

Distributed Simulations for 3D Ultrasound Computer Tomography

zur Erlangung des akademischen Grades eines
DOKTORS DER NATURWISSENSCHAFTEN
von der Fakultät für Physik
des Karlsruher Instituts für Technologie

genehmigte

DISSERTATION

von

Dipl. Phys. Marcus Hardt
aus Jülich

Tag der mündlichen Prüfung: 22.06.2012

Referent: Prof. Dr. T. Müller, Institut für Experimentelle Kernphysik, KIT

Korreferent: Assistant Prof. Dr. K.W.A. van Dongen, Delft University of Technology

Abstract

More than 10% of all women in the western world get breast cancer. Mammography as a screening system is able to detect breast cancer at a diameter of 1cm with a probability of 68% (1σ). Higher sensitivity at the order of 5mm and the use of non ionising radiation is highly desirable to avoid the development of metastases. Ultrasound Computer Tomography (USCT) currently developed at KIT promises to be such a device.

The USCT project requires precise simulation data of realistic breast models for the optimisation and further development of its reconstruction algorithms. Up to now it was not possible to get such simulations within the allotted time frame of one or a few days of computing time.

For the solution of the ultrasound wave equation the integral equations were numerically solved. The main approximation which is made in this numerical solution is the assumption that density changes are negligible in human tissue. The errors introduced by this assumption were analysed and found to introduce no change in time of arrival. The error of the spatial energy distribution differs by up to 10%. Furthermore, the absorption is not included in the model.

An existing analytical solution for a plane wave scattered at a sphere was used to verify the results of the developed numerical solution. The resulting differences of the energy distribution remain below 3% and were 0.06% in average. In addition, the numerical solution was used to simulate the directivity characteristics of a surface source, yielding results of less than 6% difference when compared to a solution in Fraunhofer approximation.

The presented simulations were accelerated by distributed processing. The required development environment was implemented in this thesis, and enables Matlab users to distribute arbitrary tasks to a cluster or to cloud computing resources. The overhead of the implemented solution is measured to be 70ms per remote call using a local cloud. This enables efficient speed-ups even for short tasks, e.g. 192 tasks of 60s , the solution is 160 times faster than sequential processing.

Using the available infrastructure of 720 cores it was for the first time possible to conduct several full simulations for the full USCT system at a resolution of 128^3 voxels and $f_{max} = 140\text{kHz}$ for all 628 emitters in less than one day, corresponding to a speed-up of 650.

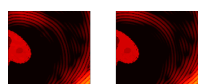
This infrastructure was used to create several test-cases using a realistic breast model. The test cases were used for challenging the reconstruction using the Synthetic Aperture Focussing Technique (SAFT) and to explore the imaging limits of the USCT system. Since SAFT is based on the Born approximation, multiple scattering is not included in the reconstruction. Our simulations show that only 1% of the scattered energy goes into multiple scattering.

The computational limits allowed only frequencies of up to 140kHz to be simulated. Scaled to the 3.5MHz of USCT the sidelength of one voxel ($dr = 2.1\text{mm}$) corresponds to $dr = 60\mu\text{m}$. This was used to simulate microcalcifications with $60\mu\text{m}$ and $120\mu\text{m}$, the larger of which can be clearly seen.

This thesis provides the first precise simulation system of the full USCT and creates artificial measurement data (890 000 A-scans) in less than one day. The simulations were used to evaluate the image quality achievable with SAFT and influence the future development for breast cancer imaging with USCT.

CONTENTS

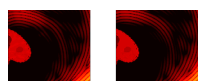
1	Introduction	9
2	Ultrasound Computer Tomography	13
2.1	Breast Cancer Detection	13
2.1.1	Standard imaging methods for breast cancer	14
2.2	Advantages and aims of USCT	16
2.2.1	Synthetic Aperture Focussing Technique	16
2.2.2	Limitations of SAFT	19
3	Theory	21
3.1	Helmholtz equation for inhomogeneous media	22
3.1.1	The fundamental equations of acoustics	22
3.1.2	Boundary conditions	22
3.1.3	The Helmholtz equation for inhomogeneous media	23
3.1.4	The Helmholtz equation for inhomogeneous media	23
3.2	Solution of the Helmholtz equation	24
3.2.1	The Sommerfeld radiation condition	24
3.2.2	The Green's function	24
3.2.3	The spatially averaged Green's function	25
3.2.4	The solution to the Helmholtz equation	25
3.2.5	Constant density approximation	25
3.2.6	Born-Approximation	26
3.3	Analytical solution for plane wave scattering	28
3.3.1	Solution of the Helmholtz Equation	28
3.3.2	Boundary conditions for $r \rightarrow 0$	31
3.3.3	Boundary conditions at the sphere ($r = R$)	32
4	Numerical Method	33
4.1	Reformulation of the problem	33
4.1.1	The Adjoint of L	34
4.2	Conjugate Gradient Method	35
4.2.1	The update directions	36
4.2.2	The step length	37
4.2.3	The resulting CG scheme	37
4.3	Implementation	38



4.3.1	Numerical precision	38
4.3.2	Memory considerations	38
4.3.3	Convolutions	39
4.3.4	Testing correctness	39
4.4	Discretisation and signal processing boundaries	40
4.4.1	Spatial discretisation	40
4.4.2	Temporal discretisation	41
4.4.3	Simulations in temporal Fourier domain	42
4.4.4	Incident Pulse	42
4.4.5	Parameters summarised	42
5	Computing Infrastructure	45
5.1	Requirements	45
5.2	Distributed Computing Infrastructures	47
5.2.1	Grid Computing	47
5.2.2	Cloud Computing	48
5.2.3	Cluster Computing	49
5.3	Testing the Integration of Matlab into the grid	49
5.3.1	First results	51
5.4	Final integration of DCI and Matlab	51
5.4.1	PARallel INfrastructure – PAIN	52
5.4.2	Concluding remarks	55
5.4.3	Additional Tools	56
5.4.4	PAIN usage within this thesis	57
5.5	Performance measurement	57
5.5.1	Cluster partitioning	58
5.5.2	Scalability of PAIN	60
5.5.3	Result	61
6	Results	63
6.1	Verification of the numerical solution	63
6.1.1	Result of the comparison	64
6.2	Impact of neglecting density changes	68
6.2.1	Impact on the travel time	69
6.2.2	Result	69
6.3	Impact of the Born approximation	70
6.3.1	Travel time analysis	74
6.3.2	Energy analysis	78
6.3.3	Impact on imaging	79
6.3.4	Result	86
6.4	The effect of surface sources	86
6.4.1	Energy distribution	87
6.4.2	Comparison with a far field simulation	87
6.4.3	Result	90
6.5	Visibility of microcalcifications	90
6.5.1	Result	92



7 Discussion and Conclusion	95
7.1 Discussion of Results and Methodology	96
7.1.1 Discussion of the approximations	96
7.1.2 Discussion of the scaled frequencies	97
7.1.3 Discussion of the computing infrastructure	98
7.1.4 Discussion of the limited resolution	99
7.2 Conclusions and Recommendations for Future Work	100
Bibliography	101
A Figures	109
A.1 Energy distribution for various Objects in the USCT	109
A.2 Additional A-scans	114
A.3 Plane wave scattering	120
A.4 Directivity patterns of the surface source	121
A.5 Technical Drawings	122
B Derivations and tools	123
B.1 The spatially averaged Green's function	123
B.2 The Fourier transform	124
B.3 Energy in a pressure wave	124
C Source Code	127
C.1 mpain – the Matlab interface to PAIN	127





INTRODUCTION

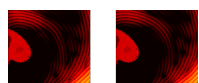
This thesis concerns the simulation of the 3D Ultrasound Computer Tomography (USCT) system, a novel device for medical breast cancer detection.

The early detection of breast cancer is very important because the earlier it can be detected the lower is the probability that the cancer has spread and created metastases. Once metastases are created the patient's survival rate drops significantly. Today the average size at which cancer is found using mammography is at 10 *mm* with a probability of 68 % (1σ) [MSM⁺03], however mammography can not frequently be applied because of its ionising radiation [GN11, OG01] and therefore misses fastly growing and therefore aggressive cancers [VSHS11]. Palpation, which can be made more frequently finds tumours at a size of 20 *mm* in average, a size at which the average probability for metastases is at 60 % [FAB93]. The goal of the USCT project is to develop a new medical device that does not make use of ionising radiation and which is capable to detect breast cancer tumors reliably at a size of 5 *mm*, where the average probability of spreading is low.

When building novel devices it is desirable to simulate the device before it is built, so that its design can be evaluated and optimised. Additionally, simulated data for which the ground truth is known provides valuable knowledge for the implementation of algorithms that reconstruct the measured data into images.

The goal of this thesis is the simulation of measurement data for ultrasound applications with a special focus on USCT. The scientific contribution is two fold. One part is the implementation of a scalable simulation for the scattering of pressure waves in human tissue which guarantees to deliver correct results within the chosen approximation. In addition, the second part is the design and implementation of a computational architecture capable to support the development and USCT simulations at medium scale of 64^3 to 256^3 voxels which correspond to frequencies in the range of 70 *kHz* to 280 *kHz*. The combination of both parts allows the evaluation of the imaging capabilities of USCT with respect to the approximations specific to reflection tomography.

Such simulations were earlier not possible. Finite elements methods (FEM) are frequently used to compute solutions to similar problems [LBPT05], i.e. solutions to the Helmholtz equation. However, FEM fail to obtain results of engineering accuracy unless 10 nodes per wavelength or more are used [LBPT05, IB95]. This is due to both,



memory problems which occur even when the simulated volume or the frequency range are reduced and also to computing problems, because the computing time required scales with the number of voxels. Finite differences are less precise than finite element methods but offer shorter computing times [Mar84]. Therefore, an attempt of using finite differences with wave3000 [Inc12] was conducted within the USCT project. For successful runs, the USCT had to be scaled down from a volume of $26 \times 26 \times 16 \text{ cm}^3$ to $2.6 \times 2.6 \times 1.6 \text{ cm}^3$ for simulations with pulses using frequencies of up to 3 MHz [WRS⁺06].

Several software products implement the spatial impulse response (SIR) method developed by Stephanishen [Ste73]. It computes the convolution of the incident field with the object function. Several simulation tools implement this method, such as Ultrasim [Hol01] which is a Matlab software or Field and FieldII [Jen96, JN00] with interfaces to Matlab. The latter two apply the far field approach when taking surface sources into account. The SIR methods, however, are not comparable to the simulations conducted within this thesis, since they do not provide exact results because they work within the assumptions of the Born approximation [AD09]. Many other software products exist which claim to provide “real-time” ultrasound imaging. It can be said in general that they can only provide results with limited accuracy due to approximations necessary to deliver real-time results.

Simulations of pressure waves in media can also be done by solving the Helmholtz equation of the problem using conjugate gradient methods. Much work in this field is published by the Acoustical Imaging faculty of TU Delft [FB93, BZ92, KB91a, KB91b, AD09]. These formulations form the basis upon which the scalable simulations within this work are developed.

Chapter 2 gives an introduction into the medical background of breast cancer detection. A short summary about currently practices in breast cancer detection methodologies leads to the motivations of the design of the USCT system. This device and the algorithm used for reconstruction are described in detail.

The simulation is created in collaboration with TU Delft and is based on solving the wave equations for pressure waves in liquids and is shown in chapter 3. The wave equations are expressed in temporal Fourier domain, because each stationary case can be solved independently. The corresponding Helmholtz equation [MF53] of the problem leads to a differential equation in integral form for the pressure field. To solve this equation we firstly neglect absorption. Secondly, we make the assumption that scattering on density changes can be neglected. But to keep the right speed of sound in the different media we adapt the compressibility accordingly. Thirdly, the Born approximation can optionally be introduced at this point. This approximation is the first step of the exact solution which is used within this thesis. The Born approximation is only used in specific cases, that are explicitly mentioned. In addition, an analytical solution for the case of a plane wave scattered on a sphere is derived for verification of the numerical results.

The remaining Fredholm integral of the second kind, is solved using a conjugate gradient method which is derived in chapter 4. This method was originally developed by Hestenes and Stiefel in 1952 [HS52]. The approach we use in this thesis is based on the formulations of Kleinman and van den Berg [KB91a, KB91b]. To obtain an efficient implementation all convolutions are computed as multiplications in Fourier domain, without which simulations of USCT would not be possible at all.



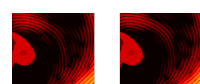
Still, a lot of computing power is required. This is why in chapter 5 access to powerful computing resources is investigated and implemented. We start with a short survey of the existing architectures normally used for accelerating and distributing computational problems. This includes traditional High Performance Computers (HPC), grid, clusters and cloud computing but also the feasibility of using graphics processors (GPUs) which became popular more recently. The final solution called PAIN (PARallel INfrastructure), which was implemented provides a simple but very comfortable access to cluster and cloud computing resources directly from Matlab. It enables straightforward parallelisation of existing Matlab programs, with minimal changes to existing source code. Additionally, no licenses need to be installed on the distributed infrastructure. Only the Matlab compiler is required on the developers computer. It is included in the campus license. The implementation of PAIN was targeted at the integration of clusters and cloud computing, but it was already extended within the SLA for D-Grid project [DGT11, THH12], to also use grid resources via the gLite middleware. The chapter concludes with benchmarks that characterise the scalability of PAIN.

In chapter 6 we use the distributed computing infrastructure and the implemented numerical algorithm. Firstly we verify our results with those of the numerical solution. Subsequently, the approximation in the numerical solution of constant density is quantified. This is done using the analytical solution for the two possible cases of reflecting the missing density changes either in the compressibility or in the speed of sound. Thereafter, the numerical solution is used to simulate several cases of different objects inside the USCT to evaluate its quality as an imaging system. Since the USCT reconstruction algorithm SAFT (Synthetic Aperture Focussing Technique) is based on the Born approximation artefacts have to be expected when reconstructing data simulated without making that approximation.

To analyse the effects of the Born approximation in SAFT, we create simulated data using the same approximation. When reconstructing this data with SAFT we commit the so called “inverse crime” [Wir04] which refers to the fact of obtaining better results than possible because the errors in simulation are compensated by the same errors in reconstruction. In our case this is intended so that we can use a reconstruction with ideal results for comparison with reconstructions of exact simulations. The USCT implementation of SAFT has one improvement over the Born approximation which is to take the spatial distribution of the speed of sound into account. This “sound speed map” option requires knowledge of the real speed of sound distribution of the object which is to be imaged. In case of simulations this is the known ground truth, in case of measurements it can be obtained by time of flight tomography [DGR12]. We compare the errors introduced by the Born approximation in SAFT to the corrections introduced by the sound speed map option.

The correct modelling of the emitter geometry is verified by comparing their directivity pattern with a tool developed specifically for the simulation of the ultrasound emitters of USCT. We furthermore investigate the feasibility of visualising microcalcifications, which pose the highest challenge in respect to imaging for USCT and appear in 30 % of the malign tumors in the breast.

Chapter 7 finishes with a discussion of the approximation made, their impact on simulations and imaging as well as a discussion of the discretisation limits within



which the simulations were made. Recommendations for future developments are discussed.



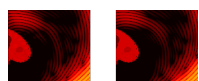
ULTRASOUND COMPUTER TOMOGRAPHY

This chapter gives an introduction into breast cancer, the conventional methods used today for breast cancer detection and introduces the device and the goals of the Ultrasound Computer Tomography (USCT) device. The reconstruction algorithm Synthetic Aperture Focussing Technique (SAFT), used to create images from measured data, is introduced. For in-depth information and wherever appropriate the reader is provided with links to external documentation.

2.1 Breast Cancer Detection

The mamma carcinoma, is the most likely cancer in women. In 2008 the world-wide yearly incident rate reached 1 384 155 (0.2 ‰ of total population), and 458 503 died [WHO08] [Cam08]. Figures for Germany from 2008 report 71 660 incidents (0.9 ‰ of total population) and 17 209 breast cancer caused deaths. This indicates a much higher percentages of affected women [RKI08]. This can be understood, since many other death causes have already been eliminated in the western world. Therefore, for women, breast cancer remains both, the most probable cancer to be obtained and the most likely cancer to die from [KB06].

To reach the goal of reducing deaths due to breast cancer it is essential to detect the primary tumour before it spreads and creates metastases, for example in the lymph system. The probability of a primary tumour to spread is correlated to its size. It was shown [FAB93] that breast cancer tumours with a diameter of 20 mm have an average spreading probability of 60 %, while a tumour with 5 mm diameter only spreads with an average likelihood of 5 %. This suggests that the tumour size has to be small when it is found.



2.1.1 Standard imaging methods for breast cancer

Palpation

For palpation a physician or the woman herself senses the breast for lumps included inside the breast. Palpation by a physician is the standard breast cancer screening method in Germany [RKI07].

Palpation often leads to first indications of a breast cancer (80 – 90 % of the cases). Using palpation cancer is found in average at sizes of 20 *mm* or more. At this size the average probability for spreading is already at 60 % [FAB93]. Only 10 % of these lumps are actual cancer tumours [MKAF07]

Mammography

In mammography the breast is deformed, by two plates so that a 2D X-ray projection image (such as shown in figure 2.1(A)) is taken of the deformed breast. The images are visually analysed by radiologists for signs of breast cancer.

In Mammography the detection rate for breast cancer tumours decreases significantly for dense breast tissue. Dense breasts contain a large amount of glandular tissue. The amount of glandular tissue decreases with age. Therefore, women younger than 50 are often not suitable for mammography. Unfortunately, the most aggressive breast cancers are found in dense breast tissue.

Mammography is widely used in breast cancer screening programs which have helped to reduce the mortality rate by 15 % [GN09]. The aim is detection of the cancer at an early stage, at which it is more likely that it can be cured by an appropriate treatment. However, the effectivity of mammography screening programmes has been questioned by recent reviews, because significance of mammography screening appears to be questionable [GN11, GO00, OG01, GD05, McP10]. This is also because mammography is not specific enough to distinguish between cancer and harmless cysts, which results in often unnecessary surgery.

Even though today the average size of the primary tumour detected by mammography screening is as small as 10 *mm* with a probability of 68 % (1σ) [MSM⁺03], the analysis of tumour doubling times [VSHS11] finds that “regular interval mammographies may be missing a high proportion of fast growing tumours”. This suggests that screenings should be conducted at higher frequencies. However, theories exist that exposure to ionizing radiation by screening includes a small but significant increase in breast cancer induced by the ionising radiation [Fri00].

Finally, mammography has the difficulty to gain sufficient knowledge about the position of the cancer, because the breast is deformed and imaging is only 2D. Extensions such as breast CT [BNLS01] and tomosynthesis [NCN⁺97] aim to overcome this problem but are not widely available and either expose the patient to higher radiation doses or provide less image contrast.

Medical sonography

Medical sonography is widely used in medical practice, covering use in diagnosis, therapy and to guide interventional surgery [Der02].

Imaging devices for sonography are hand scanners which contain arrays of ultrasound transducers. These are piezo ceramic elements which can emit and



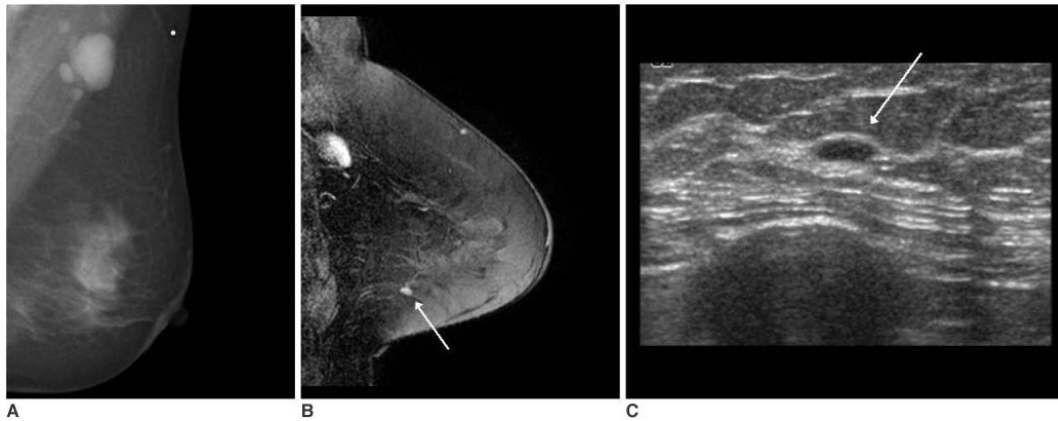


Figure 2.1: A 53-year-old woman with palpable left axillary lymph node metastasis. The mammogram (A) and US showed no abnormal findings in the breast. The contrast-enhanced MR image (B) showed 5 mm nodular enhancement without a washout pattern (arrow) in the left lower outer breast. On the MR-guided second-look US examination (C), a well-defined flat nodular lesion (arrow) was identified, and invasive ductal carcinoma was diagnosed by US-guided localization and excision. [KHSK07]

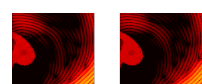
receive ultrasound wave pulses. The array emits directed wave pulses. The travel time and amplitude of the pulses is used to create the images such as shown in figure 2.1(C). Images are available in real time. Sonography is used to differentiate between various tissue types based on their reflectivity properties, which result in either sharp contrasts or in absorption which is detected by reduced speckle noise. Sonography has a high specificity for the detection of cancer and can distinguish well between cancer lesions and cysts.

In medical sonography the images contain speckle noise, shadowing artefacts and are of low spatial resolutions when compared to other methods. This is why sonography images are difficult to interpret and why the quality of a diagnosis depends largely on the skills of the operator. The low cost of devices makes them available to nearly every medical office, thereby leading to a high number of operators with varying experience.

Sonography can be applied to patients regardless of their age, therefore also younger women with dense breast tissue can be diagnosed.

Magnetic Resonance Imaging (MRI)

MRI is an imaging method based on the atomic nuclear spin. In a strong external magnetic field of $0.5 - 4 T$, the energetical niveaus of the nucleus with spin I splits into $2I + 1$ energy levels (Zeeman effect [Zee97]). State changes can be induced by an external electromagnetic field with a resonance frequency that corresponds to a specific nucleus and the external magnetic field (Larmor frequency [Lar97]). Using inhomogeneous external fields, the resonance frequency not only depends on the nucleus but also on its spatial position. This is used to create MRI images (see figure 2.1(B)) where the density distributions of atoms with different nuclear spin I and different magnetic field at the nucleus are depicted. Often the spatial concentration of hydrogen atoms is shown. Therefore, MRI provides a good contrast



between different soft tissues and is often applied to image brain, heart or breast. Contrast agents are used to improve the images.

In some countries MRI is used for screening of genetically predisposed patients or those with dense breast tissue. However, MRI is much more expensive than mammography and very much more expensive than sonography. It is therefore not used widely for screening but when an indication of cancer is given by mammography.

Breast MRI should not substitute mammography or ultrasound as some lobular cancers and ductal carcinoma in situ may not be seen with MRI [Ima12].

2.2 Advantages and aims of USCT

USCT aims to combine the advantages of conventional sonography with the advantages of a fixed setup. The 3D transducer aperture for USCT is shown in figure 2.2 (left). It consists of a half-ellipsoid on which 157 transducer array systems (TAS) are mounted. Each TAS holds four ultrasound emitters and nine receivers. The USCT transducers were designed to have a centre frequency of 2.5 MHz at 1.5 MHz bandwidth. All receivers record an amplitude scan (A-scan) for each emitter sending an ultrasound pulse into the device. The USCT ellipsoid can be rotated and lifted relative to the patient to obtain a large amount of measurements. With four different positions this results in total of more than 3.5 million A-scans.

This setup introduces the possibility to obtain 3D images by using emitter receiver combinations from many different directions. Additionally, the fixed setup surrounds the breast and thus enables to measure ultrasound waves which traverse the breast, not only the reflected ones as in classical sonography. This allows to determine additional tissue parameters such as speed of sound and attenuation. These additional modes of measurement can further improve the specificity for the detection of cancer. Furthermore, a fixed setup ensures constant image quality, independent of an operator. This allows to take several images over longer time periods and to compare them to each other.

This device does not use ionising radiation, is cheap in comparison to MRI and can record reproducible 3D images. These facts make USCT a strong candidate for breast cancer screening. Furthermore, the use of ultrasound allows the examination of women which cannot participate in mammography screening, like young or pregnant women or women with implants.

The goal of the USCT project is to detect cancer with an average size of 5 mm, to decrease the probability of metastases significantly. Therefore, a high image resolution and a good image quality are required. Good images in this sense are characterised by correctness and few artefacts.

2.2.1 Synthetic Aperture Focussing Technique

Several methods exist to reconstruct the 3.5 million measured A-scans (see figure 2.3) into a 3D image. The algorithm used in the USCT project is based on Synthetic Aperture Focussing Technique (SAFT). It uses the reflected signals for image creation. From the time at which a pulse is registered by a receiver the distance which the pulse has travelled is computed using an average speed of sound c . With the positions of emitter and receiver as focal points, the reflection has occurred on the surface of a 3D ellipsoid with the travel distance as sum of distances from the focal



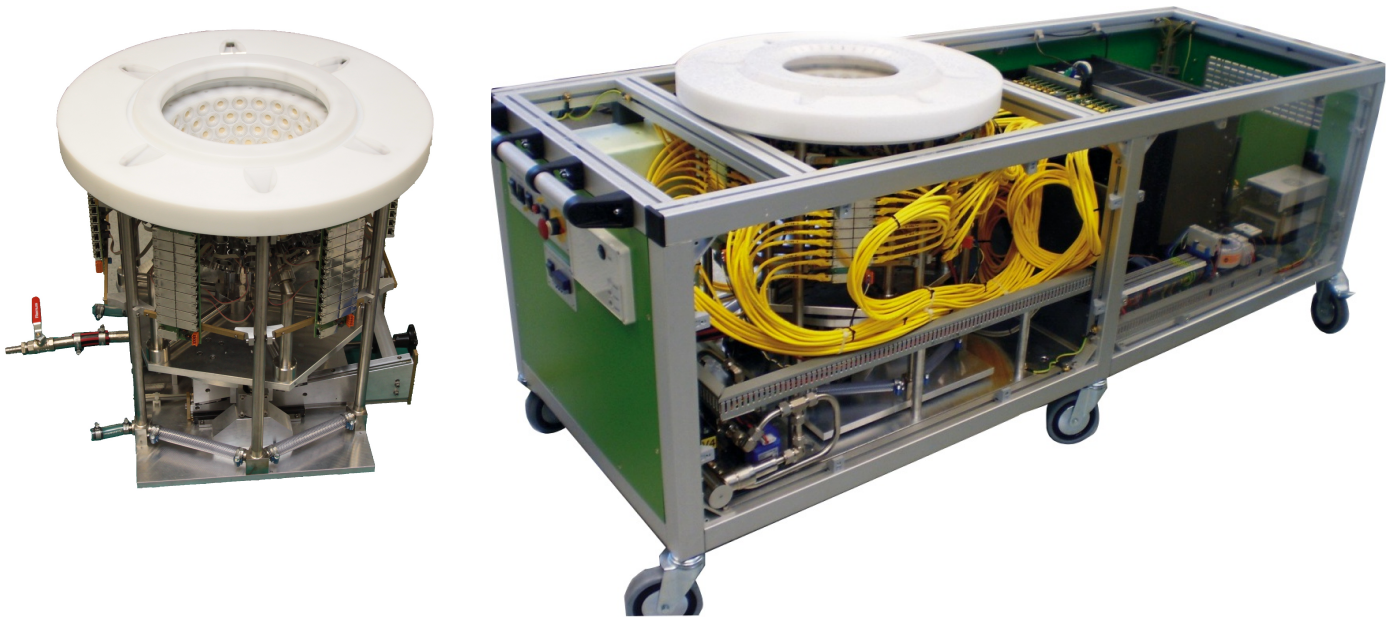


Figure 2.2: The USCT transducer aperture (left) and the overall device built (right) which holds all necessary components required for the measurement. The patient lies down on a couch on top of it (not mounted, therefore not shown) to place the breast into the USCT.

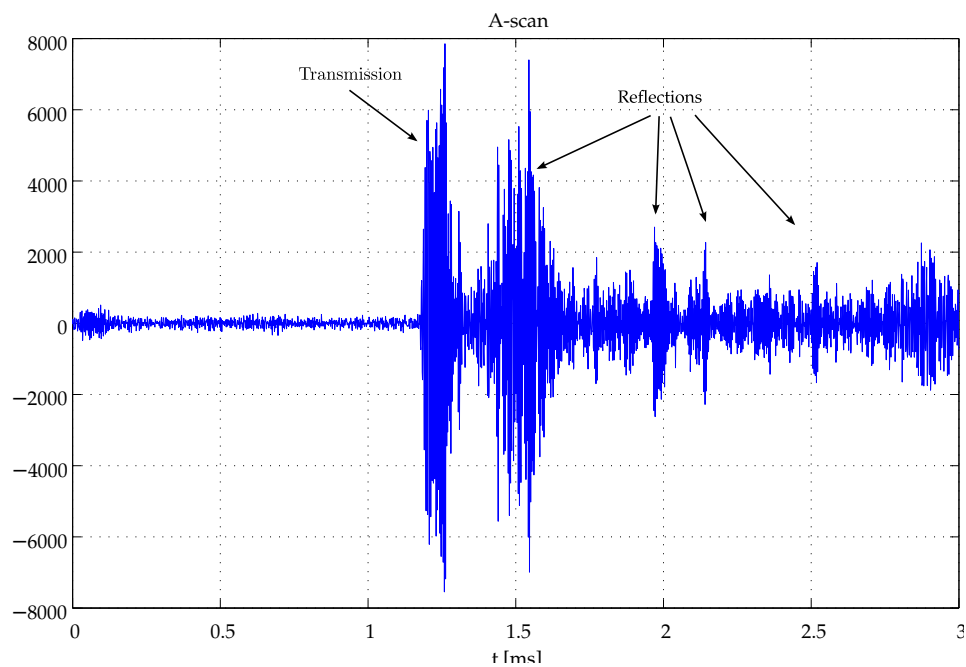
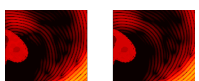


Figure 2.3: A-scan, measured in the USCT. The y -axis corresponds to the pressure amplitude (in relative units)

points to the ellipsoid. This ellipsoid is added to the resulting 3D image. Ellipsoids are created and added for all peaks of an A-scan and for selected A-scans from the



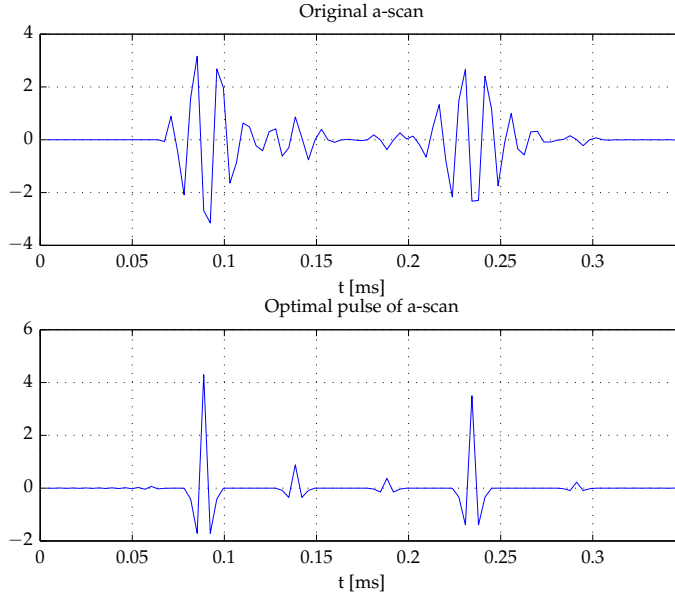


Figure 2.4: The optimal pulse preprocessing option replaces the original A-scan (top) with a sharper pulse (bottom) [RSZG08b,NL79]. The shown example was simulated with the same parameters used to create and reconstruct the datasets of this chapter.

3.5 million emitter-receiver combinations. The selection of A-scans is an option of SAFT which is based on geometrical considerations [RSZG08a]. In the resulting 3D image all sources of reflection will add up. This results in an image which shows the sources of reflection.

In practice the signals need to be processed for this algorithm to work. For this the USCT implementation of SAFT offers a lot of preprocessing options. Most notable of these are the matched filter and the optimal pulse filter [RSZG08b,NL79]. The matched filter is a convolution of the measured signal with the emitted signal. This translates the A-scan in time so that the maximum of the matched-filtered transmission pulse is at the time which is required to travel from emitter to receiver. The resulting signal is convoluted with a pulse with the optimal pulse p_{opt} for reflectivity imaging as introduced by Norton and Linzer [NL79]:

$$p_{opt} = 2\text{sinc}(2f_{bw}t) - \text{sinc}^2(f_{bw}t), \quad (2.1)$$

where b_{bw} is the bandwidth of the optimal pulse. The effect of the optimal pulse filter on a simulated pulse is shown in figure 2.4.

The geometrically motivated procedure of SAFT is described by the USCT equation:

$$f(\mathbf{x}) = \sum_{e,r} T \left[A_{e,r} \left(\frac{|\mathbf{x}_e - \mathbf{x}| + |\mathbf{x} - \mathbf{x}_r|}{c} \right) \right], \quad (2.2)$$

with the reflectivity map $f(\mathbf{x})$, preprocessing filters T , emitters and receivers e, r , their positions $\mathbf{x}_e, \mathbf{x}_r$, the A-scans $A_{e,r}(t)$, the speed of sound c and bold letters indicating vectors. Equation 2.2 shows that for every voxel all A-scans have to be processed for the time value to which this voxel corresponds.



2.2.2 Limitations of SAFT

According to Norton and Linzer [NL79] image creation using SAFT is done within four assumptions, which are repeated here in short for reference. The object is weakly reflecting, the medium is uniformly absorbing, the velocity of sound is constant and the object may be modelled as a collection of isotropic point scatterers. With exception of the absorption assumption, this corresponds to the Born approximation for imaging.

This introduces two types of visible artefacts. One is that multiple scattered signals will lead to ellipses which are added to wrong locations in the result image. The second one is the assumption of constant speed of sound, which leads to a wrong position of the ellipses in the resulting image. This error is larger than the one due to multiple scattering, because it affects the primary scattered signals which are much stronger (two orders of magnitude). For simulated signals this is shown in several A-scans of various objects in appendix A.2.

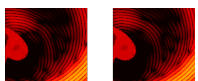
If the spatial distribution of the speed of sound is known, this information can be used in the reconstruction of images:

$$f(\mathbf{x}) = \sum_{e,r} T \left[A_{e,r} \left(\frac{|\mathbf{x}_e - \mathbf{x}| + |\mathbf{x} - \mathbf{x}_r|}{\int c(\mathbf{x}') dS(\mathbf{x}_e, \mathbf{x}_r, \mathbf{x}')} \right) \right], \quad (2.3)$$

where $S(\mathbf{x}_e, \mathbf{x}_r, \mathbf{x}')$ represents the path along which the wave pulse travels from emitter to receiver. This equation illustrates, that it is much more compute intensive to compute $f(\mathbf{x})$ with speed of sound correction than without. This is also because the ellipsoids are now deformed by the spatial variations in speed of sound.

In USCT the spatial speed of sound distribution $c(\mathbf{x})$ can be found using the time of arrival of signals which have travelled through the breast. An early implementation for such a transmission tomography algorithm exists in the USCT group [DGR12].

The effect of the missing speed of sound correction is shown for simulated data in figure 2.5. It can be seen that a small error in the speed of sound results in large degradation of the contrast and resolution of the resulting images.



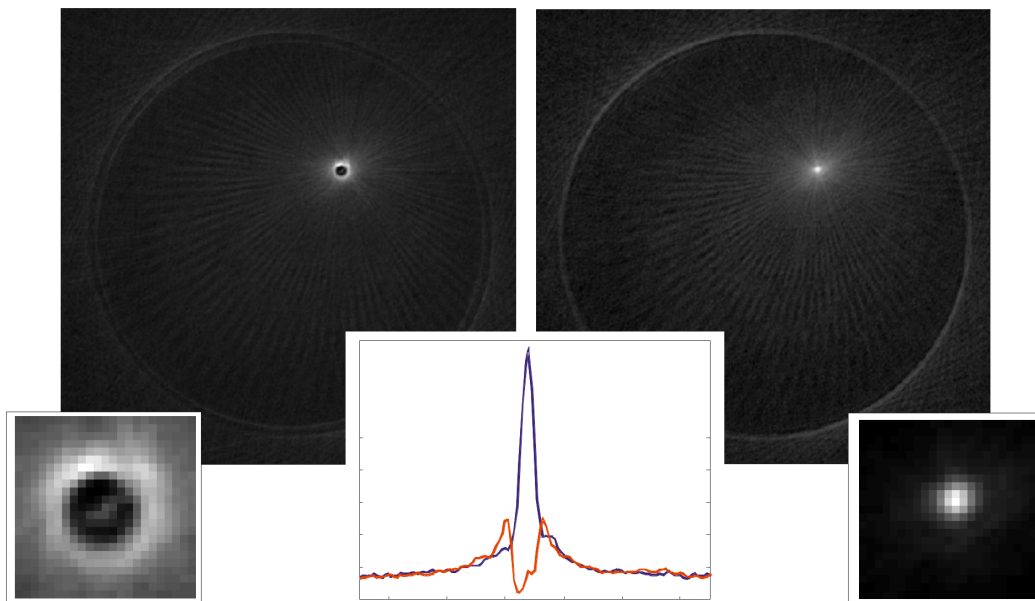


Figure 2.5: Reconstruction of reflection images of a gelatine phantom with an embedded nylon thread using (left) constant speed of sound and (right) the speed of sound correction. The small images show a nylon thread without (left) and with (right) correction. The centre diagram shows a profile through the nylon thread without (red) and with (blue) correction [RSZG07].



THEORY

In this chapter the required equations for scattering of acoustic waves in inhomogeneous media will be derived. The formulations used are based on the Book of J.T. Fokkema and P.M. van den Berg [FB93] of Delft.

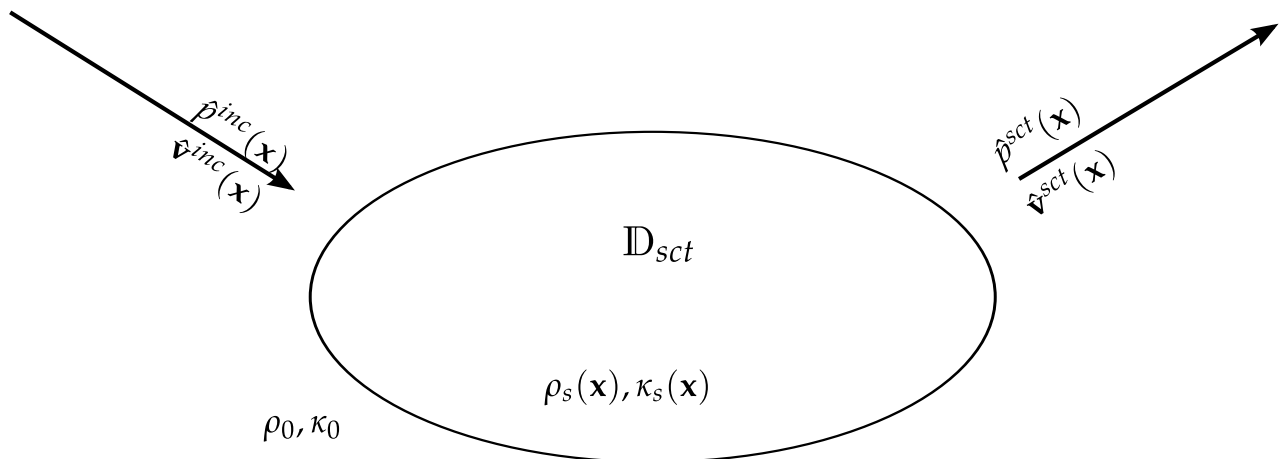
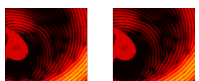


Figure 3.1: The region of scattering is embedded in a homogeneous background. The physical material parameters are denoted by volume density of mass $\rho(\mathbf{x})$ and compressibility $\kappa(\mathbf{x})$. The background parameters are subscripted with “0”, while for the inhomogeneities I use “s”. The inhomogeneous region is limited to \mathbb{D}_{sct} . The acoustic pressure wave fields are characterised by particle velocity $\mathbf{v}(\mathbf{x}, t)$ and pressure $p(\mathbf{x}, t)$ or their temporal Fourier transforms $\hat{\mathbf{v}}(\mathbf{x})$ and $\hat{p}(\mathbf{x})$. The incoming waves are superscripted by “inc”, while the scattered field is denoted by “sct”. The sum of both fields, the total field is denoted by “tot”.



3.1 Helmholtz equation for inhomogeneous media

3.1.1 The fundamental equations of acoustics

The fundamental equations for acoustic waves in liquids can be derived from Newton's [New86] and Hooke's [Hoo78] laws in continuous form, when assuming certain properties of fluids. The properties in terms of this thesis are those that can be observed in most fluids. I.e. fluids behave linear, time invariant, react instantaneously, react locally and are isotropic in their acoustic behaviour. This leads to the following fundamental acoustic equations [FB93]:

$$\begin{aligned}\nabla p(\mathbf{x}, t) + \rho(\mathbf{x})\partial_t \mathbf{v}(\mathbf{x}, t) &= \mathbf{f}(\mathbf{x}, t) \\ \nabla \mathbf{v}(\mathbf{x}, t) + \kappa(\mathbf{x})\partial_t p(\mathbf{x}, t) &= q(\mathbf{x}, t),\end{aligned}\quad (3.1)$$

with: $\rho(\mathbf{x})$ the volume density of mass ($[\frac{kg}{m^3}]$), $\kappa(\mathbf{x})$ the compressibility ($[Pa^{-1}]$), $\mathbf{f}(\mathbf{x})$ the volume source density of volume force ($[\frac{N}{m^3}]$), $q(\mathbf{x})$ the volume source density of injection rate ($[s^{-1}]$), ∂_t the partial derivative with respect to time, ∇ the spatial differential operator and bold font identifying vectors.

For further calculations, we consider these equations in Laplace domain, because there, the partial derivative with respect to time can be expressed by the Laplace Parameter s . In steady state s becomes $s = i\omega$, leading to

$$\nabla \hat{p}(\mathbf{x}) + i\omega\rho(\mathbf{x})\hat{\mathbf{v}}(\mathbf{x}) = \hat{\mathbf{f}}(\mathbf{x}) \quad (3.2)$$

$$\nabla \hat{\mathbf{v}}(\mathbf{x}) + i\omega\kappa(\mathbf{x})\hat{p}(\mathbf{x}) = \hat{q}(\mathbf{x}), \quad (3.3)$$

where $\hat{}$ indicates the temporal Fourier transform and the inherent dependence on the angular frequency $\omega = 2\pi f$.

3.1.2 Boundary conditions

Here, we consider only static liquids, i.e. situations in which no motion inside the liquids takes place.

If looking closely at the interface \mathcal{S} between two static liquids with different acoustic properties, we can derive boundary conditions for both the pressure and for the particle velocity across the interface.

