

**LUÍS HENRIQUE CAMARGO QUIROZ**

**ULTRASOUND TOMOGRAPHY OF THE  
BREAST USING A FINITE ELEMENTS MODEL**

São Paulo  
2023

**LUÍS HENRIQUE CAMARGO QUIROZ**

**ULTRASOUND TOMOGRAPHY OF THE  
BREAST USING A FINITE ELEMENTS MODEL**

**REVISED EDITION**

PhD thesis presented to Escola Politécnica da Universidade de São Paulo in fulfillment of the requisites for the Title of PhD in Sciences.

Concentration Area:

Mechanical Automation and Control Engineering

Advisor:

Prof. Dr. Raúl González Lima

São Paulo  
2023

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Permission is here granted for the total or partial reproduction and dissemination of this work, by any conventional or electronic means, for for study and research purposes, provided the original source is acknowledged.

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 8 de janeiro de 2024

---

autor

---

orientador

Catálogo na publicação

Divisão de Biblioteca

Escola Politécnica da Universidade de São Paulo

Camargo Quiroz, Luís Henrique

Ultrasound Tomography of the Breast Using a Finite Elements Model / L. H. Camargo Quiroz – São Paulo, 2024.

155 p.

Tese (Doutorado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica. Versão corrigida.

1. Tomografia. 2. Tomografia por ultrassom. 3. Elementos finitos. 4. Otimização. 5. Programação Sequencial quadrática. I. Universidade de São Paulo. Escola Politécnica. II. t.

Nome: Camargo Quiroz, Luís Henrique

Título: Ultrasound tomography of the breast using a finite elements model

Tese apresentada à Escola Politécnica da Universidade de São Paulo para a obtenção do título de Doutor em Ciências. Área de concentração: Engenharia de Controle e Automação Mecânica.

Aprovado em: \_\_\_\_\_

Banca Examinadora

---

Prof. Dr. \_\_\_\_\_  
Instituição: \_\_\_\_\_  
Julgamento: \_\_\_\_\_

---

Prof. Dr. \_\_\_\_\_  
Instituição: \_\_\_\_\_  
Julgamento: \_\_\_\_\_

---

Prof. Dr. \_\_\_\_\_  
Instituição: \_\_\_\_\_  
Julgamento: \_\_\_\_\_

---

Prof. Dr. \_\_\_\_\_  
Instituição: \_\_\_\_\_  
Julgamento: \_\_\_\_\_

---

Prof. Dr. \_\_\_\_\_  
Instituição: \_\_\_\_\_  
Julgamento: \_\_\_\_\_

for Ukraine

# ACKNOWLEDGMENTS

A few words of acknowledgment here.

Before officially starting my studies I took two courses as a hearer, did the assignments and at the very end of that year i was victim of a robbery at home. My notebook was stolen, with all my work, and with the external backup drive attached to it, as I was just updating it. Many overdue thanks to the brave policemen from COE – Comando de Operações Especiais (Special Operations Commando), who spotted the burglar fleeing (in my car), pursued and arrested him: corporal MP Sidnei, third Sgt. MP Caetano, and all their patrol colleagues whose names I don't know – there were at least 4 police cars involved. I was unaware that my notebook was also stolen until I received it back at the Police Department, with the external drive still and dangerously hanging by the USB cable!

Thanks to CAPES, who paid my PhD scholarship and a training scholarship for almost two months in Colorado, USA.

Thanks to Timpel S.A. for the desktop computer they put at the lab for my research and for an additional trip to Fort Collins, when I was able to implement big changes to the simulation software and it began to run much faster.

Thanks to my mother, who helped to pay my bills and expenses, they were higher than my humble scholarship!

Thanks to prof. Jennifer Mueller, from Colorado State University (CSU), who received me in her Electrical Impedance Tomography (EIT) laboratory. More than that: when Covid-19 stroke and I had to leave Fort Collins in a hurry – in the last plane! –, it was impossible to bring back all my belongings, and she keeps those things in safety till now. Thanks to my neighbor Brendan DeJonge, who stored the backpack with my photo gear I forgot on the sidewalk in front of our houses. Almost a year later Talles Rattis would bring my camera and one lens to me, but, instead, surprised me with the complete backpack: thank you Talles, very much! Special thanks to my colleague Ely, who went to Fort Collins a few weeks earlier than me, and managed to rent a house from CSU and allowed me to be there with him – this made my stay viable, within my expenses allowance –; it was a great experience, with much work and also fun!

Thanks to my colleague Rafael Mikio Nakanishi, who helped a lot giving me access to his studies and results, therefore I could compare my finite elements simulation with real data and, later, try to implement new techniques to speed up my simulations, solving the problem in the frequency domain. Thanks to Tayran Milá, also my colleague from the laboratory, who told about softwares I used to create a human breast model.

Thanks to my family. Thanks also to all my former teachers, since the first years of school, who always taught me lessons for the life.

Thanks to Julio Cezar Neves, a real Linux guru. You are a monster! In our short talk you pointed me in the right direction, and now I can generate lots of graphics using parallelism emphin the shell. You saved me a lot of time!

Thanks to Tatiane Tamura, who brought me the first and big books I bought to begin the study of ultrasound. I saved a lot on international shipping!

Thanks to my friends. Even if we meet ourselves not as much as we would like, the values and shared ideals help to proceed moving ahead; especially thanks to Valdecir Quarcioni, whose advice in some tough moments helped me to see and think with the light of the Gospels, so I could make better decisions.

To my advisor, Raúl González Lima, thanks for opening me a door. Literally, too! I was trying to know your research group and so I knocked at your door; then you opened and put the seeds for this work. Now it is finished: thank you!

# ABSTRACT

This work presents an algorithm for ultrasound tomography (UST), applied to the human breast. The aim is a way for early and portable screening of tumors. Each tissue is characterized by its acoustic properties.

The acoustic tomograph will have a series of transducers around the perimeter of a water filled vat where the breast is immersed. Each of the transducers will, one by one, excite the system, meanwhile all transducers will capture sound signals. After all the excitation patterns, the set of readings is processed in order to determine the properties of what was put inside the vat.

An experimental setup is not yet ready, therefore the system of transducers, water and breast, is numerically simulated by a finite element model, frequently called “direct problem”.

The tomography algorithm is based on a finite element model (FEM), with a coarser mesh than the one used as reference, and on an optimization process via sequential quadratic programming. The sound speed propagation on each voxel is adjusted in order to match the read signals from the optimization model to that of the reference or physical experiment, when available. Sound dissipation properties are considered in this work, for simplicity, as uniform properties inside the domain of analysis.

Propagation of sound is a dynamic process. The model states are propagated in time via the central difference integration scheme. For each time step the FEM optimization model has to be solved, and this is the most time consuming step of this algorithm: for each change of properties of each voxel, all the transducers have to fire, and all the numerically simulated readings are then compared to the reference readings. The FE (finite element) model is used thousands of times. The breast numerical phantom is a small cylinder of a few centimeters in diameter and in height, the “tumor” is an even smaller cylinder placed inside it, and the ultrasound frequency is low, 25 kHz, therefore the FE elements can be bigger, reducing the number of elements in the model, while concomitantly having enough elements per wavelength, allowing for faster running times with good representation of sound pressure in the media. Eight and latter twelve elements per wavelength are used.

Sound waves traversing a media will reflected at the borders, therefore the region of interest needs to be surrounded by layers of absorbing elements to attenuate echoes; this adds extra size and complexity to the complete model, and penalties to a faster optimization.

Initial coarse images were obtained using Bayes’ Inference and a sampled anatomical atlas, representing mean values from 100,000 drawings with uniform variation in sizes and position of tumor and breast. Therefore, the 3D finite element method is a viable platform to represent the acoustic partial differential equation for the development of Ultrasound Tomography image algorithms.



**Keywords:** Ultrasound. Tomography. Finite element method. Mathematical optimization. Mammography. Sequential quadratic programming.

# RESUMO

Este trabalho apresenta um algoritmo para tomografia por ultrassom aplicada a uma mama, visando uso em equipamento portátil para a triagem precoce de tumores. Cada tecido é caracterizado por suas propriedades acústicas.

Um tomógrafo para esta aplicação terá uma série de transdutores ao redor de uma cuba com água, na qual a mama será imersa. Cada um dos transdutores emite uma excitação, enquanto os demais capturam as respostas. Ao fim dos padrões de excitação as leituras são processadas para gerar a tomografia.

Um protótipo de tal equipamento não está pronto, portanto o conjunto dos transdutores, água e mama será representado numericamente por um modelo de elementos finitos, comumente chamado de “problema direto”.

O algoritmo de tomografia é baseado num modelo de elementos finitos com uma malha mais grosseira que a de simulação, e em otimização por programação quadrática sequencial. A velocidade de propagação do som em cada voxel será ajustada (otimizada) para fazer corresponder as leituras do modelo de otimização com as do modelo de simulação ou experimento físico, quando disponível. Para simplificar o problema, as propriedades de dissipação do som nos diferentes elementos dentro do domínio de análise são consideradas uniformes e iguais.

A propagação de onda é um fenômeno dinâmico. Os estados dos modelos são propagados usando uma integração por diferenças centrais. Para cada passo no tempo o modelo de otimização deve ser resolvido, e é este processo de simulação por elementos finitos o que mais exige tempo de processamento. Para cada variação no valor de uma propriedade, em cada voxel, cada transdutor emite a excitação, as leituras são numericamente simuladas integrando os passos no tempo com a malha de otimização, e estas leituras comparadas com as de referência. O modelo de elementos finitos será usado milhares de vezes. O fantoma da mama é um cilindro de poucos centímetros de diâmetro e de altura, e o tumor é um cilindro ainda menor em seu interior. O ultrassom é de baixa frequência, 25 kHz, assim os elementos finitos são maiores, reduzindo o número de elementos no modelo mas preservando uma adequada relação de elementos por comprimento de onda. Assim têm-se um modelo rápido com boa representação dos fenômenos acústicos. São usados 8 e depois 12 elementos por comprimento de onda.

Ondas sonoras atravessando um meio qualquer são refletidas com a máxima intensidade nas suas fronteiras externas, e por isso a região de interesse deve ser revestida com elementos extras, cujas propriedades atenuem os ecos. Isto acrescenta complexidade e número de elementos ao modelo completo, comprometendo a velocidade de simulação e de otimização.

Imagens iniciais de baixa resolução foram geradas com inferência bayesiana e um atlas anatômico elaborado com amostras da população, com 100000 sorteios supondo distribuição uniforme nas variações de tamanho e posição de mama e tumor. Portanto, um modelo 3D de elementos finitos é uma plataforma viável para representar as equações de derivadas parciais acústicas, e isto permite o desenvolvimento de algoritmos para geração

de imagens de tomografia por ultrassom.

**Palavras-chave:** Ultrassom. Tomografia. Método dos elementos finitos. Otimização matemática. Mamografia. Programação sequencial quadrática.

# LIST OF FIGURES

|  | Page |
|--|------|
| Figure 1 – Mass element and mechanical stresses of a plane wave in the $z$ direction                 | 26   |
| Figure 2 – Mass flow in an elemental fluid volume . . . . .  | 29   |
| Figure 3 – Conservation of momentum for the $x$ axis in an elemental fluid volume .                  | 30   |
| Figure 4 – Sample of a result from the simulation program . . . . .                                  | 41   |
| Figure 5 – Some of the first finite element models used with the simulation program                  | 43   |
| Figure 6 – 3D mesh for one of the first finite element models . . . . .                              | 44   |
| Figure 7 – The first models had echoes at borders; they had no ALID layer . . . . .                  | 44   |
| Figure 8 – Angular position $\theta_\ell$ of a transducer “ $\ell$ ” for the D-bar problem . . . . . | 48   |
| Figure 9 – Dimensions of the thorax model and of the transducers’ ellipse . . . . .                  | 49   |
| Figure 10 – Comparison of the original and the simplified contours . . . . .                         | 49   |
| Figure 11 – Some equipments for breast tomography and for breast measurements .                      | 51   |
| Figure 12 – Geometry for the simplified breast model (reference and tomography) .                    | 54   |
| Figure 13 – Composition of an ALID layer “ $N_1/N_2/N_3$ ” of $N_1$ sub-layers of $N_2$ elements     | 63   |
| Figure 14 – Visual identification of echo at the model borders . . . . .                             | 68   |
| Figure 15 – Temporal evolution of an excitation pulse generating echoes . . . . .                    | 69   |
| Figure 16 – Temporal evolution of the excitation pulse with an ALID layer 3/3/3 . .                  | 70   |
| Figure 17 – Behavior of the echo model, ALID 6/2/4 . . . . .   | 71   |
| Figure 18 – Temporal evolution of the excitation pulse with an ALID layer 6/2/4 . .                  | 72   |
| Figure 19 – Excitation and reading with initial pulse and echo signal . . . . .                      | 73   |
| Figure 20 – Readings with echo attenuated by the use of an ALID wrapping . . . . .                   | 74   |
| Figure 21 – Tumor for the realistic model, described by a STL file . . . . .                         | 75   |
| Figure 22 – Big variation in the size of surface triangles in one of the realistic breasts           | 77   |

|   |     |
|---|-----|
| Figure 23 – A more uniform size of surface triangles in another of the realistic breasts                  | 77  |
| Figure 24 – Wireframe depicting cuba_2020 breast in the vat, viewed from below                            | 78  |
| Figure 25 – Structured meshes for comparisons with experimental data                                      | 80  |
| Figure 26 – Pressures inside a sphere with Nakanishi’s simulator  | 80  |
| Figure 27 – Pressures in a sphere with the simulator from the tomography algorithm                        | 81  |
| Figure 28 – Excitation pulse with sharp transition from steady state                                      | 82  |
| Figure 29 – Sample of a “rings and sectors” grid of voxels for a 16x16 grid with 4 rings                  | 83  |
| Figure 30 – Voxel numbers in the same “rings and sectors” 16x16 grid with 4 rings                         | 83  |
| Figure 31 – Finite element mesh used as reference with water, breast and tumor visible                    | 85  |
| Figure 32 – Tomography with tumor in the fourth quadrant  | 86  |
| Figure 33 – Comparison of optimizers: SLSQP <i>versus</i> Simplex   | 87  |
| Figure 34 – Effectiveness of the simple objective function used   | 87  |
| Figure 35 – Mean values from 100,000 random elements on a 14 x 14 voxel grid                              | 88  |
| Figure 36 – Tomography with 24 transducers and tumor anywhere in half of the breast                       | 89  |
| Figure 37 – Mesh of the direct problem after last optimization, including the ALID                        | 90  |
| Figure 38 – Improvement in a tomography going from 6 to 24 transducers                                    | 90  |
| Figure 39 – Plot of the “time sensitivity” $t_{V_1} - t_{V_{1\Delta}}$ for various voxel grid resolutions | 95  |
| Figure 40 – Echoes in the cut planes of the thorax section in a model without ALID                        | 118 |
| Figure 41 – Principles of the finite element modeling   | 129 |
| Figure 42 – Time grid in the central difference method  | 138 |
| Figure 43 – Comparison of regular and tumor cells   | 152 |
| Figure 44 – Metastatic pancreatic cells (pale) in a liver   | 152 |
| Figure 45 – Anatomy of a human female breast  | 154 |
| Figure 46 – Lymphatic system in the breast region   | 155 |

# LIST OF TABLES

|  | <b>Page</b> |
|--|-------------|
| Table 1 – Sound speeds in the thorax model components. Longitudinal waves . . .                                | 50          |
| Table 2 – Sound speeds in the simplified breast model. Longitudinal waves . . . .                              | 52          |
| Table 3 – Number of elements and nodes in the simplified breast model . . . . .                                | 53          |
| Table 4 – Sizes of the thorax simulation models with different ALID layers . . . .                             | 64          |
| Table 5 – Execution times varying mesh size with serial and parallelized programs                              | 66          |
| Table 6 – Properties of some of the realistic breast meshes . . . . .  | 78          |
| Table 7 – Influence of the grid type in the number of optimization parameters . .                              | 84          |
| Table 8 – High contrast speeds in the simplified breast model. Longitudinal waves                              | 86          |
| Table 9 – “Time sensitivity” $t_{V_1} - t_{V_{1\Delta}}$ for various voxel grid resolutions $N \times N$ . . . | 94          |
| Table 10 – Execution times: mesh with 1,061,828 tetrahedrons, 8 threads computer                               | 116         |
| Table 11 – Execution times: mesh with 1,061,828 tetrahedrons, 12 threads computer                              | 116         |
| Table 12 – Execution times: mesh with 1,061,828 tetrahedrons, 48 threads computer                              | 117         |

# LIST OF SYMBOLS

|                |  |
|----------------|--|
| $\nabla$       | gradient operator  |
| $\vec{\nabla}$ | divergence operator  |
| $\bar{A}$      | reciprocal of the imposed acoustic impedance                                       |
| $a_i$          | coefficient at node $i$ of the finite element                                      |
| $[B]$          | gradient of the global shape functions   |
| $c$            | sound speed  |
| $[C]$          | acoustic damping matrix  |
| $C_a(e)$       | attenuation coefficient of the finite element $e$ in the ALID layer                |
| $C_{a_0}$      | reference attenuation coefficient (water)  |
| $C_{i,j}$      | element at line $i$ and column $j$ of the acoustic damping matrix                  |
| $c_\ell$       | longitudinal wave propagation speed  |
| $d(e)$         | distance from the center of element $e$ to the beginning of the ALID layer         |
| $f$            | frequency (Hz)   |
| $\{F_{ai}\}$   | “acoustic force” vector  |
| $f_{vi}^e$     | number of faces of element $e$ which contains node $i$                             |
| $i$            | imaginary number ( $\sqrt{-1}$ )   |
| $k$            | wave number  |
| $[K]$          | acoustic stiffness matrix  |
| $K_{i,j}$      | element at line $i$ and column $j$ in the acoustic stiffness matrix                |
| $K_{i,j}^e$    | element at line $i$ and column $j$ of the stiffness matrix of a finite element $e$ |
| $L$            | transducers’ total   |
| $[M]$          | acoustic mass matrix   |
| $M_{i,j}$      | element at line $i$ and column $j$ of the acoustic mass matrix                     |
| $m_{i,j}$      | number of elements which contains the nodes $i$ and $j$                            |
| $M_{Tr}$       | maximal excitation amplitude of the transducers’ set                               |

|                   |   |
|-------------------|---|
| $m_{vi}$          | number of elements which contain the node $i$ on a given face                                     |
| $m_{Z_{i,j}}$     | number of faces of an element, containing nodes $i$ and $j$ , subjected to an impedance condition |
| $[N]$             | vector of global shape functions  |
| $\vec{n}$         | unity vector normal to a surface, at some point   |
| $n_e$             | nodes of a finite element   |
| $n_f$             | total of nodes in the discretized domain  |
| $N_i$             | global shape function at node $i$   |
| $N_i^e$           | shape function $i$ of finite element $e$  |
| $n_p$             | number of shape functions   |
| $[N_{ve}^f]$      | shape functions matrix of the nodes at face $f$ of element $e$                                    |
| $\bar{p}$         | imposed pressure  |
| $P$               | power in the increment of the ALID attenuation  |
| $p$               | acoustic pressure   |
| $\hat{p}$         | aproximate expansion of $p$   |
| $\tilde{p}$       | acoustic pressure weighting   |
| $P_i$             | element at line $i$ of the imposed pressure vector  |
| $\{\hat{p}_i\}$   | nodal pressure vector   |
| $\{P_i\}$         | imposed pressure vector   |
| $\{\hat{p}_i^e\}$ | pressure vector at node $i$ of element $e$  |
| $q$               | acoustic excitation source  |
| $\{Q_i\}$         | acoustic source vector  |
| $\{\bar{q}_i\}$   | point acoustic source at node $i$   |
| $T$               | mechanical stress; operator of a forward problem  |
| $[\cdot]^T$       | transpose of vector or matrix   |
| $t$               | time  |
| $T_{sl}$          | thickness of an ALID sublayer   |
| $T_X^n(\ell)$     | amplitude of the signal emitted by transducer $\ell$ in the $n$ excitation pattern                |
| $u$               | particle displacement in $z$ direction, relative to the equilibrium position                      |
| $V$               | wave propagation domain   |
| $V_e$             | domain of a finite element  |



|                   |  |
|-------------------|--|
| $\{\bar{v}_e^f\}$ | velocity components vector at the nodes of face $f$ of element $e$   |
| $\bar{v}_n$       | imposed normal speed   |
| $V_{ni}$          | line $i$ of the vector of imposed normal velocities  |
| $\{V_{ni}\}$      | vector of imposed normal velocities  |
| $\bar{Z}$         | imposed acoustic impedance   |
| $z$               | $z$ axis of the cartesian coordinate system; value of $z$ coordinate                                       |
| $\Omega$          | domain external surface  |
| $\Omega_e^f$      | set of the faces of element $e$  |
| $\Omega_p$        | surface subjected to a pressure boundary condition   |
| $\Omega_v$        | surface subjected to a velocity boundary condition   |
| $\Omega_Z$        | surface subjected to an acoustic impedance boundary condition  |
| $\delta$          | Dirac delta function (at node $i$ )  |
| $\theta_\ell$     | angular position of transducer $\ell$ , relative to transducer 1 and to the center of the transducers' set |
| $\lambda$         | first Lamé parameter (shear modulus) <i>or</i> wavelength  |
| $\mu$             | second Lamé parameter  |
| $\rho$            | material's density   |
| $\omega$          | angular frequency  |
| $\phi(\cdot)$     | value, corresponding to $\phi_m$ , obtained with the voxel values of the tomography                        |
| $\vec{\phi}$      | vector field of $\phi$   |
| $\phi_m$          | measured values (X-ray intensity, sound pressure signal, etc) used for the tomography                      |

# CONTENTS

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>21</b> |
| 1.1      | Objectives . . . . .  | 22        |
| <b>2</b> | <b>Revision – basic concepts</b>                                  | <b>24</b> |
| 2.1      | Acoustic waves – equations and finite elements modeling . . . . . | 24        |
| 2.1.1    | Plane wave equation . . . . .                                     | 25        |
| 2.1.2    | Acoustic dissipation . . . . .                                    | 27        |
| 2.1.2.1  | Dissipation with an imaginary wavenumber . . . . .                | 27        |
| 2.1.2.2  | Some remarks on acoustical dissipation . . . . .                  | 28        |
| 2.1.3    | Acoustic PDE in 3 dimensions . . . . .                            | 28        |
| 2.1.4    | Definition of the problem in 3 dimensions . . . . .               | 32        |
| 2.1.5    | Finite elements for acoustics . . . . .                           | 33        |
| 2.1.5.1  | Finite elements with viscous losses . . . . .                     | 34        |
| 2.2      | The central difference method . . . . .                           | 35        |
| 2.3      | Program parallelization . . . . .                                 | 35        |
| 2.4      | Ultrasound tomography in modern practice . . . . .                | 36        |
| <b>3</b> | <b>Method</b>   | <b>38</b> |
| 3.1      | Direct problem . . . . .  | 39        |
| 3.1.1    | General description of the simulation program . . . . .           | 39        |
| 3.1.2    | Boundary conditions: the penalty method . . . . .                 | 43        |
| 3.1.3    | Echo absorbing layer . . . . .                                    | 46        |
| 3.1.4    | Mesh development . . . . .  | 47        |
| 3.1.4.1  | Meshes for ultrasound inside the thorax . . . . .                 | 47        |

|          |  |           |
|----------|--|-----------|
| 3.1.4.2  | Meshes for the breast with a tumor . . . . .                     | 50        |
|          | Realistic human breast: . . . . .                                | 51        |
|          | Simplified human breast: . . . . .                               | 51        |
| 3.1.5    | Parallel computing . . . . .                                     | 55        |
| 3.1.5.1  | Parallelism using OpenMP . . . . .                               | 55        |
| 3.2      | Inverse problem . . . . .  | 55        |
| 3.2.1    | Initial guess . . . . .  | 56        |
| 3.2.1.1  | How voxels and finite elements are correlated . . . . .          | 56        |
| 3.2.2    | Regularization: sample based prior . . . . .                     | 56        |
| 3.2.3    | Search for the maximum <i>a posteriori</i> probability . . . . . | 59        |
| 3.2.3.1  | Selection of algorithms . . . . .                                | 60        |
|          | Optimization problem statement. . . . .                          | 60        |
| <b>4</b> | <b>Results</b>   | <b>62</b> |
| 4.1      | Development of an ALID model . . . . .                           | 62        |
| 4.1.1    | ALID equation . . . . .  | 62        |
| 4.1.2    | ALID sizes . . . . .   | 63        |
| 4.2      | Use of parallelism in the program . . . . .                      | 64        |
| 4.2.1    | Influence of the number of threads . . . . .                     | 64        |
| 4.2.2    | Parallelization for very small problems? . . . . .               | 65        |
| 4.2.3    | Parallelism using the GPU . . . . .                              | 66        |
| 4.3      | Simulation results for the thorax model . . . . .                | 67        |
| 4.3.1    | Effectiveness of the ALID layer . . . . .                        | 73        |
| 4.4      | Realistic breast . . . . .                                       | 75        |
| 4.5      | Simulations and experimental data . . . . .                      | 79        |
| 4.5.1    | Numerical comparison . . . . .                                   | 79        |
| 4.6      | Tomography grids for faster results . . . . .                    | 82        |

|  |            |
|--|------------|
| 4.7 Tomography results . . . . .   | 85         |
| <b>5 Analysis and discussion</b>   | <b>91</b>  |
| <b>6 Conclusions and continuity of the work</b>                                | <b>97</b>  |
| 6.1 Conclusions . . . . .  | 97         |
| 6.2 Continuity of the work . . . . .   | 99         |
| <b>Bibliography</b>  | <b>102</b> |
| <b>Appendix A – Acoustic finite elements with dissipation</b>                  | <b>112</b> |
| <b>Appendix B – Execution times: OpenMP <i>versus</i> number of threads</b>    | <b>115</b> |
| <b>Appendix C – Echoes in the upper and lower surfaces of the thorax slice</b> | <b>118</b> |
| <b>Annex A – Ultrasound in medicine: the beginning</b>                         | <b>120</b> |
| <b>Annex B – Finite Element formulation for sound waves</b>                    | <b>125</b> |
| B.1 Basic equation and boundary conditions . . . . .                           | 125        |
| B.2 Weighted residual formulation of the Helmholtz equation . . . . .          | 126        |
| B.3 Approximation of the field variables . . . . .                             | 128        |
| B.4 Uncoupled finite element model for acoustic problems . . . . .             | 129        |
| B.4.1 Acoustical stiffness matrix . . . . .                                    | 130        |
| B.4.2 Acoustical mass matrix . . . . .   | 132        |
| B.4.3 Acoustical excitation vectors . . . . .                                  | 132        |
| B.4.4 Acoustical damping matrix . . . . .                                      | 134        |
| B.4.5 Acoustical model by finite elements . . . . .                            | 135        |
| <b>Annex C – The central difference method</b>                                 | <b>137</b> |
| <b>Annex D – Mathematical details about SLSQP, TOMP’s optimizer</b>            | <b>139</b> |

|  |  |            |
|--|--|------------|
| D.1  | Sequential quadratic programming               | 139        |
| D.1.1                                      | Interactive optimization                       | 140        |
| D.1.2                                      | Search direction                               | 140        |
| D.1.3                                      | Step size                                      | 141        |
| D.1.4                                      | Update of the Matrix B                         | 142        |
| D.1.5                                      | Least Squares Subproblems                      | 143        |
| D.1.6                                      | Motivation of SQP Methods                      | 143        |
| D.1.6.1                                    | Optimality Conditions.                         | 143        |
| D.1.6.2                                    | Nonlinear Equations and Quadratic Programming. | 145        |
| D.1.7                                      | Active Set Strategy                            | 145        |
| D.2  | Quadratic programming algorithm                | 147        |
| D.2.1                                      | Problem Equivalence                            | 147        |
| D.2.2                                      | Problem Transformation                         | 148        |
| <b>Annex E – Facts about breast cancer</b> |  | <b>151</b> |
| E.1  | What is cancer                                 | 151        |
| E.2  | The human breast                               | 153        |
| E.2.1                                      | Types of breast carcinoma                      | 153        |

# 1 INTRODUCTION

The generation of a medical image by tomography is a problem in the mathematical category of *inverse problems*, those where, by means of an adequate set of measurements, one can assess, for a known set of excitations, what is the distribution of physical properties inside a subject which, with said excitations, best approximates the taken measurements.

Many groups study to date the subject of ultrasound tomography, for breast examination ([1, 2, 3]) and in many other fields, medical or not ([4, 5]), besides the characterization of ultrasound properties in the concerned tissues or media ([6, 7, 8]).

Ultrasound is already established as an valuable mean for many medical diagnostic equipments. It is safe, equipments are small and even portable, and not too expensive. *An ultrasound tomography (UST) of the breast may screen early tumors, before calcification<sup>1</sup>, therefore before they begin to show up in conventional mammography exams by means of X-rays.* And it is well known that early treatments are of utmost relevance for a successful outcome. Breast cancer is, still today, one of the biggest health problems for women, and one of the main causes of deaths that could be avoided. An UST will distinguish tissues by their sound properties, allowing to distinguish between tumor and health tissue. Note that a conventional US exam can give an contour image, however based purely on echoes.

The ultrasound tomography software uses data representative of the real problem: the complete sound signal captured by a transducer. The data is obtained by measurements, or else simulated by means of a *direct problem*, with an adequate and precise model in terms of its formulation, equations and parameters. With the desired excitation, it gives the surrogate measures that would be given by the real system. The measurements, simulated or real, are the inputs for the *inverse model*, where the parameters are adjusted until the measurements obtained by the inverse model match the measurements obtained by the direct model. When this match succeeds, in a certain sense, the properties correspond to the maximum *a posteriori* probability conditioned to the measurements of pressure, and will be considered the best distribution of sound speed in the domain. The direct

---

<sup>1</sup>See annex E.

problem is a finite element method (FEM) model.

In this work are presented the basics of sound propagation and its equations, a review of the finite element method (applied to sound problems), the sequential quadratic programming used to solve the optimization problem (generate a tomography), the program developed, results and conclusions about this process.

It is worth to emphasize that ultrasound is absolutely safe for use with the human body. This enables, if necessary, to submit someone to a great number of US readings to elaborate a high-definition tomography; on the contrary, X-rays and computed tomography pose a risk because of the ionizing radiations employed.

Next chapters will cover:

- a review of equations for acoustic waves, the modeling of the problem via the finite elements method and the ultrasound tomography,
- creation of the finite element (FE) meshes for simulation (reference or “real problem”) and for optimization (generation of the tomography),
- optimization using sequential quadratic programming, and how it is used in the tomography optimization,
- results obtained,
- analysis, discussion, and
- conclusions.

## 1.1 Objectives

The main goal is to verify the feasibility of an ultrasound tomography algorithm using the Finite Element Method (FEM) models. If this kind of tomography reveals viable, it will allow a broad range of subjects to undergo UST, as the FE method is extremely powerful and with it geometries of any complexity can be modeled, and many less trivial physical phenomena can be included in the problem: sound interaction with structures (outside of the human body) in a device or experimental set-up; viscous, relaxation, multiple relaxation viscoelastic losses, Voigt or Maxwell wave equations (see [9, chap. 4]), thermal effects, realistic transducer to skin modeling with gel layers, etc.

Additional objectives are:

- to develop appropriate FEM meshes for the refined model and the inversion model;
- to develop and numerically test a regularization procedure;

*Note: figures and tables without reference to other sources were created by the author.*



## 2 REVISION – BASIC CONCEPTS

In this chapter some basic concepts which allow the development of the simulation and ultrasound tomography algorithm are presented. Equations for sound propagation in one dimension and in three dimensions are developed, and the strategy of parallel computations, OpenMP are explained.

The partial differential equation (PDE) of the acoustic wave relates sound pressure in time and space and propagation speed of the wave in tissues. The ultrasound tomography aims to estimate the distribution of sound speeds in a domain of interest. Sound speed is the principal property of clinical interest, because it is associated with the tissue: bones, fat, healthy tissue, tumors, each one has its own sound speed. Based on the acoustic PDE equation, the Finite Elements Method will be applied to model the breast and define the direct problem, including the boundary conditions.

Ultrasound Tomography may be clinically relevant if the estimation of an image happens within hours. In order to speed up the computer calculations, one strategy is to use parallelization techniques . OpenMP is a relatively recent technology for parallelization that has an easy and effective way to use the multiple cores modern CPUs have, even in today's personal computers and notebooks.

### 2.1 Acoustic waves – equations and finite elements modeling

The problem must have its domain properly described by physical equations that adequately represent all the phenomena of interest. The propagation of sound is governed by partial differential equations (PDEs), and analytical solution of PDEs are possible only for simple geometries, included simple configurations for the forcing terms, for instance, in the center of the geometry. For more general situations a numerical solution must be sought using finite elements.

### 2.1.1 Plane wave equation

The grounds for the most basic equations of sound wave propagation in solids, that is, plane waves, can be seen e. g. in [10, chap. 1.7], [11, chap. 5.1.3], [12]. For the finite element method, the equations in the actual program are from a more basic and general nature, based mainly in [13], [14] and [15]. In Annex B there is a brief introduction to the FE method and the development of equations of finite elements for sound waves, based on material from a seminar given by Desmet and Vandepitte [16].

A sound wave is a mechanical vibration that propagates through some material medium, causing displacements of particles<sup>1</sup> and stresses due to the elasticity of this medium. In what follows it is assumed no heat transfer and very small displacements. Therefore, linear models are valid.

Let us develop the equation for the plane longitudinal wave, where all the particles of an homogeneous and isotropic material have the same displacement in time in a given geometric plane of the continuum and tridimensional medium.

Aligning the  $z$  axis of the coordinate system with the propagation direction, that is, perpendicular to the wave plane, and taking an elementary volume of the medium, shown in figure 1, where  $u$  is the particle displacement,  $\rho$  the medium density and  $T$  the mechanical stress due to the displacement, applying the second Newton law we get:

$$\frac{\partial T}{\partial z} = \rho \frac{\partial^2 u}{\partial t^2} , \quad (2.1)$$

and, by the elasticity properties of the medium, stress is related to the strain that the particle displacement causes,

$$T = (\lambda + 2\mu) \frac{\partial u}{\partial z} , \quad (2.2)$$

where  $\lambda$  and  $\mu$  are respectively the first and the second Lamé parameters from the constitutive equations of the material.

Substituting (2.2) in (2.1) we get the *equation of a longitudinal plane wave*,

$$\frac{\partial^2 u}{\partial z^2} = \frac{1}{c_\ell^2} \frac{\partial^2 u}{\partial t^2} , \quad (2.3)$$

where the propagation velocity  $c_\ell$  of a longitudinal sound wave is

$$c_\ell = \sqrt{\frac{\lambda + 2\mu}{\rho}} . \quad (2.4)$$

---

<sup>1</sup>Or small amounts of a continuous medium.

**Figure 1** – Mass element and mechanical stresses of a plane wave in the  $z$  direction.

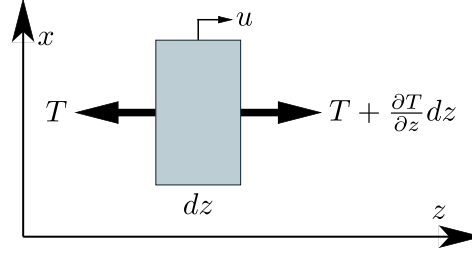


Figure modified from ([17], class 1)

Harmonic solutions for (2.3) can be obtained by separation of variables, with a solution  $u$  function of position  $z$  and time  $t$  such as

$$u(z, t) = u(z) \cdot e^{i\omega t} , \quad (2.5)$$

where  $\omega$  is the angular frequency of the phenomenon and  $i = \sqrt{-1}$  is the imaginary number.

Substituting (2.5) in the wave equation (2.3) we can write

$$\frac{d^2 u(z)}{dz^2} e^{i\omega t} = \frac{1}{c^2} (-\omega^2 u(z) e^{i\omega t}) \implies \boxed{\frac{d^2 u}{dz^2} + k^2 u = 0} \quad (2.6)$$

that is the *Helmholtz equation*. The parameter  $k = \omega/c = 2\pi/\lambda$  is the *wave number*, where  $\lambda$  is the wavelength.

One solution for this equation is the exponential  $u(z) = e^{r \cdot z}$ , which, applied to equation 2.6, allows us to write the following equation and its roots,

$$r^2 e^{r \cdot z} + k^2 e^{r \cdot z} = 0 \implies r^2 + k^2 = 0 \implies r = \pm ik . \quad (2.7)$$

One solution in the frequency domain is

$$u(z) = A e^{-ik \cdot z} + B e^{ik \cdot z} \quad (2.8)$$

and in the time domain it is

$$\begin{aligned} u(z, t) &= A e^{-ik \cdot z} e^{i\omega t} + B e^{ik \cdot z} e^{i\omega t} \\ &= A e^{-ik(z-ct)} + B e^{ik(z+ct)} \end{aligned} \quad (2.9)$$

The term  $Ae^{-ik(z-ct)}$  is a wave going in the positive  $z$  direction and  $Be^{ik(z+ct)}$  a wave in the opposite direction.

## 2.1.2 Acoustic dissipation

When an ultrasound propagates in the human body it loses energy as it travels. Many types of losses exist, being the more important in this case the effects of viscosity and effects of relaxation.

Dissipation can be included in the acoustic equations. This is demonstrated here for viscosity of the media.

### 2.1.2.1 Dissipation with an imaginary wavenumber

Ultrasound energy may be dissipated when the waves reach a boundary (an abrupt change in acoustic impedance) by effects of scattering and absorption. In real fluids it also occurs due to internal losses, e. g. by viscosity.

A general procedure<sup>2</sup> for development of wave equations with dissipation is to include the effect of this dissipation in the conservation equations, which are combined to obtain a wave equation, and this wave equation is then solved by assuming a time-harmonic solution:

$$u = u_0 e^{i(\omega t - k'x)} . \quad (2.10)$$

Doing this the wave equation is transformed into a *dispersion relation*, that is, an algebraic equation in the wave number  $k'$ , now with real and imaginary parts:

$$k' = \beta - i\alpha . \quad (2.11)$$

Substituting equation 2.11 in equation 2.10 yields

$$u = u_0 e^{-\alpha x} e^{i(\omega t - \beta x)} = u_0 e^{-\alpha x} e^{i\omega(t - \frac{x}{\omega/\beta})} , \quad (2.12)$$

and this evidences that  $\alpha$  is the absorption coefficient and  $\beta$  is related to the phase speed by  $c_{ps} = \omega/\beta$ .

---

<sup>2</sup>See e. g. [18].

### 2.1.2.2 Some remarks on acoustical dissipation

For simplicity the development above deals with a simple plane wave. A more general analysis of dissipation entails the solution of the Navier-Stokes equations for a compressible fluid (according to [19]) therefore here only one-dimensional waves in the frequency domain are examined. This reduces the problem to an ordinary differential equation. Another simplification comes from the fact that dissipation effects are relatively weak for most fluids of everyday interest. This property simplifies accounting for the combined effects of the various dissipation mechanisms.

The book of Pierce ([10]) details the complete derivation of the field equation and extends the concepts to more general types of signals and dissipations.

The parameter  $\beta$  is the *propagation wavenumber* and corresponds to a wavelength  $\lambda_\beta = 2\pi/\beta$ . The wave is rigorously not periodic anymore in space because of attenuation, but  $\pi/\beta$  is the distance between zeros.

When  $\alpha$  is nonzero, the pressure decays exponentially. After  $n$  wavelengths the amplitude reduces by a factor of  $e^{(-2\pi n\alpha/\beta)}$ . If  $n > (2\beta)/(\pi\alpha)$  this factor is less than  $e^{-4} \approx 0.0183$  for one wavelength. Attenuation is usually weak, in which case  $\alpha \ll \beta^3$ .

The absorption coefficient  $\alpha$  is frequency dependent and proportional to the square of the sound frequency ( $f^2$ ) for dissipation by effects of viscosity or heat conduction, proportional to  $f^2/(f^2 + f_r^2)$  for dissipation by relaxation (with  $f_r$  the “relaxation frequency”, depends on the composition of material the sound traverses), and proportional to  $\sqrt{f}$  for dissipation by boundary layer effects (see [18, chap. 9]).

The derivations are done here with plane waves, however the expressions for  $\alpha$  and  $c_{ps}$  are also valid for spherical and other unidirectional waves.

### 2.1.3 Acoustic PDE in 3 dimensions

Now let us develop the acoustic equation in three dimensions. Assume a fluid, where no shear forces exist, and the waves are longitudinal; this means that the particle movements are in the same direction the waves travel in space. There is an initial ambient state  $s_0$  (where  $s$  is a placeholder for the state variables) and small perturbations to the state,  $s'$ . The inviscid medium is fully described by three state variables that are functions of

---

<sup>3</sup>However  $\alpha$  and  $\beta$  are not independent, they must obey to the Kramers-Kronig relations to ensure the principles of causality and linearity.

the position in space and functions of time: pressure  $p$ , mass density  $\rho$  and velocity  $v$ :

$$p(x, y, z, t) = p_0(x, y, z, t) + p'(x, y, z, t) , \quad (2.13a)$$

$$\rho(x, y, z, t) = \rho_0(x, y, z, t) + \rho'(x, y, z, t) , \quad (2.13b)$$

$$\vec{v}(x, y, z, t) = \vec{v}_0(x, y, z, t) + \vec{v}'(x, y, z, t) . \quad (2.13c)$$

Being the fluid homogeneous, and initially at rest, the initial states can be written as

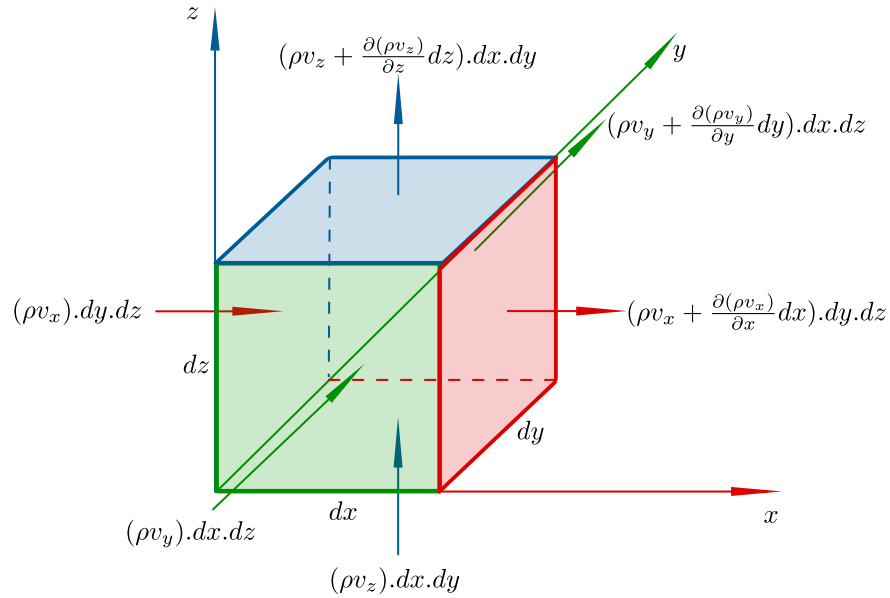
$$p_0(x, y, z, t) \equiv p_0 , \quad (2.14a)$$

$$\rho_0(x, y, z, t) \equiv \rho_0 , \quad (2.14b)$$

$$\vec{v}_0(x, y, z, t) \equiv \vec{0} . \quad (2.14c)$$

The acoustic equation is developed applying the law of mass conservation, momentum conservation and the constitutive pressure-density relation to an infinitesimal volume  $dx.dy.dz$  of the fluid. For the sake of simplicity, no acoustical sources are assumed in the elemental volume.

