. T. Porter. Self-and mutual-radiation impedance and beam patterns for flexural disks in a rigid plane. *The Journal of the Acoustical Society of America*, 36(6):1154–1161, 1964.

[51] K. K. Park, M. Kupnik, H. J. Lee, B. T. Khuri-Yakub, and I. O. Wygant. Modeling and measuring the effects of mutual impedance on multi-cell CMUT configurations. In *Ultrasonics Symposium (IUS), 2010 IEEE*, pages 431–434. IEEE, 2010.

[52] C. Meynier, F. Teston, and D. Certon. A multiscale model for array of capacitive micromachined ultrasonic transducers. *The Journal of the Acoustical Society of America*, 128(5):2549–2561, 2010.

[53] A. Caronti, A. Savoia, G. Caliano, and M. Pappalardo. Acoustic coupling in capacitive microfabricated ultrasonic transducers: modeling and experiments. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 52(12):2220–2234, 2005.

[54] H. Oguz, A. Atalar, and H. Köymen. Equivalent circuit-based analysis of CMUT cell dynamics in arrays. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 60(5):1016–1024, 2013.

[55] A. Lohfink and P.-C. Eccardt. Linear and nonlinear equivalent circuit modeling of CMUTs. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 52(12):2163–2172, 2005.

[56] MEMCAD Manual. Coventor, inc., cary, nc 27513.

[57] H. Koymen, A. Atalar, E. Aydogdu, C. Kocabas, H. Oguz, S. Olcum, A. Ozgurluk, and A. Unlugedik. An improved lumped element nonlinear circuit model for a circular CMUT cell. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 59(8), 2012.

[58] A. Caronti, R. Carotenuto, and M. Pappalardo. Electromechanical coupling factor of capacitive micromachined ultrasonic transducers. *The Journal of the Acoustical Society of America*, 113(1):279–288, 2003.

[59] G. Yaralioglu, S. A. Ergun, and B. T. Khuri-Yakub. Finite-element analysis of capacitive micromachined ultrasonic transducers. *ieee transactions on ultrasonics, ferroelectrics, and frequency control*, 52(12):2185–2198, 2005.

[60] W. Zhang, H. Zhang, F. Du, J. Shi, S. Jin, and Z. Zeng. Pull-in analysis of the flat circular CMUT cell featuring sealed cavity. *Mathematical Problems in Engineering*, 2015, 2015.

[61] B. Bayram, G. Yaralioglu, M. Kupnik, A. Ergun, O. Oralkan, A. Nikoozadeh, and B. T. Khuri-Yakub. Dynamic analysis of capacitive micromachined ultrasonic transducers. *ieee transactions on ultrasonics, ferroelectrics, and frequency control*, 52(12):2270–2275, 2005.

[62] E. Chan, K. Garikipati, and R. Dutton. Characterization of contact electromechanics through capacitance-voltage measurements and simulations. *Journal of microelectromechanical systems*, 8(2):208–217, 1999.

[63] Z. Yao, S. Chen, S. Eshelman, D. Denniston, and C. Goldsmith. Micromachined low-loss microwave switches. *Journal of Microelectromechanical Systems*, 8(2):129–134, 1999.

[64] J. R. Gilbert, R. Legtenberg, and S. D. Senturia. 3d coupled electro-mechanics for MEMS: applications of cosolve-em. 1995.

[65] N. Kacem, S. Baguet, S. Hentz, and R. Dufour. Computational and quasi-analytical models for non-linear vibrations of resonant MEMS and NEMS sensors. *International Journal of Non-Linear Mechanics*, 46(3):532–542, 2011.

[66] A. Bozkurt. Finite element modeling of CMUTs using a perfectly matched layer for fast simulation. In *Ultrasonics, 2003 IEEE Symposium on*, volume 2, pages 1979–1982. IEEE, 2003.

[67] B. Bayram, G. Yaralioglu, A. Ergun, M. Oralkan, and B. T. Khuri-Yakub. Dynamic fem analysis of multiple CMUT cells in immersion [capacitive micromachined ultrasonic transducers]. In *Ultrasonics Symposium, 2004 IEEE*, volume 1, pages 252–255. IEEE, 2004.

[68] S. Ballandras, M. Wilm, W. Daniau, A. Reinhardt, V. Laude, and R. Armati. Periodic finite element/boundary element modeling of capacitive micromachined ultrasonic transducers. *Journal of applied physics*, 97(3):034901, 2005.

[69] C. O'Mahony, M. Hill, R. Duane, and A. Mathewson. Analysis of electromechanical boundary effects on the pull-in of micromachined fixed-fixed beams. *Journal of Micromechanics and Microengineering*, 13(4):S75, 2003.

[70] S. Antman. Nonlinear problems of elasticity, 1995.

[71] P. Naghdi. The theory of shells and plates. In *Linear Theories of Elasticity and Thermoelasticity*, pages 425–640. Springer, 1973.

[72] K.-J. Bathe. *Finite element method.* Wiley Online Library, 2008.

[73] C. Johnson. *Numerical solution of partial differential equations by the finite element method.* Courier Corporation, 2012.