Suppose two different pressures on both sides of \mathcal{S} . This would require a force density on the interface to compensate the differences in pressure. Since the interface is of zero width, this force density would furthermore be infinitely large. We therefore know, that the pressure is continuous across boundaries between static liquids:

$$\hat{p}(\mathbf{x}) \text{ is continuous across } \mathcal{S}. \quad (3.4)$$

Regarding the particle velocity we can also derive a boundary condition. Let's consider its component normal to \mathcal{S} , i.e. $\mathbf{v}_n \cdot \hat{\mathbf{v}}(\mathbf{x})$, where \mathbf{v}_n is the vector normal to the surface of \mathcal{S} . If the normal components of the velocities of the particles on both sides were different, the liquids would mix. This does not happen in the static case, hence

$$\mathbf{v}_n \cdot \hat{\mathbf{v}}(\mathbf{x}) \text{ is continuous across } \mathcal{S}. \quad (3.5)$$



3.1.3 Derivation of the Helmholtz equation for homogeneous media

First we derive the Helmholtz equation in the absence of scatterers, i.e. in a homogeneous medium, where $\rho(\mathbf{x}) = \rho_0$ independent of \mathbf{x} , which only contains the sources. Therefore, we separate $\hat{p}(\mathbf{x})$ and $\hat{\mathbf{v}}(\mathbf{x})$ in equations (3.2) and (3.3) by multiplying (3.2) with ∇ and (3.3) with $i\omega\rho_0$:

$$\begin{aligned} \nabla^2 \hat{p}(\mathbf{x}) + i\omega\rho_0 \nabla \hat{\mathbf{v}}(\mathbf{x}) &= \nabla \hat{\mathbf{f}}(\mathbf{x}) \\ i\omega\rho_0 \nabla \hat{\mathbf{v}}(\mathbf{x}) - \omega^2 \kappa_0 \rho_0 \hat{p}(\mathbf{x}) &= i\omega\rho_0 \hat{q}(\mathbf{x}), \end{aligned} \quad (3.6)$$

which can then be combined to

$$\left(\nabla^2 + \hat{k}_0^2 \right) \hat{p}(\mathbf{x}) =: -\hat{S}(\mathbf{x}), \quad (3.7)$$

with the source term $\hat{S}(\mathbf{x}) = i\omega\rho_0 \hat{q}(\mathbf{x}) - \nabla \hat{\mathbf{f}}(\mathbf{x})$, $c_0^2 = \frac{1}{\kappa_0 \rho_0}$ and the wave number \hat{k}_0 . Here, the wavenumber of the background medium $\hat{k}_0 = \frac{\omega}{c_0}$ is used. Equation (3.7) is the inhomogeneous Helmholtz Equation for $\hat{p}(\mathbf{x})$ in absence of scatterers, which are also called contrasts.

3.1.4 Derivation of the Helmholtz equation for inhomogeneous media

To obtain a Helmholtz equation in presence of scatterers, we proceed as before, but this time with spatial dependence of density $\rho_s(\mathbf{x})$ and compressibility $\kappa_s(\mathbf{x})$:

$$\begin{aligned} \nabla^2 \hat{p}(\mathbf{x}) + \nabla \{i\omega\rho_s(\mathbf{x}) \hat{\mathbf{v}}(\mathbf{x})\} &= \nabla \hat{\mathbf{f}}(\mathbf{x}) \\ i\omega\rho_s(\mathbf{x}) \nabla \hat{\mathbf{v}}(\mathbf{x}) - \omega^2 \kappa_s(\mathbf{x}) \rho_s(\mathbf{x}) \hat{p}(\mathbf{x}) &= i\omega\rho_s(\mathbf{x}) \hat{q}(\mathbf{x}), \end{aligned} \quad (3.8)$$

which can be combined to

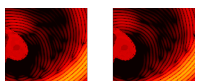
$$\begin{aligned} \nabla^2 \hat{p}(\mathbf{x}) + i\omega \nabla \{ \rho_s(\mathbf{x}) \} \hat{\mathbf{v}}(\mathbf{x}) + \omega^2 \rho_s(\mathbf{x}) \kappa_s(\mathbf{x}) \hat{p}(\mathbf{x}) + i\omega \rho_s(\mathbf{x}) \hat{q}(\mathbf{x}) &= \nabla \hat{\mathbf{f}}(\mathbf{x}) \\ \nabla^2 \hat{p}(\mathbf{x}) + \frac{\omega^2}{c_0^2} \hat{p}(\mathbf{x}) &= -i\omega \rho_s(\mathbf{x}) \hat{q}(\mathbf{x}) + \nabla \hat{\mathbf{f}}(\mathbf{x}) - i\omega \nabla \{ \rho_s(\mathbf{x}) \} \hat{\mathbf{v}}(\mathbf{x}) \\ &\quad - \frac{\omega^2}{c_0^2} \left(\frac{c_0^2}{c_s^2(\mathbf{x})} - 1 \right) \hat{p}(\mathbf{x}), \end{aligned} \quad (3.9)$$

using the relation between the speed of sound and the material parameters

$$c_s(\mathbf{x}) = \frac{1}{\kappa_s(\mathbf{x}) \rho_s(\mathbf{x})}. \quad (3.10)$$

We want to derive an expression for the pressure field when scattered at a contrast. If the sources of this pressure field are located outside the domain of scattering \mathbb{D}_{sct} , we can then use equation (3.2) with $\hat{\mathbf{f}}(\mathbf{x}) = 0$, to eliminate the particle velocity in equation (3.9). Furthermore using the identity $\nabla \{f\} \frac{1}{f} = \nabla \{\ln f\}$ we obtain:

$$\begin{aligned} \left(\nabla^2 + \hat{k}_0^2 \right) \hat{p}(\mathbf{x}) &= - \left[i\omega \rho_s(\mathbf{x}) \hat{q}(\mathbf{x}) - \nabla \hat{\mathbf{f}}(\mathbf{x}) \right] + \nabla \{ \ln [\rho_s(\mathbf{x})] \} \hat{p}(\mathbf{x}) \\ &\quad - k_0^2 \left(\frac{c_0^2}{c_s^2(\mathbf{x})} - 1 \right) \hat{p}(\mathbf{x}), \end{aligned} \quad (3.11)$$



Equation (3.11) is the desired Helmholtz equation with the source terms $\hat{S}(\mathbf{x}) = \hat{S}^{pr}(\mathbf{x}) + \hat{S}^{cs}[\hat{p}(\mathbf{x})]$, with the primary sources (pr) and the contrast sources (cs).

$$\begin{aligned}\hat{S}^{pr}(\mathbf{x}) &= i\omega\rho_s(\mathbf{x})\hat{q}(\mathbf{x}) - \nabla\hat{\mathbf{f}}(\mathbf{x}) \\ \hat{S}^{cs}[\hat{p}(\mathbf{x})] &= k_0^2\left(\frac{c_0^2}{c_s^2(\mathbf{x})} - 1\right)\hat{p}(\mathbf{x}) - \nabla\{\ln[\rho_s(\mathbf{x})]\}\nabla\hat{p}(\mathbf{x}).\end{aligned}\quad (3.12)$$

It should be noted that the gradient in second term of equation (3.12) can become large at locations where abrupt changes in the volume density of mass occur.

Characteristic for medical imaging are abrupt changes in material parameters, which can be either attributed to changes in the speed of sound $c(\mathbf{x})$ or to changes in the volume density of mass $\rho(\mathbf{x})$.

3.2 Solution of the Helmholtz equation

To solve the Helmholtz equation we can use the two equations (3.11) and (3.12). For this additional definitions are required, which are derived as follows.

3.2.1 The Sommerfeld radiation condition

If a field has the property that it vanishes for infinite distance to its source, i.e.

$$\lim_{|\mathbf{x}|\rightarrow\infty}|\mathbf{x}|\left(\frac{\partial}{\partial|\mathbf{x}|} - ik\right)p(\mathbf{x},t) = 0, \quad (3.13)$$

the Sommerfeld radiation condition is fulfilled. This is the case for acoustic waves, because the energy radiated through any surface around their sources is constant. According to Abramowitz [AS65] this is sufficient for the Helmholtz equation to have a solution, which is given by the convolution (*) of the source term with the Green's function:

$$\begin{aligned}\hat{p}(\mathbf{x},\omega) &= \hat{G}(\mathbf{x}) * \hat{S}(\mathbf{x}) \\ &= \int_{\mathbb{R}^3}\hat{G}(\mathbf{x}-\mathbf{x}')\hat{S}(\mathbf{x}')dV(\mathbf{x}'),\end{aligned}\quad (3.14)$$

where $\hat{G}(\mathbf{x})$ is the Green's function.

3.2.2 The Green's function

The Green's function is defined as the impulse response of a medium. Written in the context of the Helmholtz equation for a point source we have:

$$\left(\nabla^2 + \frac{\omega^2}{c_0^2}\right)\hat{G}(\mathbf{x}-\mathbf{x}',\omega) = -\delta(\mathbf{x}-\mathbf{x}'), \quad (3.15)$$

with $\delta(\mathbf{x})$ the Dirac delta. The Green's function for $\mathbf{x}-\mathbf{x}'\rightarrow\mathbf{x}$ is:

$$\hat{G}(\mathbf{x},\omega) = \frac{e^{-\frac{i\omega}{c}|\mathbf{x}|}}{4\pi|\mathbf{x}|}. \quad (3.16)$$



3.2.3 The spatially averaged Green's function

The Green's function as given in equation (3.16) has a singularity for $\mathbf{x} = 0$. This will lead to problems in simulations, whenever this position needs to be computed. We therefore compute the average of the Green's function inside a volume of a sphere with radius R , equivalent to the volume of a voxel [BZ92]. This value can be used instead of the singularity because the energy radiated out of the volume is the same:

$$\begin{aligned} [\hat{G}(\mathbf{x})] &= \frac{\int_{\|\mathbf{x}'\| \leq R} \hat{G}(\mathbf{x} + \mathbf{x}') dV(\mathbf{x}')}{\frac{4}{3}\pi R^3} \\ &= \frac{3}{4\pi R^3} \int_{\varphi'=0}^{2\pi} \int_{\vartheta'=0}^{\pi} \int_{r'=0}^R \frac{e^{-ik|\mathbf{x} + \mathbf{x}'|}}{4\pi|\mathbf{x} + \mathbf{x}'|} r'^2 \sin \vartheta' dr' d\vartheta' d\varphi'. \end{aligned} \quad (3.17)$$

Integration needs to be done for the two cases of $r = 0$ and $r \neq 0$. As shown in Appendix B.1 this yields:

$$[\hat{G}(\mathbf{x})] = \begin{cases} \frac{3 e^{-ikr}}{4k^3 \pi R^3 r} [\sin(kR) - kR \cos(kR)] & \forall r \neq 0 \\ \frac{3}{4k^2 \pi R^3} [(1 + ikR)e^{-ikR} - 1] & \forall r = 0. \end{cases} \quad (3.18)$$

This procedure is valid, because, according to the Sommerfeld theorem, the energy radiated through a surface of a sphere in which all sources are contained is constant – independent of size and shape of the surface.

3.2.4 Integral representation of the solution to the Helmholtz equation

Using equation (3.12) and (3.14), we can now express the pressure field using the primary and the contrast sources:

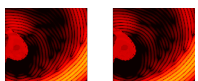
$$\begin{aligned} \hat{p}^{tot}(\mathbf{x}) &= \int_{\mathbb{D}_{sct}} \hat{G}(\mathbf{x} - \mathbf{x}') [\hat{S}^{pr}(\mathbf{x}') + \hat{S}^{cs}(\mathbf{x}')] dV(\mathbf{x}') \\ &= \hat{p}^{inc}(\mathbf{x}) + k_0^2 \int_{\mathbb{D}_{sct}} \hat{G}(\mathbf{x} - \mathbf{x}') \left[\frac{c_0^2}{c_s^2(\mathbf{x}')} - 1 \right] \hat{p}(\mathbf{x}') dV(\mathbf{x}') \\ &\quad - \int_{\mathbb{D}_{sct}} \hat{G}(\mathbf{x} - \mathbf{x}') \nabla \{ \ln[\rho_s(\mathbf{x}')] \} \nabla \hat{p}(\mathbf{x}') dV(\mathbf{x}') \end{aligned} \quad (3.19)$$

We have used the fact that the primary sources $\hat{S}^{pr}(\mathbf{x})$ convolved with the Green's function yield the incident field $\hat{p}^{inc}(\mathbf{x})$. With this we can define the scattered field as difference between the total field $\hat{p}^{tot}(\mathbf{x})$ and the incident field $\hat{p}^{inc}(\mathbf{x})$:

$$\hat{p}^{sct}(\mathbf{x}) = \hat{p}^{tot}(\mathbf{x}) - \hat{p}^{inc}(\mathbf{x}) \quad (3.20)$$

3.2.5 Constant density approximation

If we can assume constant density for our medium, we can neglect the last term in equation (3.19). In this case all changes in speed of sound $c(\mathbf{x}) = \frac{1}{\sqrt{\rho \kappa(\mathbf{x})}}$, i.e. all sources for scattering, would be attributed to changes in the compressibility $\kappa_s(\mathbf{x})$.



Tissue	$c[\frac{m}{s}]$	$\rho[\frac{kg}{m^3}]$	$\kappa[10^{-10}Pa^{-1}]$
Water	1524	993	4.34
Tissue	1550	1060	3.93
Fat	1470	950	4.87
Cancer	1580	1100	3.64
$\sigma[\%]$	3.0	6.4	12.7

Table 3.1: Material parameters of human tissue [Duc90, HS11], also indicating σ the relative standard deviation. (The compressibility was computed from given values for speed of sound and density.)

As can be seen in table 3.1, for various tissue types inside the breast, the changes in $\rho_s(\mathbf{x})$ are smaller than those in $\kappa_s(\mathbf{x})$.

However, for this approximation, we require in fact that the gradient of the logarithm of the density is small in comparison to the changes in speed of sound. This is not the case for all parts of the human body, especially for bones. In this thesis we formulate the equations for ultrasound scattering in the breast and therefore assume that the $\nabla \{\ln[\rho_s(\mathbf{x})]\}$ remains small in comparison to $k_0^2 \left(\frac{c_0^2}{c_s^2(\mathbf{x})} \right)$ in the female breast. Thus, we can rewrite equation (3.19) as

$$\hat{p}^{tot}(\mathbf{x}) = \hat{p}^{inc}(\mathbf{x}) + \hat{p}^{sct}(\mathbf{x})$$

$$\hat{p}^{tot}(\mathbf{x}) = \hat{p}^{inc}(\mathbf{x}) + k_0^2 \int \hat{G}(\mathbf{x} - \mathbf{x}') \left(\frac{c_0^2}{c_s^2(\mathbf{x}')} - 1 \right) \hat{p}^{tot}(\mathbf{x}') dV(\mathbf{x}') , \quad (3.21)$$

$$\text{with } \chi^c(\mathbf{x}) := \left(\frac{c_0^2}{c_s^2(\mathbf{x}')} - 1 \right) , \quad (3.22)$$

the contrasts based on changes in the speed of sound. Equation (3.21) will be the main equation used for computing pressure fields in this thesis.

3.2.6 Born-Approximation

There are cases, in which we do not want to solve the differential equation in integral form (3.21). This can be due to computational constraints or if we want the simulation to ignore certain aspects, or when it is required to express the scattered field directly.

In these cases, the total field $\hat{p}^{tot}(\mathbf{x})$ has to be replaced on the right hand side of equation (3.21) under the integral by the Born approximation, which consists of replacing the unknown total field under the integral of equation (3.21) with the known incident field $\hat{p}^{inc}(\mathbf{x})$. That means that we neglect multiple scattering:

$$\hat{p}_{Born}^{tot}(\mathbf{x}) = \hat{p}^{inc}(\mathbf{x}) + k_0^2 \int \hat{G}(\mathbf{x} - \mathbf{x}') \chi^c(\mathbf{x}') \hat{p}^{inc}(\mathbf{x}') dV(\mathbf{x}') . \quad (3.23)$$



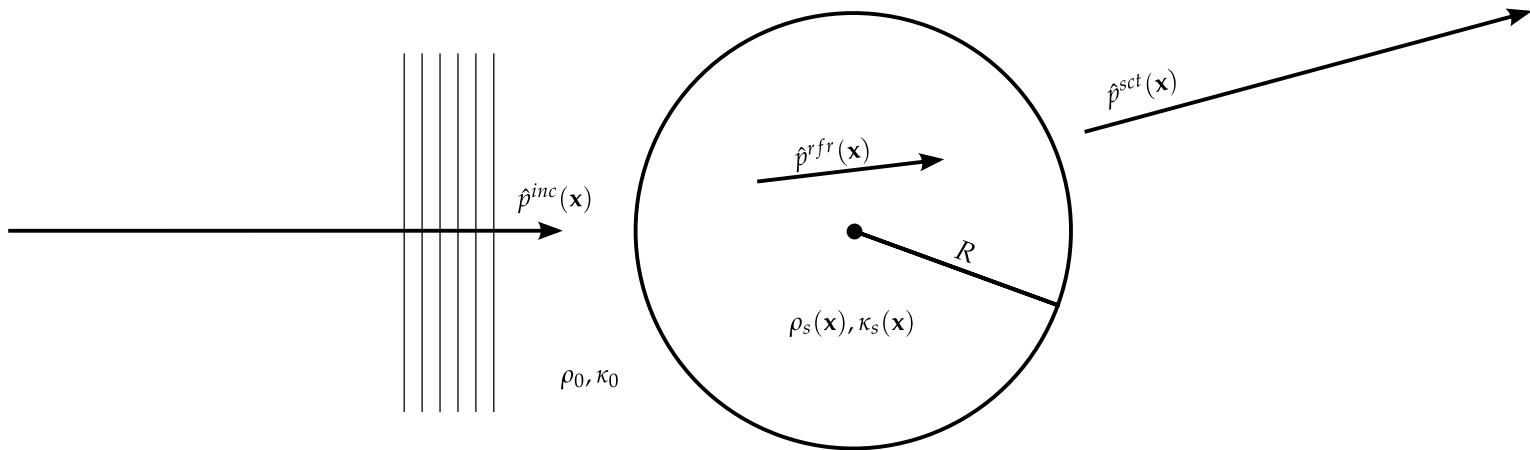


Figure 3.2: Geometrical setup for the analytical solution. The incident plane pressure wave $\hat{p}^{inc}(\mathbf{x})$ travels along the positive z-axis and is scattered on the sphere with radius R . The scattered or refracted field inside the sphere is denoted $\hat{p}^{rfr}(\mathbf{x})$, outside the sphere it is $\hat{p}^{sct}(\mathbf{x})$

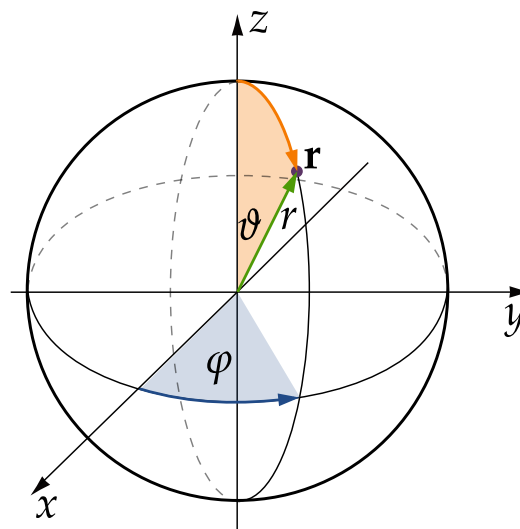
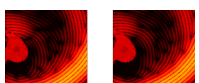


Figure 3.3: Illustration of the axes in the spherical coordinate system.



3.3 Analytical solution for plane wave scattering

To verify the numerical solution which will be applied to equation (3.21), an analytical solution is also computed. Only for a few geometries it is possible to derive an analytical solution. We choose a plane wave which is scattered on a sphere, which is located in the centre of a spherical coordinate system (see figure 3.2).

According to the geometry of the problem, the following formulations are done in a spherical coordinate system (see figure 3.3):

$$\begin{aligned} x &= r \sin \vartheta \cos \varphi \\ y &= r \sin \vartheta \sin \varphi \\ z &= r \cos \vartheta . \end{aligned} \quad (3.24)$$

The sphere with radius R is located at the origin. The incident pressure field $\hat{p}^{inc}(\mathbf{x})$ is a plane wave, travelling along the positive z-axis:

$$\begin{aligned} \hat{p}^{inc}(\mathbf{x}) &= \hat{p}_0 \exp(-i\hat{k}_0[z - d]) \\ &= \hat{p}_0 \exp(-i\hat{k}_0[r \cos \vartheta - d]) \text{ (in spherical coordinates) .} \end{aligned} \quad (3.25)$$

3.3.1 Solution of the Helmholtz Equation

The generic homogeneous Helmholtz equation serves as our starting point:

$$\left(\nabla^2 + \hat{k}^2 \right) \hat{p}^{tot}(\mathbf{x}) = 0 , \quad (3.26)$$

Because of the symmetry of the problem the pressure can be separated into a radial and an angular part:

$$\begin{aligned} \hat{p}(\mathbf{x}) &= \hat{p}(r, \vartheta, \varphi) \\ &=: \Gamma(r)\Psi(\vartheta, \varphi) . \end{aligned} \quad (3.27)$$

The Helmholtz equation can be rewritten, using the definition of the Laplace operator in spherical coordinates:

$$\begin{aligned} 0 &= \frac{1}{r^2} \partial_r [r^2 \partial_r \Gamma(r)\Psi(\vartheta, \varphi)] + \frac{1}{r^2 \sin \vartheta} \partial_\vartheta \{ \sin \vartheta \partial_\vartheta [\Gamma(r)\Psi(\vartheta, \varphi)] \} \\ &\quad + \frac{1}{r^2 \sin^2 \vartheta} \partial_\varphi^2 [\Gamma(r)\Psi(\vartheta, \varphi)] + \hat{k}^2 \Gamma(r)\Psi(\vartheta, \varphi) \\ &= \frac{1}{\Gamma(r)} \partial_r [r^2 \partial_r \Gamma(r)] + \frac{1}{\Psi(\vartheta, \varphi) \sin \vartheta} \partial_\vartheta [\sin \vartheta \partial_\vartheta \Psi(\vartheta, \varphi)] \\ &\quad + \frac{1}{\Psi(\vartheta, \varphi) \sin^2 \vartheta} \partial_\varphi^2 \Psi(\vartheta, \varphi) + r^2 \hat{k}^2 , \end{aligned} \quad (3.28)$$

with ∂_a the derivative with respect to variable a . Equation (3.28) can be split into two separate differential equations, one only depending on r and one only depending on ϑ and φ . Both equations are determined up to an arbitrary constant which is chosen to be $n(n+1)$.



The radial part of the Helmholtz equation

For the r dependent equation we obtain:

$$\begin{aligned} \frac{1}{\Gamma(r)} \partial_r [r^2 \partial_r \Gamma(r)] + r^2 \hat{k}^2 &= n(n+1) \text{ with: } n \in \mathbb{N} \\ \Leftrightarrow 2r \partial_r \Gamma(r) + r^2 \partial_r^2 \Gamma(r) + [r^2 \hat{k}^2 - n(n+1)] \Gamma(r) &= 0. \end{aligned} \quad (3.29)$$

This differential equation looks very similar to Bessel's differential equation for fractional numbers n . The definition below is from [AS65] (Abramowitz of 1966, p.437):

$$2r \partial_r \Gamma(r) + r^2 \partial_r^2 \Gamma(r) + [r^2 - n(n+1)] \Gamma(r) = 0 \quad (3.30)$$

The similarity between equations (3.29) and (3.30) means we can solve the radial part of the Helmholtz equation with Bessel functions. The full solution to equation (3.30) are the spherical Bessel functions of the first and the second kind, also known as Bessel and Neumann functions. They form an orthogonal basis on the surface of the sphere for each index n :

$$\begin{aligned} j_n(kr) &= \sqrt{\frac{\pi}{2kr}} J_{n+\frac{1}{2}}(kr) \\ y_n(kr) &= \sqrt{\frac{\pi}{2kr}} Y_{n+\frac{1}{2}}(kr). \end{aligned} \quad (3.31)$$

J_n and Y_n are the cylindrical Bessel functions of the first and second kind while j_n and y_n are their spherical counterparts.

A different basis are the spherical Hankel functions of the first and the second kind:

$$\begin{aligned} h_n^{(1)} &= j_n + iy_n \\ h_n^{(2)} &= j_n - iy_n. \end{aligned} \quad (3.32)$$

The useful property of the Hankel functions the existence of a comparatively simple formula to compute them:

$$h_n^{(1)}(x) = (-i)^{n+1} \frac{e^{ix}}{x} \sum_{m=0}^n \frac{i^m}{m!(2x)^m} \frac{(n+m)!}{(n-m)!}. \quad (3.33)$$

Summing up, we can solve the radial part of the Helmholtz equation with:

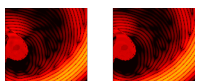
$$\Gamma(r) = \sum_{n=0}^{\infty} \alpha_n j_n(kr) + \beta_n y_n(kr), \quad (3.34)$$

with α and β complex constants.

The angular parts of the Helmholtz equation

Starting with the remaining part of equation (3.28) and the Ansatz

$$\Psi(\vartheta, \varphi) = \Theta(\vartheta) \Phi(\varphi), \quad (3.35)$$



we obtain:

$$\begin{aligned} \frac{1}{\Theta(\vartheta)} \frac{1}{\sin \vartheta} \partial_{\vartheta} [\sin \vartheta \partial_{\vartheta} \Theta(\vartheta)] + \frac{1}{\Phi(\varphi)} \frac{1}{\sin^2 \vartheta} \partial_{\varphi}^2 \Phi(\varphi) &= -n(n+1) \\ \Leftrightarrow \frac{\sin \vartheta}{\Theta(\vartheta)} \partial_{\vartheta} [\sin \vartheta \partial_{\vartheta} \Theta(\vartheta)] + \frac{1}{\Phi(\varphi)} \partial_{\varphi}^2 \Phi(\varphi) &= -n(n+1) \sin^2 \vartheta . \end{aligned} \quad (3.36)$$

This equation is separable in the variables ϑ and φ . Again, both equations are determined by an arbitrary constant which we choose to be m :

$$\frac{1}{\Phi(\varphi)} \partial_{\varphi}^2 \Phi(\varphi) = -m^2 , \quad (3.37)$$

which is solved by

$$\Phi(\varphi) = e^{\pm im\varphi} \quad (3.38)$$

With the plane wave travelling along the z-axis the problem is symmetric against rotation around φ (see figure 3.2). Therefore, we choose $m = 0$ and $\Phi(\varphi) = 1$ to simplify the further calculation. The fact that the problem is independent of φ requires $\Phi(\varphi) = 1$ from which follows that m equals zero.

The ϑ dependent solution

The solution for $\Phi(\varphi)$ leaves from equation (3.36):

$$\begin{aligned} \frac{\sin \vartheta}{\Theta(\vartheta)} \partial_{\vartheta} [\sin \vartheta \partial_{\vartheta} \Theta(\vartheta)] + n(n+1) \sin^2 \vartheta &= m^2 \\ \frac{1}{\sin \vartheta} \partial_{\vartheta} [\sin \vartheta \partial_{\vartheta} \Theta(\vartheta)] + n(n+1) \Theta(\vartheta) &= 0 \\ \frac{1}{\sin \vartheta} (\cos \vartheta \partial_{\vartheta} \Theta(\vartheta) + \sin \vartheta \partial_{\vartheta}^2 \Theta(\vartheta)) + n(n+1) \Theta(\vartheta) &= 0 \\ \frac{\cos \vartheta}{\sin \vartheta} \partial_{\vartheta} \Theta(\vartheta) + \partial_{\vartheta}^2 \Theta(\vartheta) + n(n+1) \Theta(\vartheta) &= 0 . \end{aligned} \quad (3.39)$$

The final expression in equation (3.39) looks similar to the associated Legendre polynomials [AS65](Abramowitz, p.332, equation (8.1.1)). Using this equation (8.1.1) for $\mu = 0$: and identifying w with f and z with $\cos \vartheta$, we obtain:

$$\partial_{\vartheta}^2 \Theta(\vartheta) + \frac{\cos \vartheta}{\sin \vartheta} \partial_{\vartheta} \Theta(\vartheta) + \left[\nu(\nu+1) - \frac{\mu^2}{\sin^2 \vartheta} \right] \Theta(\vartheta) = 0 . \quad (3.40)$$

Thus our solution for the ϑ dependent part of the Helmholtz equation is:

$$\Theta(\vartheta) = P_n^{\mu=0}(\cos \vartheta) , \quad (3.41)$$

where $P_n(\cos \vartheta)$ are the Legendre Polynomials.



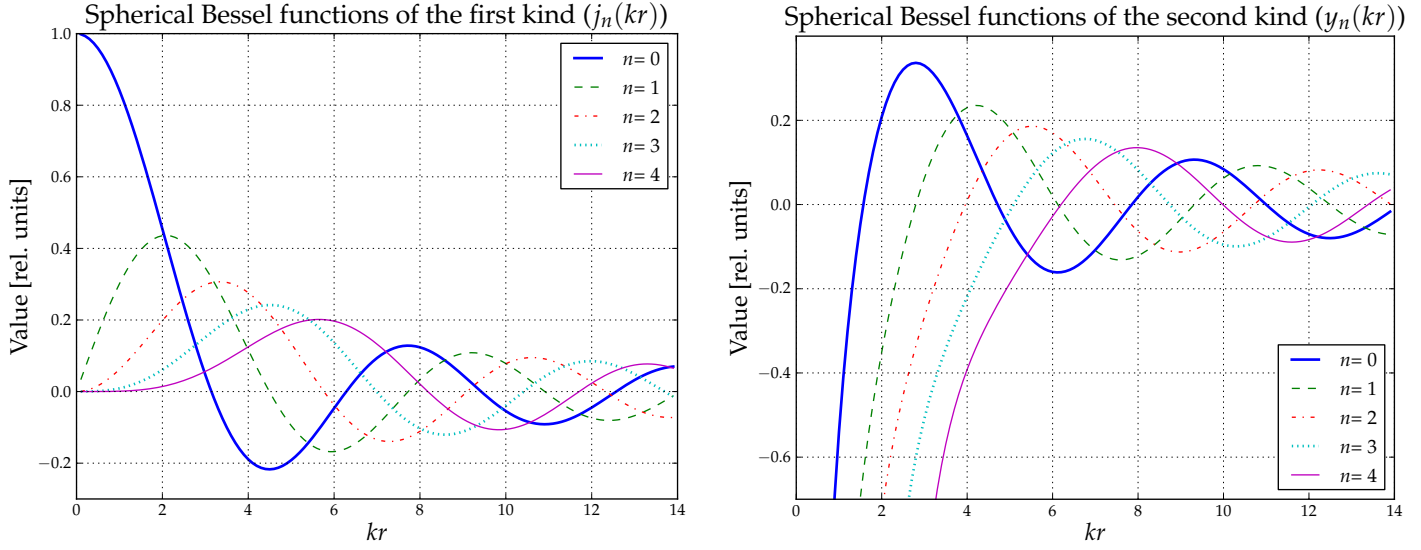


Figure 3.4: Spherical Bessel functions of the first and the second kind for $n = 0, \dots, 4$

Joining the partial solutions

According to the definitions of equations (3.27) and (3.35), we can write the full solution as:

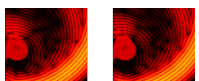
$$\begin{aligned} \hat{p}^{tot}(r, \vartheta, \varphi) &= \sum_{n=0}^{\infty} \hat{p}_n(r, \vartheta, \varphi) \\ &= \sum_{n=0}^{\infty} [j_n(kr) + y_n(kr)] P_n(\cos \vartheta) \end{aligned} \quad (3.42)$$

3.3.2 Boundary conditions for $r \rightarrow 0$

The boundary conditions for $r \rightarrow 0$ and $r \rightarrow R$ define the coefficients of the Bessel and Hankel functions. The total pressure field can be decomposed into the field inside and outside the sphere:

$$\hat{p}^{tot}(\mathbf{x}) = \begin{cases} \hat{p}^{inc}(\mathbf{x}) + \hat{p}^{sct}(\mathbf{x}) & |\mathbf{x}| > R \\ \hat{p}^{rfr}(\mathbf{x}) & |\mathbf{x}| < R. \end{cases} \quad (3.43)$$

Any pressure field can be expressed as a linear combination of the solutions p_n to the Helmholtz equation (3.42), because they form the linear independent basis of solutions. However, some solutions cannot be applied, because they do not match the physical boundary conditions. As can be seen in figure 3.4, the Bessel functions of the second kind diverge for $kr \rightarrow 0$. We can therefore deduce that those parts of the pressure fields which exist at $kr = 0$ can only be expressed by linear combinations of Bessel functions of the first kind. In equation (3.44) this is the case for $\hat{p}^{inc}(\mathbf{x})$ and $\hat{p}^{rfr}(\mathbf{x})$ with known complex constants a_n and unknown c_n . The scattered field outside the sphere $\hat{p}^{sct}(\mathbf{x})$ can be expressed by both, the Bessel



functions of the first and the second kind. Therefore the Hankel functions of the second kind (see equation 3.32) are chosen, with complex coefficients b_n :

$$\begin{aligned}\hat{p}^{inc}(\mathbf{x}) &= A e^{-i\hat{k}_0 r \cos \vartheta} \\ &= \sum_{n=0}^{\infty} a_n j_n(\hat{k}_0 r) P_n(\cos \vartheta) \\ \hat{p}^{sct}(\mathbf{x}) &= \sum_{n=0}^{\infty} b_n h_n^{(2)}(\hat{k}_0 r) P_n(\cos \vartheta) \\ \hat{p}^{rfr}(\mathbf{x}) &= \sum_{n=0}^{\infty} c_n j_n(\hat{k}_s r) P_n(\cos \vartheta).\end{aligned}\tag{3.44}$$

3.3.3 Boundary conditions at the sphere ($r = R$)

Using the boundary conditions as derived in (3.4) and (3.5) for our sphere of radius R it is possible to obtain explicit expressions for the pressure fields.

$$\begin{aligned}p(R_<) &= p(R_>) \\ \mathbf{v}_n \cdot \mathbf{v}(R_<) &= \mathbf{v}_n \cdot \mathbf{v}(R_>),\end{aligned}\tag{3.45}$$

with

$$\begin{aligned}R_> &= \lim_{\delta \downarrow 0} R + \delta \\ R_< &= \lim_{\delta \uparrow 0} R + \delta.\end{aligned}\tag{3.46}$$

This definition of the total field (3.43) can be used within the boundary conditions. The boundary conditions have to be fulfilled for every individual order of n , because the Bessel functions j_n , the Hankel functions of the second kind $h_n^{(2)}$ and the Legendre Polynomials $P_n(\cos \vartheta)$ each are orthogonal for each n . I.e. $\langle j_n | j_m \rangle \sim \delta_{n,m}$. This leads to a set of $2n$ equations:

$$\begin{aligned}a_n j_n(\hat{k}_0 R) + b_n h_n^{(2)}(\hat{k}_0 R) &= c_n j_n(\hat{k}_s R) \\ a_n \frac{1}{\rho_0} \partial_r j_n(\hat{k}_0 R) + b_n \frac{1}{\rho_0} \partial_r h_n^{(2)}(\hat{k}_0 R) &= c_n \frac{1}{\rho_s} \partial_r j_n(\hat{k}_s R),\end{aligned}\tag{3.47}$$

Which can be solved for c_n and b_n for each n . The a_n are known coefficients given by the properties of the incident plane wave.

$$\begin{aligned}b_n &= a_n \frac{\frac{1}{\rho_0} j_n(\hat{k}_s R) \partial_r j_n(\hat{k}_0 R) - \frac{1}{\rho_s} j_n(\hat{k}_0 R) \partial_r j_n(\hat{k}_s R)}{\frac{1}{\rho_s} h_n^{(2)}(\hat{k}_0 R) \partial_r j_n(\hat{k}_s R) - \frac{1}{\rho_0} j_n(\hat{k}_s R) \partial_r h_n^{(2)}(\hat{k}_0 R)} \\ c_n &= a_n \frac{\frac{1}{\rho_0} h_n^{(2)}(\hat{k}_0 R) \partial_r j_n(\hat{k}_0 R) - \frac{1}{\rho_0} j_n(\hat{k}_0 R) \partial_r h_n^{(2)}(\hat{k}_0 R)}{\frac{1}{\rho_s} h_n^{(2)}(\hat{k}_0 R) \partial_r j_n(\hat{k}_s R) - \frac{1}{\rho_0} j_n(\hat{k}_s R) \partial_r h_n^{(2)}(\hat{k}_0 R)},\end{aligned}\tag{3.48}$$

where the partial derivative of a function at radius R is to be understood as the derivative of a function in evaluated at $r = R$ in the sense of $\partial_r j_n(kR) = \partial_r j_n(kr)|_{r=R}$.

Equation (3.48) can be used in equation (3.43) to compute the solution to the analytical problem. The implementation of this analytical case is presented in detail, including Matlab sourcecode in [Don10]. Results of this implementation are included in this thesis to verify the numerical results.



NUMERICAL METHOD

The goal is to solve the integral equation (3.21) using established methods. This chapter will show how this integral equation can be formulated in a way so that it can be solved like any equation system. The numerical methods used for solving equation systems will be introduced and applied to the problem.

4.1 Reformulation of the problem

For reference equation (3.21) is shown again:

$$\hat{p}^{tot}(\mathbf{x}) = \hat{p}^{inc}(\mathbf{x}) + k_0^2 \int \hat{G}(\mathbf{x} - \mathbf{x}') \chi^c(\mathbf{x}') \hat{p}^{tot}(\mathbf{x}') dV(\mathbf{x}') . \quad (3.21)$$

Equation (3.21) is a Fredholm integral equation of the second kind [Bak83], which are often solved using Neumann iterations [AWH05]. However, Neumann iterations are not always converging [KB91b]. The approach taken in this work is therefore to solve the integral equation (3.21) like an equation system. For this equation (3.21) needs to be brought into the form of an equation system. Well known methods for solving equation systems exist. They solve systems of the form

$$\mathbf{L}\mathbf{u} = \mathbf{f} , \quad (4.1)$$

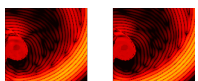
where \mathbf{f} is the known solution, the integral operator \mathbf{L} is the kernel of the integral equation and \mathbf{u} is the unknown. If equation (3.21) can be rewritten in terms of equation (4.1) existing methods can be used to solve it efficiently.

In discretised space with N voxels ($N = N_x \cdot N_y \cdot N_z$) we can rewrite it as a sum over all equally sized voxels:

$$\hat{p}^{tot}(\mathbf{x}_g) - k_0^2 \sum_{j=1}^N \hat{G}(\mathbf{x}_g - \mathbf{x}_j) \chi^c(\mathbf{x}_j) \hat{p}^{tot}(\mathbf{x}_j) \Delta V = \hat{p}^{inc}(\mathbf{x}_g) , \quad (4.2)$$

where \mathbf{x}_g and \mathbf{x}_j correspond to a vector pointing to the g -th and j -th voxel. Accordingly, a shorter notation is introduced:

$$\hat{p}_g^{tot} - k_0^2 \sum_j \hat{G}_{g,j} \chi_j^c \hat{p}_j^{tot} \Delta V = \hat{p}_g^{inc} . \quad (4.3)$$



Using the definition of matrix-vector multiplication, the g -th element of (4.1) can be rewritten as

$$\begin{aligned} (\mathbf{L}\mathbf{u})_g &= f_g \\ \sum_j L_{g,j} u_j &= f_g . \end{aligned} \quad (4.4)$$

Using the same definition, the g -th element of equation (4.3) can be rewritten as

$$\sum_j \left(\delta_{g,j} - k_0^2 \hat{G}_{g,j} \chi_j^c \Delta V \right) \hat{p}_j^{tot} = \hat{p}_g^{inc} . \quad (4.5)$$

Therefore equation (3.21) can be expressed like equation (4.1), with the following identifications:

$$f_g := \hat{p}_g^{inc} \quad (4.6)$$

$$u_j := \hat{p}_j^{tot} \quad (4.7)$$

$$L_{g,j} := \delta_{g,j} - \frac{\omega^2}{c_0^2} \hat{G}_{g,j} \chi_j^c \Delta V . \quad (4.8)$$

It should be noted that in matrix notation the size of $\hat{G}_{g,j}$ is $N \times N$, i.e. the number of all voxels squared. Already for coarse grained discretisations of 64^3 , the RAM required to store $\hat{G}_{g,j}$ is 512 GB. The computational effort required to fill this matrix will exceed the time required to solve it. Therefore, we can only use methods that use matrix-vector products and hence do not require the full matrix to be known.

4.1.1 The Adjoint of L

For solving equation (4.1) using a conjugate gradient method L needs to be self-adjoint. This is not necessarily the case. However, it can be accomplished, by using $L^\dagger L$ instead of L , where † denotes the adjoint and $L^\dagger L$ is selfadjoint.

To find the adjoint operator of L we use the definition of the scalar product:

$$\langle \mathbf{a} | \mathbf{b} \rangle = \sum_n a_n b_n^\dagger \quad (4.9)$$

and the definition of the adjoint operator

$$\langle \mathbf{f} | \mathbf{L}\mathbf{u} \rangle = \langle \mathbf{L}^\dagger \mathbf{f} | \mathbf{u} \rangle . \quad (4.10)$$



Together with equation (4.3) and (4.8) the adjoint operator can be derived:

$$\begin{aligned}
\langle \mathbf{f} | \mathbf{L} \mathbf{u} \rangle &= \sum_g \mathbf{f}_g (L \mathbf{u})_g^\dagger & (4.11) \\
&= \sum_g f_g \left(u_g^\dagger - k_0^2 \sum_j \hat{G}_{g,j}^\dagger \chi_j^\dagger u_j^\dagger \Delta V \right) \\
&= \sum_g f_g u_g^\dagger - k_0^2 \sum_g \sum_j f_g \hat{G}_{g,j}^\dagger \chi_j^\dagger u_j^\dagger \Delta V \\
&= \sum_j f_j u_j^\dagger - k_0^2 \sum_g \sum_j f_g \hat{G}_{g,j}^\dagger \chi_j^\dagger u_j^\dagger \Delta V \\
&= \sum_j \left(f_j - k_0^2 \sum_g f_g \hat{G}_{g,j}^\dagger \chi_j^\dagger \Delta V \right) u_j^\dagger \\
&= \sum_j (\mathbf{L}^\dagger \mathbf{f})_j u_j^\dagger \\
&= \langle \mathbf{L}^\dagger \mathbf{f} | \mathbf{u} \rangle
\end{aligned}$$

$$\text{with: } (\mathbf{L}^\dagger \mathbf{f})_j = f_j - k_0^2 \chi_j^\dagger \sum_g f_g \hat{G}^\dagger(\mathbf{x}_g - \mathbf{x}_j) \Delta V$$

$$\text{and therefore: } \mathbf{L}^\dagger \mathbf{f} = \mathbf{f}(\mathbf{x}) - k_0^2 \chi^\dagger(\mathbf{x}) \int \mathbf{f}(\mathbf{x}') \hat{G}^\dagger(\mathbf{x} - \mathbf{x}') dV(\mathbf{x}') . \quad (4.12)$$

Most notably is that $\chi^\dagger(\mathbf{x})$ is not part of the integral in equation (4.12).