**Figure 2** – Mass flow in an elemental fluid volume.



The conservation of mass implies that the net accumulation of mass in the element equals the balance of mass that flows through the element faces. The mass flows are

shown in figure 2, and the mass conservation equation in the  $x$  axis is

$$\begin{aligned} \frac{\partial \rho_x}{\partial t} dx \cdot dy \cdot dz &= (\rho v_x) dy \cdot dz - \left( \rho v_x + \frac{\partial(\rho v_x)}{\partial x} dx \right) dy \cdot dz \\ &= - \frac{\partial \rho v_x}{\partial x} dx \cdot dy \cdot dz , \end{aligned} \quad (2.15)$$

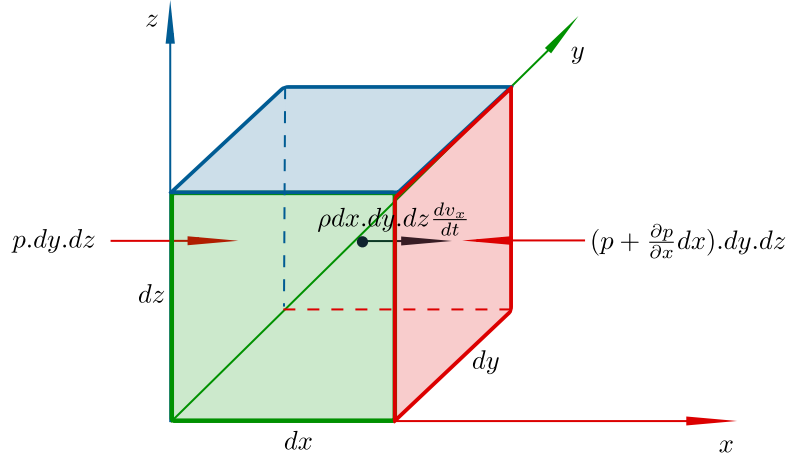
where  $\partial \rho_x / \partial t$  is the density variation in time due to the  $x$  axis flow alone; adding the equations of the  $x$ ,  $y$  and  $z$  axes,

$$\begin{aligned} \frac{\partial \rho}{\partial t} dx \cdot dy \cdot dz &= - \left( \frac{\partial(\rho v_x)}{\partial x} + \frac{\partial(\rho v_y)}{\partial y} + \frac{\partial(\rho v_z)}{\partial z} \right) dx \cdot dy \cdot dz \implies \\ \boxed{\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot (\rho \vec{v})} , \end{aligned} \quad (2.16)$$

where  $\rho$  is the density of the elemental volume.

The conservation of momentum gives an equation for the acceleration of the elemental volume; this acceleration is imparted by the balance of the forces on the element.

**Figure 3** – Conservation of momentum for the  $x$  axis in an elemental fluid volume.



With the forces and acceleration for the  $x$  axis in figure 3 the equation is

$$\rho dx \cdot dy \cdot dz \frac{dv_x}{dt} = - \left( \frac{\partial p}{\partial x} dx \right) \cdot dy \cdot dz . \quad (2.17)$$

The total time derivative of  $v_x$  can be written in terms of partial derivatives,

$$\begin{aligned} \frac{dv_x}{dt} &= \frac{\partial v_x}{\partial t} + \frac{\partial v_x}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial v_x}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial v_x}{\partial z} \cdot \frac{dz}{dt} \\ &= \frac{\partial v_x}{\partial t} + \frac{\partial v_x}{\partial x} \cdot v_x + \frac{\partial v_x}{\partial y} \cdot v_y + \frac{\partial v_x}{\partial z} \cdot v_z . \end{aligned} \quad (2.18)$$

Adding the force balance in the three directions,  $x$ ,  $y$  and  $z$ ,

$$\boxed{\rho \cdot \left( \frac{\partial}{\partial t} + \vec{v} \cdot \vec{\nabla} \right) \vec{v} = -\vec{\nabla} p} . \quad (2.19)$$

The constitutive pressure-density relation, for an adiabatic condition, is

$$p = f(\rho) = K \cdot \rho^\gamma , \quad (2.20)$$

where  $K$  is a constant,  $\rho$  is density and  $\gamma$  is the relation between specific heat at constant pressure  $c_p$  and the specific heat at constant volume  $c_v$ . The initial state allows us to write  $K$  as

$$K = \frac{p_0}{\rho_0^\gamma} . \quad (2.21)$$

Equation 2.20 can be expanded as a Taylor series,

$$p = f|_{\rho_0} + \left( \frac{df}{d\rho} \Big|_{\rho_0} \right) \cdot (\rho - \rho_0) + \left( \frac{1}{2} \frac{d^2 f}{d\rho^2} \Big|_{\rho_0} \right) \cdot (\rho - \rho_0)^2 + \left( \frac{1}{3!} \frac{d^3 f}{d\rho^3} \Big|_{\rho_0} \right) \cdot (\rho - \rho_0)^3 + \dots . \quad (2.22)$$

For a first order approximation, this equation reduces to

$$\begin{aligned} p &= f|_{\rho_0} + \left( \frac{df}{d\rho} \Big|_{\rho_0} \right) \cdot (\rho - \rho_0) = p_0 + K \cdot (\gamma \cdot \rho_0^{\gamma-1}) \cdot (\rho - \rho_0) \\ &= p_0 + \frac{\gamma p_0}{\rho_0 \rho_0^{\gamma-1}} \cdot \rho_0^{\gamma-1} \cdot (\rho - \rho_0) \implies \boxed{p = p_0 + \frac{\gamma p_0}{\rho_0} \cdot (\rho - \rho_0)} . \end{aligned} \quad (2.23)$$

Now equations 2.13 and 2.14 are applied to equations 2.16, 2.19 and 2.23; if the effects of a small perturbation over another small perturbation are neglected, that is, only the first order terms are kept, then

$$\frac{\partial \rho'}{\partial t} = -\rho_0 \vec{\nabla} \cdot \vec{v}' , \quad (2.24a)$$

$$\rho_0 \frac{\partial \vec{v}'}{\partial t} = -\vec{\nabla} p' , \quad (2.24b)$$

$$p' = \frac{\gamma p_0}{\rho_0} \cdot \rho' . \quad (2.24c)$$

Applying time derivation to equation 2.24a and substitution of equations 2.24b and 2.24c in it yields the acoustic partial differential equation, in three dimensions, for longitudinal sound waves, in a adiabatic condition and with no sources of mass in the medium,

$$\nabla^2 p' - \frac{1}{c^2} \frac{\partial^2 p'}{\partial t^2} = 0 , \quad (2.25)$$



where  $\nabla^2$  is the Laplace operator,

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (2.26)$$

and  $c$  is the sound wave propagation speed,

$$c = \sqrt{\frac{\gamma p_0}{\rho_0}}; \quad (2.27)$$

if there is no chance to misinterpret the pressure  $p'$  as a small perturbation relative to the static pressure value, the prime ' can be dropped,

$$\boxed{\nabla^2 p - \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} = 0}. \quad (2.28)$$

Again, an harmonic expression for the pressure  $p$  can be sought by separation of variables,

$$p(s, t) = p(s) \cdot e^{i\omega t}, \quad (2.29)$$

with  $s = (x, y, z)$  a position in three dimensional space,  $\omega$  the angular frequency and  $i$  the imaginary number.

Substituting equation 2.29 in equation 2.28, the Helmholtz equation in three dimensional space for pressure is obtained:

$$\nabla^2 p(s) + k^2 p(s) = 0 \quad (2.30)$$

with the wave number  $k = \omega/c$ , and no mass sources in the space.

### 2.1.4 Definition of the problem in 3 dimensions

The pressure field  $p$  in any point  $(x, y, z)$  in a limited domain  $V$ , closed by a surface  $\Omega$ , generated by an external and distributed volume source  $q$  with frequency  $\omega = 2\pi f$ , in an adiabatic situation (no energy dissipation) is governed by the second order Helmholtz equation:

$$\nabla^2 p(x, y, z) + k^2 p(x, y, z) = -i\rho_0\omega \cdot q(x, y, z), \quad (2.31)$$

where  $k = \omega/c = 2\pi f/c$  is the wave number,  $c$  the sound wave propagation speed, and  $\rho_0$  the mass density of the propagation medium.

In order to define the pressure field in  $V$ , one boundary condition must be specified in each subarea of the closed boundary  $\Omega$ . This border can be subdivided in regions  $\Omega_p$ ,  $\Omega_v$  and  $\Omega_Z$  where respectively the constraints of pressure  $p$ , normal velocity  $v_n$  and acoustical

impedance  $Z$  (or its inverse, the admittance  $A$ ) are applied,  $\Omega = \Omega_p \cup \Omega_v \cup \Omega_Z$ :

- imposed pressure  $\bar{p}$ :

$$p = \bar{p} \text{ in } \Omega_p, \quad (2.32)$$

- imposed normal velocity  $\bar{v}_n$ :

$$v_n = \frac{i}{\rho_0 \omega} \frac{\partial p}{\partial n} = \bar{v}_n \text{ in } \Omega_v, \quad (2.33)$$

- imposed acoustic impedance  $\bar{Z}$ :

$$p = \bar{Z} v_n = \frac{v_n}{A} = \frac{i \bar{Z}}{\rho_0 \omega} \frac{\partial p}{\partial n} = \frac{i}{\rho_0 \omega A} \frac{\partial p}{\partial n} \text{ in } \Omega_Z, \text{ with } i = \sqrt{-1}. \quad (2.34)$$

For the problem addressed in this thesis, the entire closed boundary  $\Omega$  is maintained at ambient static pressure ( $\bar{p} = 0$ ). This is justifiable whenever the media around and outside of the model is of a much lower acoustical impedance than the inside, as air or vacuum against water or steel; in this situation the exterior media material can not offer any significative resistance to the waves propagating inside the model region, therefore the pressure (or tension, for solids) is 0. The inconvenience is wave reflection with full intensity<sup>4</sup>. This is typical of boundaries free of tension or pressure, where the outside has a impedance  $Z_o$  much lower than the inside impedance  $Z_i$ , that is,  $Z_i \gg Z_o$ .

As pressure source  $\bar{p}$  there are the FE nodes corresponding to the transducers<sup>5</sup>; the values are applied to these nodes via the penalty method [20]. Alternatively, those nodes – with known pressures – can be removed from the unknowns and the system of rearranged equations solved only for the remaining nodes with unknown values. In the UST problem there is no volume source,  $q(x, y, z) \equiv 0$ .

### 2.1.5 Finite elements for acoustics

The method of weighted residuals informs that a set of scalar products can be used to state an equivalent mathematical formulation for equation 2.31,

$$\int_V \tilde{p} (\nabla^2 p + k^2 p + i \rho_0 \omega q) dV = 0 \quad (2.35)$$

where  $\tilde{p}$  is a weighting or test function,  $p$  is a trial function and  $V$  is the domain.

<sup>4</sup>But in opposition of phase.

<sup>5</sup>Each transducer is defined by approximately 12 nodes in the smaller dimensions models, to approximately 24 nodes in the bigger models, and also according to the number of elements per wavelength.

The FEM method divides the domain in small subdomains, the so called *finite elements*, where the field variable – in our problem, acoustic pressure  $p$  – is approximated by functions, valid only inside the FE, which interpolates the values of the nodes (points) which defines the geometry of the elements. These interpolations are done by “shape functions”  $N_i^e$  of each element  $e$  and node  $i$  of the element. This way the pressure  $\hat{p}$  is approximated by the sum of the values in all elements,

$$\hat{p}(x, y, z) = \sum_{i=1}^{n_f} N_i(x, y, z) \{\hat{p}_i\} = [N] \{\hat{p}_i\} \quad (x, y, z) \in V, \quad (2.36)$$

where  $n_f$  is the total of nodes in the discretization of the domain,  $[N]$  is a vector ( $1 \times n_f$ ) of global shape functions and  $\{\hat{p}_i\}$  a vector ( $n_f \times 1$ ) of unknown nodal pressure values.

Annex B provides additional details and steps for deriving the FE equation of the acoustical problem without internal losses for a harmonic excitation of frequency  $\omega$ ,

$$\boxed{([K] + i\omega [C] - \omega^2 [M]) \{\hat{p}_i\} = \{Q_i\} + \{V_{ni}\} + \{P_i\}}. \quad (2.37)$$

where  $[K]$ ,  $[C]$  and  $[M]$  denotes the acoustic stiffness, damping and mass matrices<sup>6</sup> of the FE model, and  $\{Q_i\}$ ,  $\{V_{ni}\}$  and  $\{P_i\}$  are vectors of internal volume sources at the nodes (or, considering density, internal mass sources) and imposed conditions of velocity and pressure to the nodes at the boundary surface of the model. In annex B the damping matrix C conveys the imposed acoustical impedance at the boundary, in appendix A is derived the equation to incorporate the viscous dissipation in the domain.

### 2.1.5.1 Finite elements with viscous losses

Derivation of acoustical FEM equations for viscous loss is presented in appendix A.

Equation 2.37 can be easily applied to problems with viscous losses with an dissipation  $\alpha$ ; the only change is in the factor that multiplies the term with the second derivative in time, which becomes

$$c^2\alpha^2 + 2i.c.\alpha\omega - \omega^2, \quad (2.38)$$

and the FEM equation with viscous loss is

$$\boxed{\left([K] + (c^2\alpha^2 + 2i.c.\alpha\omega - \omega^2) [M] + i\omega [C]\right) \{\hat{p}_i\} = \{V_{ni}\} + \{P_i\}}, \quad (2.39)$$

without volume sources  $\{Q_i\}$  – which can be easily added – where  $c$  is the sound wave propagation speed,  $i = \sqrt{-1}$ , and  $[K]$ ,  $[M]$ ,  $[C]$ ,  $\{V_{ni}\}$  and  $\{P_i\}$  are respectively the

<sup>6</sup>These names are used by analogy with the mechanical structural problem.

stiffness, mass and damping matrices, and the loading is the vector of imposed normal velocities and imposed pressure at nodes of the boundary, in the same way they were calculated for equation 2.37.

## 2.2 The central difference method

The dynamic of the pressure distribution in the domain is propagated in time with use of the central difference method. The values are calculated at regular time intervals  $\Delta t$ . This is an explicit method, which means that the updated value depends on the previous ones.

The recurrence equation is

$$[M_{\text{eff}}]\{X\}_{t+\Delta t} = \{F_{\text{eff}}\}_t, \quad (2.40)$$

where  $\{X\}_{t+\Delta t}$  is the vector of variables at the *next* time interval. The “effective mass” matrix  $[M_{\text{eff}}]$  is a combination of the real mass and damping matrices of the problem, and the “effective force” vector  $\{F_{\text{eff}}\}$  is a combination of the forcing term at the instant  $t$  and of the mass, damping and stiffness matrices of the problem and the values of the variables at the current instant  $t$  and the previous instant  $t - \Delta t$ .

In our problem, the variables in the vector  $\{X\}$  are the pressure values at the nodes of the FE mesh. The system of equations in 2.40 is solved by the preconditioned conjugate gradient algorithm.

Deduction of the equations of the central difference method are in annex C.

## 2.3 Program parallelization

It is very important to perform the lengthy simulations in the shortest possible time. Current computers employ multi-core central processing units (CPUs), and they should be used whenever possible. This is known as parallel computing, where program instructions run in many or all of the available cores simultaneously.

One modern technology to implement parallelization in a program is OpenMP.

OpenMP ([21]) is an “application programming interface” that allows use of multiprocessing in computer programs written in C, C++ and Fortran. Some advantages of OpenMP are the possibility of its implementation in just a few parts of the original

program at a time, and no big changes to the original program must be done; indeed, OpenMP instructions or “directives” will appear to be program comments to a compiler which is not OpenMP capable, therefore the same code can be compiled on different machines with older compilers without problems. The current versions of OpenMP allows also for execution in GPUs ([22]).

It is easy to parallelize `for` loops in a C source code, or, in the case of Fortran code, a `DO` loop, but also different program functions or code regions can be executed concomitantly (using the `sections` construct). Some conditions must be meet, but the necessary changes to the original (serial) program, if any, are not difficult to implement.

The finite element program comprises multiple variables in the form of vectors and matrices. As a result, computations using these variables necessitate `for` loops that operate on a considerably large number of elements. Parallelizing these loops significantly accelerates the program.

## 2.4 Ultrasound tomography in modern practice

For a few historical aspects of ultrasound in general and its applications in medicine see [9, chap. 1] and annex A.

Articles about ultrasound tomography published in recent years show that many are based on stacking of two-dimensional slices of a subject in order to obtain a 3D tomography of it, as is done with X-ray tomography; however, in general effects of diffraction, refraction and reflection of ultrasound are disregarded in the methods used, albeit they are not negligible. This creates artifacts in such tomographies.

The idea of stacking slices of the object, or layer by layer tomography, is the method used, for instance, in [23, 24, 25, 26, 27, 28], published around ten years ago, and also in the very recent [29, 30] and [31] which, albeit using transducers located in a circular ring that moves perpendicularly to the slice plane, started to account for 3D effects using transducer focusing. Each of the transducers are arranged in a low height strip, approximately 2 cm high, and can emit a pulse in other directions than the slice plane using an acoustical lens in front of rigid flat transducers and time delays. According to [29],

“... artifacts may appear in the reconstructed images due to scattering and diffraction from out-of-plane structures. Such effects are not accounted for in the 2D imaging model, where point-like transducers are typically assumed and

the spatial impulse responses (SIRs) of transducers are not modeled. Moreover, there are also significant differences in the way acoustic waves propagate in 2D compared to 3D. To reduce artifact levels and improve image quality in ring-array-based USCT [ultrasound computed tomography], there remains an important need to develop reconstruction methods based on 3D wave physics models that can incorporate the elevational focusing properties of the transducers.”

Note that the above cited works use transducers positioned around a plane ring. They can improve the results of purely plane slice by slice tomographies accounting for sound scattering from structures out of the transducer plane [29, 30]. For a tomograph to be considered truly 3D, it should take into account the waves emitted from a transducer at one height and captured at another. Therefore, the transducers must be distributed across the entire surface of the vat, that is, they must be spread circumferentially and along the height. Such equipments and prototypes already exist, as in [32, 33].

Many of these studies use full wave inversion and the BIM (Born iterative method) or DBIM (distorted Born iterative method) methods to generate a tomography. No other ultrasound tomography studies were found using finite elements to model and solve the physics of acoustical propagation.

In this thesis, through the use of tetrahedral finite elements, consideration of 3D physics phenomena is implied, even with transducers at a single plane<sup>7</sup>, as they clearly can generate echoes at the upper and lower surfaces of the domain – that must be avoided with the ALID (see also appendix C). Of course, transducers can be distributed at different heights in more elaborated models when desired.

---

<sup>7</sup>As our transducers are a circular flat disc, they already have most of their active nodes above and below the nominal transducer height in our simulations.

### 3 METHOD

The objective of the present work is to show that the Finite Element Method is suitable to solve the inverse problem associated to the the Ultrasound Tomography.

To effectively solve an inverse problem, such as the ultrasound tomography, it is beneficial to have an accurate direct problem, together with the meshes, which support and constitute the FE the program used to solve the acoustic equation. It is necessary a scheme to apply into the FE model, iteratively, the tentative sound properties and adjust these properties adequately until a criterion is met and, as result, a spatial distribution of sound properties inside the domain is considered optimal. The sound properties are specific to each tissue in the breast, and this specificity allows for the screening for neoplasms. Through a multivariable optimization process, the distribution of sound properties is modified, maximizing an *a posteriori* conditional probability density function. Therefore, some methodological tasks seem necessary:

- for the direct problem development:
  - boundary conditions imposition,
  - echo absorbing layer,
  - mesh development, and
  - parallel computing;
- and for the inverse problem development:
  - initial properties,
  - regularization strategy, and
  - multivariate optimization strategy.

## 3.1 Direct problem

The direct problem is the simulation of the propagation of an ultrasound wave in the breast. This simulation includes the water around the breast and the firing patterns of the transducers, creating a virtual ultrasound tomograph apparatus for breast examination. In this kind of breast UST the patient will lie down and her breast will hang down and be immersed in a vat with water (see figures 11a and 11b<sup>1</sup> and note the hole in the stretcher: it is the vat opening). As surrogate for real physical measurements in an US tomography examination, a more elaborated and higher precision direct problem is used, the “reference model”, and for faster computer processing, and to avoid inversion crime, in the tomography algorithm a less elaborated direct problem is employed. The models in these direct problems are finite element numerical models. The reference model has smaller finite elements, for a higher elements per wavelength relation, and a stronger dissipation wrapping around the domain (water and breast), to avoid echoes from the boundary of the whole FE model; the tomography model has (potentially) less and bigger elements, and a smaller echo dissipation layer.

### 3.1.1 General description of the simulation program

The program for ultrasound simulation by finite elements used in this work is based on Logan [15], [13] and implements an equation in the form of 2.39.

The physical hypothesis implicit by using 2.39 are that the damping model is viscous and linear, the overall model is non dispersive and, since the model is linear, the superposition principle is valid.

The finite element meshes can be created, for instance, using programs such as Gmsh ([34]), and this is described in the following section. Only elements of the linear tetrahedron type, that is, 4-nodes tetrahedrons, are considered in the calculations, with equations according to, for instance, [15, chap. 11.2].

The programming language is C. Some functions, as the ones used to solve systems of linear equations, were based on algorithms described in “Numerical Recipes” ([35]), modified for sparse matrices when needed<sup>2,3</sup>.

---

<sup>1</sup>Although this is not a tomograph, but a device for breast measurements.

<sup>2</sup>Dedicated mathematical libraries could be used, however we prefer to have comprehensive control of how the program works, being able to modify the code as needed, streamlining the code and avoiding overheads.

<sup>3</sup>The storage structure for sparse matrices is the *compressed row storage (CRS)*, according to [36].



A file with the final values at the nodes of the mesh was created, for visualization with Gmsh. The nodes of the transducers had to be known beforehand and given to the FEM program as input; to discover the number of such nodes the mesh had to be scrutinized using Gmsh.

The numerical simulation program, which represents the direct problem, has the following features: **a)** it has the capacity to read the plain mesh file that Gmsh creates – the file contains elements other than hexahedrons, such as the 2D triangular elements for the faces of any geometry we wanted to create a 3D mesh for – dismissing the need to edit and remove the “invalid” elements beforehand, **b)** it has the definition of transducers by means of a file with the respective nodes, **c)** it has a simplified assignment of each transducer role, which can be an emitter only, a receiver, or both, **d)** for the emitters, individual excitation amplitudes can be specified (however, by now, with the same frequency and phase), **e)** many excitation patterns (“*excitation cycles*”) can be defined, in succession, in a single simulation, **f)** definition of an echo absorbing/reducing layer, of ALID type (*absorbing layer using increasing damping*), **g)** creation of a detailed simulation report and saving of backup files with initial data and parameters of the simulation, **h)** data files and results have their own folders, reducing clutter and keeping an organized environment, **i)** capability to resume any interrupted simulation (due to power interruptions, full disk, etc) from a state close to the interruption<sup>4</sup>, **j)** refinement of the visualization output file. This visualization file has a certain number of “time frames” of the nodal values, at equal time intervals from the start to the end of a complete simulation, and because it has values for all the nodes this can be a big file. If more details are needed, a previous and “coarser” (only in terms of visualization time frames in the file) simulation can be re-started near a desired point in time, for a shorter simulation, however with more time frames around the time range of interest, for instance when a sound wave presumably reaches some organ or tumor, **k)** records the transducers’ readings at each time step, **l)** the visualization file, when desired, can have a specified number of equally spaced time frames.

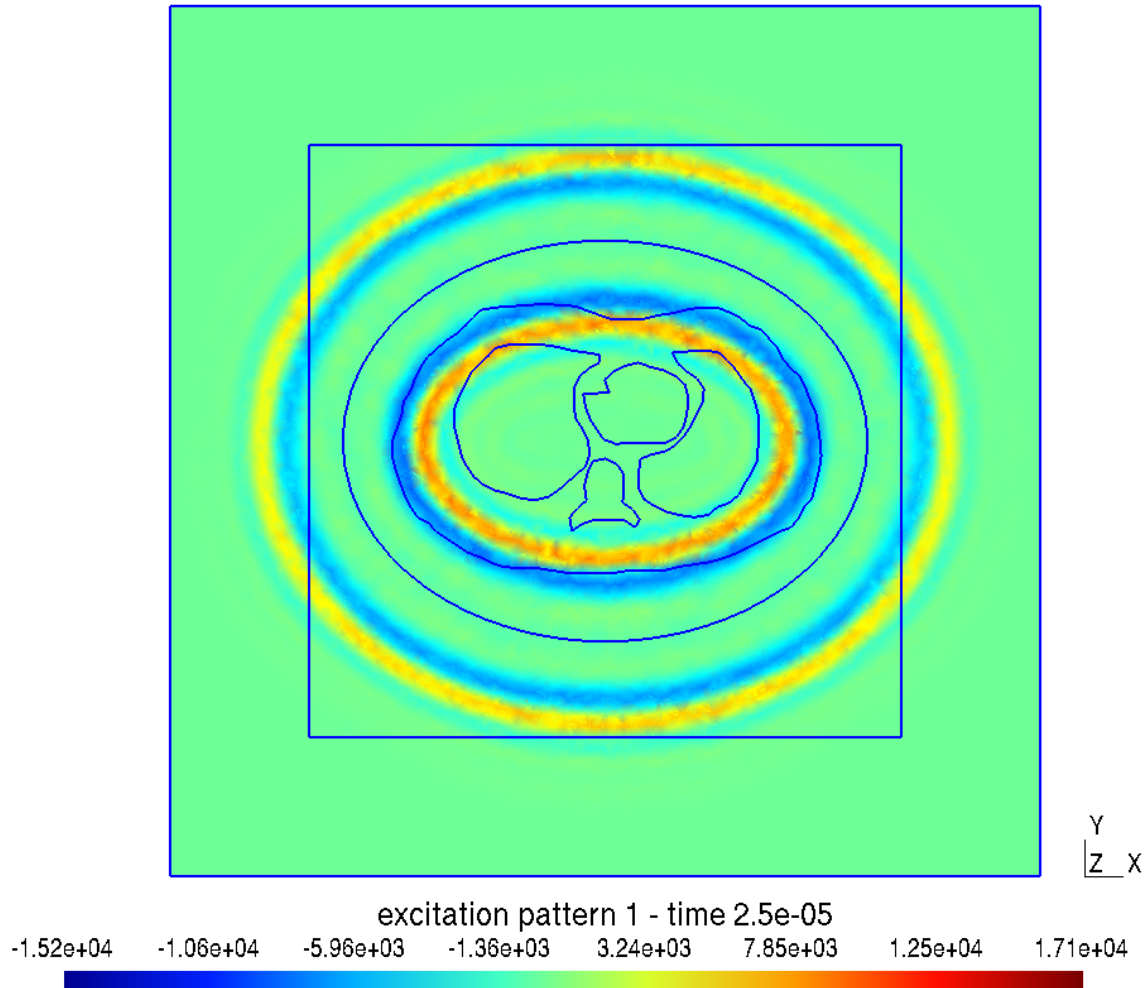
The program creates a file with the results in a format which allow visualization of the pressures inside the domain, as can be seen in figure 4. For the visualization it is used the Gmsh program, the same used to create the finite element mesh.

The program now automatically ignores any elements that were not linear tetrahe-

---

<sup>4</sup>This resume capability, for a specific *simulation*, which sometimes can be lengthy, was later removed when the program became much faster, and especially because this capability introduced a great level of complexity to the code, making the development of the tomography algorithm, which requires numerous simulations, much harder. When the tomography algorithm started to work, a new resume capability was introduced, this time for the optimization process, because the number of simulations increases with the number of voxels and number of transducers and the total run time became lengthy again.

**Figure 4** – Sample of a result from the simulation program, depicting the object contours (a thorax slice with lungs, heart and spine), and some acoustic pressure waves. Pressure in Pa.



drons, telling about any unexpected element found in the file. Therefore the mesh file created by Gmsh can be used as is.

The desired number of transducers can be defined with ease, item **b)**: an auxiliary program reads the mesh file – which for this functionality must have been exported in the `inp` file format by Gmsh, a file type used by the Abaqus FEA<sup>5</sup> program – and extracts a list of transducer nodes; this list, modified with some identification headers and the the nodes of each transducer (quantity and their numbers), is one of the input data files for the simulation program. Another data file indicates how each transducer will work, item **c)**, where a sequence of those indications allows a single run of the direct problem numerical simulator to change which transducers acts as emitters, (item **e)**); in an analogous way, a discrete time series of amplitudes (item **d)**) allows modulation of the excitation in each

<sup>5</sup><https://www.3ds.com/products-services/simulia/products/abaqus/>.

transducer, with the remark that (at the current state) a single frequency is used for all the transducers.

Itens *g)* and *h)* are not essential for the simulations, but help to keep the results tidied up and with greater control of how each result was generated. The simulation report identifies all data files (mesh, excitation cycles and amplitudes, etc.), size of the problem (number of tetrahedrons), total of time steps, frequency and maximal amplitude of the excitation, among other informations. All the files the program creates<sup>6</sup> have identification headers; also, as will be explained in section 3.1.4, the geometry file is fully commented, and all other data files have identification headers and an organized internal structure, therefore the program can catch some data errors or inconsistencies, and the management of research data, for instance as required by FAPESP<sup>7</sup> [37], could be met<sup>8</sup>.

The simulations can be affected by echo provenient from the artificially finite borders of the numerical simulation domain, avoided when using layers of ALID, item *f)*. ALID layers (*absorbing layer using increasing damping*) is a solution for echoes which is easy to implement in the program [38]<sup>9</sup>.

Item *i)* is useful in the event of any interruption of the program, as at restart it will identify the presence of a recovery file with the “state” of the simulation at a point close to the interruption. The user has the option to load this state and proceed with the simulation, without restarting from the first time step. Another improvement, item *l)*, fixes the inconvenience of having only the final step to see the obtained pressure field; since acoustics is a propagation phenomenon it is useful the capability to see the process over time, with a specified number of time frames selected at the start of the simulation.

As those time frames could not be enough to “capture” and show a particular instant, say, when the wavefront reaches one tissue of interest, item *j)* was created. With this functionality, and a few recover files (copies of the ones used to proceed an interrupted simulation), a new simulation can begin close to the desired time interval, creating a supplementary visualization file with more frames for a detailed visualization; those files have information about *all the nodes* of the model.

As just stated, the readings of each transducer are recorded in a file, at each time

---

<sup>6</sup>transducer readings, complete excitation signal in time at each time step, even when the excitation is read and interpolated from a given file, etc.

<sup>7</sup>FAPESP, Fundação de Amparo à Pesquisa do Estado de São Paulo – The São Paulo Research Foundation.

<sup>8</sup>This is not required with this work.

<sup>9</sup>It is worthy to remark that in [38] (e. g. equation (3-15)) Drozd uses Rayleigh damping, a combination of the finite elements matrices of mass and stiffness. The use of Rayleigh damping is usual for structures; in the numerical simulation program, it is used a damping or absorptivity parameter.

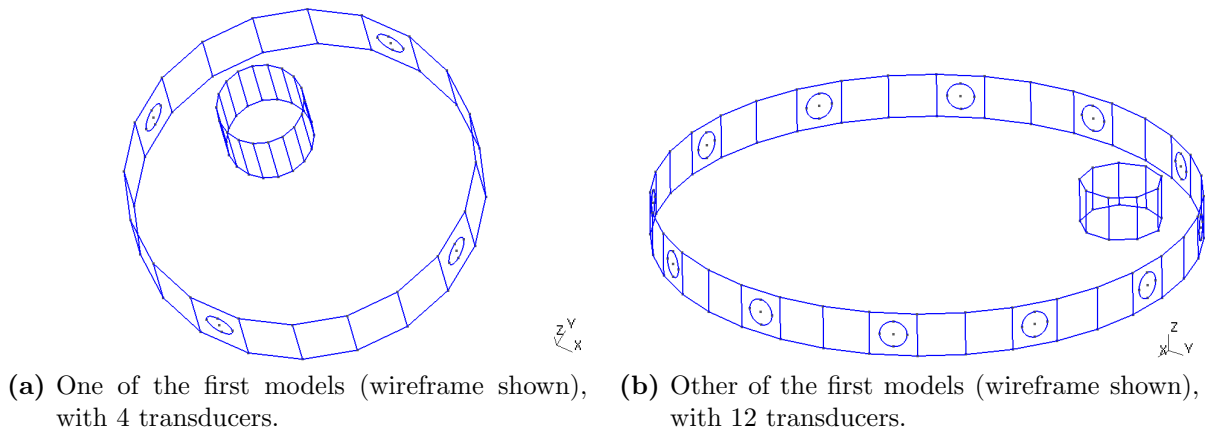
step of the simulation. Those readings are the mean of pressure values of all transducer interface nodes. Therefore, there are files for visualization inside the domain, as in figure 4 and it is possible to plot the measured values in the model, as in figure 19 on page 73.

### 3.1.2 Boundary conditions: the penalty method

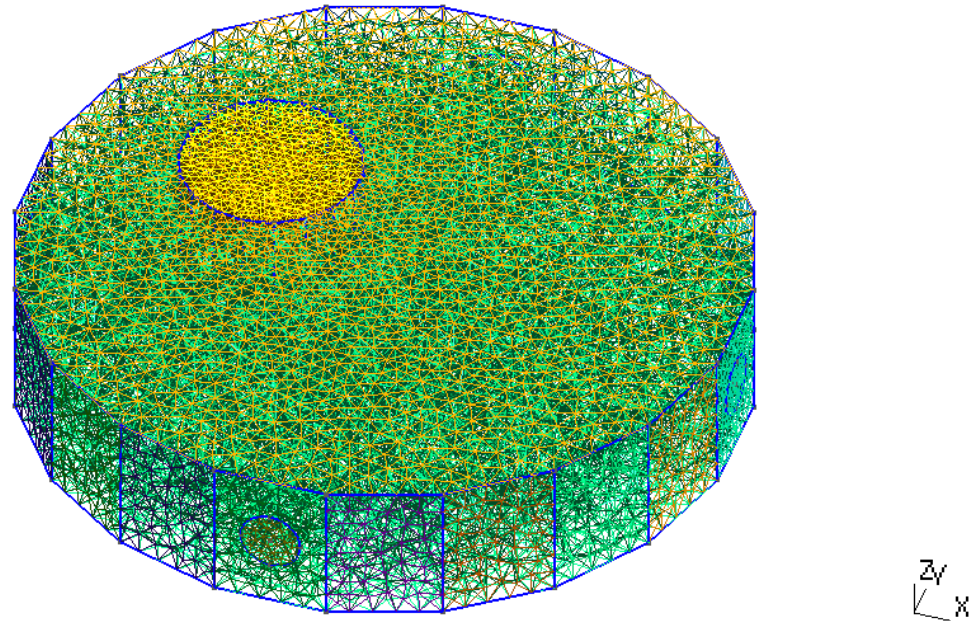
The finite element formulation shown in equation 2.37 assumes that the excitations, pressure, normal velocity and impedance, given by equations 2.32–2.34, are applied (or imposed) at the *boundaries* of the model, and are considered *external* sources.

Some of the first simulations with the finite elements program were made using a very simplistic model, with the transducers placed at its border, as seen in figure 5. One of the 3D meshes is shown in figure 6. At the time, there was no ALID around the model.

**Figure 5** – Some of the first finite element models used with the simulation program.

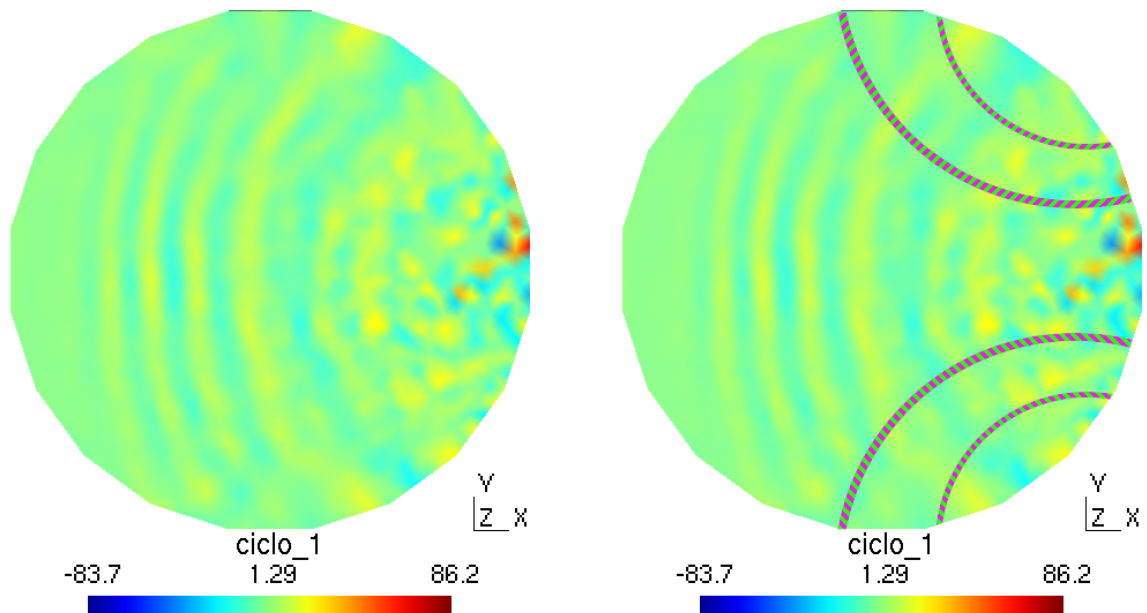


**Figure 6** – 3D mesh with tetrahedral elements for the model in figure 5a. Generated using Gmsh, with Netgen optimization algorithm.



Looking at the results one could soon realize that an echo is generated when a sound wave reaches the model border, as shown in figure 7. This echo is strong.

**Figure 7** – The first models were susceptible to echoes at the borders, because of the lack of an ALID layer. One emitter active at the right position. This figure shows the pressure wave inside a cylindrical domain. Pressure in Pa.



(a) Sound waves propagating through the model and creating echoes at the borders.

(b) Same picture, notice the echoes marked here for clarity.

Therefore the ALID<sup>10</sup> layer was introduced in the model to avoid the echoes. The ALID covers all sides of the original model, but then groups of nodes of the transducers became *internal* to the model, while in the equations they are considered at the boundary of the model, the imposed pressure  $\{P_i\}$  at the right of equation 2.37. In this situation, internal nodes with known or imposed values, it is usual to exclude the respective variables from the system of equations – as these nodes values are no longer unknowns – so the remaining system of equations is determined and can be solved. However, in the actual direct problem numerical simulator and tomography program, an alternative method is used, the *penalty method*. This method is presented in [20]. Besides boundary conditions it applies also to constraints, and the nodes that represent the transducer impose the excitation pressure.

Let  $\bar{p}_i$  be the excitation pressure at the node  $i$  – the rationale applies to more than a single node as well. If a large number  $\bar{\omega}$  is taken, a modified functional  $\Pi^*$  of the problem (see equation An-B.40) can be written:

$$\Pi^* = \frac{1}{2} \{\hat{p}\}^T [\hat{Z}] \{\hat{p}\} - \{\hat{p}\}^T \{\hat{f}\} + \frac{\bar{\omega}}{2} (\hat{p}_i - \bar{p}_i)^2, \quad (3.1)$$

where  $\{\hat{p}\}^T$  is the transpose of  $\{\hat{p}\}$ .

The functional will be minimized when equation 3.1 is stationary,

$$\delta\Pi^* = \{\delta\hat{p}\}^T \left( [\hat{Z}] \{\hat{p}\} - \{\hat{f}\} \right) + \bar{\omega} \delta\hat{p}_i (\hat{p}_i - \bar{p}_i) = 0. \quad (3.2)$$

The pressure at node  $i$  can be written in terms of the vector of nodal pressures and a “localization vector”  $\{e_i\}$  of the same dimension of  $\{p\}$  and defined such as

$$\hat{p}_i = \{e_i\}^T \{\hat{p}\}; \quad \{e_i\}^T = (0, \dots, 1, \dots, 0), \quad (3.3)$$

with the value 1 in the position matching the node  $i$ . Substitution of 3.3 in 3.2 yields

$$\delta\Pi^* = \{\delta\hat{p}\}^T \left( [\hat{Z}] \{\hat{p}\} - \{\hat{f}\} \right) + \bar{\omega} \{\delta\hat{p}\}^T \{e_i\} \{e_i\}^T \{\hat{p}\} - \bar{\omega} \{\delta\hat{p}\}^T \{e_i\} \bar{p}_i = 0. \quad (3.4)$$

The vector  $\{\delta\hat{p}\}$  is not a null vector; the equation can be divided by it and the terms of equation 3.4 regrouped to obtain

$$\left( [\hat{Z}] + \bar{\omega} \{e_i\} \{e_i\}^T \right) \{\hat{p}\} = \{\hat{f}\} + \bar{\omega} \{e_i\} \bar{p}_i. \quad (3.5)$$

---

<sup>10</sup>Absorbing layer using increasing damping.

If the matrix and vectors in equation 3.5 were expanded, we get

$$\begin{bmatrix} \hat{Z}_{1,1} & \hat{Z}_{1,2} & & \cdots & \hat{Z}_{1,n_{eq}} \\ & \ddots & & & \\ & & \hat{Z}_{i,i} + \bar{\omega} & & \vdots \\ & & & \ddots & \\ & & & & \hat{Z}_{n_{eq},n_{eq}} \end{bmatrix} \begin{Bmatrix} \hat{p}_1 \\ \vdots \\ \hat{p}_i \\ \vdots \\ \hat{p}_{n_{eq}} \end{Bmatrix} = \begin{Bmatrix} \hat{f}_1 \\ \vdots \\ \hat{f}_i + \bar{\omega} \bar{p}_i \\ \vdots \\ \hat{f}_{n_{eq}} \end{Bmatrix} \quad (3.6)$$

(symmetry)

The equation which corresponds to the  $i$  row of 3.6 is

$$\sum_{\substack{j=1 \\ j \neq i}}^{n_{eq}} \hat{Z}_{i,j} \hat{p}_j + \hat{Z}_{i,i} \hat{p}_i + \bar{\omega} \hat{p}_i = \hat{f}_i + \bar{\omega} \bar{p}_i . \quad (3.7)$$

For a sufficiently large value of  $\bar{\omega}$  we get  $\bar{\omega} \hat{p}_i \approx \bar{\omega} \bar{p}_i$ , and therefore the constraint  $\hat{p}_i = \bar{p}_i$  is practically satisfied. Generalizing for constraints at many nodes, it suffices to add  $\bar{\omega}$  to the diagonal of the  $i$ -th element and at the same time add  $\bar{\omega} \bar{p}_i$  to the load vector at the right hand side, at the positions affected, in the system of equations.

One caveat is that a really big value for  $\bar{\omega}$  can lead to a ill-conditioned system of equations. An adequate value for this parameter can be established through tests. For the finite elements problem in this thesis the values of  $3 \times 10^{10}$ ,  $5 \times 10^9$ ,  $2.5 \times 10^4$  and  $1 \times 10^4$  were evaluated, and for the latest meshes (for a very simplified “breast” model) the value  $\bar{\omega} = 200$  was used. For the biggest values there was a very close approximation  $\hat{p}_i \approx \bar{p}_i$ , but the pressures appeared to concentrate and remain around the emitters instead of being propagated to the neighboring finite elements. The value chosen, 200, gives a still good approximation for the imposed pressures, and had no concentration of nodes with persistent high pressures in the region of the emitters.

### 3.1.3 Echo absorbing layer

It is know that a sound wave can be reflected as echo when it reaches a obstacle. In our models, one obstacle to be aware of are the model boundaries. According to imposed conditions at these boundaries, the wave can be attenuated or may reflect with full intensity.

After a few initial tests with the simulator program, it was evident that some kind of echo attenuation was required at the boundaries. Two common solutions exist: PML, for perfectly matching layer, and ALID, absorbing layer using increasing damping. The best

solution is the PML, however its mathematical perfection does not exist in approximate solutions, as is the case with FEM, and its implementation is more complex than the use of ALIDs. For the simulator, a simpler ALID was implemented. The main drawback is the use of a ticker layer, with more finite elements than in the case of a PML layer.

It is very important that any increase in the attenuation of the absorbing layer be smooth, otherwise the wave starts to reflect as it encounters a material (or a finite element) with different physical properties.

### 3.1.4 Mesh development

During initial development of the simulation program there was the opportunity to model and simulate a correlated problem, the direct problem of the thorax and lungs. The thorax problem was used to see how FE meshes could be generated and what functionalities the ultrasound simulator must have in order to be adequate for UST. This know-how was later used for the breast UST.

#### 3.1.4.1 Meshes for ultrasound inside the thorax

During testing stages of the direct problem simulation program, the model used was a 2D thorax slice, with the vertebral spine, heart and lungs in it. The thorax section was centered inside a water filled rectangle. The solutions to the requirements and problems addressed with this model would later apply to a breast model, as multiple transducers around the region of interest, definition of internal boundaries for different materials, and definition of an ALID layer because of the echoes.

