NTNU – Trondheim
Norwegian University of
Science and Technology

A framework for the simulation and validation of acoustic fields from medical ultrasound transducers

Ole Bakstad

Master of Science in Communication Technology
Submission date: June 2012
Supervisor: Ilangko Balasingham, IET
Co-supervisor: Lasse Løvstakken, ISB
Hans Torp, ISB

Norwegian University of Science and Technology
Department of Electronics and Telecommunications

A framework for the simulation and validation of acoustic fields from medical ultrasound transducers

Ole Bakstad

June 15, 2012

Master thesis TTT4920
NTNU – Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Electronics and Telecommunications

Abstract

The unified simulation framework for medical ultrasound, FieldSim, currently supports linear and non-linear simulations by using Field II and Abersim, respectively. In this thesis the quasi-linear simulation tool, Propose, is incorporated in FieldSim and verified. The implementation uses Field II to generate the initial pressure propagated by Propose. It is shown to produce satisfactory results when compared to standalone Propose, Field II and Abersim for both the fundamental and second harmonic. The results are also verified in the water tank. The running time is found to be slower than standalone Propose, but still substantially quicker than Abersim for non-linear simulations. By combining core features of the FieldSim framework and Propose new features are presented. It is now possible to easily simulate the second harmonic from a transducer with a measured impulse response and arbitrary excitation pulse using Propose in minutes, compared to hours with Abersim. By rotating the initial field steering can now be achieved in Propose in few lines of MATLAB code.

A link between a research scanner and the FieldSim framework is presented. When finalized a FieldSim simulation can be converted to a file read by PyTexo making it possible to use the exact same setup for both simulations and measurements in the water tank. The current implementation supports single ultrasound beams and B-mode with fixed focus.

Sammendrag

Simuleringsrammeverket for medisinsk ultralyd, FieldSim, støtter i dag lineære og ulineære simuleringer ved bruk av henholdsvis Field II og Abersim. I denne avhandlingen er det kvasi-lineære simuleringsverktøyet, Propose, integrert i FieldSim og verifisert. Implementasjonen bruker Field II for å generere initialtrykket som propageres av Propose. Det vises at implementasjonen gir tilfredsstillende resultater sammenlignet med originale Propse, Field II og Abersim. Resultatene er også verifisert med målinger i vanntank. Kjøretiden er vist å være dårligere enn originale Propose, men vesentlig bedre enn Abersim for ulineære simuleringer. Ved å kombinere kjernefunksjonene i FieldSim og Propose nye muligheter er presentert. Det er nå mulig å simulere den andreharmoniske fra en transduser med målt impulsrespons og en arbitrær eksitasjonspuls i løpet av minutter med Propose mot timer for Abersim. Ved å rotere initialtrykket kan nå styring brukes med Propose.

En link mellom en forskning scanner og FieldSim-rammeverket er presentert. Når linken er ferdigutviklet vil en FieldSim-simulering kunne bli konvertert til en fil som kan leses av PyTexo. Dette vil gjøre det mulig å bruke det samme oppsettet i simuleringer som i målinger i vanntank. Dagens implementasjon støtter enkle ultralydstråler og B-mode med fast fokus.

Contents

List of Figures

Preface

With this thesis my five years at NTNU comes to and end. It has been five great and interesting years that I will remember for the rest of my life. A special thanks to everybody whom I've lived with at Singsaker Studenterhjem for making the stay in Trondheim truly great.

During the work on this thesis many people have helped me out in varying ways. I would like to thank my supervisor, Lasse Løvstakken, for great advice and a positive attitude from the day I started as a summer intern at ISB[1] to the very end of my master thesis. Fabrice Prieur helping out with questions regarding the theory and implementation of Propose. Bastien Denaire for helping out in general, but specially with FieldSim and PyTexo. Øyvind Krøvel-Velle Standal for helping me with measurements in the water tank. Thor Andreas Tangen for helping out with questions related to Texo. Hans Torp for insightful discussions about Propose. Mats Lindh, Espen Øybo and Ida Liv Rognstad Trangsrud for reading through the thesis with a critical eye.

Sist men ikke minst vil jeg takke mine foreldre, Per og Randi Bakstad, for upåklagelig støtte gjennom alle disse årene. Dere kan kanskje ikke hjelpe til med de faglige problemene, men dere er alltid der. Takk.

[1]Department of circulation and medical imaging, NTNU

Chapter 1

Introduction

In the field of medical ultrasound imaging the acoustic output must be kept below the safety limits set by regulatory institutions such as the Food and Drug Administration (FDA) in the USA. This is ensured by measuring several parameters related to the acoustic output in a water tank using a hydrophone to collect data. Further, when designing new transducers and new imaging techniques (pulsing schemes and scan sequences), researchers commonly start out using a set of simulation tools. Different simulation tools include different and often simplified modelling schemes of transducers, and different implementations of linear and non-linear wave propagation. Hence, varying trade-offs between simulation time and accuracy exists between them, limiting the usefulness of each individual simulation tool. Each tool often has its own setup that may can vary from tool to tool. This requires the user to learn how particular simulation tools are configured making it necessary to remember multiple configurations even though the same scenario is to be simulated.

The motivation for this thesis is the development of a unified simulation framework for medical ultrasound, FieldSim, which offers a standardized simulation setup used by the underlying simulators. FieldSim offers a high level of abstraction, but at the same time gives the user the power to change every detail of a simulation setup. By having a standardized way of setting up simulations it becomes easy to use the same setup with different simulation tools.

Instead of thinking about probes and materials as numbers specifying the number of elements, pitch sizes, lens focus, the speed of sound or frequency dependent attenuation constants, the user is abstracted away from this and handed objects that represent actual probes and materials. If a specific probe is to be used in a simulation, the probe is chosen by referring the actual name of the probe and the specifics are automatically initialized without the user having to worry about using the right values that specifies the probe. If needed, all the details are still available and can be changed to fit the need of the user.

The goals for this thesis can be broken down into three parts:

- Implement support for the simulation tool Propose in FieldSim. Propose is substantially quicker than Abersim for non-linear simulations, but also more inaccurate.
- Develop a link between the research scanner system and simulation framework, so that acoustic measurements can be made using the identical setup as in simulations.
- Validate the non-linear simulation framework in a watertank, including both Abersim and Propose, in terms of predicting acoustic parameters related to mechanical and heating effects.

Reading through this thesis Propose, FieldSim and Abersim will be frequently referred to. In order to separate the standalone version of Propose from the Propose incorporated in FieldSim a convention has to be made. *Whenever talking about FieldSim in the context of comparing against Propose, it is the Propose simulator implemented in FieldSim that is being compared to the standalone version of Propose*, unless otherwise stated. When refering to Abersim in this thesis it is referred to the Abersim simulator implemented in FieldSim.

The structure of the thesis is as follows: Background material and theory is given in Chapter 2, in Chapter 3 the implementation of Propose in FieldSim is presented. Test cases for verifying the implementation against Field II and Abersim is given and the running time is discussed and measured. Chapter 4 introduces the laboratory setup and a new link between a research scanner and FieldSim is presented. Measurements are carried out in a water tank and compared to Propose and Abersim. In Chapter 5 the results from previous chapters are presented. In Chapter 6 the results are discussed and the report is concluded in Chapter 7. Ideas and further work is listed in Chapter 8.

Chapter 2

Background

This chapter gives an introduction to the FieldSim framework and a brief overview of the theory behind Field II, Abersim and Propose.

2.1 FieldSim

FieldSim is a simulation framework for medical ultrasound developed at the Department of Circulation and Medical Imaging at NTNU. When the project started the aim was to make a unified simulation framework. It should be easily configurable and contain default setups for scanner configurations, post processing and probes typical for modern 3D ultrasound scanners. This setup should then be employable across a range of different simulation tools.

When using multiple standalone tools the process of setting up a scanner configuration for the individual tools can be both time consuming and error prone. By introducing a standardized way to set up a scanner configuration and running simulations the learning curve goes down and the configuration error should go down as well because the scanner configuration is only done once.

FieldSim is written in object-oriented MATLAB since many of the existing tools already interfaces well with MATLAB. By using an object-oriented design a highly modular architecture makes it easy to extend existing features and/or implement new features. However, MATLAB version 2009b or higher is required because the implementation uses some of the new features released in this version. The code is planned to be released as open source.

2.1.1 Overview

An architectural overview of FieldSim is shown in Figure 2.1. From the figure it can be seen that the system mainly consist of six blocks: front-end, scan definition, scan geometry, the biological setup, the simulator and the post-processing and display.

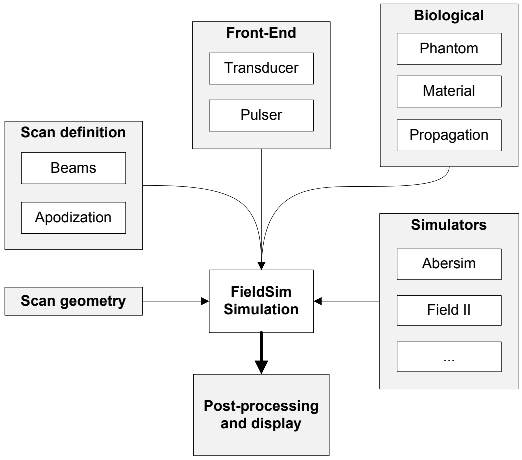


Figure 2.1: Overview of the FieldSim3 framework.

Front-end – When the user selects a probe the transducer geometry, scan shape, center frequency and bandwidth and more are automatically set up. This is imported from an XML file that contains the probe definitions. Probes can be assigned applications as well. For example a probe can have different settings if used to image the brain or the heart. By changing the probe application the probe parameters are automatically changed accordingly.

The probe specification also contains information about the impulse response. This can either be set as a formula or be a measured impulse response. Some probe setups from both GE Healthcare and Ultrasonix are included, but these contain proprietary information and have to be shipped separately.

To add a probe all the user has to do is to add a probe specification to the XML file. This way it is very easy to study probe design.

Scan definition – The scan definition contains different information dependent on what mode is selected. In the current version of FieldSim the two modes supported are B-mode and beamprofile. For a B-mode simulation the scan definition has information about the scan shape, transmit and receive beams and their layout, number of beams used, apodization on transmit and receive, multiple line acquisition (MLA), number of frames, packets and frame rate.

A beam profile however requires less information, in this case the scan definition contains information about the beams and apodization for transmit and receive.

Also here does the probe definition play a role. In the probe definition it is possible to set wanted f-number for the beams on transmit and receive and wanted scan shape.

Scan geometry – In FieldSim three types of scan geometries are supported; linear, sector and curvilinear in both 2D and 3D. The scan geometry can be thought of as the field of view for the simulation. The scan geometry can be limited by the simulator used. Abersim and Propose, for example, only support linear (rectangular) scan geometry.

Biological setup – To simulate biological effects FieldSim has mainly three ways of defining a medium: phantoms, materials and aberrators. Phantoms can be designed by point scatterers, both stationary and moving. Some phantoms are already implemented, but users are also encouraged to make their own.

The implementation of materials is based on the materials found in Abersim. Here the speed of sound, attenuation, coefficient of non-linearity, temperature, etc. is defined. The parameters are dynamic and can for example be dependent on temperature. Some of the materials already implemented include fat, liver, muscle, blood and some generic cases with $c = 1540$m/s, 0.3dB/Mhz/cm and 0.5dB/Mhz/cm constant frequency dependent attenuation.

Aberrations can be introduced in both Field II and Abersim, but does not yet have a standardized setup. In Abersim aberrations is implemented by using delay screens with random phase delays, while the aberrator for Field II is a collection of scripts which introduces phase and amplitude delays in their own way. Propose can take advantage of the Field II aberrator when creating the initial pressure.

Simulators – As of today (June 2012) FieldSim comes with support for the two simulators Field II and Abersim. In this paper Propose is incorporated and verified. The implementation and verification is discussed in detail in Chapter 3. Work is also being done on implementing support for the Cole simulator.

Post-processing and display – In post-processing several filters can be applied to simulation results. Depending on the mode used for a simulation, default filters are now set up. For example when simulating a beam profile the filters interpolation, depth normalize, power and log10, in that order are used. Future plans include a more robust and generalized post-processing library that can be used on generic data as well as data inside FieldSim.

Results can be displayed by using the built-in plot function in MATLAB. This works with most objects in FieldSim like simulation data, scan definition, phantoms and gives a consistent way for visualization. When using the plot function on simulation data the post-processing filters are automatically applied to generate better looking result, this can however be turned off.

2.2 Field II

Field II is a program for simulating ultrasound tranducer fields and ultrasound imaging using linear acoustics. The program consists of a C program and a number of Matlab m-functions that calls this program. All calculations are performed by the C program, and all data is kept by the C program. Field II is released as citationware which requires the citation of [3] and [4]. The details can be found at `http://server.oersted.dtu.dk/personal/jaj/field/?copyright.html`.

This section gives a brief introduction to the theory behind Field II. Only the relevant theory is presented as Field II also include features like pulse-echo imaging and flow estimation. The material presented in this section is based on [1].

2.2.1 Model

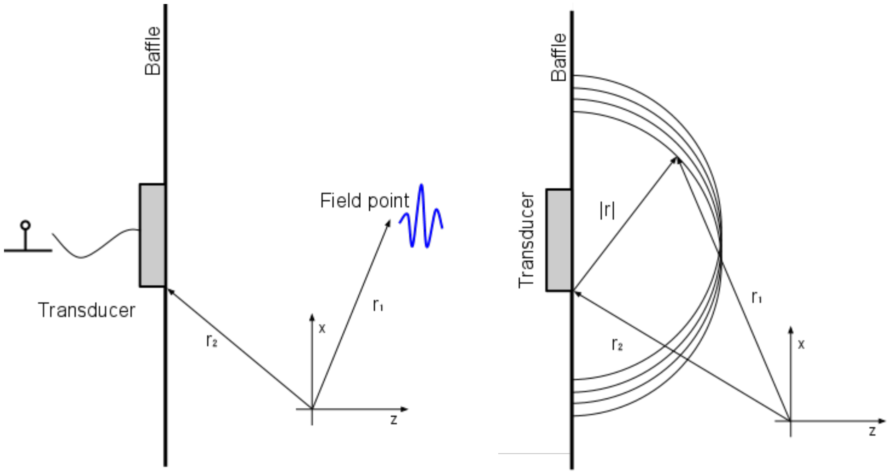
Field II is based on a linear acoustic model as shown in Figure 2.2a. The transducer is assumed to be mounted in an infinite rigid baffle at position $\mathbf{r}_2$. The propagation is assumed to be through a homogeneous medium with a constant speed of sound c and density ρ. At the point $\mathbf{r}_1$ the acoustic pressure is measured from a small point hydrophone. If the transducer is excited by a delta function the measured pressure at $\mathbf{r}_1$ is the acoustic impulse response for the given system. By moving the hydrophone or the transducer relative to each other a different acoustic impulse response will be measured. This gives rise to the name spatial impulse response since it is dependent on the relative position $(\mathbf{r}_2 - \mathbf{r}_1)$ of the sender and receiver, respectively.

By Huygens' principle the perception of a sound field at a fixed time is that every point on the radiating surface is the origin of a spherical wave as illustrated in Figure 2.2b. Mathematically each sphere is given by

$$p_s(\mathbf{r}_1, t) = \delta\left(t - \frac{|\mathbf{r}_2 - \mathbf{r}_1|}{c}\right) = \delta\left(t - \frac{|\mathbf{r}|}{c}\right). \tag{2.1}$$

To calculate the exact spatial impulse response a triangular aperture is assumed, placed in an infinite rigid baffle on which the velocity orthogonal to the plane is zero everywhere, except the aperture. The pressure at $\mathbf{r}_1$ is then given by the Rayleigh integral [5, 1]

$$p(\mathbf{r}_1, t) = \frac{\rho_0}{2\pi} \int_S \frac{\frac{\partial v_n\left(\mathbf{r}_2, t - |\mathbf{r}|/c\right)}{\partial t}}{|\mathbf{r}|} \,\mathrm{d}S, \tag{2.2}$$



(a) A linear acoustic system.

(b) Illustration of Huygens' principle for a fixed time instance. A spherical wave with a radius of $|\mathbf{r}| = ct$ is radiated from each point on the aperture.

