

Optimization of software for modeling acoustic monitoring systems

Monitoring polar areas for climate change

Frida Tryggestad Klockmann

Master's thesis in Software Engineering at
Department of Computing, Mathematics and
Physics,
Bergen University College
Department of Informatics,
University of Bergen

June 2020



Western Norway
University of
Applied Sciences



Abstract

To understand the rapid changes in the climate and in the ocean underneath the ice in the Arctic, it is important to get more and better observations from this area. The Nansen Environmental Remote Sensing Center (NERSC) has an existing program that is meant for analyzing and model acoustic propagation. This thesis describes the process of improving this package by adding several acoustic models to the program, adding functionality for saving executions, adding new ways to visualize the data and improving the graphical user interface (GUI).

The thesis also describes how we made it possible to compare simulations done on several time steps and how one can visualize how the sound propagates from one source to several receivers surrounding it in a circle.

Acknowledgments

I would like to thank my supervisors at Western Norway University of Applied Sciences (HVL), Rogardt Heldal and Kjell-Eivind Frøysa, for their guidance throughout this thesis. Their frequent feedback and assistance has helped me greatly. I would also like to give a special thank Espen Storheim for meeting with me almost every week throughout this process and and Hanne Sagen for her engagement in this project. Without their assistance, expertise and feedback, this thesis would not have been possible. Nansen Environmental and Remote Sensing Center has also provided us with office space on their premises, which we have been very grateful for.

I would also like to thank my fellow master student Emilia Botnen Van den Bergh for a good cooperation on the programming part of this project. Her help and assistance has helped me greatly, and discussing things with her has helped me to get a better overview of what we have been doing through the project.

Contents

1	Introduction	1
1.1	Research questions	3
1.2	Thesis Outline	3
2	Background	5
2.1	Underwater acoustic networks	5
2.2	The Arctic Package	6
2.3	Acoustic Propagation Models	7
2.3.1	Eigenray	9
2.3.2	Kraken	12
2.3.3	Range-dependent Acoustic Model	12
2.3.4	Bellhop	15
3	Related Work	17
3.1	SOLWEIG	17
3.2	MERLIN	18
3.3	CAATEX	19
4	Methodology: Design science	21
4.1	The seven guidelines	21
4.1.1	Guideline 1: Design as an Artifact	22
4.1.2	Guideline 2: Problem Relevance	22
4.1.3	Guideline 3: Design Evaluation	23
4.1.4	Guideline 4: Research Contributions	23
4.1.5	Guideline 5: Research Rigor	24
4.1.6	Guideline 6: Design as a Search Process	24
4.1.7	Guideline 7: Communication of Research	24
4.2	The four quadrants	25
5	Design and Implementation	28
5.1	MATLAB	28

5.2	Architecture	29
5.3	The User Interface	33
5.4	Visualizations	36
6	Results and Interpretation	40
6.1	Saving simulation results	40
6.2	User Interface	41
6.3	The new models	46
6.4	Time Variation	49
6.5	Simulating in space	52
6.6	Visualization of saved results	55
6.7	Answer to Research Question 1	58
7	Feedback from NERSC	60
7.1	The Interviews	60
7.2	Interpretation of interview answers	63
7.3	Discussion	64
7.4	Answer to Research Question 2	65
8	Threats to Validity	66
8.1	Construct validity	66
8.2	Internal validity	67
8.3	External validity	67
8.4	Reliability validity	67
9	Conclusion	69
10	Further Work	71
10.1	MATLAB	71
10.2	GUI	72
10.3	Machine Learning	72
10.4	Visualizations	72
10.5	More	73

List of Figures

2.1	Acoustic tomography with four sources (S) and 5 receivers (R), giving 20 different acoustic paths[1].	6
2.2	A map of the Arctic with a path from a source to a receiver in which the following models will simulate on.	8
2.3	The three panel visualization of results from Eigenray. The upper panel contains arrival angles for each path, the middle panel shows the ray paths of the eigenrays and the bottom panel shows the SSPs for the path.	10
2.4	The three panel visualization of results from Eigenray with time fronts. The upper panel contains information about number of turning points for each path, the middle panel shows the eigenrays time fronts and the bottom panel shows the SSPs for the path.	11
2.5	Krakens three-panel plot. The upper plot contains a turning point filter, the middle plot displays the time-fornts and the bottom plot contains SSPs.	13
2.6	Visualization of results from Ram. It's a two panel plot whereas the upper plot shows the transmission loss, and the bottom panel shows the SSPs.	14
2.7	This is one of the possible results from running Bellhop. The upper panel shows the ray paths, while the bottom plot shows the SSPs.	16
4.1	The Generate/Test Cycle.	25
4.2	The four quadrants in DSR.	26
5.1	Block diagram of the original version of Arctic package.	30
5.2	Block diagram of the Arctic package after adding more models, saving and other functionalities.	31
5.3	An overview of how the plotZRT-file handles several receivers and time steps.	38

5.4	Screenshot of the old GUI.	39
6.1	The button for saving the results. Here shown for a simulation of transmission loss with RAM.	41
6.2	This is the old GUI on Windows.	42
6.3	The new GUI on mac OS.	43
6.4	The new GUI on Windows.	44
6.5	Window for advanced settings for sources and receivers.	45
6.6	Sepraate windows for model specific input for Eigenray and RAM.	46
6.7	Example of a result when running MPIRAM	47
6.8	Example of a result when running RAM	48
6.9	The drop down where the user can choose to run the model over several time steps	50
6.10	A visualization of Eigenrays Arrival time for the different months.	51
6.11	The window that controls information about receivers when running a model on several receivers	52
6.12	The map with 100 receivers surrounding a source with a radius of 75 km.	53
6.13	The transmission loss from a source to several receivers in a circle surrounding it.	54
6.14	The transmission loss form a source to several receivers in a circle surrounding it.	56
6.15	The eigenrays plotted on top of the sound speed profile. Bathymetry shown in black.	56
6.16	The transmission loss on a given depth as a function of range.	57
6.17	The average transmission loss on a path from the surface to a given depth (here 800 m), to a the shallowest point on the path and to 4500 m.	58

Chapter 1

Introduction

Climate change is the most significant environmental challenge the world has ever faced [2]. The melting ice of the Arctic is a barometer for the global warming that may cause unimaginable damage to our planet [3]. The extent of these changes over individual regions will vary over time and the ability of the different systems to mitigate or adapt to the changes [4]. The Arctic is an especially vulnerable area, as it is covered in ice which reflects the solar energy. When the ice is melting, it exposes a darker surface and increases the amount of solar energy absorbed in this area [5]. Scientists predict there may be virtually no summer sea ice in the Arctic within a generation [6]. This will bring enormous changes and challenges not just to the Arctic, but also to the entire world [3].

To understand what is happening in the Arctic area, we are in need of improved ocean-ice-atmosphere data. There is a lack of such observations from the ocean under the Arctic sea ice. To improve this, Nansen Environmental and Remote Sensing Center (NERSC) are using multipurpose acoustic networks to measure in this area. These acoustic networks are used to, among other things, measure the average temperature in the under-ice sea over distances ranging from 100 to 2500 km. Analyzing the results of these measurements, such as ocean temperature, is complicated due to the vast amount of complex data. Theories and acoustic models support measurements, since they tend to be better behaved and more consistent than experiments, and they are useful to acquire better knowledge about the underlying physical principle [7].