4.2 Conjugate Gradient Method

Many methods exist for solving equation systems. In this work a conjugate gradient (CG) scheme will be used. The conjugate gradient method is an algorithm for finding an optimal solution of equations like equation (4.1). Instead of the steepest descent here conjugate directions for updating the initial estimate are used. Furthermore, the search directions are each \mathbf{L} -orthogonal (i.e. orthogonal with respect to \mathbf{L}) to each other, which guarantees convergence after N steps, with N the size of the problem, the number of voxels in our case. Often much less steps are required. It depends on the properties of the kernel matrix \mathbf{L} , how efficiently it can be solved and which CG schemes can be applied. [KB91a] and [KB91b] describe and compare various iterative solutions to the problem in equation (4.1).

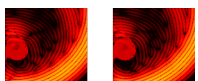
Requirements made on \mathbf{L} are that it is selfadjoint and positive definite. These criteria are problem specific and not fulfilled in general. However, the problem can be turned into a self-adjoint problem. This can be done by left multiplication of both sides of equation (4.1) with \mathbf{L}^\dagger and therefore solving

$$\mathbf{L}^\dagger \mathbf{L} \mathbf{u} = \mathbf{L}^\dagger \mathbf{f} \quad (4.13)$$

This equation is also known as the normal equation. It can be solved because \mathbf{L} and \mathbf{L}^\dagger are both known.

To derive the iterative scheme, the desired solution \mathbf{u} is thought to be composed of a previous solution which is updated with each step of the iteration:

$$\mathbf{u}_n = \mathbf{u}_{n-1} + \alpha_n \mathbf{d}_n , \quad (4.14)$$



where $n = 1..N$ is the iteration step, \mathbf{u}_0 arbitrary, \mathbf{d}_n the update directions and α_n the size of each step into direction \mathbf{d}_n . The difference between the correct solution \mathbf{f} and the current estimate \mathbf{u}_n is called residual:

$$\begin{aligned}\mathbf{r}_n &= \mathbf{f} - \mathbf{L}\mathbf{u}_n \\ &= \mathbf{r}_{n-1} - \alpha_n \mathbf{L}\mathbf{d}_n.\end{aligned}\quad (4.15)$$

The normalised residual error made at each iterative step is defined as

$$F_n = \frac{\|\mathbf{f} - \mathbf{L}\mathbf{u}_n\|^2}{\|\mathbf{f}\|^2}.\quad (4.16)$$

Equation (4.16) is also called energy norm, because the quadratic forms of \mathbf{f} and \mathbf{r} relate to expressions for the energy.

4.2.1 The update directions

To express the update step in terms of our problem, we use the Fréchet derivative on F_n for an arbitrary directional vector \mathbf{v}_n and use equations (4.15) and (4.16):

$$\begin{aligned}\partial F_{n-1} &= \lim_{\varepsilon \rightarrow 0} \frac{F_{n-1}(\mathbf{u}_{n-1} + \varepsilon \mathbf{v}_n) - F_{n-1}(\mathbf{u}_{n-1})}{\varepsilon} \\ &= \lim_{\varepsilon \rightarrow 0} \frac{\|\mathbf{f} - \mathbf{L}\mathbf{u}_{n-1} - \varepsilon \mathbf{L}\mathbf{v}_n\|^2 - \|\mathbf{f} - \mathbf{L}\mathbf{u}_{n-1}\|^2}{\varepsilon \|\mathbf{f}\|^2} \\ &= \lim_{\varepsilon \rightarrow 0} \frac{-2\varepsilon \operatorname{Re} \langle \mathbf{f} - \mathbf{L}\mathbf{u}_{n-1} | \mathbf{L}\mathbf{v}_n \rangle}{\varepsilon \|\mathbf{f}\|^2} + \frac{\varepsilon^2 \|\mathbf{L}\mathbf{v}_n\|^2}{\varepsilon \|\mathbf{f}\|^2} \\ &= -\frac{2 \operatorname{Re} \langle \mathbf{L}^\dagger \{ \mathbf{f} - \mathbf{L}\mathbf{u}_{n-1} \} | \mathbf{v}_n \rangle}{\|\mathbf{f}\|^2} \\ &= -\frac{2 \operatorname{Re} \langle \mathbf{L}^\dagger \mathbf{r}_{n-1} | \mathbf{v}_n \rangle}{\|\mathbf{f}\|^2}\end{aligned}\quad (4.17)$$

This equation is fulfilled for a vector \mathbf{v}_n parallel to the expression on the other hand of the scalar product.

$$\mathbf{v}_n = -\frac{2}{\|\mathbf{f}\|^2} \mathbf{L}^\dagger \mathbf{r}_{n-1}.\quad (4.18)$$

Re is the real-value operator, returning the real part of the expression it acts on. A vector parallel to \mathbf{v}_n is chosen as update direction. The length of the vector is not relevant yet, because we still need to define the length of the step to be taken. This liberty is used to drop the constant factor $-\frac{2}{\|\mathbf{f}\|^2}$, thereby arriving at the update directions for each step n :

$$\mathbf{d}_n = \mathbf{L}^\dagger \mathbf{r}_{n-1}.\quad (4.19)$$



4.2.2 The step length

To determine the length of the step along \mathbf{d}_n we need to know two things. Firstly F_n needs to be minimized along \mathbf{d}_n . Basic calculus requires $\partial_\alpha F_n = 0$. Due to the requirement of positive definiteness of our problem, we already know that the point at which $\partial_\alpha F_n = 0$ is a minimum. Combined with equation (4.15) this gives:

$$\begin{aligned}
0 &= \partial_\alpha F_n \\
&= \partial_\alpha \langle \mathbf{f} - \mathbf{L}\mathbf{u}_n | \mathbf{f} - \mathbf{L}\mathbf{u}_n \rangle \\
&= \partial_\alpha \langle \mathbf{r}_{n-1} - \alpha_n \mathbf{L}\mathbf{d}_n | \mathbf{r}_{n-1} - \alpha_n \mathbf{L}\mathbf{d}_n \rangle \\
&= \partial_\alpha \{ \|\mathbf{r}_{n-1}\|^2 + \alpha_n^2 \|\mathbf{L}\mathbf{d}_n\|^2 - 2\alpha_n \mathcal{R}e \langle \mathbf{r}_{n-1} | \mathbf{L}\mathbf{d}_n \rangle \} \\
&= 2\alpha_n \|\mathbf{L}\mathbf{d}_n\|^2 - 2 \mathcal{R}e \langle \mathbf{r}_{n-1} | \mathbf{L}\mathbf{d}_n \rangle \\
\Leftrightarrow \alpha_n &= \frac{\mathcal{R}e \langle \mathbf{r}_{n-1} | \mathbf{L}\mathbf{d}_n \rangle}{\|\mathbf{L}\mathbf{d}_n\|^2}.
\end{aligned} \tag{4.20}$$

4.2.3 The resulting CG scheme

Writing the previous results together in one place, we arrive at the following scheme

$$\begin{aligned}
\mathbf{u}_0 &= \mathbf{0} \\
\mathbf{r}_0 &= \mathbf{f} \\
\text{for each } n &\geq 1 \\
\mathbf{d}_n &= \mathbf{L}^\dagger \mathbf{r}_{n-1} \\
\alpha_n &= \frac{\langle \mathbf{r}_{n-1} | \mathbf{L}\mathbf{d}_n \rangle}{\|\mathbf{L}\mathbf{d}_n\|^2} \\
\mathbf{u}_n &= \mathbf{u}_{n-1} + \alpha_n \mathbf{d}_n \\
\mathbf{r}_n &= \mathbf{r}_{n-1} - \alpha_n \mathbf{L}\mathbf{d}_n \\
&= \mathbf{f} - \mathbf{L}\mathbf{u}_n.
\end{aligned} \tag{4.21}$$

The residual \mathbf{r}_n can be determined in two ways and represents a trade-off between computational speed and accuracy. While the first form requires only one Matrix-Vector product ($\mathbf{L}\mathbf{d}_n$) to be computed per iteration, the second form requires two (because also $\mathbf{L}\mathbf{u}_n$ is required). The advantage of the second form, however, is that the direct computation of the residual from the current solution is more precise. Therefore this form will be used. An alternative method is suggested by Shewchuk [She94] which uses the first definition most of the time, but updates \mathbf{r}_n from time to time, using the second expression.

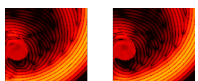
Other methods exist to overcome problems with numerical accuracy. Fletcher and Reeves ("FR") as well as Polak and Ribière ("PR") suggest the following modifications to the update direction \mathbf{d}_n :

$$\mathbf{d}_n = \mathbf{L}^\dagger \mathbf{r}_{n-1} + \beta_n \mathbf{d}_{n-1} \tag{4.22}$$

where β is defined as follows by the respective authors:

$$\beta_{n,FR} = \frac{\|\mathbf{d}_{n-1}\|^2}{\|\mathbf{d}_n\|^2} \tag{4.23}$$

$$\beta_{n,PR} = \frac{\langle \mathbf{d}_n | \mathbf{d}_n - \mathbf{d}_{n-1} \rangle}{\|\mathbf{d}_{n-1}\|^2}. \tag{4.24}$$



4.3 Implementation

In principle a solution to equation (3.21) can be implemented straight forward using the formulas derived up to here. However, a few areas mainly related to implementational aspects, have not been addressed so far.

4.3.1 Numerical precision

One important aspect is the precision of the computations, which is finite on computers. Throughout this work floating point variables with 64 bit precision are used. The IEEE standard IEEE-754 (see [Gol91]) standardises compatible implementation of floating point numbers.

Numerical rounding errors occur whenever a value is stored in a floating point variable. Additionally, adding variables, especially with largely different values, causes rounding errors (see [Gol91]). This is due to the nature of floating point values which are described by their significand multiplied with an exponent. When adding values with largely different exponents, the value of the smaller one has to be stored within the significand of the resulting value. It is therefore likely to lose the information of the smaller value.

To not increase the errors made two precautions are taken in this thesis. As discussed in (4.2.3), values will be computed from original sources rather than from derived values.

The other attempt to reduce errors is to define the physical values so that their order of magnitude is close to one. This is done to reduce errors introduced by adding or multiplying values of different magnitude. A concrete example for this is the definition (3.22), which is often defined like

$$\chi^{\text{bad}}(\mathbf{x}) = \left(\frac{1}{c_s^2(\mathbf{x})} - \frac{1}{c_0^2} \right). \quad (4.25)$$

However, the following definition expresses the same physics:

$$\begin{aligned} \chi^{\text{good}}(\mathbf{x}) &= \left(\frac{c_0^2}{c_s^2(\mathbf{x})} - 1 \right) \\ &= \frac{1}{c_0^2} \chi^{\text{bad}}(\mathbf{x}), \end{aligned} \quad (4.26)$$

but, the factor $\frac{1}{c_0^2}$ introduces a difference of more than 6 orders of magnitude.

Sorted additions, finally, where smaller values are added before larger ones are not utilised in this thesis.

4.3.2 Memory considerations

The Matrix \mathbf{L} when generated for all g and j , where $g, j \in 1..N$ and N is the total number of voxels. The memory required for keeping this matrix in Memory grows very fast. Already for $N = 64^3$ voxels 512 MB RAM are required, while for realistic resolutions of 1024^2 , 8 Zetabyte RAM would need to be allocated. Of course, computing all entries for the non-sparse Matrix \mathbf{L} would not be possible.

Fortunately, \mathbf{L} never appears alone anywhere in the derived formulae. Therefore we can always use equation (4.4) to compute the g -th entry of any desired operator.



4.3.3 Convolutions

The Green's function (see equation (3.16)) is used, always as a function of the distance between two points (see equation (3.21) and equation (4.3)). The resulting integral to compute the pressure therefore has the form of a convolution. It can be shown [Wei, Wik, Bra00, AW05] that the convolution of two functions is equivalent to the multiplication of their Fourier representations, followed by an inverse Fourier transform.

$$u(\mathbf{x}_g) = \int \hat{G}(\mathbf{x}_g - \mathbf{x}')f(\mathbf{x}')dV(\mathbf{x}') \quad (4.27)$$

$$= \mathcal{F}^{-1} \{ \mathcal{F}[\hat{G}(\mathbf{x})]\mathcal{F}[f(\mathbf{x})] \} |_{\mathbf{x}=\mathbf{x}_g} . \quad (4.28)$$

Equation (4.27) introduced a computational advantage. We no longer need to compute the Green's function on all possible combinations of \mathbf{x} and \mathbf{x}' , which in practice comes close to filling the whole $N \times N$ matrix as mentioned in 4.3.2. Only $\hat{G}(\mathbf{x})$ needs to be computed for all voxels \mathbf{x} . I.e. instead of our problem growing with N^2 it only grows with N .

This advantage comes at the cost of computing the Fourier transformation three times. But as is shown in figure 4.1(left), the Fourier transform (fftw [FJ05], Matlab implementation) scales with $N^{1.2}$. Therefore, and for memory consumption reasons it will always be faster to use the convolution theorem.

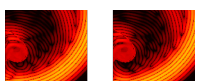
It should be noted that the fftw algorithm is supposed to scale with $N \log N$. However, it was not possible to reproduce this in the measurements made.

The convolution theorem furthermore requires the range of x in equation (4.27) to be larger than the range x_g which is required to be computed. This is due to the periodicity of the Fourier transform. If the range of x is not extended accordingly, a wave leaving the region on one side would enter the region from the other side. This results in a factor of two for each spatial dimension, which has to be computed and to be kept in memory.

Furthermore, for the discretisation in each direction x, y and z we use numbers that are powers of two, because the fast Fourier transform [BR76] performs best for these sizes.

4.3.4 Testing correctness

One simple but reliable way to test the correctness of the implementation and the achievable numerical precision simultaneously is to make use of the definition of the adjoint operator L^\dagger , equation (4.10). By computing both sides of (4.10) it is possible to see if both sides yield the same values within the error margin. This verifies the correct implementation of the adjoint operator. Both values will not be identical due to the finite precision of computers. The amount of equal digits on both sides serves as a relative metric for the numerical errors which have to be expected in the later result. The implementations in this thesis are made in Matlab, using double precision according to the IEEE-754 standard. According to [Kah96], two values are equal within this double specification, if their first 15 digits are equal.



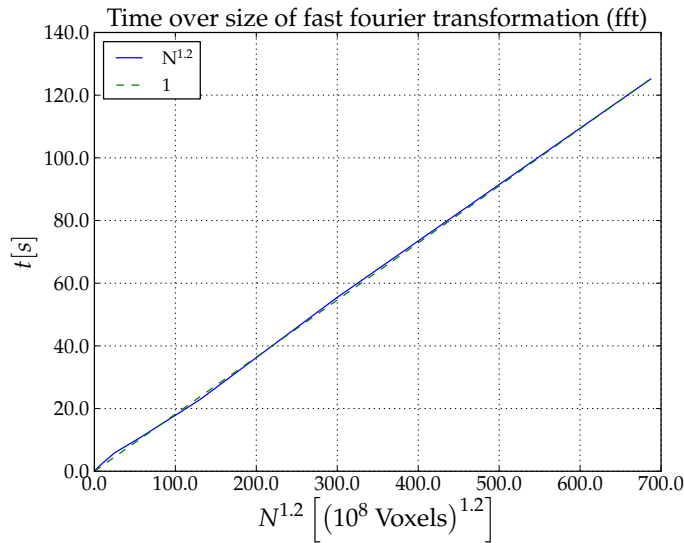


Figure 4.1: Performance of the fast Fourier transform Matlab. N was chosen as powers of two, where the fftw algorithm performs best. It can be clearly seen that the fft performance in Matlab scales with $N^{1.2}$, rather than the expected $N \log N$.

4.4 Discretisation and signal processing boundaries

Simulations have to work on a discretised copy of the problem and to be implementable on a computer. This section documents which boundary conditions have to be respected and why they are necessary.

The ultrasound waves simulated are discrete in space and time. When discretising waves it must be ensured that its shortest wavelength can still be properly sampled. This was quantified by Nyquist and Shannon in their Nyquist-Shannon sampling theorem [NS28]. The requirement is that any sampled signal can be reconstructed from a minimum of discrete steps. The sampling theorem states that at least two discrete points per wavelength are required to properly sample a signal.

Since several frequencies have to be taken into account in the simulations, it should be clarified that a wave pulse consists of a whole spectrum of frequencies. Two points are used for the shortest wavelength, i.e. for f_{max}). Typically, the shortest wavelength has an amplitude of less than a factor 10^5 less than the central wavelength.

This has several consequences on temporal and spatial discretisation that shall be discussed in this subsection.

4.4.1 Spatial discretisation

The spatial discretisation step is described by the length of a step Δx taken along spatial direction. In the cartesian coordinate-system used, the steps taken into each direction are equal, unless otherwise mentioned. The step length is determined by the amount of steps and the size of the Region Of Interest (ROI) which we want



to simulate. For a cube with side length L_{ROI} divided into $N = N_x N_y N_z$ discrete points, the side length Δx of each Voxel is

$$\begin{aligned}\Delta x &= \frac{L_{ROI}}{\sqrt[3]{N}} \\ &= \frac{L_{ROI}}{N_x},\end{aligned}\quad (4.29)$$

if $N_x = N_y = N_z$.

According to the sampling theorem, the shortest wavelength λ_{min} which can be modelled within a given spatial discretisation Δx , taking into account that the wave might travel diagonally across the pixels, is:

$$\begin{aligned}\lambda &\geq \lambda_{min} \\ &\geq 2\sqrt{3}\Delta x.\end{aligned}$$

Furthermore, to avoid any errors to be caused by insufficient sampling, in this work four or more points per wavelength will be used, leading to:

$$\begin{aligned}\lambda &\geq \lambda_{min} \\ &\geq 4\sqrt{3}\Delta x.\end{aligned}$$

4.4.2 Temporal discretisation

Also for temporal sampling the Nyquist-Shannon theorem is valid. It requires that the sampling frequency has to be twice as high as the highest frequency modelled. Often sample frequencies rather than timing values are used. They are related via

$$f_{Sample} = \frac{1}{\Delta t}.\quad (4.30)$$

For a wave of the minimal wavelength λ_{min} in a medium with speed of sound c the frequency is

$$f_{max} = \frac{c}{\lambda_{min}}.$$

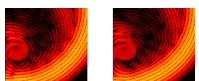
This leads to a relation between the sample rate and the spatial discretisation:

$$f_{Sample} \geq 2f_{max}\quad (4.31)$$

$$\begin{aligned}&\geq \frac{c}{2\sqrt{3}\Delta x} \\ &\geq \frac{cN_x}{2\sqrt{3}L_{ROI}}.\end{aligned}\quad (4.32)$$

To derive an expression for the amount of time samples which are needed we stipulate that we want to simulate the wave this long that it can cross the ROI twice, diagonally. For this we need to simulate a timespan $T = N_t \Delta t$:

$$N_t \Delta t = 2 \cdot \sqrt{3} L_{ROI} \cdot \frac{1}{c}.\quad (4.33)$$



With the expressions for Δt in equation (4.30) and the sample frequency in equation (4.32) an astonishingly simple expression for the number of temporal sampling points can be derived:

$$\begin{aligned} N_t &= T f_{\text{Sample}} \\ &= 2\sqrt{3} L_{\text{ROI}} \frac{1}{c} \cdot \frac{c N_x}{2\sqrt{3} L_{\text{ROI}}} \end{aligned} \quad (4.34)$$

$$= N_x . \quad (4.35)$$

To be able to exploit fast implementations of the Fourier transform N_x and therefore also N_t will be chosen as powers of 2.

4.4.3 Simulations in temporal Fourier domain

We know that the Fourier transform of real valued signals is symmetric in the real part and antisymmetric in the imaginary part. Since all simulations take place in temporal Fourier domain, we can take advantage of this by only simulating the first half and constructing the rest of it. For a signal of length $2N$, we have to compute $N + 1$ values and can interpolate the $N - 1$ remaining values. In cases where Matlab is used, this can be accomplished by using the “symmetric” option of the “ifft” command.

4.4.4 Incident Pulse

Up to now we have only taken monochromatic waves into account. Wave pulses can be described by a superposition of many monochromatic waves each with a different frequency and amplitude. In fact the Fourier transform (see B.2) is the relation between the wave pulses and the amplitude spectrum of monochromatic waves.

In terms of this section, care must be taken to properly choose the frequencies, especially towards f_{max} . We therefore choose a Gaussian distribution of the frequency spectrum around the centre frequency f_{centre} .

The bandwidth definition used is FWHM (full width half max), i.e. the width of the power spectrum at the half maximum.

To avoid unwanted effects, we take care that the amplitude of the frequency spectrum towards the boundaries $f = 0$ and $f = \frac{1}{2}f_{\text{sample}}$ is lower than -40 dB.

4.4.5 Parameters summarised

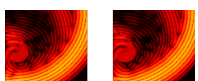
The spatial discretisation or resolution is the only parameter which is modified. All other values are either given by the dimensions of the system or can be derived.

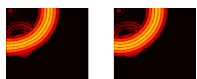
Table 4.1 shows the values and formulas used to determine all simulation parameters.



Parameter name	Value / Formula	Reference	Comment
N_x	32-1024		
L_{ROI}	0.2 m		
Δx	$\frac{L_{ROI}}{N_x}$	(4.29)	
c_0	1570 $\frac{m}{s}$	Table 3.1	
c_s	1460-1590 $\frac{m}{s}$	Table 3.1	
N_t	N_x	(4.35)	Rounded to the next power of 2
f_{sample}	$\frac{cN_x}{2\sqrt{3}L_{ROI}}$	(4.32)	
f_{max}	$\frac{1}{2}f_{sample}$	(4.32)	
f_{centre}	$\frac{1}{2}f_{max}$	(4.31)	
Bandwidth	16 % of f_{centre}		

Table 4.1: List of all simulation parameters including their values or formulas and reference.





COMPUTING INFRASTRUCTURE

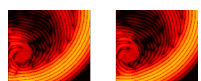
This chapter focuses on utilising distributed computing infrastructures (DCI) with emphasis on the interactive use of infrastructures from within problem solving environments such as Matlab. Several solutions will be analysed with respect to problems which users often experience when using parallel facilities. For this a list of requirements will be compiled in 5.1 and serve as guideline for the work described later in this chapter. The used DCIs will be introduced in section 5.2. Section 5.3 describes a pilot-implementation which served to understand the important aspects which lead to the final implementation shown in 5.4.

5.1 Requirements

Many scientific and engineering problems today are too large to be solved on a single computer. Despite the fact that workstations provide multiple cores on several CPUs, still the development of demanding scientific and engineering applications reaches the point at which results cannot be obtained in time on a single computer. A cluster of workstations may be required already at an early development stage of the code. In many situations such clusters are fully booked in advance and require computing bundled into jobs. Similar jobs of other users wait in queues before they can finally be processed. In many cases the time spent in the queue is longer than the jobs runtime on the cluster – but often this is the fastest option for users.

However, for software development, frequent runs and tests are essential. Long turnaround times between implementation and obtaining results have therefore a strong impact on software development efficiency. This often underestimated use-case provides the motivation for this chapter.

The demanding software algorithm, on whose implementation I will show the usefulness of the developed approach, is the 3D simulation of ultrasound waves as described in the previous chapters. These simulations are required for two scenarios important to the USCT project. One is the simulation of ultrasound measurement data for which by principle the ground truth is known. This data can be used for the



evaluation and further development of reconstruction algorithms, which are not part of this work. The other is a future reconstruction algorithm, which requires the simulated wave-field based on a first estimate of the object function to take into account multiple scattering and to reduce artefacts otherwise created.

The results of the 3D wave simulation algorithm have to be verified by comparing them to those obtained with alternative methods. In our case an analytical solution implemented by Koen van Dongen of the University of Delft was available.

The results of both simulations can only be compared when their spatial and temporal discretisations are fine-grained enough, because otherwise differences in between both approaches can not be visualised. This is the case for spatial and temporal coordinates. Since the time discretisation directly corresponds to the spatial discretisation (see section 4.4.2), few discrete steps dictate a low bandwidth for the pulse being used. The problem with low bandwidths is that the pulse length increases. Therefore, coarse grained discretisations lead to aliasing effects. The errors thereby introduced prevent the comparison with the results of the analytical solution. Furthermore, the algorithm developed in this thesis requires to be run in full 3D to provide correct results.

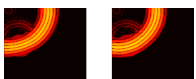
These considerations lead to minimal discretisation boundaries of 64^3 Voxels and 32 frequencies. Even when efficiently implemented this requires roughly 10 min computing time on a dual-core (Intel T9400 @ 2.53GHz) CPU. The development of a correct sound wave simulation is a difficult task. Therefore, the development cycle requires frequent tests, code changes and the results need to be returned while the programmer is still aware of the changes made to the code. If the same result can be obtained in only one minute, ten times more test cycles can be run in the same time. This results in a profound increase in developer performance.

The work within this thesis provides a solution to the aforementioned problems insofar as it offers a simple to use and comfortable interface to make use of remote resources and to offload computations to them.

The experience gathered in past grid projects and throughout this thesis have led to a list of eight requirements. These requirements are imposed on the interface for using distributed computing infrastructures from the users' point of view, but with the technological capabilities in mind.

Nr.	Requirement
1	Access distributed resources with neglectable queueing time
2	Synchronise the latest version of the users software to all distributed resources
3	Optionally compile the software
4	Provide interfaces to run the software remotely
5	Transport input data to the remote resources
6	Run and monitor the remote execution
7	Return the remotely obtained simulation results
8	Integrate as smoothly and user-friendly as possible

Table 5.1: Requirements to be fulfilled by the ideal interface to distributed resources



To address the points of this list, the existing Distributed Computing Infrastructures (DCIs) are introduced and tools for the respective DCIs are evaluated with respect to this list.

5.2 Distributed Computing Infrastructures

This section introduces the three different types of DCIs used in this thesis. The grid is the largest infrastructure available to scientists today. However, a strong trend towards using cloud resources can be observed also in the scientific community [SBB⁺10]. Cluster computing is still available because this is just another way to access cloud or grid resources. It can be provided without additional effort. It is important to note that the tools I developed in this thesis were tested to work on all three types of resources: local clusters, grid and also cloud resources.

5.2.1 Grid Computing

The idea of grid computing emerged in the end of the 1990s, most notably pursued by Ian Foster and Carl Kesselman [FK98, FK04]. Their leading paradigm was to decouple users from resources. The goal was, that computer centres would join their resources into one big grid of resources. In return, any user who was allowed to use computational resources at one computer centre would be allowed to also use resources at the other computer centres, as well. Two problems were addressed in this way. One was that computations which require more resources than one single computer centre can provide could be run by allocating several computer centres. The other was the increase of resource efficiency, because more users can be scheduled more efficiently if more resources are available. Maintenance work would affect less users. The likelihood of running jobs without queue time increases.

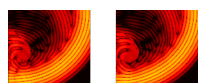
Enabled by a sophisticated authentication and authorisation infrastructure based on X.509 [Mo189] certificates, the paradigm of “acting on behalf of the user” was established. This is required for example, when a job needs to authenticate at a remote storage system for writing results.

Kesselman and Foster created the de facto standard for grid computing by bundling the developed tools into the Globus toolkit [FK97].

Building on top of this toolkit and substantially extending it, the LHC computing grid project (LCG) [ABB⁺02] has established its grid middleware gLite [Lau04] and runs the largest grid infrastructure available to scientists today. At time of writing, 140 computer centres offered (317 139 CPU cores, 186.4 PB disk space and 178 PB tape space¹). Most important improvements of gLite over globus were logical file catalogues and automated resource brokering. A file catalogue stores all physical replicas of a file. This information is used for resource brokerage for scheduling jobs to a specific site which already holds a file.

The grid middleware Unicore [ES01] focuses on HPC, where resource are typically reserved well in advance based on tendering procedures. Additionally, dedicated hardware requires to adapt the codes for optimal performance. Therefore, Unicore is not in the focus of this work. However, it should be noted that consid-

¹<http://wlcg-rebus.cern.ch/apps/capacities/sites/>



erable effort [Kra09, NKG07, Hof11] was taken to make Unicore, gLite and other flavours of globus (e.g. ARC [SBa11]) compatible to each other.

Extension to the LHC-Grid

The infrastructure offered by computational grids provides a rich set of services on resources distributed across the world.

With respect to the requirements, defined in the beginning of this chapter, the grid offers access to remote resources without queueing-time, i.e. requirement 1 is fulfilled, but all other points are not addressed so far.

To overcome these limitations, this work was launched with partial funding of the Interactive European Grid Project [MCC⁺08] (int.eu.grid). This project focussed on interactive extensions and also MPI support on the grid. The developments for interactive grid access in this thesis were developed within the Interactive European Grid Project (int.eu.grid).

5.2.2 Cloud Computing

Conceptually newer is Cloud Computing. The name and concept are based on the infrastructure built up at Amazon.com, which was frequently depicted using a cloud symbol. The conceptual novelties were the use of virtualisation and the introduction of a business model. From a technological point of view there were – besides the use of virtual machines – no improvements when compared to grid computing. Instead of introducing new interfaces, protocols, tools and authentication mechanisms, cloud computing uses available tools that were well known by users already before cloud computing. This lack of novel developments made the use of cloud infrastructure a lot easier. Little new knowledge is required to obtain first results.

The existence of a working business model led to several companies offering cloud computing infrastructures as a service (IaaS). Market competition led to attractive prices for using cloud computing.

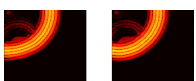
Virtualisation enables users to gain administrative rights on the rented virtual machines. Users can provide their own virtual disk image, which is then run at the providers computer centre. Compared to the grid infrastructure, users are now free to decide which software to use without the need of requesting grid administrators to install it grid-wide. Even commercial software and alternative operating systems (such as Windows) can be used within the cloud scenario. These are the advantages that make cloud computing attractive for scientific users.

To allocate cloud resources, a virtual machine image needs to be created. The image contains all required software and user customisations.

Cloud Infrastructure Service at KIT

The cloud resources used for tasks within this thesis are those provided by the KIT computer centre SCC (Steinbuch Centre for Computing). Using the cloud middle-ware OpenNEbula [SMLF09] users can request virtual machines. The underlying virtual disk image is provided for own customisation. To create a virtual cluster many virtual machines based on the same disk image are started.

The hardware to provide cloud services consists of 55 machines, each equipped with 2 Intel Xeon 2.27GHz CPUs and 36 GB RAM. Use of hyperthreading offers



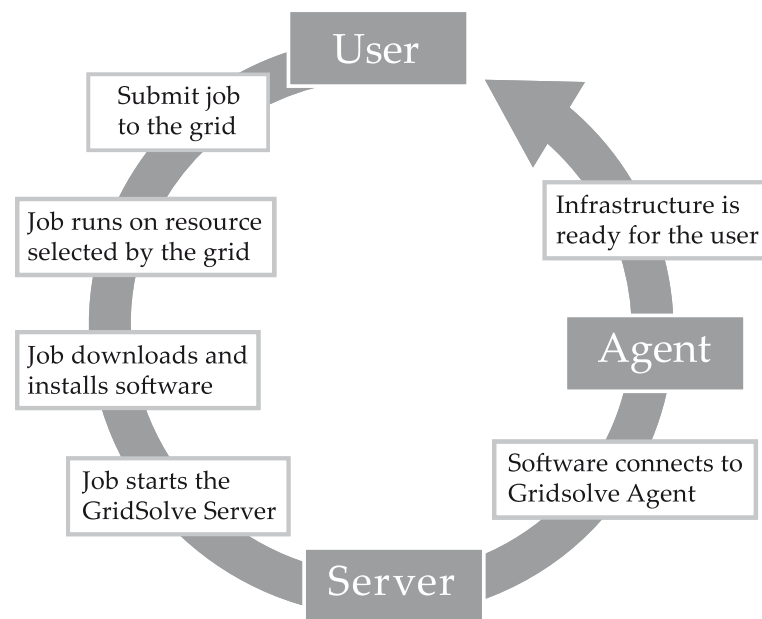


Figure 5.1: Integration of the grid and the GridSolve middlewares using software developed within this thesis.

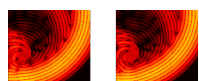
16 CPU cores to be used. The cluster is shared with a Hadoop batch-system, but for benchmarks and larger runs for this thesis exclusive access was granted on the cluster.

5.2.3 Cluster Computing

Cluster Computing is the most traditional way to scale from a workstation to a larger setup. Computing clusters are the foundation of scientific grid sites and cloud providers. The difference is the way in which access is provided. In this work the cluster model is used to compare performance metrics with the cloud resources. For ideal comparison direct access to the same hardware as used in the cloud setup is necessary

5.3 Testing the Integration of Matlab into the grid

To test the available tools and make use of the resources provided by int.eu.grid interactively from within Matlab, an integrative architecture was designed and implemented as part of this thesis. It was decided to conduct a pilot implementation first which only targets basic requirements to study the feasibility of the approach taken. Therefore it was imperative to use available software components where possible so that feasibility can be determined quickly. The integration architecture and the steps it defines are displayed in fig. 5.1.



GridSolve

The tool GridSolve [YDS07] is a development of the Innovative Computing Laboratory (ICL) at University of Tennessee at Knoxville (UTK). It was chosen for the prototypical implementation, because it provides grid-RPC based remote procedure calls with an easy to use interface. It implements a “client + agent + server” architecture. On the client side many interfaces exist to support Matlab, C, Fortran and more. To run a remote procedure, the client requests a resource which can solve a given problem (i.e. execute a previously compiled and deployed remote procedure). Resources are requested via an agent to which GridSolve Servers are connected. The servers allocate resources and execute the remote calls. They correspond to the workernodes in a grid or cluster architecture.

The (hidden) semantics of a GridSolve request are:

1. Client contacts the agent for a list of capable servers.
2. Client contacts server and sends input parameters.
3. Server runs appropriate service.
4. Server returns output parameters or error status to client.

From the user’s perspective, the call to GridSolve acts just like the call to the original function. [AAB⁺02]

The interface to the developer very easy to use as shown in this example:

```

1 foo = bar (x, y, z) % original call (locally)
2 foo = gs_call('bar', x, y, z) % remote call using GridSolve

```

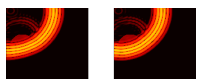
Listing 5.1: Matlab code for calling a local function *foo* (line 1) compared to calling the same function remotely, using GridSolve (line 2)

However, the creation of a remote procedure requires several additional steps. It is required to define an interface of the procedure and to deploy a compiled version of it. Furthermore, only C and Fortran are supported for creating remote services. Existing Matlab code cannot be used for this, which makes distribution of existing code a major obstacle.

GridSolve was developed independently of gLite, and was therefore not meant to be compatible. Therefore, the integrative architecture has to implement components which allow to allocate LCG workernodes to GridSolve using gLite job-submission.

To allocate resources on the grid, an LCG pilot job is sent via the grid’s Resource Broker to an appropriate server in a previously not determined computer centre. The job contains a minimal piece of software called “start”, developed within this work. *start* ensures the smooth integration of both otherwise not compatible tools. This involves the following tasks, which have been published more detailed in [HZR08, HHH10, Har08]:

1. Download and install GridSolve on the selected grid computing nodes (Workernodes).
2. Fetch the up-to-date functions that the scientific developer needs to run.
3. Start the GridSolve server and connect via a proxy to the GridSolve agent.



One technical problem that typically inhibits the use of such techniques on grid is that Workernodes typically have private IP addresses. Thus it is not easily possible to connect to the Workernode from outside. The GridSolve proxy provides a solution to this. By forwarding the servers connections of the servers to the agent and keeping it open, it allows users to connect back from the outside.

5.3.1 First results

Extensive performance benchmarks of the tested integration were conducted within the Master thesis of Torsten Hopp [Hop09]. The results cover the analysis of the speedup if running codes on the grid. The work utilised advanced interactive grid features, so that even differences of using the Message Passing Interface (MPI) or Symmetric Multi Processing (SMP) remotely are analysed.

In terms of the initially defined eight requirements, GridSolve on top of gLite, allows to fulfil nicely the additional Requirements 4 (interface to remote resources), 5 (transport input), 6 (run and monitor execution) and 7 (return results).

Before integrating higher level requirements 2 (synchronisation of software to remote resources), 3 (compile software) and 8 (user-friendliness), I decided to test the components available at this stage. The higher level requirements can easily be integrated, if it is ensured that the low-level components will not need to be changed anymore.

Requirement	Fulfilled
1 Access without queue time	<input checked="" type="checkbox"/>
2 Synchronise software	<input type="checkbox"/>
3 Compile software	<input type="checkbox"/>
4 Interface to remote resources	<input checked="" type="checkbox"/>
5 Transport input	<input checked="" type="checkbox"/>
6 Run and monitor execution	<input checked="" type="checkbox"/>
7 Return results	<input checked="" type="checkbox"/>
8 User-friendliness	<input type="checkbox"/>

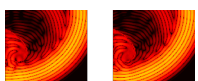
Table 5.2: Requirements fulfilled by the pilot implementation

With regard to the eight requirements defined in table 5.1, the pilot implementation performs as targeted. The higher level requirements 2 (synchronisation of software to remote resources), 3 (compile software) and 8 (user-friendliness) can now be addressed.

5.4 Final integration of DCI and Matlab

During the hardware shift from 32-bit to 64-bit the development and support for the used middleware GridSolve was discontinued. A working middleware solution had therefore to be implemented as part of this thesis. This middleware utilised the same architecture as GridSolve. As part of the new implementation the previously missing higher level requirements were also addressed:

- 2: Synchronise correct version



3: Compile the software

4: Interface to resources, but instead of the requirement of implementing in C or Fortran, allow the use of existing Matlab code.

8: Smoothly and user-friendly

Furthermore the chance was taken to adapt the already developed tools to use cloud computing resources. This section will describe the implemented solution in detail and characterise its performance.

5.4.1 PArallel INfrastructure – PAIN

PAIN is written in Python. The interface from Matlab to Python is implemented by using system calls from within Matlab. This was done to reduce the complexity introduced by using the additional programming language C and to loosen the dependency on the specific Python version.

PAIN only requires two external libraries to work. One is the network communication which is handled via Python Remote Objects (Pyro) [Jon12]. Pyro is a widely used library, so that longer term support is very likely.

The second external library is python-psutil, which is used to collect monitoring information on the servers.

PAIN is based on the client + agent + server architecture, which is shown in fig. 5.2.

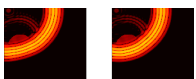
PAIN Server

The PAIN Server is the component which allocates the computing resources. One PAIN server is run on each cluster node in case of a cluster, on each worker node in case of grid, and in each virtual machine (VM) in case of cloud. For deployment, the same component as for the grid can be used: “start”. To prove the general applicability of the approach, and because resources were available, cloud resources are used in case of PAIN rather than grid resources. For this a VM image is prepared.

To start a server, the only mandatory parameter is the agent to which the server will connect. Optional parameters include specification of scratch space, which is used to store temporary files. In many cases this is required to avoid storing temporary files on a shared filesystem, which would impair system performance.

After startup, the PAIN server continuously collects monitoring information about the hardware on which it is running. These include the amount of available CPU cores, available memory, CPU utilisation and the result of a short benchmark. The latter is required, because on the cloud infrastructure the performance available to the running virtual machine can vary over time. Collecting benchmark information can help to avoid sending computing tasks to resources slowed down by other virtual machines on the same physical hardware.

An advantage introduced by using virtual machines on the cloud is additional services can be easily added. The virtual machines used in this thesis had a web-server activated which gave access to detailed monitoring information as well as to the progress of the computational tasks. All monitoring information is frequently sent to the PAIN Agent.



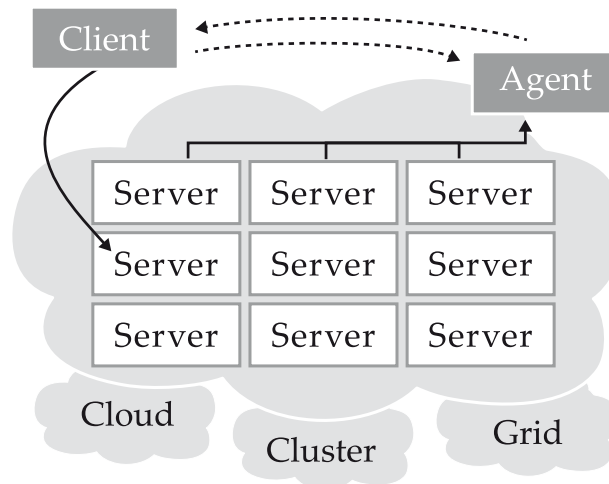


Figure 5.2: The “client + agent + server” architecture of PAIN. The servers may be deployed using cloud, cluster or grid based resources and start a private communication topology.

PAIN Agent

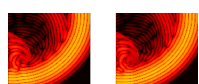
At the core of the system is the PAIN Agent. It stores the list of servers that are connected, their monitoring information and IDs of the tasks currently running on each server. Whenever a client wants to run a new task on one of the servers, it requests the optimal server from the agent. The agent returns the fastest free server for the client to be used. Servers are free when the amount of concurrently running tasks is lower than a configurable maximum. The fastest server is determined by the benchmark result, provided periodically by the servers.

The agent stores copies of the status information of the jobs in the system. This information can be used by the client to display and to manage the system. The authoritative source, however, is the server who periodically updates job status information.

Because of its central role, only one agent instance was allowed for a group of servers. It is possible to allow multiple agent instances. This would increase the complexity of the implementation to a degree not adequate for the size of the problem and therefore beyond the scope of this thesis.

PAIN Clients

Two clients were implemented within this thesis. A commandline client and a Matlab client. The commandline client is designed to display monitoring information, stored in the agent and to send administrative commands to the agent. Most notably, the client can request the agent to restart the agent and all servers. When used in combination with `start` (see 5.3) this can be used to restart the whole infrastructure with the latest software release within the order of 20 seconds. This is very fast compared to restarting via the infrastructure layer (i.e. grid, cloud) which requires several minutes.



Job submission is also supported by the commandline client. All Unix-shell commands can be used. For this it requests the best server from the agent. Then it directly connects to the server and transfers the input data. This is implemented this way to avoid a bottleneck by the agent when transmitting large amounts of data or when submitting many jobs.

The Matlab client (`mpain`) fulfils two functionalities. One focuses on the communication with the agent and the servers and is very similar to the commandline client. The notable enhancement is that the Matlab client can compile and transfer the executable together with required libraries to the server prior to running this executable. This is one of the two key advantages over the GridSolve solution presented in the previous section. The second advantage is that existing Matlab code can be used for remote and parallel runs.

`Mpain` was designed for a user who would like to run his locally developed function bar on one or more remote servers. Minimal modifications to the code are required on the calling side (see listing 5.2).

```

1 foo =          bar (a, b, c); % call function bar in matlab
2 foo =      mpain(@bar ,a, b, c); % remote call to function bar
3 id  = mpain_async(@bar ,a, b, c); % asynchronous remote call
4 foo = mpain_wait(id);           % get output of asynchronous call

```

Listing 5.2: Changes on the calling side, when using `mpain`. For simple remote calls only 8 characters need to be changed.