Figure 2.2: Basic description of a ultrasound system. Both figures are based on [1].

where v_n is the velocity at the surface of the aperture. Here it is assumed that the propagation is linear in a loss less homogeneous medium.

By interchanging the integration and the partial derivative and introducing the velocity potential

$$\Psi(\mathbf{r}_1, t) = \int_S \frac{v_n\left(\mathbf{r}_2, t - |\mathbf{r}|/c\right)}{2\pi|\mathbf{r}|} \,\mathrm{d}S, \tag{2.3}$$

the pressure can be expressed as:

$$p(\mathbf{r}_1, t) = \rho_0 \frac{\partial \Psi(\mathbf{r}_1, t)}{\partial t}. \tag{2.4}$$

Introducing a convolution in time with delta function as

$$\Psi(\mathbf{r}_1, t) = \int_S \int_T \frac{v_n\left(\mathbf{r}_2, t_2\right)\delta(t - t_2 - |\mathbf{r}|/c)}{2\pi|\mathbf{r}|} \,\mathrm{d}t_2 \,\mathrm{d}S, \tag{2.5}$$

the excitation pulse can be separated from the transducer geometry. Assuming that the velocity is independent from $\mathbf{r}_2$, that is, the velocity is the same everywhere on the transducer surface:

$$\Psi(\mathbf{r}_1, t) = v_n(t) * \underbrace{\int_S \frac{\delta\left(t - |\mathbf{r}|/c\right)}{2\pi|\mathbf{r}|} \,\mathrm{d}S}_{h}. \tag{2.6}$$

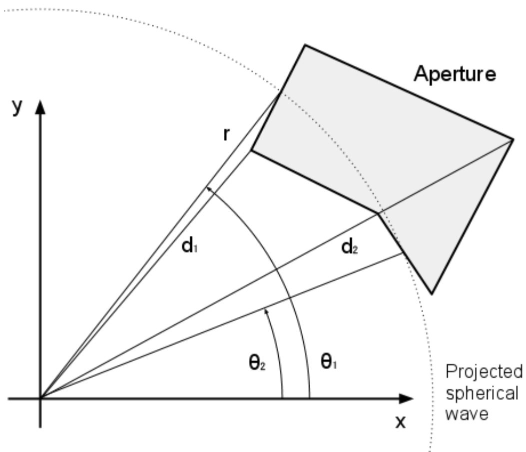


Figure 2.3: Illustration of a spherical wave emitted from (0,0) in the local coordinate system. d_1 and d_2 is the projected distances furthest and closest, respectively, determined by the aperture. θ_1 and θ_2 is the corresponding angles for a given time. The figure is based on [1].

The last term in Equation (2.6) is called the spatial impulse response $h(\mathbf{r}_1, t)$. Using this, Equation (2.3) and (2.4) the pressure can be written as:

$$p(\mathbf{r}_1, t) = \rho_0 \frac{dv_n(t)}{dt} * h(\mathbf{r}_1, t) \tag{2.7}$$

2.2.2 Calculation of spatial impulse responses

Due to the linearity assumed in Field II arbitrary complex apertures can be calculated by sub dividing in smaller sub-apertures and adding the responses. The acoustic reciprocity theorem [6] states "If in an unchanging environment the locations of a small source and a small receiver are interchanged, the received signal will remain the same." That is, the spatial impulse response can be found by emitting a spherical wave at the field point and finding where the wave intersects the aperture as illustrated in Figure 2.3.

In the previous section the spatial impulse response was found by evaluating the Rayleigh integral. By rewriting the Rayleigh integral in polar coordinates the spatial impulse response can be stated as:

$$h(\mathbf{r}_1, t) = \int_{\theta_1}^{\theta_2} \int_{d_1}^{d_2} \frac{\delta\left(t - R/c\right)}{2\pi R} r \, \mathrm{d}r \, \mathrm{d}\theta, \tag{2.8}$$

where r is the radius of the projected circle and $R = |\mathbf{r}| = |\mathbf{r}_2 - \mathbf{r}_1|$ the distance to the aperture from the field point.

By substituting $2RdR = 2rdr$ and $t' = R/c$ the spatial impulse response can be written as [1]:

$$h(\mathbf{r}_1, t) = \frac{c}{2\pi} \int_{\theta_1}^{\theta_2} \int_{t_1}^{t_2} \delta(t - t')\, \mathrm{d}t'\, \mathrm{d}\theta. \tag{2.9}$$

By looking at the spatial impulse response for a given time instance it becomes clear that the contribution along the arc is constant and gives:

$$h(\mathbf{r}_1, t) = \frac{\theta_2 - \theta_1}{2\pi} c. \tag{2.10}$$

Thus, the spatial impulse response can be found by keeping track of the intersections as a function of time, and can be found for a arbitrary transducer geometry when no apodization i used. Solutions for polygons, circular surfaces and circular concave surfaces can be found in [1].

2.2.3 Apodization

To introduce apodization Equation (2.8) is written as [7]

$$h(\mathbf{r}_1, t) = \int_{\theta_1}^{\theta_2} \int_{d_1}^{d_2} a_p(r,t) \frac{\delta\left(t - R/c\right)}{2\pi R} r\, \mathrm{d}r\, \mathrm{d}\theta, \tag{2.11}$$

where $a_p(r,t)$ is the apodization over the aperture. By doing the same substitutions as in Equation (2.9) and noting that the inner integral is a convolution in time with a delta pulse the impulse response becomes [1]:

$$h(\mathbf{r}_1, t) = \frac{c}{2\pi} \int_{\theta_1}^{\theta_2} a_p(t,\theta)\, \mathrm{d}\theta. \tag{2.12}$$

The spatial impulse response can thus be found for a given time instance by numerical integration of the apodization function. Any apodization can therefore by applied to a arbitrary transducer geometry.

2.2.4 Fields from array transducers

Nowadays ultrasound scanners use arrays to generate and receive ultrasound fields. In Field II the spatial impulse response $h_e(\mathbf{r}, t)$ for each element is assumed to be known. Since a linear system is assumed the total response of an array can be found as the sum of N elements:

$$h_a(\mathbf{r}_p, t) = \sum_{i=0}^{N-1} h_e(\mathbf{r}_i, \mathbf{r}_p, t), \tag{2.13}$$

where $\mathbf{r}_i$ denotes the element position and $\mathbf{r}_p$ the field point.

If very small elements and a field point far away is assumed, the spatial impulse response can be approximated as a delta pulse $\delta(t)$ and Equation (2.13) can be written as [1]:

$$h_a(\mathbf{r}_p, t) = \frac{k}{R_p} \sum_{i=0}^{N-1} \delta\left(t - \frac{|\mathbf{r}_i - \mathbf{r}_p|}{c}\right), \tag{2.14}$$

where $R_p = |\mathbf{r}_a - \mathbf{r}_p|$, k is a proportionality constant and $\mathbf{r}_a$ is the position of the array. If the distance between the elements are D, h_a is a train of delta pulses,

$$h_a(\mathbf{r}_p, t) = \frac{k}{R_p} \sum_{i=0}^{N-1} \delta\left(t - \frac{|\mathbf{r}_a - iD\mathbf{r}_e - \mathbf{r}_p|}{c}\right), \tag{2.15}$$

where $\mathbf{r}_e$ is a unit vector in the direction of the elements as illustrated in Figure 2.4.

By looking at the time differences in arrival times the spatial impulse response for the array can be expressed as [1]:

$$h_a(\mathbf{r}_p, t) \approx \frac{k}{R_p} \sum_{i=0}^{N-1} \delta\left(t - \frac{R_p}{c} - i\Delta t\right), \tag{2.16}$$

where $\Delta t = \frac{D \sin\theta}{c}$. Thus, Δt and the shape of the excitation pulse determines whether the signal is added or cancel out for each element. If this difference is exactly one period of a sine wave peaks occur when

$$\frac{n}{f} = \frac{D \sin\theta}{c}.$$

This is what's causing grating lobes to appear. The main lobe is found for $\theta = 0$ and the next maximum can be found for

$$\theta = \arcsin\left(\frac{c}{fD}\right) = \arcsin\left(\frac{\lambda}{D}\right).$$

For the first grating lobe to be outside the imaging area the elements must be spaced less than a wavelength apart. For examples and more detailed explanation see [1].

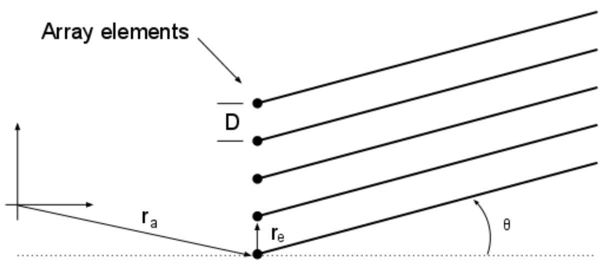


Figure 2.4: Simplified geometry of a linear array (based on [2], Copyright Cambridge University Press)

2.3 Abersim

Abersim is a toolkit for simulating 3D nonlinear acoustic forward wave propagation through attenuating medium. It comes in a pure MATLAB version, a MATLAB + core routines in C compiled via the MATLAB MEX interface, and a stand-alone C version for batch processing [8]. The development of Abersim has been carried out by a group of scientists[1] at Department of Circulation and Medical Imaging, Faculty of Medicine, The Norwegian University of Science and Technology. Abersim is released as open source under the GNU General Public License[2].

This section gives a brief introduction to the underlying theoretical model and implementation in Abersim. The material presented in this section is based on [9] and [10], more detailed information can be found there.

2.3.1 Model

The underlying differential equation for wave propagation in Abersim is the Westervelt equation [9, 11]

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} - \frac{1}{c^2}\frac{\partial^2 \mathcal{L}p}{\partial t^2} = -\frac{\beta_n \kappa}{c^2}\frac{\partial^2 p^2}{\partial t^2}, \qquad c = \frac{1}{\sqrt{\rho\kappa}}, \tag{2.17}$$

where p is the pressure, c is the speed of sound in the medium, $\mathcal{L}$ is the kernel of a convolution operator accounting for attenuation, β_n is the coefficient of nonlinearity, ρ is the mass density and κ is the compressibility.

By introducing retarded time, $\tau = t - z/c$, Equation (2.17) can be written as the directional Westervelt equation [12, 10]

$$\frac{\partial^2 p}{\partial \tau \partial z} = \frac{1}{2}\left(\nabla^2 - g\right) p - \epsilon_t \frac{\partial^2 p}{\partial \tau^2} + \frac{\epsilon_n}{2}\frac{\partial^2 p^2}{\partial \tau^2} + \epsilon \frac{\partial^2 \mathcal{L}p}{\partial \tau^2}, \tag{2.18}$$

where ϵ_t, ϵ_n, ϵ are scaling constants and g accounts for heterogeneities. Integrating with respect to τ gives

$$\frac{\partial p}{\partial z} = \frac{1}{2}\int_{-\infty}^{\tau} \left(\nabla^2 - g\right) p d\tau' + (\epsilon_n p - \epsilon_t)\frac{\partial p}{\partial \tau} + \epsilon \frac{\partial \mathcal{L}p}{\partial \tau} \tag{2.19}$$

In the above equation the terms on the right hand side represent diffraction, non-linearity and absorption, respectively; each of these terms can be solved separately using operator splitting [12, 10].

[1] http://www.ntnu.no/isb/abersim/people
[2] http://www.gnu.org/copyleft/gpl.html

2.3.2 Implementation

In the implementation computation starts at depth $z = 0$ with the initial condition $p(x, y, 0, t) = p_0(x, y, t)$ which is propagated in steps of length Δ such that $z_k = k\Delta$. The step size should not be chosen arbitrary small, but ideally selected such that the splitting error is of the same order of magnitude as the accuracy of each of the numerical solution operators [10].

Absorption

The convolution operator $\mathcal{L}$ in Equation (2.19) follows a frequency dependent power law attenuation of the form

$$\alpha(f) = af^b, \tag{2.20}$$

where a and b are constants of the power law. And is defined through it's Fourier transform as

$$\mathcal{F}_\tau\left\{\frac{\partial \mathcal{L}p}{\partial \tau}\right\} = -|\omega|^b \mathcal{F}_\tau\{p\}. \tag{2.21}$$

The scaling constant ϵ is defined as:

$$\epsilon = \frac{\ln 10}{20}\frac{a}{(2\pi)^b} \tag{2.22}$$

Diffraction

Looking at the diffraction term in Equation (2.18),

$$\frac{\partial^2 p}{\partial \tau \partial z} = \frac{1}{2}\left(\nabla^2 - g\right)p, \tag{2.23}$$

it is clear that this is the diffraction for a linear wave equation in a loss less medium.

In Abersim heterogeneous effects are introduced as phase shifts between the propagation steps using a delay screen body wall. Hence all propagation is homogeneous and $g = 0$.

By expanding the ∇-operator and setting $g = 0$ Equation (2.23) can be written as an second order PDE[3] coupled in x, y and τ:

$$\frac{\partial^2 p}{\partial z^2} - \frac{2}{c}\frac{\partial^2 p}{\partial \tau \partial z} + \nabla_\perp^2 = 0, \tag{2.24}$$

where $\nabla_\perp^2 = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2}$.

To decouple Equation (2.24) the Fourier transform is used. First in time (τ) and then in space (x,y). This gives a fully decoupled second-order ODE[4] as shown below:

[3] Partial Differential Equation
[4] Ordinary Differential Equation

$$\frac{\partial^2 \tilde{p}}{\partial z^2} - 2ik_t \frac{\partial \tilde{p}}{\partial z} - \left(k_x^2 + k_y^2\right)\tilde{p} = 0, \tag{2.25}$$

where $k_t = \omega/c$ and k_x and k_y are the wave numbers in the x and y respectively.

This type of ODE has the solution on the form [9]

$$p(z_0 + \Delta) = Ae^{i\left(\sqrt{k_t^2 - k_x^2 - k_y^2} - k_t\right)\Delta} + Be^{-i\left(\sqrt{k_t^2 - k_x^2 - k_y^2} - k_t\right)\Delta}, \tag{2.26}$$

where Δ is the step size. By choosing the solution propagating in the positive direction the solution can found as

$$\tilde{p}(z_0 + \Delta) = \tilde{p}(z_0)e^{i\mathcal{K}_z\Delta}, \quad \mathcal{K}_z = \sqrt{k_t^2 - k_x^2 - k_y^2} - k_t. \tag{2.27}$$

Hence the solution at $z_0 + \Delta$ can be found by using the inverse Fourier transform in x, y and τ. However, by employing the FFT algorithm in the implementation periodicity in the frequency domain appears. The periodic effect is suppressed by applying a spatial window function which tapers the solution to zero at the edges.

Non-linearity

The non-linearity term in Equation (2.19) is solved by the method of characteristics as described in [10]. This introduces samples which are not equally spaced in the temporal direction. In order to preserve a equally spaced grid the solution is resampled by interpolation. This introduces an error, but as long as the sampling frequency is sufficiently high the error is negligible.

2.3.3 Abersim in FieldSim

When using Abersim in the FieldSim framework Field II is used to generate the initial pressure field. This done by calculating the pressure field in the azimuth-elevation plane at the initial depth z_0. This pressure field is then propagated by Abersim as discussed in Section 2.3.2.

In FieldSim the step size when using the Abersim simulator is assumed to be equidistant in a rectangular grid as illustrated in Figure 2.5 and can be expressed as

$$\Delta = \frac{\text{range\_max} - \text{range\_min}}{N - 1}. \tag{2.28}$$

However, Abersim internally has the ability to change the step size if needed. This is specially the case when doing simulations with aberrations or non-linear simulations. Abersim has no predefined grid for the output that is produced, the step size controls how many samples to be made in the depth dimension. For example, if the simulation range in depth from $z_0 = 0$

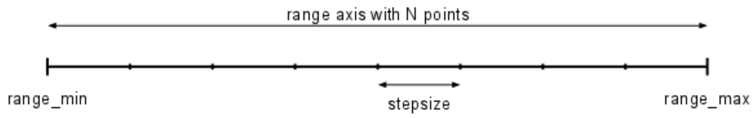


Figure 2.5: Range axis in FieldSim illustrating how the stepsize is computed. Range_min denotes the starting depth for the simulation and range_max denote the endpoint.

to $z_1 = 100$mm with step size 1 mm, the output will have 100 samples in the Z direction.