There exists a package called Arctic Package for acoustic propagation modeling. This package is produced at NERSC and contains tools using theories

and acoustic models to support the calculation of measurements such as the average temperature and sound propagation. The Arctic Package has previously been used to simulate sound propagation in the design phase of acoustic experiments. Acoustic modeling is an important tool for planning experiments, as well as in processing and analysis of results from e.g. acoustic thermometry experiments. Various models are available, based on different mathematical principles.

These tools were made for the purpose of running the calculations and models, and not necessary to have user-friendly tools — a bit *ad hoc* produce by a non-software developer. The need for making user-friendly tools have become more and more critical, to simplify the task of using the tool, permit more users, and to permit new calculations and visualizations.

This master thesis, based on software engineering principles, will describe the process of designing and implementing more user-friendly and extendable tools than used today based on the existing tools in the Arctic Package written in MATLAB.

The key issues were to obtain better user interface, an improved interface towards environmental data from databases and ice-ocean model systems as well as the ability to run more acoustic models. This work was done in collaboration with NERSC. NERSC is an independent non-profit research foundation conducting basic and applied environmental and climate research [8]. They are leading or involved in several projects investigating and map marine, cryospheric and atmospheric conditions using, among other methods, polar acoustic measurements [9].

To be able to assess the role of the ocean in climate variability and changes, monitoring and understanding the rapid changes underway in the Arctic Ocean and in the Southern Ocean surrounding Antarctica are of crucial importance [10]. Due to these changes, human activities will increase in polar areas [11]. This puts pressure on the vulnerable Arctic environment.

In order to help the issue, we need to know more about what is happening in the Arctic area. The Arctic Package can help the researchers plan their experiments as well as analyzing the results they get from their experiments. By simplifying the graphical user interface of the package it will improve the usability of the program. The users may therefore be able to complete their tasks faster and more efficiently and with less confusion. The process of plotting the different results and models in such a way that it is easy to visualize and compare has been both complicated and tedious. With an improved program this process can be more efficient and therefore leave the

scientists with more time to focus on their research.

The essence of the work is further development of the Arctic Package to make the researchers job easier. The primary users of the tools will be the researchers and students at NERSC, but in the long run, we also hope other research groups or organizations interested in measurements of the ocean will start using them.

The research and development of the package is done in collaboration with Emilia Botnen Van den Bergh, who has also written a thesis about this process from her perspective [12]. Her thesis is not published at the time of writing this thesis, however the title of her thesis will be: “Further development of a software for acoustic propagation modelling”.

1.1 Research questions

The research questions we want to answer in this thesis:

1. How well can existing tools in the Arctic Package written in MATLAB be improved regarding usability and extendibility?
2. How well can the way of working at NERSC be improved by improving the tools in the Arctic Package?

1.2 Thesis Outline

Introduction: Chapter 1 provides a brief introduction to this thesis. The goal of the thesis is given along with the research questions.

Background : Chapter 2 describes the background information needed for this thesis. Information about the the Arctic Package and the different acoustic models is given.

Related work: Chapter 3 presents a few articles and projects that are related to this thesis.

Methodology: Chapter 4 gives information about the design science methodology which is used in this project.

Design and Implementation: Chapter 5 presents the design of the framework and describes the implemented solution.

Results and Interpretation: Chapter 6 presents the results and the new features added to the arctic package. The results are interpreted and the chapter looks at how the tool can be used. In this chapter the first research question is answered.

Feedback from NERSC: Chapter 7 presents the interviews held with the researchers at NERSC and discusses the answers. The chapter also answers the second research question.

Threats to Validity: Chapter 8 gives a description of four types to classify potential threats to validity in this study.

Conclusion: Chapter 9 gives a conclusion on the thesis as a whole.

Further work Chapter 10 gives a summary of improvements and extensions that can be implemented in further work.

Chapter 2

Background

2.1 Underwater acoustic networks

Underwater acoustic networks consists of a given number underwater sound sources and listening devices, called hydrophones. These instruments are often referred to as sources and receivers and can be placed up to thousands of kilometers apart form each other, depending on the frequency of the source.

The main goal for the networks referred to in thesis is to measure the average temperature over large distances in the ocean underneath the ice in the Arctic. To measure the temperature of the water, the time it takes from a sound signal is sent from a source until it is received by the receiver is measured [1]. By knowing the distance between source and receiver, the sound speed can be calculated. The speed of sound in water depends on the temperature of the water, the salinity and pressure. The speed will increase with higher temperatures, more salt and higher pressure. The pressure is directly dependent on the depth.

With precise measurements from numbers of crossing acoustic paths it is possible to generate ocean temperature maps. This is called acoustic tomography [1]. There is an example of this in Figure 2.1. When assuming that the shaded area on the figure is warmer than the surroundings, the sound signal going through this area will have a higher speed than the ones that does not travel through this area. By combining these travel times, it is possible to conduct a map that displays where the warm and cold areas are.

In the Arctic, placing a lot of sources and receivers in the ocean is more

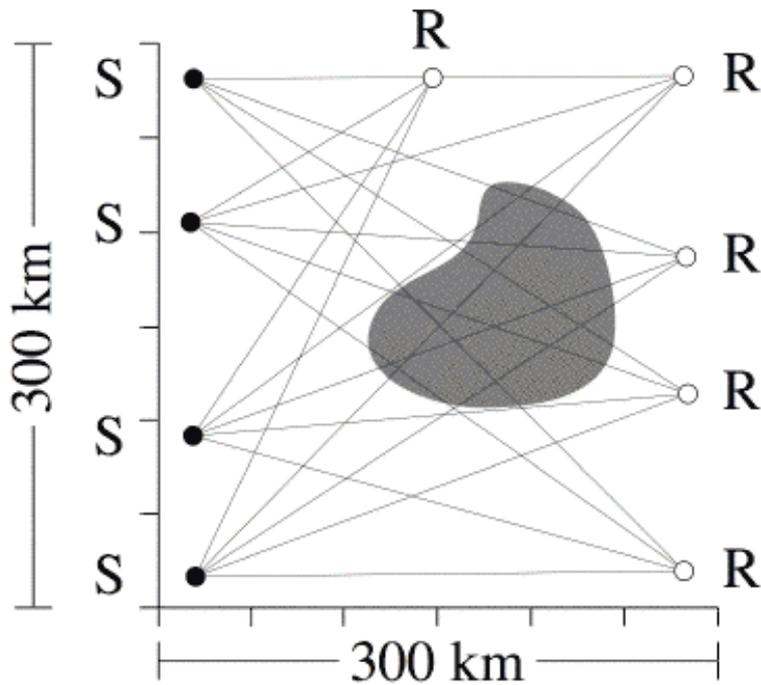


Figure 2.1: Acoustic tomography with four sources (S) and 5 receivers (R), giving 20 different acoustic paths[1].

complex and more expensive than other places because of the lack of infrastructure, the rough climate and the sea ice. Therefore it is not always feasible to have the number of sources and receivers required to make a detailed ocean temperature maps. However, even if there is just one or very few such paths, they are still very helpful and can be used to measure the mean temperature on these single paths. By finding the mean temperature, one can observe large seasonal changes in the ocean temperature. By continuing to do so for many years, these measurements can be used to detect large-scale climate changes in the ocean.