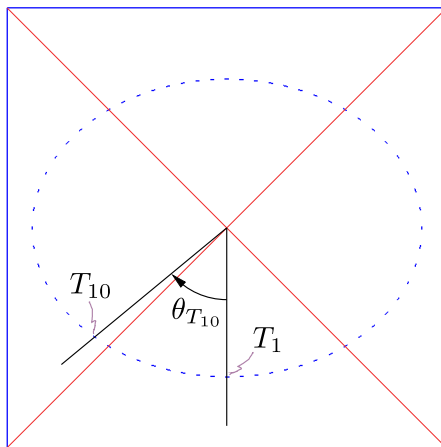
The contour of the organs was given by Cárdenas [39] and is the same used in his PhD thesis and in other works from researchers from his group, therefore, some results comparisons could be made later on. Cárdenas solves the ultrasound problem using the k-Wave software ([40]).

In this model the positions of the transducers are defined by the angle between a vertical line and the center of the transducer considered, as depicted in figure 8. The first transducer is the bottom one, and numbering is clockwise. The  $x$  axis points to the right, the  $y$  to the top.

The excitation pattern uses a single frequency for all the transducers, in phase, with amplitude varying according to the position of the transducer. For the inverse problem by the D-bar method, a series of excitation patterns are used where the amplitudes follows



**Figure 8** – Angular position  $\theta_\ell$  of a transducer “ $\ell$ ” (here shown the tenth) for the D-bar problem.



the equation

$$T_X^n(\ell) = \begin{cases} M_{Tr} \cos(n\theta_\ell) & \text{for } n = 1, 2, \dots, L/2 \\ M_{Tr} \sin((n - L/2)\theta_\ell) & \text{for } n = L/2 + 1, L/2 + 2, \dots, L \end{cases} \quad (3.8)$$

where the transducers' total number  $L$  must be even and  $M_{Tr}$  is the maximum amplitude (energy) used in the excitations. The number of excitation patterns is equal to the number of transducers.

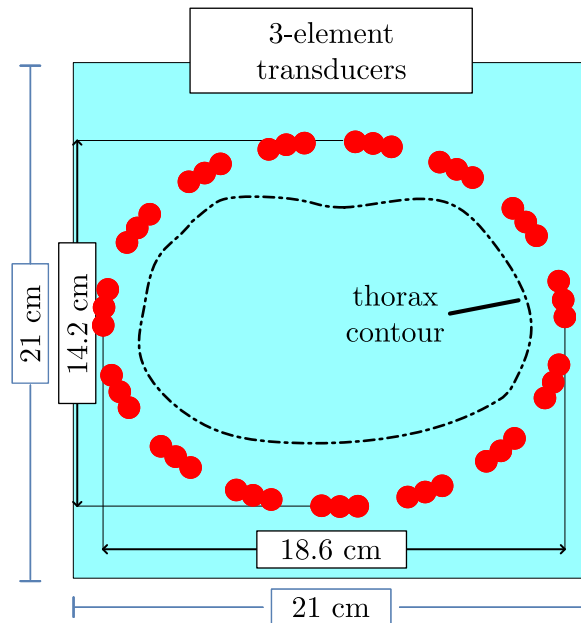
In the finite element model the transducers are, like in [39], over an ellipse and without direct contact with the thorax, but the space in between is filled with water, as shown in figures 9 and 10. In a real situation of ultrasound for thorax and lung tomography, however, most probably the transducers – unless if they were very big – will be placed in a belt, and this belt fastened around the thorax, putting the transducers in touch with the skin, with only a tiny layer of gel for impedance matching.

Rigorously speaking, the finite element model of this thorax section is a tridimensional one. However, with 2.5 cm in height and 21.0 cm in length, a ratio height/length of 12%, it can be considered to behave as two dimensional model. On the other hand, the models solved by [39] are intrinsically of two dimensions.

Initially meshes had 6 elements per wavelength, according to [41], later meshes were used with 10 elements per wavelength, looking for greater fidelity in the representation of the wave phenomenon, according to [42].

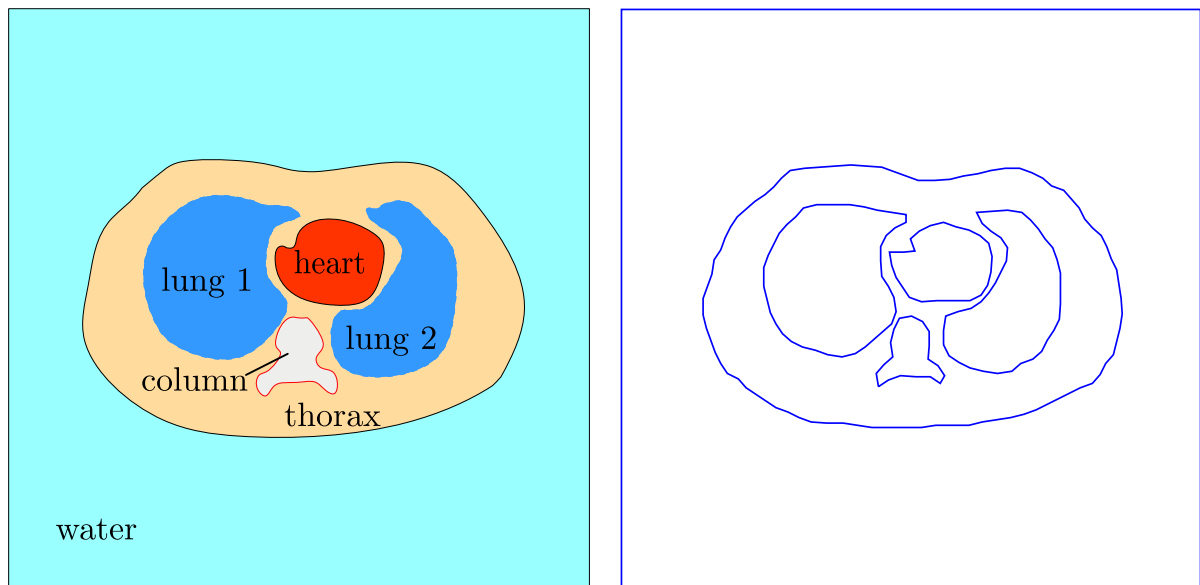
The sound propagation speeds in each of the model media are, except for the lungs, the same used by [39], *without variations in the spine sound speed*; for the lungs the speed

**Figure 9** – Dimensions of the thorax model and of the transducers' ellipse.



Source: adapted from [39].

**Figure 10** – Comparison of the original high resolution contour with the contour sampled at every 15<sup>th</sup> point from the original.



(a) High resolution contour (modified from [39]).

(b) Simplified contour, with one point for every 15 from the original.

is like in [43], valid for a frequency range from 10 kHz to 1 MHz; frequencies in which lungs are translucent for ultrasound. The actual sound speed values are in table 1.

**Table 1** – Sound propagation speeds  $c_\ell$  in each thorax model component (m/s). Longitudinal waves.

| medium          | $c_\ell$ (m/s) |
|-----------------|----------------|
| water           | 1540           |
| lungs           | 1500           |
| vertebral spine | 1630           |
| thorax interior | 1520           |
| heart           | 1570           |

Source: data from [39] and from [43].

As already stated, to avoid echoes in the upper and lower planes of the thorax slice, in order to simulate a 3D domain, and on the sides beyond the walls containing the water around the thorax, it was needed to wrap this whole volume with a 3D ALID layer. With the inclusion of this wrapping the mesh size increases considerably.

In order to generate the finite element meshes with Gmsh, one must first define the geometry of the model, that is, the complete set of points, lines, curves and surfaces that define the figures and volumes of the model. In our case, the geometry files are created by commands in a script file read and executed by GNU Octave ([44]); this command file is commented and has the important parameters clearly identified: dimensions of the transducers' ellipse axis, height of the thorax slice, length of the square region with water, quantity and size of the transducers, etc., therefore with the edition of a few values many different models can be created for study. Comments are also included in the generated geometry file, and the most important values like the number of transducers and how many elements per wavelength are displayed in the Gmsh message area when the geometry file is opened, therefore a user can double check if the desired data are being dealt with.

Selection of how many elements per wavelength to use is based on calculation of a “*characteristic length*” which Gmsh utilizes; this value regulates the (approximate) size of finite elements in free regions of the geometry, while close to points which define the geometry the elements have to conform to the geometry. That is why sampled contours are used, to avoid an excess of very small elements close to the high resolution geometry contour, with bigger elements (with the desired size) far away from the geometry.

### 3.1.4.2 Meshes for the breast with a tumor

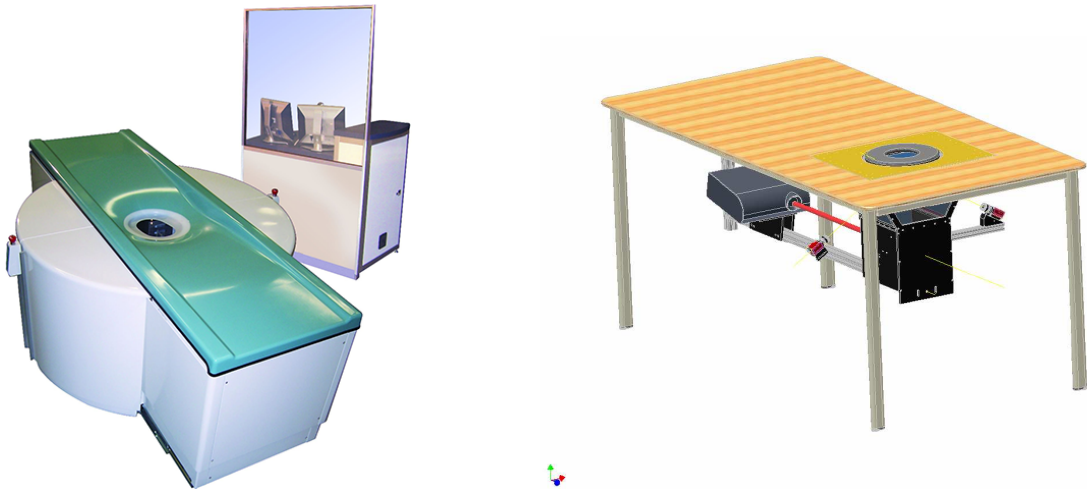
One of the phases of this work is the development of numerical phantoms to be used in the simulations. Initially a realistic phantom was pursued. The actual model does not

resemble a breast at all. It is simpler and more suitable for software development and initial tests.

**Realistic human breast:** an attempt to create a realistic breast involved the use of computer generated models and also some stereo-lithography (.STL) files for 3D scanned bodies or torsos. These STL files contain only information about the surface of the objects.

**Deficiencies of the realistic models:** all the realistic phantoms above share a common pitfall. It is reasonable to think that in an ultrasound breast tomography the person undergoing the examination will have to lie over a table, or bed, with their breast (one at a time) suspended through a table hole and immersed in liquid, according to figure 11. In this situation, de Kleijn ([45]) states that the breast shows some little concavities, and, with time<sup>11</sup>, the breast will even swell and deform, because of blood accumulation in it due to gravity. The tomograph shown in 11a does a complete scanning in about 10 seconds.

**Figure 11** – Some equipments for breast tomography and for breast measurements.



(a) Koning Breast computed tomograph (old model, [46]).

(b) CAD representation of the test bed from [45].

**Simplified human breast:** the attempts to create a realistic finite element model of a human breast have a very high number of FE elements. Notice that the values shown in table 6 already include the breast, the water around it and an additional layer of ALID elements. This high number of elements has potential to make the computational run time considerable, even considering parallelization in the program.

<sup>11</sup>In his study the time for the measurement is around 15 minutes, consequently breathing also affected measurements [45].

For practical reasons, a simpler breast model is more suitable for the development of the tomography algorithm, allowing for faster program development and testing. Once the algorithm is finished and working well, more effort can be directed to realistic models (for real world experimentation of the program) and to improvements in parallelization and faster execution time.

Therefore the breast was substituted by a small cylinder, with a smaller cylinder inside it to act as a tumor. Again, as in the thorax model, the breast model will be of small height. This will allow for smaller meshes to simulate, speeding up the tests and development of the tomography algorithm.

At early stages of the tomography algorithm development – and before regularization – some crude tests of the optimization gave bad results. As the sound speed in water was then the same used in the thorax simulations, 1540.0 m/s, and, according to some sources<sup>12</sup> that for the breast was chosen as 1545.0 m/s and for the tumor 1570.0 m/s, the small difference of only 5.0 m/s from water to breast was considered to be a problem.

Since breast data had big variations from one reference to another, or were much specific and not adequate for the breast as a whole, e. g. 1440.0 m/s for breast fat and 1505.0 m/s for the mammary gland, a decision has been made regarding sound speeds of 1510.0 m/s for the breast [47, cited as “average value”], 1480.0 m/s for water [2] and 1560.0 m/s for the tumor [48], which allow for 30.0 m/s differences from water to breast and 50.0 m/s from breast tissue to tumor. These speeds were chosen for use with the simplified model and are in table 2.

**Table 2** – Sound propagation speeds  $c_\ell$  in the simplified breast model (m/s). Longitudinal waves.

| medium        | $c_\ell$ (m/s) |
|---------------|----------------|
| water         | 1480.0         |
| breast tissue | 1510.0         |
| tumor         | 1560.0         |

The ultrasound frequency is 25 kHz. Higher frequencies, with shorter wavelengths, would imply in more elements in the mesh for the same number of elements per wavelength.

<sup>12</sup>Among them, for instance, <http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/usound2.html>, <https://itis.swiss/virtual-population/tissue-properties/database/acoustic-properties/speed-of-sound/>, <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470561478.app1>, <https://asa.scitation.org/doi/pdf/10.1121/1.1336896> and <http://www.rfcafe.com/references/general/velocity-sound-media.htm>.

The diameter of the models (reference and tomography) was increased from 3 to 10 cm, this change was thought to help with the optimization process.

There are two meshes: a reference mesh and a simulation one. The reference is used as a standard or surrogate of real world measurements; the simulation mesh is used as direct problem in the tomography algorithm and is simpler to allow faster (compared to the reference model) execution times.

The mesh for the reference model has 15 elements per wavelength (for a very accurate representation of the wave using linear elements), an ALID of type 5/5/3<sup>13</sup> whose thickness is 9.8667 cm – a thick ALID in order to minimize echoes. The tumor has a radius of 0.7678 cm, in a breast of radius 4.0 cm and vat of radius 5.0 cm and height 1.5 cm.

The mesh used for the direct problem (simulation) in the tomography program has 6 elements per wavelength and an ALID of type 3/2/3 with thickness 6.24 cm. Breast and vat are the same as in the reference model. This very simplified model has less than 50,000 elements, including water and ALID. This is 34 times smaller than the model cuba2020 and 115 times smaller than cuba\_2020B<sup>14</sup>. The number of transducers is 24. The number of elements and nodes of both reference and tomography meshes are in table 3. Their geometry is in figure 12.

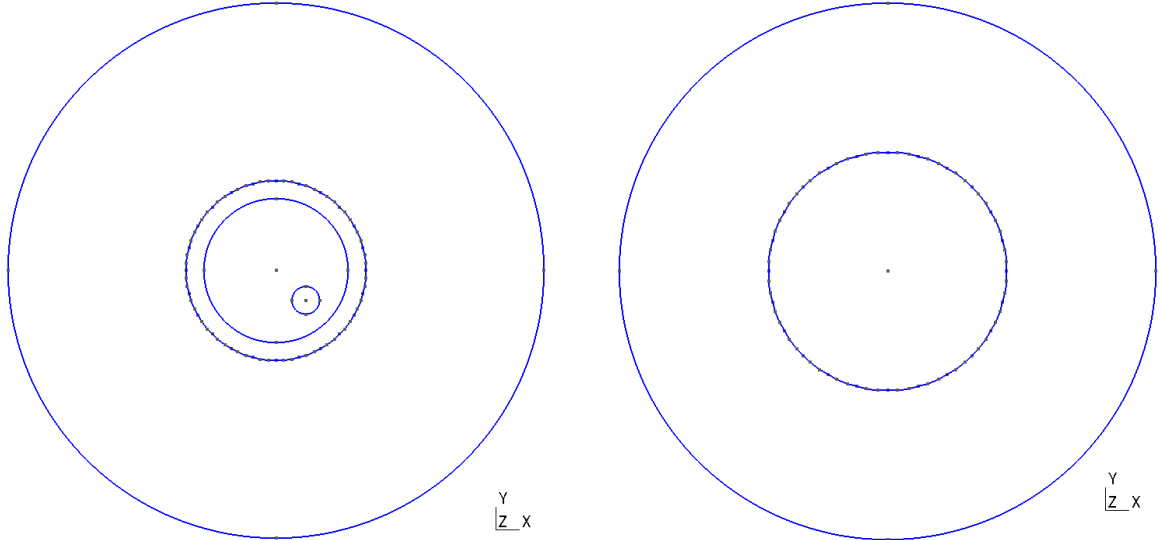
**Table 3** – Number of elements and nodes in the simplified breast model, including the ALID layer.

| mesh       | elements | nodes  |
|------------|----------|--------|
| reference  | 1348973  | 227965 |
| tomography | 49093    | 8952   |

<sup>13</sup>Recall equation 4.1 and the description in its paragraph, at page 62, and figure 13.

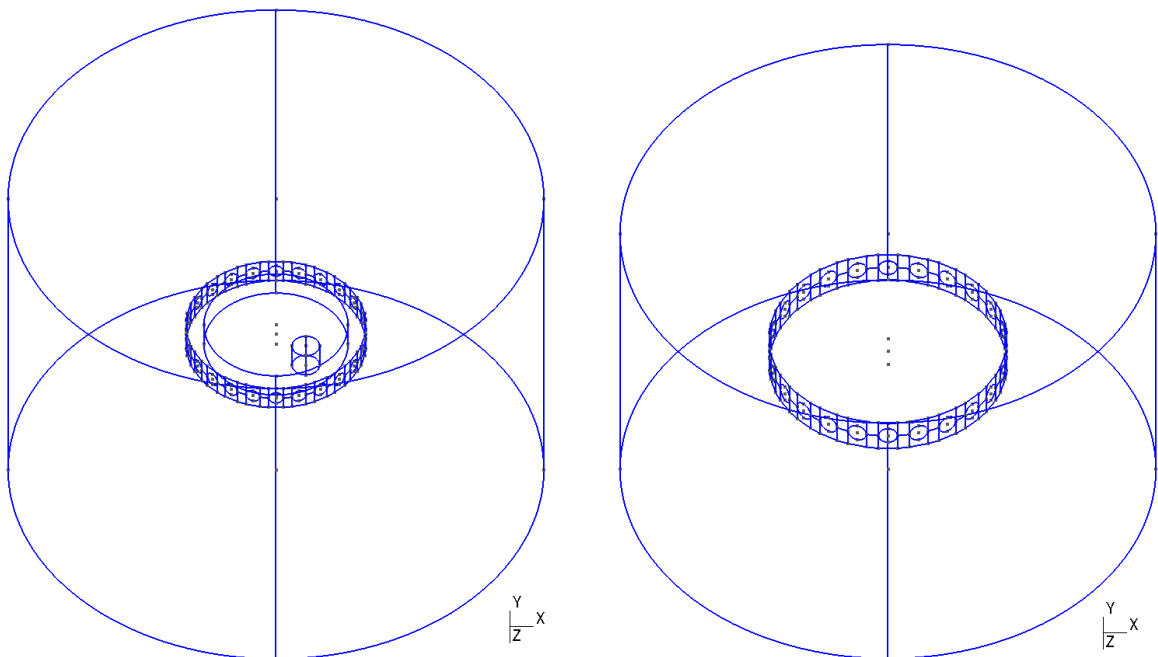
<sup>14</sup>However, the realistic breast models were created for an ultrasound of 50 kHz, with 10 elements per wavelength, therefore for 25 kHz with 6 elements per wavelength an adjustment factor of  $(\frac{25}{50} \times \frac{10}{6})^3 \approx 0.579$  can be applied to the number of elements, and the simpler breast would be, for a fair comparison with the same conditions, 19,6 times smaller than cuba2020 and 66.5 times smaller than cuba\_2020B for the same ultrasound and elements per wavelength.

**Figure 12** – Wireframe geometry for the simplified breast model. At left the reference model, at right the model used for the direct problem in the tomography program. Not in the same scale – the disc surrounded by small lateral discs (the transducers) is the same size in both models.



(a) Simplified breast and tumor used as reference, in top view. From the outside to the center: ALID, water, breast, and tumor in the breast, down and to the right. The circle with black points (because of the transducers) is the vat wall.

(b) Simplified vat model used in the tomography as direct problem, with ALID layer, in top view. The tomography algorithm will have to correctly assign properties of water, health breast and an likely tumor to the elements inside the vat.



(c) Simplified breast and tumor used as reference, in oblique view.

(d) Simplified vat model used in the tomography, in oblique view.

### 3.1.5 Parallel computing

Parallel computing is the use of many processor units, simultaneously, to perform calculations and in this way speed-up program throughput. This means the use of all available processor cores that nowadays CPUs (computer processing units) have, and also the use of simultaneous CPUs and networked computers to do calculations.

The first finite element program was serial, that is, unaware of how many CPU cores and simultaneous multithreading (SMT)<sup>15</sup> the computer could offer. Only one of the many cores of modern computers was used. After most of the changes in the list *a)–l)* were done, and the usability of the program deemed good, the development of the code was focused on better performance, via parallelization of the program.

Some parallelization technologies as OpenACC ([49], [50], [51]) and CUDA were initially evaluated. Then another parallelization option, OpenMP ([21]), was investigated.

#### 3.1.5.1 Parallelism using OpenMP

OpenMP is used in the FEM simulation and tomography program. The GNU compilers used to develop the program are compatible, implementing many of the most recent specifications of OpenMP.

The current versions of OpenMP allow also for execution in GPUs ([22]).

## 3.2 Inverse problem

In order to obtain a tomographic image Baye’s Theorem will be used, the image should maximize the posterior probability density function, given that a set of pressure measurements had happened. The prior probability density function will be used as a regularization of the problem that, otherwise, has an infinitude of solutions [52]. Maximizing the posterior probability is consistent with minimizing a performance index.

From a computational point of view, a way to assign properties of each voxel to the corresponding finite elements in the model was devised. As the properties in each voxel affect the simulated measurements, they must be changed appropriately to minimize the performance index through an iterative process. For that purpose a performance index is devised and an optimization algorithm is adopted.

---

<sup>15</sup>Or “hyper-threading” in Intel parlance.



### 3.2.1 Initial guess

The initial guess tells the program the starting values, in each voxel, to begin the optimization process. The standard values are the sound propagation speeds of water and of an homogeneous breast tissue, but some voxels could start with tumor values if desired. Also, the final tomography of a previous run of the program could be used to proceed with the optimization.

A small program was developed which asks for the type of voxel grid to be created, the sound speeds, the dimensions of the model elements, and the grid discretization in the XY plane and in the height (Z axis). As the simulation model uses very simple geometric figures for the breast inside the vat, only the radius and height of the vat and of the breast are needed as input dimensions. This program then saves a file with the initial sound speed values, assuming only water and breast exist in the vat<sup>16</sup>. This file is used as input for the tomography program.

#### 3.2.1.1 How voxels and finite elements are correlated

The tomography algorithm loads the initial values of each voxel at startup. When the individual finite elements have their properties calculated by the program to build up the mass, damping and stiffness matrices of the FE problem, they use the sound speed corresponding to the voxel where the geometric center of the element lies.

Inside the program, the voxel values are stored as a vector variable; each vector element is an “optimization parameter” for the optimizer, and only the optimizer changes the values in this vector. For each new iteration of the program, the elements have their acoustical properties calculated with updated vector values. Also at each iteration a new file is saved with updated voxel values.

### 3.2.2 Regularization: sample based prior

It is known that inverse problems are ill-posed and their solution requires special methods ([53], [54], [52], [55]). Generation of a tomographic image is an ill-posed problem.

Regular, ordinary mathematical problems are, on the other hand, well posed. Most of the mathematical problems usually dealt with, and most problems of engineering interest, belong to this class.

---

<sup>16</sup>A graphical representation of the file created by this program is in figure 32a on page 86.

Being  $T$  the (linear) forward operator of which the inversion is wanted,  $T : X \rightarrow Y$ ,  $T$  bounded and  $X, Y$  separable Hilbert spaces, and  $\mathcal{R}$  and  $\mathcal{N}$  respectively the range and nullspace of  $T$ . According to [53], the inversion of  $T$  is said to be well posed if it satisfies these conditions<sup>17</sup>:

- *existence*: for all  $g \in Y$ , there exists  $f \in X$  such that  $Tf = g$ , that is,

$$\mathcal{R}(T) = Y, \quad (3.9)$$

- *uniqueness*: for all  $g \in Y$ , there is a unique solution,

$$\mathcal{N}(T) = \{0\}, \quad (3.10)$$

- *stability*: the solution depends *continuously* on the data,

$$T^{-1} \in \mathcal{L}(Y, X), \quad (3.11)$$

where  $\mathcal{L}(Y, X)$  is the set of all bounded linear transforms mapping from  $Y$  to  $X$ . If any of the three conditions is not met, the problem is ill-posed.

In practical terms, non-uniqueness is undesirable because, if the same image or tomography could be created by two or more different objects, this is problematic for the (correct) reconstruction.

Lack of stability leads to numerical problems. And real world measurements are subjected to noise, and have limited accuracy; even computer simulated values are subjected to numerical accuracy and truncation errors. A non-stable solution signifies that any small deviation, in any variable, causes a great variation in the values of the solution, therefore it can be far from the true (and unknown) solution.

The theory of regularization deals with the problems of stability. So, the solution will depend on the noise level or measurement accuracy, but will be “attracted” to viable values. If the first condition is violated, a “classical” solution can be relaxed and, instead, a generalized solution may be seek. As stated by [53], “...how one can aim at extracting information as stably as possible from an unstable system. This is the goal of regularization.” The results for nonlinear inverse problems are inspired by the linear problem.

The problem of ultrasound tomography of the breast is ill-posed. For instance, even with an initial guess with voxel values close to the desired solution, known from the

---

<sup>17</sup>Known as Hadamard conditions for well posedness.

forward problem used by the simulator as the reference, the optimization could not give good results most of the time, with voxels usually going to the maximum or minimum allowed limits.

If an exact solution  $g$  can not be found, it is usual to look for the best fit  $f$ , a least-squares solution, which minimizes the Euclidean norm,

$$\|Tf - g\| \leq \|Th - g\|, \quad \forall h \in X. \quad (3.12)$$

The objective function to be optimized depends on the sound propagation speed  $c_l$ , but of course the function can have any parameters of interest, for instance damping (dissipation) of the different tissues subjected to the tomography process. Let  $r$  be this function; before regularization it is of the form

$$r(c_\ell) = \|\phi_m - \phi(c_\ell)\|^2 = (\phi_m - \phi(c_\ell))^T (\phi_m - \phi(c_\ell)) \quad (3.13)$$

where  $\phi_m$  is a performance index that depends on the reference measured pressure values – which are simulated by a precise model in this work – and  $\phi(c_\ell)$  are the values from the tomography direct model, function of the sound wave speeds  $c_\ell$  in each voxel. For simplicity, the adopted idea is to match in time the position of the pressure peak in the reference and in the tomography direct problem; the performance considers the squared delays of the peaks in the time history of all available signals: for each transducer as listener, and for each emitter.

A generalized Tikhonov regularization term

$$\lambda^2 \|L_2(\theta - \bar{\theta})\|^2 \quad (3.14)$$

is added to  $r(c_\ell)$ . In this regularization term,  $\theta$  is the vector of sound wave speeds ( $c_\ell$ ) at the tomography voxels, and  $\bar{\theta}$  the vector of expected mean values of sound wave speed at each voxel. Those expected values are from samples of a population of interest, computed using normal distributions with given mean values and standard deviations for the breast and tumor radii and also varying the position of the tumor inside the breast; only a given fraction of the individuals has tumors. The breasts are always at the center of the vat. Care was taken to avoid biased results due to bad choice of random number generators<sup>18</sup>. In the present work,  $L_2$  is the inverse of a sampled covariance matrix  $I$ ; this regularization term tries to avoid big discrepancies, in regard to the values found in the population, of

---

<sup>18</sup>In [35, chap. 7.1] the authors alert against bad implementations of “random number” generators that are very common; these generators have “sequential correlations” in successive calls.

the  $\theta'$  obtained in the optimization process, as the problem is ill-posed.

The program which calculates the mean values  $\bar{\theta}$  also determines the covariances  $\Gamma$  for the same random sample of breasts, therefore a second regularization term can be added to  $r(c_\ell)$ ,

$$\gamma^2(\theta - \bar{\theta})\Gamma^{-1}(\theta - \bar{\theta}) \quad (3.15)$$

where  $\Gamma^{-1}$  is the inverse of the covariances, and  $\bar{\theta}$  the mean values. However,  $\Gamma$  has a big condition number and is difficult to invert, so it is modified as  $(\Gamma + \beta I)^{-1}$ ,  $\beta$  being “white noise” added to the covariances.

With the regularization terms, the objective function becomes

$$r(c_\ell) = (\phi_m - \phi(c_\ell))^T (\phi_m - \phi(c_\ell)) + \lambda^2 \|L_2(\theta - \bar{\theta})\|^2 + \gamma^2(\theta - \bar{\theta})(\Gamma + \beta I)^{-1}(\theta - \bar{\theta}) \quad (3.16)$$

and  $\lambda^2$  and  $\gamma^2$  are the weightings for the regularization terms. A set of faster simulations with various combinations of  $\gamma$  and  $\lambda$  was executed, with a lower voxel count, using a  $16 \times 16$  grid, 44 parameters to adjust, see table 7.

Note that equation (3.16) can be multiplied or divided by any non-zero value at any time, this way making its first term, the error from measurements to the values given by the tomography model, to appear always with unitary weighting<sup>19</sup>, with no loss of generality. However, the optimization program has parameters which affect the way this model error is calculated, and care must be taken to avoid this term to numerically vanish and become negligible; also, the optimizer from next section stops if the performance index (objective function) does not diminish below a given tolerance, therefore it made sense to weight the model error with values other than 1.

### 3.2.3 Search for the maximum *a posteriori* probability

The regularization acts to avoid the voxels to acquire undesired, unfeasible or unreal values. But the algorithm has to provide a mean for the values to evolve from the initial guess to a final one, where a tumor, if present in the breast, will be shown in its approximate position and size, according to the grid size chosen for the voxels. An optimizer will drive the choice of adequate voxel values, minimizing the objective function 3.16.

---

<sup>19</sup>The actual tomography program *has* a weighting value for the first term, the model error. This parameter was not removed when weightings for mean values were later added.

### 3.2.3.1 Selection of algorithms

Some algorithms were considered for this task: TOMP (Trajectory Optimization by Mathematical Programming, [56, 57]); VE08, from the Harwell Subroutine Library ([58]), good at non-linear unconstrained optimization problems; Subplex, subspace-searching simplex method for unconstrained optimization ([59]); BBVSCG, a variable storage Fortran subprogram for function minimization ([60]); lmdif, from the MINPACK library, which performs the Levenberg-Marquardt algorithm, with Jacobian calculated via finite differences ([61]); and even simulated annealing with simann ([62]).

Due to previous experience with the software TOMP, e. g. [63], solving a complex control problem where other methods, albeit carefully implemented from [64], were unsuccessful, it was chosen as optimizer.

While all the FEM simulation and the remaining of the inverse problem are C code; just the optimizer is Fortran. Mixed programming can easily be implemented with GNU compilers.

TOMP is divided in two main parts: one is a simulator (called d\_TOMP) and the other is the optimizer, which name is SLSQP, for Sequential Least Squares Programming; the simulator d\_TOMP is not used at all for the tomography algorithm.

**Optimization problem statement.** TOMP's optimizer SLSQP receives a vector  $x$  of all the optimization parameters, the gradient  $G$  of the cost (objective) function to be minimized, and the Jacobian  $J$  of constraint violations. For the purpose of the tomography optimization there are no constraint conditions, therefore  $J$  is 0 whenever it appears in the program. Also there are lower and upper bounds for the parameters. For the tomography, each parameter is simply the sound propagation speed to be optimized<sup>20</sup>.

Formally, the optimizer will solve the nonlinear programming problem

$$\min_{x \in R^n} f(x) \quad (3.17)$$

with each component of the vector  $x$  bounded by lower and upper limits

$$l_i \leq x_i \leq u_i, \quad i = 1, \dots, n \quad (3.18)$$

---

<sup>20</sup>As already stated, dissipation values can be used too.

and also satisfying constraints as

$$g_j(x) = 0, \quad j = 1, \dots, m_e \quad (3.19)$$

and

$$g_j(x) \geq 0, \quad j = m_e + 1, \dots, m \quad (3.20)$$

with  $f : R^n \rightarrow R^1$ . In our tomography problem  $m$  has value 0, and also  $m_e$ .

As the optimizer does not have to deal with constraint violations in the breast ultrasound tomography, its internal variables for  $m$  and  $m_e$  will have value 0, the subroutines involved with the constraints return immediately, and some loops in the program will not be executed. There is no extra simulation penalty, although some useless variables are still used in subroutine calls. In other words, SLSQP can be streamlined for use in this tomography algorithm, but the overall program performance may not improve at all.

The method for optimization is sequential quadratic programming, a efficient computational method to solve general nonlinear minimization problems, equation (3.17). A comprehensive explanation of the SLSQP algorithm is available in annex D.

## 4 RESULTS

The first simulations, done with the thorax model, show abundant echoes at the model borders. This artificial echo interferes with the transducers' readings, and because of that they must be diminished to acceptable levels in order to allow readings without extraneous signals.

The direct problem numerical simulator is compared to a set of simulations (from another simulator) which in turn were compared to physical laboratory experiments and agreed very precisely. This comparison addresses whether the tomography simulator is of an equivalent level of accuracy.

The tomography algorithm is applied to simple phantoms of the breast (represented by small cylinders) and its ability to indicate the position of a tumor is tested.

### 4.1 Development of an ALID model

A simple equation was devised for the ALID attenuation, with increase according to a power law, and a very subtle increment in the start of the layer. The ALID layer increases the size of the model, but is effective to avoid echoes.

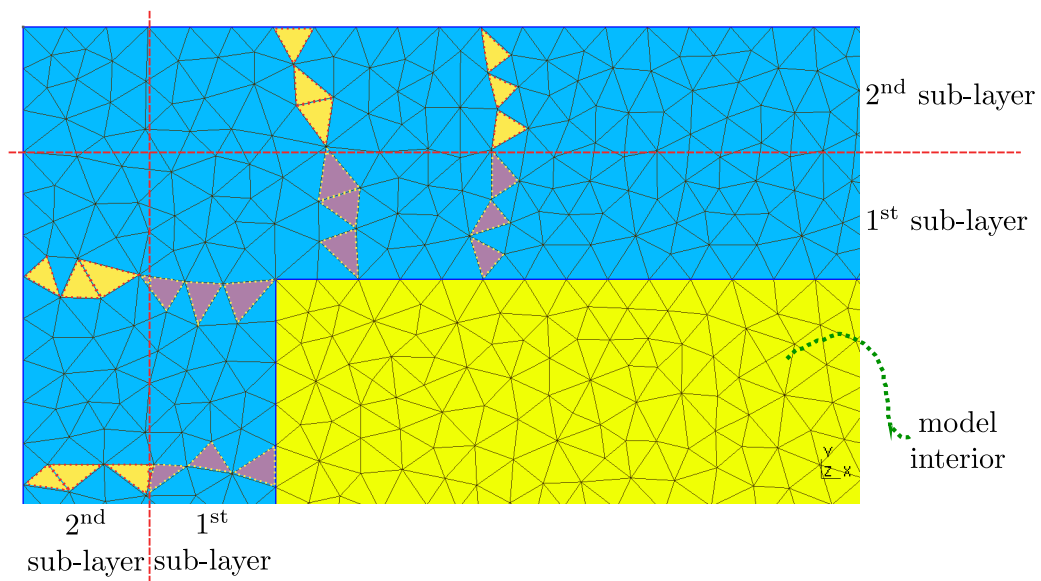
#### 4.1.1 ALID equation

In the simulations the ALID is indicated by a numeric code in the format  $N_1/N_2/N_3$  meaning that the layer is made up by  $N_1$  sub-layers, each one with a width of  $N_2$  elements (with their given "characteristic length") and power  $N_3$  for absorption increase, according to figure 13 and equation 4.1.

$$C_a(e) = C_{a_0} \times \left( 1 + \left( \frac{d(e)}{T_{sl}} \right)^P \right), \quad (4.1)$$

where  $C_a(e)$  is the attenuation coefficient of the finite element  $e$ ,  $C_{a_0}$  the water attenuation coefficient, which borders the ALID layer,  $d(e)$  is the distance from the start of the absorbing layer to the geometric center of the element  $e$ ,  $T_{sl}$  the thickness of one sub-layer of the ALID, given by the characteristic length<sup>1</sup>  $L$  of the elements and number  $N_2$  (in the direction of the thickness) of elements in the sub-layer, that is,  $T_{sl} = N_2 \times L$ , and  $P$  (parameter  $N_3$ ) the power for the attenuation increment.

**Figure 13** – Composition of an ALID layer “ $N_1/N_2/N_3$ ” of  $N_1$  sub-layers of  $N_2$  elements, here showing an ALID 2/3/3. Each one of the 2 sub-layers has approximately 3 elements (highlighted) or exactly 3 characteristic lengths in width.



#### 4.1.2 ALID sizes

Comparisons of the sizes and processing times for meshes with different ALID layers are in table 4.

In order to check the effectiveness of the ALID wrapping some meshes were created, without the thorax and internal organs, containing only water and the ALID layer, with a single point emitter in the geometric center of the model. The transducer readings file will show the occurrence of any echo and its intensity. The excitation is a single pulse.

<sup>1</sup>A parameter in Gmsh for the creation of the mesh.



**Table 4** – Sizes of the thorax simulation models with different ALID layers.

| model        | ALID wrapping thickness (m) | tetrahedrons total (comparison, %) | nodes total (comparison, %) | simulation time (s) |
|--------------|-----------------------------|------------------------------------|-----------------------------|---------------------|
| Qual553      | 0,077000                    | 3606232 (100%)                     | 624919 (100%)               | 25960               |
| Qual443      | 0,049280                    | 1974992 (54,77%)                   | 348688 (55,80%)             | 13245               |
| Qual343      | 0,036960                    | 1497299 (41,52%)                   | 265854 (42,54%)             | 9584                |
| Qual333      | 0,027720                    | 1149905 (31,89%)                   | 205710 (32,92%)             | 8029                |
| without ALID | absent                      | 276096 (7,66%)                     | 56239 (9,00%)               | 2164                |

Simulation times using an older, serial version of the simulation program.

## 4.2 Use of parallelism in the program

Parallelism technologies were developed for use with computationally intensive tasks. Problems involving finite elements and the generation of tomographic images can be considered computationally intensive, unless they have a small FE mesh as a model or a small domain for the tomography.

In this work models of different sizes were used: smaller models for testing and faster program development, and larger models for the tomographies. The performance of these various models allow for the elaboration of guidelines for the efficient use of parallelism, in general, and particularly for the problem of tomography using FEM.

### 4.2.1 Influence of the number of threads

OpenMP is “well suited for expressing loop-level parallelism” (cf. [21], 6.1 Scalability Challenges for OpenMP), but care must be taken with the nature of the problem to solve. As example, some tests were done in a “supercomputer” from CITI/USP<sup>2</sup>, with 48 threads available, and there is a clear trade-off where the number of processor cores (or threads) being used is optimal, and the use of all available resources can be counterproductive, due to the overhead of splitting, say, a `for` loop among all 48 threads, when the size of the mesh is not big enough to compensate for this overhead. In other words, the management of all the cores/threads takes an amount of time that is not compensated by the very short execution of the work by each core; by the brief moment each core finishes its small share of the `for` loop, the added time to start and then conclude the parallelization in

<sup>2</sup>CITI - Centro Interdisciplinar de Tecnologias Interativas (Cross-disciplinary Interactive Technologies Center) from USP.

all cores is, in proportion, too big. Illustrative results are available in appendix B for a medium to big FE model used when the simulation program was in development.

### 4.2.2 Parallelization for very small problems?

The same parallelization overhead that impacted the results shown in appendix B when a big number of threads is used<sup>3</sup> can be noticed, keeping now a fixed number of threads, but with problems of varying sizes.

Take a big problem, which will have long loop iterations, like a complex finite element mesh with some millions of elements. In this circumstance, more computer cores or threads lead to smaller execution times. The time in each loop is greater than the overhead, and more threads will reduce those loop times.

For very small problems, on the other hand, the overhead of preparing and starting a parallel region (a `for` loop) and, when it concludes, finishing this region and proceeding to another one or to serial code can easily be much larger than the time spent in a short duration loop with relatively few elements or vector variables to iterate. In extreme cases, a serial program will have better performance than a parallel one.

Table 5 shows results for some mesh sizes; when there are less than 100,000 tetrahedrons the serial version of the program is faster than the parallelized program.

---

<sup>3</sup>How “big” or “good” this number is depends and varies in proportion to the amount of calculations to do, or with the size of the FE model.

**Table 5** – Execution times for meshes of various sizes, serial and parallelized programs. Computers with 8 and with 12 threads for the parallelized programs. In **bold face** the fastest program.

| CPU i7-3610QM, 8 threads    CPU i7-8700K, 12 threads |                                     |                                     |
|--|-------------------------------------|-------------------------------------|
| mesh name  |                                     |                                     |
| size (tetrahedrons/nodes)                            | execution times (s)                 | execution times (s)                 |
| integration steps                                    |                                     |                                     |
| cuba_2020B<br>5545240/907937<br>500                  | <b>parallel: 689</b><br>serial: 991 | <b>parallel: 379</b><br>serial: 581 |
| trans_ely_NG<br>477437/85742<br>5000                 | <b>parallel: 456</b><br>serial: 518 | <b>parallel: 246</b><br>serial: 306 |
| trEconc<br>94283/18311<br>10000                      | parallel: 186<br><b>serial: 161</b> | parallel: 101<br><b>serial: 88</b>  |
| MiniA<br>47623/9565<br>10000                         | parallel: 118<br><b>serial: 77</b>  | parallel: 62<br><b>serial: 45</b>   |
| MiniB<br>8825/1958<br>10000                          | parallel: 31<br><b>serial: 16</b>   | parallel: 18<br><b>serial: 9</b>    |
| MiniC<br>4576/1038<br>30000                          | parallel: 31<br><b>serial: 15</b>   | parallel: 17<br><b>serial: 8</b>    |

### 4.2.3 Parallelism using the GPU

Programs parallelized with OpenMP can, without much additional effort, be executed in the GPU. There are two drawbacks: data transfer from the computer RAM memory to the GPU memory is slow – and occurs twice: uploading data to be processed in the GPU and later downloading the results to the RAM – and there is no input/output capabilities in the GPU for reading or writing files.

There are OpenMP directives which allow data to be uploaded to the GPU and then remain there for repeated use later on, reducing the amount of data (and time) to upload

in later calls of the function using the GPU. In the FEM simulation, the GPU is used for the multiplication of a matrix by a vector; the matrix elements are constant values (as long as the model properties do not change), uploaded only once to the GPU, and the vector is the only data which is uploaded at every function call; the result of the multiplication is always downloaded to the computer RAM.

In the same way the use of all available CPU cores may be counterproductive, the use of the GPU for parallelism can not be justified unless the problem at hand is big enough, because the time for upload/download of data between CPU and GPU may be dominant and surpass any parallelism gains.

We did limited testing of GPU parallelism, in the machine with the CPU i7-8700K CPU (12 threads), with a NVIDIA RTX 2070 GPU. This GPU model has 40 “streaming processors” with 64 cores each, for a total of 2560 “shader processors”<sup>4</sup>. With this hardware, and only the matrix times vector operation using the GPU, a simulation with a 4,592,407 tetrahedrons mesh took 444 s. using only CPU parallelism, and 357 s., almost 20% faster, using CPU+GPU. At that time the decision to use much smaller meshes was done, and in this situation the use of the GPU would not be beneficial.

OpenMP allows the use of NVIDIA and newer Radeon GPUs for parallelism. This is a positive point over CUDA, which works only with NVIDIA GPUs.

### 4.3 Simulation results for the thorax model

Results for thorax simulations are presented in this section. These results show how echoes manifest in the model, and how an adequate ALID layer, wrapping the model at all sides, is effective in attenuating echoes.