If Abersim decreases the step size, which initially is set by Equation (2.28), the number of output samples in depth will become higher than initially set up in FieldSim. To maintain the grid initially set up FieldSim keeps track of the changes in stepsize made by Abersim. The output is then interpolated from the depth axis in Abersim to the axis set up in FieldSim by using cubic splines.

2.4 Propose

Propose is a simulation tool designed for fast 3D simulations of the second harmonic field. This is achieved by using the quasi-linear approximation and not having to propagate the pressure field stepwise from the transducer surface to a wanted observation depth. Propose is implemented in MATLAB.

This section is largely based on [13] and [14], a more detailed explanation can be found here.

2.4.1 Theory

The underlying differential equation for Propose is the same as for Abersim, here written slightly different.

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} + \mathcal{L}(p) = -\frac{\beta_n}{\rho c^4}\frac{\partial^2 p^2}{\partial t^2} \tag{2.29}$$

where p, ρ, c, β_n represent the acoustic pressure, the speed of sound, the density of the medium and the coefficient of non-linearity, respectively. $\mathcal{L}(p)$ is a linear operator representing the loss. If $\mathcal{L}(p) = \frac{\delta \partial^3 p}{c^4 \partial t^3}$ where δ represents the diffusivity of sound, equation (2.29) is the Westervelt equation [14, 15]. In complex media like biological tissues the loss operator obeys the power law.

In quasi-linear theory the right-hand side of equation (2.29) is considered a small correction to the linear equation [14]. Only focusing on the fundamental and the second harmonic signal the acoustic pressure can be written $p = p_1 + p_2$. p_1 is the pressure at the fundamental frequency f_0 that satisfies the linear propagation equation and p_2 is the pressure at the second harmonic frequency $2f_0$ that satisfies the non-linear propagation equation. p is approximated to be p_1 in the non-linear term [14, 16].

$$\nabla^2 p_1 - \frac{1}{c^2}\frac{\partial^2 p_1}{\partial t^2} + \mathcal{L}(p_1) = 0 \tag{2.30}$$

$$\nabla^2 p_2 - \frac{1}{c^2}\frac{\partial^2 p_2}{\partial t^2} + \mathcal{L}(p_2) = -\frac{\beta_n}{\rho c^4}\frac{\partial^2 p_1^2}{\partial t^2} \tag{2.31}$$

To solve the above equations the Angular Spectrum Method (ASM) is used. The ASM decomposes a pulse of frequency f_0 into monochromatic plane waves and allows for the definition of the complex pressure $P(x, y, z, t)$ as a sum of complex exponentials [14]:

$$p(x, y, z, t) = \frac{1}{2}P(x, y, z, t) + \text{c.c.}, \tag{2.32}$$

where c.c. stands for complex conjugate. ASM works by taking the Fourier transform in the two spatial dimensions x and y, and the temporal dimension t. The Fourier transform of $P(x, y, z, t)$ is given as:

$$\hat{P}(\mathbf{k}, z) = \iiint P(x, y, z, t) \exp\left[-j(\omega t + k_x \cdot x + k_y \cdot y)\right] \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}t, \tag{2.33}$$

where $\mathbf{k}$ is a vector with the components $(\omega/c, k_x, k_y)$ with ω, k_x and k_y being the temporal angular frequency and spatial frequency components in x and y, respectively.

By applying the Fourier transform to Equation (2.30) and (2.31) the equations can be written in the frequency domain as

$$\frac{\partial^2 \hat{P}_1(\mathbf{k}, z)}{\partial^2 z} + K^2(\mathbf{k})\hat{P}_1(\mathbf{k}, z) = 0, \tag{2.34}$$

$$\frac{\partial^2 \hat{P}_2(\mathbf{k}, z)}{\partial^2 z} + K^2(\mathbf{k})\hat{P}_2(\mathbf{k}, z) = \frac{\beta_n \omega^2}{2\rho c^4}\hat{P}_1(\mathbf{k}, z) * \hat{P}_1(\mathbf{k}, z), \tag{2.35}$$

where $K(\mathbf{k}) = \sqrt{k^2 - k_x^2 - k_y^2}$ and $*$ represents convolution in all dimensions of $\mathbf{k}$. The imaginary part of $K(\mathbf{k})$ represents the attenuation and is the formulation in the frequency domain of the loss operator $\mathcal{L}$. To account for attenuation in biological tissue, that is known to follow the frequency power law, $K(\mathbf{k})$ can be written as [14]

$$K(\mathbf{k}) = \sqrt{k^2 - k_x^2 - k_y^2} - ja(f/10^6)^b, \tag{2.36}$$

where a is the attenuation factor given in neper per meter for a wave with frequency 1 Mhz.

The fundamental pressure is found from Equation (2.34) which is a well known ODE where one solution is

$$\hat{P}_1(\mathbf{k}, z) = \hat{P}_1(\mathbf{k}, z_0)e^{-jK(\mathbf{k})(z-z_0)}, \tag{2.37}$$

where z_0 is the observation depth. The sign in the exponential is chosen such that for $z \to \infty$ divergence is avoided.

The solution to Equation (2.35) is on the form $\hat{P}_2 = \hat{P}_{2h} + \hat{P}_{2p}$ where $\hat{P}_{2h}$ is the homogeneous solution which has the same of as $\hat{P}_1$ and $\hat{P}_{2p}$ is the particular solution. To solve for $\hat{P}_{2p}$ it is shown in [17] that Equation (2.35) can be expressed in terms of an integral equation using one-dimensional Green's functions. By taking into account the sign convention used for $K(\mathbf{k})$, $\hat{P}_{2p}$ can be written as[14]:

$$\begin{aligned}\hat{P}_{2p} = \frac{jM}{2K(\mathbf{k})}\Bigg(&\int_0^z e^{-jK(\mathbf{k})(z-z')}\mathcal{F}(\hat{P}_1) \\ &- \int_0^z e^{-jK(\mathbf{k})(z+z')}\mathcal{F}(\hat{P}_1) \\ &+ \int_z^\infty e^{-jK(\mathbf{k})(z-z')}\mathcal{F}(\hat{P}_1) \\ &- \int_z^\infty e^{-jK(\mathbf{k})(z+z')}\mathcal{F}(\hat{P}_1)\Bigg),\end{aligned} \tag{2.38}$$

where $M = \beta_n\omega^2/(2\rho c^4)$ and $\mathcal{F}(\hat{P}_1) = \hat{P}_1(\mathbf{k}, z') * \hat{P}_1(\mathbf{k}, z')$.

It turns out that the three last integrals in Equation (2.38) can be neglected in the case of weak non-linearity [17] making the first integral the dominant contribution, and by neglecting back propagation, the particular solution can approximately be found[14]:

$$\hat{P}_{2p}(\mathbf{k}, z) \approx \frac{jM}{2K(\mathbf{k})}\int_0^z e^{-jK(\mathbf{k})(z-z')}\hat{P}_1(\mathbf{k}, z') * \hat{P}_1(\mathbf{k}, z')\,\mathrm{d}z' \tag{2.39}$$

By assuming that $\hat{P}_{2p}(\mathbf{k}, 0) = 0$ and $\hat{P}_2(\mathbf{k}, 0) = 0$ it follows that $\hat{P}_{2h}(\mathbf{k}, 0) = 0$. Hence the solution to (2.35) is the particular solution $\hat{P}_{2p}(\mathbf{k}, z)$. Using $\hat{P}_1$ found in Equation (2.37) and writing out the convolution; $\hat{P}_2$ can be expressed as a function of the linear field $\hat{P}_1$ at depth z_0[14]:

$$\begin{aligned}\hat{P}_2(\mathbf{k}, z) = \frac{jM}{2K(\mathbf{k})}\int_0^z \int_{-\infty}^\infty &\hat{P}_1(\mathbf{k}', z_0)\hat{P}_1(\mathbf{k}-\mathbf{k}', z_0) \\ &e^{-jK(\mathbf{k}')(z'-z_0)}e^{-jK(\mathbf{k}-\mathbf{k}')(z'-z_0)} \\ &e^{-jK(\mathbf{k})(z-z)}\,\mathrm{d}z'\frac{\mathrm{d}\mathbf{k}'}{(2\pi)^3}\end{aligned} \tag{2.40}$$

The integral along z' to the point of interest z can be solved analytically and hence the final form of $\hat{P}_2$ is found:

$$\hat{P}_2(\mathbf{k}, z) = \frac{jM}{2K(\mathbf{k})} \int_{-\infty}^{\infty} \hat{P}_1(\mathbf{k}', z_0)\hat{P}_1(\mathbf{k} - \mathbf{k}', z_0)H(\mathbf{k}, \mathbf{k}', z, z_0)\frac{\mathrm{d}\mathbf{k}'}{(2\pi)^3}, \quad (2.41)$$

where

$$H(\mathbf{k}, \mathbf{k}', z, z_0) = z \cdot e^{-jK(\mathbf{k})(z-z_0)} e^{-j\Lambda(\mathbf{k},\mathbf{k}')(z_0-z/2)} \operatorname{sinc}\left(\Lambda(\mathbf{k}, \mathbf{k}')\frac{z}{2\pi}\right), \quad (2.42)$$

$$\Lambda(\mathbf{k}, \mathbf{k}') = -K(\mathbf{k}) + K(\mathbf{k}') + K(\mathbf{k} - \mathbf{k}'), \quad (2.43)$$

and

$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \quad (2.44)$$

By not having to do any stepwise computation this method allows for fast simulation of lateral profiles or pulse shapes at a given depth. Quasi-linear propagation with $p_1 \gg p_2$ and homogeneous medium are the conditions for using this method[14].

2.4.2 Pressure generation

The linear field in Equation (2.37) can be found for any depth z_0, however by taking z_0 to be the focal depth the wave at z_0 is proportional to the Fourier transform of the transducers aperture function $A(x, y)$. This comes from the Fraunhofer approximation of the Huygens principle which is valid in the far-field of an unfocused transducer or at the focal depth of a focused transducer and can be written as [14, 18]:

$$P_1(x, y, d) \approx \frac{dcf}{j\omega^2} e^{j\omega d/c} e^{\frac{j\omega}{2dc}(x^2+y^2)} \iint A\left(-\frac{k_x dc}{\omega}, -\frac{k_y dc}{\omega}\right) e^{j(k_x x+k_y y)}\, \mathrm{d}k_x\, \mathrm{d}k_y, \quad (2.45)$$

where d is the focal depth, c is the speed of sound and f is the frequency of a monochromatic wave.

By neglecting the proportionality factor and the phase term, and recognizing that the double integral can be seen as the inverse Fourier transform of $A(-k_x dc/\omega, -k_y dc/\omega)$ a spatial Fourier transform in the x and y dimensions give

$$\hat{P}_1(k_x, k_y, d) \propto A\left(-\frac{k_x dc}{\omega}, -\frac{k_y dc}{\omega}\right) * \hat{C}(\omega, k_x, k_y), \quad (2.46)$$

with

$$\hat{C}(\omega, k_x, k_y) = \mathbb{F}\left\{e^{\frac{j\omega}{2dc}(x^2+y^2)}\right\}, \quad (2.47)$$

where $\mathbb{F}$ denotes the spatial Fourier transform in x and y dimensions. Assuming that the aperture is symmetric, $A(-x, -y) = A(x, y)$, and generalizing from a monochromatic frequency to a pulse $\hat{P}(\omega)$ in the temporal frequency domain the linear field P_1 at focus depth d can be expressed in K-space as:

$$\hat{P}_1(\mathbf{k}, d) \propto \hat{P}(\omega) A\left(\frac{k_x dc}{\omega}, \frac{k_y dc}{\omega}\right) * \hat{C}(\omega, k_x, k_y). \tag{2.48}$$

In Equation (2.48) the focal distance in azimuth and elevation was assumed to be the equal. In the case where the focal distances in azimuth and elevation differ the aperture function needs to be corrected. Let d_x and d_y be the focal distances in azimuth and elevation, respectively. The pressure at $d = (d_x + d_y)/2$ is approximated by the pressure from a 2D array with focus distance, d. This is done by removing the phase shifts responsible for the azimuth and elevation focus at d_x and d_y and replace them with a delay corresponding to focus at d. These delays are found to be[14]:

$$\tau_x(x) = \frac{d_x - \sqrt{d_x^2 - x^2}}{c} \tag{2.49}$$

$$\tau_y(x) = \frac{d_y - \sqrt{d_y^2 - y^2}}{c} \tag{2.50}$$

$$\tau_{xy}(x, y) = \frac{d - \sqrt{d^2 - x^2 - y^2}}{c} \tag{2.51}$$

The corrected aperture function, $A'(x, y)$, becomes [14]

$$A'(x, y) = A(x, y) e^{-j\omega\Delta(x,y)/c}, \tag{2.52}$$

where $\Delta(x, y) = \tau_{xy}(x, y) - \tau_x(x) - \tau_y(y)$.

2.4.3 Implementation

In Propose the fundamental and second harmonic field can be treated as a monofrequency wave modulated by an envelope characterized by the pulse bandwidth B. The maximum frequency in the temporal dimension can be taken to be B[14]. The maximum spatial frequencies can be expressed in terms of the maximum radial temporal frequency, ω_m as

$$k_{xm} = \frac{\omega_m}{c}\frac{D_x}{d} \tag{2.53}$$

$$k_{ym} = \frac{\omega_m}{c}\frac{D_y}{d}, \tag{2.54}$$

where D_x and D_y are the aperture dimensions in azimuth and elevation, respectively. The maximum radial frequency should be set to $2\pi(f_0+B/2)$ and $2\pi(2f_0 + B/2)$ for the fundamental and second harmonic field, respectively.

In the implementation the maximum radial frequency is given by the highest frequency component, $f_{\max}$, in the temporal frequency axis of the initial field for the linear case. For the calculation of the second harmonic the maximum frequency is given by $2f_{\max}$ which is used to define new frequency axis' used in the computations.

The computational domain is set by the user and is defined by L_x, L_y and L_z for azimuth, elevation and time, respectively. The number of samples in each of the dimensions is defined as [14]

$$N_x = L_x \frac{D_x}{cd} \frac{\omega_m}{\pi}, \tag{2.55}$$

$$N_y = L_x \frac{D_y}{cd} \frac{\omega_m}{\pi}, \tag{2.56}$$

$$N_z = L_z \frac{2B}{c}. \tag{2.57}$$

If L_x and L_y is not chosen large enough spatial aliasing appear due to the discrete Fourier transform used in the implementation[14].

By default the pulse frequency response on transmit is given by a simple window function

$$P_{\text{tx}}(x) = e^{-x^4}, \tag{2.58}$$

where $x \in [-1.5, 1.5]$ defined by N_z samples.

Chapter 3

Propose in FieldSim

As explained in Section 2.4.2 Propose generates the initial linear field, $\hat{P}_1$, in K-space by applying the Fraunhofer approximation at the focal depth. In FieldSim Field II is used to generate the initial field which is linearly or non-linearly propagated by Propose. This chapter will explain in detail explain the implementation done in FieldSim, verify the FieldSim implementation against known simulators, compare execution times between the standalone Propose and Propose in FieldSim and at the end give an overview of new possibilities that arise from integrating Propose in the FieldSim framework.

3.1 Basic setup

The FieldSim framework is design to be easy to use and set up, and at the same time offer the features of a modern ultrasound scanner. Even though the framework has support for multi line acquisition (MLA), B-mode and other features as presented in Section 2.1 not everything is supported in the implementation of Propose in the framework.

The implementation discussed here focuses on the one way simulation of *one* ultrasound beam propagated in a homogeneous medium. When using Propose in the FieldSim framework, a rectangular 3D grid of size $N'_x \times N'_y \times N'_z$ where N'_x, N'_y and N'_z is number of samples in azimuth, elevation and depth, respectively, has to be defined by the user. The user also has to define the size of the computational domain, in meters, for all the dimensions stated above.