2.2 The Arctic Package

The Arctic Package is a software package that is meant to make acoustic modeling easy. It is installed on a Linux machine and the software is mainly written in MATLAB. The Arctic Package was at the time being made to simplify modeling by automatically retrieve data, execute models and plot

results.

The program has a main GUI where the user selects a map and where sources and receivers are to be placed. In addition to this, the main GUI also contains choices for input on the Eigenray and Kraken models, which are the two models that originally was included in the package. The map is shown in a separate dialog window. When the user presses the “get predict”-button, he or she has to choose from which source to which receiver he or she wants to simulate the model on. The results of the simulation is visualized in a separate window.

There are other acoustic models that are well established in the research field as well, but they were not available in the original version of The Arctic Package.

2.3 Acoustic Propagation Models

The transmission of sound in the ocean, is much more complicated than in the simplest situation in air. The interactions between the sound and the water results in reflection, bending (refraction) and scattering of the sound [13]. Sound waves traveling through the ocean are bending whenever it encounters changes in the speed of sound. Because the sound speed changes with temperature, salinity, and pressure, a sound wave will refract as it moves through the ocean [14]. Because of reflection, refraction and scattering of the sound, the sound does not travel in a straight line from the sound source to the hydrophones on the receiver. Therefore one cannot simply measure the distance from source to receiver when calculating the sound speed.

The complexity of sound transmission raises the need for acoustic models to simulate sound propagation in the design phase of the experiments, i.e. to optimize the location and depths of the sources and receivers, and to understand the data received from the experiments. The Arctic Package is a software package that is meant to make acoustic modeling easy. It is installed on a Linux machine and the software is mainly written in MATLAB. The next sections will describe some of the models that are used in, or that are added to, the Arctic Package. There will be an example of how the result will look like for each of the models, when running on the path shown on the map in Figure 2.2. The sound is sent from the source which is the top of the path on the map, and to the receiver which is on the bottom.

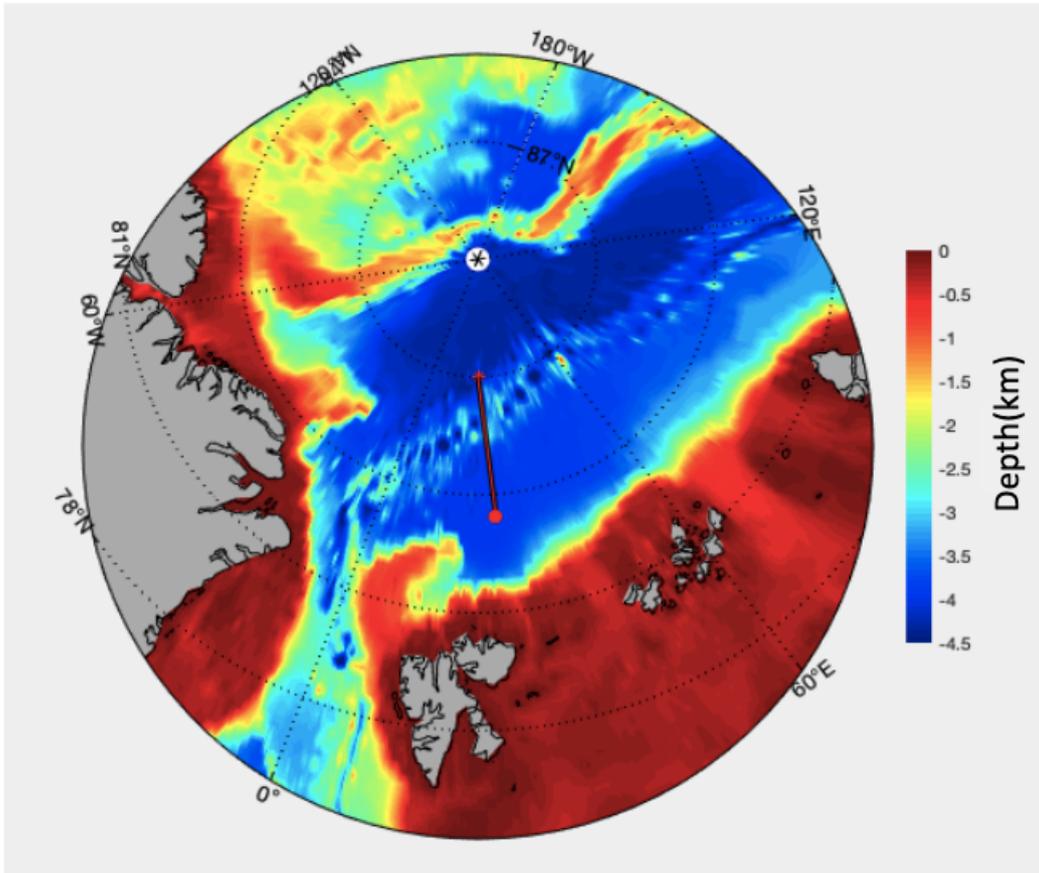


Figure 2.2: A map of the Arctic with a path from a source to a receiver in which the following models will simulate on.

2.3.1 Eigenray

The goal of the Eigenray model is to calculate smooth acoustic wavefront propagation through a realistic model of the ocean [15]. The wavefront is the set of all points where the wave has the same vibration. The Eigenray program that is used in our package is a ray-based acoustic prediction program. It is written by Brian Dushaw. It is designed to find eigenrays at long ranges accurately and efficiently by using sophisticated logic and exhaustive search [16]. The model creates accurate calculations of ray paths through range-varying ocean sound channel [15]. A sound channel is a horizontal layer in the ocean in which the sound speed is at a minimum, and it acts like a wave-guide for the sound [17].

The Eigenray program is written in Fortran with a view toward parallel processing. Because of this, and its sophisticated logic, the Eigenray program can potentially process difficult cases quickly [16].

Plotting of Eigenray predictions results in a three-panel plot. This is shown in Figure 2.3, where the model is running on the path pictured in Figure 2.2. The upper panel shows the different arrival times of the rays and the angle in which the sound is received by the hydrophone with respect to the horizontal. The middle panel contains a plot of the bathymetry from the sound source to the receiver as well as the plot of ray paths of the eigenrays found by the Eigenray program. The term “bathymetry” is the ocean’s depth relative to sea level, and describes the depths and shapes of underwater terrain [18]. Bathymetric maps illustrate the land that lies underwater in the same way that topographic maps represent the three-dimensional features (or relief) of overland terrain [18].

The bottom panel contains the same bathymetry plotting as the middle panel, and sound speed profiles (SSPs) along the line from source to receiver.

When running the Eigenray program, one can also choose to run it with timefronts. If one chooses to do so, the result will be slightly different. This is visible on Figure 2.4. The top panel contains a plot of number of turning points of each eigenray vs. travel time and the angle in which the sound is received by the hydrophone with respect to the horizontal. The turning-point filter is a technique used when comparing measurements [19]. The middle panel shows the eigenray timefronts. That is a distribution of ray travel times at the observation range. Every blue dot represents one ray presented in the time-depth plane. This has resulted in a visualization of 2492 individual rays.