All simulations are done for a 50000 Hz ultrasound, of only one pulse, modulated by a Gaussian curve. The integration time step was  $1.0 \times 10^{-8}$  s, with a total of 12,500 steps for models similar to the ones solved by k-Wave and 30,000 steps for models used for echo evaluation. Models for echo evaluation do not have thorax contours nor the internal organs, being a simple closed volume of water with the ALID wrapping around and a single point transducer in the geometric center of the model.

The thorax model represents one model used by Cárdenas [39], for use in the D-bar tomography ([65]). The simulations, unless otherwise specified, are made for only the first

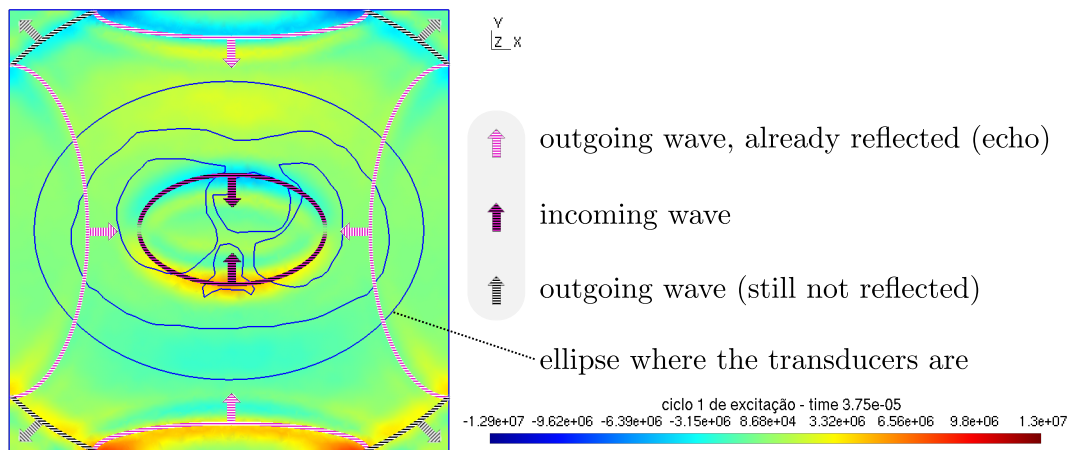
---

<sup>4</sup>However, not all “processors” are equally capable, e. g. only a few of them can work with transcendental functions. This is a general purpose gaming GPU, not one designed for heavy data mining.

excitation pattern, that is, with  $n = 1$  in equation 3.8, with amplitudes proportional to  $\cos(\theta)$ .

The first simulation is of a model without ALID wrapping. In this case echoes are abundant and can be easily recognized, see figures 14 and 15. There are 64 transducers, in an elliptic arrangement (figure 8 on page 48) and angularly equi-spaced.

**Figure 14** – Visual identification of echo at the model borders. Single pulse excitation in the 64 transducers with amplitude according to  $\cos(\theta)$ . Pressure in Pa.



In figure 15 there is a selection of some frames recorded in the visualization result file, which helps to understand how the sound propagation evolved.

With an ALID absorbing layer as is described in section 3.1.3 it is clear that echoes at the model borders are eliminated or at least greatly diminished, as can be seen in figures 16 and 17, where the figures include the ALID layer and, because of that, figures appear to have a smaller region of interest (transducers, water, thorax and internal organs) than in figure 15; however the dimensions are exactly the same. All the apparent size variation in the following figures is due to varying ALID layer thicknesses, see table 4 on page 64.

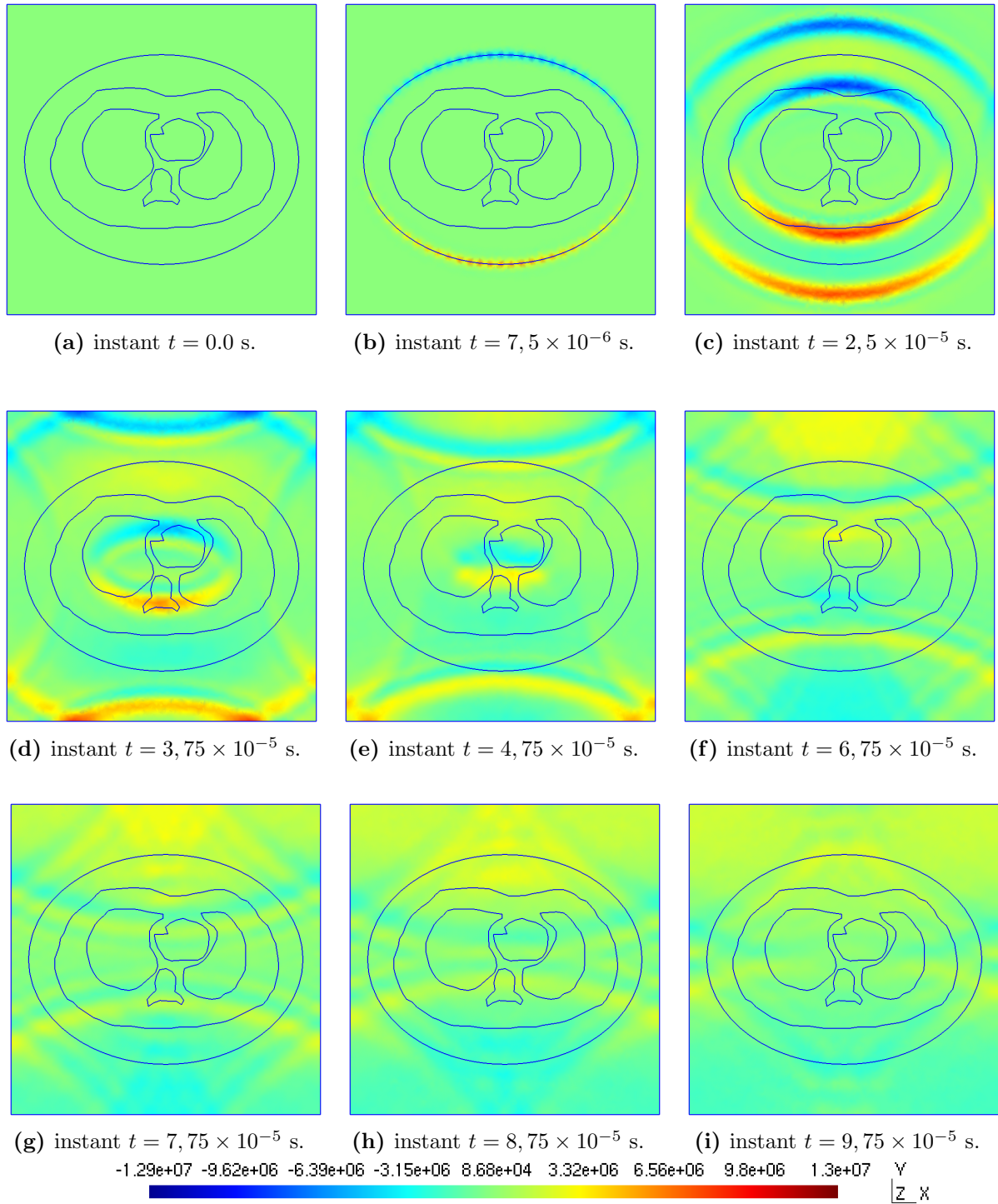
One objection that can be made is that, in the thorax simulation models, the transducers are numerous, and are in an ellipse, at a distance from the center and closer to the model lateral borders. In the ALID evaluation models it is used a single transducer, in the center point of the model, therefore the total sound intensity reaching the ALID layer is smaller in this situation.

One echo evaluation result with an ALID 6/2/4 is shown in figure 17, where the value scale is adjusted to encompass the minimum and maximum values of the depicted instant in time.

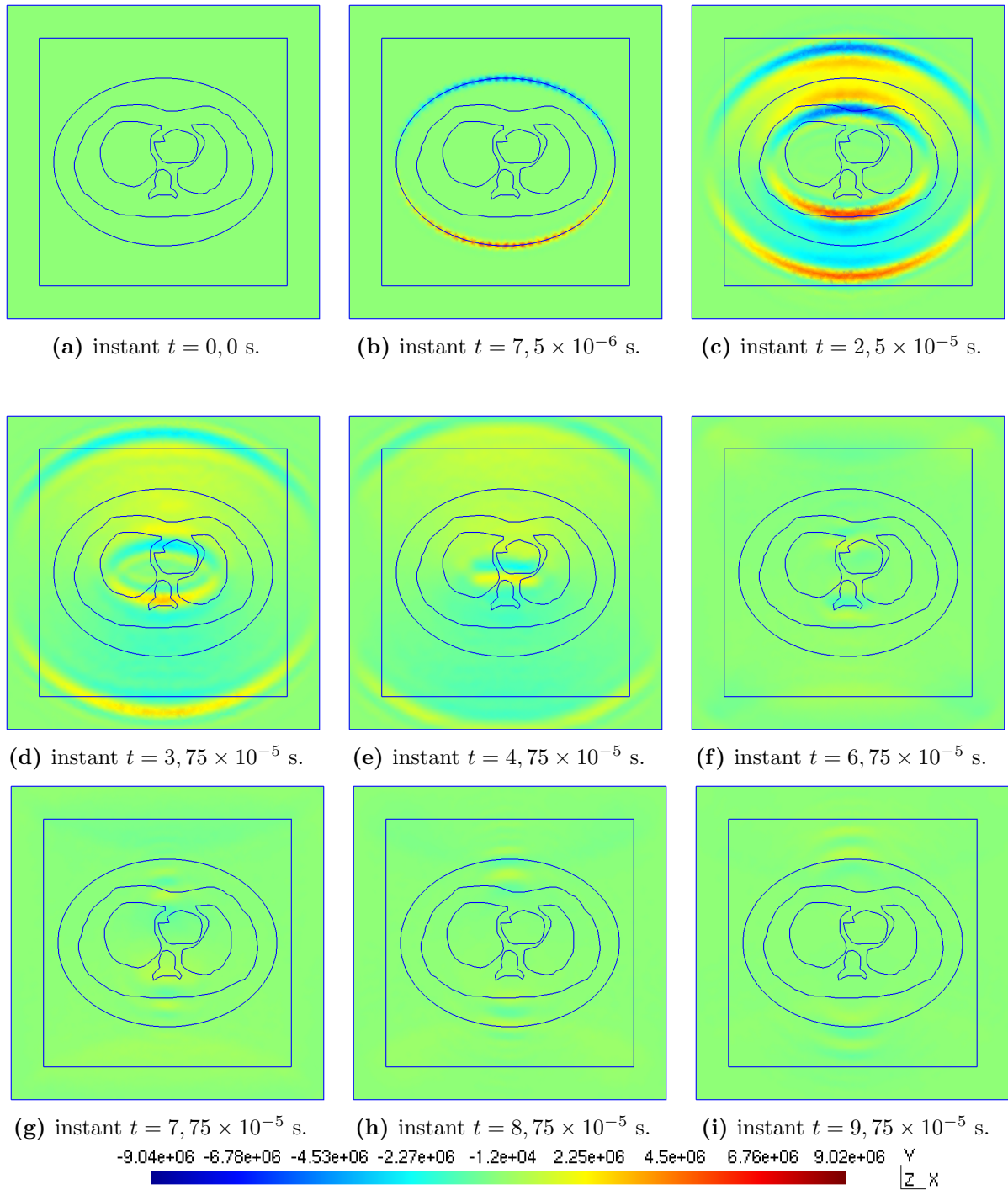
There is no echo, and the waves have their intensity vastly reduced inside the absorbing

layer. *However*, with the same ALID layer 6/2/4, but in the thorax simulation, with all the transducers emitting, still some echo can be noticed, as is shown in the sequence of figure 18; for the increased power from many emitters the ALID *must* be bigger.

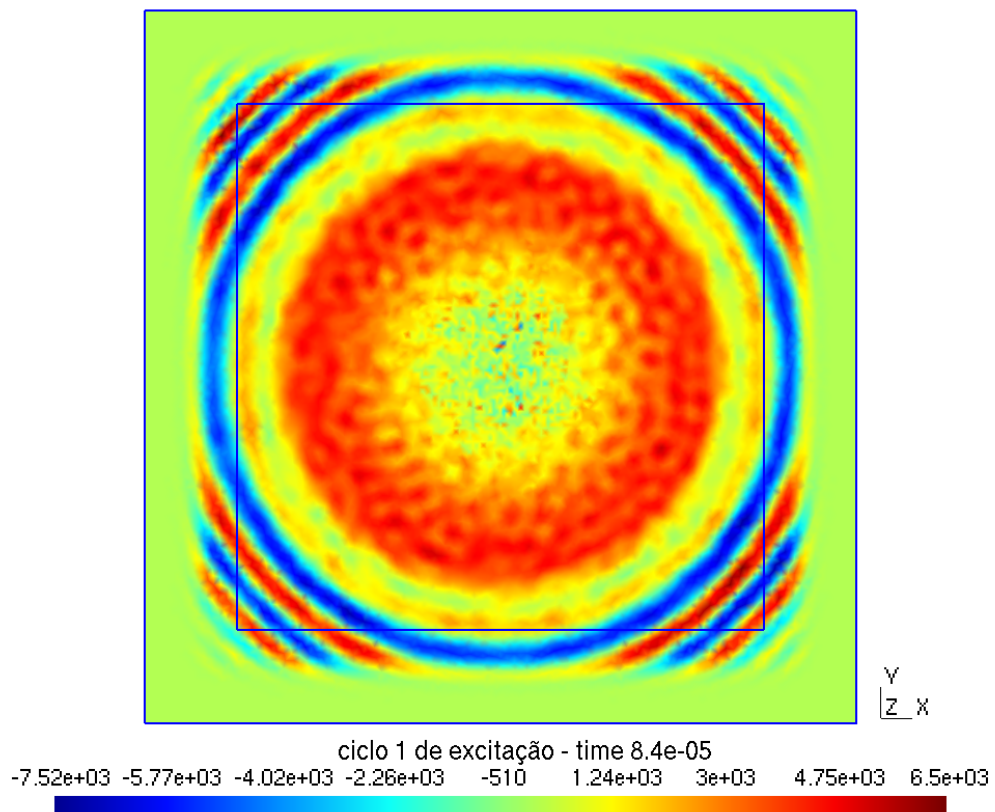
**Figure 15** – Evolution of an excitation pulse generating echoes. Model without ALID layer, pressure in Pa.



**Figure 16** – Temporal evolution of the excitation pulse with an ALID layer 3/3/3. Pressure in Pa.

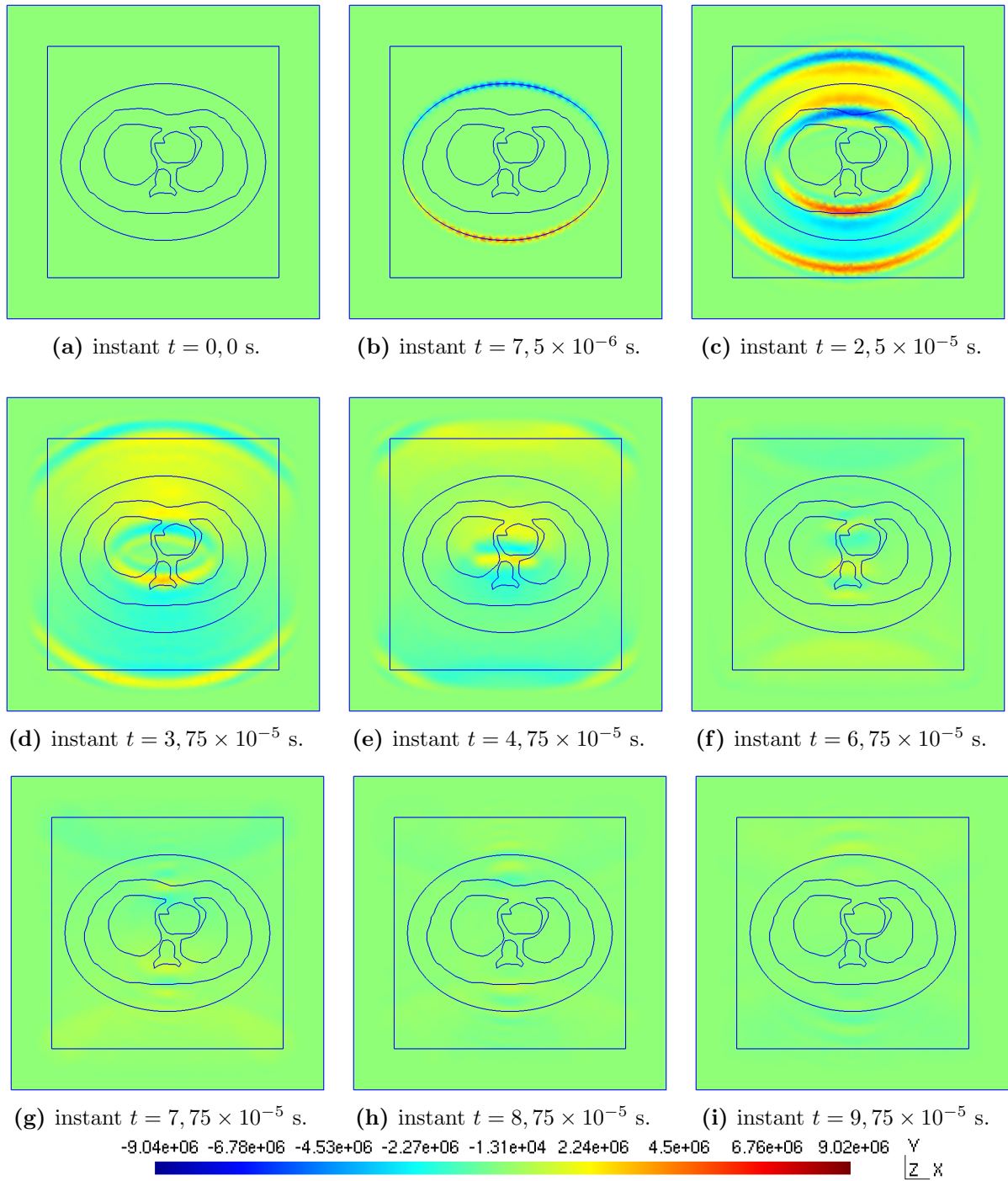


**Figure 17** – Behavior of the echo model, ALID 6/2/4, instant  $t = 8,4 \times 10^{-5}$  s. Pressure in Pa.





**Figure 18** – Temporal evolution of the excitation pulse with an ALID layer 6/2/4. Pressure in Pa.

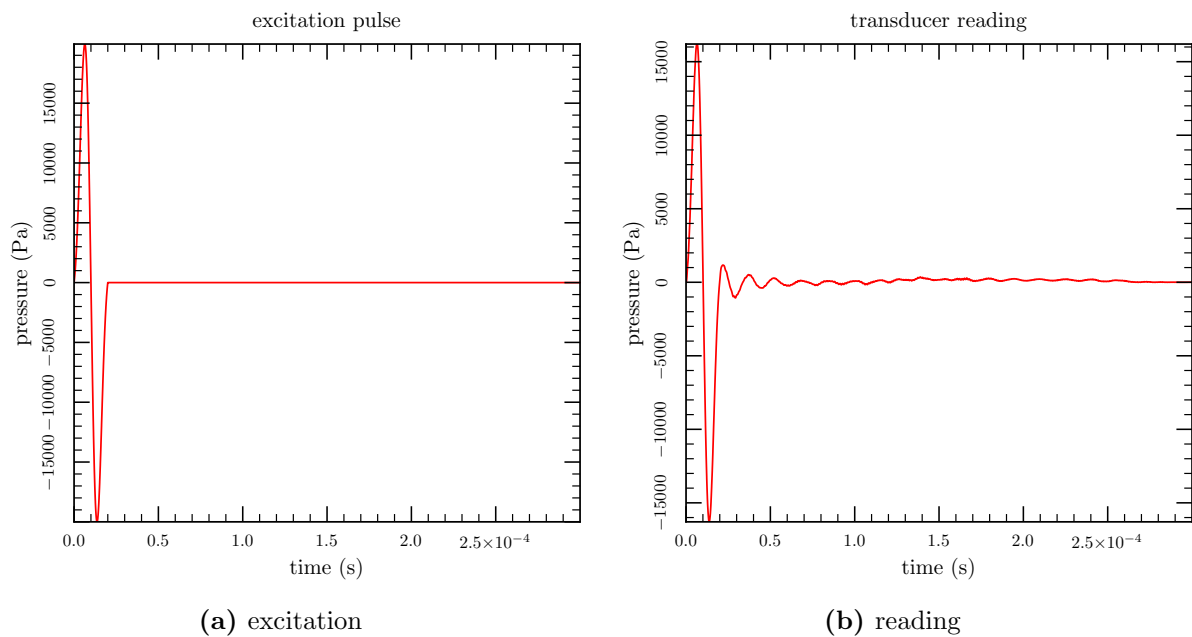


### 4.3.1 Effectiveness of the ALID layer

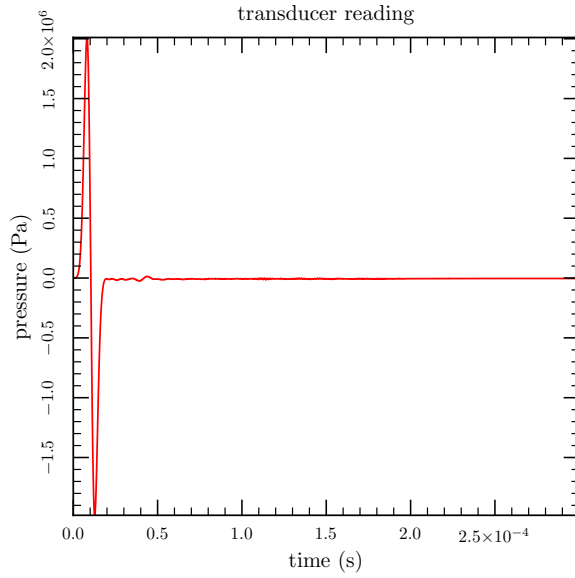
The echo also appears very clearly in the reading signal from the echo evaluation model, without ALID, figure 19, where it is shown in 19b the excitation pulse followed by a series of small waves – which are not seen in simulations with ALID layers. These disturbances, however, are not echoes from the far borders, because they appear already at the moment the excitation ends, at instant  $t = 2 \times 10^{-5}$  s, whereas the first echo signal from the borders, due the size of the model, should only appear at the time  $t = 1,3636 \times 10^{-4}$  s. Such disturbances are due to echoes between the planes that define the thorax slice, as shown in appendix C.

With the use of an ALID 3D wrapping the measured sound pressures is reduced. Keeping the same power for the increase in the absorption, figures 20a-20c show how a thicker ALID avoids echoes.

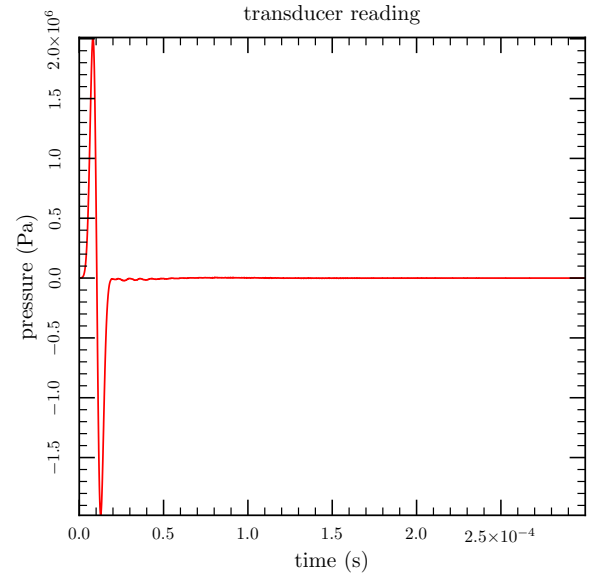
**Figure 19** – Excitation and reading with initial pulse and echo signal. Model for echo evaluation, *without* ALID layer.



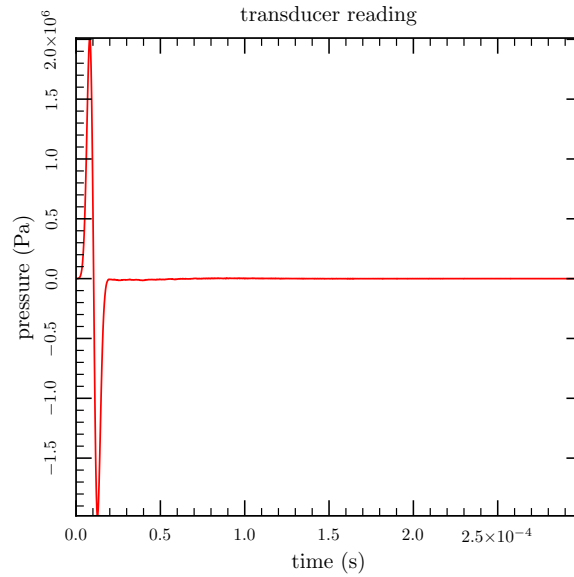
**Figure 20** – Readings with echo attenuated by the use of an ALID wrapping. Same conditions of figure 19 except for the different ALIDs.



(a) ALID 2/2/3



(b) ALID 4/4/3



(c) ALID 5/5/3

## 4.4 Realistic breast

One breast mesh was developed using a STL tumor with 18,498 polygons and 18,580 vertices, shown in figure 21 ([66]) and a breast extracted from a human character created with the software MakeHuman ([67]).

**Figure 21** – Tumor for the realistic model, described by a STL file, from [66].



Source: [66].

A series of operations had to be performed in order to obtain the final model: **a)** as can be seen in figure 21, the tumor has veins and arterioles of no value for our application, and the fact that they are opened at their extremities will provoke errors when trying to create a (closed) volume in Gmsh, and for this reason they have to be removed, **b)** no character in MakeHuman was found or could be created, to our very limited knowledge, in the desired position, with the breast hanging below the horizontally laid torso; because of that, when an individual in an acrobatic posture – named “Gym01” – was found it was assumed the best that could get at the time; this character was exported as a .dae (Collada) mesh, **c)** this character had to be rotated from its default position, in order to make the breast point vertically downwards, using Blender, **d)** the whole character was combined with simpler volumes (cubes, cylinders, etc.) and operations of subtractions and intersections made in order to isolate one of the breasts; for this the program Blender ([68]) was used, **e)** the tumor had to be scaled down, and then positioned inside the breast, again using Blender, **f)** the isolated breast and the tumor are then saved, each one

in an independent file, **g**) an script/program was created and executed in Octave ([44]) to create a vat, with the chosen dimensions and number of transducers as parameters, and this information saved in a geometry file for Gmsh, **h**) a method was devised in Gmsh to allow that the STL files of the breast and the tumor were loaded and combined with the vat geometry, and finally a FE mesh was generated, while preserving the ability to access a water volume, the breast and the tumor independently, **i**) as the resulting FE mesh had bad shaped tetrahedrons (Gmsh displays an alert when the mesh just created has these problems), simplifications, corrections and improvements to the breast and tumor STL files were done using Meshlab<sup>5</sup> ([69]) and step **h**) redone, iteratively, until a good FEM mesh was obtained with Gmsh, with a small number of bad elements<sup>6</sup>.

An adequate balance has to be found between finer surface meshes, for the breast and for the tumor, and coarser ones. Obviously a fine mesh has a large number of elements and requires more computer power or time to simulate, and a coarse mesh would not work well to simulate the wave phenomena (see [42]).

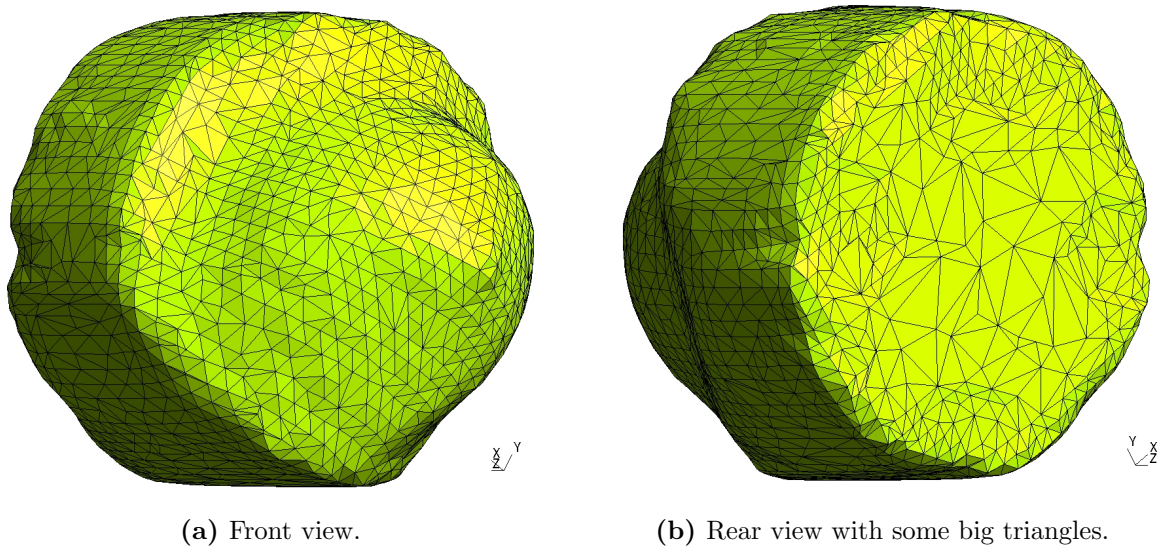
Other problems appeared too: in figure 22 a big difference in mesh size exists in the breast. The original MakeHuman surface, which corresponds to the skin of the breast, has small or (say) adequate size, while the surfaces for the “cuts” done with simple geometric volumes in order to isolate the breast from the whole body have a very coarse mesh, maybe because of the more regular surface and curvatures of the volumes. As the surface mesh drives the creation of volumetric elements in Gmsh, the resulting tetrahedral mesh is of bad quality. Also, some triangles from the STL surfaces are very pointy, and this also causes difficulties as a low quality mesh and later numerical problems in the finite element program.

---

<sup>5</sup>Several resampling and remeshing algorithms were used: Catmull-Clark, Butterfly subdivision, TwoStep smoothing, etc, to improve the breast and tumor meshes.

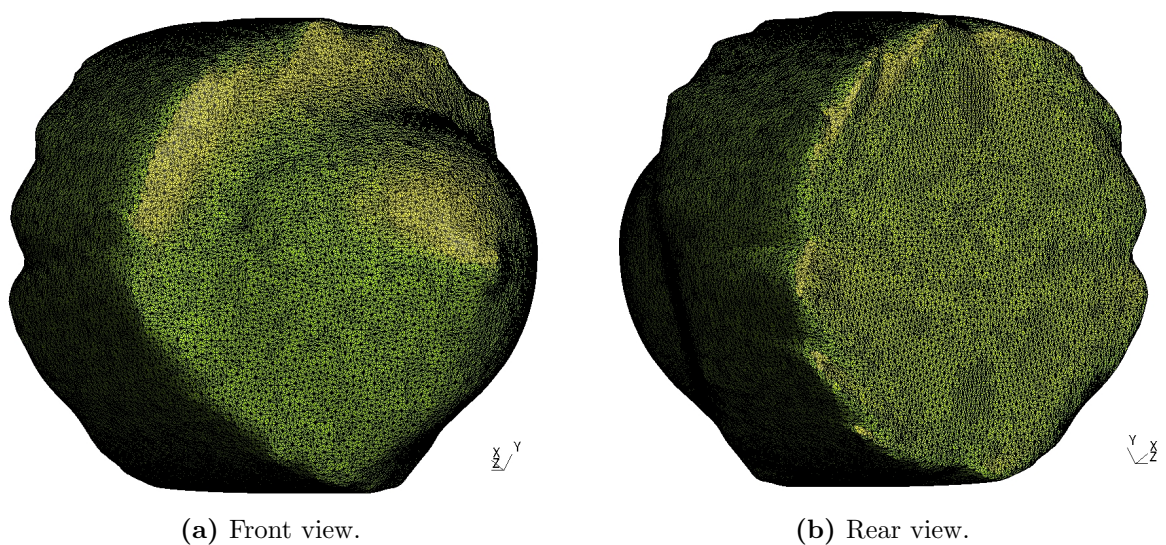
<sup>6</sup>A background field was used in Gmsh, however this could not correctly drive the element sizes near the boundaries defined by the STL triangles; in attempts with B-rep files for the breast and tumor the background field worked better, but then it was impossible to correctly subtract volumes and there was water and breast volumes sharing the same region in space, at the same time.

**Figure 22** – Big variation in the size of surface triangles in one of the realistic breasts. The mesh file for the complete model using this breast, with water and ALID, has 1,735,789 elements.



A more regular surface mesh can be seen in figure 23, obtained with different remeshing algorithms applied to the same starting stereo-lithography file from figure 22; however, the resulting 3D mesh created in Gmsh, even after all the optimization stages, still has bad tetrahedrons, as shown in table 6; the breast used in mesh cuba2020 is the one in figure 22, and from mesh cuba\_2020B is in figure 23.

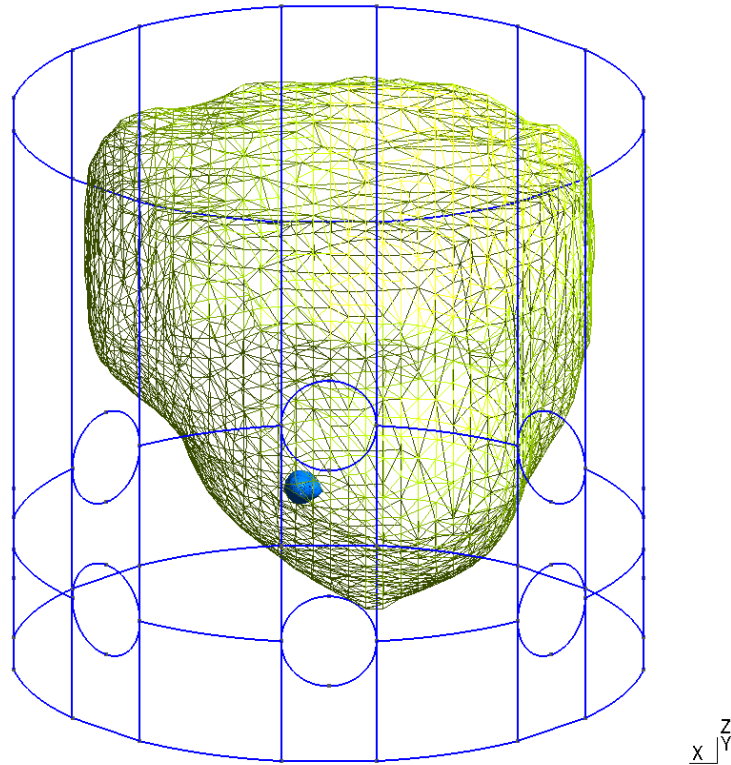
**Figure 23** – A more uniform size of surface triangles in another of the realistic breasts. With this breast, the complete mesh file (with water and ALID) has 5,881,631 elements.



In figure 24 one of the breasts models can be seen inside his vat, slightly from below the level  $Z = 0$ . Breast and tumor are shown as surfaces made up from many triangles –

that is the information their STL files contains – and only the triangles’ edges were drawn in this figure.

**Figure 24** – Wireframe geometry depicting cuba\_2020 breast in the vat, viewed from below. A bluish blob inside the breast is the tumor.



**Table 6** – Properties of some of the realistic breast meshes.

| mesh file  | tetrahedrons |            |
|------------|--------------|------------|
|            | total        | ill-shaped |
| cuba2020   | 1735789      | 12         |
| cuba_2020B | 5881631      | 61         |

Numbers with Gmsh 4.8.4, may vary with other versions.

MakeHuman, according to Wikipedia, has been used in industrial design, to verify the anthropometry of a project, and in virtual reality research, to quickly produce avatars from measures or camera views; MakeHuman characters are used in biomechanics and biomedical engineering, to simulate the behavior of the human body under certain conditions or treatments; full-body 3D virtual reconstructions have been performed using MakeHuman, and 3D analysis of early Christian burials, among other applications ([70]).

## 4.5 Simulations and experimental data

In section 4.3 some results from the simulator were already shown. In figures 14 (page 68) and 15 the echoes can be clearly seen, because of the absence of an ALID layer. Also an ALID was developed, with a damping according to equation 4.1. After a good set of values for the parameters  $N_1/N_2/N_3$  was established empirically the echoes are effectively dissipated, as can be seen in figure 18 (on page 72).

### 4.5.1 Numerical comparison

More simulations were done with the simulation model (direct problem) and these results compared to experimental data available in the work of Nakanishi ([71]), where the author used a pair of know transducers to emit a pulse and receive it at various controlled distances from the emitter, both transducers in a water tank. The complete tests were repeated 16 times and the mean values obtained then used to ascertain the precision of a computer simulation of the problem, made with finite elements.

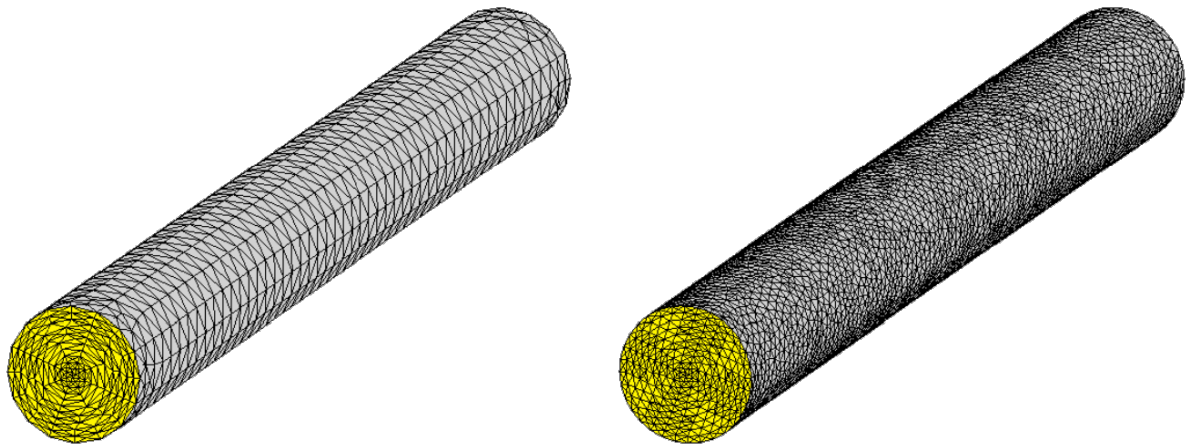
It should be noted that the mesh used for comparisons was not the same from [71]; there, by the figures, the mesh seems to be somehow *structured* – see figure 25 – as some points of measurement are equally and regularly spaced along the length of the cylinder, and any section of the mesh, between two consecutive measurement points, had a visual pattern which repeat itself for the other segments between measurement points; this is more clearly visible in the coarser mesh with 5 points per wavelength. Much more noticeable, however, even for the finer meshes (10, 20 and 30 points per wavelength) are the concentric layers of attenuating elements used to avoid echoes, as Nakanishi created “tubes” of PML (perfectly matching layer) around the central cylinder where he simulated sound propagating in pure water. In a PML, damping properties are a function of distance, therefore he calculated those properties for thin cylindrical shells.

The shape of the excitation signal in [71] is also different from what was used before in our simulations with thorax sections. New parameters were tested until the excitation used in the test resembled Nakanishi’s one.

In addition to cylindrical models, [71] also includes comparisons with a spherical mesh. The use of PML elements is not necessary in this mesh. Comparing his computer results with experimental data the agreement is very good, with greatest error of about 6%, explained by differences in the sound propagation speed, exact shape of the real excitation signal, multiple wave reflections between emitter and sensor transducers in the



**Figure 25** – Structured meshes used in comparisons between simulated and experimental data.



(a) Mesh with 5 elements per wavelength.

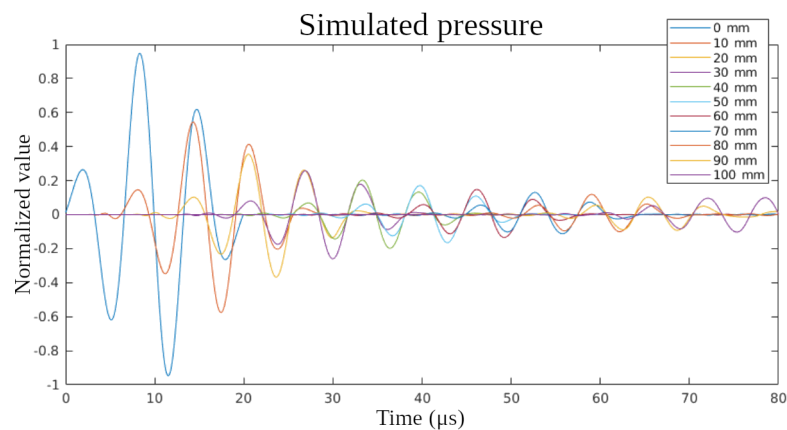
(b) Mesh with 5 elements per wavelength.

Source: [71].

experimental setup, among other factors. Most of the position measurements have less than 1% errors. Nakanishi used the mean of 16 experimental samples, and his values were normalized by dividing all the signals by the peak value of each experiment and this is what was compared to his simulations.

Figure 26 shows the simulated measurements for the spherical model from [71], and figure 27 shows numerically simulated readings for the corresponding positions computed by the FEM direct problem simulator. In both situations, the present work and Nakanishi's work, the excitation signal is applied by a disc surface at the center of the sphere.

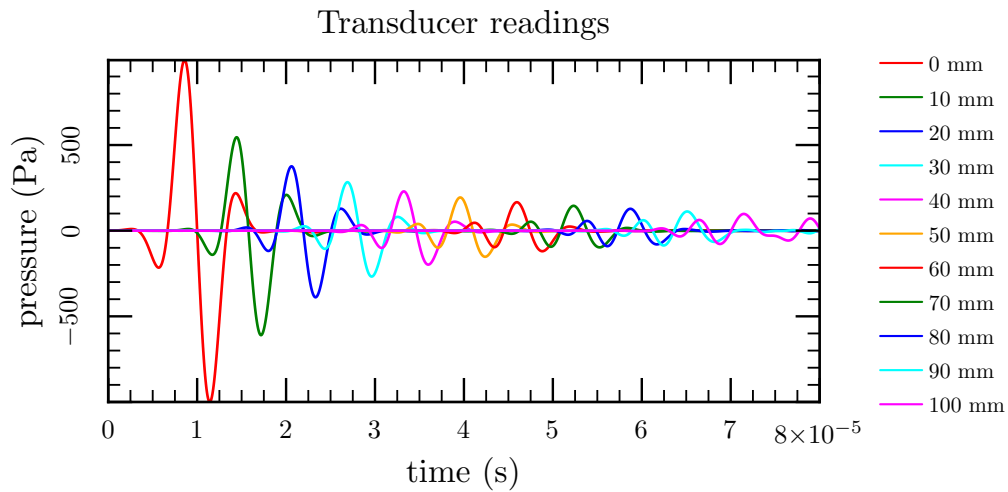
**Figure 26** – Pressures at various distances from the center of a sphere, simulated by Nakanishi. Emitter at the center of the sphere (position 0 mm).



Source: modified from [71].

The excitation signal in [71] is a 150 kHz pulse made with 3 cycles of a sine wave, attenuated at the beginning and end of the pulse. However, comparing figures 26 and

**Figure 27** – Pressures at various distances from the center of a sphere with the simulator used for the tomography algorithm. Emitter at the center (position 0 mm).



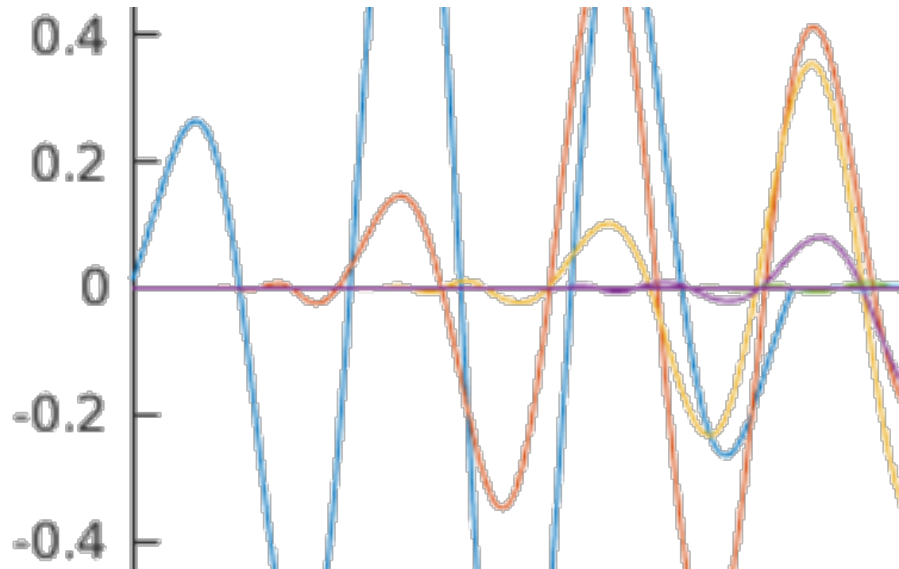
27 for the pressure at the center of the sphere, position 0 mm, where the excitation is applied to the surrounding water, the signal in [71] appears to lack enough attenuation for a very smooth transition from steady state at rest (constant zero pressure) to the pulse, see the detail in figure 28. The pulse used in the tomography simulator has a much more attenuated transition from and to steady state – around half of a complete sine disappears. This signal is a sine wave attenuated by a Gaussian curve. Indeed in [71] the author wrote about the same convolution of a sine wave with a Gaussian, and shows a graph much closer<sup>7</sup> to the excitation in figure 27; however, this is *not* the signal that can be seen in figure 26 or figure 28.