The computational domain is illustrated in Figure 3.1 where only the azimuth and depth dimensions are shown. `openingSize` is a tuple holding the values for the spatial dimensions azimuth and elevation, respectively. Propose by default only generates the pressure and profiles at one given depth. In the FieldSim implementation it is possible to set up a depth axis and simulate the pressure and profiles at given number of steps equally spaced between `range_min` and `range_max`. If the user only want to simulate

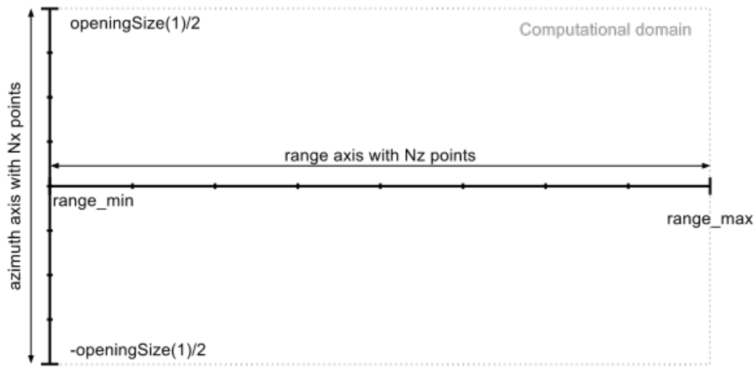


Figure 3.1: Illustration of the computational grid used in FieldSim when using the Propose simulator. Only azimuth and depth dimensions are shown, the actual computational domain is in three dimensions with an elevation axis similar to the one shown for azimuth.

one depth $N_z' = 1$ and `range_min` = `range_max` = z_0, where z_0 is the wanted depth, has to be defined. Listing 3.1 features an example of a simulation done with Propose in FieldSim.

3.2 Implementation

In Propose the computational domain is specified by the user in spatial and temporal dimensions. The initial field $\hat{P}_1$ is generated in K-space using the minimal amount of samples required to satisfy the Nyquist criteria in both spatial and temporal dimensions. The implementation in FieldSim tries to mimic this approach of using the minimal amount of samples by creating the initial pressure in Field II and decimating it to achieve fast running times.

As the implementation of Propose works in the frequency domain the pressure generated by Field II needs to be transformed from space and time into K-space, preferably as small and compact as possible. The default behaviour in FieldSim is to generate the initial field at the same focus as Propose, $d = (d_x + d_y)/2$, where d_x and d_y are the focal depths in azimuth and elevation, respectively. The user can however define d to be any depth.

The number of samples needed in the spatial dimensions are found by calculating the maximal spatial frequencies with the added option to oversample in the spatial dimensions. The highest frequency component in azimuth and elevation is approximated to

$$F_{x,\max} = \frac{f_0 + B/2}{2cF_{\#,x}} \tag{3.1}$$

$$F_{y,\max} = \frac{f_0 + B/2}{2cF_{\#,y}}, \tag{3.2}$$

where f_0 is the center frequency, B is the bandwidth, c is the speed of sound, $F_{\#,x}$ amd $F_{\#,y}$ is the F-number in azimuth and elevation, respectively. The bandwidth B is assumed to be the same in the fundamental and harmonic signal, this is also assumed in Propose [14].

The spatial sampling frequencies is thus given by

$$F_{s,x} = O_s \cdot 2F_{x,\max} \quad \text{and} \tag{3.3}$$

$$F_{s,y} = O_s \cdot 2F_{x,\max}, \tag{3.4}$$

where O_s is the factor of oversampling in space. Hence the number of samples needed to satisfy the Nyquist criteria for the spatial dimensions are

$$N_x = \lceil D_x/\Delta_x \rceil \quad \text{and} \tag{3.5}$$

$$N_y = \lceil D_y/\Delta_y \rceil, \tag{3.6}$$

where D_x and D_y is the computational domain for azimuth and elevation given in meters, respectively, $\Delta_x = 1/F_{s,x}$ and $\Delta_y = 1/F_{s,y}$. Note that the number of samples N_x and N_y, respectively, can be substantially less than the grid size defined (N'_x and N'_y) by the user as discussed in Section 3.1 due to N_x and N_y representing the minimum of samples needed.

The user is responsible for choosing D_x and D_y and should beware that by defining the computational domain too small, periodical effects can occur due to the use of the Angular spectrum method and FFTs used in the implementation of Propose. However by increasing the size of the computational domain the running time for the simulation increases. This is discussed in Section 3.4.

Having determined the number of samples needed to represent the computational domain in the spatial dimensions, Field II is used to generate the initial pressure at depth d. The default sampling frequency for the temporal dimension in FieldSim is set to 100 MHz and is higher than the Nyquist criteria for medical ultrasound transducers which typically operate in the 1 - 12 MHz range. The default sampling frequency is due to the spatial impulse method used by Field II which requires high sampling frequency [19].

The generated pressure from Field II, p_1, has the wanted size in the spatial dimensions but the time dimensions is sampled at 100 MHz as explained earlier. By decimating the signal a sampling frequency closer to the Nyquist limit is achieved. The sampling frequency in time can be expressed as

$$F_{s,t} = O_t \cdot 2(f_0 + B/2), \tag{3.7}$$

where O_t is the factor of oversampling in the temporal dimension, by default this factor is set to 1.1, which is 10% oversampling. The decimation factor used is

$$N_{\text{des}} = \lfloor F_s/F_{s,t} \rfloor, \tag{3.8}$$

where F_s is the default sampling frequency in FieldSim normally set to 100 MHz. The decimation is done by MATLABs built in function `decimate()` which reduces the sampling rate by a factor N_{des}, applies a lowpass to remove the periodic frequency components that appear when decimating and then resamples the signal to the wanted rate[20]. After decimation the initial pressure from Field II, $p_{1,\text{des}}$, is defined by $N_x \times N_y \times N_t$ samples in azimuth, elevation and time, respectively, where $N_t = \lceil N/N_{\text{des}} \rceil$ and N is the number of samples in time before decimation. The sampling frequency is now reduced to

$$F_{s,t,\text{des}} = F_{s,t}/N_{\text{des}}. \tag{3.9}$$

Propose internally operates on a nearly ideal K-space for the initial field $\hat{P}_1$ with spatial frequency axis' defined as $[-f_x/2, f_x/2]$ and $[-f_y/2, f_y/2]$, where $f_x = 1.1F_{s,x}$ and $f_y = 1.1F_{s,y}$, where the spatial frequency axis' is oversampled by 10%. The temporal frequency axis in Propose is defined by using positive frequencies and is band limited to only the needed signal in the interval $[f_0 - B/2, f_0 + B/2]$.

By construction the pressure field generated in the FieldSim implementation has the same spatial frequency axis' as Propose, but the temporal frequency axis is $[-F_{\text{des}}/2, F_{\text{des}}/2]$ where $F_{\text{des}} = F_{s,t,\text{des}}$ from on now for convenience. The solution chosen by FieldSim is to implement two options. One option is to use the full positive frequency axis, $[0, F_{\text{des}}/2]$, the other option is to use only the band pass signal as done in Propose, $[f_0 - B/2, f_0 + B/2]$.

To remove the negative frequencies the Hilbert transform is applied to the decimated initial pressure. The N-dimensional Fourier transform is then applied to transform $p_{1,\text{des}}$ to $P_{1,\text{des}}$ which is the decimated initial pressure in K-space with negative temporal frequencies removed.

$$P_{1,\text{des}} = \mathcal{F}\left\{\mathcal{H}_t\left(p_{1,\text{des}}\right)\right\}, \tag{3.10}$$

where $\mathcal{F}$ denotes the N-dimensional Fourier transform and $\mathcal{H}_t$ is the Hilbert transform done in the temporal dimension.

The final K-space for the option which uses the full frequency axis can now be obtained by only choosing the temporal frequencies in the range $[0, F_{\text{des}}/2]$. The band pass signal is obtained by filtering $p_{1,\text{des}}$ with a 10th order butterworth filter with cut-off at $f_{\text{lower}} = (1-\epsilon)(f_0 - B/2)$ and $f_{\text{upper}} = (1+\epsilon)(f_0 + B/2)$ for the lower and upper part of the pass band, respectively. ϵ is a "extra band" parameter that can be used to extend the band if wanted, by default $\epsilon = 0.025$. The wanted frequency axis is found by selecting samples in the range $[N_{\text{lower}}, N_{\text{upper}}]$ where

$$N_{\text{lower}} = (1-\epsilon)(f_0 - B/2)\Delta f \tag{3.11}$$

$$N_{\text{upper}} = (1+\epsilon)(f_0 + B/2)\Delta f \tag{3.12}$$

and $\Delta f = (N_t - 1)/F_{\text{des}}$.

As explained in Section 2.1.1 FieldSim has a standardized setup for materials that is based on how Abersim implements different materials. Propose has chosen the same approach. The only difference between the way FieldSim and Propose handles materials is that the constant α in Equation (2.20) is given with the unit dB/MHz/cm in FieldSim while Neper/m is used in Propose. The conversion is as follows:

$$a = 100\frac{\ln(10)}{20}\alpha, \tag{3.13}$$

where a is the constant in Equation (2.36) and α is the constant in Equation (2.20).

To summarize; there are two different options for generating the initial K-space that is propagated by Propose. Both signals are constructed the same way in spatial dimensions and have the same spatial frequency axis'. For the option that uses the full frequency domain the frequency axis is $[0, F_{\text{des}}/2]$ and for the band pass option (later denoted by BP) the frequency axis is $[f_0 - B/2, f_0 + B/2]$. This information is sent to Propose by using the `Propose('setparam')` and `Propose('setdata')` functions. By using the built in function to set parameters and data like frequency axis', initial field and material properties the implementation required minimal changes in the original Propose code. The only logic added are an option to show/hide the progress bar from Propose (FieldSim has its own progress bar) and let Propose return the propagated field without doing any interpolation, as this is done in FieldSim to control how the field is interpolated.

As mentioned earlier the number of samples in azimuth and elevation, N_x and N_y, respectively, for the initial field is not necessarily the same as the number of samples, N'_x and N'_y, defined by the user as the grid size. After the initial field has been propagated by Propose it is interpolated in the spatial dimensions from N_x and N_y samples to N'_x and N'_y samples in azimuth and elevation, respectively. This is done by the same approach chosen in Propose which is symmetric zero-padding in the frequency domain. To ensure symmetry the grid will automatically be resized to N'_x+1 or N'_y+1 if either N'_x or N'_y is even, respectively.

Note that only the spatial directions are interpolated, not the temporal dimension. This is due to the number of samples in the temporal dimension in the initial field from Field II which causes overhead vs. the standalone Propose as the time it takes to do the interpolates increases notably. As the implementation stands at the moment the only way to increase the number of samples in the temporal dimension is to increase the oversampling factor in time O_t in Equation (3.7). More on this in chapter 6.

While the spatial dimensions can be treated the same way, the temporal dimension is dependent on whether or not the band pass option is used. When the band pass option is enabled the pressure is stored in IQ form. To get the actual pressure the user has to modulate the signal to the center frequency $f = f_0$ for the fundamental harmonic and $f = 2f_0$ for the second harmonic and the real part of the modulated signal, mathematically expressed as:

$$p_{\text{bp}}(x, y, z_0, t) = \text{Re}\left\{p'_{\text{bp}}(x, y, z_0)e^{j2\pi ft}\right\}, \tag{3.14}$$

where z_0 is simulation depth of interest, $p'_{\text{bp}}(x, y, z_0)$ is the unmodulated pressure at z_0 and t is the decimated time axis defined by $F_{s,t,\text{des}}$. For the signal using the full frequency axis only the real part is needed

$$p_{\text{full}}(x, y, z_0, t) = \text{Re}\left\{p'_{\text{full}}(x, y, z_0)\right\}. \tag{3.15}$$

3.3 Verification

The FieldSim incorporation of the Propose simulator is compared against other known simulation tools, namely the standalone version of Propose, Field II and Abersim in 4 test cases. The first two simulations focuses on the fundamental while the other two focuses on the second harmonic. Both the fundamental and the harmonic case have variations with and without frequency dependent attenuation.

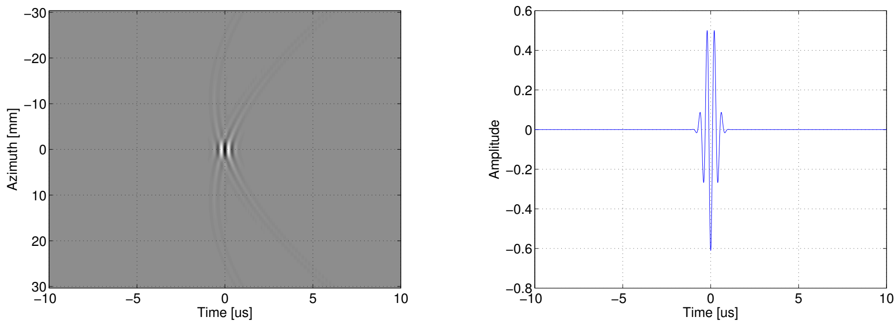
All simulations, with Abersim as the exception, were carried out on a machine running Ubuntu 12.04 with a 3.3 GHz Dual-Core Intel i5-2500K and 8 GB RAM running MATLAB 2011b. The Abersim simulations were carried out on a machine running Ubuntu Server 11.04 with a 2.66 GHz 6 core Intel Xeon X5650 and 36 GB RAM running MATLAB 2011b.

Due to problems with getting the C version of Abersim to compile on the 64 bit machine all the simulations was done using the MATLAB version of Abersim, making them orders of magnitude slower. This is important to have in mind when talking about the running times for the different simulators.

In this section the Propose implementation in FieldSim is compared against the original Propose tool, Field II and Abersim for linear and non-linear propagation, with and without attenuation.

In all the test cases in this section the M5S probe was used. The M5S is a cardiovascular probe with 96 × 3 elements and dimensions 22 mm × 13 mm in azimuth and elevation, respectively, and has a lens focus of 86 mm in elevation.

For Propose in FieldSim the initial pressure was generated by Field II as explained in section 2.4.2. The focus depth follows the convention used in Propose, that is $z_{\text{focus}} = (d_x + d_y)/2$, where d_x is the focal depth in azimuth and d_y is the focal depth in elevation. For the test cases the azimuth focus d_x = 75 mm was chosen and the elevation focus d_y = 86 mm was given by the probe which together gives z_{focus} = 80.5 mm. Even if the FieldSim framework allows the use of measured impulse responses, a Gaussian response was chosen as this is the response used by the standalone Propose.



(a) (b)

Figure 3.2: The initial pressure field generated by Field II at depth $z_{\text{focus}} = 80.5$ mm. Center frequency $f_0 = 2.5$ MHz, bandwidth $B = 0.6$ and 1.5 periods. **a)** The initial pressure field in the XZ-plane. **b)** The RF-signal at the center of the transducer.

3.3.1 Case 1: Linear without attenuation

The initial pressure for Propose in FieldSim was created by using Field II, and can be seen in Figure 3.2. A excitation pulse with center frequency of $f_0 = 2.5$ MHz, bandwidth $B = 0.6$ (60% fractional bandwidth) and 1.5 period was used. In the simulation the number of samples used in azimuth and elevation were $N'_x = 300$ and $N'_y = 300$ with a computational domain with dimensions 60 mm × 60 mm (D_x, D_y) in the spatial dimensions azimuth and elevation, respectively. The beam profile was simulated from depth $z_0 = 1$ mm to $z_1 = 100$ mm with $N'_z = 100$ samples. The simulation was run two times with two different spatial oversampling factors, $O_s = 1$ (no oversampling) and $O_s = 1.5$ (50% oversampling).

For the standalone Propose the aperture size and focus was set according to the M5S probe. The computational domain was specified to be the same as for the FieldSim implementation, 60 mm × 60 mm in azimuth and elevation, respectively. However as explained in section 2.4.3 Propose requires the user to specify the extent of the time dimension, L_z, as well. L_z were chosen to be 8 mm. The resolution was set to (0.2 mm, 0.2 mm, 0.1 mm) in the (x,y,z) dimensions, respectively. The resolution in the spatial dimensions x and y were calculated by D_x/N'_x and D_y/N'_y, respectively. Using Propose two depths were simulated at 40 mm and 75 mm.

For Field II the setup is nearly identical to the way Propose in FieldSim is set up. This can be done by using the `copy()` function and change the simulator from Propose to Field II. However, instead of simulating the whole 3D space, two simulations were set up, one for the azimuth plane and an other for the elevation plane.

The setup of Abersim was also very similar to the one of Propose in

FieldSim, again due to the standardized setup in the FieldSim framework. Field II was used to generate the initial pressure near the transducer surface in 3D. This pressure field was then propagated by Abersim. When simulating pressure fields near the transducer with Field II it is important to have enough mathematical elements in the simulation so that the far-field approximation holds[19, 21]. Here 12 elements in azimuth and elevation was used and the sampling frequency were set to 100 MHz.