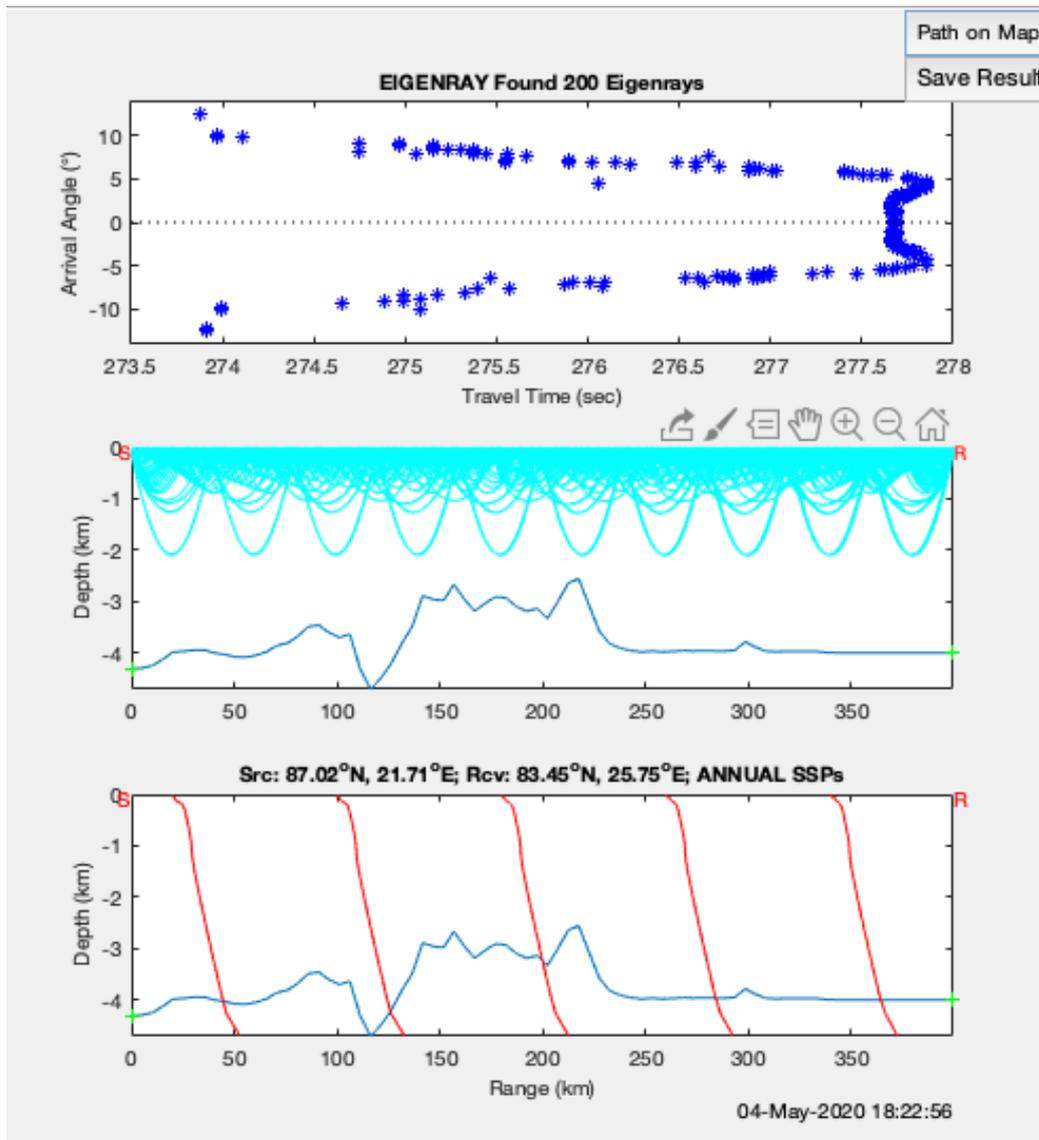


Figure 2.3: The three panel visualization of results from Eigenray. The upper panel contains arrival angles for each path, the middle panel shows the ray paths of the eigenrays and the bottom panel shows the SSPs for the path.