[74] D. Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics.* Cambridge University Press, 2007.

[75] G. Dhatt, E. Lefrançois, and G. Touzot. *Finite element method.* John Wiley & Sons, 2012.

[76] S. C. Brenner and L.-Y. Sung. Linear finite element methods for planar linear elasticity. *Mathematics of Computation*, 59(200):321–338, 1992.

[77] E. Durand and C. Masson. Électrostatique, tome i. *Les Distributions*, 1964.

[78] F. Henrotte. *Modélisation des efforts électromagnétiques et de leurs effets dans les structures quelconques.* PhD thesis, 2000.

[79] F. Henrotte, G. Deliége, and K. Hameyer. The eggshell approach for the computation of electromagnetic forces in 2D and 3D. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, 23(4):996–1005, 2004.

[80] L. Kraehenbuehl, H. Timouyas, M. Moussaoui, and F. Buret. Coins et arrondis en éléments finis-une approche mathématique des coins et arrondis pour les solutions par éléments finis de l'équation de Laplace. *RIGE*, 8:35–45, 2005.

[81] S. H. Schot. Eighty years of sommerfeld's radiation condition. *Historia mathematica*, 19(4):385–401, 1992.

[82] F. V. Atkinson. Lxi. on sommerfeld's radiation condition. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 40(305):645–651, 1949.

[83] M. J. Gander, F. Magoules, and F. Nataf. Optimized schwarz methods without overlap for the helmholtz equation. *SIAM Journal on Scientific Computing*, 24(1):38–60, 2002.

[84] Y. Boubendir, X. Antoine, and C. Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the helmholtz equation. *Journal of Computational Physics*, 231(2):262–280, 2012.

[85] A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An exact bounded perfectly matched layer for time-harmonic scattering problems. *SIAM Journal on Scientific Computing*, 30(1):312–338, 2007.

[86] I. Singer and E. Turkel. A perfectly matched layer for the helmholtz equation in a semi-infinite strip. *Journal of Computational Physics*, 201(2):439–465, 2004.

[87] C. Geuzaine and J.-F. Remacle. Gmsh: A 3d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[88] C. A. Felippa and K. C. Park. Staggered transient analysis procedures for coupled mechanical systems: formulation. *Computer Methods in Applied Mechanics and Engineering*, 24(1):61–111, 1980.

[89] S. David A. Hannot. *Modeling strategies for electro-mechanical Microsystems with uncertainty quantification.* TU Delft, Delft University of Technology, 2010.

[90] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85(3):67–94, 1959.

[91] K-J. Bathe and E. L. Wilson. Numerical methods in finite element analysis. 1976.

[92] S.-Y. Chang. Studies of newmark method for solving nonlinear systems:(i) basic analysis. *Journal of the Chinese institute of engineers*, 27(5):651–662, 2004.

[93] B. Meys. *Modélisation des champs électromagnétiques aux hyperfréquences par la méthode des éléments finis, application au probleme du chauffage diélectrique.* PhD thesis, Ph. D. Thesis, Université de Liege, 1999.

[94] D. M. Copeland and U. Langer. Domain decomposition solvers for nonlinear multiharmonic finite element equations. *Journal of Numerical Mathematics*, 18(3):157–175, 2010.

[95] F. Bachinger, M. Kaltenbacher, and S. Reitzinger. An efficient solution strategy for the HBFE method. *Proceedings of the IGTE*, 2:385–389, 2002.

[96] H. De Gersem, H. Vande Sande, and K. Hameyer. Strong coupled multi-harmonic finite element simulation package. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, 20(2):535–546, 2001.

[97] J. Gyselinck, P. Dular, C. Geuzaine, and W. Legros. Harmonic-balance finite-element modeling of electromagnetic devices: a novel approach. *Magnetics, IEEE Transactions on*, 38(2):521–524, 2002.

[98] G. Paoli, O. Biro, and G. Buchgraber. Complex representation in nonlinear time harmonic eddy current problems. *Magnetics, IEEE Transactions on*, 34(5):2625–2628, 1998.

[99] S. Yamada and K. Bessho. Harmonic field calculation by the combination of finite element analysis and harmonic balance method. *Magnetics, IEEE Transactions on*, 24(6):2588–2590, 1988.

[100] J. Gyselinck, C. Martis, and R. Sabariego. Using dedicated time-domain basis functions for the simulation of pulse-width-modulation controlled devices–application to the steady-state regime of a buck converter. 2013.

[101] B. Cochelin and C. Vergez. A high order purely frequency-based harmonic balance formulation for continuation of periodic solutions. *Journal of sound and vibration*, 324(1):243–262, 2009.

[102] S. Karkar, B. Cochelin, and C. Vergez. A high-order, purely frequency based harmonic balance formulation for continuation of periodic solutions: The case of non-polynomial nonlinearities. *Journal of Sound and Vibration*, 332(4):968–977, 2013.

[103] F. Bachinger, U. Langer, and J. Schöberl. Numerical analysis of nonlinear multiharmonic eddy current problems. *Numerische Mathematik*, 100(4):593–616, 2005.