3.3.2 Case 2: Linear with attenuation

The same setup as in Section 3.3.1 was used with the addition of enabling frequency dependent attenuation. The attenuation were set to be 0.5 dB/MHz/cm or 5.76 Neper/m in a material where $c = 1540$ m/s at 37 °C. The power in the power law, b, was set to $b = 1$.

3.3.3 Case 3: Non-linear without attenuation

For the non-linear test case the same setup as in Section 3.3.1 was used. Non-linear in this setting means the use of simulating the second harmonic with Propose as section 2.4 explains, with Abersim the simulation is done as explained in Section 2.3 where the second harmonic is filtered out. It is these two beamprofiles that are compared with this test case.

3.3.4 Case 4: Non-linear with attenuation

This test case is the same as Case 3 with attenuation enabled. The material used was assumed to have the following properties: $c = 1540$ m/s, $\rho = 1.05$ mg/mm^3, $\kappa = 4.0158 \cdot 10^{-10}$ and $\beta_n = 3.5$ which are the constants in Equation (2.35) and (2.17) for Propose and Abersim, respectively. The frequency dependent attenuation is the same as in Case 2, that is 5.76 Neper/m or 0.5 dB/MHz/cm in Propose or Abersim, respectively. This gives $\epsilon_n = 0.0091$, $\epsilon = 0.0092$ and $\epsilon_t = 1$ in Equation (2.18).

3.4 Speed evaluation

Propose is designed to be used for fast 3D simulation of the second harmonic field. In this section the running times for the standalone version of Propose and Propose in FieldSim are compared.

When looking at the running time as a function of center frequency f_0 it can be seen from Equation (3.1) and (3.2) that the maximum spatial frequency in azimuth and elevation, respectively, is dependent on f_0. The number of samples in all the dimensions, both spatial and temporal, is thus dependent on f_0.

The computation of the fundamental field is a straightforward matrix multiplication. The calculation of the second harmonic however is more expensive due to the convolution in Equation (2.41). The number of operations used to calculate the second harmonic field is on the order of $N_x^2 \cdot N_y^2 \cdot N_t^2$ [14]. Since the number of samples in the spatial dimensions depend on the center frequency, which also determines the number of samples in the temporal dimension, the calculation of the second harmonic can be expressed as $O(n^6)$ using asymptotic notation. This indicates that the running time of the algorithm should increase notably when increasing the center frequency.

To compare the standalone version of Propose against Propose in FieldSim a computational domain of 40 mm × 40 mm in azimuth and elevation, respectively, was used. In standalone Propose the temporal dimension was defined by $L_z = 3 \cdot c/B$ as done in [14]. For Propose in FieldSim the same computational domain was used in the spatial dimensions and the oversampling factors used were $O_s = 1$ and $O_t = 1.1$. The results are presented in Section 5.2.

3.5 New possibilities

By combining different simulation tools new possibilities arise. For example, Propose can calculate the second harmonic field, but by default assumes a simple window function as the pulse frequency response on transmit as explained in Section 2.4.3. Field II however simulate linear propagation, but supports the use of arbitrary transducer designs with a measured impulse response convolved with the excitation pulse. By using Field II to initialize Propose, calculation of the second harmonic field with a measured impulse response is made possible. This isn't necessarily new since it is possible to define a custom pulse frequency response in Propose, but having both simulators implemented in the same framework makes this easier and standardized. This section discusses some of the new features by implementing Propose in FieldSim.

3.5.1 Standardized setup

With FieldSim the user has a standardized way of setting up simulations. This makes it easy to use the same simulation setup in multiple settings and should be less error prone since the user only has to relate to one setup instead of one for each possible simulation tool.

In FieldSim abstraction makes it more natural to work with different components of a setup. For example a probe can be initialized only by using the name of the probe without having to remember any details about the probe like number of elements, pitch sizes and geometry. The same goes for materials. The user can choose from a range of materials including water, muscle, liver and more. The material properties will automatically be recalculated if the temperature is changed.

The abstraction is done without having to sacrifice detailed control. Any parameter that defines a component like a probe or a material, for example, can be changed after initialized. For a probe the pitch or the number of active elements could be changed. This could be done by either changing the F-number or the minimum and maximum aperture.

By integrating Propose in FieldSim the user will get the power and features from Propose with the ease of use of FieldSim.

3.5.2 Measured impulse response

As mentioned in the overview of FieldSim, the probe specification contains information about the impulse response. The impulse response can be defined in two different ways. It can either be specified as a function or it can be a measured impulse response. This way it is possible to measure the impulse response of a transducer and use it to simulate the pressure generated by a given transducer. If the user has access to proprietary impulse responses from GE Vingmed or Ultrasonix, this is supported as well. This makes it possible to simulate with measured impulse responses in Propose.

The pulse from the transducer is generated by convolving the impulse response and a excitation pulse. The user is free to define the excitation pulse and its period.

3.5.3 Steering

FieldSim has the possibility to rotate the Field II generated pressure field. This way steering can be added to Propose by creating a rotated initial pressure field and then using Propose to propagate the field. When steering the user has to be aware of the underlying method for propagation in Propose which is the Angular Spectrum Method as discussed in Section 2.4.1. This can cause periodical effects if the computational domain is not chosen big enough. As a consequence the running time of the simulation will increase because of the need for more samples in the spatial dimensions as shown in Section 2.4.3. Listing 3.1 shows how to simulate a steered beam using Propose in FieldSim.

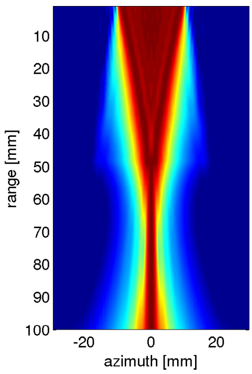
3.6 Example

In Listing 3.1 the code used to simulate a beam profile in FieldSim using the Propose simulator is shown. A linear simulation using the 4V probe in a material with the properties of muscle is set up. The number of samples is defined to be 300×100×100 in azimuth, elevation and depth, respectively. The computational domain is chosen to be 60 mm× 60 mm in the spatial dimensions, with an oversampling factor in space of 1.5. The beam profile is set up to be simulated from depth $z_0 = 1$ mm to $z_1 = 100$ mm. As explained in Section 3.5.3 by rotating the initial pressure steering can be achieved. In this example the steering angle is set to 5 °C.

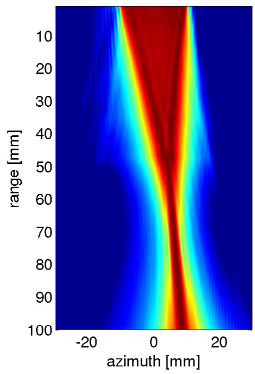
Since the FieldSim uses a standardized setup the only change needed to do the same simulation with Field II is to change the simulator by defining `sim.simulator = 'FieldII'` and rerun the simulation. If a comparison is wanted the whole simulation can be copied by doing `sim2 = sim.copy()`. This will copy all the variables and parameters from the original simulation, `sim`, to a new simulation instance, `sim2`. By changing the simulator used in `sim2` two simulators can be compared in few lines of MATLAB code. This way multiple simulations can be based on the same setup, without having to define multiple setups.

Listing 3.1: Example of a beam profile simulated with and without tilt in FieldSim using the Propose simulator.

```
% Create new FieldSim instance
sim = FieldSim.Simulation();

% Use Propose simulator
sim.simulator = 'Propose';

% Simulate a beam profile
sim.selectMode('BeamProfile');

% Set transmit
sim.scan.type = 'Tx';

% Select probe
sim.probe = '4V';

% Choose material
sim.propagation.material = FieldSim.Medium.Material.Muscle;

% Select mode, linear or non-linear
sim.simulator.nonlinear = 0;

% Set grid size in samples [az,el,depth]
sim.scan.grid.resolution = [300 100 100];

% Define the computational domain in meters
sim.scan.grid.openingSize = [0.06 0.06];

% Oversample?
sim.simulator.oversampleSpace = 1.5;

% Simulate from z = 1mm to 100mm
sim.scan.grid.range_min = 1e-3;
sim.scan.grid.range_max = 100e-3;

% To tilt, or not to tilt, that is the question
sim.scan.txBeam.tilt(1) = 5*pi/180; % [rad]

% Start simulation
sim.doBeamProfile();

% Plot the beam profile
sim.data.plot;
```



(a)



(b)

Figure 3.3: Results from Listing 3.1. beam profile propagated by using the Propose simulator without and with tilted beam.

Chapter 4

Measurements in water tank

The implementation of Propose in FieldSim will be compared against measurements done in the water tank. This chapter describes the laboratory setup, how the measurements were carried out and introduces a link between the FieldSim framework and research scanners supported by the Texo interface.

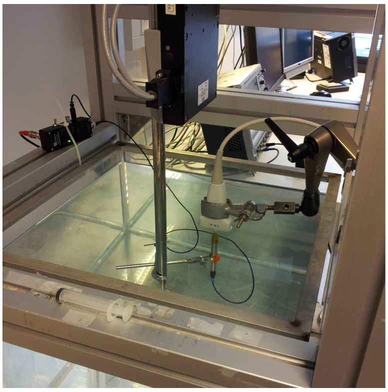
4.1 Laboratory setup

To do measurements in a water tank certain equipment and tools are needed. The measurements are made by recording the ultrasonic field from a transducer connected to a ultrasound scanner with a hydrophone. The hydrophone is mounted in a robot rig that allows a computer to position the hydrophone with high accuracy in the spatial dimensions X, Y and Z. The hydrophone is connected to a preamp which is connected to an oscilloscope. The oscilloscope and the robot rig is connected to a computer running ProbeLab. ProbeLab is a program developed at ISB[1], which controls the robot rig and in effect the position of the hydrophone. At the same time the measured field from the hydrophone is recorded via the oscilloscope. A picture of the laboratory setup is shown in Figure 4.1.

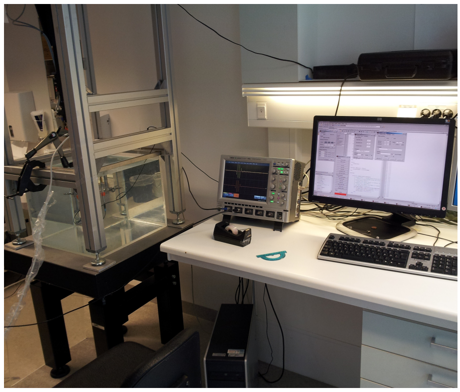
4.2 Texo interface

Any ultrasound scanner and transducer can be used in the water tank as the probe is mounted in a stationary position near the water surface. However, at ISB an Ultrasonix research scanner is available. This is a scanner with

[1]Department of Circulation and Medical Imaging, NTNU



(a)



(b)

Figure 4.1: Pictures of the laboratory setup. **a)** The water tank with the L9-4/38 probe and Onda hydrophone. **b)** The LeCroy oscilloscope and a computer with ProbeLab installed.

the possibility to change parameters that normally would be restricted on a regular ultrasound scanner at the hospital.

To ease the setup of the research scanner an interface named Texo was developed. This is an interface against the scanners API written in C++. Later a wrapper for the C++ interface was written in Python, PyTexo, to ease the use further. The possible parameters the interface can control is listed in Appendix A. A scan line is defined by the parameters listed in Appendix A, multiple scan lines from a scan sequence and can be used to define a whole frame. The current version (as of June 2012) of PyTexo a scan sequence can be loaded from a HDF5[2] file.

With PyTexo being able to load a scan sequence from a file it was possible to develop a link between the research scanner and the FieldSim framework. This is done by converting the standardized simulation setup in FieldSim to a HDF5 file containing the parameters listed in Appendix A. FieldSim has two different scan modes: a single beam profile and B-mode scan. For the beam profile mode only one scan line is created, while for the B-mode the number of scan lines created depends on how many beams are used in the simulation.

There are limitations to both the Texo interface and the FieldSim link. Texo only supports 1D transducers with a maximum of 128 elements, which means that only 2D B-mode simulations in FieldSim can be converted the PyTexo file format. For B-mode dynamic focusing is typically used in simulations, but the current version only support fixed focus.

The development is in the early stages and only basic features has been implemented as of today (June 2012). By implementing full support of the

[2]http://www.hdfgroup.org/HDF5/

features offered by FieldSim, or as much as the Texo interface allows, the set up of measurements in the water tank will become more accessible to the user. By learning to use one framework, FieldSim, the user will be able to simulate using multiple simulation tools with different niches, take these simulations and transfer them to the research scanner and do the actual measurements with the exact same setup used in simulations.

4.3 Simulations vs. measurements

To compare Propose in FieldSim against measurements in the water tank the FieldSim link to the research scanner was to be used. However, the HDF5 file generated by MATLAB was storing the scalar data values as one dimensional arrays. PyTexo then loaded the values as arrays containing one scalar value in Python causing PyTexo to crash. Having only limited time on the lab the decision was made to drop the file generated by FieldSim and use PyTexo instead. A discussion around the failure of the file generated by FieldSim is given in Section 6.2.

The measurements were carried out by using a LeCroy WaveSurfer 44Xs 400 MHz 2.5 GS/s oscilloscope and a calibrated Onda HGL-0200 S/N 1263 hydrophone. The research scanner was an Ultrasonix SonixMDP with the L9-4/38 probe connected. Using PyTexo the center frequency was set to 5 Mhz, PRF = 500 Hz, 1.5 period excitation, focus distance of 2 cm and 32 active elements at the center of the transducer. The number of active elements was chosen to get a F-number of approximately 2. The water temperature was measured to be 22.6 °C giving $c = 1490$ m/s.

The measurements were carried out at the focal depth of 2 cm. The hydrophone was moved incrementally from (-1.5 mm, 0 mm, 20 mm) (x,y,z) to (1.5 mm, 0 mm, 20 mm) with 1.5 mm increments in azimuth and from (-2 mm, 0 mm, 20 mm) to (-2 mm, 0 mm, 20 mm) with 1.9 mm increments in elevation.

The simulation with Propose in FieldSim was done with both 0% and 50% oversampling in space with a computational domain with axis' [-2 mm, 2 mm] in both azimuth and elevation. The grid size used was 300×300. The center frequency used was the same as in the measurements with a fractional bandwidth of 65%, same as the L9-4/38 probe.

For Abersim the initial pressure was generated by Field II close to the transducer using 12 mathematical elements in both azimuth and elevation to satisfy the far field approximation. The initial pressure was propagated non-linearly from $z_0 = 0.5$ mm to $z_1 = 20$ mm. A computational domain with the same dimensions as Propose was chosen. The material was set to water which has the following properties at 22.6 °C: $c = 1490$ m/s, $\rho = 997.63$ mg/mm^3, $\kappa = 4.5150{\cdot}10^{-10}$ and $\beta_n = 3.5179$ which is the constant in equation (2.17). The frequency dependent attenuation is given by $a = 0.002$ and $b = 2$ in the power law. This gives $\epsilon_n = 0.0107$, $\epsilon = 5.8553{\cdot}10^{-6}$ and $\epsilon_t = 2$ in Equation (2.18).

Chapter 5

Results

Results from the different test cases in Section 3.3, speed evaluation in Section 3.4 and the measurements made in the water tank, in Section 4, is presented in this chapter. A discussion of the results is given in Chapter 6.

For this chapter a reference to FieldSim where Propose is present refers to the Propose implementation FieldSim, while Propose refers to the standalone version of Propose. When referring to "FieldSim bp" the "bp" part indicates that the band pass option for Propose in FieldSim, as explained in Section 3.2, is enabled, while only "FieldSim" indicates that the full frequency axis is being used.

5.1 Verification of Propose

5.1.1 Case 1: Linear without attenuation

The simulated profile at 40 mm is shown in Figure 5.1. Figure 5.1a and 5.1b shows the lateral RMS profile in azimuth and elevation, respectively, with no oversampling in space. It can be seen that there is a different between Propose and the other simulators. This is caused by the other simulators being based on an initial pressure from Field II. With no oversampling in space both FiledSim and FieldSim bp lies slightly above Field II while Abersim compared to Field II is nearly identical in azimuth but in elevation is slightly above Field II.