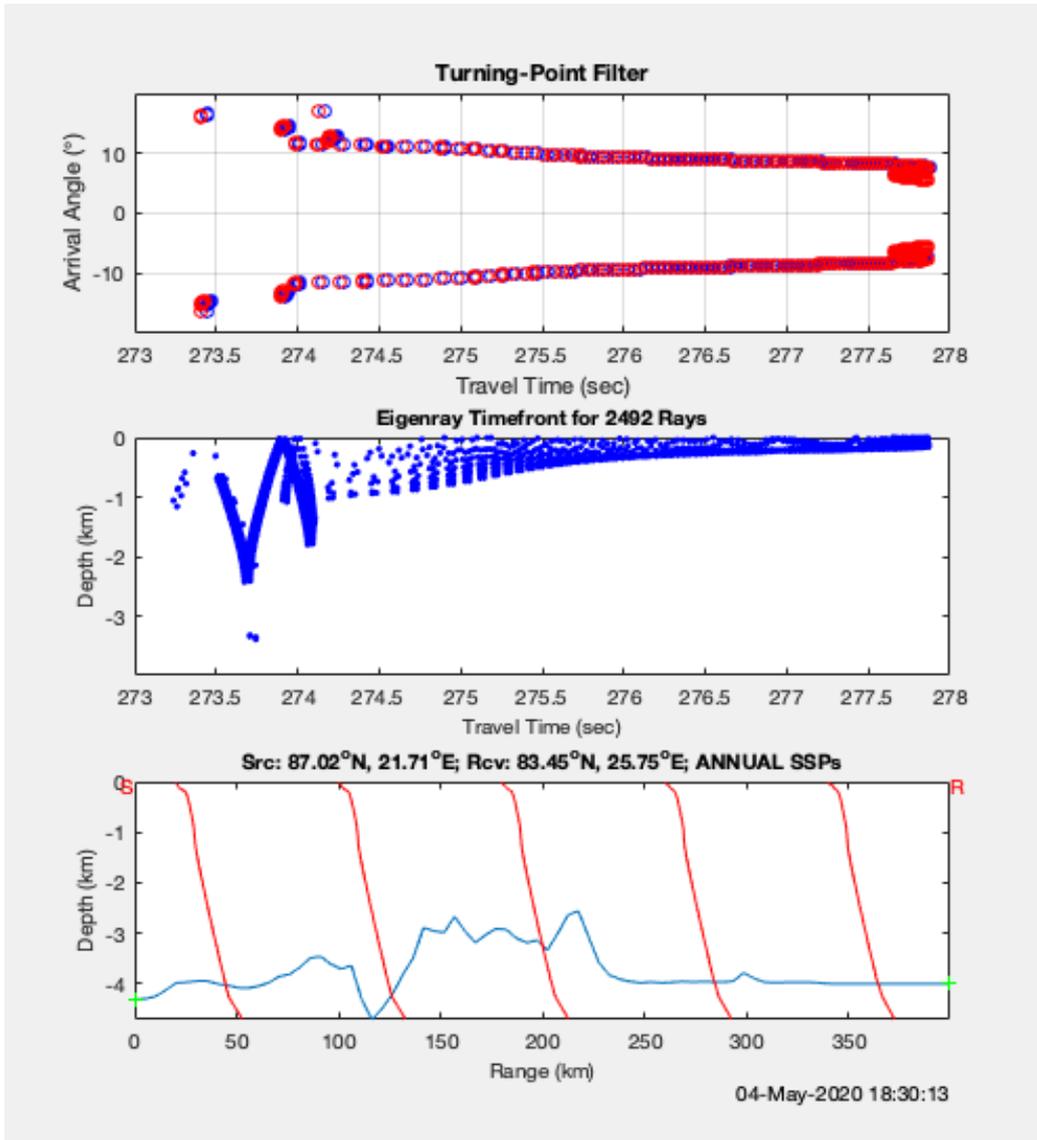


Figure 2.4: The three panel visualization of results from Eigenray with time fronts. The upper panel contains information about number of turning points for each path, the middle panel shows the eigenrays time fronts and the bottom panel shows the SSPs for the path.

The bottom panel is the same as the bottom panel in Figure 2.3.

2.3.2 Kraken

Kraken is another program that was already in the Arctic Package. It is a normal mode method. Normal mode methods have been used for many years in underwater acoustics. Kraken was developed with the objective of developing a normal mode model which was more robust, accurate and efficient [20]. The version of Kraken that is included in the Arctic Package, MATLAB-KRAKEN, is Brian Dushaws version. It has a MEX wrapper which makes it easier to execute from MATLAB. MEX (“MATLAB executable”) files are computer files that provides an interface between MATLAB and functions written in C, C++ or Fortran.

Running the Kraken program on the path shown in Figure 2.2 results in a three-panel plot, which you can see in Figure 2.5.

The upper panel displays a turning point filter. The middle panel displays the time front. Just like the in the time-front version of Eigenray, this is a distribution of ray travel times at the observation range. The bottom panel contains the bathymetry plotting and sound speed profiles (SSPs) along the bathymetry from source to receiver as well as normalized mode shapes for the five first modes.

2.3.3 Range-dependent Acoustic Model

The Range-dependent Acoustic Model, usually referred to as RAM, is a FORTRAN code based on the latest techniques in the parabolic equation (PE) modeling. PE methods represent one-way propagation and is widely used in radio wave propagation modeling [21]. The model is written by Michael D Collin and was not originally a part of the Arctic Package.

The result of plotting the result from RAM is a 2 panel plot. Figure 2.6 shows the result after running the model on the same path as in Eigenray and Kraken. The bottom plot is exactly the same as in Eigenray, but the upper plot, is a lot more interesting. It shows the propagation loss from the sound source in the depth-range plane. Propagation is the same as transmission loss and is the reduction in sound pressure at a given point from a source. The yellow/green line at the very top of the model shows a sound channel. Low frequency sound waves within the channel may travel thousands of miles

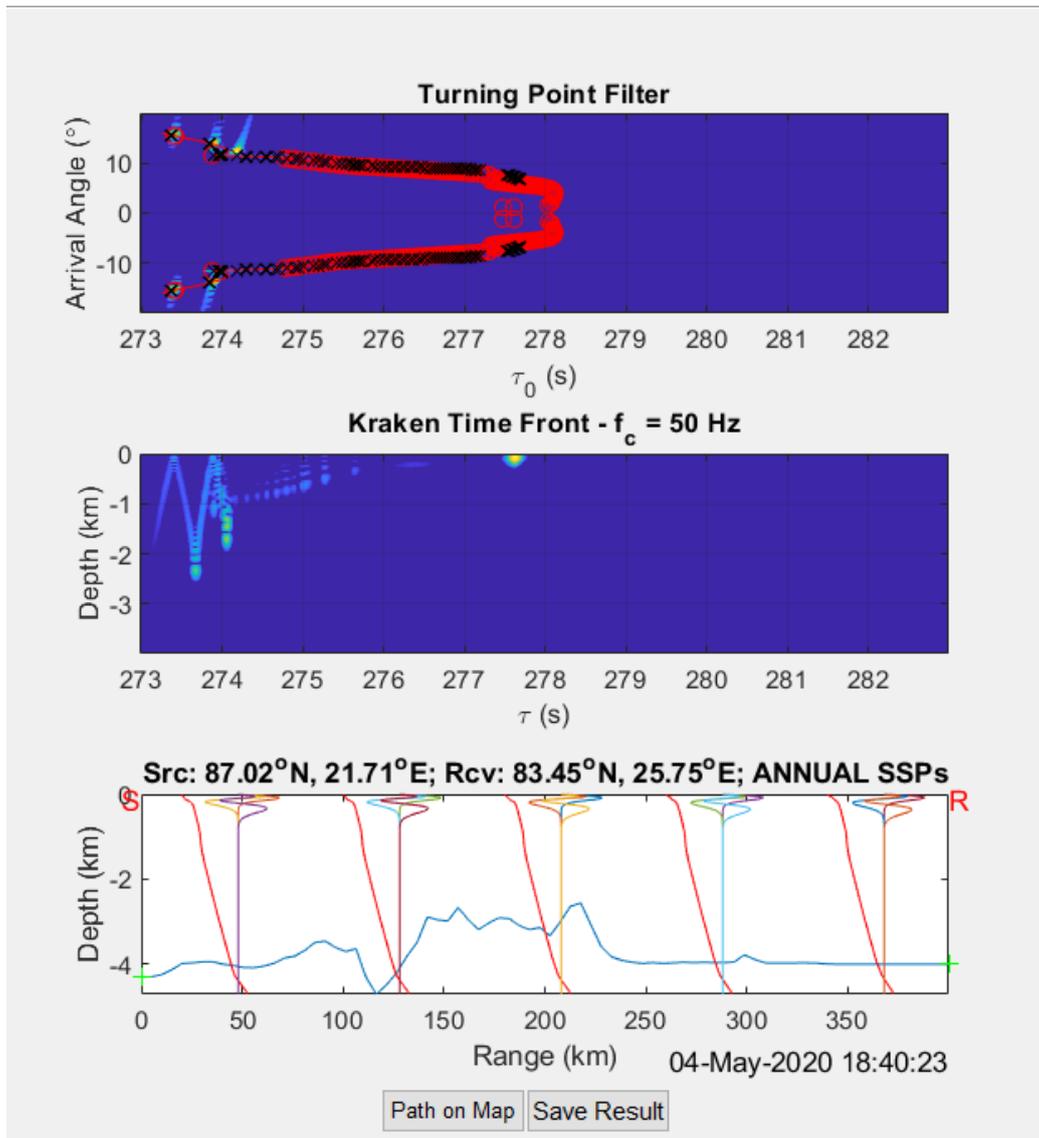


Figure 2.5: Krakens three-panel plot. The upper plot contains a turning point filter, the middle plot displays the time-fronts and the bottom plot contains SSPs.