The “@”-sign before `bar` passes a function handle to `mpain`. This handle is used to parse the source code of `bar` so that the names for the passed parameters are known. Then `mpain` assigns the passed values to the parameter names and saves them into a `.mat` file. In this way it can be guaranteed that complex variable types such as “cell” or future variable types will be handled properly. The function handle is furthermore used to run the Matlab compiler `mcc`, which is included in our Matlab Campus license, to create a binary executable of the remote function – `bar` in this example. The output of `mcc` is packaged into a tar file. Together with the saved parameters the tar file is transferred to the server. On the server both files are extracted, installed into the proper directories and the binary version of `bar` is executed.

At this point the deployed Matlab function needs to support PAIN. The required modifications are minimal (see listing 5.3). All that is required is loading the parameters previously saved before the actual code starts as well as saving the return values to a file after the original code finished. The saved file is returned via the network, loaded from the `mpain` client as return values (i.e. to assign `foo` in listing 5.2).

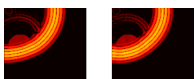
To run many remote functions at the same time, `mpain_async` is provided. After calling it immediately returns an ID. The ID can be used to retrieve the output variables after all tasks have been distributed (lines 3 and 4 in listing 5.2).

Appendix C.1 shows an example in listing C.1 on how `mpain_async` can be used to parallelise existing Matlab code.

```

1 function [foo_1, foo_2 ] = bar (param1, param2, param3, ...)
2 if isdeployed
3     % The first two parameters contain the names of the files via
4     % which input and output are handled.
5     input_filename = param1;
6     output_filename = param2;

```



```

7|     clear param1 param2;
8|     load (input_filename)
9| end
10| %
11| % Original code goes here
12| %
13| if isdeployed
14|     my_save (output_filename, 'foo_1', 'foo_2'); % get the order right
15|     exit % This extra exit is required for Matlab newer than 2008a
16| end

```

Listing 5.3: Changes required in the remote function, when using *mpain*. The first two parameters contain the input and the output filenames. The two “*isdeployed*” blocks ensure proper handling of i/o data. The code between these two blocks remains unchanged.

Step	Action
1.	Parse function handle to get parameter names
2.	Save parameters to .mat file
3.	Compile the function using the Matlab compiler <i>mcc</i>
4.	Package and transfer the output to the best server
5.	On the server: run the binary created by the Matlab compiler
7.	Load parameters before running the main code
8.	Save return values into a specified filename
9.	Transfer return value file back to client
10.	Load file and return values as if it was called locally

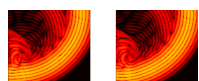
Table 5.3: *Mpain* actions for running existing Matlab code remotely

It is important to note, that due to the usage of the Matlab compiler, no Matlab licenses are required on any of the servers. In contrast to the distributed computing toolbox (which costs around 4000 EUR) only the compiler toolbox (which is included in the Matlab campus license) is required. The Matlab Compiler Runtime environment (MCR) is required to be installed on the servers for each supported Matlab version. Currently three versions are supported: Matlab 2011a, Matlab 2009b and Matlab 2008a

5.4.2 Concluding remarks

Coming back to the eight requirements defined in table 5.1, here is an analyse whether the presented solution fulfils the requirements. These points are summarised in table 5.4.

1. “Allocate distributed resources without queueing time”: The VM image has to be prepared in advance. Once the virtual machine is started (or the grid or cluster job running) it can instantly be used. Queueing only occurs when not enough resources are available. Using *start*, allocation is easily possible on grid, cloud and cluster resources.



2. *“Synchronise the correct version of the users software to all distributed resources”*: Mpain uploads the most recent version of the software right before the distributed resources are utilised.
3. *“Optionally compile the software”*: The integrated compilation in the Matlab client ensures that the most recent version of the software is compiled when it is needed. Furthermore, mpain only compiles the software, if the sourcecode was modified.
4. *“Provide interfaces to run the software remotely”*: The mpain interface works in the same way as GridSolve. Both are easy to understand and to use.
5. *“Transport input data to the remote resources”*: Mpain supports direct transfers from client to the servers, bypassing the agent for optimal scalability.
6. *“Run and monitor the remote execution”*: The agent collect monitoring information which can be visualised in several ways using the pain commandline client.
7. *“Return the results remotely obtained”*: In asynchronous mode stored output data remains at the servers until the mpain client retrieves it. In synchronous mode, the output is immediately returned.
8. *“All of this should work as smoothly and user-friendly as possible”*: This is what pain was designed for.

	Requirement	Fulfilled
1	Allocate distributed resources	☑
2	Synchronise software	☑
3	Compile software	☑
4	Interface to remote resources	☑
5	Transport input	☑
6	Run and monitor execution	☑
7	Return results	☑
8	User-friendliness	☑

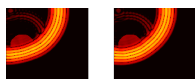
Table 5.4: Requirements table for PAIN: All requirements are fulfilled

5.4.3 Additional Tools

In the context of pain, several additional tools were developed, the two most useful are introduced below.

Test suite

To ensure proper functioning of an PAIN installation, a test suite is provided. Since testing is never easy, this testsuite is named “painful”. It performs tests of all required systems, including compilation, synchronous and parallel submission and visualises the created results.



Memory Profiler

The memory profiler for Matlab is very useful to find the places in the code where most memory is allocated. This often limits the size of the problem that can still be computed. The memory profiler “mem_stats” requires the user to instrument his code with calls to mem_stats. The profiler then logs the variable names together with their memory usage to a file. The logfile can be visualised to plot the memory usage throughout the runtime of the program. Fig 5.3 shows a typical plot created using mem_stats.

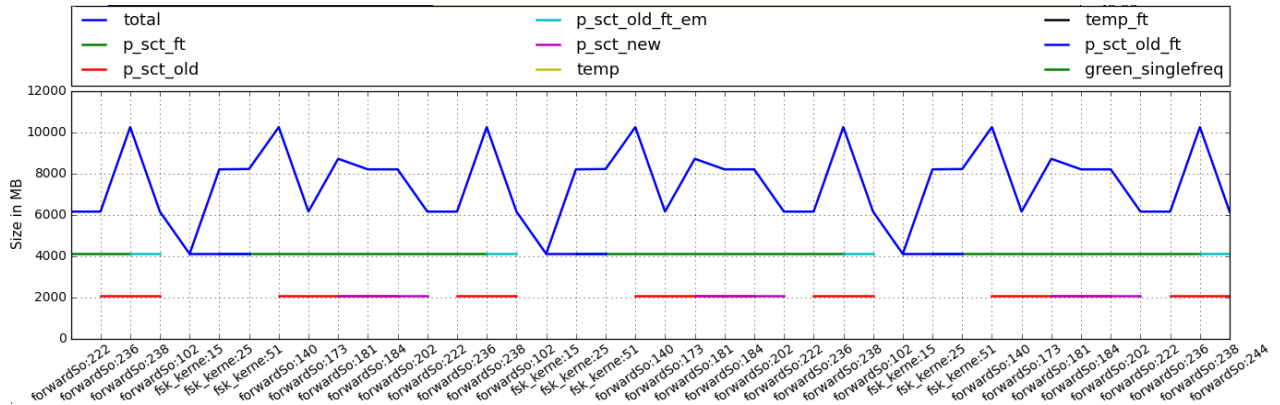


Figure 5.3: Memory plot generated using `mem_stats`. The largest nine variables are plotted over source file name and code line. Some variables are not defined in all parts of the monitored code and therefore cause the interrupted lines.

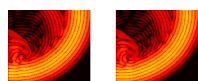
5.4.4 PAIN usage within this thesis

Within this thesis PAIN was used to speed up certain parts of the simulation:

- Distribute large simulations by emitter
- Distribute smaller simulations (less than 20 emitters) by frequency
- Distribute the point sources used to simulate the surface source
- Distribute the analytical solution by frequency
- Distributed parameter search for the best fit of the interpolation from spheric to cartesian coordinate system
- For running the benchmarks of the distributed infrastructure

5.5 Performance measurement

Understanding the performance of the system is required to characterize the efficiency of the implemented parallel infrastructure (PAIN). For the chosen setup cloud resources gives an additional degree of flexibility, because the physical nodes in the cluster are partitioned when using virtual machines.



Number of partitions [nVM]	Partition size [nCPU]	Total number of CPUs
1	16	16
2	8	16
4	4	16
8	2	16

Table 5.5: *The used partition sizes.*

In this section two performance measurements are conducted. One is aimed at finding the optimal performance of the possible partitions. The other will be used to evaluate the performance improvement which can be obtained when using PAIN.

5.5.1 Cluster partitioning

The use of cloud computing allows to request virtual machines with an arbitrary number of virtual CPUs and RAM. Of course this has to be possible within the physical limits of the underlying physical hardware. On the available cloud cluster this means that the 16 cores of a cluster node can either be used by one VM with 16 virtual cores or by 16 VMs with one core each. Thus the question is, which partition size is optimal. To find this optimal partition size, a benchmark was run on several partition sizes. In all partition configurations one physical computer was filled with VMs of the same size and every physical CPU was assigned to exactly one virtual CPU so that no idle CPUs and no duplicate allocations have occurred (see table 5.5).

Each partition configuration was setup twice to obtain the results faster and to average over short lived local jobs on the physical hardware.

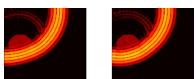
Synthetic Benchmark

The target algorithm is described in chapters 3 and 4. Profiling [Fit77] indicates that 77% of the computing time is spent running the routine “fft”. This is based on the highly optimised “fasted Fourier transform in the west” [FJ05], which automatically allocates all available CPU cores and exploits hardware acceleration wherever possible.

To obtain clearer results, a synthetic benchmark is used. To mimic the performance of the target algorithm, the benchmark consists of creating a $1000 \times 10\,000$, or 152.6 MB random number matrix which is Fourier transformed using “fft” N_{loops} times. Caching effects are avoided by using the output of one Fourier transform as input for the next.

The measurement was repeated 50 times to obtain a statistical mean value. To judge which configuration performs best, the normalised time (i.e. $t_{measurement} \cdot N_{cores} / N_{fft}$) is plotted over the number of CPU cores in fig 5.4.

It can be observed that the non virtualised configuration outperforms the other setups. This behaviour is to be expected, because virtualisation is known to introduce a small overhead [Har06]. Unexpected was the fact that the virtual machines (VMs) with eight cores are performing better than the 16 core VMs. This can be attributed to the use of hyperthreading [SD96]. This technology was introduced



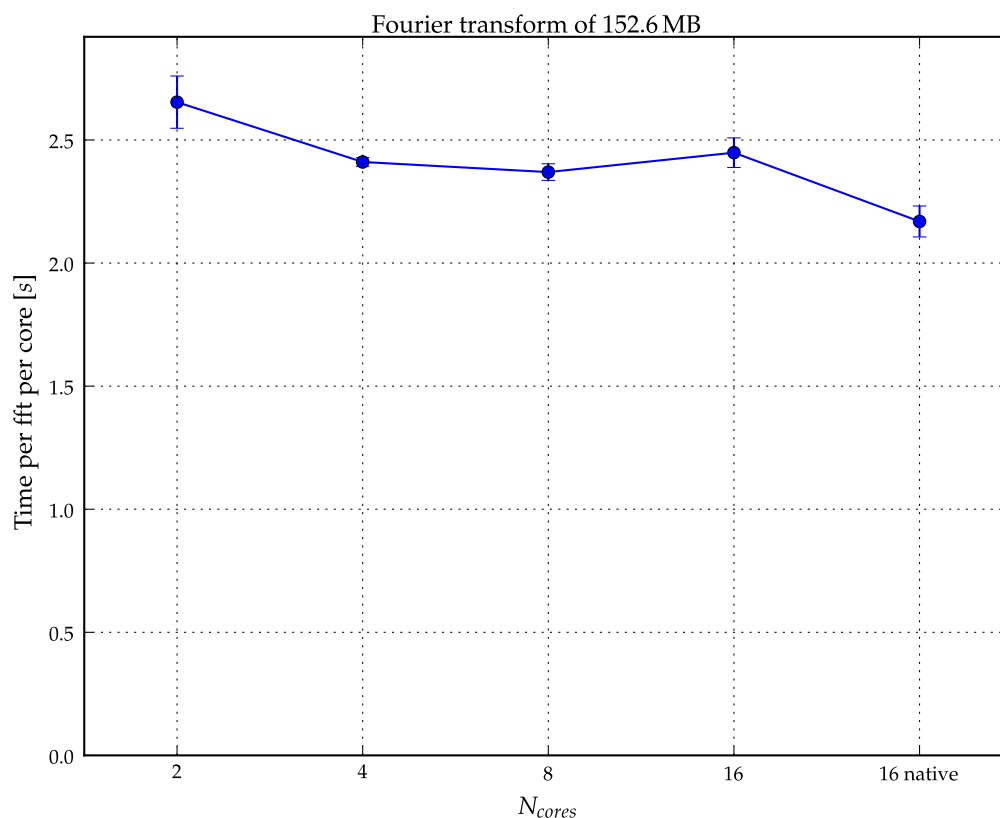
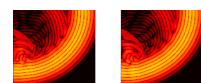


Figure 5.4: Time per Fourier transform of 152.6 MB per core over number of cores. The Fourier transform was run 5000 consecutive times to keep the overhead of distribution to a minimum. “16 native” indicates the benchmark run on the physical hardware rather than inside virtual machines. The error bars represent the statistical error of 50 measurements.

by Intel to improve the utilisation of CPU cores. In most cases a performance improvement can be observed.

I assume that the highly optimised `fftw` code tries to use optimisations on virtualised hardware it would not apply on physical hardware, because virtualisation hides the information about hyperthreading. When trying in this setup to optimise for 16 cores on 8 native cores performance can be worse than when optimising for 8 cores on 8 native cores – even when two concurrent machines are active on the same hardware. While deeper analysis of this effect is beyond the scope of this thesis, the knowledge that eight core VMs are slightly faster was exploited for the large simulations. The performance of the dual core machines is the worst because the overhead of virtualisation is larger when more VMs are run on one host.

Performance degradations were observed when sharing a physical node with other non virtual machine users, who only allocated one CPU. The degraded performance was about 10% of the original. In contrast to the previous situation in which the virtualisation layer separated physical CPUs between individual virtual machines, the non virtual machine user is scheduled by the Linux kernel. This will schedule tasks of both, the benchmark inside and the user outside the virtual



machine to the same CPU core and thereby alter the content of the cores caches. This is the possible reason for the large performance decrease of the `fftw` code. Measurements were taken to avoid hosts allocated with virtual machines and non-virtual jobs.

5.5.2 Scalability of PAIN

The scalability measurement of PAIN is conducted to understand its scalability limits. For the benchmark to be used a constant remote runtime is more important than the actual CPU performance. This is why the benchmark used in this case is the “pause” function, because its runtime is independent of how heavily the system is loaded. This allows to measure scalability on larger configurations than the 56 nodes available in the cluster. Cluster sizes of up to 192 VMs were deployed.

The problem to be distributed was modelled to be similar to the target algorithm in the way that a fixed number of tasks had to be computed on a variable amount of remote machines. In this setup the amount of data transferred via the network is constant.

To demonstrate the influence of data transfers the amount of input and output data was varied between 1 *kB* and 16 *MB* for each cluster size.

Statistic fluctuations were mitigated by running the benchmark five times per cluster size, removing up to one obviously incorrect measurements and taking the average of the remaining values.

The remote runtime per remote function T_p was chosen to be $T_p = 60$ s. This value is lower than the runtime of the target algorithm, which is in the order of 800 *min*. The low value was chosen for two reasons. Firstly, the total time of the measurement become significantly high. Secondly, the low value of T_p will result in the earlier saturation of the speedup graphs. The results shown in fig 5.5 therefore represent the worst case scenario.

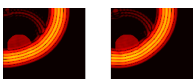
To be able to obtain results in a timely manner, all measurements submitted $N_{jobs} = 3 \cdot N_{hosts}$ jobs. The result was extrapolated to $N_{jobs} = 256$. This is why the statistical error of small cluster sizes is so big. The VMs used were dual-core machines with 2 GB RAM.

Amdahl’s law

Amdahl’s law [Amd67] is well known to calculate the maximal speedup which can be obtained, when knowing the relation between the parallelisable and the serial parts of a program. Unfortunately, the computer architect Amdahl did not take into account timing penalties to be paid when data transfers are involved. To be useful in our context, we have generalised Amdahl’s law to incorporate the time required for output and input data transfer:

$$\frac{1}{\bar{T}} = \frac{1}{T_s + \frac{T_p}{N_{hosts}} + T_d + T_{pain}} \quad (5.1)$$

with T the total runtime of the program, T_s and T_p the time required for the sequential and the parallel part, N_{hosts} the amount of resources to which tasks can be sent, $T_d = D_r \cdot N_{data}$ the time required for transfer of input and output data,



computed, using the data rate D_r . T_{pain} is the time overhead introduced by the parallel infrastructure.

5.5.3 Result

For obtaining T_{pain} an assumption for T_s had to be made, because only the sum of both can be discovered. For a worst case estimate of T_{pain} T_s was set to 1 ms. T_p is known to be 60 s.

T_{pain} and D_r were found by a 2D least square fit to the measurements. I found an average data rate of $D_r = 18 \text{ MB/s}$ and $T_{pain} = 70 \text{ ms}$. These values are used in equation 5.1 to plot the dotted lines together with the measurements in figure 5.5. It can be seen that the general trend of both can be modelled. The simulated data is outside the error bars. This means, that additional to the statistic error, a systematic error is present. This can be understood, because the cluster on which the measurements took place was used by other users at the time of measurement.

For the measured case the 192 tasks ran 160 times faster than sequential execution, which corresponds to 83 % efficiency. The short time of 60 s represents the worst case scenario.

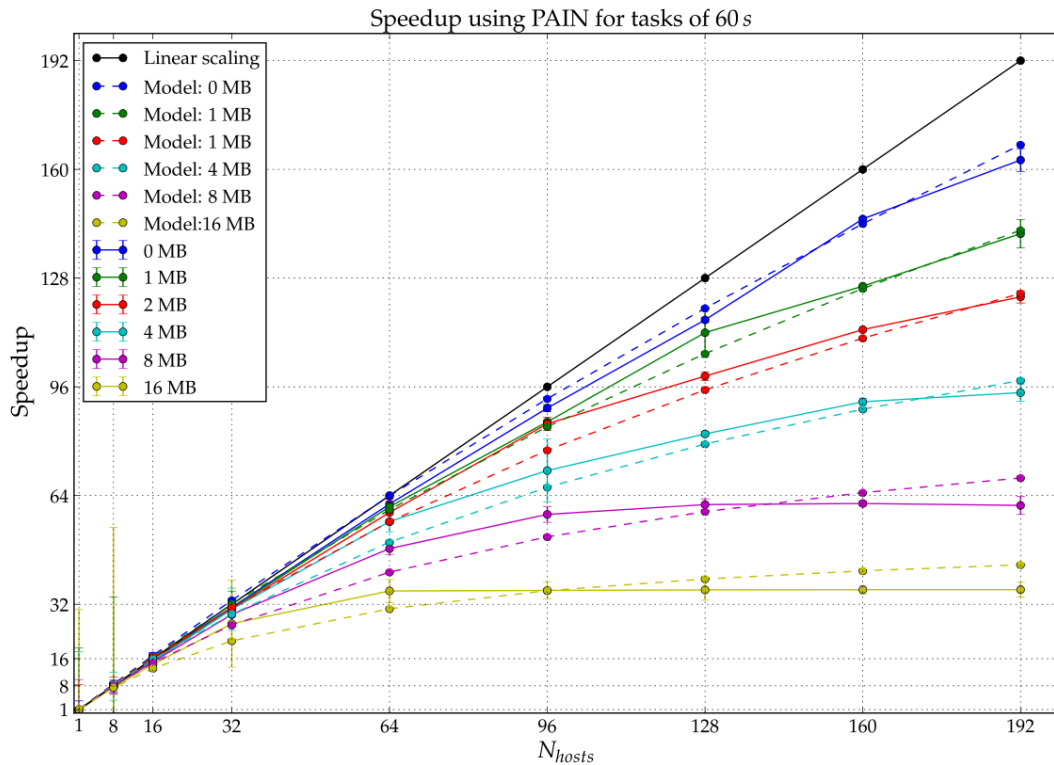
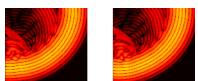
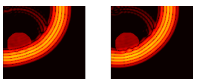


Figure 5.5: Speedup over cluster size and data transfer The amount of input and output data is equal. The 0 MB measurement contains upload of the executable and the minimal amount of data to steer the remote function of 1380 bytes. The fitted curves use the generalised Amdahl's law with $T_s = 1 \text{ ms}$, $T_p = 60 \text{ s}$ a data rate of 18 MB/s and $T_{pain} = 70 \text{ ms}$





RESULTS

In this chapter the derived theory for the propagation of sound waves in matter (chapter 3), the required numerical methods (chapter 4) and the computing infrastructure (chapter 5) will be utilised to answer the scientific questions of this thesis and to discuss the obtained results.

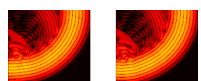
Section 6.1 will verify the implemented numeric solution against an analytical solution derived in section 3.3. Henceforth, the analytical solution will be used in section 6.2 to evaluate which influence the constant density approximation has on the specific example of cancer embedded in fat.

Once the validity of the numeric solution is proved, it will be used in section 6.3 to analyse the impact of the Born approximation on the forward simulation of signals and also on the inverse problem of reconstructing images from data. The simulation will be adapted to account for the spatial dimensions of real existing ultrasound transducers in section 6.4.

Finally, we will investigate the possibility of visualising microcalcifications with the USCT system in section 6.5.

6.1 Verification of the numerical solution

The central question to be answered is whether the numerical algorithm developed in chapter 3 provides correct results. The results generated are therefore compared with those of the analytical calculation also derived in chapter 3. While it is possible that the same error is made in both algorithms, the likelihood is low for two reasons. Firstly, both algorithms are based on completely different derivations in different coordinate systems. Therefore it is unlikely, that both contain errors which result in the same erroneous result. Secondly, the analytical algorithm was developed by a different person, thereby reducing the probability of systematic errors to be introduced in both algorithms. if in doubt – the analytical solution was the correct one. Access to the analytical algorithm is provided with friendly permission of Koen van Dongen at the TU Delft, who implemented the theory and developed the algorithm as part of his research work.



To obtain comparable results, identical input into both algorithms is required. Due to the approximation that scattering on density changes can be neglected under which the numerical solution is currently implemented, constant density is used to model the object function $\chi_c(\mathbf{x})$. In that case the approximation cannot lead to different results in the comparison of both algorithms.

The geometrical setup chosen is a plane wave scattered by a sphere in the centre of the coordinate system. This scenario is shown in figure 3.2. It is one of the few setups that can be solved analytically. All dimensions are chosen to be in the order of the size of the USCT setup. The sphere has a diameter of 8 cm and has the speed of sound of fat but the density of the background medium. It is immersed in water at body temperature. The corresponding tissue parameters are shown in table 6.3 on page 81.

The sphere used in cartesian coordinates was therefore created in spherical coordinates and interpolated to cartesian coordinates. Experiments with an equivalent sphere in cartesian coordinates showed that a systematic error between both definitions exists. This was due to the discrete definition of the sphere. Rounding errors led to a different value for single voxels on the boundary of the sphere, resulting in a slightly smaller sphere.

The simulated regions of interest (ROI) were different in both cases. While the analytical solution is exact and every point can be computed independently, this is not possible in the numerical case because the results from neighbouring voxels influence each other. The consequence is that the analytical solution needs to be computed only for those points that need to be analysed. This is the $x-z$ -plane at $y=0$ corresponding to the $\vartheta-r$ -plane in spherical coordinates (see figure 3.3 for both coordinate systems). The numerical solution had to be computed for all voxels (in all three dimensions). From its result the $x-z$ plane was extracted. The ROI for the analytical case had a diameter of 20 cm. The radial unit vector \mathbf{r} was discretised into 64 steps of 1.6 mm and ϑ into 203 angular steps of 1.8° .

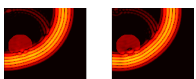
The cartesian ROI for the numerical solution was created so that the voxels along the z -axis are identical in both coordinate systems. This required 128 discrete steps of 1.6 mm along all three axes.

Temporal sampling was done for 128 time steps at a sample rate of 285.7 kHz for a period of 0.45 ms. The pulse used for p^{inc} had a centre frequency of $f_{centre} = 71.4$ kHz as shown in figure 6.1. To avoid aliasing effects, a short bandwidth of 16% of f_{max} was chosen for the pulse so that the highest frequencies, at which aliasing could occur, are attenuated with -150 dB in comparison to the centre frequency.

In the iterative scheme the normalised residual error F_n (see equation 4.16) is the value which indicates the difference between the exact solution and the current iteration. F_{min} is the value below iterations are stopped and the current solution is considered final. This value was set to $F_{min} = 10^{-3}$.

6.1.1 Result of the comparison

The results of the Fourier domain are compared for the centre frequency f_{centre} in (figure 6.2). The time domain results (figure 6.3) are compared at several time steps. It can be seen that in the Fourier domain the error stays below 3% and in the time domain below 1%. While these values indicate the errors at the given snapshots, a



Domain	Norm	(relative units)	Average percental change
Fourier	Numeric	575210	–
Fourier	Analytic (spherical)	574978	0.04 %
Time	Numeric	69549	–
Time	Analytic (spherical)	69514	0.06 %

Table 6.1: Differences of energy norms for Fourier- and time-domain for interpolated and non-interpolated total field in relative units. The time domain energy norm covers all time steps, while the Fourier domain energy norm only used the centre frequency.

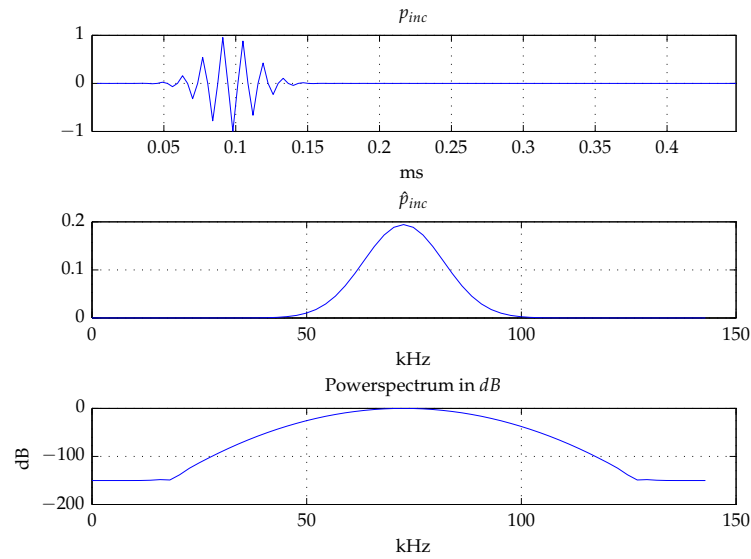


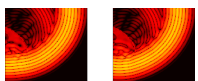
Figure 6.1: Incident pulse used for the verification of the numerical solution. Centre frequency: $f_{centre} = 71.4 \text{ kHz}$, sample rate: $f_{sample} = 285.7 \text{ kHz}$.

more quantitative metric for the errors is given by comparing the energy norms in Fourier domain (FD) and time domain (TD)

$$E_{norm}^{FD} = \sum_{\mathbf{x}} |\hat{p}^{tot}(\mathbf{x}, f_{centre})|^2 \quad E_{norm}^{TD} = \sum_{\mathbf{x}} \sum_t |p^{tot}(\mathbf{x}, t)|^2 \quad (6.1)$$

The sum is taken over all spatial coordinates to compare the Fourier-domain results and over all spatial and temporal coordinates to compare time domain results.

Since the analytical result was interpolated to cartesian coordinates it is possible to compute its energy norm before and after interpolation, thereby obtaining a value for the quality of the interpolation. Table 6.1 shows the results. The differences between both solutions are well below 0.1 %. It can therefore be deduced that that the chosen normalised residual of $F_{min} = 10^{-3}$ is sufficient for the verification of the numerical solution and that a higher precision would not improve accuracy.



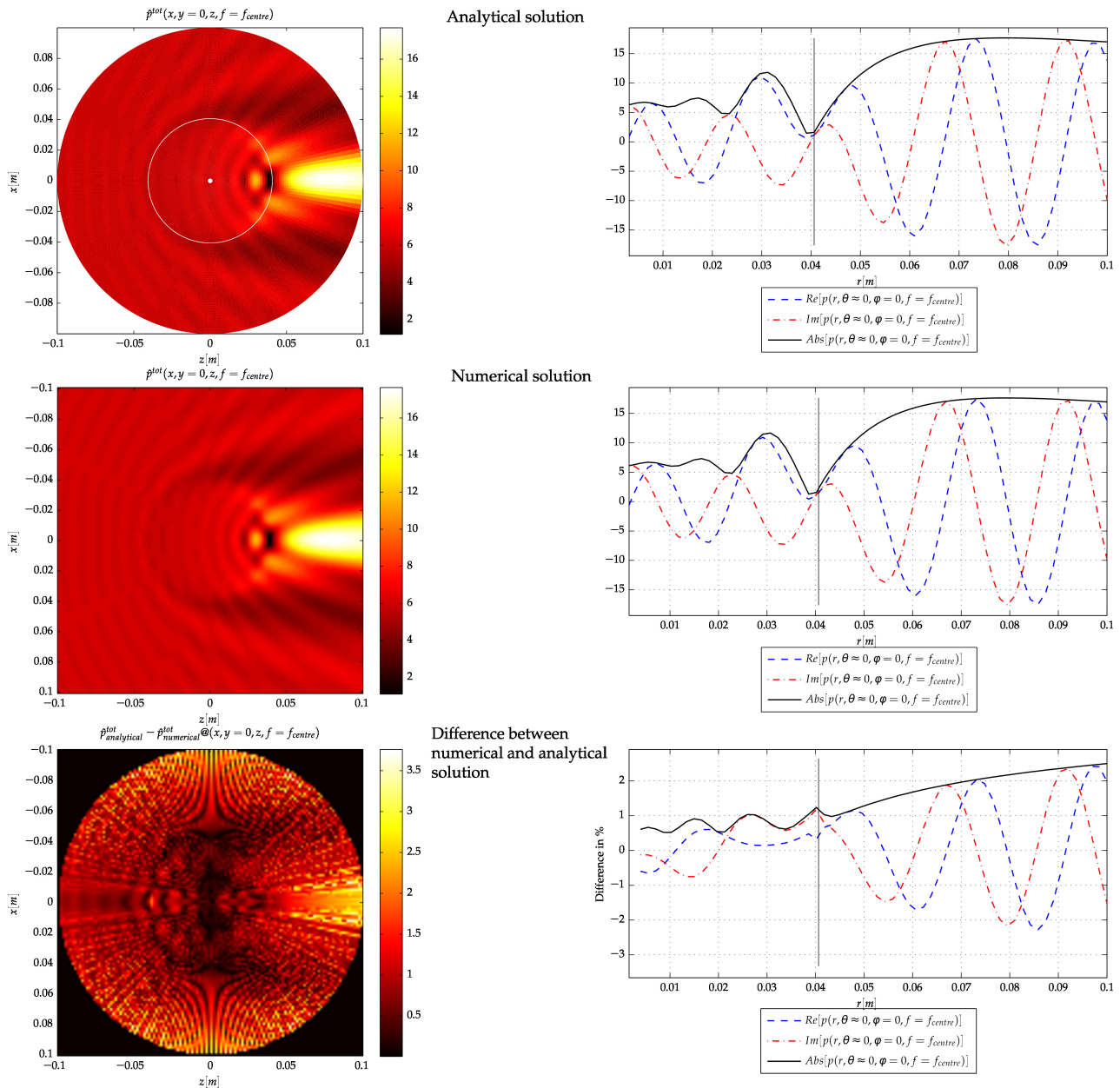
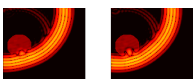
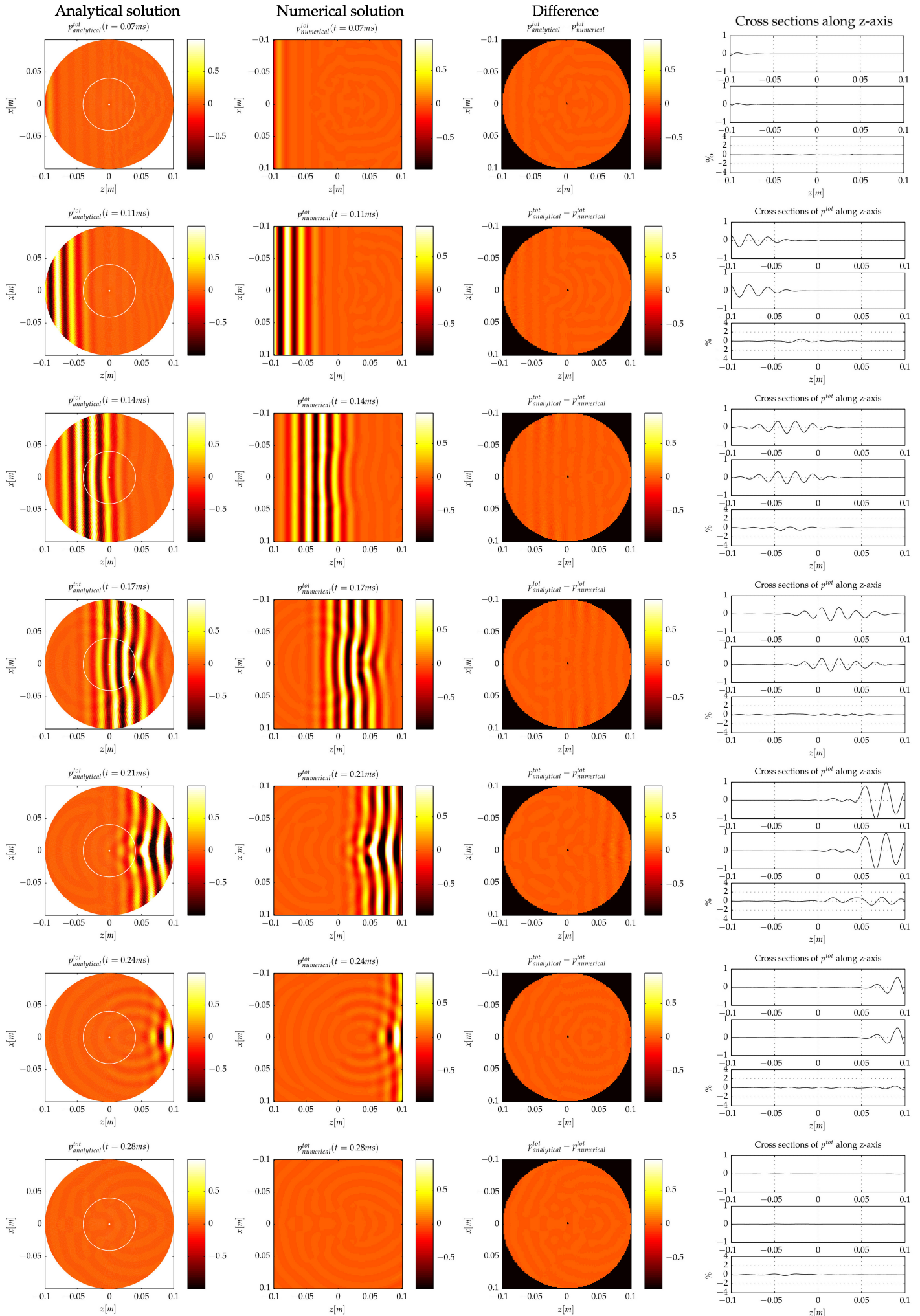


Figure 6.2: Total pressure fields in temporal Fourier domain (see appendix B.2) at $f = f_{\text{centre}}$. The top part shows the analytical solution in spherical coordinates, in the middle is the numerical solution in cartesian coordinates, the bottom part show their difference in percent. The cross sections on the right show the positive z -axis, starting in the centre of the sphere. The cross sections show real, imaginary and absolute values of the total pressure field. All of them are continuous across the boundary of the sphere.

Figure 6.3: (Next page) Total pressure fields are shown for several time steps (one per row). The columns show from left to right the total pressure field of the analytical solution, the total field of the numerical solution, and their difference. The plot on the right part shows the cross-sections along the z -axis (from top-to bottom: analytical, numerical, percental difference). The sphere at which scattering takes place is highlighted in the leftmost plots.





It should be noted that the percentages shown in the third line of figure 6.2 and in the fourth column of figure 6.3 compare both solutions in cartesian coordinates. For this the results of the analytical solution had to be transformed into cartesian coordinates, using the Matlab function “griddata” with linear interpolation.

The observed maximal differences of up to 2.5 % in the plots on the right side of figure 6.3 originate from the fact that these values are shown for the focal plane of the sphere. In this plane errors are larger than at more peripheral positions, which are included in the energy norm.

The numerical solution with residual $F_{min} = 10^{-3}$ can be used to simulate waves travelling through matter within the given approximations. The average error made in this case stays below 0.1 %, the maximal error remains below 2.5 %.

We have therefore shown that the numerical Ansatz is capable of precise simulation for the scattering of pressure waves in matter at spatial variations in speed of sound $c(\mathbf{x})$.

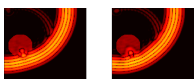
6.2 Impact of neglecting density changes

The numerical solution in equation 3.21 neglects changes in the density. In this section the analytical solution is used to evaluate the error made by using this approximation. The reason to investigate this effect is that cancer has a higher density than the surrounding tissues. Hence, we will use a sphere of cancer immersed in fat in the otherwise unmodified setup of the previous section.

The analysis of the effect of neglected density changes is investigated with respect to the USCT device, which operates at a centre frequency of 2.5 MHz, while the simulations in this section are conducted with a centre frequency of 71.43 kHz. The cancer model is therefore scaled to have the same size in wavelengths. The used size of the sphere has a diameter of 10 cm. This corresponds to a diameter of 2.9 mm when insonified with 2.5 MHz. This is the size at which cancer already may be found with USCT. Another reason to choose such a small inclusion is that we want to see if the neglect of density changes can be neglected. If it is found to be too large for this size, it cannot be neglected at all. The wave pulse used is shown in figure 6.1.

To analyse the effect of the approximation we compare three cases. The reference case is the sphere with changes in density ρ and in compressibility κ . This is compared with two spheres in which density is kept constant. Since density, speed of sound and compressibility depend on each other as described by equation 3.10, we can either keep c unchanged and account for the constant density by modifying κ or the other way around. Both cases will be compared to the original sphere, table 6.2 shows the values used.

The wave field scattered on the unmodified cancer sphere is shown in figure 6.4 at several time steps. The cases of the simulations with constant ρ and unmodified c and with constant ρ and unmodified κ are shown in figure 6.6 (b) and (c) as temporal integral of the spatial energy distribution and are compared with the reference case 6.6 (a). Figure 6.6 (d) shows the comparison for unmodified c and figure 6.6 (e) those for unmodified κ . The colour maps are set to scale to the maximum value outside the sphere, because the region inside the sphere is not relevant for imaging.



Case	Tissue	$c[\frac{m}{s}]$	$\rho[\frac{kg}{m^3}]$	$\kappa[10^{-10}Pa^{-1}]$
Cancer-like sphere Figure 6.6(a)	Fat	1470	950	4.87
	Cancer	1580	1100	3.64
Constant ρ unmodified c Figure 6.6(b)	Fat	1470	950	4.87
	Cancer	1580	950	4.21
Constant ρ unmodified κ Figure 6.6(c)	Fat	1470	950	4.87
	Cancer	1700	950	3.64

Table 6.2: Tissue parameters

The changes in energy outside the sphere with unmodified speed of sound remain below 10%, while those with unmodified compressibility reach 50%.

6.2.1 Impact on the travel time

Imaging methods which use the arrival times of pulses such as SAFT rely on correct timing values. If the constant density approximation affects the speed of sound, the attainable resolution decreases. Figure 6.5 shows the A-scans recorded on the leftmost (upper half) and rightmost (lower half) points of the z -axis in the previous images. These points correspond to unscattered and scattered fields. This is not exactly the case, because the backscattered signals can be seen in the upper plot starting at 0.17 ms .

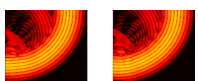
It can also be seen in the upper plot that the incoming pulses are identical in all three cases, while the scattered pulse in the lower plot for unmodified κ arrives $40\ \mu\text{s}$ earlier. This is because in that case the speed of sound was increased to cope for the unmodified compressibility (see table 6.2).

The pulse of the unmodified speed of sound is difficult to distinguish from the exact simulation, it is therefore evident that no reasonable error in arrival time occurs in that case.

6.2.2 Result

When simulating objects using a simulation in which changes of the density cannot be modelled, two options exist. Either the speed of sound c or the compressibility κ have to be modified to reflect the omitted density change. The least errors are made when using a known value for speed of sound, thereby accepting a modification of the effective value of the compressibility. In case of the simulated example with values shown in table 6.2 the error of the spatial error distribution of the energy remains below 10%, while in the other case this error reaches 50%.

The same choice is best when the arrival time of the pulses is concerned. In the case of unmodified c the arrival time does not change, while in the case of unmodified κ an error of $40\ \mu\text{s}$ is observed in the discussed example.



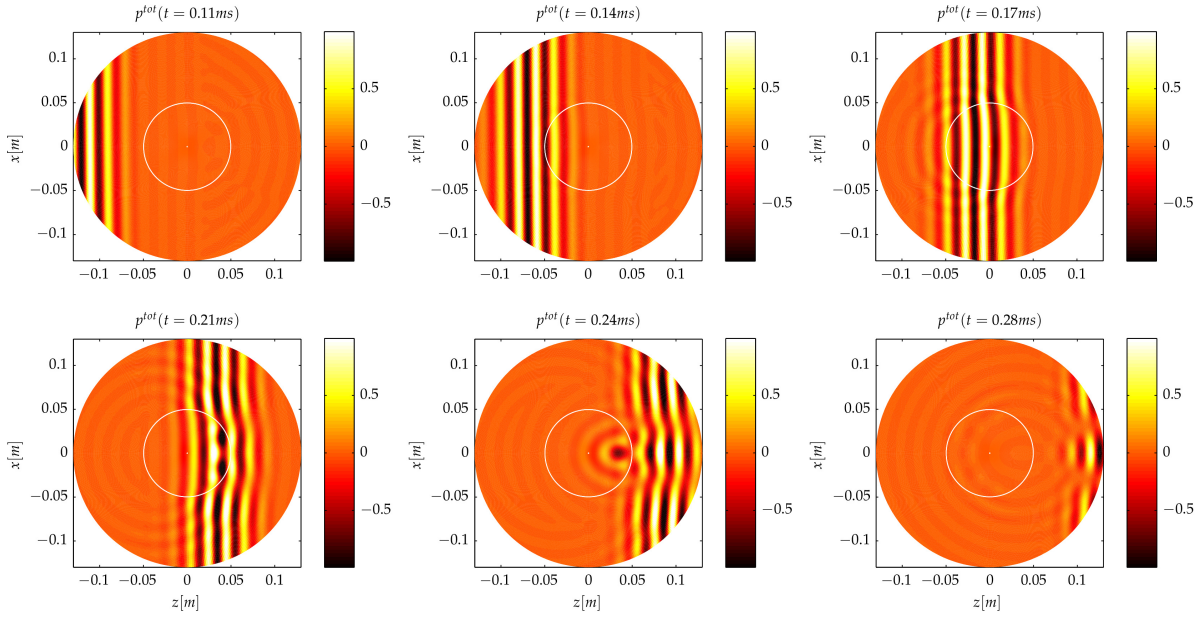


Figure 6.4: Analytical solution of scattering on a spherical cancer region with a diameter of 10 cm embedded in fat. The images show the pressure wave pulse at different points in time. The sphere is marked in white.