The MSE of the normalized profiles over the lateral extent -10 mm to 10 mm is below 1.5 dB and 2.5 dB in azimuth and elevation, respectively, for both FieldSim profiles compared to Field II and below 2 dB and 2.5 dB when compared against Propose. When 50% oversampling in space was used, as shown in Figure 5.1c and 5.1d, Field II and FieldSim are nearly identical and below 0.5 dB. Figure 5.1e and 5.1f shows the cross sectional RMS profile for Propose and FieldSim. The ripples caused by the band limited signal when using Propose are gone when FieldSim with a full frequency axis is used.

Figure 5.2 presents the same data as above, but at the focal depth 75 mm. The same trends as in Figure 5.1 are present, however it can be seen that FieldSim bp has lower zero-point than FieldSim and Field II. In Figure 5.2e the ripple effect from a band limited signal is even more clear than in Figure 5.1e. All the normalized profiles for FieldSim and FieldSim bp when compared against Field II was below 0.5 dB. Compared against Propose the profiles were below 2 dB.

The axial RMS beam profile is shown in Figure 5.3. With no oversampling used in space it can be seen that both FieldSim and FieldSim bp is below Field II. When the the oversampling factor is increased to 50% the profile for FieldSim using the full frequency axis is nearly identical to Field II. FieldSim bp is closer to Field II with oversampling but deviates more than FieldSim using the full frequency axis. For Abersim it can be seen that the profile is similar to that of Field II except in the range between 20 mm and 50 mm. When comparing both normalized axial RMS profiles from FieldSim against Field II the MSE are found to be below 2 dB with no oversampling in space and below 0.5 dB for FieldSim and below 1 dB for FieldSim bp.

It should be noted that the running time for FieldSim for simulating both at 40 mm and 75 mm depths were 4 - 5 times greater than Propose.

5.1.2 Case 2: Linear with attenuation

With attenuation enabled FieldSim and FieldSim bp are very similar without any oversampling, as shown in Figure 5.4. When the oversampling factor in space is increased to 1.5 the same effect as noted for case 1 occurs. MSE for the normalized axial RMS profile is below 0.5 dB for FieldSim and 2 dB for FieldSim bp with 50% oversampling in space. The running time did not increase notably when enabling attenuation in the simulation, Propose is 4 - 5 times faster than FieldSim.

5.1.3 Case 3: Non-linear without attenuation

Non-linear simulations were compared as explained in Section 3.3.3 at the focal depth 75 mm. Figure 5.5 shows lateral RMS profiles of the second harmonic propagated by Propose, FieldSim and Abersim as well as the fundamental from Field II. In Figure 5.5a and 5.5b the profiles generated with no oversampling in space are presented. In the azimuth dimension it can be seen that Propose is similar to both FieldSim and FieldSim bp. Abersim has slightly higher side lobes than FieldSim and Propose. In elevation Propose has higher side lobes than FieldSim, but lower than Abersim. FieldSim and FieldSim bp are similar as expected due to the same initial pressure. For Abersim the side lobes are notably higher than FieldSim, but the peaks are at the same position along the elevation axis.

With 50% oversampling in space the change in FieldSim and FieldSim bp are minimal, but the profiles are slightly broader in azimuth as can be seen at the bottom of Figure 5.5c. The same effect is present in elevation where the peaks are slightly higher in Figure 5.5d.

The MSE of the normalized profiles over the lateral extent -10 mm to 10 mm in azimuth is below 1.5 dB and 2 dB for FieldSim and FieldSim bp, respectively. In elevation the average mismatch was found to be below 3 dB for both FieldSim and FieldSim bp. Compared against Propose the MSE was found to be below 1 dB in azimuth and elevation.

The cross sectional RMS profiles at focus are shown in Figure 5.2e and 5.2f. The profiles are similar except from the ripple effect as commented earlier.

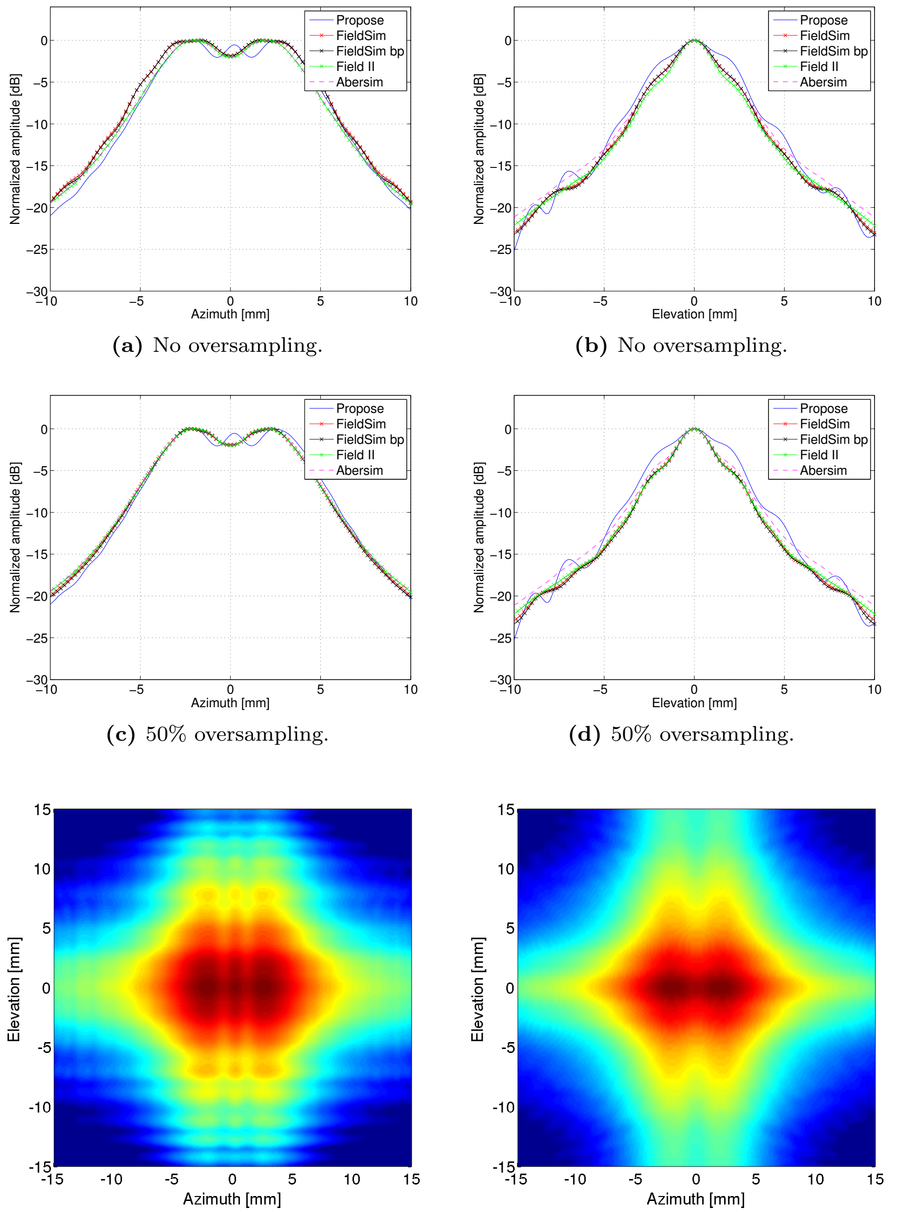
The running time without oversampling in space were 3 times longer for FieldSim when using the full frequency axis compared to the standalone version of Propose. With the band pass option enabled the running time was equal to Propose. By increasing the oversampling factor to 1.5 the running times increased to 9 - 10 and 2 - 3 times slower for FieldSim and FieldSim bp, respectively, when compared to Propose. The MATLAB version of Abersim used close to 40 hours to complete the simulation.

5.1.4 Case 4: Non-linear with attenuation

Figure 5.6 shows the lateral RMS profiles at 75 mm and 40 mm, respectively, with an oversampling factor of 1.5. At both 40 mm and 75 mm Propose and FieldSim can hardly be distinguished from each other. For FieldSim bp and Abersim in focus the observations are the same as for the case where no attenuation was taken into account. The measured MSE between Propose and FieldSim was below 0.5 dB in azimuth and 1 dB in elevation.

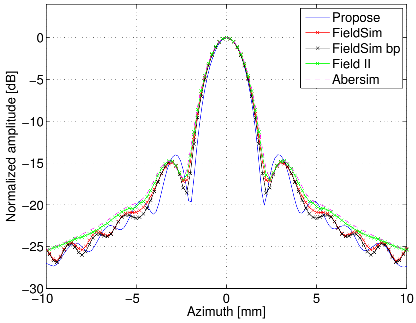
At 40 mm FieldSim bp has a wider profile than Abersim in azimuth as shown in Figure 5.6c. FieldSim using the full frequency axis has a more similar curvature to Abersim than the band limited version as can be seen at the center of the azimuth profile. From Figure 5.6d it can be seen that Abersim is wider than both FieldSim and FieldSim bp in the elevation dimesion and FieldSim bp being wider than FieldSim. In azimuth and elevation the MSE compared to Abersim was below 1.5 and 3 dB, respectively, for FieldSim. With band pass enabled the mismatch was below 2.5 and 3 dB in azimuth and elevation, respectively.

Figure 5.6e and 5.6f shows axial RMS profiles with oversampling by a factor of 1 and 1.5, respectively. The profiles generated using FieldSim are similar in shape, but the lens focus is more present in Abersim. By increasing the oversampling factor to 1.5 it can be seen from Figure 5.6f that this effect is more present in FieldSim as well. In Figure 5.6f only the band passed initial pressure is oversampled due to the running time causing the simulation to hang when using the full frequency axis. With 0% and 50% oversampling the error was found to be below 2 and 1.5 dB, respectively. The calculation of the axial profile with band pass enabled finished in 1 hour and 10 minutes. Abersim completed the simulation in approximately 42 hours.

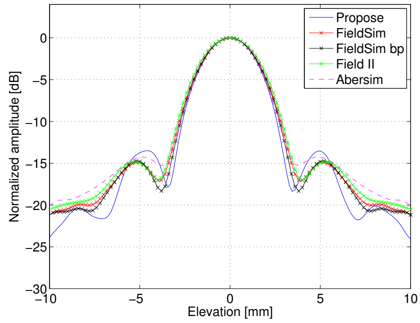


(a) No oversampling.
(b) No oversampling.
(c) 50% oversampling.
(d) 50% oversampling.
(e) Propose
(f) FieldSim

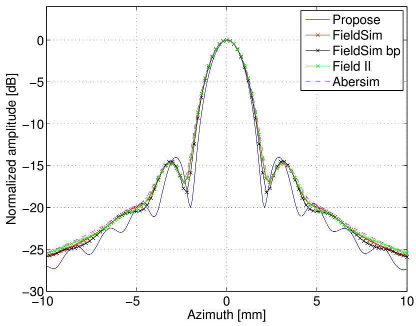
Figure 5.1: Results from Case 1: Linear simulation without attenuation at 40mm. **a)** Lateral RMS azimuth profile with no oversampling. **b)** Lateral RMS elevation profile with no oversampling. **c)** Lateral RMS azimuth profile with 50% oversampling in spatial dimensions. **d)** Lateral RMS elevation profile with 50% oversampling in spatial dimensions. **e)** Cross sectional RMS profile in focus from Propose. **f)** Same as e) but simulated by FieldSim.



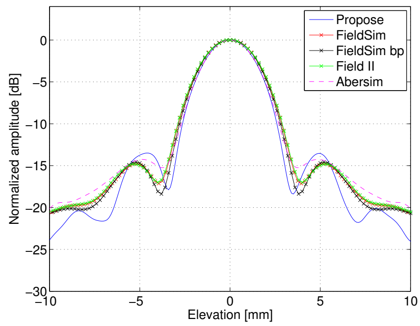
(a) No oversampling.



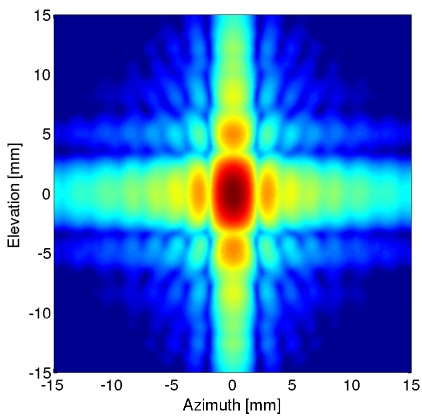
(b) No oversampling.



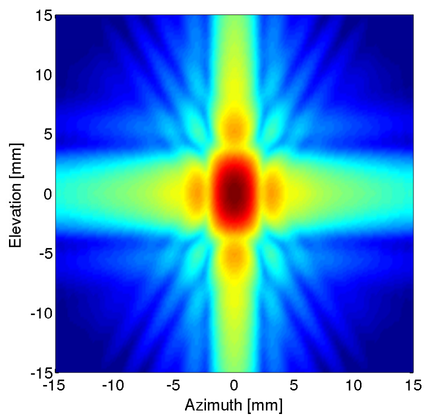
(c) 50% oversampling.



(d) 50% oversampling.



(e) Propose



(f) FieldSim

Figure 5.2: Results from Case 1: Linear simulation without attenuation at 75mm. **a)** Focal plane RMS azimuth profile with no oversampling. **b)** Focal plane RMS elevation profile with no oversampling. **c)** Focal plane RMS azimuth profile with 50% oversampling in spatial dimensions. **d)** Focal plane RMS elevation profile with 50% oversampling in spatial dimensions. **e)** Cross sectional RMS profile in focus from Propose. **f)** Same as e) but simulated by FieldSim.

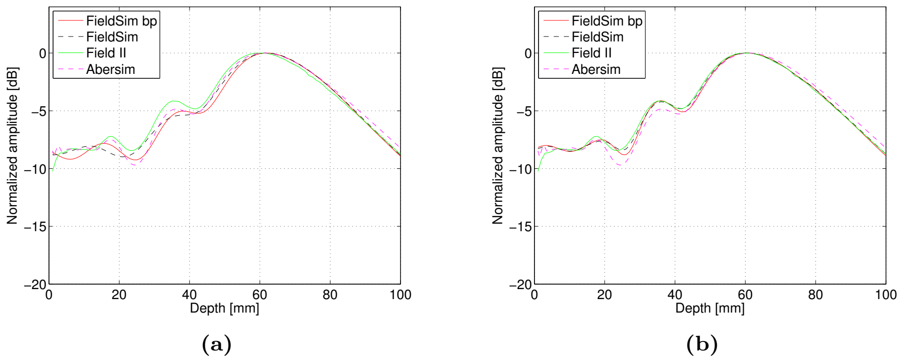


Figure 5.3: Results from Case 1: Linear simulation without attenuation. **a)** RMS axial profile with no oversampling in space. **b)** RMS axial profile with 50% oversampling in space.

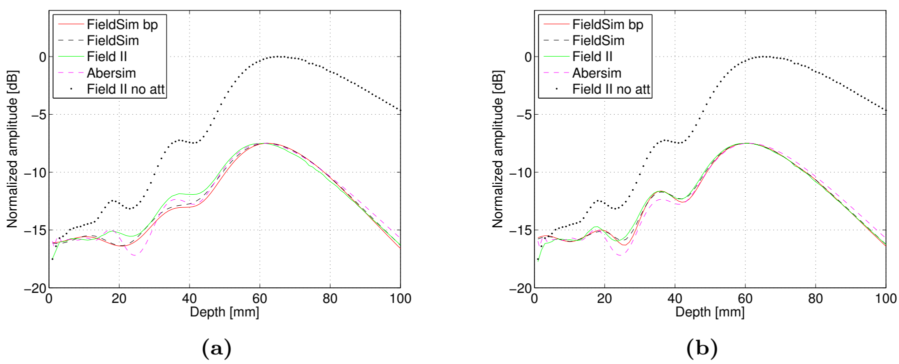
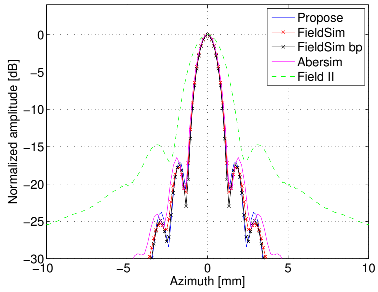
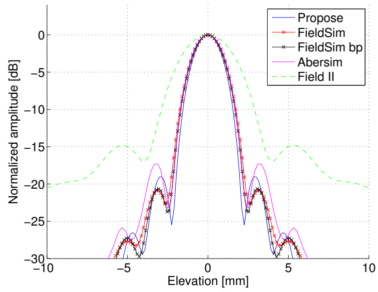


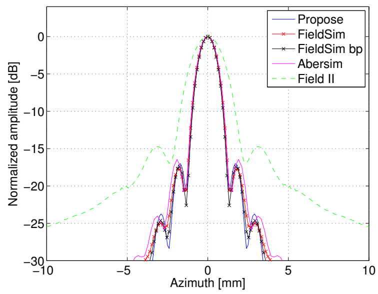
Figure 5.4: Results from Case 2: Linear simulation with attenuation. **a)** Axial RMS profile with no oversampling in space. **b)** Axial RMS profile with 50% oversampling in space.