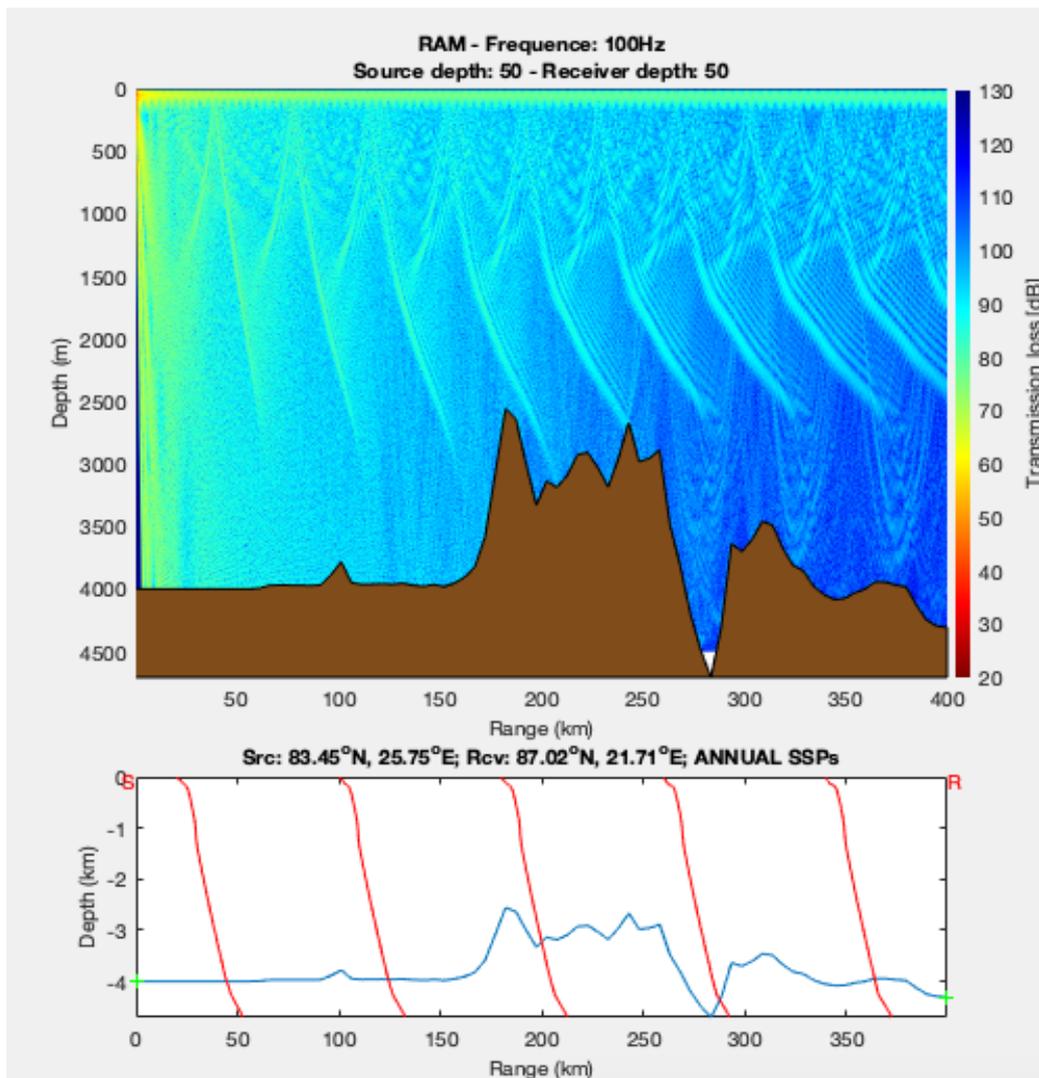


Figure 2.6: Visualization of results from Ram. It's a two panel plot whereas the upper plot shows the transmission loss, and the bottom panel shows the SSPs.

before dissipating. That is why it's less loss in that area [17]. When looking at this figure, it is important to keep in mind that the ratio between depth and range is not representative to how it would look in the real world. The depth is up to 4.5km in this figure, while the range is 400km. Therefore the wave pattern that is visible in the figure is flatter in reality.

Comparing the pattern in this plot with the pattern in the middle plot of the eigenray results, one can tell that there is a lot of similarities in the pattern. The bathymetry in brown is also included in this model.

2.3.4 Bellhop

The Bellhop model is yet another model that was not originally a part of the Arctic Package. Bellhop is a beam tracing model for predicting acoustic pressure fields in ocean environments [22]. The model can produce several outputs. This includes transmission loss, like RAM, as well as eigenrays, such as the Eigenray model does. In addition to these outputs, it can produce ray tracing and arrival and received time-series. Bellhop is implemented in Fortran, Matlab, and Python [22].

Figure 2.7 shows one of the plots you can get when running the Bellhop model. In this example the model has produced ray-tracing plots. Ray tracing is a method for calculating the path of the sound waves through the ocean [23]. The ocean has varying propagation velocity, absorption characteristics, and reflecting surfaces. Therefore, wavefronts may bend, change direction, or reflect off surfaces, complicating analysis. The black and green lines on the figure represent ray paths through the ocean. The green lines has been reflected in the surface, while the black ray paths has been reflected in both the surface and on the bottom some time during the path. If there where any ray paths that was not reflected in the surface nor the bottom, they would appear red. Ray paths that only hits the bottom would appear blue.

As stated above, Bellhop can also produce transmission loss plots. These plots looks a lot like the plot that was produced by the RAM model in Figure 2.6. However, as Bellhop is a ray-tracing model and consequently a high-frequency approximation. Use of either Bellhop or RAM would depend on the application and any requirements on accuracy.