With respect to imaging data simulated under this approximation using SAFT, the energy errors might alter the amplitudes of A-scans, and thereby lead to wrong weight of A-scans by up to 10 %.

Much more relevant is the impact of the arrival time. The effect of a wave pulse which arrives $4 \mu\text{s}$ too early may not appear to be large at the frequency for which this effect was visualised in figure 6.5. The real USCT operates at 35 times shorter wave and pulse lengths. In the case $4 \mu\text{s}$ correspond to twice the pulse length. The error thereby introduced in USCT imaging is illustrated in figure 2.5.

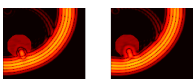
These errors can be avoided when measures speed of sound values are used, not measured values of the compressibility.

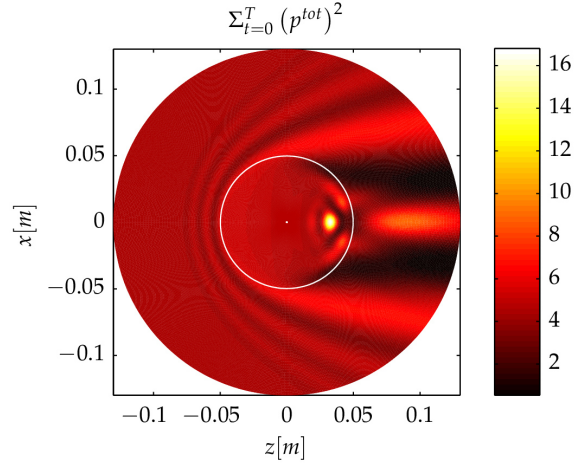
Finally it should be noted, that a sphere does not model all aspects of cancer because cancer does not have a smooth surface. It is more spotty. Nonetheless, when using the analytical solution this is the only model that can be used.

Figure 6.6: (Next page) Comparison of the time integral of the spatial total energy equivalent for scattering on the spherical cancer region with a diameter of 10 cm embedded in fat. The sphere is marked in white. All three cases. (a) shows the reference case, (b) and (c) the possible alternatives, (d) and (e) show the percental differences to the reference case.

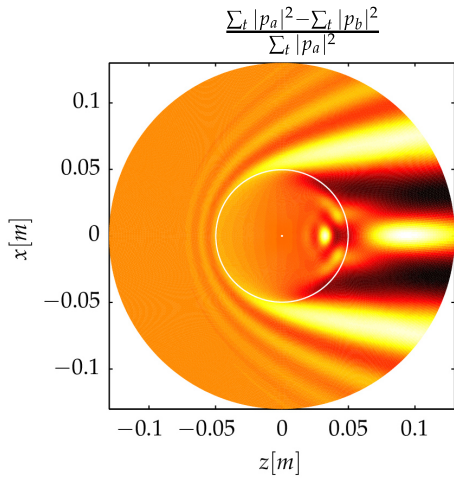
6.3 Impact of the Born approximation

The implemented numerical algorithm allows the simulation of scattered wave pulses. As proven in section 6.1 this simulation is exact as long as variations in the

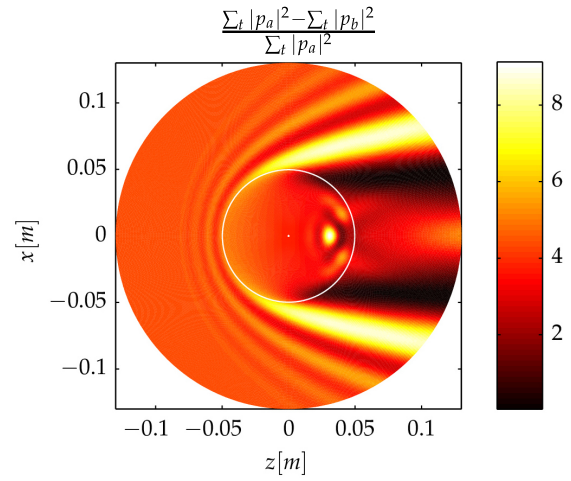




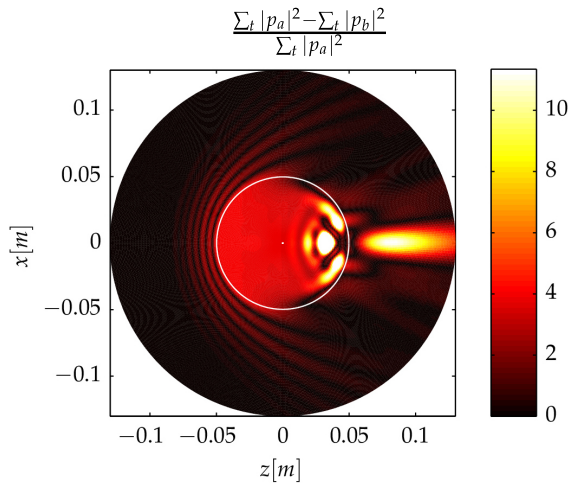
(a) Time integral of the energy equivalent for the fully modelled spherical cancer region



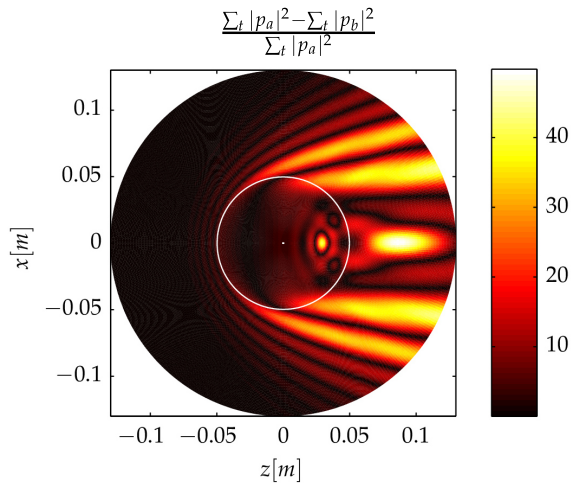
(b) Energy norm for unmodified speed of sound.



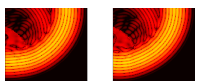
(c) Energy norm for unmodified compressibility.



(d) Percentual difference of energy norm between cancerous sphere and sphere with unmodified speed of sound. The colour bar is scaled to the maximal values outside the sphere, because errors inside the sphere have no impact on imaging by signals measured outside the breast.



(e) Percentual difference of energy norm between cancerous sphere and sphere with unmodified compressibility.



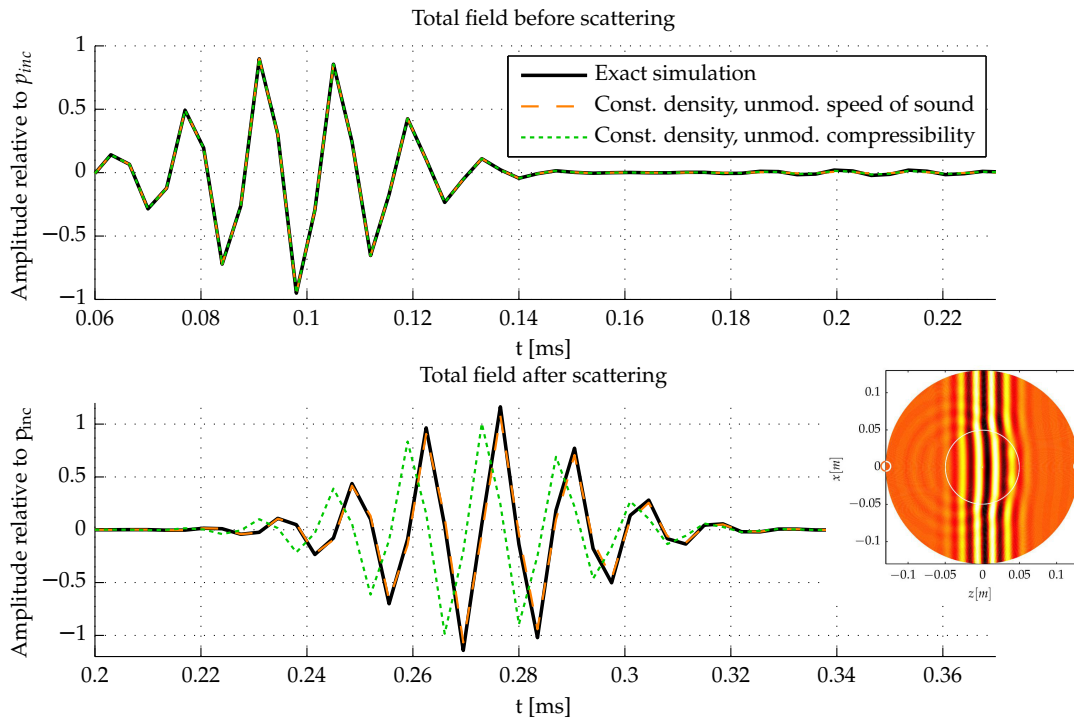


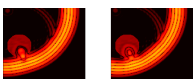
Figure 6.5: A-scans before and after scattering i.e. at the left and right end of the circular region of the previous figures (marked by the rings in the embedded image).

speed of sound are of interest. It is possible to apply the Born approximation (see equation 3.23) to the numerical solution and to create approximated results which provides the opportunity to analyse the effects caused by this approximation. This is of special interest, because the reconstruction algorithm SAFT (Synthetic Aperture Focussing Technique) works within this approximation, too.

The goal of this section is to analyse the differences between the Born approximation and the exact solution. This will be evaluated with respect to the reconstruction of images. We will also analyse, how well shortcomings in the reconstruction (which are due to the Born approximation) can be overcome when introducing speed of sound corrections.

The SAFT algorithm is introduced in section 2.2.1. It is used for reconstructing images from data measured in the USCT and uses reflections of ultrasound waves to depict sources of scattering. This is equivalent to applying the Born approximation within the reconstruction, because multiple reflections and variations in the speed of sound cannot be correctly handled and cause artefacts in the resulting images. Extensions beyond this simplification exist by taking the spatial distribution of speed of sound variations into account (see section 2.2.2).

In this section the impact of the Born approximation is analysed with three approaches. This includes an analysis of the simulated data regarding travel time and energy but also the reconstructed images of signals simulated exactly and within the Born approximation.



For these analyses so called A-scans and B-scans will be used. The term a-scan refers to a plot of the pressure amplitude measured at a receiver over time. B-scans are composed of several A-scans plotted next to each other along their time axis for many receivers using colour values which correspond to amplitude. Most of the B-scans in this thesis show the normalised energy equivalent on the dB scale instead of the pressure amplitude. The energy of a wave is proportional to the square of the pressure amplitude (see appendix B.3). The energy decreases with $\frac{1}{r^2}$. To remove this dependency the normalised energy equivalent is introduced as

$$N = p^2(\mathbf{x}, t) , \quad (6.2)$$

with the pressure p and the distance between emitter and receiver r . In this normalisation the B-scans show the same value, regardless of how far away a receiver is from an emitter. The dB scale is used because of better visualisation of the wide range of relevant energies on the logarithmic scale.

The object with which these effects will be investigated is the same sphere as in section 6.1: a sphere of fat immersed in water at body temperature. The model uses changes in speed of sound while density changes are not modelled. A constant density of $1000 \frac{\text{kg}}{\text{m}^3}$ is used. The sphere is placed centrally into the USCT (see figure 6.7) which requires a ROI with an edge length of 26 cm in all three dimensions. This is slightly larger than the previously used ROI. The ROI is discretised into 128 steps of $dr = 2.1 \text{ mm}$. Temporal sampling uses 128 frequencies for the pulse shown in figure 6.8. The larger bandwidth of 39% of f_{max} was used to obtain shorter pulses and better images. The maximal frequency is now suppressed with -40 dB , which equal to two orders of magnitude.

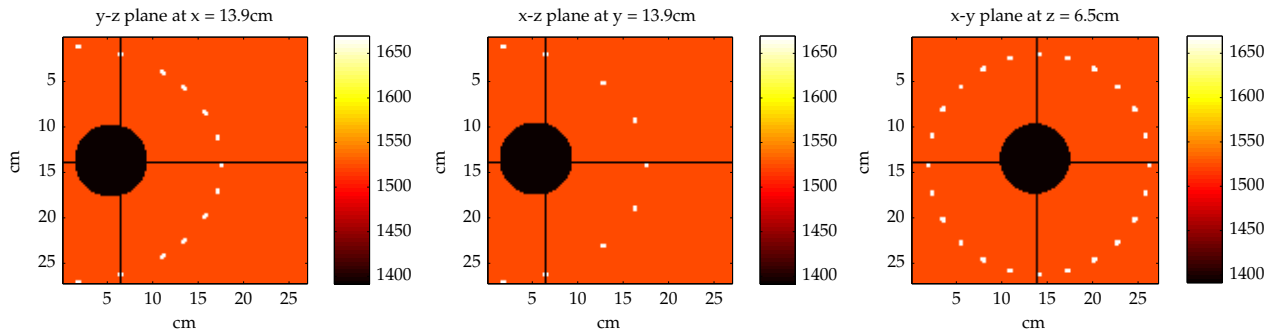
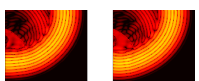


Figure 6.7: Centrally placed sphere of fat in water. The white points represent emitter positions of the USCT.

An initial comparison between a full simulation and one within the Born approximated simulation is shown in figure 6.9. It shows the normalised energy equivalent in a B-scan using the first emitter and all receivers of the USCT.

Several points can be observed in this B-scans. Firstly the ring-like geometry of the USCT can be seen. For reference a technical drawing of the USCT is provided in figure A.18 in appendix A.5. Differences between both simulations can be observed in the last graph. Large visible differences could originate from shift of only half a wavelength between two waves, while the larger error of a full wavelength in runtime could hardly be noticed. Therefore figure 6.9 is merely useful to indicate that further analysis is necessary.



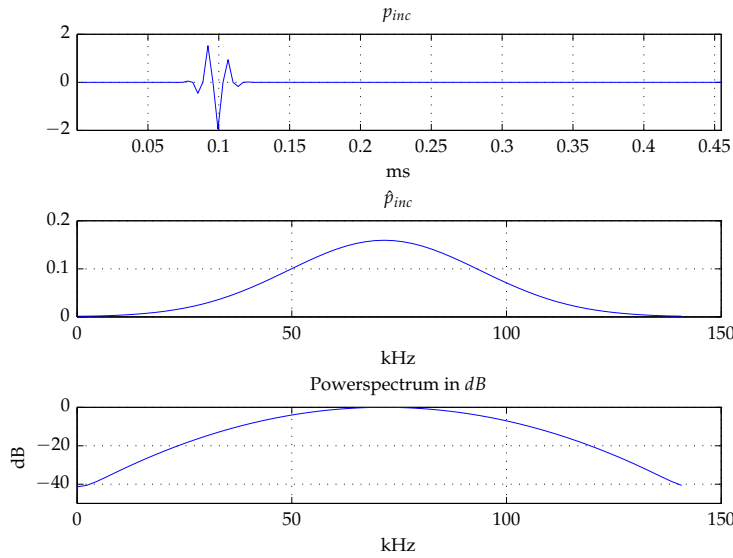


Figure 6.8: The incident field used for the subsequent simulations. It has a centre frequency of 70.4 kHz a bandwidth of 39%. The highest frequency (140.8 kHz) is sampled with five spatial points per wavelength and is -40 dB lower than f_{centre} . The sample rate is $f_{sample} = 281.6$ kHz.

The strongest signal with -60 dB originates from the transmission of the emitted pulse in a straight line from emitter to receiver. This transmission signal is of similar amplitude and phase in both simulations, because it can not be observed in the difference image.

In imaging applications this pulse has a special role because of its magnitude. figure 6.9 shows that this pulse is at least 60 dB larger than other signals. If the transmission signal is used the imaging procedure is called transmission tomography if it is removed it is reflection tomography, such as SAFT.

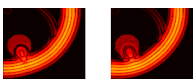
6.3.1 Travel time analysis

A better understanding of the differences between both simulations can be obtained when analysing A-scans.

For visualisation a movie of the wave travelling through the volume with the sphere is available in form of a flip-book on the bottom left of all even pages. It shows the energy equivalent of the pressure waves on the dB scale with the sphere superimposed into the images. The left movie shows the exact solution while the right one corresponds to the simulation within the Born approximation.

Figure 6.10 shows four A-scans, one for transmission and one for reflection and both with and without transmission signal. The receiver numbers correspond to those in the technical drawing in figure 6.9.

In figure 6.10 the plots start at $t_0 = 0.1$ ms, because this is the time at which the incident pulse starts. All four plots are scaled differently. The largest signal is the so called transmission signal in the first and third plot. With the speed of sound of $1524 \frac{m}{s}$ it can be verified that the transmission pulse in plot one at 0.27 ms



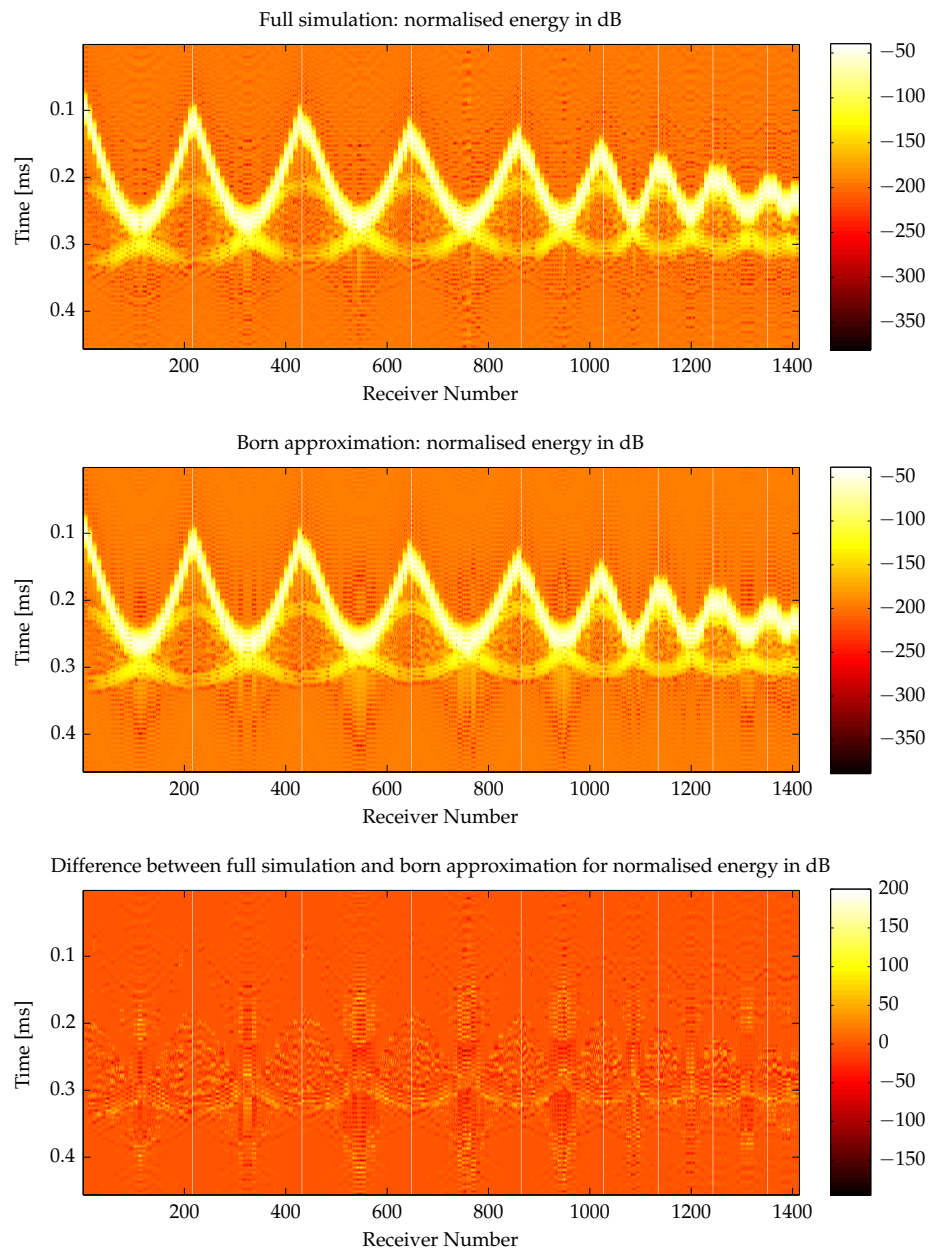
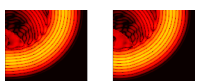


Figure 6.9: B-scans of the first emitter compared. Shown are the normalised energy equivalents in dB. From top to bottom shown are the full simulation, the Born approximated simulation and their difference. To localise the receiver positions, a technical drawing of the USCT is provided in figure A.18. The vertical white lines mark the rings of the USCT.



corresponds to 26 cm, the diameter of USCT. The transmission pulse in plot three at 0.17 ms corresponds to the 10 cm of the straight line from emitter one to receiver 225. The pulse which traversed the sphere shows a difference in time of arrival of 4 μ s which is caused by the Born approximation which does not correctly model speed of sound variations. The reflected pulse in plot three only travelled in water and is therefore identical in both ways of computation.

$$\hat{p}_{Born}^{tot}(\mathbf{x}) = \hat{p}^{inc}(\mathbf{x}) + k_0^2 \int \hat{G}(\mathbf{x} - \mathbf{x}') \chi^c(\mathbf{x}') \hat{p}^{inc}(\mathbf{x}') dV(\mathbf{x}') . \quad (3.23)$$

The transmission pulse of the first A-scan shows a very important property of the Born approximation: The speed of sound is not correctly modelled. The exact pulse arrives later than the Born-approximated one. This is due the lower speed of sound inside the sphere when compared to water. From the structure of equation 3.23 this can be directly understood, because in this equation the convolution of the incident pressure $\hat{p}^{inc}(\mathbf{x})$, the object function $\chi_c(\mathbf{x})$ and the Greens function $\hat{G}(\mathbf{x})$ are added to the incident pressure. To model a time delay, the convolution integral would need to yield a result that when added to $\hat{p}^{inc}(\mathbf{x})$ extincts it and adds the signal at a different point in time. However, the integral in equation 3.23 is a convolution integral, which models point sources at locations of the object function insonified by $\hat{p}^{inc}(\mathbf{x})$. An extinction of $\hat{p}^{inc}(\mathbf{x})$ can therefore not be modelled. The fact that the equation is notated in Fourier domain does not make a difference, because the structure of equation 3.23 does not change under the temporal Fourier transformation.

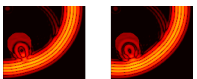
To visualise the scattered signals, the transmission signal is removed. This is also one of the preprocessing steps in SAFT. The plots without the transmission signal for the transmitted wave is shown in plot two. This names correspond to the geometric location of the receiver 109 used in this plot, because to reach the receiver, the wave has to traverse the USCT. We can see one pulse at $t = 0.32$ ms, which corresponds to 33.5 cm. It is caused by a reflection at the bottom of the sphere. The signal starting at 0.36 ms is multiple scattering with an amplitude of 10 % of the primarily scattered pulse. It is not present in the Born approximation. Additionally a phase shift of π is missing in the approximation simulation.

The fourth plot shows the reflected situation without transmission pulse. The first pulse corresponds to the reflection on the front of the shere (18.3 cm) and at the back of the sphere (32 cm).

In reflection tomography such as SAFT the multiple reflections in comparison to the primary reflections are of interest. Multiple scattering can be quantified as 10 % of the pressure and 1 % of the energy of the first order reflection.

The conclusion from the analysis of the A-scans is that three different effects have to be attributed to the Born approximation:

1. Wrong arrival time of pulses which travel through tissue with different speeds of sound.
2. Missing phase shift of π in reflected signals.
3. Multiple scattering can not be simulated at all.



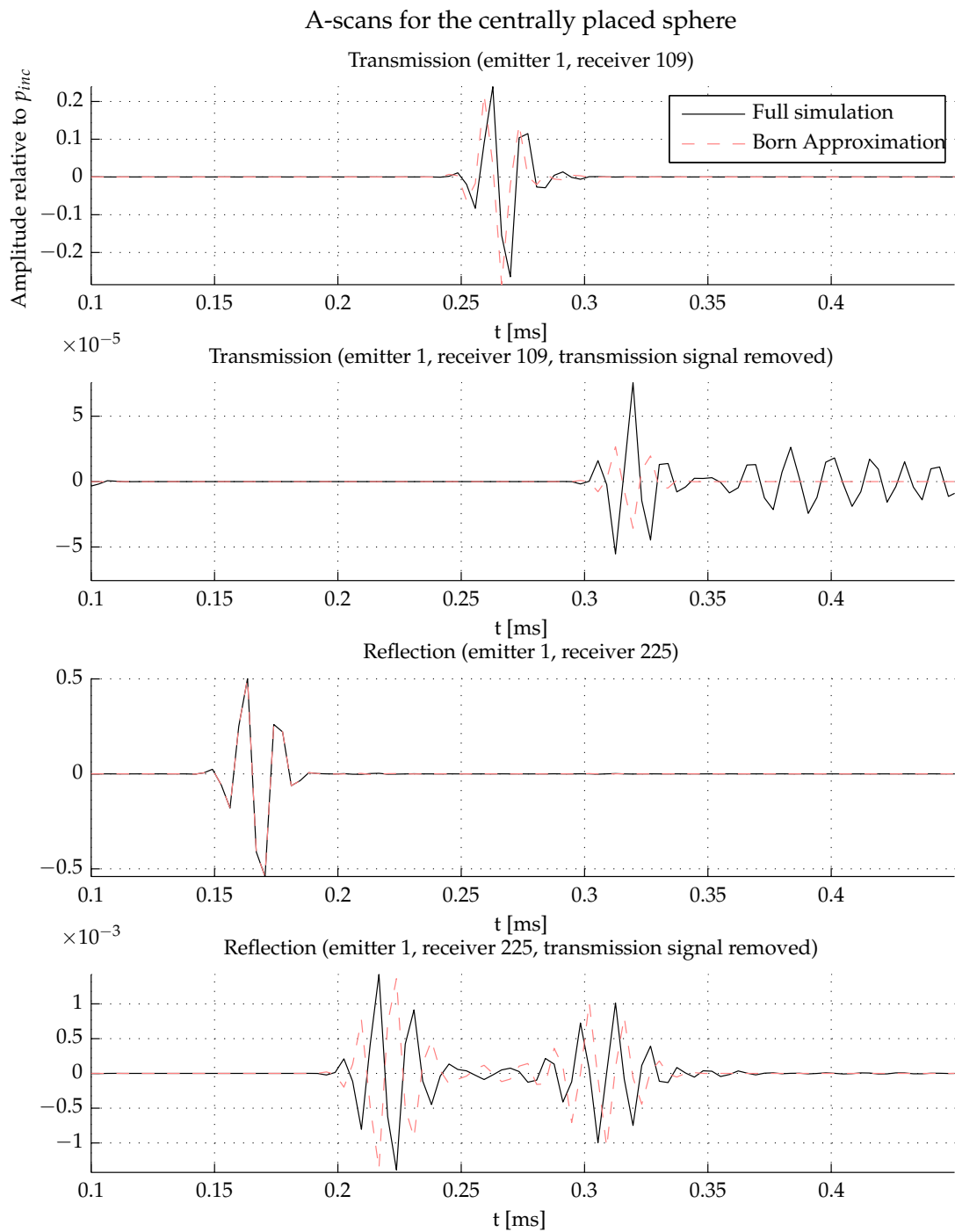
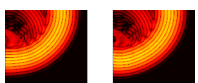


Figure 6.10: Two A-scans of the same emitter but at different receiver positions, each with and without transmission pulse. The plots start at $t = 0.1$ ms because the incident pulse starts at this time. All four plots use different scaling on the y-axis.



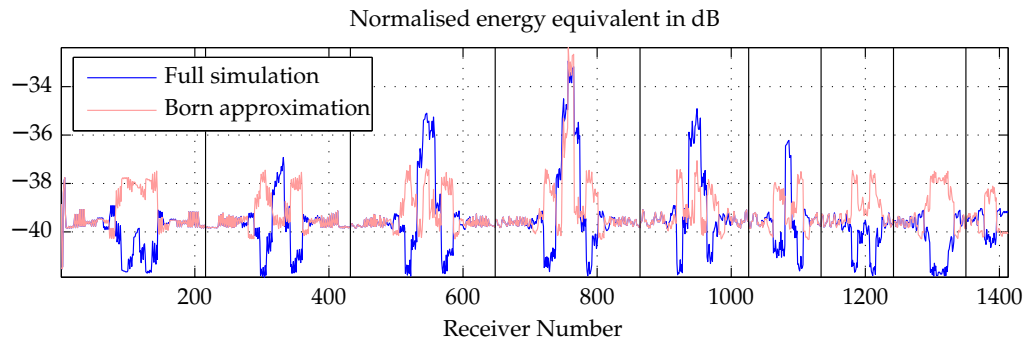


Figure 6.11: Normalised, summarised energy plots for the centrally placed sphere of fat. The vertical black lines mark the rings of the USCT.

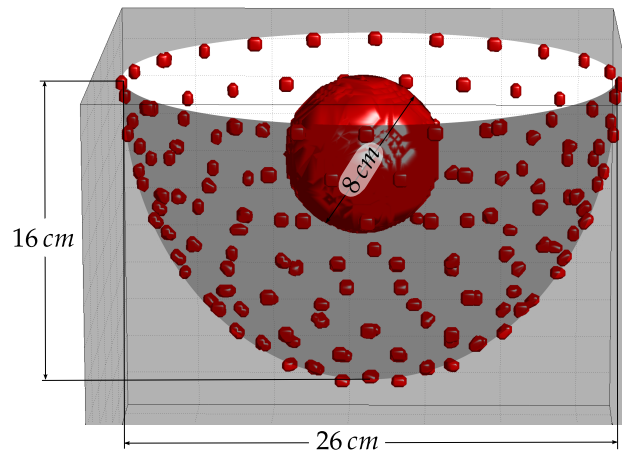


Figure 6.12: Sphere placed inside the USCT

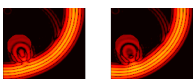
6.3.2 Energy analysis

A different aspect to look at is the amount of energy received by each receiver. To remove the dependence on the distance between emitter and receiver, the normalised energy equivalent is used and integrated over time. This is done for both simulations (full simulation and Born approximation) and plotted on the dB scale in figure 6.11.

It can be seen that the energy is distributed differently within the Born approximation. The total normalised energy (obtained by an additional integration along the receivers) is the same in both cases (less than 0.1 % difference). A characteristic difference can be observed: the energy distribution of the full simulation shows focussing and shadowing caused by the sphere while the energy distribution of the approximated simulation shows a diffraction pattern similar the scattering at a sphere with $ka > 1$. That means this effect can not be treated within the Born approximation. A comparison of a plane wave scattered on a sphere

The energy collected by all receivers is equal in both simulations when the wave reaches the receiver without traversing the sphere.

B-scans and energy plots of additional objects are shown in Appendix A.1.



6.3.3 Impact on imaging

The difference between the full simulation and the simulation within the Born approximation is relevant for imaging. Especially for imaging using the SAFT algorithm, because it is based on the Born approximation.

Up to now we have analysed the impact of the Born approximation in the simulation. From this we deduced possible problems in imaging. However, for image reconstruction, the inverse situation is found: The Born approximation is made in imaging while exact data are used as input.

Therefore, in order to quantify the impact of this approximation in imaging, we firstly create a reference case. This is done by feeding data which was simulated under the Born approximation into SAFT which works within the same approximation. Using the same approximation in both, simulation and imaging is called “committing an inverse crime” [Wir04], because it leads to better images than can be generated under realistic circumstances. The thereby created reference case is used to quantify the impact of this approximation.

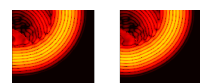
To investigate the differences between the full simulation and the Born approximation in imaging additional objects were simulated, each within the exact simulation and the Born approximation. A full simulation of the full USCT setup consisted of 628 individual simulations – one for each emitter. The workload was distributed for each emitter using PAIN on 94 virtual machines with 726 cores. The computations took roughly one day per object. An exception was the simulation with a doubled centre frequency, doubled number of time steps and larger object. This simulation was conducted to enlarge the object in terms of wavelengths. It required one week to complete.

The simulation is implemented so that its output data can be used as input for the reconstruction algorithm SAFT. It offers several options for signal preprocessing and one for improving imaging beyond the principal limitation of the Born approximation. For reconstructing images the optimal pulse option (see figure 2.4 on page 18) was used. One of the effects of this filter is that it uses the absolute value of the signal. Therefore, the phase shift which is missing within the Born approximation is not noticeable.

Three reasonable cases of reconstructions are compared with each other. The “Born reconstruction” reconstructs data simulated under the Born approximation with the standard set of parameters of SAFT. The “full reconstruction” reconstructs data from the full (exact) simulation with the same set of parameters of SAFT as the “Born reconstruction”. The “enhanced reconstruction” reconstructs data from the full simulation with the speed of sound correction option of SAFT. This option uses the ground truth of the simulation to take the spatial speed of sound distribution correctly into account.

The best images are to be expected for the reconstruction of data simulated under the Born approximation, because in this case the same approximation is made in the simulation and in the reconstruction.

Three cases of objects – all immersed in water at body temperature – are simulated. Their tissue parameters are listed in table 6.3. The first case is a sphere of fat with two inclusions that have the speed of sound of cancer with a diameter of 10 mm. The sphere has the same dimensions as the one used in previous simulations (diameter of 8 cm) but is not placed in the centre of the USCT. The decentral position



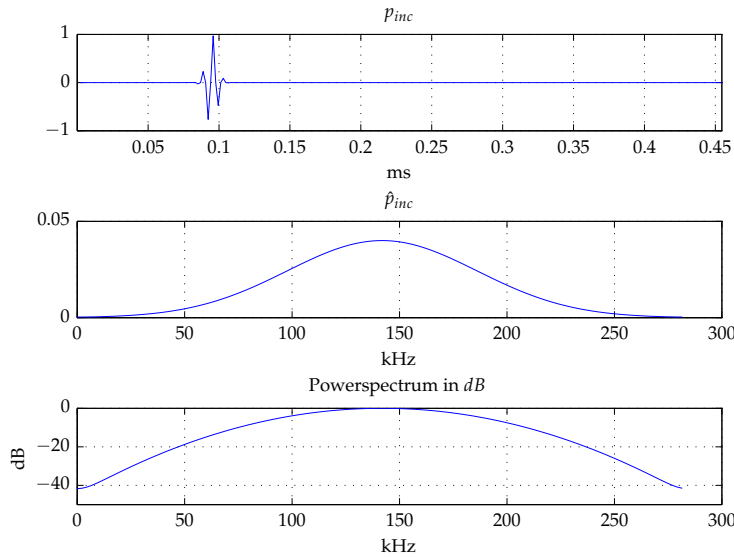


Figure 6.13: High frequency pulse used for the enlarged breast model. The spectrum and the pulse form are the same as for the normal pulse (figure 6.8), because the bandwidth is defined as a percentage of f_{max} . The centre frequency is therefore $f_{centre} = 140.8 \text{ kHz}$ and the sample rate $f_{sample} = 563.1 \text{ kHz}$.

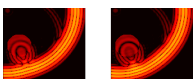
was chosen to ensure rotationally symmetric artefacts do not overlap with the object. This object is called “decentral sphere with inclusions”. It is shown in figure 6.14.

The second object is a breast phantom. It was hand segmented from an MRI volume image. The size of the breast phantom was scaled so that it fills the USCT like a patient with cup size “C” would. The phantom contains a cancerous region of 1.8 cm diameter as well as an inner structure which consists of fat and glandular tissue. This breast model is shown in figure 6.15. The tissue parameters are also listed in table 6.3. Scattering on this object is shown in the flip-book on the bottom right of all odd pages. Left is the exact solution, right the one under the Born approximation.

The third object is the same breast but enlarged to almost fill the USCT. This object was insonified with the double centre frequency. This was done to enlarge the object in terms of wavelengths. While the normal breast phantom has a diameter of 4λ in the $x - y$ -plane at the position of the cancer (which has a diameter of 1λ) the enlarged high-frequency object has a diameter of 15λ at the position of the cancer which measures 3λ in diameter. In comparison to the wavelength of the USCT the normal breast phantom had a diameter of 143λ at the position of the cancer, which had a diameter of 36λ in that case. This object is shown in figure 6.16.

The shift to higher frequencies is done keeping in mind that aliasing artefacts have to be expected, because the shortest wavelength with 5.44 mm is now only sampled with 2.6 points, while 4 are required [VH09].

The results for the decentral sphere with inclusions are shown in figure 6.14. Most clearly can be seen that all reconstruction types do not properly reconstruct that part of the sphere which is close to the emitters. This is because in the reconstruction



Tissue	$c[\frac{m}{s}]$	$\rho[\frac{kg}{m^3}]$	$\kappa[10^{-10}Pa^{-1}]$
Water (37°C)	1524	993	4.34
Tissue	1550	1060	3.93
Fat	1470	950	4.87
Cancer	1580	1100	3.64
Breast phantom fat	1478	–	–
Breast phantom glandular tissue	1485	–	–
Breast phantom skin	1610	–	–
Hydroxyl apatite	6790	–	–

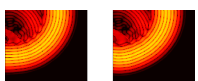
Table 6.3: Material parameters of human tissue [Duc90, HS11]

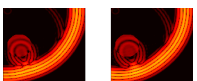
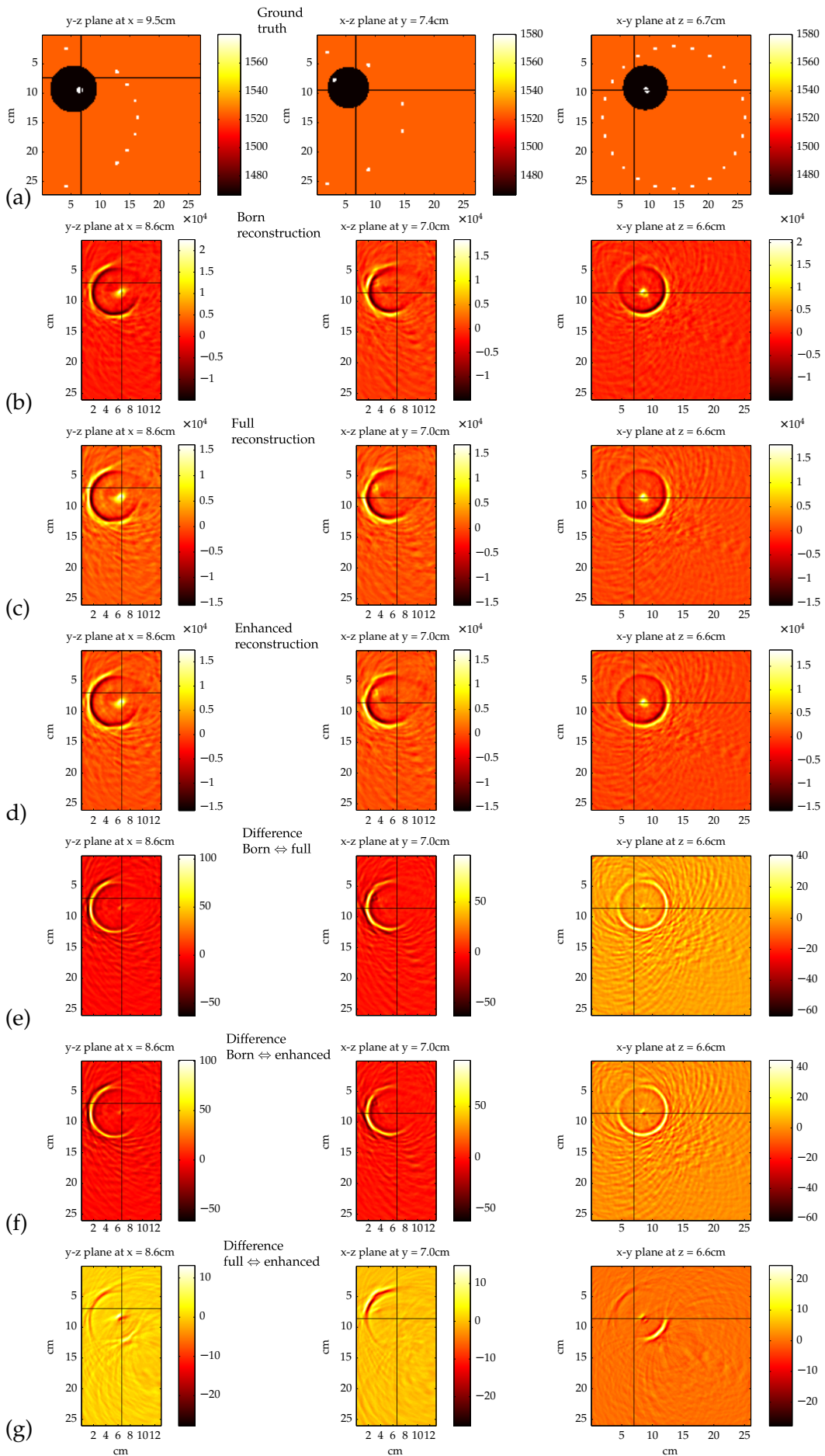
using SAFT each peak in an A-scan corresponds to a possible source of reflection. These reflections create a 3D ellipsoid with emitter and receiver as focal points and the time value of the peak as radius in the result-image. These ellipsoids are added into the image for all emitter-receiver combinations selected in the reconstruction. The part of the sphere which is closer to the transducer array systems (TAS) is under-represented in this form of reconstruction. Therefore it does not appear. This effect is known and visible only for placements of the sphere too close to the USCT wall. The effect is the same in all reconstructions of this object.

Regarding the differences between the reconstructions, it can be seen that the Born-reconstruction (b) yields the best images with respect to two metrics. Firstly the total amplitude of the result image is highest. This means that effects as shown in figure 2.5 on page 20 occur to a lesser extent than in the reconstruction of data from the exact solution. This behaviour is to be expected, because of the Born approximation in simulation and reconstruction. Due to the same effect the surface of the sphere is imaged sharper in the Born reconstruction.

Secondly, it can be seen that the reconstructions of the exact simulation (c) and (d) contain more artefacts in the whole image. These are also visible in the comparison between the Born reconstruction and the reconstruction of the exact simulation (e) and the reconstruction of the exact simulation and speed of sound correction (f), which show the percental differences. There are two possible causes for these artefacts. One is the multiple scattering which is modelled by the exact simulation, but not within the Born approximation. The second are so called “grating lobes” [RDZG10] and originate from the sparse aperture of the USCT [RSZG08a]. Furthermore, the artefacts are better visible due to the lower amplitude of the result image.

Figure 6.14: (Next page) Decentrally placed sphere of fat with two inclusions of cancer embedded in water. The sphere is simulated and reconstructed in several ways. From top to bottom shown are: Ground truth (a), reconstruction of Born simulation (b), reconstruction of full simulation (c), reconstruction of full simulation using the known ground truth in speed of sound correction (d), percental difference between Born and full reconstruction (e), percental difference between Born and speed of sound reconstruction (f), percental difference between full and speed of sound reconstruction (g). The black lines mark the positions at which the other visualised planes are placed.