[104] F. Bachinger, U. Langer, and J. Schöberl. Efficient solvers for nonlinear time-periodic eddy current problems. *Computing and Visualization in Science*, 9(4):197–207, 2006.

[105] G. Li and N. R. Aluru. A lagrangian approach for electrostatic analysis of deformable conductors. *Microelectromechanical Systems, Journal of*, 11(3):245–254, 2002.

[106] M. Frigo and S. G. Johnson. FFTW user manual. *Massachusetts Institute of Technology*, 1999.

[107] H. A. Schwarz. Ueber einige abbildungsaufgaben. *Journal für die reine und angewandte Mathematik*, 70:105–120, 1869.

[108] P.-L. Lions. On the Schwarz alternating method I:. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. SIAM, Philadelphia, PA, 1988.

[109] A. Vion and C. Geuzaine. Double sweep preconditioner for optimized Schwarz methods applied to the Helmholtz problem. *J. Comput. Phys.*, 266:172–174, June 2014.

[110] Y. Boubendir, X. Antoine, and C. Geuzaine. A quasi-optimal non-overlapping domain decomposition algorithm for the Helmholtz equation. *Journal of Computational Physics*, 231(2), 2012.

[111] E. Efstathiou and M. J. Gander. Why restricted additive schwarz converges faster than additive schwarz. *BIT Numerical Mathematics*, 43(5):945–959, 2003.

[112] X.-C. Cai and M. Sarkis. A restricted additive schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21(2):792–797, 1999.

[113] A. St-Cyr, M. J. Gander, and S. J. Thomas. Optimized multiplicative, additive, and restricted additive schwarz preconditioning. *SIAM Journal on Scientific Computing*, 29(6):2402–2425, 2007.

[114] V. Dolean, P. Jolivet, and F. Nataf. *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*, volume 144. SIAM, 2015.

[115] M. J. Gander and L. Halpern. Méthodes de décomposition de domaine. *Encyclopédie électronique pour les ingénieurs*, 2012.

[116] X. C. Cai. Nonlinear overlapping domain decomposition methods. In *Domain Decomposition Methods in Science and Engineering XVIII*, pages 217–224. Springer, 2009.

[117] P. Brune, M. Knepley, B. Smith, and X. Tu. Composing scalable nonlinear algebraic solvers. *Argonne National Laboratory, Preprint ANL/MCS-P2010-0112*, 2013.

[118] F. Confalonieri, A. Corigliano, M. Dossi, and M. Gornati. A domain decomposition technique applied to the solution of the coupled electro-mechanical problem. *International Journal for Numerical Methods in Engineering*, 93(2):137–159, 2013.

[119] X. C. Cai, W. D Gropp, D. E Keyes, and M. D Tidriri. Newton-Krylov-Schwarz methods in cfd. In *Numerical methods for the Navier-Stokes equations*, pages 17–30. Springer, 1994.

[120] P. N. Brown and Y. Saad. Convergence theory of nonlinear Newton-Krylov algorithms. *SIAM Journal on Optimization*, 4(2):297–330, 1994.

[121] S. C. Eisenstat and H. F. Walker. Globally convergent inexact Newton methods. *SIAM Journal on Optimization*, 4(2):393–422, 1994.

[122] A. Halbach, P. Dular, and C. Geuzaine. Comparison of nonlinear domain decomposition schemes for coupled electromechanical problems. *IEEE Transactions on Magnetics*, In press, 2015.

[123] X.-C. Cai and D. E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing*, 24(1):183–200, 2002.

[124] J. O. Skogestad, E. Keilegavlen, and J. M. Nordbotten. Domain decomposition strategies for nonlinear flow problems in porous media. *Journal of Computational Physics*, 234:439–451, 2013.

[125] A. Vion, R. Bélanger-Rioux, L. Demanet, and C. Geuzaine. A DDM double sweep preconditioner for the Helmholtz equation with matrix probing of the DtN map. In *Mathematical and Numerical Aspects of Wave Propagation WAVES 2013*, June 2013.

[126] A. Vion and C. Geuzaine. Parallel double sweep preconditioner for the optimized Schwarz algorithm applied to high frequency Helmholtz and Maxwell equations. In *Submitted to the Proceedings of the 22th International Conference on Domain Decomposition Methods (DD22)*, Lugano, Italy, September 2013.

[127] T. F. Chan and T. P. Mathew. Domain decomposition algorithms. *Acta numerica*, 3:61–143, 1994.

[128] B. Smith, P. Bjorstad, and W. Gropp. *Domain decomposition: parallel multilevel methods for elliptic partial differential equations.* Cambridge university press, 2004.

[129] V. D. Kupradze. *Three-dimensional problems of elasticity and thermoelasticity*, volume 25. Elsevier, 2012.

[130] S. Zaglmayr. High order finite element methods for electromagnetic field computation.

[131] E. Oñate. *Structural analysis with the finite element method. Linear statics: volume 2: beams, plates and shells.* Springer Science & Business Media, 2013.

[132] J. T. Oden. Stability and convergence of underintegrated finite element approximations. 1984.