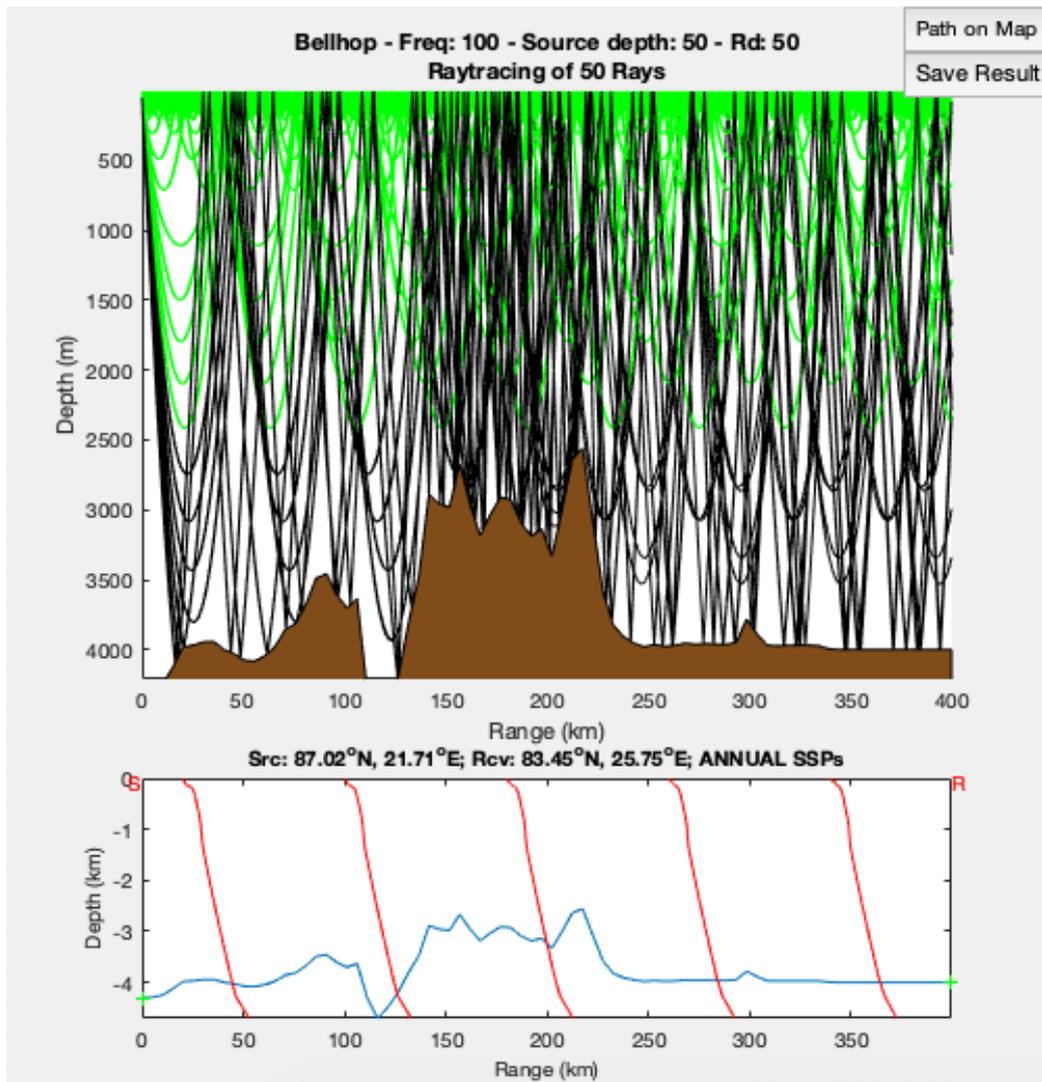


Figure 2.7: This is one of the possible results from running Bellhop. The upper panel shows the ray paths, while the bottom plot shows the SSPs.

Chapter 3

Related Work

There exists related work to both the software engineering and the acoustic aspects of this project. There is one project that is related in the sense of working on a MATLAB program and improving its user interface. Another project is related in regards to an expansion of the functionality of a program. There are also projects that relates to this one because of what they do in the acoustic fields. One of those is an ongoing project that most likely would benefit from using the Arctic Package. This chapter will describe these projects and how they relate to this thesis.

3.1 SOLWEIG

SOLWEIG comes from the acronym Solar and LongWave Environmental Irradiance Geometry. SOLWEIG is a radiation model which simulates spatial variations of 3D radiation fluxes and T_{mrt} in complex urban settings[24]. T_{mrt} is an important meteorological factor that govern human energy balance and the thermal comfort of man outdoors. It is defined as: “The sum of all short and long wave radiation fluxes (both direct and reflected), to which the human body is exposed.” [24]

The SOLWEIG-model was written in MATLAB, but in the master thesis “SOLWEIG A climate design tool” Deepak Jesawni Dewan wrote about his work building a computer software and graphical user-friendly tool that integrates the model[25]. His task was similar to the task in this project, as it is a model for climate estimations that uses maps of the real world and calculates some results. In his project there was only one model, whereas in

this project there were several models. Both models was initially written in MATLAB. Deepak Jesawni Dewan chose to write a separate graphical tool in Java which integrates the model in MATLAB[25]. In his case there were no prior GUI to start from, as the model previously had a command line based interface. The Arctic package already had a GUI written in MATLAB, so continuing using MATLAB felt like a natural choice. In Dewan’s task, the GUI was the main task, whilst in the case of the Arctic package, there where also other tasks that the researchers at NERSC wanted us to prioritize just as much. Therefore changing to another programming language would take up a lot of time.

3.2 MERLIN

MERLIN is a particle tracking software package which has gotten a lot of new functionalities in the recent years [26]. MERLIN is written in C++ and is a accelerator physics library. The “M1.4 Milestone Report: Profiling MERLIN” describes how the MERLIN code base was profiled to identify and quantify the severity of any performance and sustainability issues. Another objective stated in the report is to establish a code reform and refactoring priorities based on the profiling results[26].

Since the code functionality is increased, the size of the code base also increases. This is also the case with the Arctic Package. The added functionalities adds more code and that leads to some challenges like code complexity, readability and maintainability. Just like with the Arctic Package, the intention is to maintain MERLIN for long-term use. Therefore code sustainability is relevant and fast becoming necessary for both projects.

The report focuses on the work done to detect and identify and then remove or improve upon any performance or sustainability issues. Conventional and more advanced profiling analyses have been carried out with intention of quantifying the severity of these issues[26]. The report focuses on how one can make the already extended program more sustainable, while this thesis focuses on the actual expanding of the package. However, it’s important to keep in mind that the work that is described in the article is highly relevant for the Arctic Package as well.

3.3 CAATEX

As mentioned, there are also ongoing projects that are related to this thesis. NERSC are leading or involved in several projects investigating and map marine, cryospheric and atmospheric conditions using, among other methods, polar acoustic measurements. These are both ongoing and completed projects [9].

The Acoustics and Oceanography research group is working on fundamental acoustic research for applications within sea ice research, oceanography, environmental monitoring, underwater communication and geo-positioning. They wish to carry out fundamental oceanographic research for improved estimates of the heat, mass, and freshwater fluxes and storage in the sparsely sampled Arctic region by combining direct measurements with numerical models. Acoustic remote sensing provides one of the few ways to observe the abyssal ocean. The research group wishes to improve the understanding of small and mesoscale ocean processes and their importance for the large-scale ocean climate system by combining new observation technologies, platforms, and models [27].

CAATEX: Coordinated Arctic Acoustic Thermometry Experiment is one of the ongoing projects in this research group. One of the main focuses of the CAATEX project is to provide regional to basin scale information about the temperature in the ocean from acoustic thermometry and standard oceanographic instrumentation [28]. A goal of the project is obtaining new knowledge about the decadal changes in heat content of the central Arctic Ocean and improving our understanding of uncertainties in heat content estimates from climate models [29]. They will use two sound sources in fixed moorings to transmit a signal which will be received on listen cables in five fixed moorings and four drifting receivers hundreds of kilometers away from the sound sources [29]. The fixed mooring thermometry will provide yearlong times series of mean ocean temperature between each of the moorings [29].

This scanning concept will systematically map a very large portion of the upper and lower part of the central Arctic Ocean, manifesting spatial variability. The yearlong time series of mean ocean temperature and point measurements from the fixed moorings will provide information about the temporal variability on local to basin-wide scale.