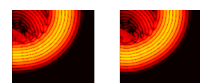
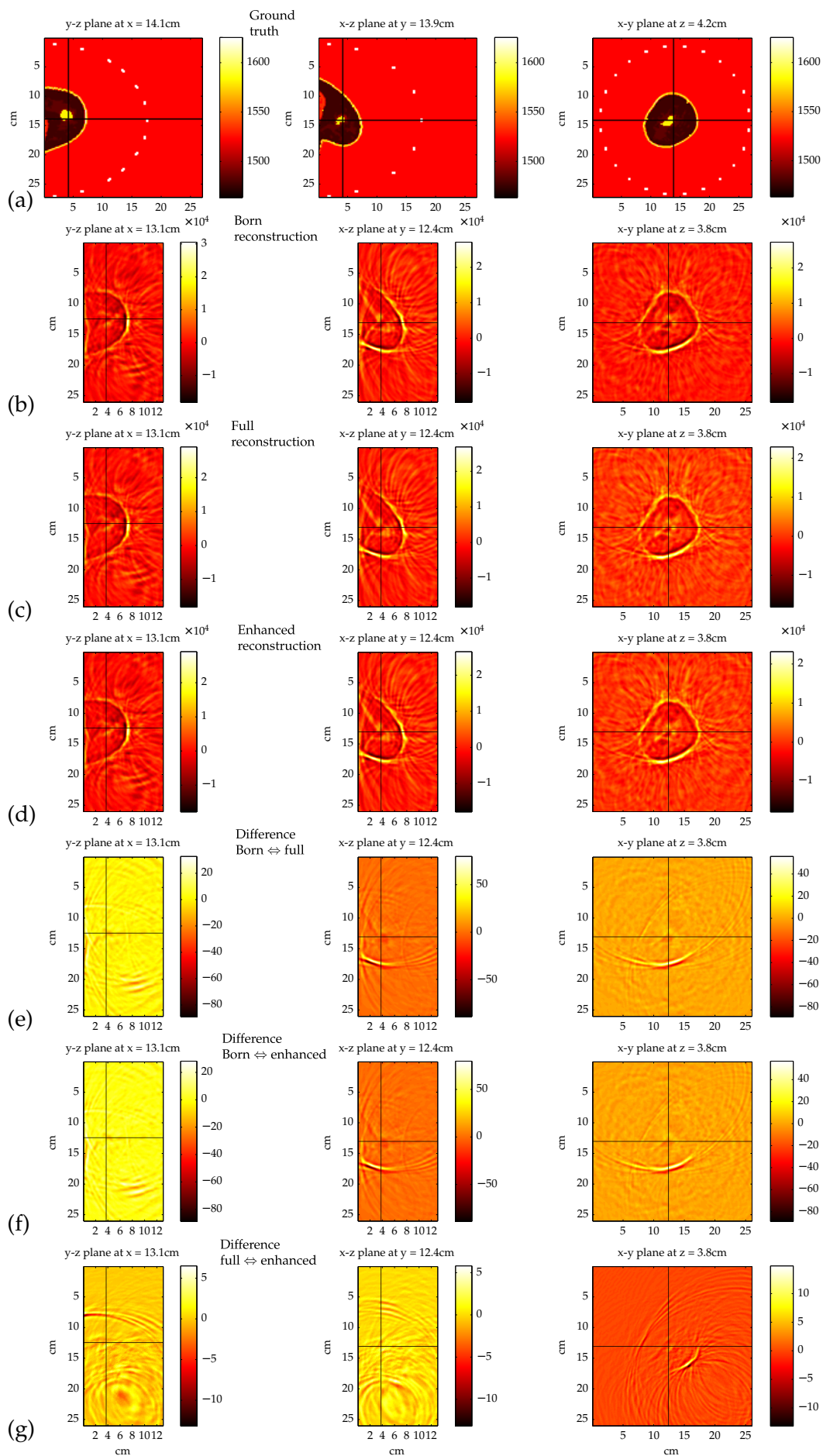


Figure 6.15: (Previous page) Breast model simulated and reconstructed in several ways. From top to bottom shown are: Ground truth (a), reconstruction of Born simulation(b), reconstruction of full simulation(c), reconstruction of full simulation using the known ground truth in speed of sound correction(d), percental difference between Born and full reconstruction(e), percental difference between Born and speed of sound reconstruction(f), percental difference between full and speed of sound reconstruction(g). The black lines mark the positions at which the other visualised planes are placed.

The differences between both reconstructions of exact data and the Born reconstruction are up to 60 % as can be seen in figure 6.14 (e) and (f).

The comparison between the full simulation reconstructed with and without speed of sound correction (g) shows the effect of this correction. The highest values correspond to larger corrections. These are locations at which the sphere was at least partially reconstructed from signals which have traversed the sphere. The parts of the sphere which are close to the USCT centre, for example, were reconstructing using many pulses which first traversed the sphere and were then scattered.

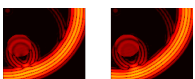
The comparison between difference images (e) and (g) shows that changes introduced by the speed of sound correction are not the same as those introduced by using the exact solution. This means that improvements beyond the speed of sound correction are still required to obtain the reference images (b).

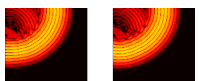
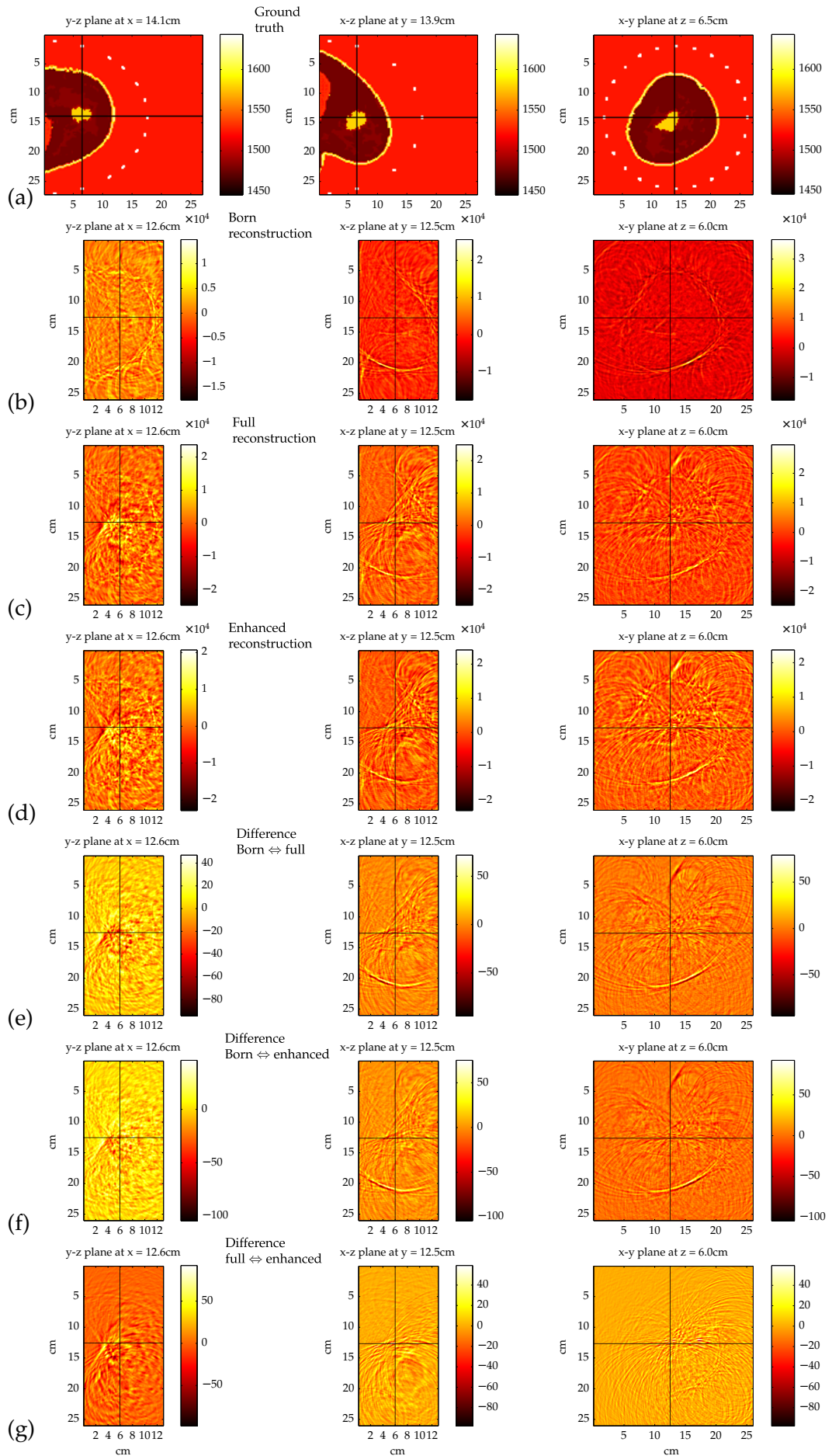
The reconstructions of the breast model are shown in figure 6.15. The breast was immersed into the USCT as foreseen by the medical measurement plan. Therefore the problems as seen in figure 6.14) do not occur. In all three types of reconstruction (b), (c) and (d) the skin of the breast can be seen. Also the cancerous region inside the breast can be seen in all three. This is due to additional sources of scattering inside the breast. Again, the Born approximated reconstruction (b) shows a higher total amplitude and slightly less artefacts than the other two reconstructions (c) and (d). Unlike the decentral sphere, the large amplitude is not as pronounced for the breast phantom.

The differences between the Born reconstruction and the two variants of the full version (e) and (f) are in the same order of up to 80 %. The changes introduced by the speed of sound correction are much smaller (up to 10 %).

More interesting results are visible in the reconstructions of the big breast in figure 6.16. It can be seen by the bad visibility of the skin, that the breast is too close to the USCT wall. This is the same problem of missing illumination as for the decentral sphere. However, this was accepted in order to have access to the simulation larger object (factor three in terms of wavelengths). The cancerous inclusion is hardly visible in the Born reconstruction and cannot be seen in the reconstruction of the full simulation. The speed of sound correction has a larger influence due to the shorter wavelengths and also because of the larger delays due to the increased size of the object. Therefore the cancer can be seen in the enhanced reconstruction. It can also seen in the difference image (g), that the speed of sound correction has a much larger impact than in the previous objects. All difference images yield values between -100% and $+75\%$.

Figure 6.16: (Next page) Enlarged breast model simulated and reconstructed in several ways. From top to bottom: Ground truth(a), reconstruction of Born simulation(b), reconstruction of full simulation(c), reconstruction of full simulation using the known ground truth in speed of sound correction(d), percental difference between Born and full reconstruction(e), percental difference between Born and speed of sound reconstruction(f), percental difference between full and speed of sound reconstruction(g). Black lines mark positions at which the other visualised planes are placed.





6.3.4 Result

The impact of the Born approximation on imaging is two fold. It has an impact on the arrival times of pulses fo $4 \mu s$, which is relevant for higher bandwidth imaging. The amplitude of multiply scattered waves is at 10 % of the first reflection. Both effects can clearly be seen, both in the simulations and in the reconstructions. It is currently possible to do an exact computation with an object dependent factor of 25 to 100 times more computing time compared to the time required to compute a simulation within the Born approximation. However, this is not possible for the reconstruction for two reasons:

1. The theory is not implemented into an algorithm which could be used to generate images from the USCT. (There is an implementation taking place currently in the TU Delft).
2. Even with such an algorithm ready, the computational requirements are too large to reconstruct the full USCT.

The speed of sound correction of SAFT appears to be the best correction currently available to obtain images in a reasonable time. To see the effect of the speed of sound correction in a simulation a finer discretisation is required. As can be deduced from the observation in section 6.2.1 at least 512^3 voxels and 512 frequencies are required for this.

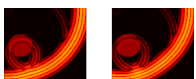
On the computational resources available for this thesis, this would require roughly one year to be computed.

6.4 The effect of surface sources

All simulations used so far were based on the assumption of point sources, i.e. a uniform emission of the pressure fields to all directions. Such emitters represent an ideal case which is not realistic. A point source is idealised by vanishing surface area. This approximation holds if the sidelength of the surface is considerably smaller than the wavelength. To obtain realistic simulations, it should be possible to simulate emitters taking their dimensions correctly into account. Therefore, it will be analysed if surface sources can properly be simulated in this section. The existing simulation will be enhanced to take the spatial dimensions of an emitter as used in the USCT device into account. The field of the surface source is created by integrating the field of point sources along the active surface of the emitter. To obtain precise results the discretisation has to be fine-grained enough. 75 points per wavelength should be used to achieve a tolerable accuracy of less than 1 % [McG04]. The simulations conducted in this thesis uses 100 points per wavelength. For each simulated wavelength the same discrete point sources were used.

The simulations in this theses are conducted at the centre frequency of $70.3 kHz$. To obtain results comparable to the USCT device, the simulated emitter was enlarged to be of the same size – when measured in wavelengths. For the centre frequency of USCT of $2.5 MHz$ this corresponds to a sidelength of $32 mm$ which has to be used in our simulations.

The emitter is sub structured into four elements of piezo ceramics which are separated partially by air and by polyurethane. Figure 6.17 shows the layout of the emitter indicating the scaled sizes.



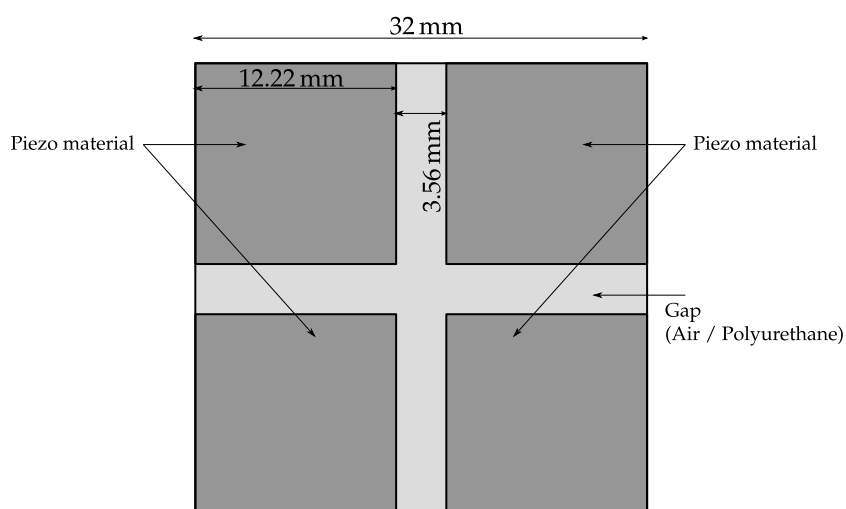


Figure 6.17: Layout of the structured emitter as used in USCT. The dimensions are scaled up from 2.5 MHz for simulation at 70.3 kHz.

Each of the four elements shown in figure 6.17 was represented by 4225 point sources. The sources were placed so that an offset is equally distributed to both sides. This setup is used for several simulations.

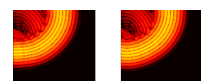
Firstly we visualise the wave front emitted by the surface emitter using the pulse, specified in figure 6.8. Results are shown in figure 6.18, from two perspectives. It can be seen that the pressure field is distributed non-uniformly into the volume. More quantitative analyses are given in the following subsections

6.4.1 Energy distribution

To analyse the difference of the energy distribution caused by a surface source and a point source, an adequate normalisation has to be used. In this case two normalisations are applied. Firstly the distance dependence is removed by multiplication of the squared pressure with r^2 , to obtain the normalised energy equivalent. The second normalisation requires the total energy received to be 1. In this norm it can best be compared which receivers will receive more or less energy. Figure 6.19 shows the b-scans in the chosen norm as well as accumulated normalised energy equivalent over receiver number. It can be seen that the surface source emits more energy to directions normal to the emitter surface. Side lobes are not visible in figure 6.19.

6.4.2 Comparison with a far field simulation

To obtain a quantitative verification of the simulations made, the radiation pattern of the surface source from the numerical solution is compared with a simulation of rectangular surface sources in the far field. This simulation is based on solving the Rayleigh integral within the Fraunhofer approximation which is valid in the far field. The theory is described in [OF89] and was implemented in a tool within the USCT



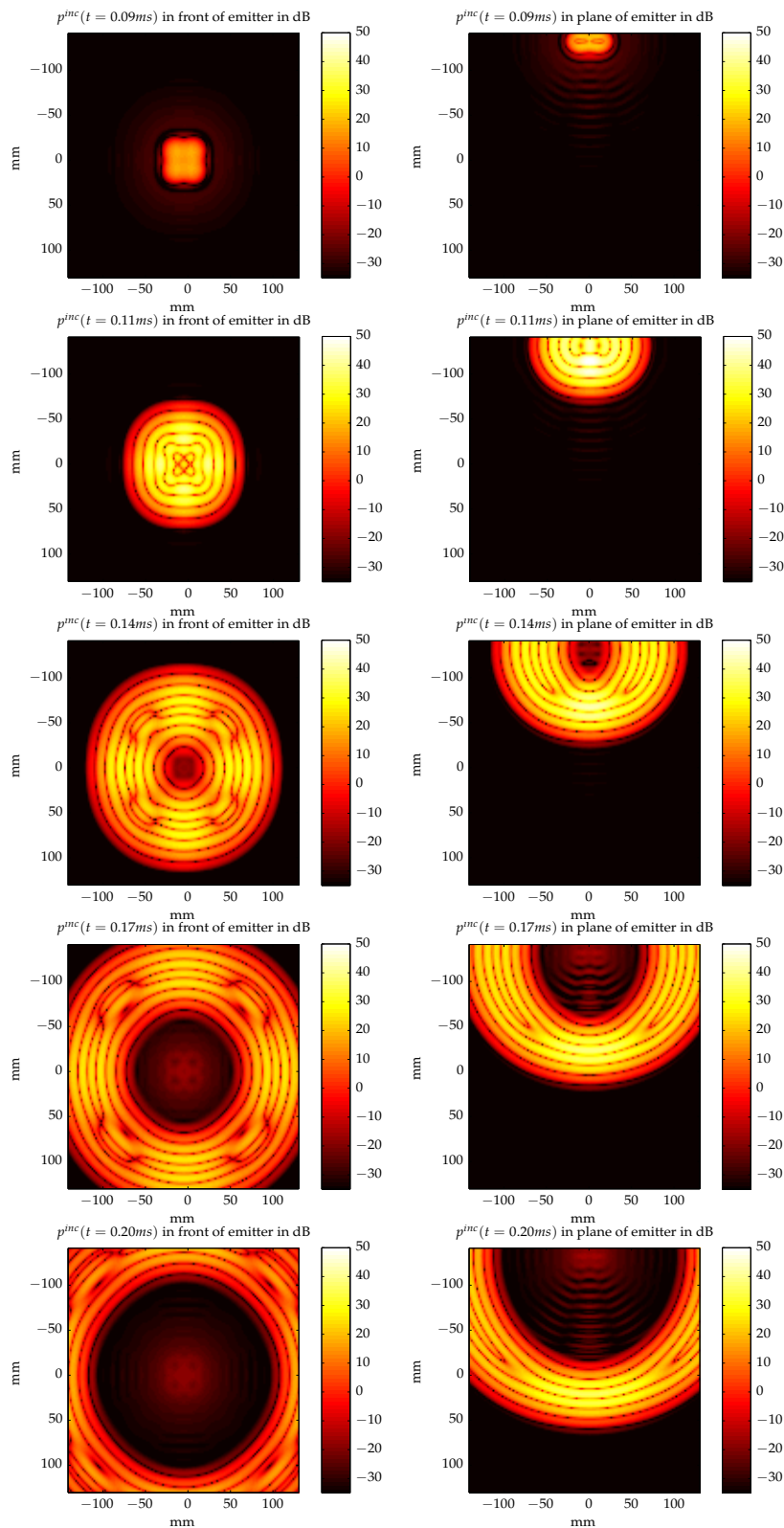
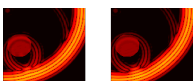


Figure 6.18: Pressure wave emitted by the surface emitter from two perspectives at given time steps. The left column shows the wave front as observed when looking into the emitter, the right column visualises the wave front in plane perpendicular to the emitter and to the left column.



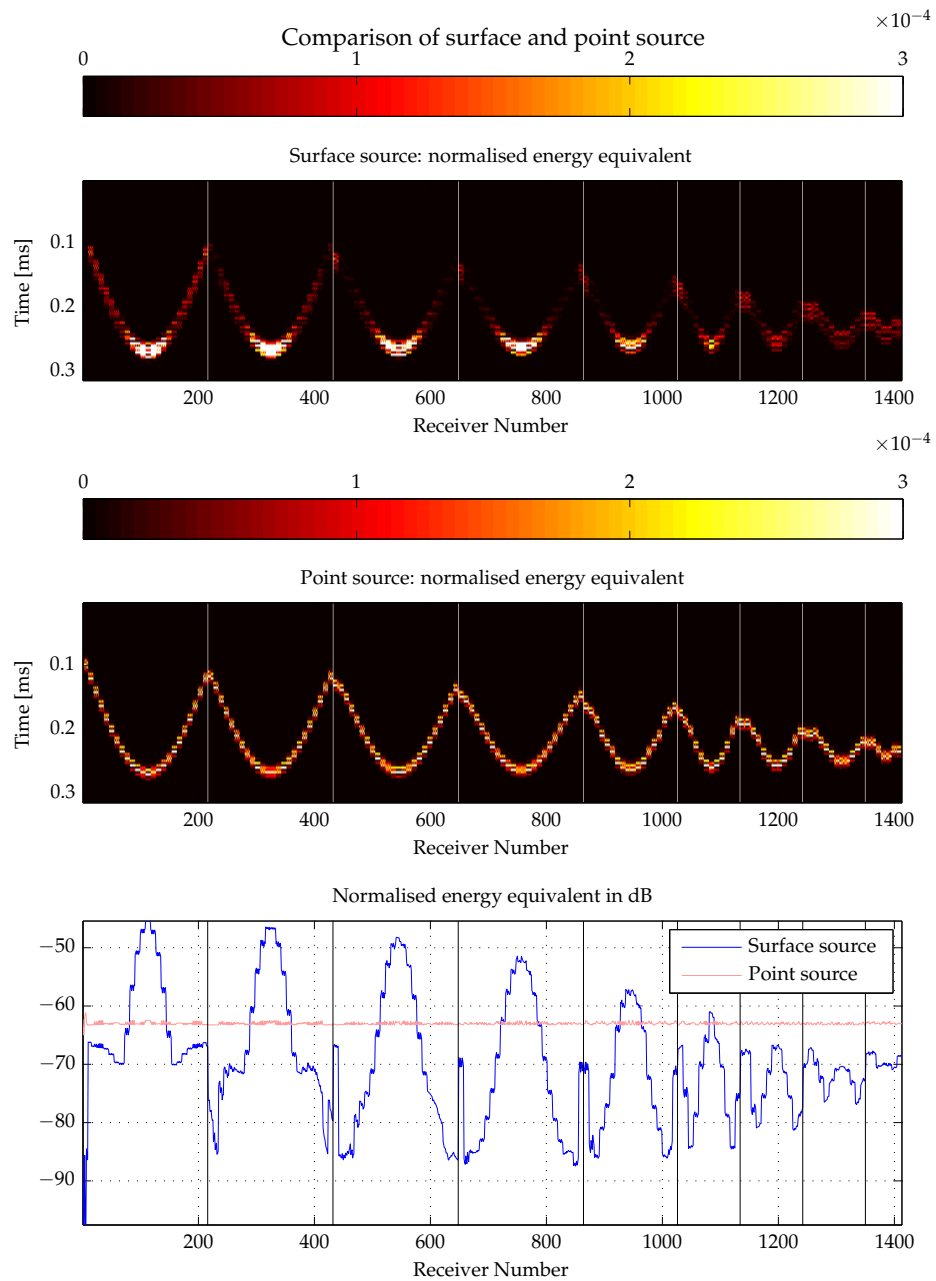
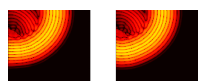


Figure 6.19: Comparison of surface source and point source. All plots show the energy which was corrected for distance dependence and normalised so that the integrals of the bottom plot yields 1. The B-scans on the top show the incident fields in the same norm. Unlike the other B-scans these are not on a logarithmic scale. Both b-scans use the same scale.



project called TAC (Transducer Array Calculation). The Fraunhofer approximation is valid [LLL10] if

$$\frac{a^2}{L\lambda} < 1, \quad (6.3)$$

with the sidelength of the emitter a and the distance L . For our case ($a = 32 \text{ mm}$, $L = 120 \text{ mm}$, $\lambda = 21 \text{ mm}$) equation (6.3) yields 0.4 and the Fraunhofer approximation is valid.

In the numerical solution the directivity pattern in several distances from the source was determined. For the given geometry and wavelength, the far field approximation is valid starting at 8 cm distance from the source (see also figure A.17 in the appendix). The directivity pattern is obtained by collecting values of the pressure amplitude in a half circle of radius 12 cm around the centre of the source.

The radiation pattern obtained with both methods for the centre frequency of 70.3 kHz is compared in figure 6.20. Both plots are normalised to one. The pressure field of a point source is also shown for reference. It can be seen that both simulations yield very similar results. The noise in the exact simulation originates from the low spatial sampling.

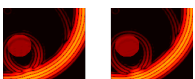
A quantitative comparison is not possible using the polar plot and the output of TAC because access to the raw data is not possible in this specific case. However, a more meaningful and quantitative comparison is possible when using the intensity distribution for many frequencies. For this, the frequency distribution of the pulse used (see figure 6.8) as input to TAC which computes the directivity pattern for each frequency weighted with the Fourier spectrum of the pulse. The same result can be obtained using the exact simulation. Both results are shown and compared in figure 6.21. It can be seen that both simulations differ less than 6%. This is due to the limited spatial resolution which causes error when computing spatial positions from angles.

6.4.3 Result

Simulations taking the spatial dimensions of the USCT emitters into account can be made within the implemented numerical solution. The results are comparable to those within the far field approximation (less than 6% difference). However, computing constraints currently limit the maximal frequency which can be simulated. In the presented case of $f_{max} = 140 \text{ kHz}$ the directivity pattern is very different for the size of the ultrasound emitter of 9 mm. To demonstrate the capability of the simulation the emitter was scaled up. In this setup a full simulation of the USCT including an object like in the previous section is not reasonable.

6.5 Visibility of microcalcifications

Some cancer types have the interesting property that due to their speed of growing, their metabolism is anaerobic. This causes the environment of the growing tumor to be acidic. If the tumor is situated within a mammary gland, the calcium of the milk it contains will precipitate and build clusters of microcalcifications. One element of such a cluster is of 35 μm to 100 μm in diameter. This is a factor six below the wavelength of the USCT. Due to the high speed of sound of hydroxyl apatite out



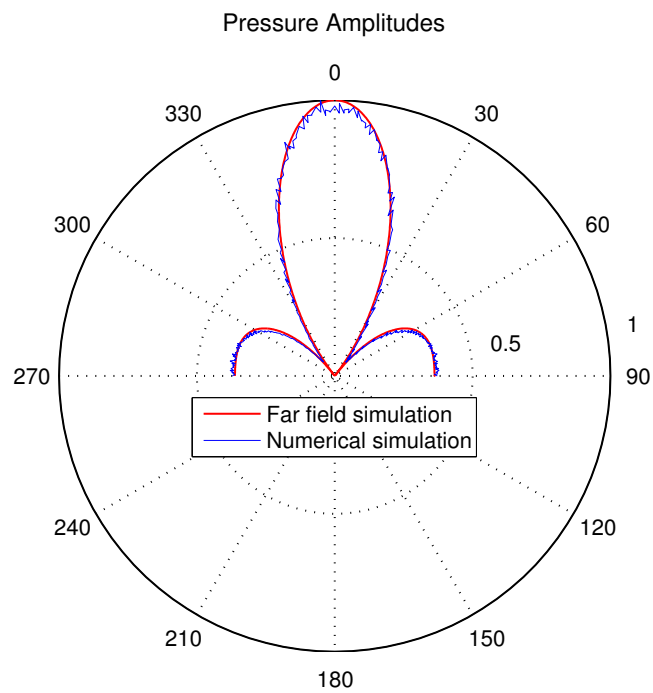


Figure 6.20: Pressure field of the surface source compared with far field and point source. Each pressure field is normalised to one.

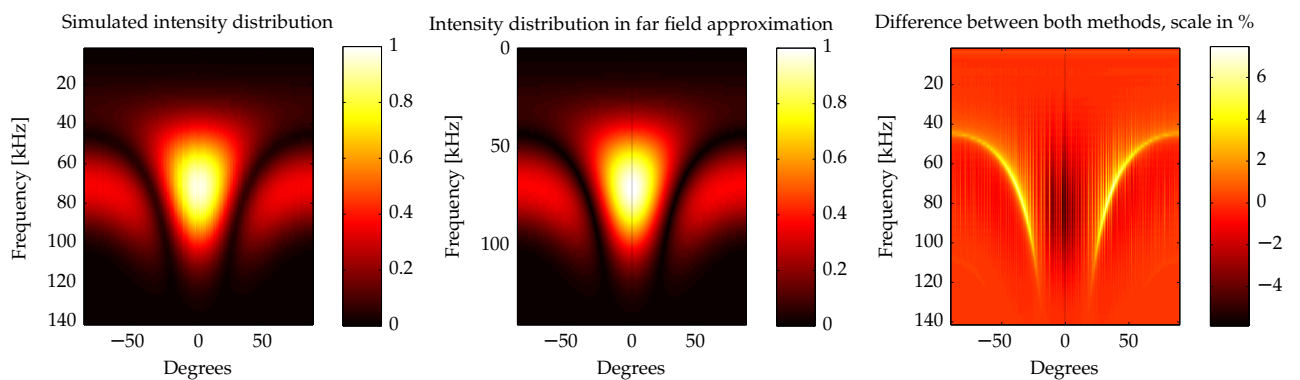
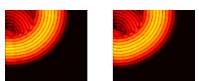


Figure 6.21: Simulated angle – frequency distribution compared with simulation in far field. Intensities are normalised to one. The difference values are shown in percent.



of which the calcifications consist ($6790\frac{m}{s}$) it might be possible to visualise their scattering signals in the USCT, despite it is smaller than a wavelength. However, microcalcifications represent the highest challenge for imaging with USCT.

To investigate the possibility of visualising microcalcifications included in the breast two modified breast models were simulated and reconstructed. One breast phantom was modified to contain one voxel (2.1 mm) of hydroxyl apatite. When scaled to the wavelength of the USCT this corresponds to a size of $60\ \mu\text{m}$. The second simulation was done with the breast model containing a cube of eight voxels of hydroxyl apatite, which models a microcalcification with a diameter of $120\ \mu\text{m}$.

Figure 6.5.1 shows the results of the Born reconstruction and the enhanced reconstruction of both breast phantoms. In (a) the ground truth is shown.

The born reconstruction (b) and the full reconstruction (c) of the phantom with a small inclusion indicate that the calcification cannot be reconstructed, while the large inclusion can be visualised in both types of reconstructions (d) and (e). However, the percental difference between the enhanced reconstructions of unmodified breast phantom and the small microcalcification (f) exhibits that the small inclusion causes changes of up to 20 %. These might be too low to be visualised but they are in the image. The large inclusion, which can be clearly seen in the image causes differences of up to 140 % when compared to the unmodified breast model (g).

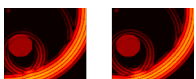
In images (e) and (g) an artefact can be seen in the x-y plane. It originates from the strong scattering of the microcalcification and is due to the ellipsoids which are added into the result image as part of the SAFT algorithm.

An additional simulation of 64^3 voxels and $f_{max} = 70\text{ kHz}$ where one voxel corresponding to a microcalcification of $120\ \mu\text{m}$ was conducted. The results are not shown. The born reconstruction of this simulation shows no sign of the hydroxyl apatite inclusion. Since this lower resolution simulation can not show the calcification which was visible at 128^3 voxels, it is likely, that the smaller calcification of $60\ \mu\text{m}$ will be visible at higher resolutions starting at 256^3 voxels. However, this needs to be shown first.

A comparison of the A-scans is shown in appendix A.2 in figures A.12 through A.14. Visible differences in the selected A-scans occur in the transmitted signals rather than in the reflected ones. It is therefore more likely that transmission tomography is suitable to find microcalcifications than reflection tomography (SAFT) which was used to create the images.

6.5.1 Result

Microcalcifications of sub wavelength sizes of $60\ \mu\text{m}$ can be detected in the USCT using SAFT, but it will be difficult to visualise them. Smaller calcifications are very unlikely to be seen, while very large calcifications ($120\ \mu\text{m}$) will be visible.



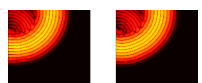
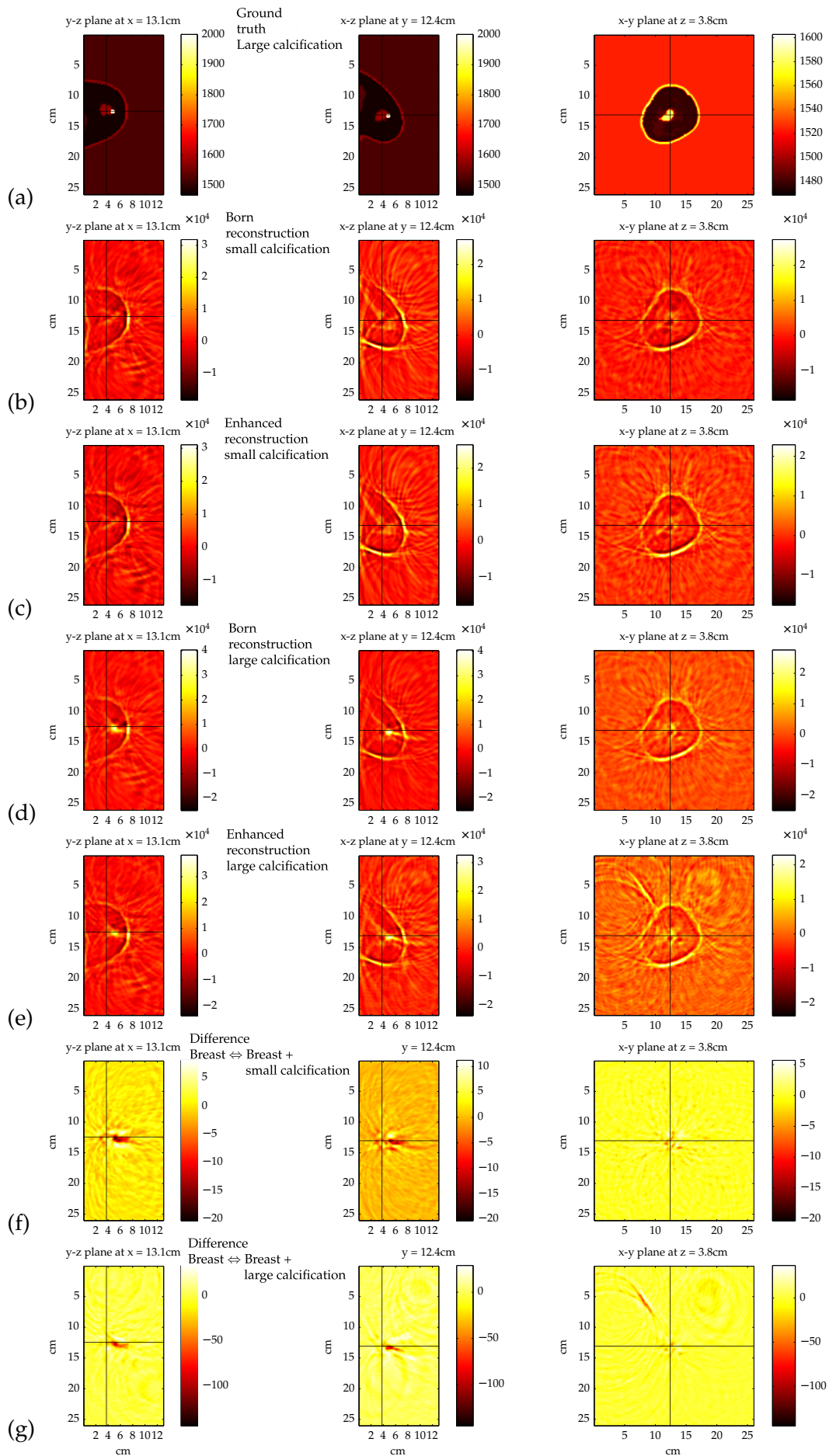
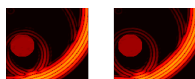


Figure 6.22: (Previous page) Breast model with an inclusion of hydroxyl apatite of $60\ \mu\text{m}$ (not shown) and $120\ \mu\text{m}$ (a). The two models with small and large inclusion are visualised with the born reconstruction and the enhanced reconstruction ((b)-(e)). The difference of the enhanced reconstruction of the unmodified breast model with the small inclusion and the large inclusion are shown in (f) and (g) in percent. For better visualisation, all regions with speed of sound above $2000\ \frac{\text{m}}{\text{s}}$ were clipped.



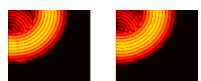
DISCUSSION AND CONCLUSION

When building new experiments or measurement devices it is desirable to simulate the properties of the device before it is finally constructed. This allows to evaluate and to improve the design of the system. Additionally, software components which are required for the analysis of measurement data can be developed while the system is built. In many cases the ground truth, which serves as input data for the simulation is valuable knowledge for the development of data analysis software. For a 3D USCT system, however, simulations are very demanding in terms of the algorithm but also in terms of its hardware requirements with respect to memory size and parallel processing facilities or computing time. This is due to the combination of the ultrasound wave simulation which is already very demanding with the fact that USCT requires 628 emitters to be simulated.

Prior attempts to simulate the system using finite differences required to reduce the system size by a factor 10 [WRS⁺06]. This is because in general finite difference or finite element methods require at least ten points per wavelength to obtain accurate results [LBPT05, IB95]. Therefore, these methods generally limit the size of the problem they can solve.

More advanced methods which employ conjugate gradient schemes to solve the Helmholtz equation of the problem are available [FB93, AD09] and require only four points per wavelength [VH09] but have not been applied to simulate the USCT system so far.

The value of this thesis beyond the state of the art is that it implements an algorithm that generates reliable simulated data for the USCT system. Full confidence in the correctness of the simulation results is guaranteed within the approximations made, so that the simulated data can be trusted and used for further development of the imaging algorithm. Additionally, an easy to use access to distributed computing resources from within Matlab is now available. This is targeted at the benefit of scientific developers who can now develop algorithms on parallel resources. Before, the time for testing software changes on distributed resources took much longer



because deployment was not supported. It was therefore not useful for productive development.

To achieve these goals an integrative architecture for PARallel INfrastructure (PAIN) was designed and implemented. High level functionalities such as automatic compilation, deployment, run and data retrieval were integrated into a user friendly Matlab interface.

derived. The numerical implementation of the derived solution was done with friendly support of Koen van Dongen of TU Delft. The implementation was made under the assumptions that changes in the density can be neglected and that absorption does not occur. The implemented numerical solution was successfully verified with results of an analytical solution. This was also used to quantify the error of the approximation of neglected density changes in the numerical solution. The overall simulation has an interface that creates an processable input for the unmodified reconstruction algorithm so that the simulated data can be used for imaging.

7.1 Discussion of Results and Methodology

The development of the numerical algorithm and the simulation of several test cases with different objects inside the USCT are discussed in the following.

7.1.1 Discussion of the approximations

The impact of two approximations was evaluated. One is made in the simulation, the other in the reconstruction.

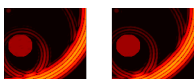
Constant density approximation in the simulations

In the simulation the assumption was made that scattering of pressure waves at density changes can be neglected for human tissue. The impact of this assumption was challenged by using the analytical solution which does not make this assumption.

It was found that the least error is obtained, if the missing changes in the density are reflected by modifying the compressibility and using measured speed of sound values. For the evaluated example of cancer embedded in fat – the simplest model of a breast with a tumor – the error in the spatial energy distribution introduced by this approximation is below 10%. Errors in arrival time were not observed.

Born approximation in the reconstruction

The analysis of the error introduced by the Born approximation which is made in the reconstruction algorithm SAFT (Synthetic Aperture Focussing Technique) was more complex. Therefore, we firstly studied the consequences which the Born approximation has on the results of the simulation. We found the known effects of missing multiple scattering and insensitivity to changes in speed of sound. Additionally, it was found that transmitted signals show a difference of a phase change of π compared to exact simulations. The impact of the neglected velocity of sound was quantified to be of $4 \mu s$ for a breast like object inside the USCT. At a centre frequency of 2.5 MHz and bandwidth of 1 MHz corresponds to more than twice the pulse length. This has a big impact on USCT imaging using SAFT, because pulses



from different directions will not add up at the same point but instead on a ring. This significantly reduces the contrast of the image.

For image reconstruction using SAFT the Born approximation is made in the reconstruction algorithm while exact data are used as input. In order to evaluate the impact of this approximation in imaging, the reconstruction algorithm is first used with Born approximated data. This means we are using the same approximation in both, simulation and imaging. This is called “committing an inverse crime” [Wir04], because it leads to better images than can be generated under realistic circumstances. The inverse crime was used to create reference images. These were compared to reconstructions using exact data as input. The used metric of comparison are percental difference images. The largest differences are found at the surfaces of the objects used in the simulations. This is due to the reflections being placed at wrong positions because of the wrong treatment of the speed of sound within the Born approximation.

The SAFT algorithm offers one extension to go beyond the Born approximation. It uses the spatial distribution of the speed of sound – the so called sound speed map – to compute the average speed of sound along the path which the wave travelled. This is used to find the correct source of the reflection. The required sound speed map is known in simulations or obtained using time-of-flight tomography in measurements. These corrections improve the reconstructed images. These improvements are visualised in percental difference images. It is found that the introduced error is only partially corrected. It was furthermore found that the errors introduced and the corrections take place at different locations and have a larger effect when the object is larger in terms of wavelengths. This suggests that the chosen discretisation of 128^3 voxels and 128 frequencies was not fine-grained enough to model the speed of sound corrections in a realistic way. It indicates that accurate simulations require a finer discretisation.

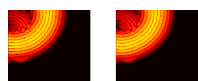
7.1.2 Discussion of the scaled frequencies

The simulations of the USCT within the maximal frequencies of $f_{max} = 140\text{ kHz}$ used have been proven to be correct within the assumption made. However, the USCT system operates at $f_{max} = 3.5\text{ MHz}$. Due to the computing constraints, the simulations can not be run at this frequency.

This results in two effects. The first is that in the simulations the objects inside the USCT are much smaller in terms of wavelengths. The consequence is that scattering is dominated by scattering at objects in the size of the wavelength rather than by refraction. This was visible in the different results the speed of sound correction mentioned above.

The second effect is that the emitters can be treated as point sources at the low frequencies. At higher frequency simulations, the spatial dimension of the emitters leads to a characteristic directivity pattern.

To prove the correctness of the implemented algorithm also for higher frequencies in terms of the directivity pattern, a comparison with a tool (Transducer Array Calculator, TAC) which is specifically designed for the simulation of characteristic directivity patterns was made. A virtual emitter was scaled up so that it had the same size in wavelengths as the real emitters used in USCT at 2.5 MHz .