Despite this difference in the excitation, the pressure peaks at each position in both figures 26 and 27 are comparable, and the decay seems a little greater than the one computed by the direct problem in the present work. A discrepancy in the ultrasound dissipation properties may justify this, and will be explained in chapter 5.

Since the numerical simulator used in the tomography algorithm generated results comparable to those of Nakanishi in [71], which in turn agree with experimental data, *indirectly*, it is presumed that the direct problem represents, up to a certain precision, results of the real world. Further investigation is needed to better assess the quality of the simulator.

<sup>7</sup>It is almost identical, but the normalized peak value is attenuated by the Gaussian and reaches about 0.9 instead of 1.0.

**Figure 28** – Excitation pulse (blue line) with 3 sinusoidal cycles and sharp transition from and to steady state. Enlarged detail of figure 26.



Source: [71].

## 4.6 Tomography grids for faster results

During the development of the tomography algorithm, even small models intended for testing the program required significant processing time, which hindered debugging and testing. To reduce program run times, some schemes were devised for creating tomography grids that allow a given number of voxels to have fewer optimization parameters, resulting in faster results.

Three types of voxel grid were used: **a)** *cylindrical*, the first created, named after the cylindrical aspect of the simpler breast model. In this grid, every voxel can be optimized independently of the others; **b)** *rings*, where the water filled region around the breast is divided into a desired number of rings, of equal width, and all the voxels corresponding to the same ring will have the same properties to be optimized; this way not only the water voxels will have, more consistent properties (at least less prone to variations from voxel to voxel) but, more importantly, the problem will have less optimization parameters – this is the first grid type intended to speed up the program; **c)** *rings and sectors*, where, besides the water rings from **b)**, the breast is divided in four quadrants, and, as the tumor is placed somewhere inside the fourth quadrant, the three first ones will have common properties for the voxels inside them, and there will be much less optimization parameters for the program to make computations with, therefore the running time will be further reduced. In figures 29 and 30 the optimization indexes and voxel numbers are shown for a rings and sectors grid with 16x16 voxel grid and 4 rings.

**Figure 29** – Sample of a “rings and sectors” grid of voxels for a 16x16 grid with 4 rings. This is the file with the optimization indices for each voxel. Note the repeated index numbers 1–4 for voxels in each of the 4 rings, 5–7 for the voxels in each of the first three sectors and the individual voxels with indexes 8 to 44 in the fourth sector. The index 0 is for ALID voxels, which are excluded from the optimization.

```

1 Z=0 - gerado por makegrid 3.6 (2023-07-10T15:05) - geometria QA - cilindrica com quadrantes e aneis. Data: 19/07/2023 - 12h24m09s em avell-b152
2 0 0 0 0 0 1 2 2 2 2 1 0 0 0 0
3 0 0 0 1 2 4 5 5 6 6 4 2 1 0 0
4 0 0 1 3 5 5 5 5 6 6 6 6 3 1 0
5 0 1 3 5 5 5 5 5 6 6 6 6 3 1 0
6 0 2 5 5 5 5 5 5 6 6 6 6 6 2 0
7 1 4 5 5 5 5 5 5 6 6 6 6 6 4 1
8 2 5 5 5 5 5 5 5 6 6 6 6 6 6 2
9 2 5 5 5 5 5 5 5 6 6 6 6 6 6 2
10 2 7 7 7 7 7 7 7 8 9 10 11 12 13 14
11 2 7 7 7 7 7 7 7 15 16 17 18 19 20 21
12 1 4 7 7 7 7 7 7 22 23 24 25 26 27 4
13 0 2 7 7 7 7 7 7 28 29 30 31 32 33 2
14 0 1 3 7 7 7 7 7 34 35 36 37 38 3 1
15 0 0 1 3 7 7 7 7 39 40 41 42 3 1 0
16 0 0 0 1 2 4 7 7 43 44 4 2 1 0 0
17 0 0 0 0 0 1 2 2 2 2 1 0 0 0 0

```

**Figure 30** – Voxel numbers in the same “rings and sectors” grid of voxels with a 16x16 grid and 4 rings from figure 29. There are 208 voxels in this domain. The number 0 is shared among ALID voxels.

```

1 Z=0 - gerado por makegrid 3.6 (2023-07-10T15:05) - geometria QA - cilindrica com quadrantes e aneis. Data: 19/07/2023 - 12h24m09s em avell-b152
2 0 0 0 0 0 1 2 3 4 5 6 0 0 0 0
3 0 0 0 7 8 9 10 11 12 13 14 15 16 0 0
4 0 0 17 18 19 20 21 22 23 24 25 26 27 28 0
5 0 29 30 31 32 33 34 35 36 37 38 39 40 41 42
6 0 43 44 45 46 47 48 49 50 51 52 53 54 55 56
7 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
8 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88
9 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104
10 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
11 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136
12 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152
13 0 153 154 155 156 157 158 159 160 161 162 163 164 165 166 0
14 0 167 168 169 170 171 172 173 174 175 176 177 178 179 180 0
15 0 0 181 182 183 184 185 186 187 188 189 190 191 192 0 0
16 0 0 0 193 194 195 196 197 198 199 200 201 202 0 0 0
17 0 0 0 0 0 203 204 205 206 207 208 0 0 0 0 0

```

To illustrate how much those types of voxel grids or patterns can reduce the number of optimization parameters, table 7 shows the effect of the three grid types in the total number of variables to optimize.

**Table 7** – Influence of the grid type in the number of optimization parameters. Type C is “cylindrical”, R is “rings” and RS “rings with sectors” – see description in the text.

| model characteristics |         |       | parameters |     |     | voxels in<br>breast |
|-----------------------|---------|-------|------------|-----|-----|---------------------|
| diameters (m)         | XY grid | rings | grid type  |     |     |                     |
|                       |         |       | C          | R   | RS  |                     |
|                       | 30 x 30 | 5     | 716        | 505 | 133 | 500                 |
| vat 0.10              | 24 x 24 | 5     | 448        | 321 | 87  | 316                 |
| breast 0.084          | 20 x 20 | 4     | 316        | 220 | 61  | 216                 |
|                       | 16 x 16 | 4     | 208        | 152 | 44  | 148                 |
|                       | 20 x 20 | 4     | 316        | 128 | 38  | 124                 |
|                       | 16 x 16 | 4     | 208        | 84  | 27  | 80                  |
| vat 0.070             | 14 x 14 | 4     | 156        | 64  | 22  | 60                  |
| breast 0.045          | 12 x 12 | 4     | 112        | 56  | 20  | 52                  |
|                       | 10 x 10 | 4     | 80         | 36  | 15  | 32                  |
|                       | 8 x 8   | 2     | 52         | 26  | 11  | 24                  |

Number of rings not applicable to the “C” grid.

The larger vat, with 0.10 m diameter, is being used to try to obtain a tomography. As some values that influence the optimization algorithm and the regularization need to be adjusted, the lower number of parameters to optimize in a coarser grid allows for faster tests; later, with ideal values for the optimization algorithm and regularization, more refined grids could be used again. Those tests are run in a relatively fast desktop computer.

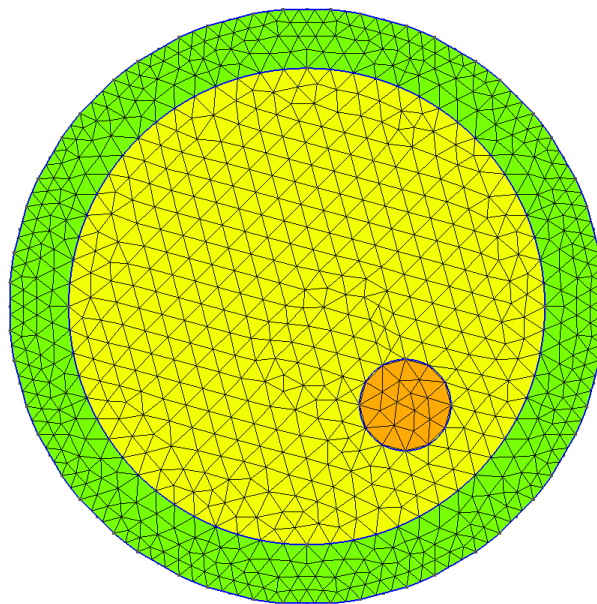
The smaller vat, which corresponds to the lower part of table 7, has a much smaller number of parameters to be optimized, linked to coarser voxel grids. Those are not intended for any useful tomography, but just for the fastest possible execution times of the whole program, allowing program development, testing and debugging in a more efficient way.

## 4.7 Tomography results

The UST algorithm was used to perform the optimizations to the voxel values and generate an image with the voxels depicting the breast with a tumor. The direct problem numerical simulation using the mesh shown in figure 31 generated the results used as reference signals for the tomography – they are used as surrogate measurements; later the diameter of the vat and height were modified and similar reference signals were used, with variations in size (vat and breast diameters) and in the size and position of the tumor.

In the statistical solution of inverse problems, regularization parameters come from the data; in the present work, that is exploratory and approximation errors are not investigated, the question is whether there is a set of parameters for the inverse problem using the FEM.

**Figure 31** – Finite element mesh used as a reference model, with regions of water, breast and tumor in top view. ALID elements not shown. Actual sizes may vary slightly according to each test done.



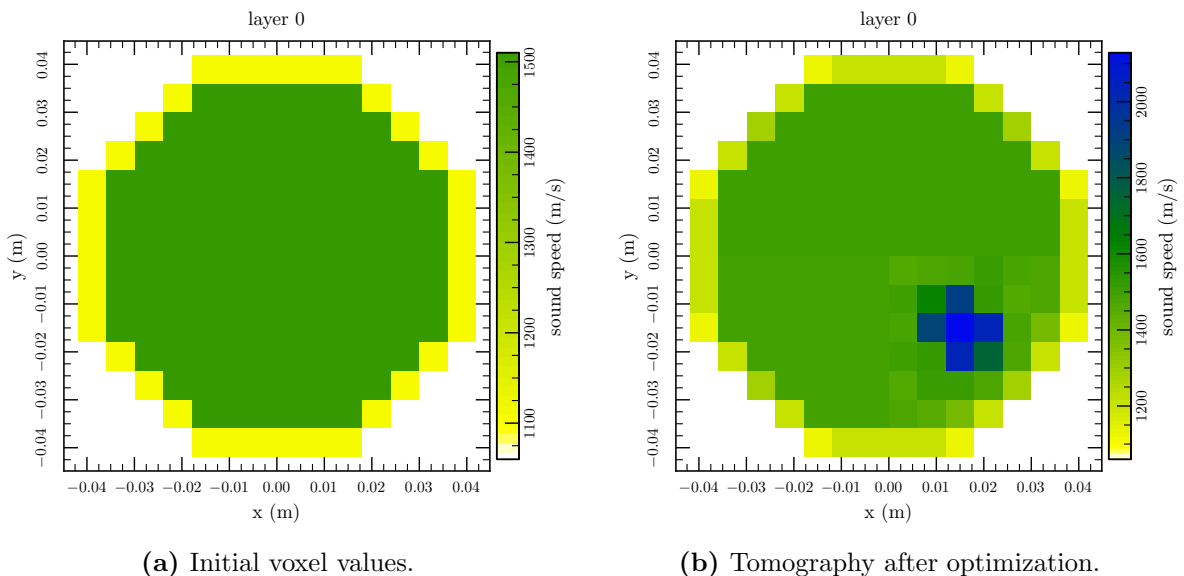
Tests were performed for the selection of adequate values for the regularization parameters (weightings). These tests were done in a model with the same size as the simplified breast model (see page 51), however with a smaller number of voxels in order to speed up the tests. Initially only 6 transducers were used. The diameter is divided in 14 parts in the direction of the  $x$  and of the  $y$  axes, and with 3 “rings” for the water around the

breast, which gives 156 voxels<sup>8</sup>. The result below is for a vat of diameter 8.4 cm, initial values considering separation between water and breast at radius 3.75 cm, and the reference model has a breast 7.5 cm in diameter and a tumor 1.4396 cm in diameter; the domain is 1.75 cm in height. The grid is of type RS (see item **c**) on page 82, which gives 30 voxels in the fourth quadrant (sector) – where the tumor must be found – but only 36 optimization parameters for the whole domain. The contrast was increased, with sound propagation speeds shown in table 8. In figure 32 there are the initial values given to the program in the left subfigure, and the solution after 119 iterations of the optimizer in the right subfigure. In the initial values, figure 32a, the voxels of the water region can be clearly seen in yellow, and the breast in green.

**Table 8** – High contrast sound propagation speeds  $c_\ell$  in the simplified breast model (m/s). Longitudinal waves.

| medium        | $c_\ell$ (m/s) |
|---------------|----------------|
| water         | 1100.0         |
| breast tissue | 1510.0         |
| tumor         | 2700.0         |

**Figure 32** – Tomography with tumor in the fourth quadrant. Initial values and tomography after 119 optimization steps.

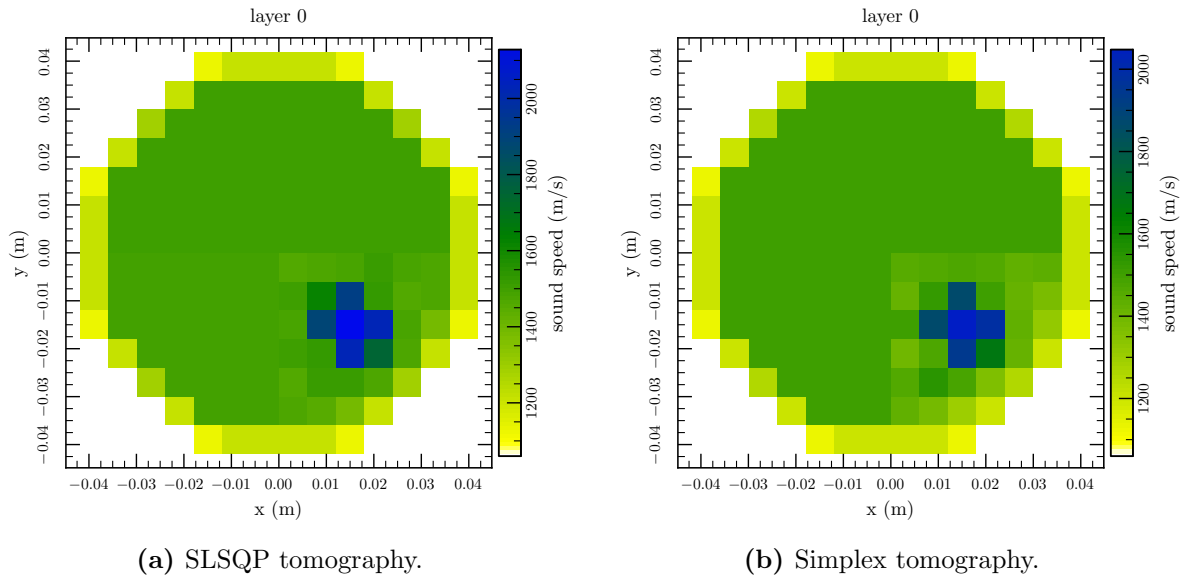


The optimizer SLSQP was also compared with the Simplex method (known as well as Nelder-Mead’s flexible polyhedra or amoeba optimization). As shown in figure 33 there

<sup>8</sup>The “missing”  $14^2 - 156 = 40$  voxels are in regions occupied by the ALID layer, and therefore are excluded from the the search of a solution.

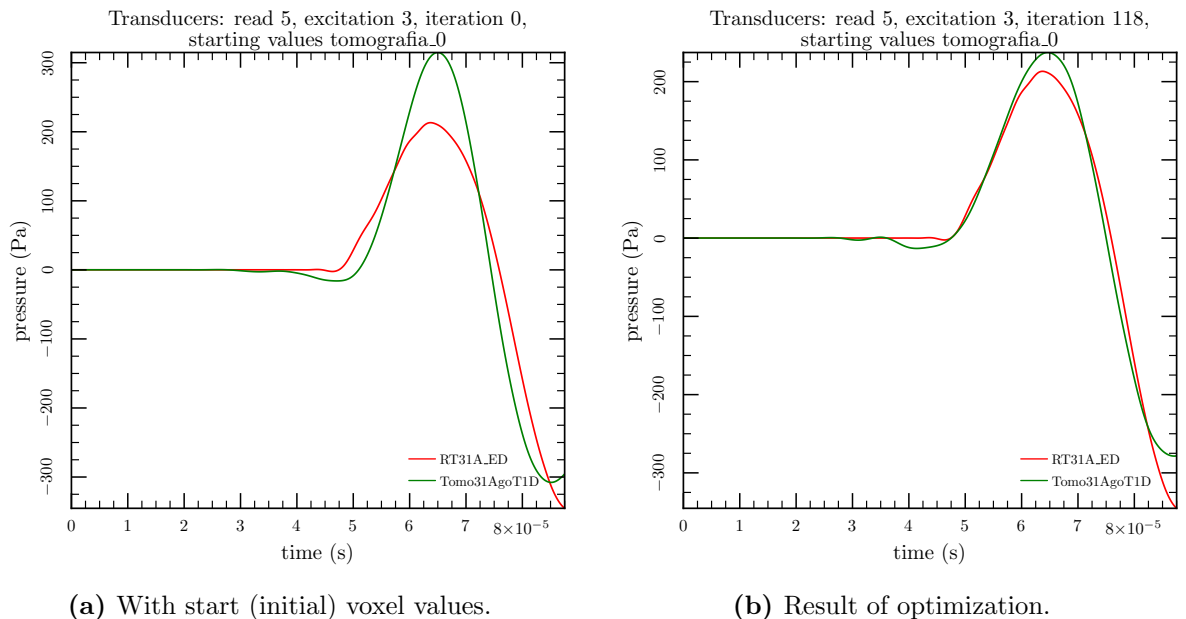
is a good agreement between both results; however, SLSQP found a solution in 7h27m, while the simplex optimizer stopped only after it reached the limit for iterations, which took 34h36m of run time.

**Figure 33** – Comparison of optimizers: SLSQP *versus* Simplex.



The objective function used here is extremely simple, yet effective. The idea is to match in time the peak values of the signals; the optimized direct problem of the tomography must match the reference. This is shown in figure 34 for a pair of random transducers, one firing and the other reading.

**Figure 34** – Effectiveness of the simple objective function used. Red curve is the reference reading, green curve is optimized for best overall match (total of 6 transducers).

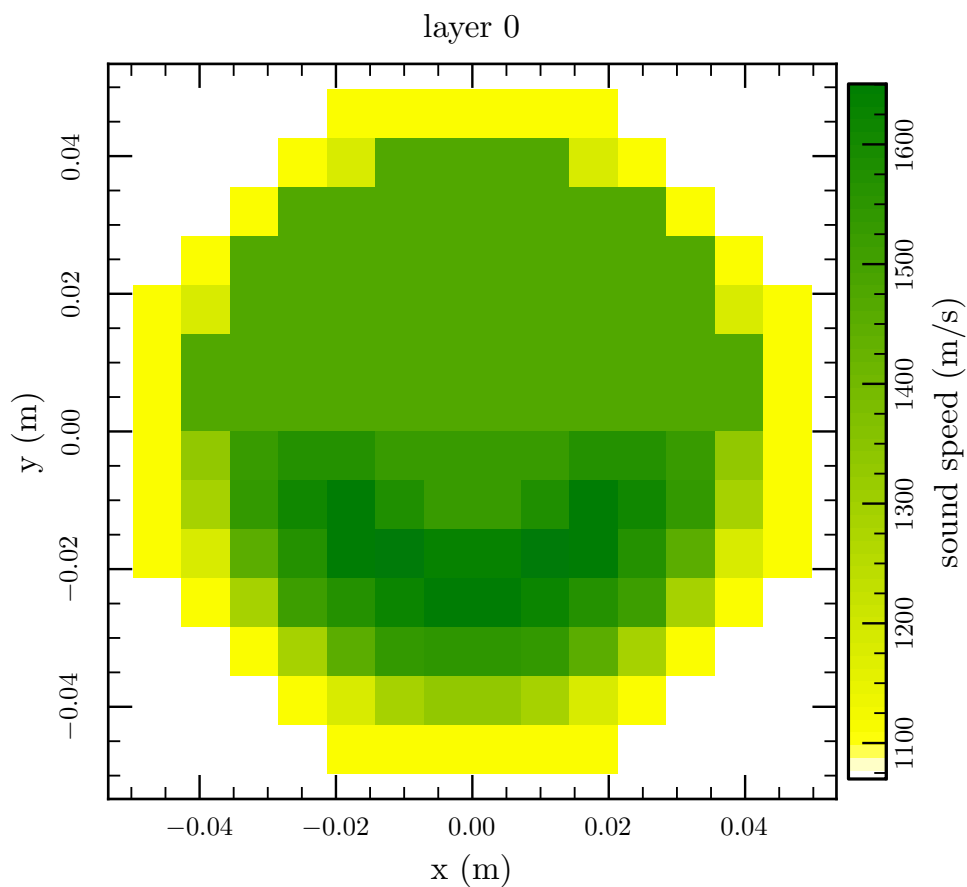




Further tests were done with a larger model. The new vat is 10.0 cm in diameter, the breast has 8.0 cm in diameter, the tumor has 1.5356 cm in diameter and 2.0 cm in height. To improve the image resolution 24 transducers are used. The tomography has a new grid type where all of the bottom half of the region representing the breast have independently optimizable values, only the upper half voxels share a single optimization value. The grid has a discretization of 14 elements (voxels) in the  $x$  and  $y$  axes, 3 rings for water, and 60 optimization parameters, 56 of them for the bottom half of the breast where the tumor must be found. The grid has water rings with 8.4 cm in internal diameter, while the representation of the numerical phantom breast has 8.0 cm in diameter.

The mean values of the population, calculated from 100,000 samples, can be seen in figure 35. Each sample comes from independent uniform distributions for breast radius, tumor radius and tumor position, and the assumed percentage of tumors in the population was 53.5%.

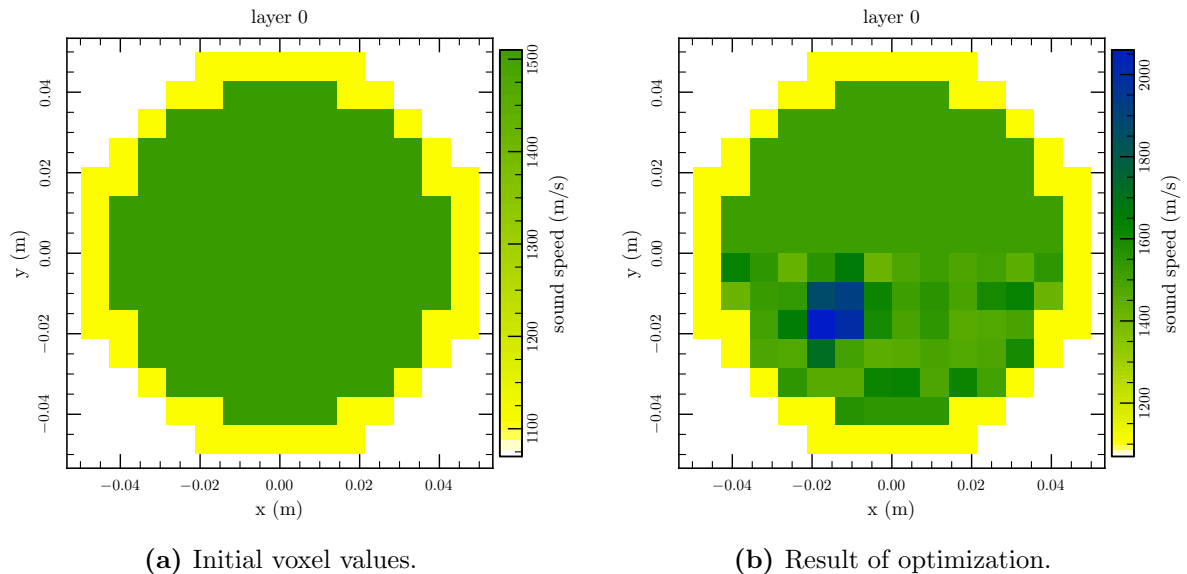
**Figure 35** – Mean values from 100,000 elements randomly drawn, on a 14 x 14 voxel grid, with uniform distribution of values for breast and tumor sizes and tumor position. The tumor can be almost anywhere in the bottom half in 53.5% of the individuals.



The initial voxel values and the solution of the inverse problem, after 50 iterations,

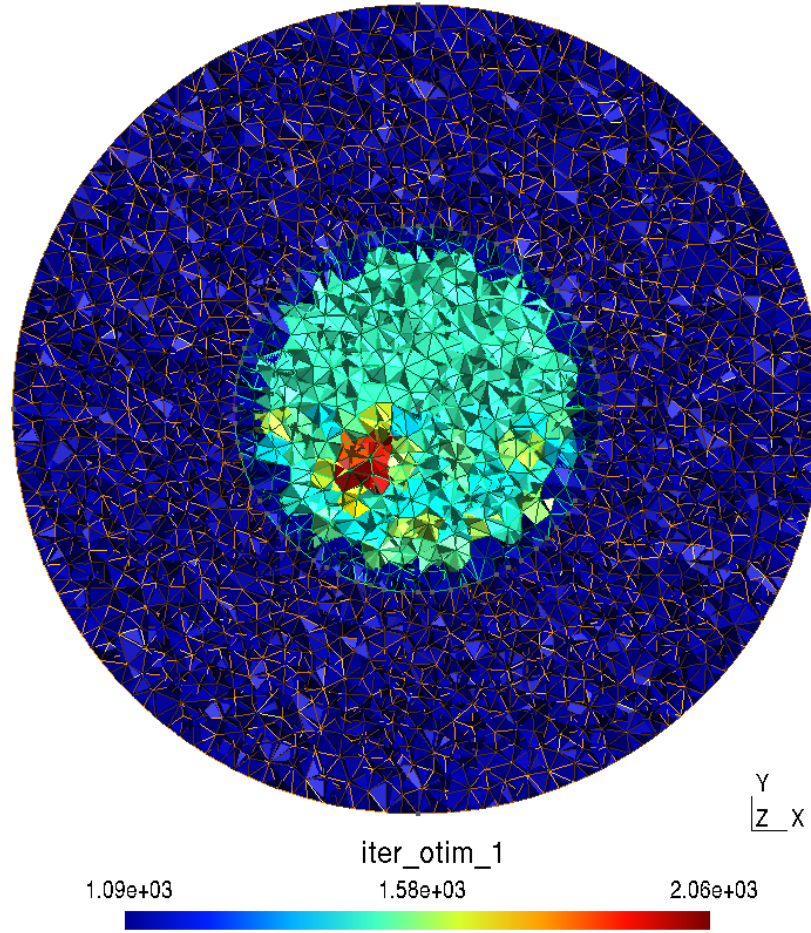
are in figure 36. The numerical phantom used in this test had the tumor in the third quadrant (bottom left quarter of the circle), and the solution of the inverse problem correctly located the tumor. The elements of the FE mesh, with their optimized values, are in figure 37.

**Figure 36** – Tomography with 24 transducers and tumor anywhere in the bottom half of the breast. Regularization with mean values from figure 35.

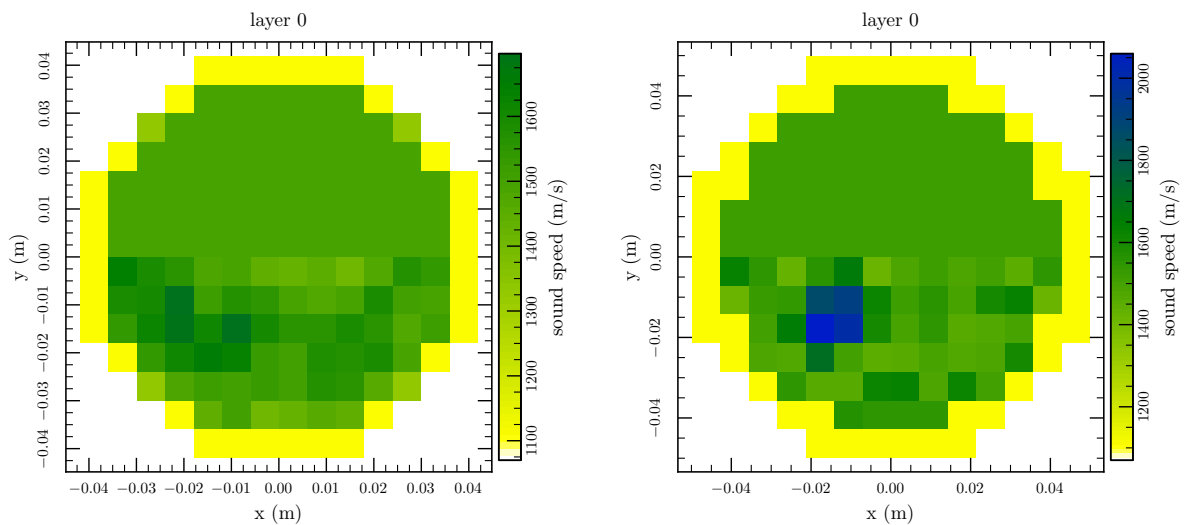


The jump in use of only 6 to 24 transducers was done after the optimization using 6 transducers stopped when it reached the limit of iterations and the result started to show higher speed voxels in the region of the tumor; besides the number of transducers, the numerical phantom had its sizes slightly increased. The tomographies with 6 and with 24 transducers are in figure 38. Whenever the numerical phantoms have the same size, the optimized tomography with 6 transducers could be used as initial guess for the 24 transducers tomography; in a similar fashion, one result could easily be used as initial values in a refined tomography with doubled resolution, that is, using a grid with twice the number of divisions in the  $x$  and  $y$  directions (and optionally twice in the  $z$  direction).

**Figure 37** – Mesh of the direct problem after last optimization by the UST algorithm – compare with figure 36b. This mesh includes the ALID. The scale shows sound wave speed in m/s.



**Figure 38** – Improvement in a tomography with more transducers, here going from 6 to 24. Problems with *similar sizes*, all other conditions are the same.



(a) Tomography using 6 transducers.

(b) Tomography using 24 transducers.

## 5 ANALYSIS AND DISCUSSION

In the present chapter the results obtained are analyzed. The time to solve the numerical inverse problem with a simple domain is reported.

After some tests of the program, when the optimization finishes and the final tomography is generated, it is evident that the complete process in this tomography algorithm takes a lot of time. One single simulation (direct problem) of ultrasound propagation, in the case of small models like the simplified breast used in this work (see 3.1.4.2), is fast<sup>1</sup>, *but* one single simulation has to be repeated by the number of transducers, if they fire one at a time. Then, for each optimization parameter, when its value is changed, the direct problem must be done again to show how this change affects the results. The number of optimization parameters (voxels, individually or grouped in one parameter) in the tomography was greatly reduced with a grid type like the RS (see c), page 82 and some values in table 7), although this limits the possible location of the tumor to a restricted region of the breast – which is reasonable in real life. Actually, the direct problem is performed  $N_{op} + 1$  times, the number  $N_{op}$  of optimization parameters plus one, accounting for the base situation without perturbed voxels/parameters and then with perturbations in each of the optimization parameters, while numerically calculating gradients for the optimization algorithm. Then, the process is driven by the iterative optimization, which in some tests required about 100 iterations to converge<sup>2</sup>. The time needed, with 12 CPU cores, ranged from 2h31m45s (44 parameters, 21 iterations, 6 transducers) to 2 days 8h26m03s (60 parameters, 24 transducers, reached the limit of 50 iterations and generated the tomography in figure 36b.)

For a more detailed tomography, with a finer voxel grid (more optimization parameters), the time will be greater. For the same dimensions and FE mesh, the “jump” from a grid of 16 x 16 divisions to one with 30 x 30 divisions, still according to table 7, with the RS grid type, is an increase of  $133/44 \approx 3$  times in the total running time of the program

---

<sup>1</sup>Only 3 seconds for 1125 integration steps, with 8 CPU cores working.

<sup>2</sup>Or around 20 iterations, with bad regularization weighting and very bad “results” as tomographies. There are also a few runs that reached the iteration limit.

– if the optimizer could maintain the same performance regardless of the problem size. A finer voxel grid can only be well represented by a mesh with more and smaller elements, otherwise there would exist optimization parameters that would not affect the mesh.

Besides the long running time of the proposed FEM tomography algorithm, there are other considerations that can be made:

- a)** lower sensitivity with higher resolution/smaller voxels. When the voxels are smaller, the effect of changes in their properties are less noticeable than with bigger voxels. Suppose an initial guess with the same sound speed in every voxel; at some point the algorithm changes, using a small increment, the value for the  $n$ -th voxel. Then a simulation of the tomographic process, with some transducers firing<sup>3</sup> is done, and the effect of the change in this  $n$ -th voxel used to determine if and how the performance index improved (numerically determining gradients, the Jacobian of the system) therefore the optimizer could establish better voxel properties in the next iteration; for a small voxel the overall effect is less noticeable than when a greater amount of material (in a bigger voxel) has its properties changed.

For a voxel  $V_1$  with a given sound propagation speed  $c$  and a length  $l_{V_1}$  (in any direction traversed by the sound wave), the time  $t_{V_1}$  it takes for the wave to traverse this voxel in this direction is

$$t_{V_1} = \frac{l_{V_1}}{c} . \quad (5.1)$$

Now, applying an increment to the sound speed,  $c + \Delta_c$ , the time is

$$t_{V_1\Delta} = \frac{l_{V_1}}{c + \Delta_c} . \quad (5.2)$$

Suppose now a more refined tomography is desired, with a higher resolution voxel grid, such that new voxel lengths  $l_{V_2}$  are now a fraction  $f$  of the former lengths,  $l_{V_2} = fl_{V_1}$ . For simplicity, consider that the new voxel  $V_2$  and it(s) neighbor(s) have both the same sound speed  $c$  as before; the *same path* from the coarser tomography takes the same amount of time, but now encompasses a smaller length  $l_{V_2}$  in the voxel  $V_2$  and the remaining  $l_{V_1} - l_{V_2}$  in it(s) neighbor(s),

$$t_{V_1} = \frac{l_{V_2}}{c} + \frac{l_{V_1} - l_{V_2}}{c} = \frac{fl_{V_1}}{c} + \frac{l_{V_1} - fl_{V_1}}{c} = \frac{fl_{V_1}}{c} + (1 - f)\frac{l_{V_1}}{c} . \quad (5.3)$$

Now, when *only* one of the new smaller voxels has its sound speed “perturbed” by

---

<sup>3</sup>According to some adequate excitation pattern, which in our case is a single transducer firing, and after that another transducer firing, until all transducers have excited the system.

the increment, the new time  $t_{V_{2\Delta}}$  will be

$$t_{V_{2\Delta}} = \frac{fl_{V_1}}{c + \Delta_c} + (1 - f)\frac{l_{V_1}}{c} . \quad (5.4)$$

and in the limit as  $f$  approaches 0 of  $t_{V_{2\Delta}}$  (an “ultra-refined” grid) and using equation 5.1,

$$\lim_{f \rightarrow 0} t_{V_{2\Delta}} = \lim_{f \rightarrow 0} \left( \frac{fl_{V_1}}{c + \Delta_c} + (1 - f)\frac{l_{V_1}}{c} \right) = \frac{l_{V_1}}{c} = t_{V_1} , \quad (5.5)$$

so, even for a big (and unnatural) increment  $c + \Delta_c$ , the new time  $t_{V_{2\Delta}}$  will be indistinguishable from  $t_{V_1}$ .

The real problem is: there is no need to reach such extreme situations. It suffices for  $t_{V_{2\Delta}} - t_{V_1}$  (or, for that matter, also  $t_{V_{1\Delta}} - t_{V_1}$ ) to be less than the time step of the simulation to have virtually identical results, and the perturbation  $\Delta_c$  will be ineffective, therefore the optimizer will give up with this voxel.

For the simplified breast, whose model has a diameter of 10 cm, assuming a sound propagation speed  $c$  of 1500.0 m/s, a perturbation (increment)  $\Delta_c$  of 10.0 m/s, table 9 shows, for various voxel grid divisions  $N \times N$ , the face to face length of the voxel  $l_{V_1}$  – the shortest distance if it is assumed the wave will pass through the whole voxel in its way from emitter to receiver –, the times  $t_{V_1}$  and  $t_{V_{1\Delta}}$  for the initial (unperturbed) and incremented (perturbed) sound wave speeds to traverse this distance, and the difference  $t_{V_1} - t_{V_{1\Delta}}$ . Figure 39 summarizes the sensitivities from table 9 for  $N$  in the range from 10 to 50. It is clear that, with an integration step of  $1.0 \times 10^{-8}$ , the optimizer should notice the effect of  $\Delta_c$  applied to a voxel until a grid of  $40 \times 40$  voxels; however, for the sake of faster tests, a time step of  $4.0 \times 10^{-8}$  is being used; it seems that all these faster simulations are wasted time, because for this step anything above  $N = 10$  will fall below the limit for perceptible effects.

**Table 9** – “Time sensitivity”  $t_{V_1} - t_{V_{1\Delta}}$  for various voxel grid resolutions  $N \times N$ . Assuming a model with diameter 10 cm, sound propagation speed of 1500.0 m/s and speed increment of 10.0 m/s.

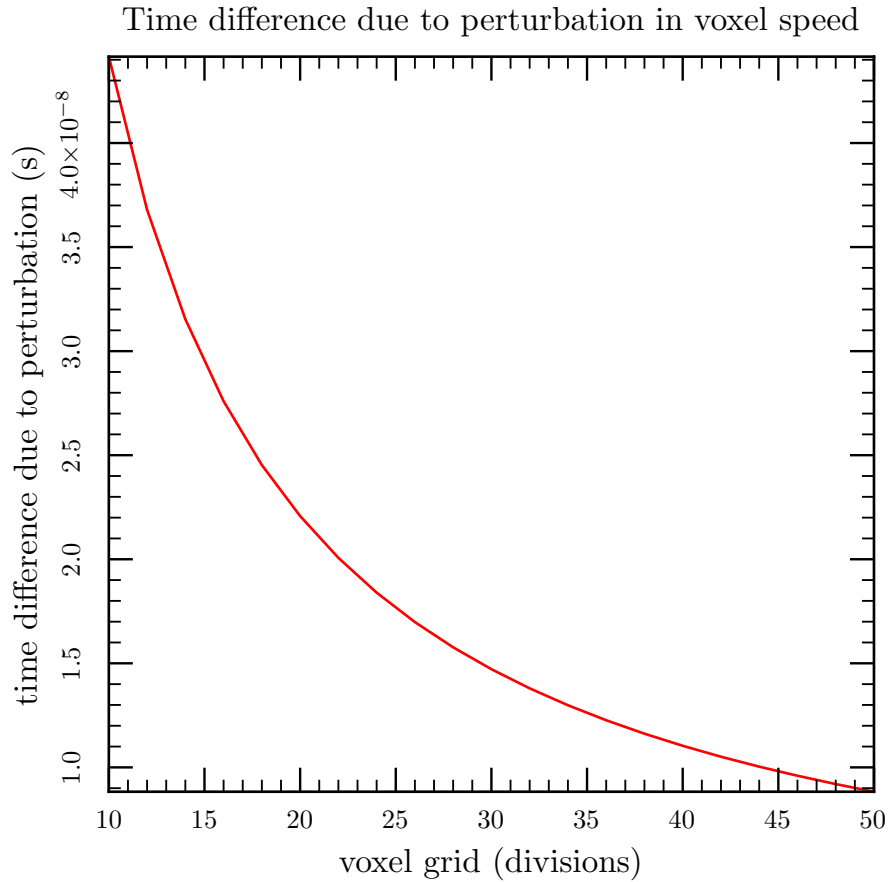
| N  | $l_{V_1}(m)$              | $t_{V_1}(s)$              | $t_{V_{1\Delta}}(s)$      | $t_{V_1} - t_{V_{1\Delta}}(s)$ |
|----|---------------------------|---------------------------|---------------------------|--------------------------------|
| 10 | $1.000000 \times 10^{-2}$ | $6.666667 \times 10^{-6}$ | $6.622517 \times 10^{-6}$ | $4.415011 \times 10^{-8}$      |
| 12 | $8.333333 \times 10^{-3}$ | $5.555556 \times 10^{-6}$ | $5.518764 \times 10^{-6}$ | $3.679176 \times 10^{-8}$      |
| 14 | $7.142857 \times 10^{-3}$ | $4.761905 \times 10^{-6}$ | $4.730369 \times 10^{-6}$ | $3.153579 \times 10^{-8}$      |
| 16 | $6.250000 \times 10^{-3}$ | $4.166667 \times 10^{-6}$ | $4.139073 \times 10^{-6}$ | $2.759382 \times 10^{-8}$      |
| 18 | $5.555556 \times 10^{-3}$ | $3.703704 \times 10^{-6}$ | $3.679176 \times 10^{-6}$ | $2.452784 \times 10^{-8}$      |
| 20 | $5.000000 \times 10^{-3}$ | $3.333333 \times 10^{-6}$ | $3.311258 \times 10^{-6}$ | $2.207506 \times 10^{-8}$      |
| 22 | $4.545455 \times 10^{-3}$ | $3.030303 \times 10^{-6}$ | $3.010235 \times 10^{-6}$ | $2.006823 \times 10^{-8}$      |
| 24 | $4.166667 \times 10^{-3}$ | $2.777778 \times 10^{-6}$ | $2.759382 \times 10^{-6}$ | $1.839588 \times 10^{-8}$      |
| 26 | $3.846154 \times 10^{-3}$ | $2.564103 \times 10^{-6}$ | $2.547122 \times 10^{-6}$ | $1.698081 \times 10^{-8}$      |
| 28 | $3.571429 \times 10^{-3}$ | $2.380952 \times 10^{-6}$ | $2.365184 \times 10^{-6}$ | $1.576790 \times 10^{-8}$      |
| 30 | $3.333333 \times 10^{-3}$ | $2.222222 \times 10^{-6}$ | $2.207506 \times 10^{-6}$ | $1.471670 \times 10^{-8}$      |
| 32 | $3.125000 \times 10^{-3}$ | $2.083333 \times 10^{-6}$ | $2.069536 \times 10^{-6}$ | $1.379691 \times 10^{-8}$      |
| 34 | $2.941176 \times 10^{-3}$ | $1.960784 \times 10^{-6}$ | $1.947799 \times 10^{-6}$ | $1.298533 \times 10^{-8}$      |
| 36 | $2.777778 \times 10^{-3}$ | $1.851852 \times 10^{-6}$ | $1.839588 \times 10^{-6}$ | $1.226392 \times 10^{-8}$      |
| 38 | $2.631579 \times 10^{-3}$ | $1.754386 \times 10^{-6}$ | $1.742768 \times 10^{-6}$ | $1.161845 \times 10^{-8}$      |
| 40 | $2.500000 \times 10^{-3}$ | $1.666667 \times 10^{-6}$ | $1.655629 \times 10^{-6}$ | $1.103753 \times 10^{-8}$      |
| 42 | $2.380952 \times 10^{-3}$ | $1.587302 \times 10^{-6}$ | $1.576790 \times 10^{-6}$ | $1.051193 \times 10^{-8}$      |
| 44 | $2.272727 \times 10^{-3}$ | $1.515152 \times 10^{-6}$ | $1.505117 \times 10^{-6}$ | $1.003412 \times 10^{-8}$      |
| 46 | $2.173913 \times 10^{-3}$ | $1.449275 \times 10^{-6}$ | $1.439678 \times 10^{-6}$ | $9.597850 \times 10^{-9}$      |
| 48 | $2.083333 \times 10^{-3}$ | $1.388889 \times 10^{-6}$ | $1.379691 \times 10^{-6}$ | $9.197940 \times 10^{-9}$      |
| 50 | $2.000000 \times 10^{-3}$ | $1.333333 \times 10^{-6}$ | $1.324503 \times 10^{-6}$ | $8.830022 \times 10^{-9}$      |

$l_{V_1}$ : size of the voxel (face to face),  $t_{V_1}$  and  $t_{V_{1\Delta}}$ : times to traverse  $l_{V_1}$  with initial and incremented sound speeds.