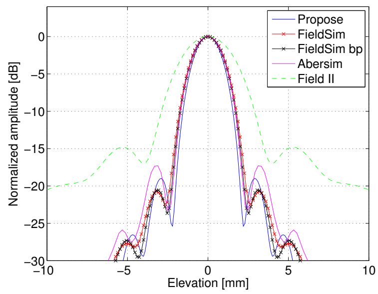
(a) 0% oversampling.



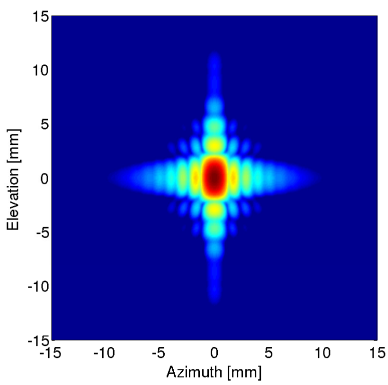
(b) 0% oversampling.



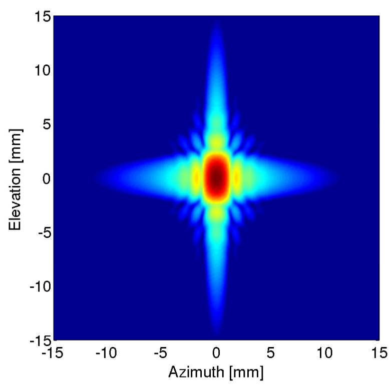
(c) 50% oversampling.



(d) 50% oversampling.

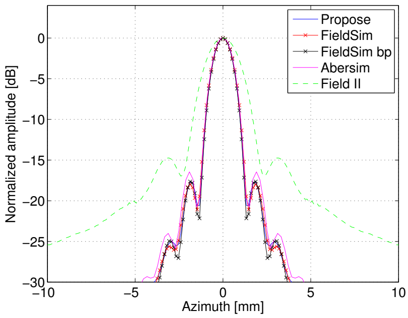


(e) Propose

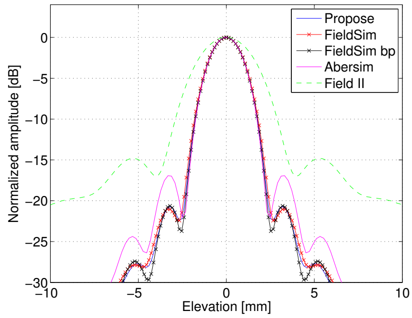


(f) FieldSim

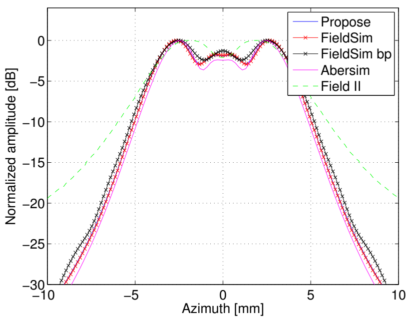
Figure 5.5: Results from Case 3: Non-linear simulation without attenuation. All profiles are calculated at focal depth 75 mm. **a)** Lateral RMS profile in azimuth with no oversampling. **b)** Lateral RMS profile in elevation with no oversampling. **c)** Lateral RMS profile in azimuth with 50% oversampling in spatial dimensions. **d)** Lateral RMS profile in elevation with 50% oversampling in spatial dimensions. **e)** Cross sectional RMS profile in focus from Propose. **f)** Same as e) but simulated by FieldSim.



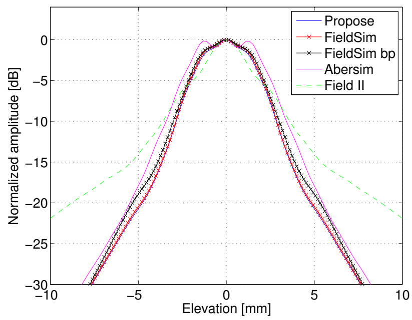
(a) 50% oversampling.



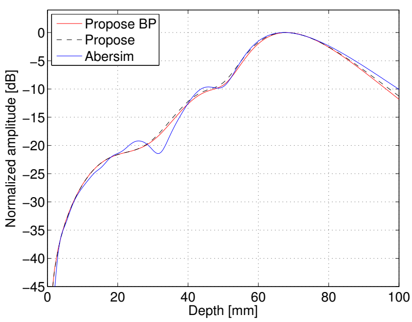
(b) 50% oversampling.



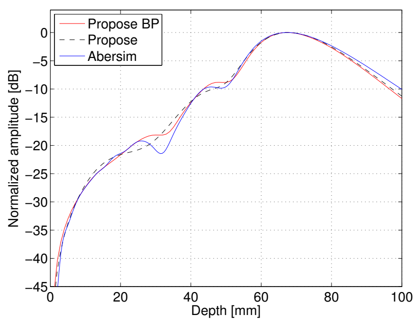
(c) 40 mm



(d) 40 mm



(e)



(f)

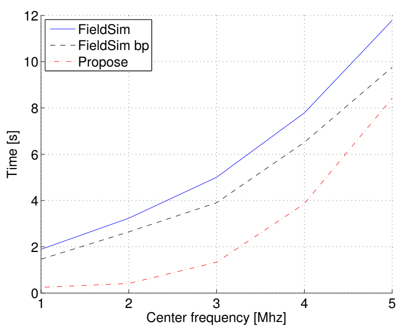
Figure 5.6: Results from Case 4: Non-linear simulation with attenuation. **a)** Lateral RMS profile at 75 mm in azimuth with 50% oversample in space. **b)** Lateral RMS profile at 75 mm in azimuth with 50% oversample in space. **c)** Lateral RMS profile at 40 mm in azimuth with 50% oversample in space. **d)** Lateral RMS profile at 40 mm in azimuth with 50% oversample in space. **e)** Axial RMS profile with no oversample in spatial dimensions. **f)** Same as e) with FieldSim bp oversampled 50%.

5.2 Speed evaluation

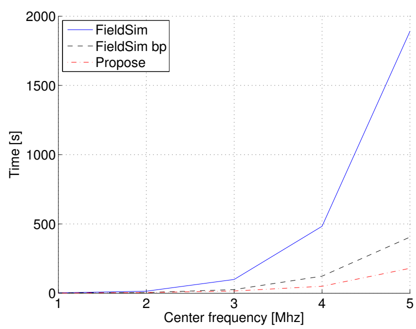
The results from the speed evaluation as explained in Section 3.4 are shown in Figure 5.7. The running time as a function of center frequency is plotted in Figure 5.7a and 5.7b for computation of the fundamental and harmonic field, respectively. For the fundamental field it can be seen that FieldSim introduces a overhead of about 2 seconds. With the band pass option enabled the simulation is 0.5 - 2 seconds faster than using the full frequency axis. As the center frequency increases, and thus the size of the computation, the overhead becomes less noticeable as more time is spent calculating the matrix multiplication in the Angular spectrum method.

For the non-linear case the effect of having complexity of $O(n^6)$ can clearly be seen as the running time increases rapidly for FieldSim using the full frequency axis. By changing the center frequency from 4 to 5 Mhz the time used for the computation increases from 8 minutes to 31 minutes and 30 seconds, a 4x increase. By using the band pass option the running time is drastically reduced, but not to the level of Propose.

It should be noted that in the range from 1 to 2.5 Mhz FieldSim using the band pass option and the standalone Propose offers the same performance, assuming that no oversampling is used. However, as the center frequency increases the relative performance becomes poor for FieldSim. Comparing Propose and FieldSim using the full frequency axis at 5 MHz shows a running time of Propose being almost 19 times faster.



(a)



(b)

Figure 5.7: Speed comparison between the FieldSim implementation and original Propose. "FieldSim bp" indicates that the bandpass option is turned on. **a)** Results for linear simulations. **b)** Results for non-linear simulations.

5.3 Watertank

This section presents the results of the measurements and simulations explained in Section 4.3. In the figures below "Propose" and "Propose BP" refers to Propse in FieldSim with and without the band pass option enabled, respectively.

Figure 5.8 shows the linearly simulated lateral RMS profiles and the measured lateral profile. Note that measured profile only has a side lobe on the left side. This can happen if the hydrophone is not exactly on the ultrasound beam axis. In this section the side lobes of the simulation will be compared against the left side lobe of the measurements.

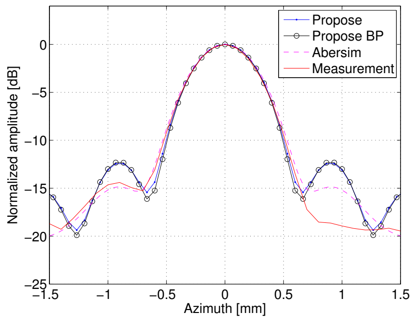
Using no oversampling the side lobes for the profile generated by Propose in FieldSim using the full frequency axis and the band limited axis is higher than the measured profile. The profile generated by Abersim is slightly below the measurement. The MSE calculated against the measurements is found to be below 1 dB for Abersim, below 3 dB for FieldSim in azimuth for both the band pass signal and the full axis. In elevation the measured profile lies above the simulated profiles with a MSE below 5 dB.

By using a oversampling factor of 2 the side lobe levels for the simulations done in FieldSim is in agreement with the measurement, however it can be seen from Figure 5.8c that the placement of the side lobes are closer to the center of the beam than the measurement. The average mismatch between the normalized profiles is found to be below 1.5 dB with 100% oversampling in the spatial dimensions.

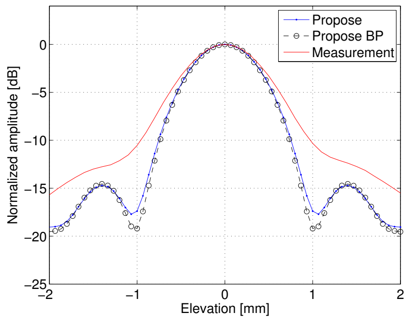
The results for the second harmonic field is shown in Figure 5.9. For the case where no oversampling is used the profile in azimuth can be seen to match the first side lobe on the left nearly perfect by Propose in FieldSim for both options. Abersim however lies slightly below the measured profile. The MSE is found to be below 4.5 dB and 10 dB in azimuth and elevation, respectively for FieldSim. For Abersim the MSE is found to be below 4 dB.

When applying 100% oversampling in FieldSim it can be seen from Figure 5.9c that the profiles simulated by FieldSim is nearly identical to Abersim. It should be noted that the profile generated by using the full frequency axis is more equal to Abersim than the band limited profile. The MSE with regards to the measurements are the same as in the case of using no oversampling in azimuth, but is found to be below 8 dB in elevation.

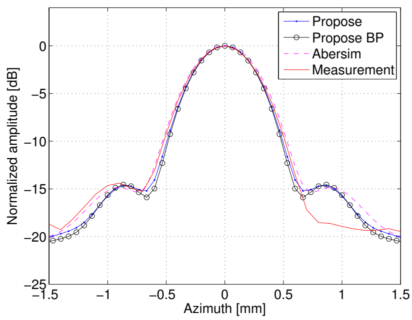
The time used to simulate the second harmonic field with no oversampling was 30 seconds for the band limited signal and 2 minutes and 30 seconds for the full frequency axis. With 100% oversampling in the spatial dimensions the time used was 5 minutes for the band limited frequency axis and 39 minutes for full axis. The MATLAB version of Abersim used two days to complete the simulation.



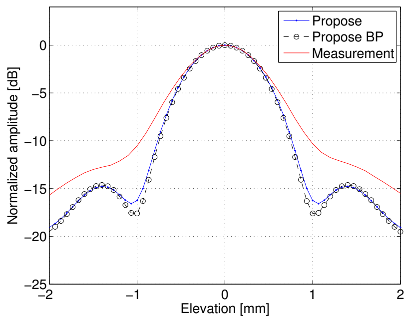
(a) No oversampling.



(b) No oversampling.

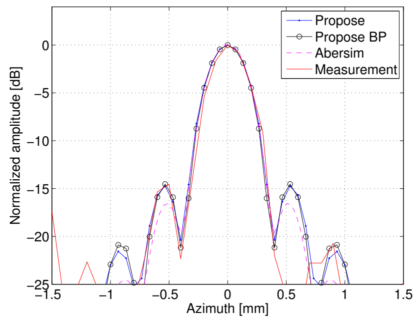


(c) 100% oversampling.

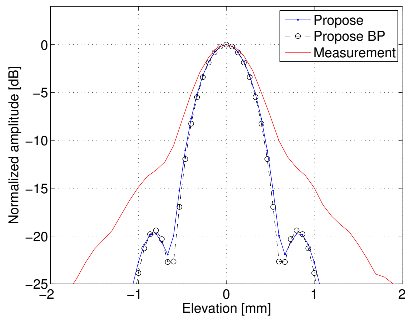


(d) 100% oversampling.

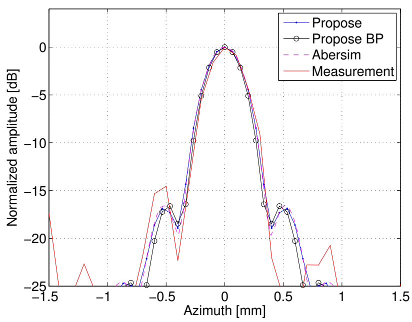
Figure 5.8: Simulated lateral RMS profiles versus measurements of the fundamental field. **a)** Lateral RMS profile in azimuth without oversampling. **b)** Lateral RMS profile in elevation without oversampling. **c)** Lateral RMS profile in azimuth with 100% oversampling. **d)** Lateral RMS profile in elevation with 100% oversampling.



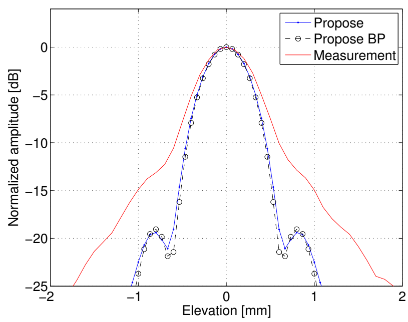
(a) No oversampling.



(b) No oversampling.



(c) 100% oversampling.



(d) 100% oversampling.

Figure 5.9: Simulated lateral RMS profiles versus measurements of the second harmonic field. **a)** Lateral RMS profile in azimuth without oversampling. **b)** Lateral RMS profile in elevation without oversampling. **c)** Lateral RMS profile in azimuth with 100% oversampling. **d)** Lateral RMS profile in elevation with 100% oversampling.

Chapter 6

Discussion

Propose has been integrated into the FieldSim framework. In this chapter the implementation of Propose in FieldSim and the results presented in Section 5 are discussed.

6.1 Results

The implementation of Propose in FieldSim is shown to produce near identical results to Field II, with and without attenuation, when using the full frequency axis option. The band limited option also gives satisfactory results, but introduces ripples effects in the spatial dimensions like Propose does, though not as distinct. By increasing the oversampling factor both versions of the FieldSim implementation show improved performance compared to Field II.

For computation of the second harmonic the results are shown to be similar in azimuth for Abersim, standalone Propose and FieldSim. In elevation the profile generated by Abersim has lower side lobe levels than Propose and FieldSim. Since the Abersim simulations was calculated using a different machine than Propose and FieldSim slightly different initial pressures were used. This may have caused the differences shown in elevation for Abersim compared to Propose and FieldSim.

When simulating the second harmonic field in focus there is little difference between using 0% and 50% oversampling in terms of MSE, however by increasing the oversampling factor the running time was increased by a factor 6 and 3 when using FieldSim and FieldSim bp, respectively. It should be noted that FieldSim with the band pass option enabled used the same amount of time as the standalone version of Propse in Case 3 when no oversampling was used. However, by choosing a smaller time domain than the one used in the simulations, Propose would be faster than FieldSim.