The tool TAC uses the Rayleigh integral within the Fraunhofer approximation, i.e. results are valid only in the far field. For the geometries used, 12 cm is well within the far field for the two cases compared. At this distance from the emitter the full spectrum was compared with the results of TAC. Less than 5% difference were found. The remaining differences were attributed to the limited spatial resolution in the numerical result, which caused errors when computing the spatial positions from angles.

The contrary effect was exploited to evaluate the visibility of microcalcifications which pose the highest challenge in respect to imaging for USCT. These calcifications indicate aggressive, i.e. fast growing types of cancer and are of sizes between 35 μm and 100 μm . The voxel sidelength of $dr = 2.1 \text{ mm}$ in frequencies with $f_{\text{centre}} = 70 \text{ kHz}$ corresponds to a sidelength of $dr = 60 \mu\text{m}$ at $f_{\text{centre}} = 2.5 \text{ MHz}$. Two cases of microcalcifications were simulated. Each was included in the breast model and had a size of 60 μm or 120 μm in diameter. The smaller inclusion can almost not be visualised, the changes introduced are below 20% of image amplitude. The larger inclusion can clearly be seen, the introduced change is up to 100%. An additional simulation at $f_{\text{centre}} = 35 \text{ kHz}$, where the large calcification corresponds to one voxel, showed that the large calcification cannot be visualised. This indicates that microcalcifications at the size of one voxel (60 μm for USCT) can not be found.

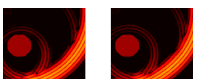
7.1.3 Discussion of the computing infrastructure

The core of a distributed solution is the distributed computing middleware. In this thesis the middleware PAIN was developed. It provides an easy to use and comfortable interface for interactive access to compute resources directly from within Matlab. PAIN was used to run many simulations of different aspects within this thesis in parallel. The performance of the distributed architecture was measured. A minimal overhead of 70 ms is introduced per access to the distributed resources, which allows to even distribute short running tasks. It was demonstrated that tasks of 60 seconds runtime a speed-up of 160 can be reached on 192 hosts. Even at the chosen runtime of 60 s still 83% efficiency can be reached.

We were able to allocate all of the 720 CPU cores provided on the cluster for simulations of the test cases. Optimisations can still be introduced for larger data transfers because only 20% of the available bandwidth were allocated.

Beyond the scalability of PAIN itself, the numerical algorithm for simulation of pressure waves in media scales remarkably well, because it can be hierarchically distributed on three levels: Each emitter can be computed independently. Its simulation can be split into the independent simulation of frequencies. The frequency simulation utilises the Fourier transform, which makes use of multi-core architectures. For example on the grid, this would lead to a distribution of the 628 emitters to 628 computer centres at which one computer would be assigned for each frequency to solve. On each computer, all cores are utilized to compute the fast Fourier transform (fft).

The interface of PAIN is very comfortable for use from within Matlab, because existing code can be used. In addition, when changes to the code are made, an automatic recompilation and redeployment will be triggered. So that the latest software of the user is always available on the remote resources. It is also possible to use PAIN from the commandline to remotely execute arbitrary Unix commands.



This also includes the execution of user provided executables. In this way PAIN can also be used for distributed C-code.

The Matlab license model permits the way in which PAIN distributes the executables. The campus license needs to be installed only on the developers computer. It includes the Matlab compiler used to create the executable which is run on the distributed resources. On these resources the Matlab compiler Runtime needs to be installed. It is free of charge.

7.1.4 Discussion of the limited resolution

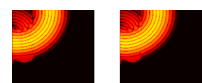
The simulations have been conducted in the coarse grained grid of 128^3 voxels and 128 frequencies ($f_{max} = 140 \text{ kHz}$). The spatial and the temporal discretisations are coupled via the requirement that four points per wavelength are required to compute the convolution without aliasing artefacts. The results of chapter 6, especially for detailed analysis of the reconstructions using simulated data with SAFT, but also the spatial dimensions of the emitters indicate that simulations at higher frequencies are required for more detailed analyses. For this spatial discretisation has to be more fine-grained. In case of USCT frequencies of up to 3.5 MHz are used. To simulate this case 4096^3 voxels and 4096 frequencies and 64 TB RAM are required on every computer in the distributed system. A rough estimate shows that the computing time of at least 40,000 years on the cluster with 720 cores is finally the largest problem.

The approach of the filtered convolution method allows to carry out convolutions with only two points per wavelength [VH09]. In this case the problem can be reduced to a discretisation of only 2048^3 voxels and still 4096 frequencies.

If additionally, only frequencies of up to 3 MHz can be accepted for the simulation, the discretisation requirements drop to 1024^3 voxels and 2048 frequencies. This represents the largest case which can be solved today. It requires computers with 256 GB RAM, which are available in some modern HPC clusters. To estimate the required computing time, we assume an additional factor two introduced by the filtered convolution method and assume that no additional RAM will be required. Due to the good scalability of the simulation, its runtime can roughly be assessed via the available number of CPU cores of a system. A factor eight increase of the number of voxels (e.g. $128^3 \rightarrow 256^3$) leads to a factor eight more memory and a factor $2 \cdot 8^{1.2}$ more CPU time. The factor two represents the increased amount of required frequencies. Additionally, more iterations are required by the algorithm at higher resolutions to reach convergence. This effect is difficult to predict and was therefore not taken into account when extrapolating computing times. For the case of 1024^3 this leads to at least 80 years of computation time on our 720 core cluster. Using cloud computing resources for this computation can be rented for 15 million € at Amazon. The computing time depends only on the number of resources which can be allocated simultaneously.

On the fastest German supercomputer (Cray XE6 in Stuttgart) this computation can be executed in approximately two months. In this case modifications of the software have to be made with regard to the HPC architecture and to the distribution of tasks.

If the whole LHC computing grid could be allocated simultaneously and if all computers had the required 256 GB RAM, this computation could be done in



approximately 40 days on the grid. In this case slight modifications of PAIN are required.

Graphical CPUs (GPUs) are used more recently by many groups to speed up computations. However, GPUs do not appear to be useful in this case, because the fast Fourier transformation (fft) is the critical part in the simulation for the performance. The problem is that for the fft all data has to fit into the memory of the graphics card. For simulations of 1024^3 voxels, 256 GB of RAM are required, which exceeds the capabilities of today's graphics cards.

7.2 Conclusions and Recommendations for Future Work

The aim of this thesis to generate exact simulated data for USCT was reached. Additionally, characteristics of the assumptions within SAFT and the speed of sound correction were quantified. Furthermore, a middleware for parallel computing was developed so that cluster or cloud access from Matlab is possible without expert knowledge in this field.

For extending the simulations towards the high frequencies used in USCT future two options are suggested. One is the evaluation and integration of the filtered convolution method so that only two points per wavelength are required for convolutions. This allows a reduction of the spatial discretisation by a factor two, resulting in a factor eight smaller problem size and is crucial for higher frequency simulations.

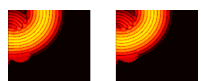
The other item for future work is to clarify the hardware platform to use for the large scale simulations. The adaptations which have to be made to exploit the full potential of the chosen architecture are very architecture specific. This also means that the general part is already implemented within this thesis.

Both topics need to be addressed so that simulations with 512^3 Voxels and $f_{max} = 1.2 \text{ MHz}$ can be carried out. Simulations with 2.4 MHz may be possible depending on the performance of the resources which can be accessed. Simulations with 3.5 MHz are unlikely to be computable in the near future.



BIBLIOGRAPHY

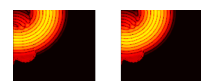
- [AAB⁺02] ARNOLD, D. ; AGRAWAL, S. ; BLACKFORD, S. ; DONGARRA, J. ; MILLER, M. ; SEYMOUR, K. ; SAGI, K. ; SHI, Z. ; VADHIYAR, S.: *Users' Guide to NetSolve V1.4.1*. Knoxville, TN, June 2002 (ICL-UT-02-05). – Innovative Computing Dept. Technical Report
- [ABB⁺02] APOSTOLAKIS, J. ; BARRAND, G. ; BRUN, R. ; BUNCIC, P. ; INNOCENTE, V. ; MATO, P. ; PFEIFFER, A. ; QUARRIE, D. ; RADEMAKERS, F. ; TAYLOR, L. u. a.: *Report of the LHC Computing Grid Project Architecture Blueprint RTAG*. Oct, 2002
- [AD09] ALLES, E. J. ; DONGEN, K. W. A.: Frequency domain perfectly matched layers for acoustic scattering integral equation problems. In: *Ultrasonics Symposium (IUS), 2009 IEEE International*, 2009, S. 1610–1613
- [Amd67] AMDAHL, Gene M.: Validity of the single processor approach to achieving large scale computing capabilities. In: *AFIPS spring joint computer conference* (1967). <http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf>
- [AS65] ABRAMOWITZ, M. ; STEGUN, I.: *Handbook of Mathematical functions with Formulas, Graphs and Mathematical Tables*. National Bureau of Standards, 1965. – ISBN 0 486 61272 4
- [AW05] ARFKEN, G.B. ; WEBER, H.J.: *Mathematical methods for physicists*. Elsevier, 2005. – ISBN 9780120598762
- [AWH05] ARFKEN, George B. ; WEBER, Hans J. ; HARRIS, Frank E.: *Mathematical Methods for Physicists, Sixth Edition: A Comprehensive Guide*. 6. Academic Press, 2005. – ISBN 0120598760
- [Bak83] BAKER, Christopher T. H. ; MILLER, Geoffrey F. (Hrsg.): *Treatment of Integral Equations by Numerical Methods*. Academic Pr, 1983. – ISBN 0120741202
- [BNLS01] BOONE, J. M. ; NELSON, T. R. ; LINDFORS, K. K. ; SEIBERT, J. A.: Dedicated Breast CT: Radiation Dose and Image Quality Evaluation1. In: *Radiology* 221 (2001), Nr. 3, S. 657–667



- [BR76] BRENNER, N. ; RADER, C.: A New Principle for Fast Fourier Transformation. In: *IEEE Acoustics, Speech and Signal Processing* 24 (1976), S. 264–266. ISBN doi:10.1109/TASSP.1976.1162805
- [Bra00] BRACEWELL, R.N.: *The Fourier transform and its applications*. McGraw Hill, 2000 (McGraw-Hill series in electrical and computer engineering). – ISBN 9780073039381
- [BZ92] BERG, P.M. van d. ; ZWAMBORN, Peter: The Three-Dimensional Weak Form of the Conjugate Gradient FFT Method for Solving Scattering Problems. In: *IEEE transactions on microwave theory and techniques*, Vol. 40, No. 9 (1992)
- [Cam08] CAMPOS, Isabel: *The int.eu.grid user guide*. Version: April 2008. <http://savannah.fzk.de/i2g-docs/i2g-UserGuide-v1.1.pdf>, Abruf: 15. August 2008. Webpublished, <http://savannah.fzk.de/i2g-docs/i2g-UserGuide-v1.1.pdf>
- [Der02] DERSHAW, David D. (Hrsg.): *Imaging-Guided Interventional Breast Techniques*. 1. Springer, 2002. – ISBN 0387954546
- [DGR12] DAPP, Robin ; GEMMEKE, Hartmut ; RUITER, Nicole: 3D refraction-corrected transmission reconstruction for 3D ultrasound computer tomography. In: *Proceedings of SPIE* 8320 (2012), Februar, Nr. 1, 832014–832014-7. <http://dx.doi.org/doi:10.1117/12.911266>. – DOI doi:10.1117/12.911266. – ISSN 0277786X
- [DGT11] D-GRID TEAM, Service Level A.: *Zwischenbericht des Projektes Service Level Agreements für das D-Grid*. 2011
- [Don10] DONGEN, Koen van: Acoustic plane wave scattering from a spherical soft contrast. In: *unpublished* (2010)
- [Duc90] DUCK, Francis A.: *Chapter 4 Acoustic Properties of Tissue at Ultrasonic Frequencies*. Academic Press, 1990. – 73–135 S.
- [ES01] ERWIN, D. ; SNELLING, D.: UNICORE: A Grid computing environment. In: *Euro-Par 2001 Parallel Processing* (2001), S. 825–834
- [FAB93] FOURNIER, D. v. ; ANTON, H.J.H.-W. ; BASTERT, G. ; BANNASCH, P. (Hrsg.): *Breast Cancer Screening*. 1993
- [FB93] FOKKEMA, J.T. ; BERG, P.M. van d.: *Seismic Applications of Acoustic Reciprocity*. Elsevier, 1993. – ISBN 0 444 89044 0
- [Fit77] FITCH, John: Profiling a large program. In: *Software: Practice and Experience* 7 (1977), Nr. 4, 511–518. <http://dx.doi.org/10.1002/spe.4380070410>. – DOI 10.1002/spe.4380070410. – ISSN 1097–024X
- [FJ05] FRIGO, M. ; JOHNSON, S.G.: The Design and Implementation of FFTW3. 93 (2005), Feb, S. 216 – 231. – ISBN doi:10.1109/JPROC.2004.840301



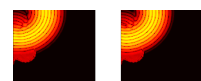
- [FK97] FOSTER, Ian ; KESSELMAN, Carl: Globus: A Metacomputing Infrastructure Toolkit. In: *International Journal of High Performance Computing Applications* 11 (1997), Juni, Nr. 2, 115–128. <http://dx.doi.org/10.1177/109434209701100205>. – DOI 10.1177/109434209701100205. – ISSN 1094–3420, 1741–2846
- [FK98] FOSTER, I (Hrsg.) ; KESSELMAN, C. (Hrsg.): *The grid: blueprint for a new computing infrastructure*. San Francisco, Calif. : Morgan Kaufmann Publ., 1998. – ISBN 1–55860–475–8
- [FK04] FOSTER, I. (Hrsg.) ; KESSELMAN, C. (Hrsg.): *The grid: blueprint for a new computing infrastructure*. Elsevier, 2004 (The Morgan Kaufmann Series in Computer Architecture and Design Series). <http://books.google.com/books?id=8-0BofIhoUOC>. – ISBN 9781558609334
- [Fri00] FRIEDENSON, B.: Is mammography indicated for women with defective BRCA genes? Implications of recent scientific advances for the diagnosis, treatment, and prevention of hereditary breast cancer. In: *Med Gen Med* 9 (2000)
- [GD05] GABE, R. ; DUFFY, S. W.: Breast cancer: Evaluation of service screening mammography in practice: the impact on breast cancer mortality. In: *Annals of oncology* 16 (2005), Nr. suppl 2, S. ii153
- [GN09] GØTZSCHE, PC ; NIELSEN, M: Screening for breast cancer with mammography. (2009). <http://dx.doi.org/10.1002/14651858.CD001877>. – DOI 10.1002/14651858.CD001877
- [GN11] GØTZSCHE, Peter C. ; NIELSEN, M: Screening for breast cancer with mammography. (2011). <http://dx.doi.org/10.1002/14651858.CD001877.pub4>. – DOI 10.1002/14651858.CD001877.pub4
- [GO00] GØTZSCHE, P. C. ; OLSEN, O.: Is screening for breast cancer with mammography justifiable? In: *The Lancet* 355 (2000), Nr. 9198, S. 129–134
- [Gol91] GOLDBERG, David: What Every Computer Scientist Should Know About Floating-Point Arithmetic. In: *Association for Computing Machinery (ACM) Computing Surveys* (1991), March, 171-264. <http://www.validlab.com/goldberg/paper.pdf>. ISBN doi:10.1145/103162.103163. – Retrieved 2012-01-04
- [Har06] HARDT, M: Virtualisation for grid-computing. In: *Proceedings of Computing in High Energy and Nuclear Physics CHEP 2006* Bd. II, Macmillan India Ltd., 2006, S. 777–782. – Query date: 15.02.2011
- [Har08] HARDT, M., SEYMOUR, K., DONGARRA, J., ZAPF, M., RUITER, N.V.: Interactive grid access using GridSolve and giggle. In: *Computing and Informatics, Vol 27, 2008, 233-248* (2008)
- [HHH10] HARDT, M. ; HAMMAD, A. ; HOPP, T.: Simple tool for application runs and transfer of data (s.t.a.r.t.). In: *Proc.of the 18th EUROMICRO*, Grosspietsch, E., 2010. – ISBN ISBN 978–3–902457–26–4, S. 443–448



- [Hof11] HOFMANN, Matthias: *IGE Requirements Management Process*. <http://www.ige-project.eu/hub/rt/IGERequirementsManagementProcess.pdf>. Version: 2011
- [Hol01] HOLM, S.: Ultrasim-a toolbox for ultrasound field simulation. In: *University of Oslo* (2001)
- [Hoo78] HOOKE, Robert: *Ut tensio, sic vis (As the extension, so the force)*. 1678 http://webapps.qmul.ac.uk/cell/Hooke/hooke_folio.php?id=80&option=both
- [Hop09] HOPP, Torsten: *Parallelisierung des Backpropagation Algorithmus "Iterative Born" mit MPI auf dem Grid*, Ruprecht-Karls-Universität Heidelberg, Diplomarbeit, 2009. <http://www.ipe.fzk.de/~stotzka/publications/publications/hopp2009.pdf>
- [HS52] HESTENES, M.R. ; STIEFEL, E.: The conjugate-gradient method for solving linear problems. In: *Journal of Research of the National Bureau of Standards* 49 (1952), S. 409–435
- [HS11] HUTHWAITE, P. ; SIMONETTI, F.: High-resolution imaging without iteration: a fast and robust method for breast ultrasound tomography. In: *The Journal of the Acoustical Society of America* 130 (2011), Nr. 3, 1721–1734. <http://dx.doi.org/10.1121/1.3613936>. – DOI 10.1121/1.3613936
- [HZR08] HARDT, M. ; ZAPF, M. ; RUITER, N.V.: *Interactive Grid-Access using Matlab*. (2008)
- [IB95] IHLENBURG, F. ; BABUŠKA, I.: Dispersion analysis and error estimation of Galerkin finite element methods for the Helmholtz equation. In: *International Journal for Numerical Methods in Engineering* 38 (1995), Nr. 22, S. 3745–3774
- [Ima12] IMAGING, Medray: *Breast MRI*. <http://www.medrayimaging.com/page/breast-mri>. Version: 2012. – Last visited May 15, 2012
- [Inc12] INC., Cyberlogic: *Website of wave3000*. <http://www.cyberlogic.org/wave3000.html>. Version: 1012. – Last visited May 2012
- [Jen96] JENSEN, Jørgen A.: FIELD: A Program for Simulating Ultrasound Systems. In: *10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING, VOL. 4, SUPPLEMENT 1, PART 1:351–353*, 1996, S. 351–353
- [JN00] JENSEN, J. A. ; NIKOLOV, I.: Fast simulation of ultrasound images. In: *Ultrasonics Symposium, 2000 IEEE Bd. 2*, 2000, S. 1721–1724
- [Jon12] JONG, Irmen de: *Python Remote Objects – pyro*. <http://packages.python.org/Pyro>. Version: 2012. – Last visited May 2012
- [Kah96] KAHAN, Prof. W.: *Lecture Notes on the Status of IEEE Standard 754 for Binary Floating-Point Arithmetic*. (1996), May



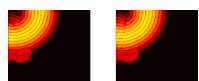
- [KB91a] KLEINMAN, R.E. ; BERG, P.M. van d.: Iterative methods for solving integral equations. In: *Radio Science, Volume 26, Number 1* (1991), S. 175–181. – doi:10.1029/90RS00934
- [KB91b] KLEINMAN, R.E. ; BERG, P.M. van d.: Iterative methods for solving integral equations. In: *PIER, Progress In Electromagnetics Research* (1991). – ISSN 1070–4698
- [KB06] KATALINIC, Alexander ; BARTEL, Carmen: *Epidemiologie Mammakarzinom*. March 2006. – Institut für Krebs epidemiologie e.V., Universität Lübeck
- [KHSK07] KO, Eun Y. ; HAN, Boo-Kyung ; SHIN, Jung H. ; KANG, Seok S.: Breast MRI for Evaluating Patients with Metastatic Axillary Lymph Node and Initially Negative Mammography and Sonography. In: *Korean Journal of Radiology* 8 (2007), Nr. 5, 382. <http://dx.doi.org/10.3348/kjr.2007.8.5.382>. – DOI 10.3348/kjr.2007.8.5.382. – ISSN 1229–6929
- [Kra09] KRANZLMULLER, D.: The future European Grid Infrastructure #x2014; Roadmap and challenges. (2009), Juni, S. 17 –20. <http://dx.doi.org/10.1109/ITI.2009.5196044>. – DOI 10.1109/ITI.2009.5196044
- [Lar97] LARMOR, Joseph: A Dynamical Theory of the Electric and Luminiferous Medium. Part III. Relations with Material Media. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 190 (1897), 205–493. <http://dx.doi.org/10.1098/rsta.1897.0020>. – DOI 10.1098/rsta.1897.0020
- [Lau04] LAURE, E.: EGEE Middleware Architecture. In: *E Dm S* 4 (2004), Nr. 7, S. 64
- [LBPT05] LAGHROUCHE, O. ; BETTESS, P. ; PERREY-DEBAIN, E. ; TREVELYAN, J.: Wave interpolation finite elements for Helmholtz problems with jumps in the wave speed. In: *Computer methods in applied mechanics and engineering* 194 (2005), Nr. 2, S. 367–381
- [LLL10] LIPSON, Ariel ; LIPSON, Stephen G. ; LIPSON, Henry: *Optical Physics*. 4. Cambridge University Press, 2010. – ISBN 0521493455
- [Mar84] MARFURT, Kurt J.: Accuracy of Finite-Difference and Finite-Element Modeling of the Scalar and Elastic Wave Equations. In: *Geophysics* 49 (1984), Mai, Nr. 5, 533–549. <http://dx.doi.org/10.1190/1.1441689>. – DOI 10.1190/1.1441689. – ISSN 0016–8033, 1942–2156
- [MCC⁺08] MARCO, J ; CAMPOS, I ; COTERILLO, I ; DÍAZ, I ; LÓPEZ, A ; MARCO, R ; MARTÍNEZ-RIVERO, C ; ORVIZ, P ; ROGRÍGUEZ, D ; GOMES, J ; BORGES, G ; MONTECELO, M ; DAVID, M ; SILVA, B ; DIAS, N ; MARTINS, J.P ; FERNÁNDEZ, C ; GARCÍA-TORRÉS, L ; VEIGA, C ; CORDERO, D ; CACHEIRO, J.L. ; LÓPEZ, I ; GARCÍA-TOBÍO, J ; COSTAS, N ; NO, J.C.M. ; GÓMEZ, A ; BOGACKI, W ; MEYER, N ; OWSIAK, M ; PŁÓCIENNIK, M ; POSPESZNY, M ; ZAWADZKI, M ; HAMMAD, A ; HARDT, M ; FERNÁNDEZ, E ; HEYMANN, E ; SENAR, M.A ; PADEE, A ; NAWROCKI,



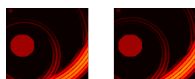
- K ; WISKLICKI, W ; HEINZLREITER, P ; BAUMGARTNER, M ; ROSMANITH, H ; KRANZLMÜLLER, D ; VOLKERT, J ; KENNY, S ; COGHLAN, B ; PAJAK, R ; MOSURSKA, Z ; SZYMOCHA, T ; LASON, P ; SKITAL, L ; FUNIKA, W ; KORCYL, K ; PIECZYKOLAN, J ; OS, K B. ; OTA, R S. ; GUZY, K ; DUTKA, L ; KITOWSKI, J ; ZIELIŃSKI, K ; HLUCHÝ, L ; DOBRUCKÝ, M ; SIMO, B ; HABALA, O ; ASTALOS, J ; CIGLAN, M ; SÍPKOVÁ, V ; GATIAL, E ; BABÍK, M ; REYNOLDS, J.M ; VALLÉS, J.M ; SERRANO, F ; TARANCÓN, A ; VELASCO, J.L ; CASTEJÓN, F ; DICHEV, K ; KELLER, R ; STORK, S: The Interactive European Grid: Project Objectives and Achievements. In: *Computing and Informatics* 27 (2008), 171-176. <http://www.cai.sk/ojs/index.php/cai/article/view/258/209>
- [McG04] MCGOUGH, Robert J.: Rapid calculations of time-harmonic nearfield pressures produced by rectangular pistons. In: *The Journal of the Acoustical Society of America* 115 (2004), Nr. 5, 1934–1941. <http://dx.doi.org/10.1121/1.1694991>. – DOI 10.1121/1.1694991
- [McP10] MCPHERSON, K.: Screening for breast cancer—balancing the debate. In: *BMJ* 340 (2010)
- [MF53] MORSE, P. M. ; FESHBACH, H.: *Methods of Theoretical Physics, Part I*. New York: McGraw-Hill, 1953. – 125–126, 271 and 509–510 S.
- [MKAF07] MITCHELL, Richard S. ; KUMAR, Vinay ; ABBAS, Abul K. ; FAUSTO, Nelson: Robbins Basic Pathology. (2007), S. 739ff. ISBN 1–4160–2973–7
- [Mol89] MOLESWORTH, R.: An overview of the CCITT X.500 directory system. In: *Telecommunications, 1989. Second IEE National Conference on*, 1989, S. 179–183
- [MSM⁺03] MICHAELSON, J. ; SATIJA, S. ; MOORE, R. ; WEBER, G. ; HALPERN, E. ; GARLAND, A. ; KOPANS, D. B. ; HUGHES, K.: Estimates of the sizes at which breast cancers become detectable on mammographic and clinical grounds. In: *JOURNAL OF WOMENS IMAGING* 5 (2003), S. 3–10
- [NCN⁺97] NIKLASON, L T. ; CHRISTIAN, B T. ; NIKLASON, L E. ; KOPANS, D B. ; CASTLEBERRY, D E. ; OPSAHL-ONG, B H. ; LANDBERG, C E. ; SLANETZ, P J. ; GIARDINO, A A. ; MOORE, R. ; ALBAGLI, D. ; DEJULE, M C. ; FITZGERALD, P F. ; FOBARE, D F. ; GIAMBATTISTA, B W. ; KWASNICK, R F. ; LIU, J. ; LUBOWSKI, S J. ; POSSIN, G E. ; RICHOTTE, J F. ; WEI, C Y. ; WIRTH, R F.: Digital Tomosynthesis in Breast Imaging. In: *Radiology* 205 (1997), November, Nr. 2, 399–406. <http://radiology.rsna.org/content/205/2/399>. – ISSN 0033–8419, 1527–1315
- [New86] NEWTON, Isaac: *Philosophiæ Naturalis Principia Mathematica*. 1686 <http://www.ntnu.no/ub/spesialsamlingene/ebok/02a019654.html>
- [NKG07] NEUROTH, Heike ; KERZEL, Martina ; GENTZSCH, Wolfgang: *German Grid Initiative D-Grid*. Universitätsverlag Göttingen, 2007. – ISBN 9783938616994



- [NL79] NORTON, Stephen J. ; LINZER, Melvin: Ultrasonic reflectivity tomography: Reconstruction with circular transducer arrays. In: *Ultrasonic Imaging* 1 (1979), April, Nr. 2, 154–184. [http://dx.doi.org/10.1016/0161-7346\(79\)90014-2](http://dx.doi.org/10.1016/0161-7346(79)90014-2). – DOI 10.1016/0161–7346(79)90014–2. – ISSN 0161–7346
- [NS28] NYQUIST, H. ; SHANNON, C.: Certain topics in telegraph transmission theory. In: *Transactions AIEE* 47 (1928), April, 617–644. <http://replay.web.archive.org/20060706192816/http://www.loe.ee.upatras.gr/Comes/Notes/Nyquist.pdf>. – Reprint as classic paper in: Proc. IEEE, Vol. 90, No. 2, Feb 2002.
- [OF89] OCHELTREE, Kenneth B. ; FRIZZELL, Leon A.: Sound Field Calculation for Rectangular Sources. 36 (1989), S. 242 – 248
- [OG01] OLSEN, O. ; GÖTZSCHE, P. C.: Cochran review on screening for breast cancer with mammography. In: *The Lancet* 358 (2001), Nr. 9290, S. 1340–1342
- [RDZG10] RUITER, N.V. ; DAPP, R. ; ZAPF, M. ; GEMMEKE, H.: A new method for grating lobe reduction for 3D synthetic aperture imaging with ultrasound computer tomography. (2010), oct., S. 2372 –2375. <http://dx.doi.org/10.1109/ULTSYM.2010.5935752>. – DOI 10.1109/ULTSYM.2010.5935752. – ISSN 1948–5719
- [RKI07] RKI, Robert Koch I.: *Verbreitung von Krebserkrankungen in Deutschland*. <http://rki.de>. Version: 2007
- [RKI08] RKI, Robert Koch I.: *Krebs in Deutschland 2007/2008 Häufigkeiten und Trends*. <http://rki.de>. Version: 2008
- [RSZG07] RUITER, N.V. ; SCHNELL, R. ; ZAPF, M. ; GEMMEKE, H.: Phase Aberration Correction for 3D Ultrasound Computer Tomography Images. (2007), Oktober, S. 1808 –1811. <http://dx.doi.org/10.1109/ULTSYM.2007.455>. – DOI 10.1109/ULTSYM.2007.455
- [RSZG08a] RUITER, N.V. ; SCHWARZENBERG, G.F. ; ZAPF, M. ; GEMMEKE, H.: Improvement of 3D ultrasound computer tomography images by signal pre-processing. In: *Ultrasonics Symposium, 2008. IUS 2008. IEEE*, 2008, S. 852 –855
- [RSZG08b] RUITER, N.V. ; SCHWARZENBERG, G.F. ; ZAPF, M. ; GEMMEKE, H.: Improvement of 3D ultrasound computer tomography images by signal pre-processing. (2008), November, S. 852 –855. <http://dx.doi.org/10.1109/ULTSYM.2008.0205>. – DOI 10.1109/ULTSYM.2008.0205
- [SBa11] SMIRNOVA, O. ; BEHRMANN, G. ; AL., M. G.: *Building and Operating a Distributed Regional Centre for LHC Computing and Data Storage*. 2011
- [SBB⁺10] SCHWIEGELSHOHN, U. ; BADIA, R. M. ; BUBAK, M. ; DANELUTTO, M. ; DUSTDAR, S. ; GAGLIARDI, F. ; GEIGER, A. ; HLUCHY, L. ; KRANZLMÜLLER, D. ; LAURE, E. u. a.: Perspectives on grid computing. In: *Future Generation Computer Systems* 26 (2010), Nr. 8, S. 1104–1115



- [SD96] STEEN, A. J. d. ; DONGARRA, J. J.: Overview of recent supercomputers. In: *NHSE Review* 1 (1996), S. 96–1
- [She94] SHEWCHUK, Jonathan R.: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain. (1994), August
- [SMLF09] SOTOMAYOR, B. ; MONTERO, R.S. ; LLORENTE, I.M. ; FOSTER, I.: Virtual Infrastructure Management in Private and Hybrid Clouds. In: *Internet Computing, IEEE* 13 (2009), sept.-oct., Nr. 5, S. 14–22. <http://dx.doi.org/10.1109/MIC.2009.119>. – DOI 10.1109/MIC.2009.119. – ISSN 1089–7801
- [Ste73] STEPANISHEN, P. R.: The impulse response and time dependent force on a baffled circular piston and a sphere. In: *Journal of Sound and Vibration* 26 (1973), Nr. 3, S. 287–298
- [THH12] TENSCHERT, Axel ; HAGEMEI,ER, Björn ; HAMMAD, Ahmad: *Die D-Grid Service Level Agreement-Schicht und ihre Dienste*. 2012
- [VH09] VERWEIJ, M. D. ; HUIJSSEN, J.: A filtered convolution method for the computation of acoustic wave fields in very large spatiotemporal domains. In: *The Journal of the Acoustical Society of America* 125 (2009), S. 1868. <http://dx.doi.org/10.1121/1.3077220>. – DOI 10.1121/1.3077220
- [VSHS11] VIEIRA, I. T. ; SENNA, V. de ; HARPER, P. R. ; SHAHANI, A. K.: Tumour doubling times and the length bias in breast cancer screening programmes. In: *Health Care Management Science* (2011), S. 1–9
- [Wei] WEISSTEIN, Eric W.: *Convolution Theorem*. <http://mathworld.wolfram.com/ConvolutionTheorem.html>. – Retrieved 2012-01-04
- [WHO08] WHO, World Health O.: *World Cancer Report*. <http://globocan.iarc.fr>. Version: 2008. – Retrieved 2011-11-16
- [Wik] WIKIPEDIA: *Convolution theorem*. https://en.wikipedia.org/wiki/Convolution_theorem. – Retrieved 2012-01-04
- [Wir04] WIRGIN, Armand: The inverse crime. In: *arXiv:math-ph/0401050* (2004), Januar. <http://arxiv.org/abs/math-ph/0401050>
- [WRS⁺06] WEBER, M. ; RUITER, N. V. ; SCHWARZENBERG, G. ; ZAPF, M. ; MÜLLER, T. O.: Ultraschallsimulation für die Ultraschall-Computertomographie. In: *Bildverarbeitung für die Medizin 2006* (2006), S. 439–443
- [YDS07] YARKHAN, A ; DONGARRA., J ; SEYMOUR, K: GridSolve: The Evolution of Network Enabled Solver. (2007), S. 215–226
- [Zee97] ZEEMAN, Pieter: The Effect of Magnetisation on the Nature of Light Emitted by a Substance. In: *Nature* 55 (1424) (1897), S. 347ff. <http://dx.doi.org/10.1038/055347a0>. – DOI 10.1038/055347a0



FIGURES

A.1 Energy distribution for various Objects in the USCT



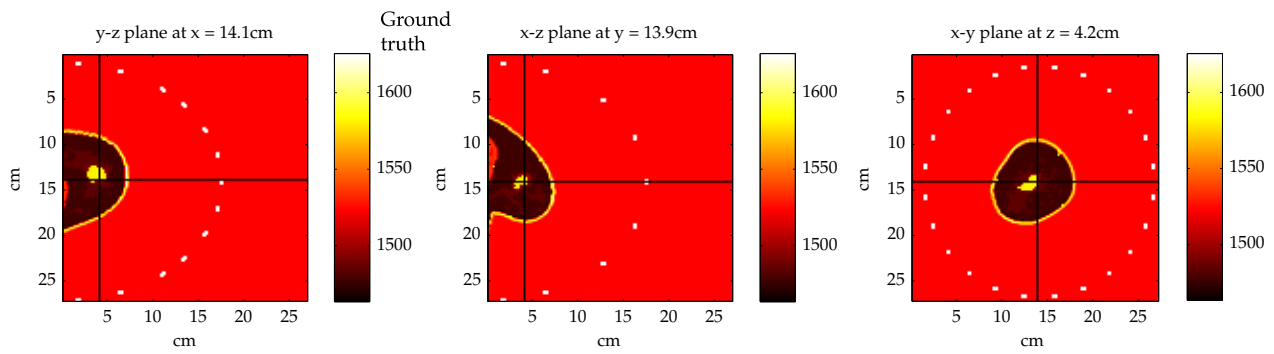


Figure A.1: Sounds speeds of the used breast model inside the USCT

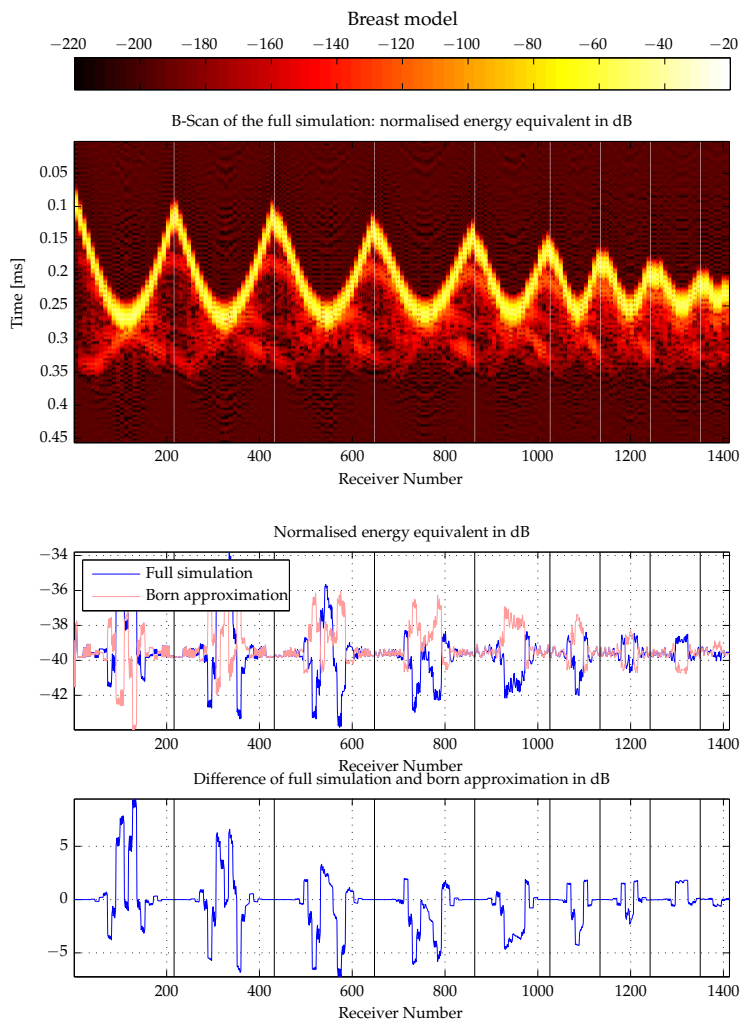
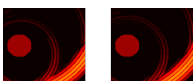


Figure A.2: Breast model with enhance speed of sound for cancer



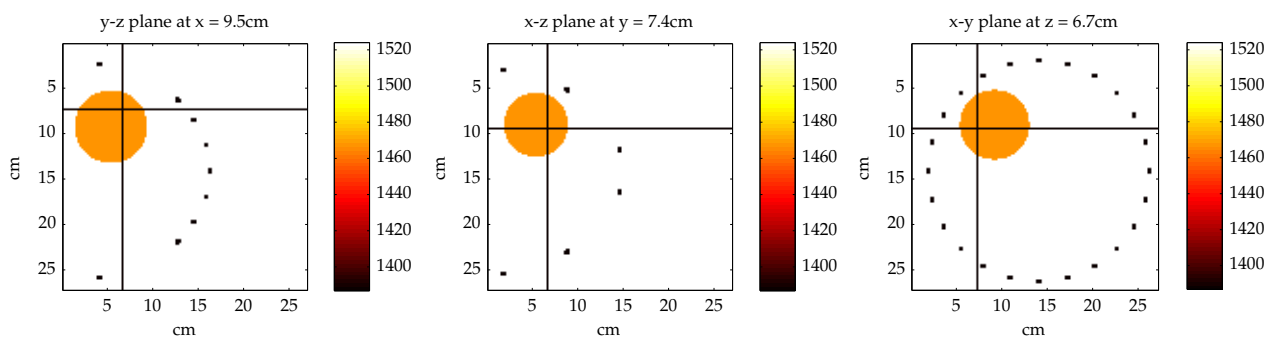


Figure A.3: Sounds speeds of the used sphere inside the USCT

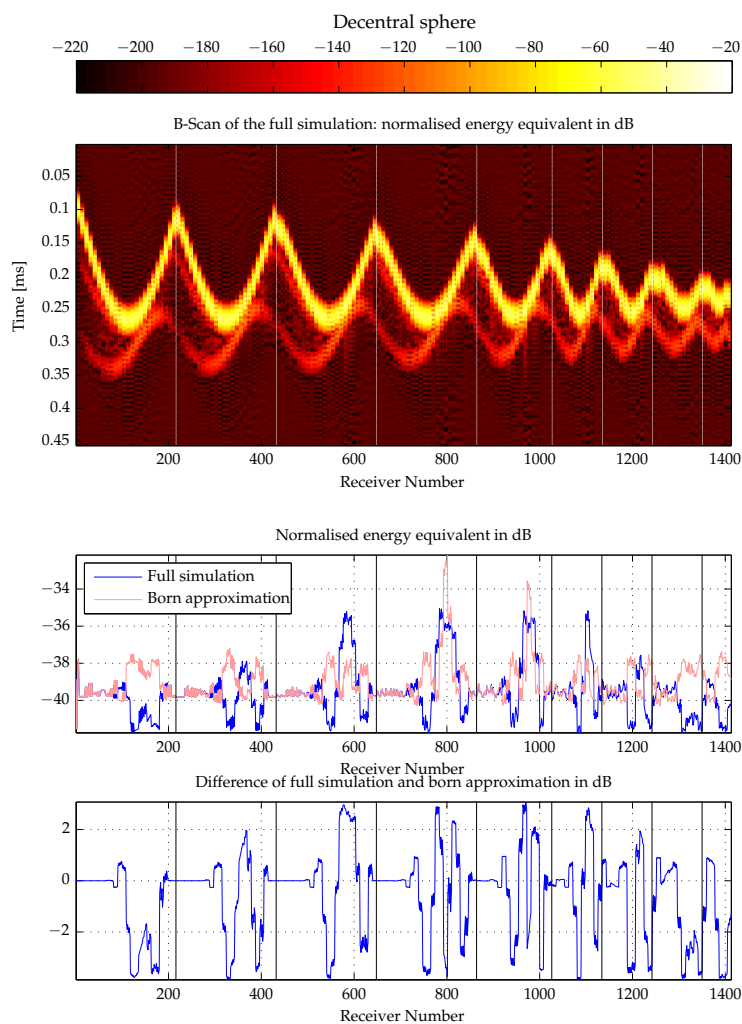
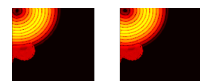