At least, in the case where many voxels share the same value of a single “optimization parameter” – as is the case with geometries of types R and RS – the sensitivity is improved in comparison with the situation of individual and independent voxels. However, where a tumor is to be detected, each voxel must have its own value to be optimized, therefore there remains a compromise mesh refinement *versus* Jacobian sensitivity.

- b)** the discretization interval affects the sensitivity also. All transducer measurements generated in the direct problem simulations are recorded in a file at each time

**Figure 39** – Plot of the “time sensitivity”  $t_{V_1} - t_{V_1\Delta}$  for various voxel grid resolutions. Same conditions as in table 9



increment. In the event of reference data from a physical experiment, the measured values will be available only at discrete time steps, according to the acquisition system capabilities, but probably the simulation steps for the direct problem will be greater than the sampling time of the transducers. Because of discretized time, errors in the instants of the peaks between reference and direct problem do not have continuous variations, but only can be detected at the greater time increment between real measurement (or reference simulation) and tomography simulation. In the algorithm those time differences are divided by the value of the time increment, resulting in an error in terms of intervalar arithmetic.

- c) one of the tricks to speed the algorithm is the possibility to use for the tomography direct problem an integration time step  $\Delta_t$  that is a multiple  $N_{ts} \times \Delta_t$  – usually the double – of the time step in the reference simulation. This way each FE simulation will be  $N_{ts}$  times faster, with  $1/N_{ts}$  steps, than if using the same reference time step and full number of integration steps. This  $N_{ts}$  multiple also reduces the sensitivity as explained in **b)** above.



**d)** better ALID dimensioning. According to [38, Chapter 3], there are occurrences of “*numerical reflection*”, physically non-existent but visible in FE results<sup>4</sup>. This reflection increases with bigger amplitude (higher power) of the ultrasound waves and decreases, but less and less, with more finite elements per wavelength. It also increases with higher coefficients in the ALID (or PML) attenuation equation<sup>5</sup>. In the current study, ALID dimensioning was made empirically, changing each of the  $N_1/N_2/N_3$  values based on previous results until a good result was achieved.

With numerical reflection the echo absorbing layer still works, but there is an optimal value for the attenuation equation coefficients; beyond this optimum, the reflection increases considerably, more than 20 dB<sup>6</sup>.

This is one reason to use thick ALID layers – at least in the reference models – so lower attenuations could be used, eventually reducing any numerical echo in the simulation.

**e)** better performance index. The optimizer minimizes the performance index where the error is defined as the sum of differences of peak time of the predicted pressures and *measured* pressures. Changing the definition of error, the sensitivity should change.

---

<sup>4</sup>It is explained in [72]: when a PDE problem, like the sound wave equations, is solved by volume discretization, as is the case in the FE method, the computational grid is truncated, which can introduce artifacts. The most difficult problems to truncate involve wave equations, where the solutions are oscillating and typically decay with distance  $r$  only as  $1/r^{(d-1)/2}$  in  $d$  dimensions. The slow decay means that simply truncating the grid with hard-wall (Dirichlet or Neumann) or periodic boundary conditions will lead to unacceptable artifacts from boundary reflections. Therefore, wave equations require something different: an absorbing boundary that will somehow absorb waves that strike it, without reflecting them, and without requiring infeasible resolution. The best of those solutions, PML (used in [71] and [38]) is reflectionless only when the exact wave equations are solved, which is not the case in a FEM discretized problem. When the medium varies slowly reflections can be made small, and increasing the resolution also increases the effectiveness of the PML, because it approaches the exact wave equation.

<sup>5</sup>Although in his work Drozd uses other types of equations, similarly, with a too big  $C_{a_0}$  in equation 4.1, this thesis ALID may have numerical reflections too.

<sup>6</sup>See [38, Chapter 3].

## 6 CONCLUSIONS AND CONTINUITY OF THE WORK

The experience with the use of FE models for the ultrasound simulation and for the US tomography algorithm has enabled us to draw some conclusions and propose alternatives or improvements for further developments on the subject.

### 6.1 Conclusions

The main conclusion is that the Finite Elements Method is a flexible three-dimensional modeling platform for Ultrasound Tomography and viable from the point of view of computational cost. The methodology and results allow the following statements to be written:

- the use of finite elements is viable to solve an inverse problem and therefore implement an ultrasound tomography algorithm in three dimensions; some aspects of the present work are in three dimensions, acoustic waves are free to propagate in the axial direction of the numerical phantoms and the transducers have finite dimensions in the axial direction;
- FE models for the acoustical direct problem can be build, and to work effectively echoes at the boundaries of the model must be eliminated. For that an ALID layer is employed; however this layer increases the size of the complete FE model. The increment of dissipation with distance in the ALID elements must be progressive, not much intense, and the thickness of the layer must be enough to absorb and dampen a sound wave that otherwise could return as echo;
- an optimization scheme was implemented and allowed an adequate set of acoustical properties to be applied to all the voxels used to represent the domain. Values of each voxel were attributed to finite elements inside them, and the direct problem solution shows the effect of the changing properties at each iteration of the optimizer;

- the algorithm optimized the *a posteriori* conditional probability density function, and a tomography correctly located regions of higher sound wave velocity of a numerical phantom, corresponding to a tumor, inside a breast; mean values from thousands of drawings of breast samples, with and without tumor, and specified distributions of sizes and positions, were used as regularization for the ill-posed inverse problem;
- acceleration of the program via parallelization is effective, but can be questionable for small problems. Using OpenMP the simulation and tomography program can run much faster than a serial version of the program, however, this depends on the size of the problem. Very small problems run faster in serial mode, medium size problems are faster using only the CPU (maybe not using all available cores if they were many) and not the GPU, and really big problems can benefit of plenty of CPU cores and GPU. The simplified breast model in this thesis, with an internal vat 10 cm in diameter and ALID, is faster in parallel mode, both in a notebook CPU with 8 cores and a desktop computer with 12 cores, both without use of the GPU;
- use of a big penalty value, as a method to impose internal pressure restrictions, is not advisable. The literature suggests the use of a very big numerical value as penalty in the penalty method (see section 3.1.2 on page 43) but the tests demonstrated that, while following the desired input more precisely at the nodes where it is applied, finite elements in the vicinity had unnatural values. The penalty was lowered and then all the elements had sensible values;
- small integration time steps are required for an effective optimization of the voxel values. From the discussion in **a)** (page 92), and with the values in table 9, the time sensitivity can be a problem with smaller voxels in the tomography. When the integration step is higher than the time sensitivity, the optimizer can not detect changes it makes to the voxels: differences in time between an original wave and a perturbed wave, after a voxel has his speed increased, will fall in the same integration step, and the optimizer can not use this information;
- as expected from similar inverse problems, more transducers improve the optimization. There is more information, more readings, for the optimizer to work with; if a single measurement is able to lower the performance index while the others don't (because of low sensitivities). In figure 38 the improvement of a tomography when going from 6 to 24 transducers illustrates the usefulness of the added information for the optimization, although this also increases the time for one complete pass of

the direct problem (i. e., all the transducers had to fire).

The increased number of finite elements due to the ALID layer, and the small time step because of the time sensitivity for the optimizer to work, lead to a FE problem in the tomography direct problem that takes longer to simulate and takes more steps too. And the numeric determination of derivatives and Jacobian for the optimizer requires a complete simulation – all transducers emitting, one after the other – for each voxel with a perturbed (increased) sound speed.

With the proposed algorithm a US tomography is numerically feasible. Although its time of execution is large at the present stage, modeling the physical environment using the FE method opens many possibilities, as e. g. use of non-linear models, dissipation that follows the power law (or others, see [9, chap. 4]), modeling of highly complex geometries and irregular shapes with different materials like the structure of a porous bone<sup>1</sup>, use of multiple types of boundary conditions, many possibilities to interpret and visually see the results with existing FEM programs, etc.

## 6.2 Continuity of the work

A problem with the proposed FEM tomography algorithm is the amount of time it takes, even with an extremely simplified and small model. One way to speed up FE simulations can be the use of Fourier series of the excitation signal.

In this particular work, the number of transducers was chosen as 24. It is probable that more transducers, capturing a richer set of information about sound propagation in the breast, could help achieve more detailed tomographies; the number used was reduced (instead of using 36 or 48 transducers) in order to reduce computational time, especially while developing/debugging the program and also while testing sets of weightings for the regularization. Some equipments (or studies) for breast UST have 200 [73], 256 [2, 74] or even 2048 transducers (SoftVue, [75]).

Another direction of improvement is the use of full wave inversion (FWI) or waveform inversion; instead of using the peak value of pressure signals, the likelihood and the performance index can be defined in terms of the complete signals. A technique initially used in geophysics problems and now spreading to UST of human tissues and specifically of the breast. FWI has not only very good spatial resolution, but can also give the sound speeds and the acoustic attenuation with good precision, and is already being used for UST

---

<sup>1</sup>See, for instance, [42].

of the breast [73, 74, 30, 29, 76]. In this technique the wave equations are solved, therefore every possible information can be extracted from the acoustics properties of subjacent media; in contrast, methods based on approximate models or on simplifications of the US phenomena like ray-tracing or time of flight algorithms can have fast convergence, but limited spatial resolution. There are FWI methods that solve only the sound propagation speed problem, others solve only the attenuation, and methods for both at the same time. FWI is computationally intensive, and some articles state that results can be obtained in minutes, with high resolution, but the hardware was not specified besides the use of GPUs.

For the numerical evaluation of derivatives, the program uses an increment  $\Delta$  in the values of the optimization parameters. This increment is a parameter passed in the program call. All tomographies until now used a value of 10.0, and a very small increment may have negligible effect, while a big increment may give results not much precise. An optimization of the  $\Delta$ , perturbation of sound speed during Jacobian computations, may bring improvements. Currently only the sound propagation speed is optimized, therefore  $\Delta$  ( $\Delta_c$  in this case) refers to this physical property, with units of m/s.

There is room for improvements concerning the direct problem numerical simulation problem:

- nodes of the transducers could be determined by the simulation program directly. Currently the nodes of a transducer are read from a “definition file”, generated by an auxiliary program. This requires the mesh to be recorded in a file with an adequate format (INP) – which is not the format used by the simulation program – and this file must be created when Gmsh generates the mesh. The auxiliary program creates an ordered list of nodes for each transducer, but missing some lines (header, end of data delimiter, a few others) and because of that this list of nodes must be inserted manually in a template, or substitute the nodes in an already existing definition file: some editing must still be done by the user. However, the mesh file the simulator uses already has embedded information about “physical surfaces”, which correspond to the transducer surfaces, and this allows for the identification of the correct nodes. This identification process is more intricate than the current method, but can be implemented and would avoid human errors, reduce the number of data files, and simplify the use of new mesh files.
- better detection of user errors. In line with the previous item, improvements to the checks already being done with data files can be implemented, reducing the

possibility of bad usage because wrong data was selected for a tomography. For example, the files with mean values and covariances for the regularization have different sizes (number and length of lines) according to the intended discretization (voxel grid), but this information is not yet available in the header of the files. All data files have headers, and some files have additional lines which helps the user when he changes the data<sup>2</sup>.

The selection of the correct files for a simulation or a tomography relies only on the name assigned to them, and a user may easily forget to change a name – after he had already changed other names in the main data file of the program –, select a file with a name similar to the intended, and so on. Today many files have short headers which the user can edit and use to know about the purpose of the file. Improved headers, with some lines with automatic content, written by the programs which create some of these files, could contain the name of the FE model, voxel grid, number of materials in the model, number of transducers, etc. With this additional information the tomography program could verify if the data files are all compatible, e. g., if an initial guess is compatible with the grid of the regularization values and vice-versa, if the reference simulation and the tomography direct model are of the same models (only with different number of elements per wavelength and ALID) and the same number of transducers.

There are more methodological suggestions:

- use as initial guess the expected mean values,
- compare reference (precise) simulations with different values of finite elements per wavelength; if the measurements do not show signal peaks at the same time, then the use of a lower number of finite elements per wavelength in the mesh used for the tomography direct problem will not work,
- in the same way, compare reference simulations with different ALIDs and see if and how much this could influence the measured signal peaks regarding to their precise time.

---

<sup>2</sup>These additional lines also allow the program to check for some errors.

## BIBLIOGRAPHY

- [1] Cuiping Li, Neb Duric, and Lianjie Huang. “Breast imaging using transmission ultrasound: Reconstructing tissue parameters of sound speed and attenuation”. In: *2008 International Conference on BioMedical Engineering and Informatics, BMEI 2008*. Vol. 2. 978-0-7695-3118-2. IEEE Computer Society, 2008, pp. 708–712. DOI: [10.1109/BMEI.2008.303](https://doi.org/10.1109/BMEI.2008.303).
- [2] Krzysztof J. Opieliński et al. “Imaging results of multi-modal ultrasound computerized tomography system designed for breast diagnosis”. In: *Computerized Medical Imaging and Graphics* 46 (2015). Information Technologies in Biomedicine, pp. 83–94. ISSN: 0895-6111. DOI: [10.1016/j.compmedimag.2015.02.004](https://doi.org/10.1016/j.compmedimag.2015.02.004). URL: <https://www.sciencedirect.com/science/article/pii/S0895611115000452>.
- [3] Kun Wang et al. “Breast Ultrasound Computed Tomography Using Waveform Inversion with Source Encoding”. In: *Medical Imaging 2015: Ultrasonic Imaging and Tomography*. Vol. 9419. Proceedings of SPIE. 2015, pp. 475–493.
- [4] Nghia Q. Nguyen and Lianjie Huang. “Ultrasound Bent-Ray Tomography Using both Transmission and Reflection Data”. In: *Medical Imaging 2014: Ultrasonic Imaging and Tomography*. Ed. by Johan G. Bosch and Marvin M. Doyley. Vol. 9040. Proceedings of SPIE. International Society for Optics and Photonics. SPIE, 2014, 90400R. DOI: [10.1117/12.2043275](https://doi.org/10.1117/12.2043275).
- [5] Hervé Carfantan and Ali Mohammad-Djafari. “A Bayesian approach for nonlinear inverse scattering tomographic imaging.” In: *ICASSP. 1995 International Conference on Acoustics, Speech, and Signal Processing*. IEEE Computer Society, 1995, pp. 2311–2314. ISBN: 0-7803-2431-5. URL: <http://doi.ieeecomputersociety.org/10.1109/ICASSP.1995.479954>.
- [6] S. A. Johnson et al. “Non-invasive breast tissue characterization using ultrasound speed and attenuation”. In: *Acoustical Imaging* 28 (2007), 147–154.
- [7] P. D. Edmonds et al. “Ultrasound tissue characterization of breast biopsy specimens”. In: *Ultrasonic Imaging* 13 (Apr. 1991), pp. 162–185.

- [8] J. Ophir et al. “Elastography: Ultrasonic estimation and imaging of the elastic properties of tissues”. In: *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine* 213.3 (1999). PMID: 10420776, pp. 203–233. DOI: [10.1243/0954411991534933](https://doi.org/10.1243/0954411991534933).
- [9] Thomas L. Szabo. *Diagnostic Ultrasound Imaging: Inside Out*. 2nd ed. Elsevier Inc., 2014. ISBN: 978-0-12-396487-8.
- [10] Allan D. Pierce. *Acoustics - An Introduction to Its Physical Principles and Applications*. Acoustical Society of America, 1994. ISBN: 0-88318-612-8.
- [11] Karl F. Graff. *Wave Motion in Elastic Solids*. Dover Books on Physics Series. Dover Publications, 1991. ISBN: 9780486667454. URL: <https://books.google.com.br/books?id=5cZFRwLuhdQC>.
- [12] L. E. Kinsler et al. *Fundamentals of Acoustics*. 4th ed. Wiley, Dec. 1999, p. 560. ISBN: 978-0471847892.
- [13] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method: Its Basis and Fundamentals*. The Finite Element Method. Elsevier Science, 2013. ISBN: 9780080951355. URL: <https://books.google.com.br/books?id=7UL5Ls9h0F8C>.
- [14] Singiresu S. Rao. *The Finite Element Method in Engineering*. 4th ed. Elsevier Science & Technology Books, 2004. ISBN: 0750678283.
- [15] Daryl L. Logan. *A First Course in the Finite Element Method*. 5th ed. United States: Cengage Learning, 2012. ISBN: 9780495668251.
- [16] Wim Desmet and Dirk Vandepitte. “Finite Element Modeling for Acoustics”. In: ISAAC 13 - International Seminar on Applied Acoustics. Leuven, Belgium: LMS International, 2002. Chap. 2, pp. 37–85. URL: <http://citeseerx.ist.psu.edu/pdf/db51845f144501cec3f8ef8fd6d3dc21fab44926>.
- [17] Flávio Buiochi. “PMR5234 - Ultrassom na Indústria e Medicina”. Post-graduate course — course notes. 2015.
- [18] David T. Blackstock. *Fundamentals of Physical Acoustics*. 1st ed. Wiley Interscience, 2000, p. 576. ISBN: 0-471-31979-1. URL: [https://books.google.com.br/books?id=e\\_8sPIEd36YC](https://books.google.com.br/books?id=e_8sPIEd36YC).
- [19] Jerry H. Ginsberg. *Acoustics – A Textbook for Engineers and Physicists, Volume I: Fundamentals*. 1st ed. ASA Press and Springer, Oct. 18, 2017, pp. XXXV, 576. 596 pp. ISBN: 978-3-319-56843-0. DOI: [10.1007/978-3-319-56844-7](https://doi.org/10.1007/978-3-319-56844-7).



- [20] N. Atalla and F. Sgard. *Finite Element and Boundary Methods in Structural Acoustics and Vibration*. CRC Press, 2015. ISBN: 9781466592889. URL: <https://books.google.com.br/books?id=oHR3CAAQBAJ>.
- [21] Barbara Chapman, Gabriele Jost, and Ruud van der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*. Cambridge, MA, USA: The MIT Press, 2007. ISBN: 978-0262533027.
- [22] Ruud van der Pas et al. "Using OpenMP – The Next Step: Affinity, Accelerators, Tasking, and SIMD". Cambridge, MA, USA: The MIT Press, 2017. ISBN: 978-0262534789.
- [23] Carri Glide, Nebojsa Duric, and Peter Littrup. "Novel approach to evaluating breast density utilizing ultrasound tomography". In: *Medical Physics* 34.2 (2007), pp. 744–753. DOI: [10.1118/1.2428408](https://doi.org/10.1118/1.2428408).
- [24] Carri K. Glide-Hurst, Neb Duric, and Peter Littrup. "Volumetric breast density evaluation from ultrasound tomography images". In: *Medical Physics* 35.9 (2008), pp. 3988–3997. DOI: [10.1118/1.2964092](https://doi.org/10.1118/1.2964092).
- [25] Youli Quan and Lianjie Huang. "Sound-speed tomography using first-arrival transmission ultrasound for a ring array". In: *Medical Imaging 2007: Ultrasonic Imaging and Signal Processing*. Vol. 6513. SPIE. 2007, pp. 48–56.
- [26] Steven Schmidt et al. "Modification of Kirchhoff migration with variable sound speed and attenuation for acoustic imaging of media and application to tomographic imaging of the breast". In: *Medical Physics* 38.2 (2011), pp. 998–1007. DOI: [10.1118/1.3539552](https://doi.org/10.1118/1.3539552).
- [27] Olivier Roy et al. "Sound speed estimation using wave-based ultrasound tomography: theory and GPU implementation". In: *Medical Imaging 2010: Ultrasonic Imaging, Tomography, and Therapy*. Vol. 7629. SPIE. 2010, pp. 177–188.
- [28] Roberto J. Lavarello and Michael L. Oelze. "Tomographic Reconstruction of Three-Dimensional Volumes Using the Distorted Born Iterative Method". In: *IEEE Transactions on Medical Imaging* 28.10 (2009), pp. 1643–1653. DOI: [10.1109/TMI.2009.2026274](https://doi.org/10.1109/TMI.2009.2026274).
- [29] Fu Li et al. "A forward model incorporating elevation-focused transducer properties for 3D full-waveform inversion in ultrasound computed tomography". In: *arXiv preprint arXiv:2301.07787* (Jan. 2023). DOI: [10.48550/arXiv.2301.07787](https://doi.org/10.48550/arXiv.2301.07787). arXiv: [2301.07787](https://arxiv.org/abs/2301.07787) [physics.med-ph].

- [30] Fu Li et al. “3D full-waveform inversion in ultrasound computed tomography employing a ring-array”. In: *Medical Imaging 2023: Ultrasonic Imaging and Tomography*. Ed. by Christian Boehm and Nick Bottenus. Vol. 12470. SPIE Proceedings. Apr. 10, 2023, p. 20. DOI: [10.1117/12.2654406](https://doi.org/10.1117/12.2654406).
- [31] Fu Li et al. “Investigation of an elevation-focused transducer model to enable accurate three-dimensional full-waveform inversion in ultrasound computed tomography”. English (US). In: *Medical Imaging 2022*. Ed. by Nick Bottenus and Nicole V. Ruiter. Progress in Biomedical Optics and Imaging - Proceedings of SPIE. SPIE, Mar. 2022. DOI: [10.1117/12.2613207](https://doi.org/10.1117/12.2613207).
- [32] H. Gemmeke and N.V. Ruiter. “3D ultrasound computer tomography for medical imaging”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 580.2 (2007). Imaging 2006, pp. 1057–1065. ISSN: 0168-9002. DOI: [10.1016/j.nima.2007.06.116](https://doi.org/10.1016/j.nima.2007.06.116). URL: <https://www.sciencedirect.com/science/article/pii/S016890020701323X>.
- [33] Nicole Ruiter et al. “Optimization of the aperture and the transducer characteristics of a 3D Ultrasound Computer Tomography System”. In: *Progress in Biomedical Optics and Imaging - Proceedings of SPIE* 9040 (Feb. 2014). DOI: [10.1117/12.2043455](https://doi.org/10.1117/12.2043455).
- [34] Christophe Geuzaine and Jean-François Remacle. “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities”. In: *International Journal for Numerical Methods in Engineering* 11 (2009), pp. 1309–1331. URL: <http://gmsh.info/>.
- [35] William H. Press et al. *Numerical Recipes in C (second edition): The Art of Scientific Computing*. 2nd ed. New York, NY, USA: Cambridge University Press, 1992. ISBN: 0-521-43108-5. URL: <http://numerical.recipes/>.
- [36] Richard Barrett et al. “4. Related Issues”. In: *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. 2nd ed. Philadelphia, PA: SIAM, 1994, pp. 57–81. DOI: [10.1137/1.9781611971538.ch4](https://doi.org/10.1137/1.9781611971538.ch4).
- [37] \_\_\_\_\_. *Plano de Gestão de Dados — FAPESP*. ”FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo”, 2018. URL: <http://www.fapesp.br/gestaodados/> (visited on 10/2018).
- [38] Mickael Brice Drozd. “Efficient Finite Element Modelling of Ultrasound Waves in Elastic Media”. PhD thesis. London: Imperial College of Science Technology and

- Medicine, Jan. 2008. URL: <https://spiral.imperial.ac.uk/handle/10044/1/7974>.
- [39] Diego Armando Cardona Cárdenas. “Investigação da Reconstrução Tomográfica Utilizando Transdutores Distribuídos de Ultrassom”. revised edition. PhD thesis. São Paulo, Brasil: Universidade de São Paulo, Jan. 2018. URL: <http://www.teses.usp.br/teses/disponiveis/3/3154/tde-10042018-084731/publico/DiegoArmandoCardonaCardenasCorr18.pdf>.
- [40] Bradley E. Treeby and B. T. Cox. “k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields”. In: *Journal of Biomedical Optics* 15.2 (2010). ISSN: 1560-2281. URL: <http://www.k-wave.org/documentation.php>.
- [41] Tomasz Lodygowski and Wojciech Sumelka. “Limitations in Application of Finite Element Method in Acoustic Numerical Simulation”. eng. In: *Journal of Theoretical and Applied Mechanics* 44.4 (2006), pp. 849–865. ISSN: 1429-2955. URL: <http://search.proquest.com/docview/28974538/>.
- [42] B. Vafaeian et al. “The finite element method for micro-scale modeling of ultrasound propagation in cancellous bone”. eng. In: *Ultrasonics* 54.6 (Aug. 2014), pp. 1663–1676. ISSN: 0041-624X. DOI: [10.1016/j.ultras.2014.02.024](https://doi.org/10.1016/j.ultras.2014.02.024).
- [43] Dirk Rüter et al. “Low-Frequency Ultrasound Permeates the Human Thorax and Lung: a Novel Approach to Non-Invasive Monitoring”. In: *Ultraschall in der Medizin* 31 (Aug. 2010), pp. 53–62. ISSN: 0172-4614. DOI: [10.1055/s-0028-1109482](https://doi.org/10.1055/s-0028-1109482). URL: [https://www.researchgate.net/publication/26652388\\_Low-Frequency\\_Ultrasound\\_Permeates\\_the\\_Human\\_Thorax\\_and\\_Lung\\_a\\_Novel\\_Approach\\_to\\_Non-Invasive\\_Monitoring](https://www.researchgate.net/publication/26652388_Low-Frequency_Ultrasound_Permeates_the_Human_Thorax_and_Lung_a_Novel_Approach_to_Non-Invasive_Monitoring).
- [44] John W. Eaton et al. *GNU Octave version 7.3.0 manual: a high-level interactive language for numerical computations*. 2022. URL: <https://www.gnu.org/software/octave/doc/v7.3.0/> (visited on 05/24/2023).
- [45] S. C. de Kleijn. *The Shape of Breasts Suspended in Liquid*. Tech. rep. PR-TN 2007/00366. Koninklijke Philips Electronics N.V, Netherlands: Philips Research Europe, June 2007.
- [46] Nick Mulcahy. *FDA Approves Breast Imager that does not Compress Tissue*. news. Medscape, Feb. 9, 2015. URL: <http://www.medscape.com/viewarticle/839502> (visited on 02/18/2016).

- [47] F. A. Duck. *Physical Properties of Tissues: A Comprehensive Reference Book*. Academic Press, Oct. 22, 2013. ISBN: 9781483288420. DOI: [10.1016/C2009-0-02755-X](https://doi.org/10.1016/C2009-0-02755-X). URL: <https://books.google.com.br/books?id=UaXpAgAAQBAJ>.
- [48] S. A. Goss, R. L. Johnston, and F. Dunn. “Comprehensive compilation of empirical ultrasonic properties of mammalian tissues”. In: *The Journal of the Acoustical Society of America* 64.2 (Aug. 1978), pp. 423–457. ISSN: 0001-4966. DOI: [10.1121/1.382016](https://doi.org/10.1121/1.382016). eprint: [https://pubs.aip.org/asa/jasa/article-pdf/64/2/423/11724490/423\\_1\\_online.pdf](https://pubs.aip.org/asa/jasa/article-pdf/64/2/423/11724490/423_1_online.pdf).
- [49] Alexander Fu, David Lin, and Russell Miller. “Introduction to OpenACC”. In: (2016). URL: <https://www.cs.ucdavis.edu/~matloff/OpenACCDir/Intros158/FuLinMiller.pdf> (visited on 09/30/2019).
- [50] Alexander Lei et al. “OpenACC”. In: (2016). URL: <https://heather.cs.ucdavis.edu/OpenACCDir/Intros158/LeiMunteanuPapadopoulosSmith.pdf> (visited on 10/02/2019).
- [51] Alabama Supercomputer Center. “Introduction to GPU Programming with CUDA and OpenACC”. In: (2016). URL: <https://www.asc.edu/files/Introduction-GPU-programming-CUDA-and-OpenACC> (visited on 10/02/2019).
- [52] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. 1st ed. New York: Springer, 2004.
- [53] Rima Alaifari. “Ill-Posed Problems: From Linear to Nonlinear and Beyond”. In: *Harmonic and Applied Analysis: From Radon Transforms to Machine Learning*. Ed. by Filippo De Mari and Ernesto De Vito. Applied and Numerical Harmonic Analysis. Switzerland: Springer International Publishing, Dec. 2021, pp. 101–148. ISBN: 978-3-030-86664-8. DOI: [10.1007/978-3-030-86664-8\\_3](https://doi.org/10.1007/978-3-030-86664-8_3).
- [54] J. Kaipio et al. “Construction of anatomy-based priors with anisotropic characteristics with application to electrical impedance tomography”. In: *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 20. IEEE. Hong Kong, 1998, pp. 1032–1035.
- [55] Erick Darío León de Camargo. “Desenvolvimento de algoritmo de imagens absolutas de Tomografia por Impedância Elétrica para uso clínico”. PhD thesis. São Paulo, Brasil: Universidade de São Paulo, July 2013, p. 104.

- [56] Dieter Kraft. “Algorithm 733: TOMP — Fortran Modules for Optimal Control Calculations”. In: *ACM Transactions on Mathematical Software* 20 (Sept. 1, 1994), pp. 262–281. ISSN: 0098-3500. DOI: [10.1145/192115.192124](https://doi.org/10.1145/192115.192124). URL: <http://dl.acm.org/citation.cfm?id=192124> (visited on 05/24/2023).
- [57] Dieter Kraft. “*Modellierung, Simulation und Optimierung von Mehrkörpersystemen*”. Mechanical Engineering course notes (Fachhochschule München - Fachbereich Maschinenbau). Original url no more available. Munich, Germany, 2003. (Visited on 07/02/2003).
- [58] A. Griewank and Ph. L. Toint. “Partitioned Variable Metric Updates for Large Structured Optimization Problems”. In: *Numer. Math.* 39.1 (Feb. 1982), pp. 119–137. ISSN: 0029-599X. DOI: [10.1007/BF01399316](https://doi.org/10.1007/BF01399316).
- [59] Thomas Harvey Rowan. “Functional Stability Analysis of Numerical Algorithms”. UMI Order No. GAX90-31702. PhD thesis. USA, 1990.
- [60] A. Buckley and A. Lenir. “Algorithm 630: BBVSCG — A Variable-Storage Algorithm for Function Minimization”. In: *ACM Transactions on Mathematical Software* 11.2 (June 1985), pp. 103–119. ISSN: 0098-3500. DOI: [10.1145/214392.214395](https://doi.org/10.1145/214392.214395).
- [61] Jorge J. Moré, Burt S. Garbow, and Ken E. Hillstom. *User Guide for MINPACK-1*. Ed. by Argonne National Laboratory. Argonne National Laboratory. Illinois, USA, Aug. 1980. URL: <http://cds.cern.ch/record/126569/files/CM-P00068642.pdf> (visited on 05/24/2023).
- [62] William L. Goffe, Gary D. Ferrier, and John Rogers. “Global optimization of statistical functions with simulated annealing”. In: *Journal of Econometrics* 60.1 (1994), pp. 65–99. ISSN: 0304-4076. DOI: [10.1016/0304-4076\(94\)90038-8](https://doi.org/10.1016/0304-4076(94)90038-8).
- [63] Rosane Moniz Piccoli et al. “Optimization of polyhydroxyalkanoates bioproduction, based on a cybernetic mathematical model”. In: *Brazilian Journal of Chemical Engineering* 37.4 (Dec. 2020), pp. 643–652. DOI: [10.1007/s43153-020-00047-5](https://doi.org/10.1007/s43153-020-00047-5).
- [64] Arthur E. Bryson and Yu-Chi Ho. *Applied optimal control: optimization, estimation, and control*. 1st ed. New York: Hemisphere Publishing Corporation, 1975. ISBN: 978-0891162285. DOI: [10.1201/9781315137667](https://doi.org/10.1201/9781315137667).
- [65] Jennifer L. Mueller and Samuli Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2012. ISBN: 978-1-61197-233-7. DOI: [10.1137/1.9781611972344](https://doi.org/10.1137/1.9781611972344).
- [66] Activepoly. *Breast Cancer Cells*. <https://www.turbosquid.com/3d-models/object-breast-cancer-cells/1051750>. 2016. (Visited on 12/01/2022).

- [67] various. *MakeHuman – Open Source tool for making 3D characters*. <https://http://www.makehumancommunity.org/>. 2000–2022. (Visited on 12/01/2022).
- [68] Blender Online Community. *Blender - a 3D modelling and rendering package*. Stichting Blender Foundation, Amsterdam: Blender Foundation, 2022. URL: <http://www.blender.org> (visited on 05/24/2023).
- [69] Paolo Cignoni et al. “MeshLab: an Open-Source Mesh Processing Tool”. In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008. ISBN: 978-3-905673-68-5. DOI: [10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136](https://doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136).
- [70] various. *MakeHuman*. <https://en.wikipedia.org/wiki/MakeHuman>. 2022. (Visited on 12/01/2022).
- [71] Rafael Mikio Nakanishi. “Comparação de Dados Experimentais e de Simulação Numérica de Sinais de Ultrassom (Comparison of Experimental Data and Numeric Simulation of Ultrasound Signals)”. Brazilian. MA thesis. Escola Politécnica, 2019.
- [72] Steven G. Johnson. *Notes on Perfectly Matched Layers (PMLs)*. July 19, 2008. URL: <https://math.mit.edu/~stevenj/18.369/spring09/pml.pdf> (visited on 12/14/2023).
- [73] M. Pérez-Liva et al. “Time domain reconstruction of sound speed and attenuation in ultrasound computed tomography using full wave inversion”. In: *The Journal of the Acoustical Society of America* 141.3 (Mar. 2017), pp. 1595–1604. ISSN: 0001-4966. DOI: [10.1121/1.4976688](https://doi.org/10.1121/1.4976688).
- [74] Olivier Roy et al. “Ultrasound breast imaging using frequency domain reverse time migration”. In: *Medical Imaging 2016: Ultrasonic Imaging and Tomography*. Ed. by Neb Duric and Brecht Heyde. Vol. 9790. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Apr. 2016, 97900B. DOI: [10.1117/12.2218366](https://doi.org/10.1117/12.2218366).
- [75] Neb Duric et al. “Clinical breast imaging with ultrasound tomography: A description of the SoftVue system”. In: *The Journal of the Acoustical Society of America* 135.4 (Apr. 2014), pp. 2155–2155. ISSN: 0001-4966. DOI: [10.1121/1.4876990](https://doi.org/10.1121/1.4876990).
- [76] Frank Natterer. “An Algorithm for the Fully Nonlinear Inverse Scattering Problem at Fixed Frequency”. In: *Journal of Computational Acoustics* 9 (Sept. 2001), pp. 935–940. DOI: [10.1016/S0218-396X\(01\)00104-2](https://doi.org/10.1016/S0218-396X(01)00104-2).

- [77] Francis Duck. “Ultrasound – The First Fifty Years”. In: *Medical Physics International Journal*. Ed. by Slavik Tabakov, Perry Sprawls, and Geoffre Ibbott. United Kingdom: The International Organization for Medical Physics (IOMP), Jan. 5, 2021, pp. 470–498.
- [78] Joseph Woo. *A short History of the development of Ultrasound in Obstetrics and Gynecology*. Dec. 14, 2023. URL: <https://www.ob-ultrasound.net/history.html> (visited on 12/13/2023).
- [79] Rao V. Dukkipati. *Matlab: An Introduction With Applications*. New Age International, 2010. ISBN: 9788122429206.
- [80] Willi Hock and Klaus Schittkowski. *Test Examples for Nonlinear Programming Codes*. 1st ed. Lecture Notes in Economics and Mathematical Systems. Germany: Springer Berlin, Heidelberg, 1981. 182 pp. ISBN: 978-3-540-10561-9. DOI: [10.1007/978-3-642-48320-2](https://doi.org/10.1007/978-3-642-48320-2).
- [81] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Classics in applied mathematics. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019. ISBN: 9781611975598. DOI: [10.1137/1.9781611975604](https://doi.org/10.1137/1.9781611975604).
- [82] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. 2e. Springer series in operations research. New York, NY, USA: Springer, 2006. ISBN: 0-387-98793-2.
- [83] Robert B. Wilson. “A simplicial algorithm for concave programming”. PhD thesis. Cambridge, MA, USA: Harvard Business School, 1963. URL: <https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37372525> (visited on 04/14/2023).
- [84] Shih P. Han. *A Globally Convergent Method for Nonlinear Programming*. Tech. rep. Ithaca, NY, USA: Cornell University, 1975. URL: <https://dl.acm.org/doi/book/10.5555/867358> (visited on 04/03/2023).
- [85] M. J. D. Powell. “A fast algorithm for nonlinearly constrained optimization calculations”. In: *Numerical Analysis*. Ed. by G. A. Watson. Vol. 630. Lecture Notes in Mathematics. Springer, 1978.
- [86] Klaus Schittkowski. “The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function. Part 1: Convergence analysis”. In: *Numerische Mathematik* 38 (1982), pp. 83–114. DOI: [10.1007/BF01395810](https://doi.org/10.1007/BF01395810).

- [87] Klaus Schittkowski. “The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function. Part 2: An efficient implementation with linear least squares subproblems”. In: *Numerische Mathematik* 38 (1982), pp. 115–127. DOI: [10.1007/BF01395811](https://doi.org/10.1007/BF01395811).
- [88] J. E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1996. DOI: [10.1137/1.9781611971200](https://doi.org/10.1137/1.9781611971200).
- [89] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995. ISBN: 9780898713565. URL: <https://books.google.com.br/books?id=AEwDbHp50FgC>.
- [90] Jorge Nocedal and Michael L. Overton. “Projected Hessian Updating Algorithms for Nonlinearly Constrained Optimization”. In: *SIAM Journal on Numerical Analysis* 22.5 (1985), pp. 821–850. DOI: [10.1137/0722050](https://doi.org/10.1137/0722050).
- [91] Philip E. Gill and Walter Murray. “Numerically Stable Methods for Quadratic Programming”. In: *Math. Program.* 14.1 (Dec. 1978), pp. 349–372. ISSN: 0025-5610. DOI: [10.1007/BF01588976](https://doi.org/10.1007/BF01588976).
- [92] Roland Zdeněk Bulirsch. *Die Mehrzielmethode zur numerischen Lösung von nicht-linearen Randwertproblemen und Aufgaben der optimalen Steuerung*. Tech. rep. Deutsches Zentrum für Luft- und Raumfahrt (DLR), 1971.
- [93] Richard J. Hanson and Charles L. Lawson. “Extensions and Applications of the Householder Algorithm for Solving Linear Least Squares Problems”. In: *Mathematics of Computation* 23.108 (1969), pp. 787–812. ISSN: 00255718, 10886842. URL: <http://www.jstor.org/stable/2004965> (visited on 04/16/2023).
- [94] A. Cline. *The Transformation of a Quadratic Programming Problem into Solvable Form*. Tech. rep. ICASE 75-14. Hampton, Virginia: NASA Langley Research Center, 1975.
- [95] C. Di Loreto et al. *Il cancro della mammella*. pdf. Rome, Mar. 2019. URL: [https://www.aimac.it/scarica-libretto.php?id=5&file=05\\_Mammella](https://www.aimac.it/scarica-libretto.php?id=5&file=05_Mammella) (visited on 04/07/2020).



# APPENDIX A – ACOUSTIC FINITE ELEMENTS WITH DISSIPATION

Here the FEM equations with viscous damping are derived. For simplicity no volume sources are considered in the domain,  $q \equiv 0$ , and an unidirectional wave is assumed, therefore the complex  $k' = \beta - i\alpha$  (see equation 2.11 and section 2.1.2.1 on page 27) holds. An harmonic problem is governed by the Helmholtz equation:

$$\nabla^2 p(x, y, z) + k'^2 \cdot p(x, y, z) = 0 , \quad (\text{Ap-A.1})$$

where  $\beta = \omega/c_{ps}$ , the real part of  $k'$ , is the propagation wavenumber and  $c_{ps}$  the phase speed.

In the same way of section 2.1.5, an equivalent formulation for equation Ap-A.1 is:

$$\int_V \tilde{p} (\nabla^2 p + k'^2 p) dV = 0 \quad (\text{Ap-A.2})$$

where  $\tilde{p}$  is a weighting and  $V$  is the domain.

The weighted residuals equation Ap-A.2 can be rewritten as

$$\int_V (\vec{\nabla} \tilde{p} \vec{\nabla} p) dV - k'^2 \int_V (\tilde{p} p) dV = \int_V \left[ \frac{\partial}{\partial x} \left( \tilde{p} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left( \tilde{p} \frac{\partial p}{\partial y} \right) + \frac{\partial}{\partial z} \left( \tilde{p} \frac{\partial p}{\partial z} \right) \right] dV \quad (\text{Ap-A.3})$$

and, using the divergence theorem and equation 2.33 for the right side integral, being  $\Omega$  the boundary surface of the domain  $V$ ,

$$\int_V (\vec{\nabla} \tilde{p} \vec{\nabla} p) dV - k'^2 \int_V (\tilde{p} p) dV = - \int_{\Omega} (i\rho_0 \omega \tilde{p} \vec{v} \vec{n}) d\Omega . \quad (\text{Ap-A.4})$$

Now, introducing in this equation the expanded  $k'^2$ ,

$$k'^2 = (\beta - i\alpha).(\beta - i\alpha) = \beta^2 - \alpha^2 - 2i\alpha\beta, \quad (\text{Ap-A.5})$$

the equation [Ap-A.4](#) can be written as

$$\int_V \left( \vec{\nabla} \tilde{p} \vec{\nabla} p \right) dV + (\alpha^2 - \beta^2 + 2i\alpha\beta) \int_V (\tilde{p}p) dV = - \int_{\Omega} (i\rho_0\omega\tilde{p}\vec{v}\vec{n}) d\Omega. \quad (\text{Ap-A.6})$$

The second integral term from equation [Ap-A.6](#) can be written as

$$c^2(\alpha^2 - \beta^2 + 2i\alpha\beta) \int_V \frac{1}{c^2} (\tilde{p}p) dV = (c^2(\alpha^2 + 2i\alpha\beta) - \omega^2) \int_V \frac{1}{c^2} (\tilde{p}p) dV, \quad (\text{Ap-A.7})$$

remembering that  $\beta = \omega/c_{ps}$  and dropping the subindex  $ps$  from the phase speed of the wave.