In Case 4 at 40 mm FieldSim using the full frequency axis is shown to be more in agreement to Abersim in azimuth than FieldSim using a band pass filtered initial pressure, while in elevation the difference is less obvious. This suggests that more accuracy can be achieved by using the full frequency axis, but as stated earlier the running time increases.

In general it is shown that by increasing the number of samples used in the simulation the results become more accurate and that FieldSim using the full frequency axis performs better than by using the band limited initial pressure. The cost paid for more accuracy is increased running time. As shown in Section 5.2 the running time, for computing the second harmonic, increases dramatically as a function center frequency, and implicitly as a function of number of samples in the simulation.

For linear simulations, where the increase is not so dramatic, the overhead of using FieldSim instead of the standalone version of Propose becomes clear. Internally FieldSim has to generate the initial pressure with Field II, decimate the pressure and then convert it to the K-space required by Propose. In addition FieldSim calculates the RMS, max envelope profiles and saves the propagated pressure field. All Propose does is to create a near optimal K-space directly and propagate the field. All post-processing is left up to the user. As the center frequency increases, and thus the number of samples, more time is used to calculate the propagation of the initial field and the overhead becomes less of a factor.

Comparing the MATLAB version of Abersim, FieldSim and Propose in terms of running times it is clear that the standalone version of Propose is the fastest tool. Comparing FieldSim against Propose the performance is at best the the same with no oversampling, in the 1 - 3 Mhz range and at worse 19 times slower than Propose for 5 Mhz. This factor will only become larger as the center frequency is increased. Comparing FieldSim against Abersim in the case of simulating lateral beam profiles at a given depth FieldSim is faster by a margin. In the non-linear test cases FieldSim was close to 500 and 3000 times faster with full frequency axis and band limited axis, respectively. When comparing simulating the whole beam profile from a range z_0 to z_1 the difference is not nearly as large. For the axial profile generated in Case 4, FieldSim, with the band pass option enabled and 50% oversampling used, was 38 times faster.

However, these results are overly optimistic because the comparisons are done against the *MATLAB* version of Abersim. The C version is more efficient and can take advantage of MPI[1] parallel computing resources[8]. In [14] the standalone version of Propose is found to be 1000 times faster for depths up to the focus point and 100 times faster at larger depths. Keeping this in mind Propose in FieldSim as presented is still faster than Abersim, but is probably more in the range of 5[2] to 1000[3] times, depending on the frequency used, oversampling factors chosen and if the full frequency axis is used or not.

[1] Message Passing Interface

[2] 100 divided by 19 (worst case) ≈ 5

[3] With no oversampling and a band limited initial pressure performance equal to standalone Propose was found

For the measurements done in the water tank of the fundamental field it can be seen that having no oversampling in space the side lobe level in the azimuth profiles generated by FieldSim are lower than that of the measurements and Abersim. With a oversampling factor of 2 the profiles generated by FieldSim are in agreement with the measurements. Due to problems finding the trigger delay from the research scanner the exact position of the hydrophone cannot be known. This introduces a source of error that effects the position of the hydrophone along the depth axis. Another source of error is the processing of defining the beam axis when setting up a measurement in ProbeLab. As can be seen from the measurements only the left side lobe is shown. This indicates that the hydrophone was not exactly positioned at the center of the ultrasound beam. In elevation it can be seen that the measurements have a different profile than the simulations. Because the effective pitch is slightly bigger than the physical pitch due to crosstalk between the elements. This combined with lens effects can cause the elevation focus to be different from the theoretical focus depth.

Looking at the second harmonic RMS profile in azimuth the profile generated by FieldSim without oversampling nearly perfectly matches the side lobe of the measurement, Abersim however lies below the measured profile. When the oversampling is increased to 100% FieldSim and Abersim generates very similar profiles. The reason for the mismatch could be from the beam axis being slightly off.

6.2 Propose in FieldSim

By integrating support for Propose in FieldSim it is now possible to calculated second harmonic fields in 3D in a relatively short time. By using some of the core features of the framework the strength of Propose and FieldSim can be combined to gain new possibilities. By using Field II to generate the initial pressure propagated by Propose features like arbitrary transducer geometries, measured impulse responses, custom excitation pulses and steering is now possible, even though it comes at a cost. The cost is increased running time compared to the standalone version of Propose.

In the current implementation of Propose in FieldSim the initial pressure generated by Field II is used directly. As shown in Figure 3.2 there are a lot of zero samples in the temporal dimension. One of the strengths of Propose is the possibility to define a computational domain in the temporal dimension which only covers the length of the pulse. This is not the case when using Field II in FieldSim as the pressure from Field II is zero padded to align recordings at the same starting time. This causes the initial pressure from Field II to have more samples than needed in the temporal dimension and in turn make the simulation in Propose slower.

As mentioned in Section 3.4 the number of operations done by Propose for the second harmonic calculation is on the order of $N_x^2 \cdot N_y^2 \cdot N_z^2$. By reducing the number of samples in the temporal dimensions by a factor of two the running time will be increased by a factor of 4, or by keeping the number of samples constant but using a temporal axis half the size the resolution is doubled without increasing the running time.

One way to remove zeros would be to crop the signal in such a way that only the wanted pulse is captured. This could be done by having the user define the extent of the temporal dimension like Propose does, but would fail for steering if the size is set to small. To calculate the bounding box for the initial pressure is not trivial since the initial depth is arbitrary and steering should also be taken into account. A practical approach could be to use envelope detection on the edge waves at the edges of the computational domain and let the lowest and highest temporal value define the start and end of the cropped temporal axis, respectively.

Another possible way to reduce the running time of Propose in FieldSim would be to use a small or no oversampling in time and upsample the signal by a user defined factor in post processing. This would decrease the time used to compute the second harmonic, but more time would be spent in post processing and in turn increasing the already noticeable overhead.

The overhead in FieldSim compared to Propose could be reduced by not automatically calculating RMS and max envelope beam profiles in post processing, however this is considered to be part of the FieldSim core. The only post processing offered by Propose is to interpolate the propagated pressure to a user defined grid.

Even though FieldSim offers a standardized way to run and set up simulations it is the underlying simulation tools that perform the calculations. By introducing features like steering and arbitrary transducers that are easy to use it is important that he user is aware of the underlying model used by the simulation tool chosen. Propose uses the Angular spectrum method implemented using the discrete Fourier transform. This can result in periodical effects if the computational domain is chosen too small while steering or using transducers that cause grating lobes to appear.

A link between FieldSim and the Ultrasonix research scanner was developed as explained in Chapter 4. However, because of problems with getting PyTexo to read the MATLAB generate HDF5 file correctly it was not used in the measurements presented in Section 4.3. As PyTexo supports both reading and writing to HDF5 files the FieldSim generated files could be parsed by a Python script that converts all the one-dimensional arrays containing scalar values to simple scalars, then write the converted data back to a new HDF5. Another solution would be to include a specific read function in PyTexo that does this automatically.

By offering a link to the research scanner comparing simulations and measurements will become easier. By converting a FieldSim setup to a real measurement scenario the need to know PyTexo and how to program the scanner is no longer needed. This will make the use of the research scanner more accessible than with the present solution of having to manually program the scanner. FieldSim can also be used as a tool to configure PyTexo, no simulations are required.

Chapter 7

Conclusion

The Propose simulation tool has been implemented in the FieldSim framework and shown to produce satisfactory results in terms of beam profiles compared to the known simulation tools Field II and Abersim. By implementing Propose in FieldSim it is now possible to simulate second harmonic fields with the FieldSim framework.

By using Field II to initialize Propose it is now possible to easily simulate pressures generated from a transducer with a measured impulse response and custom excitation pulse. Steering is also possible by taking the advantage of FieldSims ability to rotate pressures generated by Field II.

The standalone version of Propose is substantially quicker than Abersim for non-linear simulations, but also more inaccurate. The implementation developed in this thesis is shown to be just as quick as Propose at best, but in most cases slower. Even though the running times compared to Propose is found to be slow, when compared to Abersim the method is still substantially quicker.

The running time is strongly dependent on the center frequency used in the simulation and the complexity of the algorithm for calculating the second harmonic field is shown to be $O(n^6)$ using asymptotic notation. The method would be best suited for simulations in areas like cardiac ultrasound where lower frequencies are used.

A link between the FieldSim framework and a research scanner has been developed by using the PyTexo software. FieldSim simulations can be converted to a file format readable by PyTexo. However problems with compatibility stopped the link to be previewed. When finalized it will be possible to convert a FieldSim simulation to a file that automatically initializes the research scanner with an identical setup. This will make the research scanner and the water tank more accessible to users not familiar with the research scanner.

Chapter 8

Further work

This chapter lists possible further work and ideas surrounding the topics discussed in this thesis.

- One of the problems with the implementation presented in this thesis is the aspect of running time. Further work could be done on optimizing the generation of the initial field with Field II. This could be done by reducing the number of unnecessary zeros in the signal as discussed.

  An other option is optimizing the standalone version of Propose by implementing a version in C/C++. The calculation of the second harmonic would be a lot faster as the implementation today involves a triple for-loop in MATLAB. Optimization could also be done by looking at the possibility of paralleling parts of the program. This could, for example, be done by using Jacket[1] which makes it possible to do MATLAB calculations on the GPU. By utilizing the parallelism found on GPUs it would be possible to simulate multiple depths of interest at the same time.

- The link between FieldSim and the research scanner is in the early stages of development and has great potential. Further work could consist of implementing the support for B-mode scans with dynamic focusing based on a FieldSim simulation setup.

  By taking the idea of the FieldSim link further a possibility could be to link FieldSim to ProbeLab as well. Letting FieldSim control the setup of ProbeLab and allowing for automatic measurements based on the simulation set up in FieldSim.

  Tools could be built to take both a FieldSim simulation and data measured by ProbeLab and offering a interface for presenting, comparing and analysing the data. This would make comparison between simulations and measurements less time consuming and easier to do.

[1] http://www.accelereyes.com/products/jacket

Bibliography

[1] J. A. Jensen, "Linear description of ultrasound imaging systems, notes for the international summer school on advanced ultrasound imaging," tech. rep., Technical University of Denmark, 1999.

[2] J. A. Jensen, *Estimation of Blood Velocities Using Ultrasound: A Signal Processing Approach.* Cambridge University Press, 1996.

[3] J. A. Jensen, "Field: A program for simulating ultrasound systems," in *10th Nordic-Baltic Conference on Biomedical Imaging*, vol. 34, pp. 351–353, Medical & Biological Engineering & Computing, 1996.

[4] J. A. Jensen, "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers," 1992.

[5] A. D. Pierce, *Acoustics: An Introduction to Its Physical Principles and Applications.* Acoustical Society of Amer, June 1989.

[6] L. E. Kinsler, A. R. Frey, A. B. Coppens, and J. V. Sanders, *Fundamentals of Acoustics.* New York: John Wiley & Sons, 1982.

[7] G. R. Harris., *Transient field of a baffled planar piston having an arbitrary vibration amplitude distribution*, vol. 70, pp. 186–204. J. Acoust. Soc. Am., 1981.

[8] "http://www.ntnu.no/isb/abersim," June 2012.

[9] H. Kaupang, *Abersim 2.x Reference manual with tutorials*, 2008.

[10] T. Varslot and S.-E. Måsøy, "Forward propagation of acoustic pressure pulses in 3d soft biological tissue," *Modeling Identification and Control A Norwegian Research Bulletin*, vol. 27, pp. 181–190, 2006.

[11] G. Taraldsen, "A generalized westervelt equation for nonlinear medical ultrasound," 2001.

[12] G. Taraldsen and T. Varslot, "Computer simulation of forward wave propagation in soft tissue," *Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 2005.

[13] H. Torp, T. Johansen, and J. Haugen, "Nonlinear wave propagation - a fast 3d simulation method based on quasi-linear approximation of the second harmonic field," in *Ultrasonics Symposium, 2002. Proceedings. 2002 IEEE*, vol. 1, pp. 567 – 570 vol.1, oct. 2002.

[14] F. Prieur, T. F. Johansen, S. Holm, and H. Torp, "Fast simulation of second harmonic ultrasound field using a quasi-linear method.".

[15] M. F. Hamilton and C. L. Morfey, "Model equations," in *Nonlinear Acoustics* (M. F. Hamilton and D. T. Blackstock, eds.), ch. 3, pp. 41–63, Academic Press, 1988.

[16] M. F. Hamilton and C. L. Morfey, "Sound beams," in *Nonlinear Acoustics* (M. F. Hamilton and D. T. Blackstock, eds.), ch. 8, pp. 233–261, Academic Press, 1988.

[17] Y. Jing, M. Tao, and G. T. Clement, "Evaluation of a wave-vector-frequency-domain method for nonlinear wave propagation," 2011.

[18] J. W. Goodman, *Introduction to Fourier optics*, ch. 4, pp. 74–75. Roberts and Company Publishers, 3 ed., 2005.

[19] J. A. Jensen, "Speed-accuracy trade-offs in computing spatial impulse responses for simulating medical ultrasound imaging," 2001.

[20] "http://www.mathworks.se/help/toolbox/signal/ref/decimate.html," June 2012.

[21] "http://server.oersted.dtu.dk/personal/jaj/field/?faq.html," June 2012.

Appendices

Appendix A

Texo parameters

This appendix lists the possible parameters the Texo interface can control. The data types are given in C style, meaning that `int` is a single integer while `int[128]` is an array of 128 integers.

Table A.1: List of transmit parameters in Texo

Parameter	Data type	Description
`useManualDelays`	`bool`	Manual time delay usage (true = on, false = off)
`manualDelays`	`int[129]`	Time delays for up to 128 elements, with the maximum delay at the end
`focusDistance`	`int`	Focus distance in microns
`aperture`	`int`	Aperture size in elements
`frequency`	`int`	Frequency in Hz
`speedOfSound`	`int`	The speed of sound in m/s
`pulseShape`	`char[MAX]`	Pulse shape in codes of '+', '-', or '0'. The pulse shape can be maximum `MAXPULSESHAPES` number of characters
`angle`	`int`	Angle in 1/1000th of a degree
`centerElement`	`int`	Center element in 1/10th of an element
`useDeadElements`	`bool`	Enable deadElements (true = on, false = off)
`deadElements`	`int[128]`	0 = not dead, 1 = dead
`trex`	`bool`	Use transmit extension, which enables continuous transmit over a certain line duration
`tableIndex`	`int`	Transmit table index

Table A.2: List of receive parameters in Texo

Parameter	Data type	Description
`maxApertureDepth`	`int`	Maximum aperture curve depth in microns
`useManualDelays`	`bool`	Manual time delay usage (true = on, false = off)
`manualDelays`	`int[129]`	Time delays for up to 128 elements, with the maximum delay at the end
`centerElement`	`int`	Center element in 1/10th of an element
`aperture`	`int`	Aperture size in elements
`saveDelay`	`int`	Depth in microns to delay the saving of data
`applyFocus`	`bool`	Computed time delay usage (true = on, false = off) Cannot be used with manual delays.
`tgcSel`	`int`	Tgc selection by index
`speedOfSound`	`int`	The speed of sound in m/s
`lgcValue`	`int`	Digital gain value applied to entire scanline (0 - 4095)
`customLineDuration`	`int`	Custom line duration in ns
`acquisitionDepth`	`int`	Acquisition depth in microns
`angle`	`int`	Angle in 1/1000th of a degree
`channelMask`	`int[64]`	Channel masking for up to 64 elements (0 = on, 1 = off)
`decimation`	`int`	Decimation value for RF or Bmode data
`numChannels`	`int`	Number of channels (32 or 64)
`tableIndex`	`int`	Transmit table index

Table A.3: List of TGC paramters Texo defining a TGC curve defined by the points (0,`top`), (`vmid`,`mid`), (100,`btm`).

Parameter	Data type	Description
`top`	`int`	Top point defined in percent, always near the transducer.
`mid`	`int`	Middle point defined in percent between `top` and `btm`.
`btm`	`int`	Bottom point defined in percent, always at `maxApertureDepth`.
`vmid`	`int`	Point defined between 0 and 100 percent, 100% is at `maxApertureDepth`.
`depth`	`int`	Scale parameter.
`percent`	`int`	Scale parameter.