Figure A.4: Decentrally placed sphere



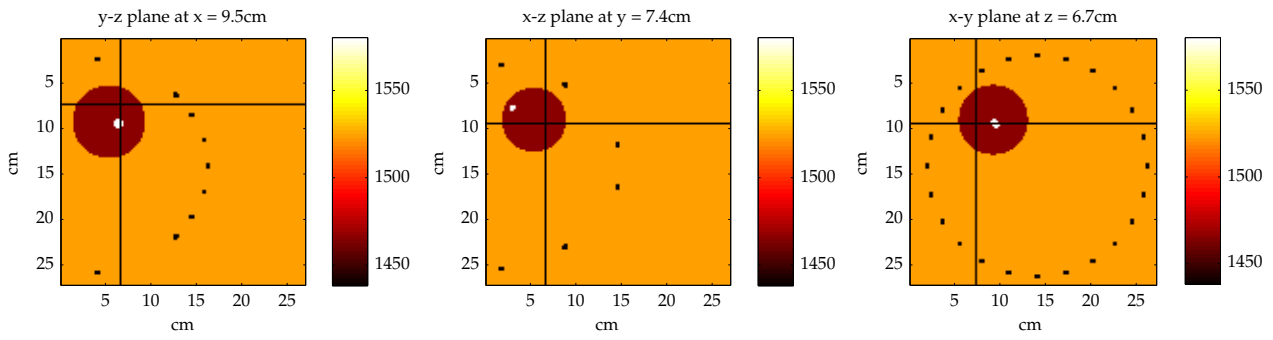


Figure A.5: Sounds speeds of the used sphere inside the USCT

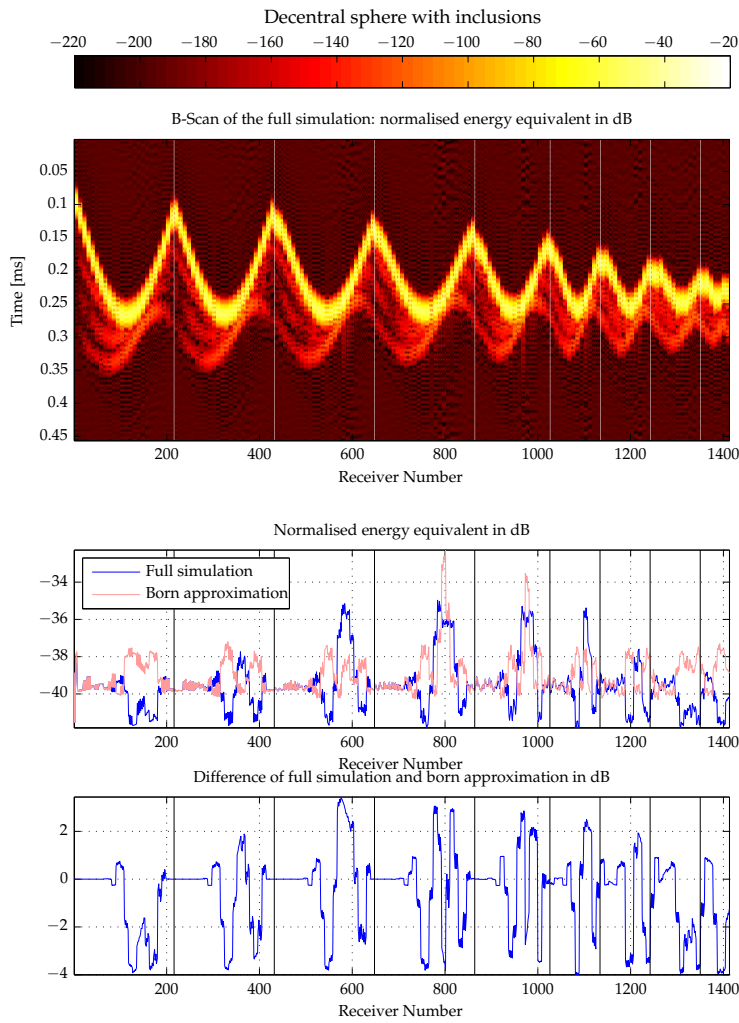
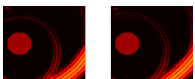


Figure A.6: Decentrally placed sphere with inclusions



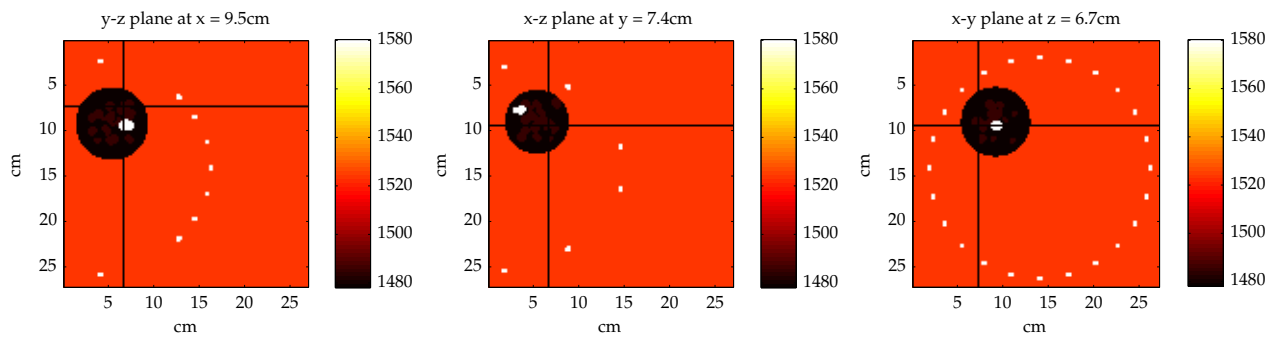


Figure A.7: Sounds speeds of the used breast model inside the USCT

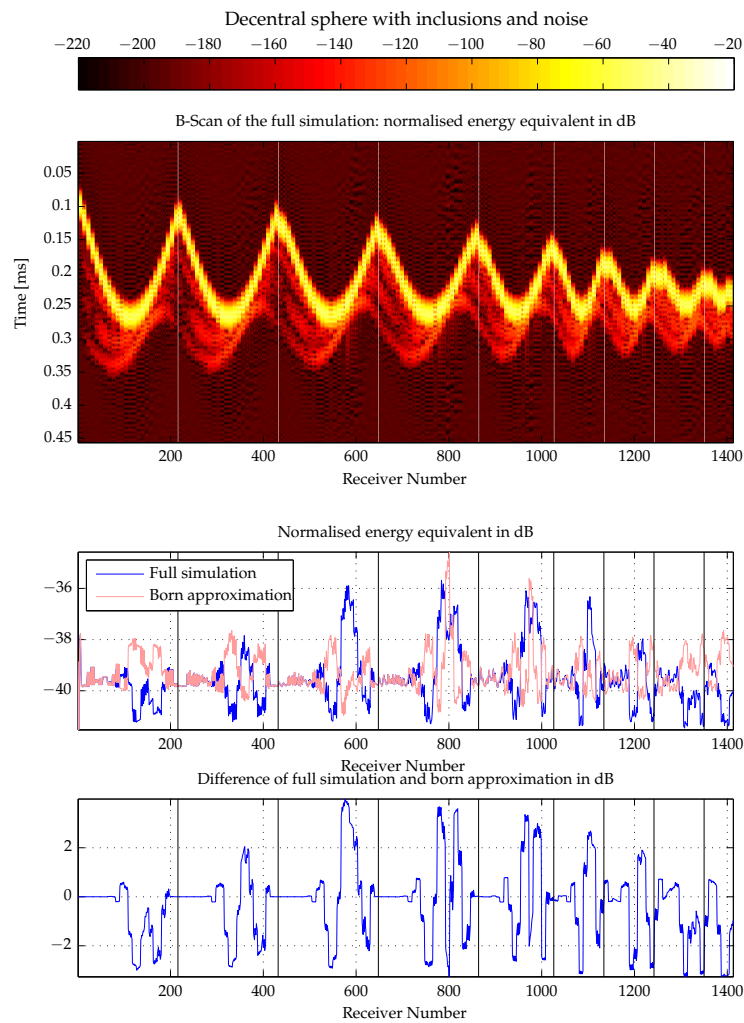


Figure A.8: Decentrally placed sphere with inclusions and noise



A.2 Additional A-scans

The amplitude scans (A-scans) on the following pages refer to the objects shown on the previous pages.

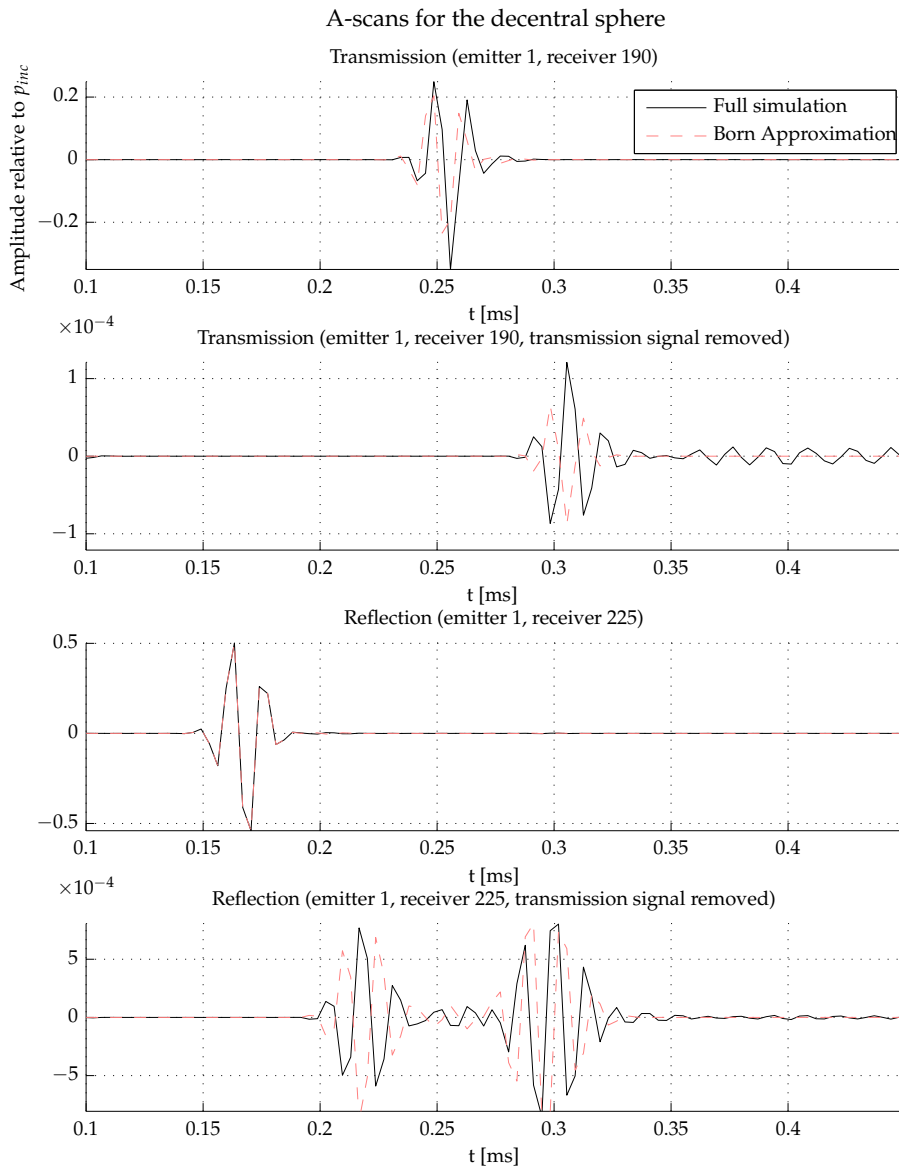
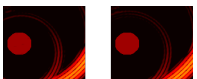


Figure A.9: A-scans for decentrally placed sphere



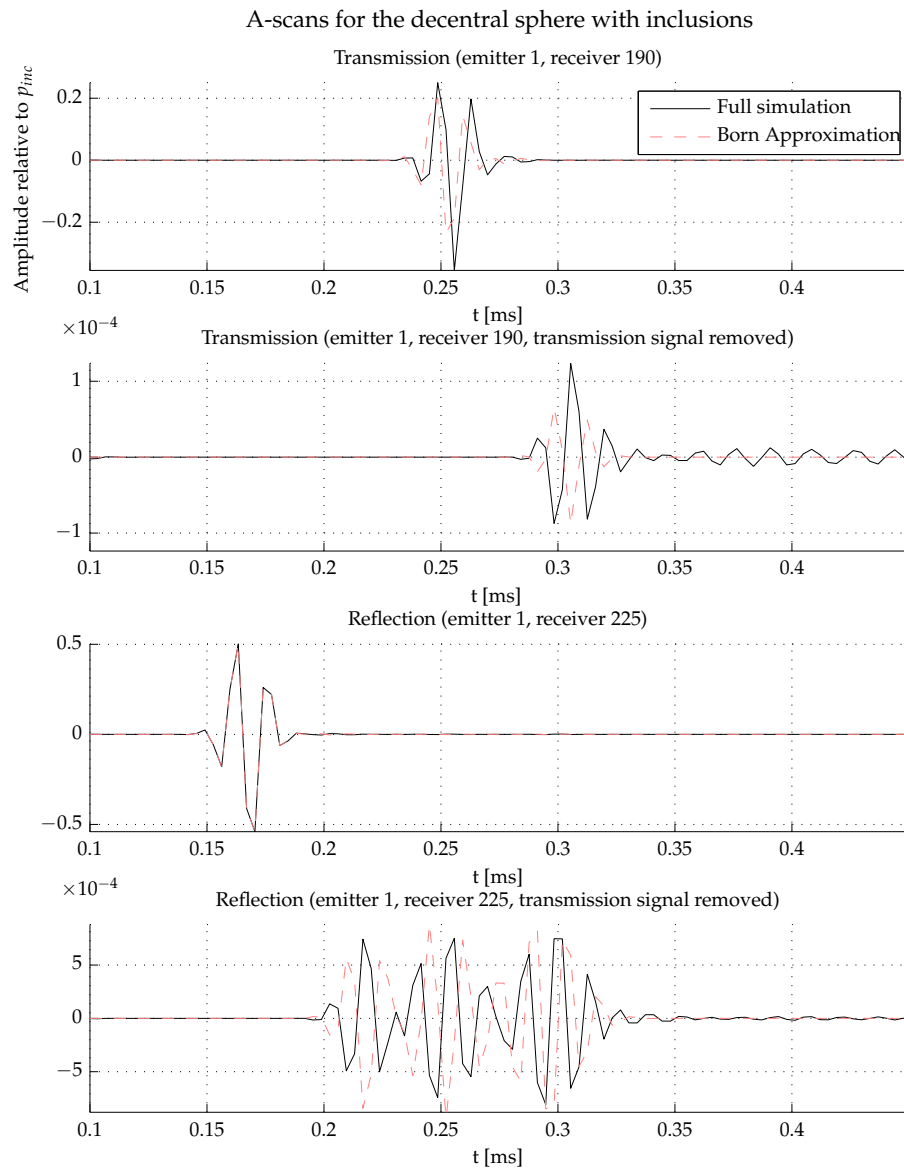


Figure A.10: A-scans for decentrally placed sphere with inclusions



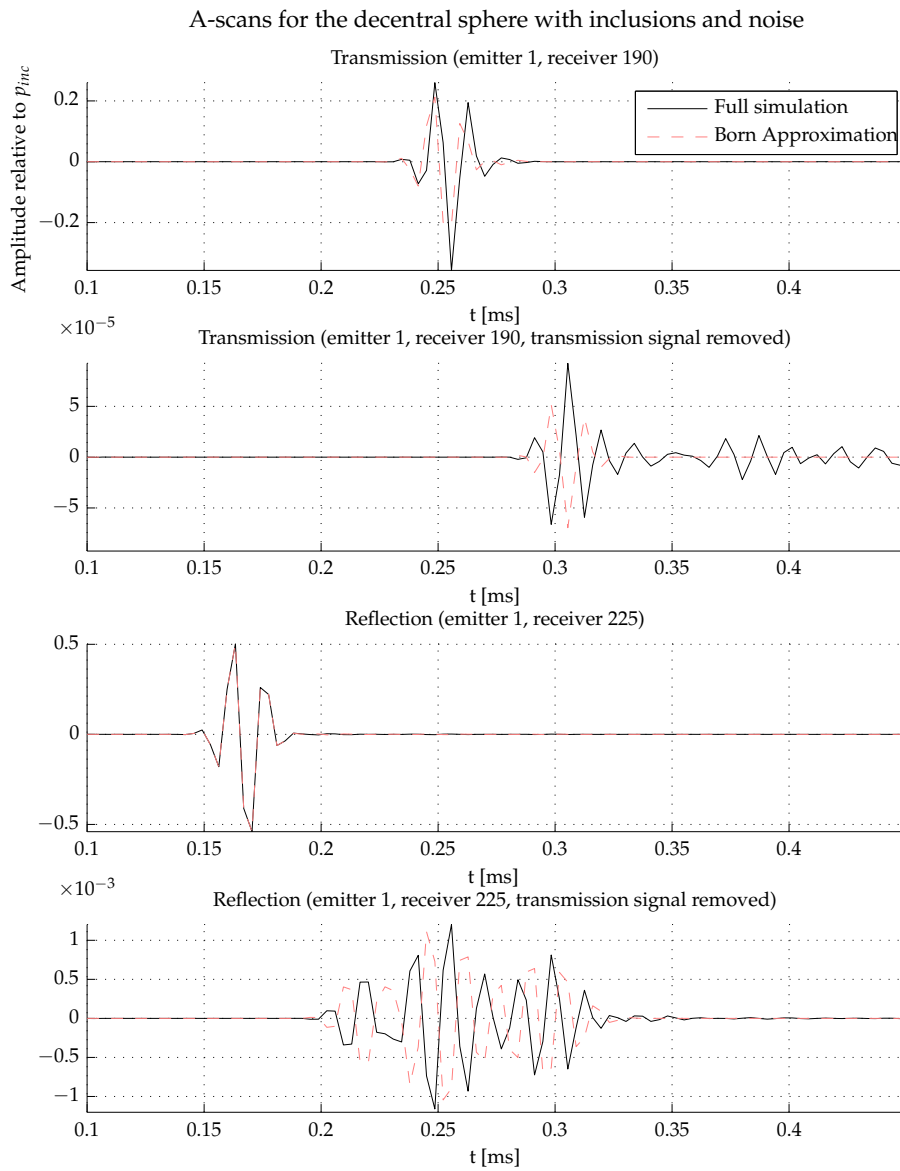
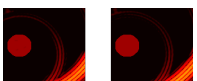


Figure A.11: A-scans for the decentrally placed sphere with inclusions and noise



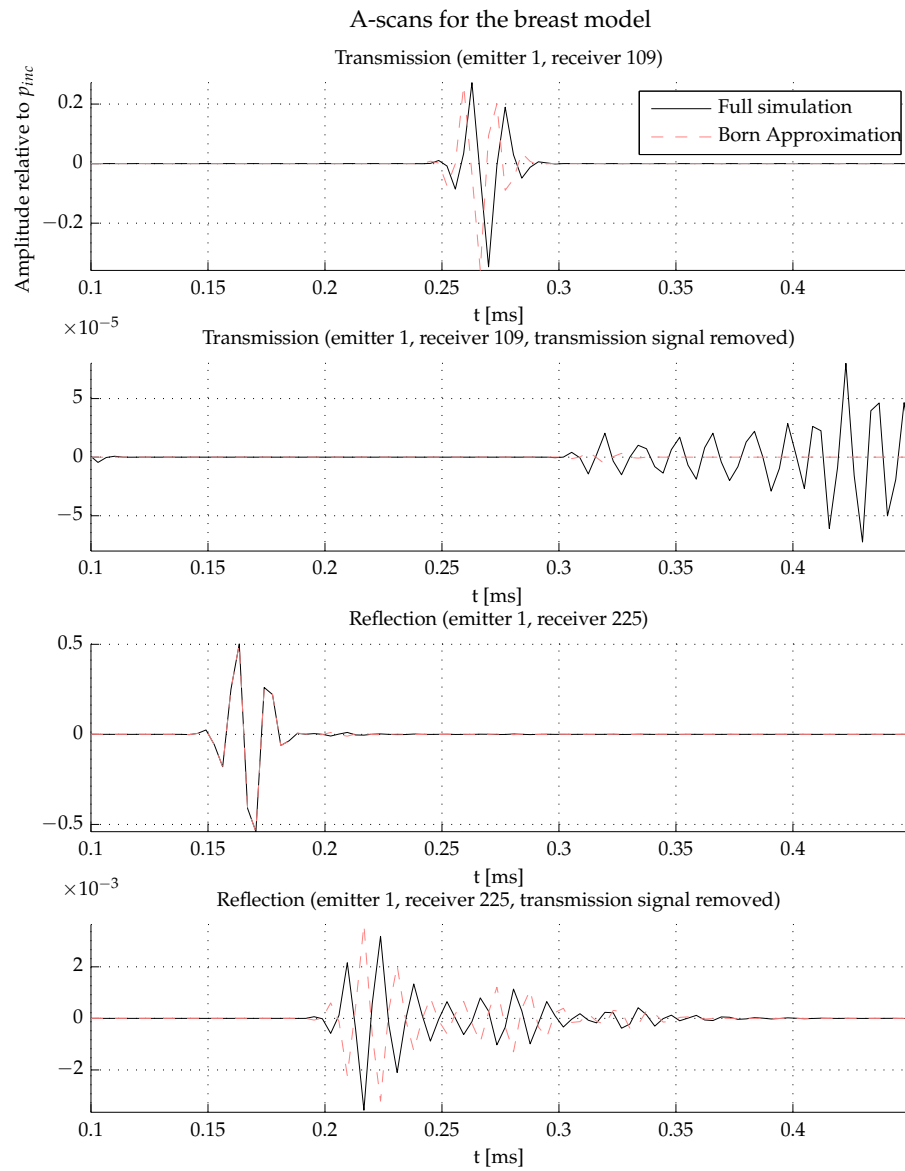


Figure A.12: A-scans for breast model with enhance speed of sound for cancer



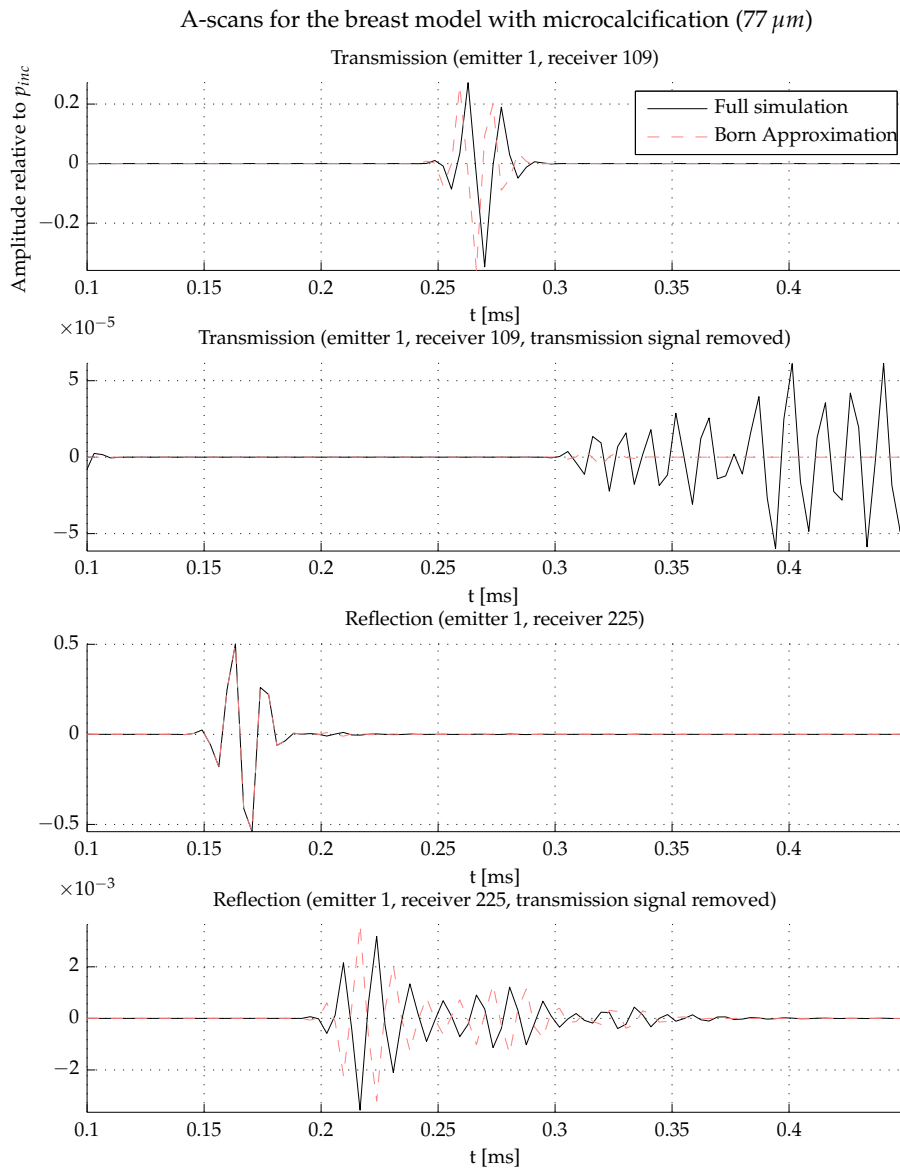
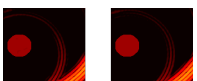


Figure A.13: A-scans for the breast model with a small microcalcification



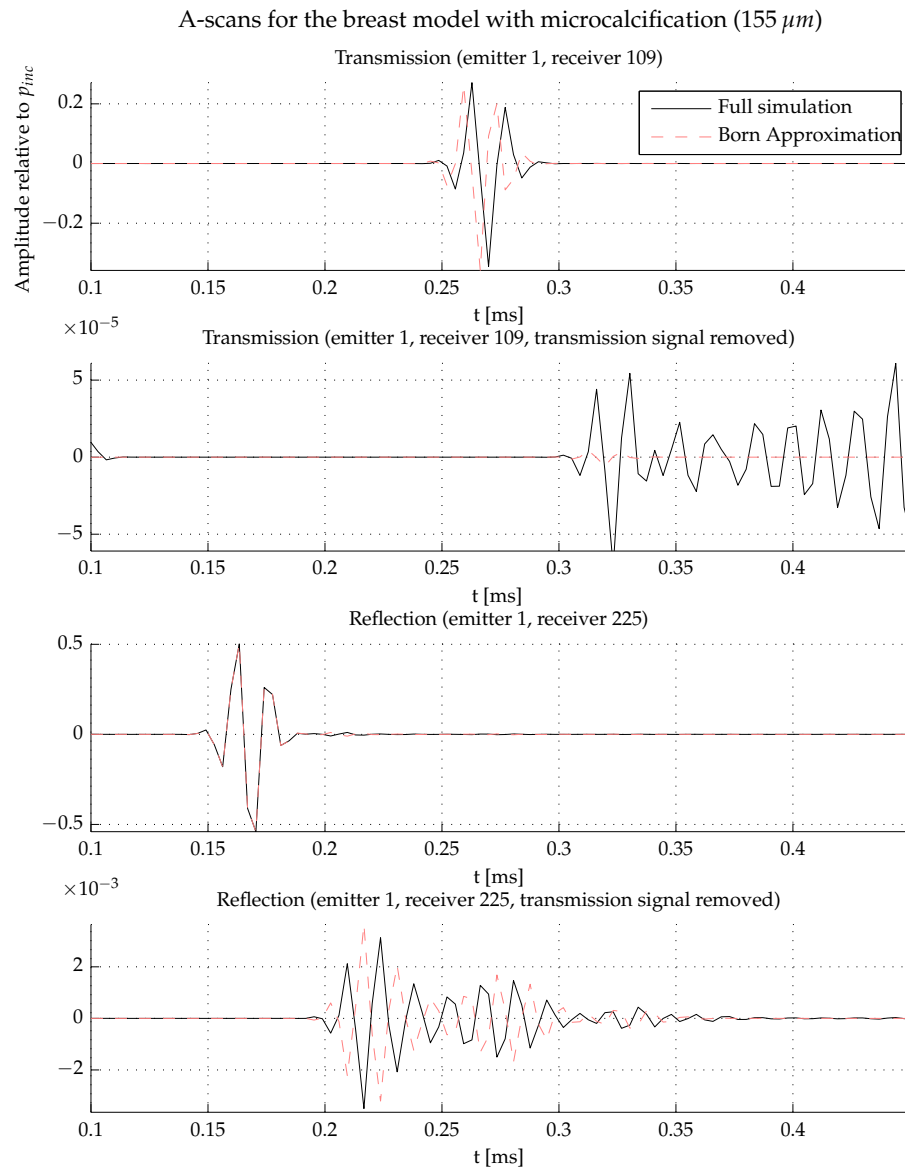


Figure A.14: A-scans for the breast model with a large microcalcification



A.3 Plane wave scattering

I therefore ran four more (numeric) simulations of a plane wave scattered on the sphere. Two with the sphere having a higher speed of sound (1703 m/s) than the surrounding medium (1524 m/s), two with a lower speed of sound (1391 m/s).

Figure A.15 shows a plane wave scattered on a sphere with higher speed of sound compared to the embedding. Top is the exact solution and bottom is the Born approximated solution. It can be seen that the Born approximation does not correctly model the speed of sound. Figure A.16 shows a plane wave scattered on a sphere with lower speed of sound compared to the embedding. Top is the exact solution and bottom is the Born approximated solution. It can be seen that the Born approximation does not correctly model the speed of sound.

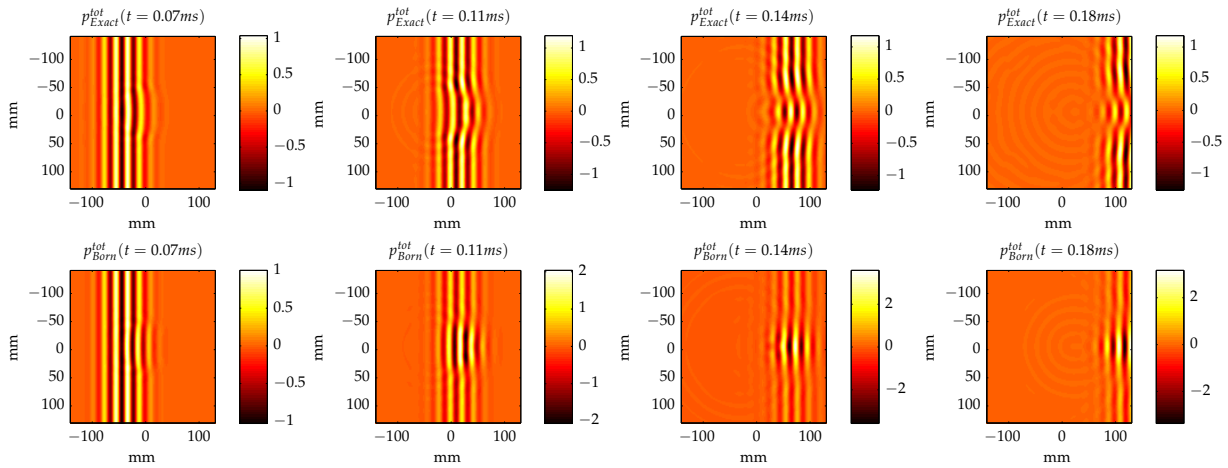


Figure A.15: Comparison of plane wave scattered on a sphere within Born approximation (bottom) and the exact solution (top) for selected time steps

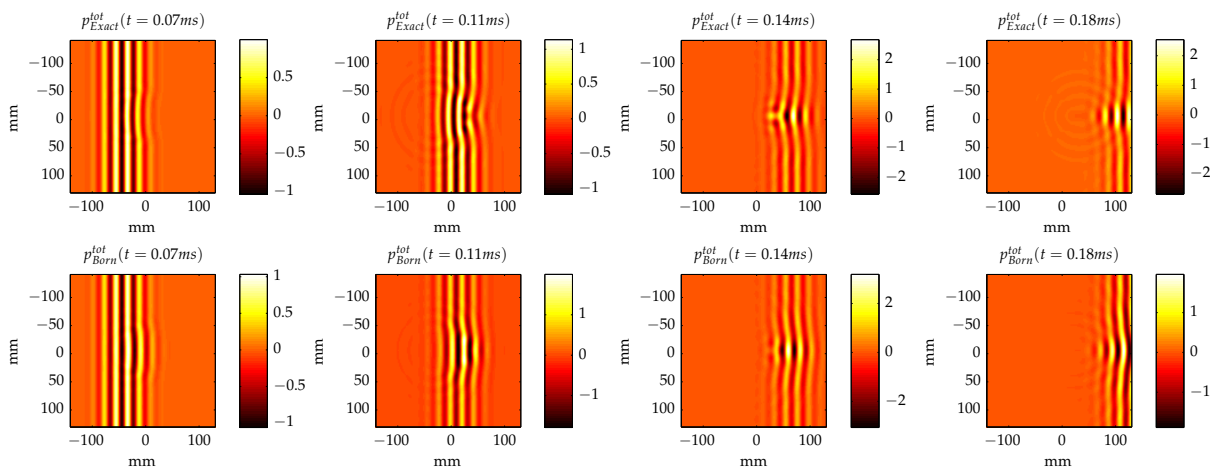
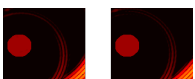


Figure A.16: Comparison of a plane wave scattered on a sphere within Born approximation (bottom) and the exact solution (top) for selected time steps



A.4 Directivity patterns of the surface source

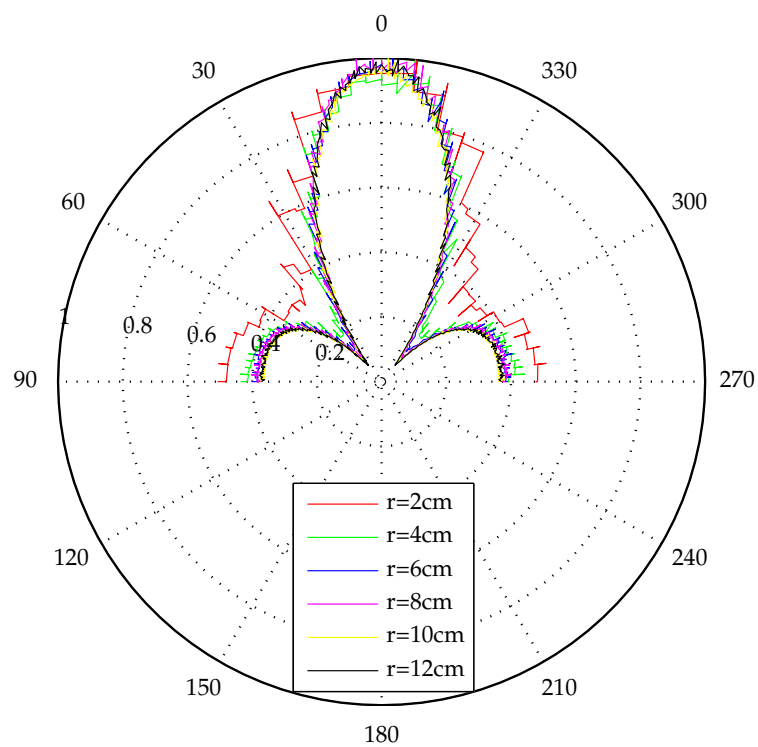


Figure A.17: Directivity pattern of the surface source in section 6.4 in various distances from the source. Each plot was normalised to one.



A.5 Technical Drawings

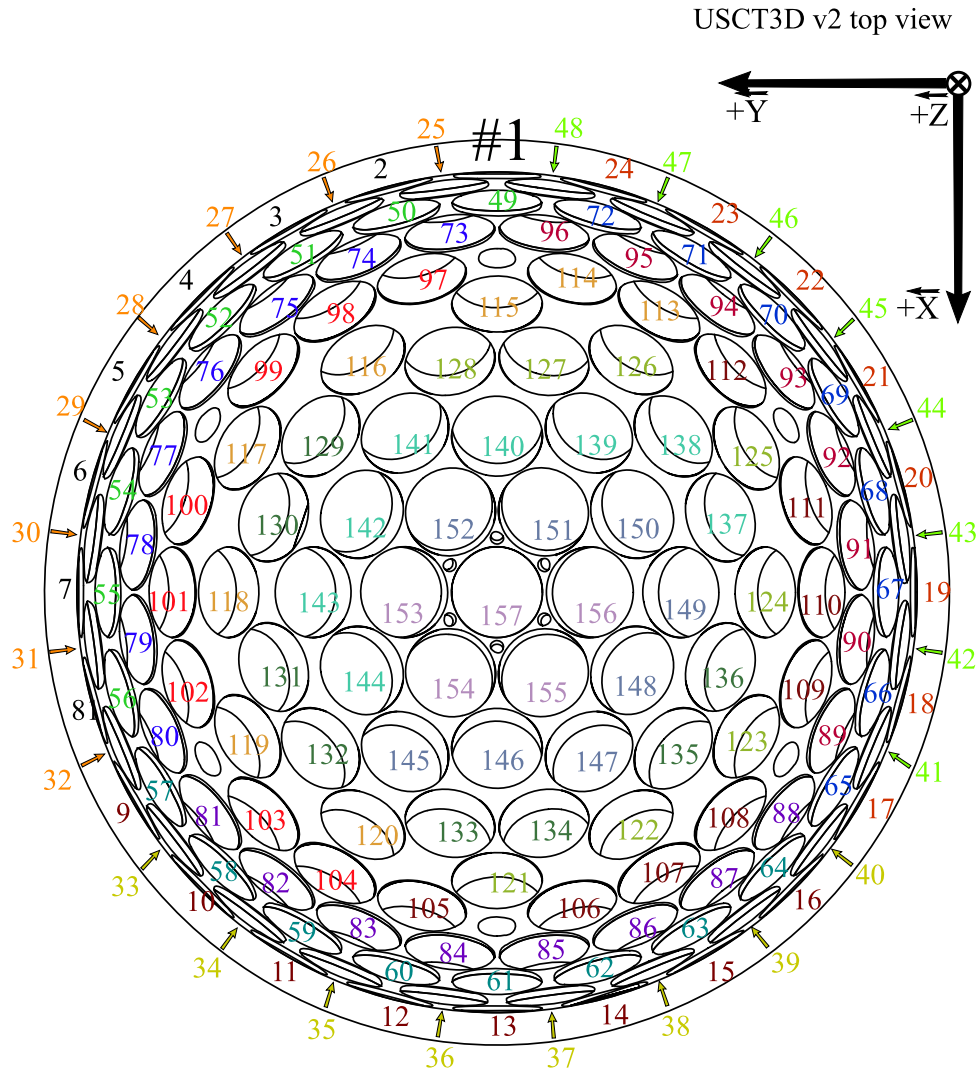
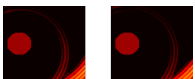


Figure A.18: A map of the USCT. Numbers indicate Transducer Array System (TAS) head numbers.



DERIVATIONS AND TOOLS

B.1 The spatially averaged Green's function

For $\mathbf{r} = 0$ this integral can be solved straightforward, while for $\mathbf{r} \neq 0$ the following derivation is necessary. Using the symmetry of the Green's function, we have freedom to choose \mathbf{r} parallel to the z -Axis. In spherical coordinates we then define

$$\begin{aligned} Q &= \|\mathbf{r} + \mathbf{r}'\| \\ &= \sqrt{r^2 + r'^2 + 2rr' \cos \vartheta'} \\ \Rightarrow \frac{dQ}{d\vartheta'} &= -rr' \sin \vartheta' \frac{1}{Q}. \end{aligned}$$

Using (B.1) in (3.17), we get

$$[\hat{G}(\mathbf{x})] = -\frac{3}{8\pi R^3} \int_{r'=0}^{r'=R} \int_{Q=r+r'}^{Q=r-r'} e^{-ikQ} \frac{r'}{r} dQ dr', \quad (\text{B.1})$$

which can be integrated. This leads to two cases for the spatially averaged Green's function:

$$[\hat{G}(\mathbf{x})] = \begin{cases} \frac{3 e^{-ikr}}{4k^3 \pi R^3 r} (\sin(kR) - kR \cos(kR)) & \forall \mathbf{x} \neq \mathbf{0} \\ \frac{3}{4k^2 \pi R^3} \left((1 + ikR)e^{-ikR} - 1 \right) & \forall \mathbf{x} = \mathbf{0}. \end{cases} \quad (\text{B.2})$$



B.2 The Fourier transform

The Fourier transform is an integral coordinate transformation which transforms a periodic function into a basis in which it is described by its frequency components. The Fourier transform has the nice property that derivatives with respect to time can be described by the multiplication of a factor $i\omega$ with the Fourier transform of that function:

$$\frac{\partial}{\partial t} f(\mathbf{x}, t) = \mathcal{F}^{-1} \{ i\omega \mathcal{F} \{ f(\mathbf{x}, t) \} \} \quad (\text{B.3})$$

$$\Leftrightarrow \mathcal{F} \left\{ \frac{\partial}{\partial t} f(\mathbf{x}, t) \right\} = i\omega \hat{f}(\mathbf{x}, \omega) . \quad (\text{B.4})$$

The Fourier transform is defined as

$$\begin{aligned} \hat{f}(\mathbf{x}, \omega) &= \mathcal{F}[f(\mathbf{x}, t)] \\ &= \int_{-\infty}^{\infty} f(\mathbf{x}, t) e^{-i\omega t} dt \end{aligned} \quad (\text{B.5})$$

and the inversion as

$$\begin{aligned} f(\mathbf{x}, t) &= \mathcal{F}^{-1}[\hat{f}(\mathbf{x}, \omega)] \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\mathbf{x}, \omega) e^{i\omega t} dt . \end{aligned} \quad (\text{B.6})$$

The $\hat{}$ indicates the Fourier transformed function with respect to time. Since this includes the frequency dependence, its dependence is only explicitly shown in the definition above and omitted elsewhere.

The shown examples apply the Fourier transform on the time, thereby transforming the function to frequency domain. This Fourier transform is also referred to as the temporal Fourier transform. The spatial Fourier transform applies to all spatial coordinates used. For this the 2D or 3D Fourier transform has to be used.

B.3 Energy in a pressure wave

The energy of a pressure wave can be computed like

$$E = 1/2 x_p^2 \frac{\omega k}{\kappa} t \cdot A, \quad (\text{B.7})$$

with x_p the particle displacement, ω the circular frequency, k the wavenumber, κ the compressibility, t the time and A the surface on which the wave energy is measured.



The derivation is simple:

$$\text{Particle movement: } s(x, t) = x_p \sin(kx - \omega t) \quad (\text{B.8})$$

$$\text{Particle velocity: } v(x = 0, t) = \frac{\partial s}{\partial t} \quad (\text{B.9})$$

$$= -A\omega \cos(\omega t) \quad (\text{B.10})$$

$$\text{Pressure difference caused by: } \Delta p = -\frac{1}{\kappa} x \frac{\Delta V}{V} \quad (\text{B.11})$$

$$= -\frac{1}{\kappa} \frac{\partial s}{\partial x} \quad (\text{B.12})$$

$$= -\frac{1}{\kappa} x_p k \cos(\omega t) \quad (\text{B.13})$$

$$\text{Power: } P(t) = vF = v\Delta p A \quad (\text{B.14})$$

$$= x_p^2 \frac{\omega k}{\kappa} \cos^2(\omega t) A \quad (\text{B.15})$$

$$\text{Averaged over one wavelength: } P = \frac{1}{2} x_p^2 \frac{\omega k}{\kappa} A \quad (\text{B.16})$$

$$\text{Energy: } E = Pt \quad (\text{B.17})$$

$$= \frac{1}{2} x_p^2 \frac{\omega k}{\kappa} t A \quad (\text{B.18})$$

Therefore image the squared pressure amplitudes is directly proportional to the energy, because all other parameters remain constant. Only the amplitude which changes over time and space gives relevant information.



Scan barcode to see
movie on youtube



SOURCE CODE

C.1 mpain – the Matlab interface to PAIN

```
1 if use_parallel_infrastructure
2     % distribute
3     for k = 2:nt/2;
4         jid(k) = mpain_async (@fsk_fd_remote, geo, freq(k), nfft);
5     end
6     % collect
7     for k = 2:nt/2;
8         [ p_inc_ft p_tot_ft ] = mpain_wait(jid(k));
9     end
10 else
11     % or run locally
12     for k = 2:nt/2;
13         [ p_inc_ft p_tot_ft ] = fsk_fd_remote (geo, freq(k), nfft);
14     end
15 end
```

Listing C.1: Practical example of parallelising code using mpain. The function `fsk_fd_remote` is either called remotely or locally. In the remote case two loops are used: one for distributing the tasks and a second one for collecting their output.