From equation [An-B.23](#) from annex [B](#) we have

$$\int_V \frac{1}{c^2} (\tilde{p}p) dV = \{\tilde{p}_i\}^T [M] \{\hat{p}_i\} \quad (\text{Ap-A.8})$$

and therefore equation [Ap-A.7](#) can be written as

$$\begin{aligned} (c^2(\alpha^2 + 2i\alpha\beta) - \omega^2) \int_V \frac{1}{c^2} (\tilde{p}p) dV = \\ (c^2(\alpha^2 + 2i\alpha\beta) - \omega^2) \{\tilde{p}_i\}^T [M] \{\hat{p}_i\}, \end{aligned} \quad (\text{Ap-A.9})$$

Equation [An-B.28](#) let us write

$$- \int_{\Omega} (i\rho_0\omega\tilde{p}\vec{v}\vec{n}) d\Omega = - \int_{\Omega_v} (i\rho_0\omega\tilde{p}\vec{v}_n) d\Omega - \int_{\Omega_p} (i\rho_0\omega\tilde{p}\vec{v}\vec{n}) d\Omega - \int_{\Omega_z} (i\rho_0\omega\tilde{p}\vec{A}\vec{p}) d\Omega. \quad (\text{Ap-A.10})$$

Using equations [An-B.29](#), [An-B.35](#) and [An-B.37](#) in the first, second and third integral terms in equation [Ap-A.10](#), we have

$$- \int_{\Omega} (i\rho_0\omega\tilde{p}\vec{v}\vec{n}) d\Omega = \{\tilde{p}_i\}^T \{V_{ni}\} + \{\tilde{p}_i\}^T \{P_i\} - i\omega \{\tilde{p}_i\}^T [C] \{\hat{p}_i\}. \quad (\text{Ap-A.11})$$

With equations [An-B.18](#), [Ap-A.9](#) and [Ap-A.11](#), equation [Ap-A.6](#) can be written as

$$\begin{aligned} \{\tilde{p}_i\}^T [K] \{\hat{p}_i\} + (c^2(\alpha^2 + 2i\alpha\beta) - \omega^2) \{\tilde{p}_i\}^T [M] \{\hat{p}_i\} = \\ \{\tilde{p}_i\}^T \{V_{ni}\} + \{\tilde{p}_i\}^T \{P_i\} - i\omega \{\tilde{p}_i\}^T [C] \{\hat{p}_i\}. \end{aligned} \quad (\text{Ap-A.12})$$

Rearranging the terms, this equation is

$$\begin{aligned} \{\tilde{p}_i\}^T \left( [K] + (c^2(\alpha^2 + 2i\alpha\beta) - \omega^2) [M] + i\omega [C] \right) \{\hat{p}_i\} = \\ \{\tilde{p}_i\}^T (\{V_{ni}\} + \{P_i\}) . \end{aligned} \quad (\text{Ap-A.13})$$

Since this equation is valid for *any* value of the weighted residuals  $\{\tilde{p}_i\}$ , equation [Ap-A.13](#) reduces to

$$\left( [K] + (c^2(\alpha^2 + 2i\alpha\beta) - \omega^2) [M] + i\omega [C] \right) \{\hat{p}_i\} = \{V_{ni}\} + \{P_i\} . \quad (\text{Ap-A.14})$$

and, recalling that  $\beta = \omega/c$ ,

$$\boxed{\left( [K] + (c^2\alpha^2 + 2i.c.\alpha\omega - \omega^2) [M] + i\omega [C] \right) \{\hat{p}_i\} = \{V_{ni}\} + \{P_i\}} . \quad (\text{Ap-A.15})$$

Since the mass, damping and stiffness matrices  $[M]$ ,  $[C]$  and  $[K]$  in equation [Ap-A.15](#) are calculated in exactly the same way as in equation [An-B.41](#), it becomes evident that the consideration of a viscous loss given by an exponential decay  $e^{-\alpha x}$ , with an absorption coefficient  $\alpha$  and distance  $x$  from a reference position, only affects the factor that multiplies the mass matrix in the system of equations of the FEM problem. This factor increases by  $c^2\alpha^2 + 2i.c.\alpha\omega$  from the original  $-\omega^2$  of the lossless problem. For consideration of volume sources in the domain it suffices to add  $\{Q_i\}$  to the forcing terms to the right of equation [Ap-A.15](#) – see also equations [An-B.27](#) and [An-B.42](#).

## APPENDIX B – EXECUTION TIMES: OPENMP *VERSUS* NUMBER OF THREADS

Tables 10, 11 and 12 show results for a mesh with 1,061,828 tetrahedrons (180,729 nodes), simulated with the FE ultrasound program and using from the first to the last thread available in 3 different computers.

The tests are automated by a Bash script and run without user intervention, in a virtual desktop in personal computers or, for the supercomputer, started in a remote terminal (ssh) with NOHUP activated, then the remote terminal was closed.

It is apparent that, when using all the threads in the CITI<sup>1</sup> supercomputer, the execution time escalates. Values for tables 10, 11 are the mean of 3 repetitions of each test for each selected number of threads; they were made in computers with no other loads at the time and the values for each repetition agreed very well<sup>2</sup>, therefore no more repetitions were done. For table 12 8 repetitions were done and the best value was selected for the table; with this computer there was greater dispersion among the results, especially above 42 cores – this is a multi-user machine, and its load clearly had varied during the tests<sup>3</sup>, therefore longer tests – those using more cores – were most impacted.

---

<sup>1</sup>See footnote 2 at page 64.

<sup>2</sup>Most of the variations within 10 seconds or less.

<sup>3</sup>Each complete test started with 1 core and proceeded until the last one available; for table 12 the tests ranged from 9h58m to 17h24m, and four of them lasted less than 10h35m in total run time.

**Table 10** – Execution times for a mesh with 1,061,828 tetrahedrons and 180,729 nodes in a computer with 8 threads. Notebook with an Intel Core™ i7-3610QM CPU, clock 2.30 GHz, 4589.72 BogoMips, SSD disk.

| number of threads | execution time (s) |
|-------------------|--------------------|
| 1                 | 3586               |
| 2                 | 2030               |
| 3                 | 1524               |
| 4                 | 1278               |
| 5                 | 1270               |
| 6                 | 1267               |
| 7                 | 1209               |
| 8                 | 1207               |

**Table 11** – Execution times for a mesh with 1,061,828 tetrahedrons and 180,729 nodes in a computer with 12 threads. Computer with an Intel Core™ i7-8700K CPU, clock 3.70 GHz, 7399.70 BogoMips, SSD disk.

| number of threads | execution time (s) |
|-------------------|--------------------|
| 1                 | 2190               |
| 2                 | 1201               |
| 3                 | 909                |
| 4                 | 767                |
| 5                 | 740                |
| 6                 | 711                |
| 7                 | 695                |
| 8                 | 695                |
| 9                 | 709                |
| 10                | 699                |
| 11                | 699                |
| 12                | 680                |

**Table 12** – Execution times for a mesh with 1,061,828 tetrahedrons and 180,729 nodes in a computer with 48 threads; value shown is the best from 8 repetitions. Computer with virtualized Intel processors, 6584.52 BogoMips.

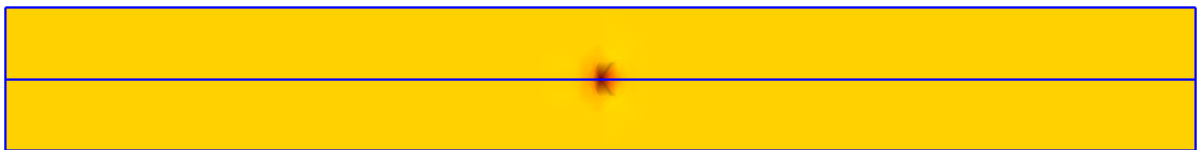
| threads | time (s) | threads | time (s) | threads | time (s) | threads | time (s) |
|---------|----------|---------|----------|---------|----------|---------|----------|
| 1       | 3396     | 13      | 451      | 25      | 344      | 37      | 366      |
| 2       | 1838     | 14      | 436      | 26      | 355      | 38      | 372      |
| 3       | 1362     | 15      | 417      | 27      | 362      | 39      | 370      |
| 4       | 1106     | 16      | 396      | 28      | 365      | 40      | 376      |
| 5       | 901      | 17      | 378      | 29      | 369      | 41      | 383      |
| 6       | 777      | 18      | 363      | 30      | 373      | 42      | 385      |
| 7       | 701      | 19      | 356      | 31      | 363      | 43      | 408      |
| 8       | 624      | 20      | 352      | 32      | 361      | 44      | 429      |
| 9       | 590      | 21      | 346      | 33      | 356      | 45      | 470      |
| 10      | 544      | 22      | 346      | 34      | 351      | 46      | 640      |
| 11      | 503      | 23      | 342      | 35      | 359      | 47      | 1669     |
| 12      | 472      | 24      | 341      | 36      | 365      | 48      | 5604     |

## APPENDIX C – ECHOES IN THE UPPER AND LOWER SURFACES OF THE THORAX SECTION

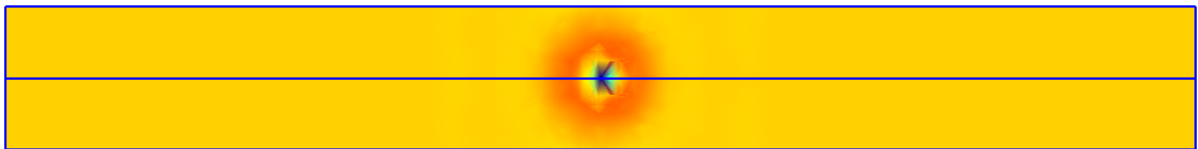
This appendix shows that echoes do appear at the upper and lower planes of the thin thorax section from 4.3, in the absence of an ALID wrapping. The same can happen with the simplified breast model.

These figures show the echo evaluation model, with a single transducer in the geometric center of the model, which has only water inside.

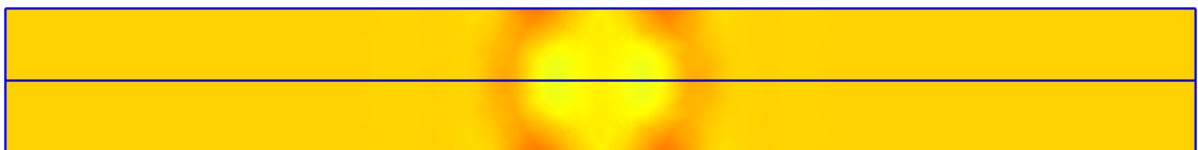
**Figure 40** – Temporal evolution of a pulse in the echo evaluation model, here without an ALID wrapping, evidences echoes at the upper and lower planes of the thorax section. Central cut view perpendicular to the  $Y$  axis of the model. Pressure values in Pa.



(a) instant  $t = 6,0 \times 10^{-6}$  s.



(b) instant  $t = 1,2 \times 10^{-5}$  s.



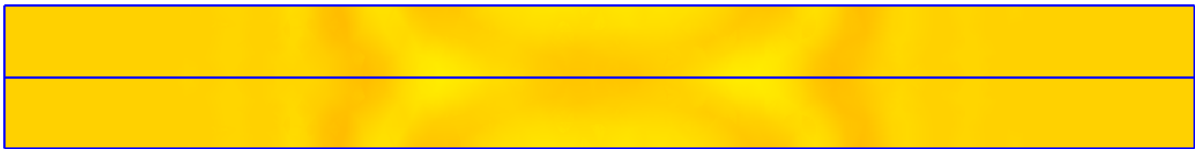
(c) instant  $t = 1,8 \times 10^{-5}$  s. The pulse just arrived to the upper and lower planes.



(d) instant  $t = 2,4 \times 10^{-5}$  s. The two echoes, upper and lower, cross the center of the model, where the transducer is.



(e) instant  $t = 3,0 \times 10^{-5}$  s. Each echo reaches the plane opposed to the original echo.



(f) instant  $t = 3,6 \times 10^{-5}$  s. The echoes, “re-echoing”, cross again the center of the model.



(g) instant  $t = 4,2 \times 10^{-5}$  s. Echoes ready to reach by the third time the sectional planes.



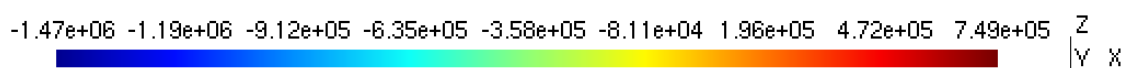
(h) instant  $t = 4,8 \times 10^{-5}$  s. Echoes ready to reach by the third time the central transducer.



(i) instant  $t = 5,4 \times 10^{-5}$  s. Echoes in the third cross of the central region, each time more attenuated.



(j) instant  $t = 6,0 \times 10^{-5}$  s. The process continues, and the initial pulse has not yet reached the side borders of the model.





## ANNEX A – ULTRASOUND IN MEDICINE: THE BEGINNING

Here a shortened version of [77], with addends from [9] and [78], is presented.

Ultrasound, the sound with higher frequencies than the human ear can hear, can be linked to physiological acoustics, which was actively investigated in the nineteenth century; in 1862 Hermann von Helmholtz, a physiologist and physicist, published ‘*Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik*’ (On the Sensations of Tone as a Physiological Basis for the Theory of Music, or simply “Sensations of Tone”), the main text on the subject. However it took time to agree on the upper frequency limit for audible sounds on 20 kHz, thank to estimates from Rudolph Koenig in 1899 and years later confirmed by Franz Schultze. In 1903 Wilhelm Altberg was the first to use radiation pressure to measure the intensity of a sound wave; this was a key experiment directly relevant to later developments in the physics of medical ultrasound. In 1907 Altberg generated ultrasonic pulses of 1 mm wavelength or 300 kHz; in 1911 Lebedev and Neklepajev explained that Altberg could not detect higher frequencies due to absorption, with a coefficient dependent on the square of the frequency. All these works were done in open air.

In 1912 Lewis Fry Richardson was the first to propose underwater detection using ultrasound<sup>1</sup>. With the first World War and the advent of submarine warfare, Constantin Chilowsky proposed, in 1915, an ultrasonic system for submarine detection, and in two years Paul Langevin and his team developed a working pulse-echo system. The efforts resulted in a capacitor transmitter that emitted 50 ms pulses of 100 MHz ultrasound, at a rate of 2 Hz, and a carbon microphone placed at the focus of a parabolic mirror

---

<sup>1</sup>He submitted two patents – the first for airborne, the second for underwater ultrasound – for detection of icebergs, soon after 15th April 1912, when the Titanic sank. However their ideas could not be implemented at the time.

of 120 cm diameter, to receive the echoes. There were many technical and operational challenges, and hydrophones remained the best option to detect submarines by the sound they emitted. Other groups developed quartz transducers and were able, in 1918, to succeed with a signaling distance of 8 km: in pulse-echo mode, clear echoes were obtained for the first time from a submarine. The development of new transducer technologies was fundamental for the medical use of ultrasound.

In September 1927 appeared the first publication citing “biological effects” and ultrasound, where a high power ultrasound was employed – it even melted glass in experiments. The German physicist Reimar Pohlman was the first to establish the use of ultrasound for therapy in 1938. He specifically rejected the high intensity approach to cancer therapy being explored at the time, and proposed that ultrasound at lower intensities could be used to stimulate healing through a combination of thermal and mechanical processes. His first experiments were designed to select the most appropriate frequency to use. From measurements at 800 kHz and at 2.4 MHz, he concluded that he would gain greater tissue penetration at the lower of these frequencies, which he selected for the clinical evaluation. Attenuation at 800 kHz, expressed as half-value thickness for intensity, was 6.8 cm for adipose tissue, and 3.6 cm for muscle. For mixed muscle and fat, the half-value thickness at 2.4 MHz was 1.5 cm, compared with 4.9 cm at 800 kHz. He found no difference between the attenuation of tissue of adults and children, within an error limit of about 10%. He confirmed observations that the frequency dependence of the absorption coefficient did not follow the square-law dependence expected from classical theory of viscous loss. The first clinical trials of his ultrasound therapy were carried out before the end of 1938, and favorable results were reported for some neurological and neuro-muscular disorders.

The first uses of ultrasound in medicine, in the late 1940s and early 1950s, involved industrial ultrasound equipments, known as “flaw detectors”, and an engineer of physicist working with the physicians. The echoes from ultrasonic pulses were captured in oscilloscopes, and heart motions could be seen for the first time.

The first “Flaw detecting device and measuring instrument” was from Floyd Firestone, in 27<sup>th</sup> May 1940, an invention which “pertains to a device for detecting the presence of inhomogeneities of density or elasticity in materials” and “may also be used for the measurement of dimensions of objects, and is particularly useful where one of the faces to which the measurement extends is inaccessible”.

In Germany, Heinrich Netheler, a physician at the Luebeck-South Hospital in Hamburg, was operating in 1945 a small repair facility for medical equipments at the Hamburg

university hospital at Eppendorf and had a mission of developing inventive medical products. Professor Hansen, his superior, suggested to him in that year to develop *an ultrasonic tomographic equipment for medical use* basing on the concept of the RADAR but, due to a lack of funds right after the war, the equipment designs had not reached the stage of actual fabrication [78].

George Ludwig was the first to report the velocity of sound transmission in animal soft tissues; it was determined to be between 1490 and 1610 meters per second, with a mean value of 1540 m/s. However, as he did research for the US Navy, his works were not released to the public domain until October 1949 by the United States Department of Defense.

By the mid 1950s, bibliographic listings of works on ultrasonic physics and engineering applications had totaled more than 6,000.

Smaller and better transducers were being assembled from the newer piezoceramics barium titanate after the mid 1940s. They were replaced by lead zirconate-titanate (PZT) when it was discovered in 1954.

John Julian Wild started investigations (after 1945) with ultrasound waves on the thickness of the bowel wall in various surgical conditions, such as paralytic ileus and obstruction. Working with Donald Neal, an engineer, Wild published their work in 1950 on uni-directional A-mode ultrasound investigations into the thickness of surgical intestinal material and later on the properties of gastric malignancies. They noted that malignant tissue was more echogenic than benign tissue and the former could be diagnosed from their density and failure to contract and relax.

John Reid, an electrical engineer, was engaged through a grant from the National Cancer Institute as the sole engineer to build and operate Wild's ultrasonic apparatus. The device which they first used was an ultrasonic instrument which had been designed by the U.S. Navy for training pilots in the use of the radar, with which it was possible to practice "flying" over a tank of water covering a small scale map of enemy territory. "We have a tissue radar machine scaled to inches instead of miles by the use of ultrasound". Wild and Reid soon built a linear hand-held B-mode instrument, a formidable technical task in those days, and were able to visualize tumors by sweeping from side to side through breast lumps. The instrument operated at a frequency of 15 megahertz. In 1952 they published the landmark paper: "Application of Echo-Ranging Techniques to the Determination of Structure of Biological Tissues". In May 1953 they produced real-time images at 15 megahertz of cancerous growths of the breast. By 1956, Wild and Reid had

examined 117 cases of breast pathology with their linear real-time B-mode instrument and had started work on colon tumor diagnosis and detection.

Pioneering designs in electronic circuits were made in conjunction with the development of the B-scan; these included the pulse-echo generator circuitry, the limiter and log amplification circuitry and the demodulator and time gain compensation circuits. Early B-scanners employed threshold detection which registered echoes on a phosphorous coated oscilloscope screen as dots of light. A “storage” or “bi-stable” cathode-ray tube was used. Echoes above a certain amplitude are displayed as dots of constant intensity and echoes of a lesser amplitude below the threshold were not depicted. The images were often recorded unsatisfactorily on ordinary 35mm photographic film. The situation was improved with the advent of the black and white “peel-apart” Polaroid instant film which had become available from 1959 onward in both England, Europe and the United States. Although there was good representation of size, shape and position, the images did not depict differences in echo amplitude. It was apparent that some sort of gray-scaling was imminently necessary to expand the diagnostic capability and accuracy of a B-scan.

The analog scan converter, which was hailed as a profound “invention” in the field of image processing and reproduction, used a silicon oxide/silicon target that acted as a capacitance matrix and was then raster-scanned by an electron beam that “read” it and displayed the information on a standard television monitor unit. By doing so, computer-processor technology, which was just up and coming at about the same time, could be applied to process the signal. Images could then be scaled, calipers moved and applied on-screen (something that had changed entirely the way measurements are made), gray-scaling applied to the images and the resultant image recorded on a variety of media including videotape, emulsion films and thermal printer devices.

With early scanners made of vacuum tubes there was often problem of drifts and numeric instability which required periodic re-calibration. Another important consideration in the early days was the assumed velocity of ultrasounds in human tissues. Different centers adopted different values, ranging from 1540 m/s, 1580 m/s and 1600 m/s, and this had to be stated in their reports and papers. By about the early 1980s and after several meetings of the experts, the ultrasound community throughout the world settled for 1540 m/s. This universal acceptance is important because with different velocity calibrations measurements like the biparietal diameter will end up with different normals.

Attenuation characteristics, axial and lateral resolution, fluid enhancement characteristics etc. were also periodically tested with “tissue phantoms” to determine the imaging

quality and accuracy of the scanner. Hospital physicists are often pre-occupied with such calibrations. With the later advent of array real-time scanners these tissue phantoms progressively became obsolete and by about the late-80s they have completely disappeared from the scene and few are bordered by the figure of 1540 m/s.

The concept of the multi-element linear electronic arrays was first described by Werner Buschmann in an ophthalmologic application in 1964 in East Berlin. His probe, fabricated in collaboration with Kretztechnik AG consisted of 10 small transducers mounted on an arc-shaped apparatus to fit over the eye. Buschmann's transducer however never became very functional in a clinical setting and did not go into serial production. It was Martin H. Wilcox, founder and engineer at the Advanced Diagnostic Research Corporation (ADR, a company founded in 1972 in Tempe, Arizona), who designed and produced one of the earliest commercially available models of a linear-array real-time scanner in 1973 and very much set the standard for subsequent designs to follow. The array contained 64 crystals in a row (3 times the number in the earlier cardiac counterparts and 3 times as long and wide), fabricated with the best material available and in the best acoustic configurations and using "stepping" crystals techniques.

The M-mode (time-motion) display was first described by Inge Edler and Hellmuth Hertz in Lund, Sweden in 1954 using a modified metal-flaw detector from Siemens of Germany.

The Doppler principle was first described over 100 years ago by Christian Andreas Doppler in Austria in 1842. Medical applications of the Ultrasonic Doppler techniques were first implemented by Shigeo Satomura and Yasuhara Nimura at the Institute of Scientific and Industrial Research in Osaka, Japan in 1955 for the study of cardiac valvular motion and pulsations of peripheral blood vessels.

Further development led to 2D color flow imaging. Marco Brandestini and his team at the University of Washington in 1975 obtained blood-flow images using a 128-point multi-gated pulsed Doppler system, where velocity waveforms and flow images were encoded in color and superimposed on M-mode and gray scale 2-D anatomical images.

Image quality of real-time ultrasound scanners made steady improvements during the mid 1980's to early 90's secondary to the increasing versatility and affordability in microprocessor technology. Nevertheless it was not until the early to mid 1990's that more substantial enhancements in image quality were seen.

# ANNEX B – FINITE ELEMENT FORMULATION FOR SOUND WAVES

In this annex, based on a seminar from Desmet and Vandepitte [16], a short introduction to the Finite Elements Method is presented, followed by the development of equations of acoustical properties for the elements.

Here, the acoustical problem is considered to be *uncoupled*, that is, there are no other objects or surfaces, neither at the boundaries nor in the interior of the region considered in the problem, which are influenced by the sound pressure waves, nor e. g. mechanical vibrations that could generate pressure interferences in the media (besides the pressure source).

We start recalling the Helmholtz equation 2.31 and the three possible boundary conditions, equations 2.32-2.34 from chapter 2.1.

## B.1 Basic equation and boundary conditions

The pressure field  $p$  in any point  $(x, y, z)$  in a limited domain  $V$ , closed by a surface  $\Omega$ , generated by an external and distributed source  $q$  with frequency  $\omega = 2\pi f$ , is governed by the second order Helmholtz equation:

$$\nabla^2 p(x, y, z) + k^2 \cdot p(x, y, z) = -i\rho_0\omega \cdot q(x, y, z) , \quad (\text{An-B.1})$$

where  $k = \omega/c = 2\pi f/c$  is the wave number,  $c$  the sound speed and  $\rho_0$  the mass density of the fluid.

In order to define the pressure field in  $V$ , one boundary condition must be specified in each position of the closed border  $\Omega = \Omega_p \cup \Omega_v \cup \Omega_z$ :

- imposed pressure:

$$p = \bar{p} \text{ in } \Omega_p, \quad (\text{An-B.2})$$

- imposed normal velocity:

$$v_n = \frac{i}{\rho_0 \omega} \frac{\partial p}{\partial n} = \bar{v}_n \text{ in } \Omega_v, \quad (\text{An-B.3})$$

- imposed acoustic impedance:

$$p = \bar{Z} v_n = \frac{v_n}{A} = \frac{i \bar{Z}}{\rho_0 \omega} \frac{\partial p}{\partial n} = \frac{i}{\rho_0 \omega \bar{A}} \frac{\partial p}{\partial n} \text{ in } \Omega_Z. \quad (\text{An-B.4})$$

For the tomographic problem in this thesis the entire closed border  $\Omega$  is kept at ambient static pressure ( $\bar{p} = 0$ ), as already justified in section 2.1.4, and as pressure source  $q(x, y, z)$  there are the FE nodes corresponding to the transducers; the values are applied to these nodes via the penalty method [20]. Alternatively, those nodes – with known pressures – can be removed from the unknowns and the system of rearranged equations solved only for the remaining nodes with unknown values.

## B.2 Weighted residual formulation of the Helmholtz equation

One equivalent formulation for An-B.1, possible due to the concept of weighted residuals, allows us to affirm that the steady state pressure field in a domain  $V$  has the integral equation

$$\int_V \tilde{p} (\nabla^2 p + k^2 p + i \rho_0 \omega q) dV = 0 \quad (\text{An-B.5})$$

satisfied for any weighting  $\tilde{p}$ , limited and unambiguously defined in the volume  $V$  and in the boundary surface  $\Omega$ .

Using the chain rule of derivation,

$$\begin{aligned} \frac{\partial}{\partial x} \left( \tilde{p} \frac{\partial p}{\partial x} \right) &= \frac{\partial \tilde{p}}{\partial x} \frac{\partial p}{\partial x} + \tilde{p} \frac{\partial^2 p}{\partial x^2} \\ \frac{\partial}{\partial y} \left( \tilde{p} \frac{\partial p}{\partial y} \right) &= \frac{\partial \tilde{p}}{\partial y} \frac{\partial p}{\partial y} + \tilde{p} \frac{\partial^2 p}{\partial y^2} \\ \frac{\partial}{\partial z} \left( \tilde{p} \frac{\partial p}{\partial z} \right) &= \frac{\partial \tilde{p}}{\partial z} \frac{\partial p}{\partial z} + \tilde{p} \frac{\partial^2 p}{\partial z^2} \end{aligned} \quad (\text{An-B.6})$$

the weighted residuals equation [An-B.5](#) can be rewritten as

$$\begin{aligned} \int_V \left[ \frac{\partial}{\partial x} \left( \tilde{p} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left( \tilde{p} \frac{\partial p}{\partial y} \right) + \frac{\partial}{\partial z} \left( \tilde{p} \frac{\partial p}{\partial z} \right) \right] dV - \\ \int_V \left( \frac{\partial \tilde{p}}{\partial x} \frac{\partial p}{\partial x} + \frac{\partial \tilde{p}}{\partial y} \frac{\partial p}{\partial y} + \frac{\partial \tilde{p}}{\partial z} \frac{\partial p}{\partial z} \right) dV + \\ \int_V k^2 \tilde{p} p dV + \int_V i \rho_0 \omega \tilde{p} q dV = 0 \end{aligned} \quad (\text{An-B.7})$$

or yet in the form

$$\begin{aligned} \int_V \left( \vec{\nabla} \tilde{p} \vec{\nabla} p \right) dV - \omega^2 \int_V \left( \frac{1}{c^2} \tilde{p} p \right) dV = \\ \int_V i \rho_0 \omega \tilde{p} q dV + \int_V \left[ \frac{\partial}{\partial x} \left( \tilde{p} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left( \tilde{p} \frac{\partial p}{\partial y} \right) + \frac{\partial}{\partial z} \left( \tilde{p} \frac{\partial p}{\partial z} \right) \right] dV . \end{aligned} \quad (\text{An-B.8})$$

According to the divergence theorem, the integral of the normal component of a vector field  $\vec{\phi}$  in a closed surface  $\Omega$  equals the integral of the field divergent, integrated in the volume  $V$  which is bound by the surface  $\Omega$ :

$$\int_V \left( \vec{\nabla} \vec{\phi} \right) dV = \int_{\Omega} \left( \vec{\phi} \vec{n} \right) d\Omega \quad (\text{An-B.9})$$

where  $\vec{n}$  is the normal unit vector which points outward of the volume  $V$ .

Application of this theorem to the last integral in [\(An-B.8\)](#) results in

$$\begin{aligned} \int_V \left[ \frac{\partial}{\partial x} \left( \tilde{p} \frac{\partial p}{\partial x} \right) + \frac{\partial}{\partial y} \left( \tilde{p} \frac{\partial p}{\partial y} \right) + \frac{\partial}{\partial z} \left( \tilde{p} \frac{\partial p}{\partial z} \right) \right] dV = \int_V \left[ \vec{\nabla} \left( \tilde{p} \vec{\nabla} p \right) \right] dV = \\ \int_{\Omega} \left( \tilde{p} \frac{\partial p}{\partial n} \right) d\Omega = - \int_{\Omega} \left( i \rho_0 \omega \tilde{p} \vec{v} \vec{n} \right) d\Omega . \end{aligned} \quad (\text{An-B.10})$$

Substitution of [\(An-B.10\)](#) in [\(An-B.8\)](#) results in the “weak form” of the weighted residual formulation of the Helmholtz equation:

$$\int_V \left( \vec{\nabla} \tilde{p} \vec{\nabla} p \right) dV - \omega^2 \int_V \left( \frac{1}{c^2} \tilde{p} p \right) dV = \int_V \left( i \rho_0 \omega \tilde{p} q \right) dV - \int_{\Omega} \left( i \rho_0 \omega \tilde{p} \vec{v} \vec{n} \right) d\Omega . \quad (\text{An-B.11})$$



### B.3 Approximation of the field variables

In the finite elements method, the domain  $V$  is discretized in small subdomains  $V_e$ , the “*finite elements*”, and some number of *nodes*  $n_e$  which are defined at some points in those elements. In each element the distribution of the field variable, in the acoustic case the pressure  $p$ , is approximated by an expansion  $\hat{p}$  in terms of a number  $n_p$  of *form functions*  $N_i^e$ , defined only inside an elementary domain  $V_e$ , and the nodal values  $a_i$ :

$$p(x, y, z) \approx \hat{p}(x, y, z) = \sum_{i=1}^{n_p} N_i^e(x, y, z) a_i \quad (x, y, z) \in V_e . \quad (\text{An-B.12})$$

For the linear elements usually employed, tetrahedrons and hexahedrons, the nodes are defined in its vertexes and the number of shape functions equals the number of nodes,  $n_p = n_e$ . Each shape function  $N_i^e$  is defined with unit value in the node  $i$  of the element and zero in the other nodes. So, each contribution  $a_i$  in the pressure expansion (An-B.12) represents the pressure approximation  $\hat{p}$  at node  $i$  of the element,

$$\hat{p}(x, y, z) = \sum_{i=1}^{n_p} N_i^e(x, y, z) \cdot \hat{p}_i \quad (x, y, z) \in V_e . \quad (\text{An-B.13})$$

Based on the shape functions of the elements  $N_i^e$ , which are defined in the elements  $V_e$ , global shape functions  $N_i$  can be built for all the fluid domain  $V$ . In each domain  $V_e$  in which the node  $i$  belongs, a global shape function  $N_i$  is identical to the elementary function  $N_i^e$ , being zero in all other elementary domains. So, the expansion of the global pressure can be defined as

$$\hat{p}(x, y, z) = \sum_{i=1}^{n_f} N_i(x, y, z) \cdot \hat{p}_i = [N] \{ \hat{p}_i \} \quad (x, y, z) \in V , \quad (\text{An-B.14})$$

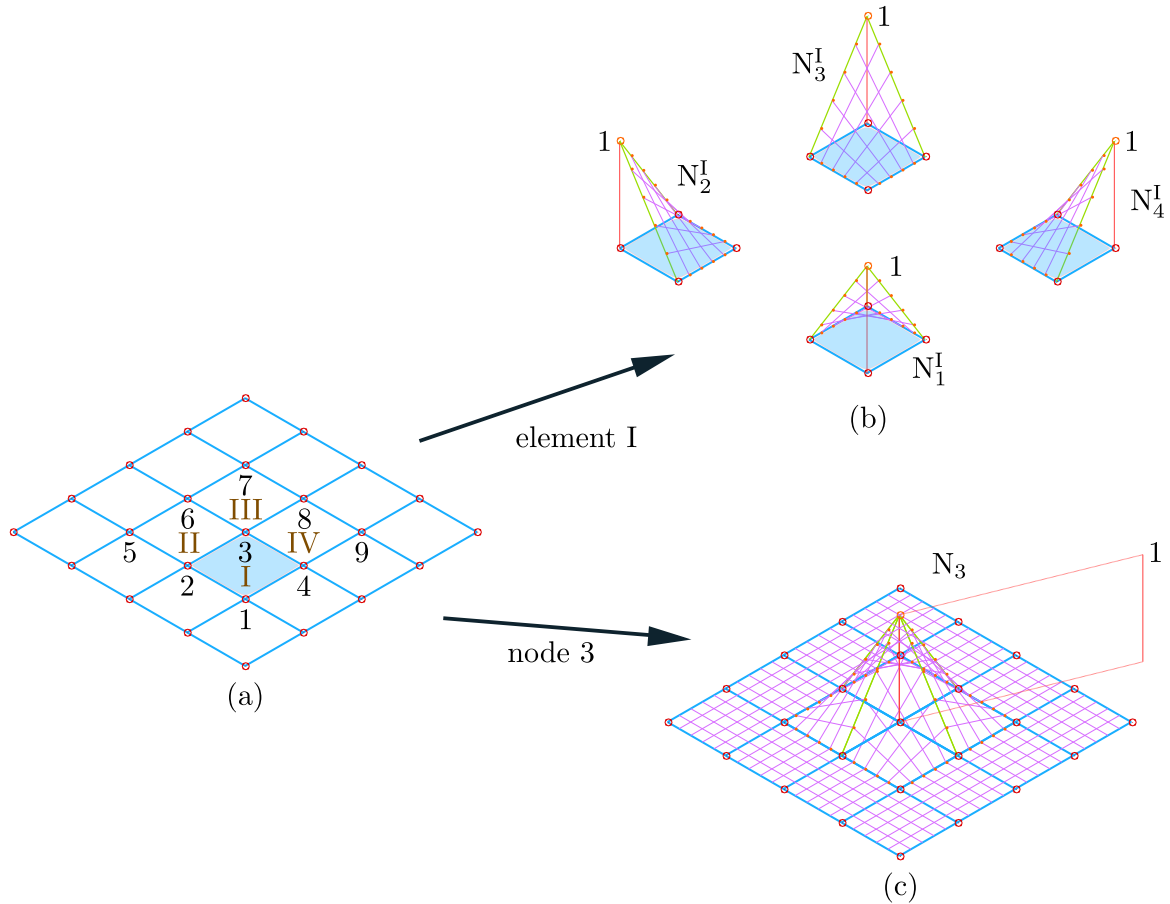
where  $n_f$  is the total of nodes in the discretization,  $[N]$  is a vector ( $1 \times n_f$ ) of global shape functions and  $\{ \hat{p}_i \}$  a vector ( $n_f \times 1$ ) of unknown nodal pressure values.

The approximation of the pressure gradient is:

$$\vec{\nabla} p \approx \vec{\nabla} \hat{p} = \begin{bmatrix} \frac{\partial \hat{p}}{\partial x} \\ \frac{\partial \hat{p}}{\partial y} \\ \frac{\partial \hat{p}}{\partial z} \end{bmatrix} = [\partial][N] \{ \hat{p}_i \} = [B] \{ \hat{p}_i \} , \quad (\text{An-B.15})$$

where  $[\partial]$  is a vector ( $3 \times 1$ ) of gradient operators and  $[B]$  is a matrix ( $3 \times n_f$ ) of components of the gradient of the global shape functions.

**Figure 41** – Principles of the finite element modeling: (a) discretization of a bi-dimensional region in nodes and elements, (b) linear shape functions for a given element, (c) global shape function, also linear.



Source: modified from [16].

Some concepts of element discretization and shape functions are depicted in figure 41 for a two-dimensional problem, discretized into linear rectangular elements.

## B.4 Uncoupled finite element model for acoustic problems

Determination of unknown pressures  $\hat{p}_i$  in expansions (An-B.14) and (An-B.15) is based in the weak formulation of weighted residues (An-B.11) from the Helmholtz equation.

In the most used method of weighted residues, the Galerkin method, the weighting function  $\tilde{p}$  and the gradient vector in equation An-B.11 are expanded with the same type

of shape functions used for the pressure and for its gradient, that is:

$$\tilde{p}(x, y, z) = \sum_{i=1}^{n_f} N_i(x, y, z) \tilde{p}_i = [N] \{\tilde{p}_i\} \quad (x, y, z) \in V \quad (\text{An-B.16})$$

$$\vec{\nabla} \tilde{p} = [\partial][N]\{\tilde{p}_i\} = [B]\{\tilde{p}_i\} . \quad (\text{An-B.17})$$

Substitution of expansions (An-B.14)—(An-B.17) in the integral terms of the weak formulation (An-B.11) leads to the definitions, as will be shown, of an *acoustic stiffness matrix*, an *acoustic mass matrix*, *acoustic excitation vectors* and an *acoustic damping matrix*. Combining those elements results in an acoustical finite element model, which when solved gives the nodal pressures  $\hat{p}_i$ .

### B.4.1 Acoustical stiffness matrix

For the first integral term on the left of (An-B.11), substitutions give:

$$\begin{aligned} \int_V (\vec{\nabla} \tilde{p} \vec{\nabla} \hat{p}) dV &= \int_V \left( ([B] \{\tilde{p}_i\})^T ([B] \{\hat{p}_i\}) \right) dV \\ &= \{\tilde{p}_i\}^T \left( \int_V ([B]^T [B]) dV \right) \{\hat{p}_i\} = \{\tilde{p}_i\}^T [K] \{\hat{p}_i\} \end{aligned} \quad (\text{An-B.18})$$

where  $[\cdot]^T$  indicates matrix transposition and  $[K]$  is a  $(n_f \times n_f)$  matrix.

Keeping an analogy with structural finite element models, the matrix  $[K]$  is called acoustical *stiffness* matrix, although, indeed, it represents an inverse of mass or mobility matrix, relating pressure to an acceleration.

The matrix element  $K_{i,j}$  at line  $i$  and column  $j$  in that matrix is

$$K_{i,j} = \int_V \left( \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right) dV . \quad (\text{An-B.19})$$

Recalling, the global shape functions  $N_i$  and  $N_j$ , as its derivatives, have non-zero values only in the domain of its respective elements, in other words, those that contain nodes  $i$  and  $j$ . So, integration in the volume of (An-B.19) is restricted to the integration in the domain of those elements. As the total volume integral can be considered the sum of the integrals of each elementary volume, and as the shape functions of each of those volumes and the global shape function (in the domain of elementary volumes) are

identical, the element  $K_{i,j}$  can be expressed as

$$K_{i,j} = \sum_{e=1}^{m_{i,j}} \left( \int_{V_e} \left( \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} + \frac{\partial N_i^e}{\partial z} \frac{\partial N_j^e}{\partial z} \right) dV \right), \quad (\text{An-B.20})$$

where  $m_{i,j}$  is the number of elements which simultaneously contain the nodes  $i$  and  $j$ .

As a given node belongs to a few elements which are defined by this node and some more adjacent nodes, only some elements  $K_{i,j}$  are not zero. Therefore the  $[K]$  matrix is sparse.

Benefiting from this sparsity, the *practical calculation* of the stiffness matrix can be made in a very efficient way. Confining the volumetric integral of (An-B.18) to the domain of one element, it can be written

$$\begin{aligned} \int_{V_e} (\vec{\nabla} \tilde{p} \vec{\nabla} \hat{p}) dV &= \{\tilde{p}_i^e\}^T \left( \int_{V_e} \left( ([\partial] [N^e])^T ([\partial] [N^e]) \right) dV \right) \{\hat{p}_i^e\} \\ &= \{\tilde{p}_i^e\}^T \left( \int_{V_e} \left( [B^e]^T [B^e] \right) dV \right) \{\hat{p}_i^e\} = \{\tilde{p}_i^e\}^T [K^e] \{\hat{p}_i^e\}, \end{aligned} \quad (\text{An-B.21})$$

where  $[N^e]$  is a  $(1 \times n_p)$  vector of shape functions of the element and  $\{\hat{p}_i^e\}$  is a  $(n_p \times 1)$  vector of unknown nodal pressures of the element. The elements of the associated  $(n_p \times n_p)$  stiffness matrix  $[K^e]$  are

$$K_{i,j}^e = \int_{V_e} \left( \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} + \frac{\partial N_i^e}{\partial z} \frac{\partial N_j^e}{\partial z} \right) dV. \quad (\text{An-B.22})$$

Notice that, as all the shape functions of an element have non-zero values in the domain  $V_e$ , each elementary stiffness matrix is full.

Calculation of the global stiffness matrix  $[K]$  can be done in two phases. On the first, all elementary stiffness matrices are evaluated. On the second phase, each non-zero element  $K_{i,j}$  of the global stiffness is evaluated according to (An-B.20) by simple addition of the values of (An-B.22) in the corresponding elements of the (global) matrix. With an adequate numbering of the nodes in the finite element discretization, the non-zero values in the global matrix will lay in a small band around the matrix diagonal, resulting in a sparse banded diagonal matrix.

### B.4.2 Acoustical mass matrix

In the same way, the second integral term to the left of (An-B.11) can be expressed as

$$\begin{aligned} -\omega^2 \int_V \left( \frac{1}{c^2} \tilde{p} p \right) dV &= -\omega^2 \{\tilde{p}_i\}^T \left( \int_V \left( \frac{1}{c^2} [N]^T [N] \right) dV \right) \{\hat{p}_i\} \\ &= -\omega^2 \{\tilde{p}_i\}^T [M] \{\hat{p}_i\} , \end{aligned} \quad (\text{An-B.23})$$

where  $[M]$  is a  $(n_f \times n_f)$  matrix. By analogy with a structural model, the matrix  $[M]$  is called as the acoustical mass matrix, although it represents compressibility, relating pressure to displacement.

As in the case of the global stiffness matrix, the *practical evaluation* of this sparse diagonal banded matrix is based in the assembly of elementary mass matrices, according to

$$M_{i,j} = \int_V \left( \frac{1}{c^2} N_i N_j \right) dV = \sum_{e=1}^{m_{i,j}} \left( \int_{V_e} \left( \frac{1}{c^2} N_i^e N_j^e \right) dV \right) . \quad (\text{An-B.24})$$

### B.4.3 Acoustical excitation vectors

The first term to the right in (An-B.11) can be written as

$$\int_V (i\rho_0\omega\tilde{p}q) dV = \{\tilde{p}_i\}^T \left( \int_V (i\rho_0\omega [N]^T q) dV \right) = \{\tilde{p}_i\}^T \{Q_i\} , \quad (\text{An-B.25})$$

where  $\{Q_i\}$  is the *acoustical source vector*, with dimension  $(n_f \times 1)$ .

When the distribution  $q$  of the external acoustic sources is confined, for instance, to a point force source  $\{\bar{q}_i\}$ , placed on node  $i$ , the distribution  $q$  of the source is

$$q(x, y, z) = \bar{q}_i \delta(x_i, y_i, z_i) , \quad (\text{An-B.26})$$

where  $\delta$  is the Dirac delta function on node  $i$ . Therefore the vector is

$$\{Q_i\} = i\rho_0\omega \left( \int_V (\bar{q}_i [N]^T \delta) dV \right) . \quad (\text{An-B.27})$$

As long as the node  $i$  is not on the surface of  $V$ , all the  $(n_f \times 1)$  components of the source vector are zero, except for row  $i$ , which is equal to  $i\rho_0\omega\bar{q}_i$ .

The second integral term to the right in (An-B.11) allows the consideration of bound-

ary conditions. As the contour surface integral  $\Omega$  can be regarded as the sum of the integrals of the component surfaces  $\Omega_v$ ,  $\Omega_Z$  and  $\Omega_p$ , and as normal velocity and normal impedance on equations [An-B.3](#) and [An-B.4](#) must be satisfied in  $\Omega_v$  and  $\Omega_Z$ , respectively, this second integral term can be expressed as

$$-\int_{\Omega_v} (i\rho_0\omega\tilde{p}\bar{v}_n)d\Omega - \int_{\Omega_p} (i\rho_0\omega\tilde{p}\bar{v}\bar{n})d\Omega - \int_{\Omega_Z} (i\rho_0\omega\tilde{p}\bar{A}\hat{p})d\Omega . \quad (\text{An-B.28})$$

Substitution from [\(An-B.16\)](#) in the first term of [\(An-B.28\)](#) results in

$$-\int_{\Omega_v} (i\rho_0\omega\tilde{p}\bar{v}_n)d\Omega = \{\tilde{p}_i\}^T \left( \int_{\Omega_v} (-i\rho_0\omega [N]^T \bar{v}_n) d\Omega \right) = \{\tilde{p}_i\}^T \{V_{ni}\} . \quad (\text{An-B.29})$$

The component on row  $i$  of the  $(n_f \times 1)$  vector of imposed velocities  $\{V_{ni}\}$  is, therefore,

$$V_{ni} = \int_{\Omega_v} (-i\rho_0\omega N_i \bar{v}_n) d\Omega . \quad (\text{An-B.30})$$

The contour surface of an element is the union of all its faces. The surface of a fluid element of linear tetrahedron or linear hexahedron types, for instance, is the union of four or six faces, respectively. For these compatible elements, the value of the global shape function  $N_i$  in a given element face will not be zero only when the node  $i$  is part of this face. Consequently, the value of the global shape function  $N_i$  in the surface  $\Omega_v$  and the vector component  $V_{ni}$  are not zero only in the nodes that make up the contour surface  $\Omega_v$ .

Thus, the practical evaluation of the imposed velocity vector is based in the expression of its component element,

$$V_{ni} = -i\rho_0\omega \sum_{e=1}^{m_{vi}} \sum_{f=1}^{f_{vi}^e} \left( \int_{\Omega_e^f} (N_i^e \bar{v}_n) d\Omega \right) , \quad (\text{An-B.31})$$

where  $m_{vi}$  is the number of elements in which node  $i$  is located in its  $f_{vi}^e$  elementary faces  $\Omega_e^f$ , which make up the contour surface  $\Omega_v$ .

The normal velocity in a certain point of an elementary face is usually specified by an expansion via shape functions, comparable to the expression [\(An-B.14\)](#),

$$\bar{v}_n(x, y, z) = \{n\}^T [N_{ve}^f] \{\bar{v}_e^f\} \quad (x, y, z) \in \Omega_e^f , \quad (\text{An-B.32})$$

where the  $(3 \times 1)$  vector  $\{n\}$  consists in the components  $x$ ,  $y$  and  $z$  of the unit vector normal to the considered face. The matrix

$$[N_{ve}^f] = \begin{bmatrix} N_1^e & 0 & 0 & \cdots & N_{n_v}^e & 0 & 0 \\ 0 & N_1^e & 0 & \cdots & 0 & N_{n_v}^e & 0 \\ 0 & 0 & N_1^e & \cdots & 0 & 0 & N_{n_v}^e \end{bmatrix} \quad (\text{An-B.33})$$

of dimension  $(3 \times n_v)$  contains the pressure shape functions of the  $n_v$  nodes from the considered face of the element. In every of those nodes, the components  $x$ ,  $y$  and  $z$  of the fluid velocity are specified, resulting in the  $(3n_v \times 1)$  vector

$$\{\bar{v}_e^f\}^T = \{\bar{v}_{x1} \quad \bar{v}_{y1} \quad \bar{v}_{z1} \quad \cdots \quad \bar{v}_{xn_v} \quad \bar{v}_{yn_v} \quad \bar{v}_{zn_v}\} . \quad (\text{An-B.34})$$

Substitution of (An-B.16) in the second term from (An-B.28) results in

$$- \int_{\Omega_p} (i\rho_0\omega\tilde{p}\vec{v}\vec{n}) d\Omega = \{\tilde{p}_i\}^T \left( \int_{\Omega_v} (-i\rho_0\omega [N]^T \vec{v}\vec{n}) d\Omega \right) = \{\tilde{p}_i\}^T \{P_i\} . \quad (\text{An-B.35})$$

Thanks to the characteristics of the global shape functions  $N_i$ , the component in the row  $i$  of the  $(n_f \times 1)$  vector of imposed pressure  $\{P_i\}$ ,

$$P_i = \int_{\Omega_p} (-i\rho_0\omega N_i \vec{v}\vec{n}) d\Omega , \quad (\text{An-B.36})$$

is not zero only when the node  $i$  lies in the contour surface  $\Omega_p$ . As this expression does not consider the contour imposed pressures, (An-B.2), but instead a alternative formulation with velocity, density and frequency, those pressures will enter the finite element model as will be shown later.

#### B.4.4 Acoustical damping matrix

The third term from (An-B.28) can be written as

$$\begin{aligned} - \int_{\Omega_Z} (i\rho_0\omega\tilde{p}\bar{A}\hat{p}) d\Omega &= -i\omega \{\tilde{p}_i\}^T \left( \int_{\Omega_Z} (\rho_0\bar{A} [N]^T [N]) d\Omega \right) \{\hat{p}_i\} \\ &= -i\omega \{\tilde{p}_i\}^T [C] \{\hat{p}_i\} , \end{aligned} \quad (\text{An-B.37})$$

where  $[C]$  is the  $(n_f \times n_f)$  matrix of acoustical damping that is caused by the contour impedance conditions.

The element  $C_{i,j}$  in the row  $i$  and column  $j$  in this matrix is

$$C_{i,j} = \int_{\Omega_Z} (\rho_0 \bar{A} N_i N_j) d\Omega . \quad (\text{An-B.38})$$

As is with the stiffness and mass matrices, the damping matrix is sparse, because its  $C_{i,j}$  elements are only non-zero when both nodes  $i$  and  $j$  lie in at least one face of the element which belongs to the contour surface  $\Omega_Z$ .

The practical evaluation of the non-zero matrix elements is, therefore, based in the expression

$$C_{i,j} = \sum_{f=1}^{m_{Z_{i,j}}} \left( \int_{\Omega_e^f} (\rho_0 \bar{A} N_i^e N_j^e) d\Omega \right) , \quad (\text{An-B.39})$$

where  $m_{Z_{i,j}}$  is the quantity of elementary faces  $\Omega_e^f$  in which the nodes  $i$  and  $j$  are contained and are parts of the contour surface  $\Omega_Z$ .

Specification of normal admittance is generally considered constant in each elementary face of  $\Omega_Z$ .

### B.4.5 Acoustical model by finite elements

Substitution of expressions (An-B.18), (An-B.23), (An-B.25), (An-B.29), (An-B.35) and (An-B.37) in equation An-B.11, the weak form of the weighted residual formulation of the Helmholtz equation, including the boundary conditions (An-B.3) and (An-B.4), results in

$$\{\tilde{p}_i\}^T ([K] + i\omega [C] - \omega^2 [M]) \{\hat{p}_i\} = \{\tilde{p}_i\}^T (\{Q_i\} + \{V_{ni}\} + \{P_i\}) . \quad (\text{An-B.40})$$

As the weighted residual formulation must be valid for any type of weighting, that is, any value set of the shape function  $\{\tilde{p}_i\}$  (see equation An-B.16), a set of  $n_f$  equations in the  $n_f$  nodal unknowns of the pressure approximation  $\hat{p}_i$  is obtained,

$$([K] + i\omega [C] - \omega^2 [M]) \{\hat{p}_i\} = \{Q_i\} + \{V_{ni}\} + \{P_i\} . \quad (\text{An-B.41})$$

The row  $i$  in this matrix equation shows the weighted residual formulation where the global shape function  $N_i$ , associated to node  $i$ , is used as weighting function  $\tilde{p}$ .

The contour imposed pressure, (An-B.2), is not yet included in the matrix equation An-B.41. This consideration is usually made by assigning the imposed pressure at each



nodal point of the contour surface  $\Omega_p$  to the corresponding nodal unknown  $\hat{p}_i$ .

When such assignment is made to the  $n_{\bar{p}}$  nodes in  $\Omega_p$ , there will remain  $n_a = n_f - n_{\bar{p}}$  pressure approximations  $\hat{p}_i$  still unknown. This means that  $n_{\bar{p}}$  equations should be eliminated in (An-B.41) in order to the system of equations to be determined.

This is generally done by elimination of each row from (An-B.41) where the weighted residuals formulation utilizes as weighting functions the global function of a node on the surface  $\Omega_p$ . Elimination from these equations, which orthogonalize the pressure error in regions close to  $\Omega_p$  in regard to the shape functions in that region, is motivated by the fact that the evaluation error is lesser than errors in other regions of the fluid domain, since the exact pressure values in nodes of the contour surface  $\Omega_p$  are assigned *a priori*.

Eliminating the adequate rows in (An-B.41) and moving the left terms which contain imposed pressures to the right side, results in the expression for the finite element model for the uncoupled acoustical problem,

$$([K_a] + i\omega [C_a] - \omega^2 [M_a]) \{p_i\} = \{F_{ai}\} , \quad (\text{An-B.42})$$

where the vector  $\{p_i\}$  with dimension  $(n_a \times 1)$  contains the nodal pressure unknowns approximations and where the acoustical stiffness, damping and mass matrices  $[K_a]$ ,  $[C_a]$  and  $[M_a]$  are now with dimension  $(n_a \times n_a)$ .

As the non-zero elements from  $\{P_i\}$  occur only in the eliminated equations from (An-B.41), the  $(n_a \times 1)$  vector of acoustical force  $\{F_{ai}\}$  contains the stiffness, damping and mass terms of the nodes where the imposed pressure was *a priori* assigned, and the contributions of the acoustical source (An-B.25) and imposed velocities vector (An-B.29) were assigned.

## ANNEX C – THE CENTRAL DIFFERENCE METHOD

The time evolution of pressures inside the domain is carried out not by an implicit integration in time, but using the central difference method.

This annex, based on [79], explains the method.

For a multi-degree of freedom system, with a mass matrix  $[M]$ , damping matrix  $[C]$ , stiffness matrix  $[K]$  and displacement vector  $\{X\}$ , the equation of motion is

$$[M]\{\ddot{X}\} + [C]\{\dot{X}\} + [K]\{X\} = \{F(t)\}, \quad (\text{An-C.1})$$

where  $\{\ddot{X}\}$  is the acceleration vector and  $\{\dot{X}\}$  the velocity vector.

Consider the time history of *one single displacement variable* of this system depicted in figure 42. The idea can be applied in the same way to the multi-variable system as well. Using the midpoint of the time interval  $\Delta t = h$  after and before the instant  $i$ , the velocity can be written as both

$$\dot{X}_{i+\frac{1}{2}} = \frac{X_{i+1} - X_i}{\Delta t} \quad (\text{An-C.2})$$

and

$$\dot{X}_{i-\frac{1}{2}} = \frac{X_i - X_{i-1}}{\Delta t}. \quad (\text{An-C.3})$$

With equations An-C.2 and An-C.3 the acceleration can be written as

$$\ddot{X}_i = \frac{\dot{X}_{i+\frac{1}{2}} - \dot{X}_{i-\frac{1}{2}}}{\Delta t}. \quad (\text{An-C.4})$$

Substitution of  $\dot{X}_{i+\frac{1}{2}}$  and  $\dot{X}_{i-\frac{1}{2}}$  from equations An-C.2 and An-C.3 into An-C.4 leads to

$$\ddot{X}_i = \frac{1}{\Delta t^2} (X_{i+1} - 2X_i + X_{i-1}). \quad (\text{An-C.5})$$

Extending these ideas to vector variables, it is now possible to express the velocity at

a given instant  $t$  as the mean value of  $\{\dot{X}_{i+1}\}$  and  $\{\dot{X}_{i-1}\}$  using equations analogous to [An-C.2](#) and [An-C.3](#), that is,

$$\{\dot{X}\}_t = \frac{1}{2\Delta t} \left( \{X\}_{t+\Delta t} - \{X\}_{t-\Delta t} \right) \quad (\text{An-C.6})$$

while, as with equation [An-C.5](#) for a single variable, the vector acceleration is

$$\{\ddot{X}\}_t = \frac{1}{\Delta t^2} \left( \{X\}_{t+\Delta t} - 2\{X\}_t + \{X\}_{t-\Delta t} \right). \quad (\text{An-C.7})$$

Substitution of equations [An-C.6](#) and [An-C.7](#) into equation [An-C.1](#), and rearranging and isolating terms at instant  $t + \Delta t$  to the left, gives

$$[M_{\text{eff}}]\{X\}_{t+\Delta t} = \{F_{\text{eff}}\}_t \quad (\text{An-C.8})$$

where the “effective mass” matrix is

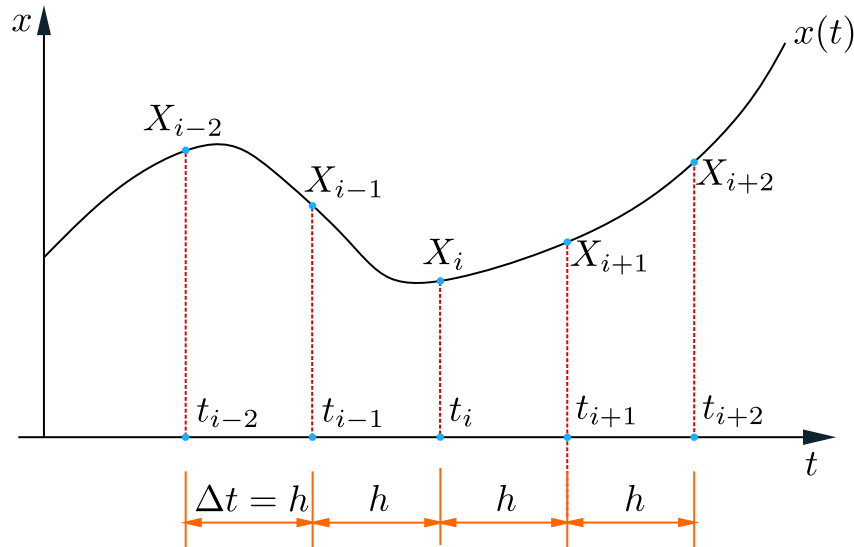
$$[M_{\text{eff}}] = \frac{1}{\Delta t^2}[M] + \frac{1}{2\Delta t}[C] \quad (\text{An-C.9})$$

and the “effective force” is

$$\{F_{\text{eff}}\}_t = \{F\}_t - \left( [K] - \frac{2}{\Delta t^2}[M] \right) \{X\}_t - \left( \frac{1}{\Delta t^2}[M] - \frac{1}{2\Delta t}[C] \right) \{X\}_{t-\Delta t}. \quad (\text{An-C.10})$$

Then, all the displacements  $\{X\}_{t+\Delta t}$  can be calculated using equation [An-C.8](#), based on current and past displacement values, and the velocities and accelerations calculated using equations [An-C.6](#) and [An-C.7](#).

**Figure 42** – Time grid in the central difference method.



# ANNEX D – MATHEMATICAL DETAILS ABOUT SLSQP, TOMP’S OPTIMIZER

This annex presents the rationale behind the SLSQP optimizer, which is available in the TOMP package [56]. This text is primarily based on [57], which is no longer available. It includes information from some of the sources cited by Kraft, the creator of TOMP, in this technical article. The text appears to be related to a Engineering course material.

The optimizer will solve the nonlinear programming problem

$$\min_{x \in R^n} f(x) \tag{An-D.1}$$

with each component of the vector  $x$  bounded by lower and upper limits

$$l_i \leq x_i \leq u_i, \quad i = 1, \dots, n \tag{An-D.2}$$

and also satisfying constraints as

$$g_j(x) = 0, \quad j = 1, \dots, m_e \tag{An-D.3}$$

and

$$g_j(x) \geq 0, \quad j = m_e + 1, \dots, m \tag{An-D.4}$$

with  $f : R^n \rightarrow R^1$ .

In our tomography problem both  $m_e$  from equation [An-D.3](#) and  $m$  from [An-D.4](#) have value 0.

## D.1 Sequential quadratic programming

The method for optimization is sequential quadratic programming, SQP, a efficient computational method to solve general nonlinear minimization problems like ([An-D.1](#)),

corroborated by extensive tests and reported by [80], apud [57].

### D.1.1 Interactive optimization

The solution should be sought iteratively, from the initial vector  $x^0$  of optimization parameters and going from iteration  $k$  to  $k + 1$  with a search direction  $d^k$  and step length  $\alpha^k$ ,

$$x^{k+1} = x^k + \alpha^k d^k . \quad (\text{An-D.5})$$

### D.1.2 Search direction

According to [57], there are two algorithms for determination of the search direction. The first is to solve a complete quadratic programming problem<sup>1</sup> in order to solve the optimality conditions, then there is a quadratic programming problem for the equality constraints alone. This is done by the behavior of a quadratic exterior point penalty function, and the equality constraint bounding the step size.

In the first approach the search direction  $d^k$  is determined by a quadratic programming subproblem, which is formulated by a quadratic approximation of the Lagrange function  $L(x, \lambda)$  of the problem (An-D.1),

$$L(x, \lambda) = f(x) - \sum_{j=1}^m \lambda_j g_j(x) , \quad (\text{An-D.6})$$

and a linear approximation of constraints  $g_j$ . This is a problem belonging to a standard form of quadratic programming:

$$\min_{d \in R^n} \frac{1}{2} d^T B^k d + \nabla f(x^k) d \quad (\text{An-D.7})$$

subject to

$$\nabla g_j(x^k) d + g_j(x^k) = 0 , \quad j = 1, \dots, m_e , \quad (\text{An-D.8})$$

$$\nabla g_j(x^k) d + g_j(x^k) \geq 0 , \quad j = m_e, \dots, m , \quad (\text{An-D.9})$$

---

<sup>1</sup>A quadratic program is the special case of linearly constrained optimization that occurs when the objective function is the quadratic function

$$F(x) = c^T x + \frac{1}{2} x^T G x ,$$

for some constant vector  $x$  and constant symmetric matrix  $G$ ; the gradient of this function is  $Gx + c$  and the Hessian is  $G$ . ([81], [82]).

Again, for the tomography algorithm, no constraints are needed besides the bounds of minimum and maximum allowed values.

where matrix  $B^k$  is symmetric and need to be determined. The gradients  $\nabla f(x)$  and  $\nabla g_j(x)$  are row vectors.

Wilson ([83]) suggested the search direction with

$$B = \nabla_{xx}^2 L(x, \lambda) . \quad (\text{An-D.10})$$

### D.1.3 Step size

Books on numerical methods like [35] draw our attention to the fact that the direction calculated by (An-D.7) has a strong analogy to the search direction while solving systems of (nonlinear) equations by Newton's method. Like there, the ideal step  $\alpha = 1$  is valid for nonlinear functions near a local optimum point, however the step has to be modified for vectors  $x^k$  far from the optimum, otherwise global convergence is not guaranteed.

It has been proved ([84], apud [57]) that a one-dimensional minimization of the non-differentiable exact penalty function

$$\phi(x; \rho) = f(x) + \sum_{j=1}^{m_e} \rho_j g_j(x) + \sum_{j=m_e+1}^m \rho_j \min [0, g_j(x)] \quad (\text{An-D.11})$$

as a merit function  $\varphi : R^1 \rightarrow R^1$

$$\varphi(\alpha) = \phi(x^k + \alpha d^k) , \quad (\text{An-D.12})$$

with both  $x^k$  and  $d^k$  fixed, leads to a step size  $\alpha$  that guarantees global convergence for values of penalty parameters  $\rho_j$  greater than a certain lower bound. Powell ([85]) proposed updating of the penalty parameters using the formula

$$\rho_j = \max \left[ \frac{1}{2}(\rho_{\bar{j}} + \mu_j), \mu_j \right], \quad j = 1, \dots, m . \quad (\text{An-D.13})$$

where  $\mu_j$  denotes the Lagrange multiplier of the  $j$ -th constraint in the quadratic subproblem and  $\rho_{\bar{j}}$  is the  $j$ -th penalty parameter in the previous iteration, starting with  $\rho_j^0 = 0$  for instance.

In view of possible difficulties in the line search of the non-differentiable merit function Schittkowski ([86], [87]), instead of (An-D.11), used the differentiable augmented Lagrange

function

$$\begin{aligned} \Phi(x, \lambda, \rho) = & f(x) - \sum_{j=1}^{m_e} \lambda_j g_j(x) \\ & - \frac{1}{2} \rho_j g_j(x)^2 - \sum_{j=m_e+1}^m \begin{cases} \lambda_j g_j(x) - \frac{1}{2} \rho_j g_j(x)^2, & \text{if } g_j(x) \leq \lambda_j / \rho_j, \\ \frac{1}{2} \lambda_j^2 / \rho_j, & \text{otherwise.} \end{cases} \end{aligned} \quad (\text{An-D.14})$$

### D.1.4 Update of the Matrix B

It is very important for the computational efficiency in practical applications of sequential quadratic programming not to evaluate the matrix  $B^k$  as in equation (An-D.10) at every iteration, but to use only first-order information to approximate the Hessian matrix of the Lagrange function. This is very common for unconstrained optimization for a long time and is known as quasi Newton methods. A very used update is the BFGS formula. Reference [88] gives a comprehensive overview on numerical methods for unconstrained optimization and the closely related field of nonlinear equations.

The analogue formula for the constrained case has been developed by Powell [85]. The difficulty in constrained optimization is that, unlike in unconstrained optimization,  $B^k$  need not remain positive definite for a positive definite initial estimate. Therefore Powell proposed this modification:

$$B^{k+1} = B^k + \frac{q^k (q^k)^T}{(p^k)^T q^k} - \frac{B^k p^k (p^k)^T B^k}{(p^k)^T B^k p^k}, \quad (\text{An-D.15})$$

with

$$p^k = x^{k+1} - x^k = \alpha^k d^k, \quad (\text{An-D.16})$$

and

$$q^k = \theta^k \eta^k + (1 - \theta^k) B^k p^k, \quad (\text{An-D.17})$$

where  $\eta^k$  is the difference in gradients of the Lagrange function

$$\eta^k = \nabla_x L(x^{k+1}, \lambda^k) - \nabla_x L(x^k, \lambda^k), \quad (\text{An-D.18})$$

and  $\theta^k$  is chosen as

$$\theta^k = \begin{cases} 1, & \text{if } (p^k)^T \eta^k \geq 0.2 (p^k)^T B^k p^k, \\ \frac{0.8 (p^k)^T B^k p^k}{(p^k)^T B^k p^k - (p^k)^T \eta^k}, & \text{otherwise.} \end{cases} \quad (\text{An-D.19})$$

which guarantees the condition

$$(p^k)^T q^k \geq 0.2(p^k)^T B^k p^k, \quad (\text{An-D.20})$$

which holds  $B^{k+1}$  positive definite within the linear manifold defined by the tangent planes to the active constraints at  $x^{k+1}$ . The multipliers  $\lambda^k$  are taken to be those of the quadratic subproblem at its solution. The choice of conditions (An-D.16)–(An-D.19) guarantees the updated  $B$  matrix (An-D.15) to remain positive definite for an arbitrary initial positive definite estimate of this matrix.

### D.1.5 Least Squares Subproblems

Schittkowski ([87], apud [57]) proposed to replace the quadratic programming subproblem (An-D.7) by a linear least squares subproblem, using a stable  $LDL^T$  factorization of the matrix  $B$ :

$$\min_{d \in \mathbb{R}^n} \left\| (D^k)^{1/2} (L^k)^T d + (D^k)^{-1/2} (L^k)^{-1} \nabla f(x^k)^T \right\| \quad (\text{An-D.21})$$

subject to

$$\nabla g_j(x^k) d + g_j(x^k) = 0, \quad j = 1, \dots, m_e, \quad (\text{An-D.22})$$

$$\nabla g_j(x^k) d + g_j(x^k) \geq 0, \quad j = m_e + 1, \dots, m, \quad (\text{An-D.23})$$

$$l_i \leq x_i \leq u_i, \quad i = 1, \dots, n. \quad (\text{An-D.24})$$

This subproblem can be solved using the linear least squares software of Lawson and Hanson [89].

### D.1.6 Motivation of SQP Methods

#### D.1.6.1 Optimality Conditions.

SQP-based methods are motivated as methods analog to Newton's method for solving systems of nonlinear equations. First we need the Kuhn-Tucker conditions for problem (An-D.7), a set of necessary optimality conditions:

$$\nabla_x L(x, \lambda) = 0, \quad (\text{An-D.25})$$

$$g_j(x) = 0, \quad j = 1, \dots, m_e, \quad (\text{An-D.26})$$

$$g_j(x) \geq 0, \quad j = m_e + 1, \dots, m, \quad (\text{An-D.27})$$



$$\lambda_j(x) \geq 0, \quad j = m_e + 1, \dots, m, \quad (\text{An-D.28})$$

$$g_j(x)\lambda_j(x) = 0, \quad j = m_e + 1, \dots, m. \quad (\text{An-D.29})$$

To start with the arguments equality-constrained problems (An-D.1) and (An-D.3) are considered first, and then inequality-constrained problems are treated introducing the notion of active constraint sets.

From the necessary conditions of the equality-constrained problem it follows that the condition

$$e(x, \lambda) = \begin{pmatrix} \nabla f(x) - A(x)\lambda \\ g(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (\text{An-D.30})$$

is satisfied, where  $A(x)$  such as

$$A^T(x) = \begin{pmatrix} \nabla g_1(x) \\ \vdots \\ \nabla g_{m_e}(x) \end{pmatrix} \quad (\text{An-D.31})$$

is the  $n \times m_e$  Jacobi matrix of equality constraints, assumed to be of full rank at the corresponding vector  $x$ .

Now Newton's method is applied to the solution of the system of  $n + m_e$  (usually nonlinear) equations (An-D.30) in the  $n + m_e$  unknowns  $(x, \lambda)^T$ . Let  $(x^k, \lambda^k)^T$  be the  $k$ -th iterate of the Newton process, defined by the equation

$$\nabla e(x^k, \lambda^k) \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} + \nabla e(x^k, \lambda^k) = 0 \quad (\text{An-D.32})$$

and the step

$$\begin{pmatrix} x^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix}, \quad (\text{An-D.33})$$

with

$$\nabla e(x^k, \lambda^k) = \begin{pmatrix} H(x^k, \lambda^k) & -A(x^k) \\ A(x^k)^T & 0 \end{pmatrix} \quad (\text{An-D.34})$$

the Jacobian of  $e(x, \lambda)$  with respect to  $(x, \lambda)^T$  within the  $k$ -th iteration, and

$$H(x^k, \lambda^k) = \nabla_x^2 f(x^k) - \sum_{j=1}^{m_e} \lambda_j^k \nabla_x^2 g_j(x^k) \quad (\text{An-D.35})$$

the Hessian matrix with respect to  $x$  of the Lagrange function  $L(x, \lambda)$ . Explicitly written,

equations (An-D.32) and (An-D.33) become

$$\begin{pmatrix} H(x^k, \lambda^k) & -A(x^k) \\ A(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} d \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} \Delta f(x^k) \\ g(x^k) \end{pmatrix}, \quad (\text{An-D.36})$$

and

$$x^{k+1} = x^k + d. \quad (\text{An-D.37})$$

Equation (An-D.36) can be solved based on matrix decomposition methods given by [90].

### D.1.6.2 Nonlinear Equations and Quadratic Programming.

It can be verified by direct evaluation of (An-D.25) that the conditions (An-D.36) and (An-D.37) are equivalent to the Kuhn-Tucker conditions of the quadratic programming problem with pure equality constraints (An-D.7) and (An-D.8):

$$\min_{d \in R^n} \frac{1}{2} d^T B^k d + \nabla f(x^k) d \quad (\text{An-D.38})$$

subject to

$$\nabla g_j(x^k) d + g_j(x^k) = 0, \quad j = 1, \dots, m_e \quad (\text{An-D.39})$$

with  $B^k = H(x^k, \lambda^k)$ .

This equivalence is the reason for the choice of the search direction (An-D.7)–(An-D.9). This is especially convenient in the inequality constrained case, where no practical implementation exists to solve the corresponding problem (An-D.36)–(An-D.37).

## D.1.7 Active Set Strategy

A complete solution for the equality-constrained quadratic programming problem (An-D.38)–(An-D.39) is given in the next section. Here, a technique for solving the general QP problem, with inequalities included, will be presented in the context of primal methods. The matrix  $B$  will be assumed to be positive definite throughout; the slightly more complex case when  $B$  is indefinite is treated by Gill and Murray ([91]) and by Bunch and Kaufman (see [92]).

First suppose a feasible iterate  $x^k$  is given, that is, a point satisfying the constraints (An-D.8) and (An-D.9). If such point is not available, it has to be provided by a suitable starting procedure, for instance a phase I method of the simplex algorithm. Now the

active set is an index set  $I_a^k$  consisting of the indices of all  $t^k$  active constraints at the point  $x^k$ :

$$I_a^k = \{j = 1, \dots, m_e\} \cup \{j = m_e + 1, \dots, m \mid a_j^T(x^k)d - g_j^k = 0\} \quad (\text{An-D.40})$$

with  $g_j^k = g_j(x^k)$ . The inactive inequalities  $a_j^T(x^k)d - g_j^k > 0$  will be temporarily disregarded.

Now the equality constrained quadratic programming problem with active constraints

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^T B^k d + \nabla f(x^k) d \quad (\text{An-D.41})$$

subject to

$$\nabla g_j(x^k) d + g_j(x^k) = 0, \quad j = 1, \dots, t^k, \quad (\text{An-D.42})$$

must be solved; let the solution be  $(d^k, \lambda^k)^T$ . Next a step is taken into this direction according to (An-D.5)

$$x^{k+1} = x^k + \alpha^k d^k. \quad (\text{An-D.43})$$

with the restrictions

$$f(x^k + \alpha^k d^k) < f(x^k), \quad (\text{An-D.44})$$

and

$$\alpha^k \leq \hat{\alpha}^k = \begin{cases} \min \frac{g_j - a_j^T x^k}{a_j^T d^k}, & \text{if } a_j^T d^k < 0 \text{ for some } j \notin I_a^k, \\ +\infty, & \text{if } a_j^T d^k \geq 0 \text{ for all } j \notin I_a^k. \end{cases} \quad (\text{An-D.45})$$

Note that  $\hat{\alpha}^k$  will have a positive value under the conditions of (An-D.45), as the index  $j$  is not within the active set.

If  $a_j^T d^k \geq 0$ , any positive step along  $d^k$  will not violate the inactive constraint  $j$ . On the other hand, if  $a_j^T d^k < 0$ , there is a step  $\alpha_j$  which activates the constraint  $j$ :  $g_j - a_j^T(x^k + \alpha_j d^k) = 0$ . This is the reasoning behind condition (An-D.45).

As in unconstrained optimization, this algorithm requires sufficient decrease of the cost function, stated in relation (An-D.44), as to prove its global convergence. For further information see reference [81, pages 168–170].

Conditions (An-D.44) and (An-D.45) may leave the active set unaltered or enlarged, according to whether  $\alpha^k < \hat{\alpha}^k$  or  $\alpha^k = \hat{\alpha}^k$ . As a third possibility the deletion of a constraint has to be considered. This is the case when the optimality conditions, especially (An-D.28)

$$\lambda_j(x) \geq 0, \quad j = m_e + 1, \dots, m, \quad (\text{An-D.46})$$

are not satisfied for the active inequality constraints. This means that the active set is not yet correct. The minimum of the cost function is sought in a subspace of too small dimension. If the Lagrange multipliers of the current solution of (An-D.41) and (An-D.42) are assumed to be sufficient accurate estimates, then condition (An-D.46) can be used as a criterion to delete a binding constraint, for instance that one with the smallest (negative) value  $l$ :

$$l = \arg \min_j (\lambda_j(x) < 0, \quad j = m_e + 1, \dots, m) . \quad (\text{An-D.47})$$

## D.2 Quadratic programming algorithm

Four main algorithms for the solution of the quadratic programming core problem of the sequential quadratic programming method are known: a primal method based on the direct formulation of the problem, a primal/dual approach based on a least squares transformation, a dual method which finds the active set beginning from an unconstrained problem and methods based on linear complementarity problems. Kraft ([56], [57]) prefers the primal/dual approach.

### D.2.1 Problem Equivalence

It can be shown that the general quadratic programming problem (An-D.7)

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T G x + h^T x \quad (\text{An-D.48})$$

subject to

$$A^T x \geq b , \quad (\text{An-D.49})$$

and

$$C^T x = d , \quad (\text{An-D.50})$$

is equivalent to the following linear least squares formulation (An-D.21)

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|E x - f\|^2 \quad (\text{An-D.51})$$

subject to

$$A^T x \geq b , \quad (\text{An-D.52})$$

and

$$C^T x = d . \quad (\text{An-D.53})$$

This problem has been extensively treated by Lawson and Hanson [89]. In the context of Sequential Quadratic Programming (SQP) methods, square matrices  $E$  with dimension  $n$  and with special structure are considered. The matrices  $A^T$  and  $C^T$  are of dimension  $(m - m_e) \times n$  and  $m_e \times n$ , respectively. All matrices are assumed to be of full rank. This assumption does not constrain the generality of the method; it has been introduced to relax the derivation from the necessary refinements in the event of rank deficiencies.

The implementation (An-D.51) is used as QP solver within the code SLSQP of DLR Oberpfaffenhofen<sup>2</sup>. Actually, the code includes simple bounds on the variables  $l_i \leq x_i \leq u_i$ ,  $i = 1, \dots, n$ .

## D.2.2 Problem Transformation

According to Lawson and Hanson [93] a transformation of variables is introduced by an orthogonal basis  $K$  of the nullspace of  $C^T$ :

$$x = K \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{matrix} m_e \\ n - m_e \end{matrix}, \quad (\text{An-D.54})$$

where the  $n \times n$  orthogonal matrix  $K$  triangularizes the matrix  $C^T$  from the right:

$$\begin{matrix} m_e \\ n \\ m - m_e \end{matrix} \begin{pmatrix} n \\ m_e \\ n - m_e \end{pmatrix} \begin{pmatrix} C^T \\ E \\ A^T \end{pmatrix} K = \begin{pmatrix} \tilde{C}_1 & 0 \\ \tilde{E}_1 & \tilde{E}_2 \\ \tilde{A}_1 & \tilde{A}_2 \end{pmatrix}. \quad (\text{An-D.55})$$

Then  $\hat{x}_1$  is determined as the solution of the lower triangular system

$$\tilde{C}_1 x_1 = d, \quad (\text{An-D.56})$$

and  $\hat{x}_2$  is the solution of the following inequality-constrained least squares problem

$$\min_{x_2 \in R^{n-m_e}} \left\| \tilde{E}_2 x_2 + \tilde{E}_1 \hat{x}_1 - f \right\| \quad (\text{An-D.57})$$

subject to

$$\tilde{A}_2 x_2 \geq b - \tilde{A}_1 \hat{x}_1. \quad (\text{An-D.58})$$

Problem (An-D.57) is now transformed into a least distance problem (LDP). First the

---

<sup>2</sup>Which is the algorithm used for the tomography.

QR factors of the  $n \times (n - m_e)$  matrix  $\tilde{E}_2$ ,

$$\tilde{E}_2 = Q \begin{pmatrix} R \\ 0 \end{pmatrix} \quad (\text{An-D.59})$$

are determined, with  $Q$  an  $n \times n$  orthogonal matrix and  $R$  an upper triangular non-singular matrix. Then a further change of variables is introduced

$$y = Rx_2 - \bar{f}, \quad (\text{An-D.60})$$

with

$$\bar{f} = \bar{Q}^T(f - \tilde{E}_1 \hat{x}_1), \quad (\text{An-D.61})$$

where the  $n \times (n - m_e)$  matrix  $\bar{Q}^T$  is composed of the first  $n \times (n - m_e)$  columns of the matrix  $Q$ . Problem LDP can now be formulated as

$$\min_{y \in R^{n-m_e}} \|y\| \quad (\text{An-D.62})$$

subject to

$$\tilde{A}_2 R^{-1} y \geq b - \left( \tilde{A}_1 \hat{x}_1 + \tilde{A}_2 R^{-1} \bar{f} \right). \quad (\text{An-D.63})$$

The solution  $\hat{y}$  of problem LDP is introduced into equation (An-D.60) from which the solution  $\hat{x}_2$  of problem (An-D.57) is obtained, which in turn is put into equation (An-D.54) to give the solution  $x$  of the original problem (An-D.51).

Problem LDP has a non-negative least squares problem (NNLS)

$$\min_{z \in R^m} \|Gz - h\| \quad (\text{An-D.64})$$

subject to

$$z \geq 0 \quad (\text{An-D.65})$$

as its dual problem, where the  $(n + 1) \times m$  matrix  $G$  is composed of the constraint matrix and the right hand side of equation (An-D.63) and the  $(n + 1)$  vector  $h$  consists of  $n$  leading zeros followed by a trailing one,  $h = (0, \dots, 0, 1)^T$ . This transformation has been proposed by Cline [94].

From the solution  $\hat{z}$  of the dual problem NNLS and its residual  $r = G\hat{z} - h$  the solution  $\hat{y}$  of problem LDP is obtained by the relation

$$\hat{y}_i = r_i / r_{n+1}, \quad i = 1, \dots, n - m_e, \quad (\text{An-D.66})$$

and the corresponding Lagrange multipliers  $\lambda$  of problem LDP are

$$\lambda_j = -u_j/r_{n+1}, \quad j = 1, \dots, m, \quad (\text{An-D.67})$$

where the vector  $u$  is the multiplier vector of problem NNLS (An-D.64). Finally, problem NNLS is solved by an active set strategy.

# ANNEX E – FACTS ABOUT BREAST CANCER

This annex, based on [95], presents information for a better understanding of what is breast cancer, including its development and symptoms. It is intended for engineers and those without medical knowledge. One purpose is to clarify that breast cancer in its early stages may not have calcifications and therefore cannot be detected by mammographic exams using X-rays.

As the odds of successful treatment of breast cancer increase dramatically with early diagnosis, and early tumors lack X-rays detectable matter, ultrasound and ultrasound tomography emerges as a valuable mean to preserve health.

## E.1 What is cancer

Cancer is a disease characterized by uncontrolled growth of cells. There are more than 200 types of cancer, each one with a specific therapy.

Body cells are of many forms and have specialized functions, but all of them grow (by cellular division) to replace old and dead cells; this is a continuous and slow process. Sometimes this cellular growth becomes disordered and at times accelerated, giving origin to a lump called *tumor*, as depicted in figure 43.

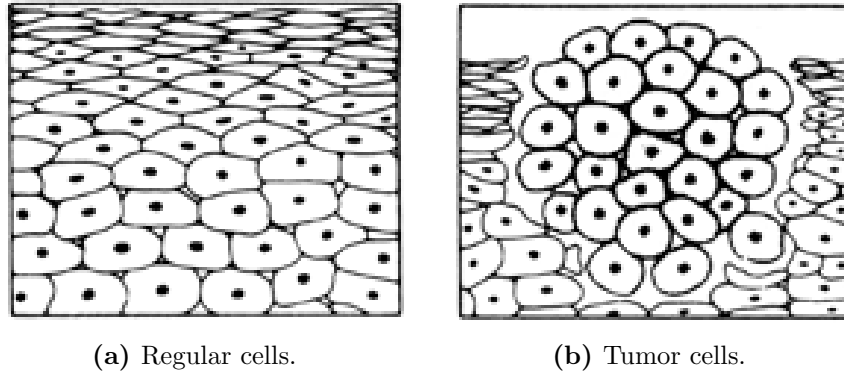
Benign tumors grow slowly and in a fixed place. They can cause troubles when big, because of pressure on adjacent organs.

Malign tumors grow not only fast, but also spread through adjacent tissues (and can destroy those tissues) and are also capable of starting to grow in other places of the body, at a distance, not connected to the primary tumor. The neoplastic cells detach from the original tumor and, via the lymphatic or the blood system, reach other organs where they could continue to grow, what is called a metastasis. Therefore – which is characteristic



of a metastasis – it's cells are not of the same type of cells from the (new) place where the metastasis is, but are cancerous, abnormal cells from the same kind of tissue of the original tumor. As an example, see figure 44.

**Figure 43** – Comparison of regular and tumor cells.



Source: [95].

**Figure 44** – Metastatic pancreatic cells (pale) in a liver.



Source: By Haymanj - Self-photographed, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=3662694>.

## E.2 The human breast

The human breast is composed mainly of skin, fatty tissue and glandular tissue, and those constituents proportions also change with age: in young women the glandular tissue prevails, in older women, instead, the fatty tissue. The breast is further subdivided, as depicted in figure 46, in lobules – which in turn are made up by smaller alveoli – milk ducts which goes from the lobules to the nipple, and veins, arteries, and connections with the lymphatic system by means of lymphatic vessels.

Mammary gland cells are of two types: lobular cells, which produce milk, and ductal cells, which compose the milk duct. If any of those cells starts to grow uncontrollably and invading and spreading to neighbouring cells, we will have a lobular or a ductal tumor or carcinoma, according to its origin. The change into a carcinoma is relatively slow and has some phases:

- in a first phase the carcinoma grows in the anatomic region of origin, and so is called *in situ*;
- in a second phase it becomes infiltrant in this region, and is then called *infiltrating* or *invasive*.

So, when someone is diagnosed with an invasive ductal or lobular carcinoma, it is still localized in a very specific region and not spreading to the whole body.

### E.2.1 Types of breast carcinoma

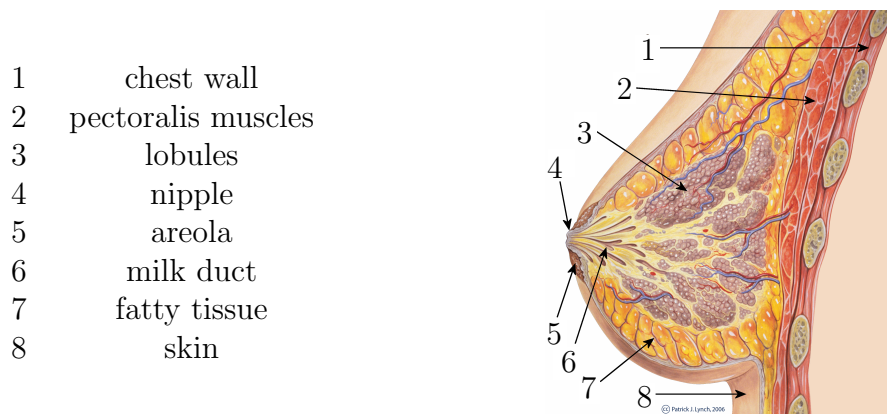
The breast carcinomas can be of four types:

1. lobular carcinoma in situ (LCIS): this is not a life risk, however is a pre-cancerous state if the carcinoma continues to develop. The treatment consists in the surgical removal of the tumor area. According to the evolution to a more malign state, it can be classified as LIN 1, LIN 2 or LIN 3, for lobular intraepithelial neoplasia. This carcinoma is not associated with calcifications.
2. ductal carcinoma in situ (DCIS): this is also considered a pre-cancerous tumor, however it is more dangerous than the lobular carcinoma in situ, because it tends to evolve and become infiltrant. Must be carefully treated, but this is not always possible, as in this stage it is not yet palpable. It frequently presents microcalcifications,

making it visible in mamographies (using X-rays). Teraphy is surgical remotion and radiotherapy.

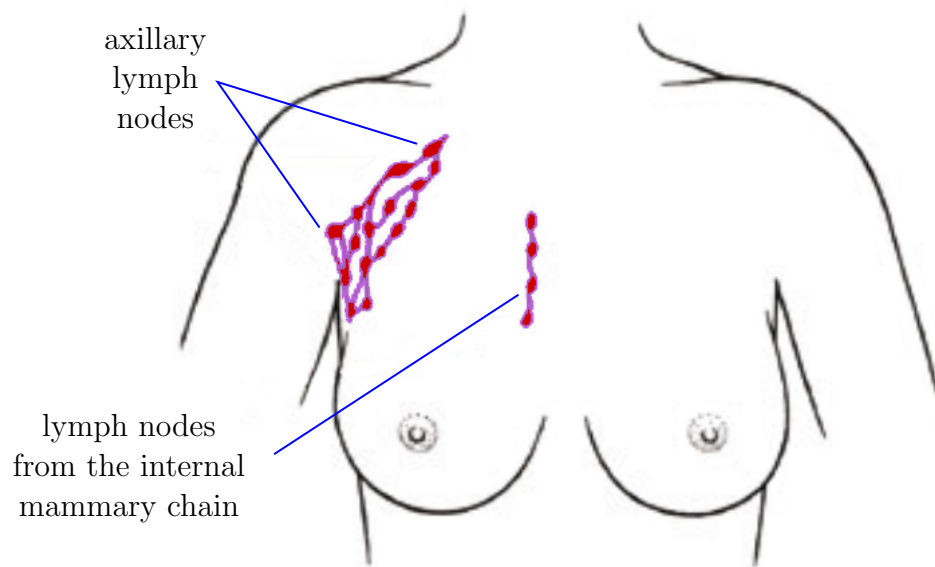
3. invasive (or infiltrant) lobular carcinoma (ILC): this is already a full-fledged carcinoma. Usually it is multi-centric (tumor foci in many quadrants of the breast) and multi-focal (many tumor foci in the same breast quadrant). This is the rarest ductal carcinoma. Usually demands magneptic resonance for a precise diagnosis. Therapy is surgical, including surgical evaluation of axillary lymph nodes.
4. invasive (infiltrant) ductal carcinoma (IDC): the most common breast carcinoma. Generally it is a single lump that grows in a specific point of the mammary gland. Therapy is surgical, including surgical evaluation of the axillary lymph nodes.

**Figure 45** – Anatomy of a human female breast.



Source: By Original author: Patrick J. Lynch. Reworked by Morgoth666 to add numbered legend arrows. - Patrick J. Lynch, medical illustrator, CC BY 3.0, <https://commons.wikimedia.org/w/index.php?curid=2676813>. Modified by the thesis author for typographical adherence to the required standard.

**Figure 46** – Lymphatic system in the breast region.



Source: modified from [95].