The CAATEX field experiment will result in baseline data on mean ocean temperature and heat content of the central Arctic Ocean, which will be used for estimation of ocean climate change and evaluation of global climate

models [28]. By comparing the observations from this experiment to similar observations made in 1994 and 1999 its possible to quantify how much the mean ocean temperature along the mooring array has changed over two decades. The new data obtained in the project will be used in combination with high resolution ice-ocean model to obtain an improved estimate of the heat content of the central Arctic ocean. This estimate will be highly valuable for bench-marking how precise the climate models are, and how well they represent the Arctic Ocean heat content[29]. Since the acoustic signals travel along a set of ray paths, there will be multiple arrivals from a single transmitted acoustic signal. The ray paths needs to be calculated using ray tracing, so that each ray path can be identified with a particular travel time [29].

The CAATEX project is an example of a project that would benefit from using the Arctic Package. Both in the design phase of the project, and after the project, when calculating the results. The results from the project could also be used to evaluate the models in the Arctic Package.

Chapter 4

Methodology: Design science

In research, methodology is the approach and means of how to solve problems and to reach new knowledge. As stated in the introduction, we want to answer two research questions in this thesis. These are:

1. How well can we improve on existing tools in the Arctic Package written in MATLAB regarding usability and extendibility?
2. How well can we improve the way of working at NERSC by improving the tools in the Arctic Package?

To answer these questions we have used the Design Science Research(DSR) method, which is a relatively new approach to research [30]. The goal of design science research is to produce prescriptive knowledge for professionals in a discipline and to share empirical insights gained from investigations of the prescriptions applied in context

4.1 The seven guidelines

Hevner *et al.* are listing seven guidelines for Design Science in Information Systems Research [31]. These guidelines also applies to Design Science regarding software engineering and development research. This chapter will go through each of these guidelines and explain what we did to conform to them.

4.1.1 Guideline 1: Design as an Artifact

The first guideline states that the result of DSR has to be a purposeful IT artifact created to address an important organizational problem [31]. Herver *et al.* defines an IT artifact very broadly. It can be either constructs (such as a vocabulary or symbols), models (such as abstractions and representations), methods (algorithms and practices) or instantiations such as implemented and prototype systems [31]. In this case there is no doubt that there is an artifact, namely the Arctic Package in which has been expanding and worked on throughout this research project. This falls under the instantiations part of the definition of IT artifacts.

The article also specifies that the artifact must be described effectively, enabling its implementation and application in an appropriate domain, in which is done in Chapter 5. The chapter describes the design- and implementation process of the the artifact from the state that it was in before, and til the new and improved version.

4.1.2 Guideline 2: Problem Relevance

The second guideline is concerning the subject of problem relevance. Herver *et al.* states in the article that the objective of design-science research is to develop technology-based solutions to important and relevant business problems [31]. For the research to be of any use, it has to answer relevant problems. The research must address the problems faced and the opportunities afforded by the interaction of people, organizations, and information technology to be relevant for the software development community.

As stated in the Chapter 1, there is a need for improved ocean-ice-atmosphere data to understand what is happening in the Arctic area. Improvement of the Arctic Package can lead to more efficient and effective research in this area. Our research of how one can improve the package is therefore of great relevance for the researchers at NERSC, as they are the ones that will be using this tool in the first place. Eventually, the tool might be mature enough to be distributed til other researchers, outside of NERSC, working in the field.

4.1.3 Guideline 3: Design Evaluation

The third guideline reads as follows: "The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods" [31]. In DSR evaluation is a circular component of the research process. The artifact is evaluated based upon the requirements the business environment has established. While we were working on the expansion of the Arctic Package, we had weekly meetings with Espen Storheim. He has been one of our supervisors at NERSC. Espen has been using the old version of the Arctic Package and will be one of the researchers using the new version. His weekly feedback has helped us evaluate and improve the tool as we were working on it. This is better documented in Chapter 5.

In Chapter 6 the results presented are discussed, and Chapter 7 presents interviews with the users where they give feedback on the new version of the Arctic Package.

4.1.4 Guideline 4: Research Contributions

An important part of DSR is the need for providing research contributions to two different fields. Both the field in which the artifact belongs and the knowledge base needs to receive research contributions. A knowledge base is defined by the Cambridge Business English Dictionary as "a collection of information about a particular subject" [32]. The Design Science in Information Systems article reads:

"Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies."[31]

The research presented in this thesis has brought contributions to the field in form of an improved package, in which the researchers can use for more efficient and effective analyzing as well as simulating in advance of experiments, in the design phase. From an acoustic perspective, the new and improved version of the Arctic Package is of great value for the researchers.

During the research we have worked outside of academia, with the industry, while also wanting to provide contributions to the knowledge base. The research project has brought us to knowledge about how one can expand a already existing MATLAB project that has been developed by others, that may not have a lot of experience in software engineering. The project has brought an example of how one can improve and extend a program by using

existing models and combine them into one tool, using MATLAB. It has also brought knowledge about how one can work with many stakeholders that may wish for different things.

4.1.5 Guideline 5: Research Rigor

Rigor can be many things. The article “Rigor The essence of scientific work” discusses the many thing rigor is, both dissatisfaction with uncertainty, being methodical commitment to experimental procedure, strict adherence to the truth and more [33]. The fifth guideline listed in Henver textitet al.’s article reads: “Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.” This means that, in order to follow this guideline, one has to draw from the knowledge base appropriately in terms of foundational theories as well as experimental design.

During the work on this project, the package has been tested continuously. Also weekly meetings with Espen Storheim, one of this projects supervisors, who is also one of the stakeholders of the package, has helped answer to the requirements.

4.1.6 Guideline 6: Design as a Search Process

A design process is usually a Generate/Test Cycle, as seen in Figure 4.1 The 6Th guideline is about the search process for a good design, and it reads: “The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.” [31]

As stated in the previous subsection, the package was tested continuously during development. These small cycles of testing and development fits nicely into Figure 4.1 The weekly meetings with Espen Storheim is an example of bigger cycles where more changes was added to the package, and feedback was given.

4.1.7 Guideline 7: Communication of Research

The 7Th and last of the guidelines presented in the article reads as follows: ”Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences” [31]. This means that

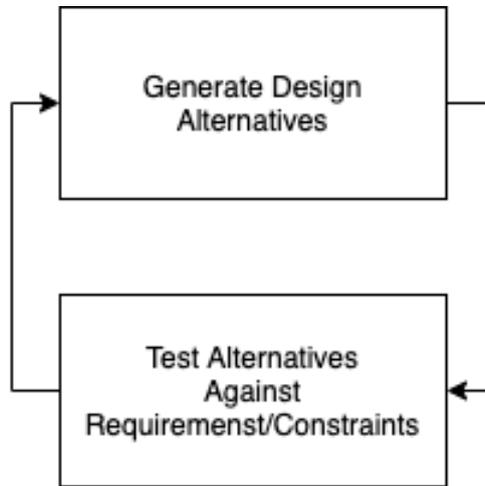


Figure 4.1: The Generate/Test Cycle.

the research needs to be communicated in a effective way to the acoustic research community as well as to the software community. Trough this thesis our research has been communicated to both parts. The background necessary to understand what is happening in the models is described in Chapter 2, and in Chapter 5 the design and implementation of the package is described.

Both in the acoustic research community and the software community collaborations between institutes and organizations across national borders is common. Therefore the choice to write this thesis in English was made in order to reach out to a larger audience in both of the communities.

4.2 The four quadrants

It is possible to position DSR with respect to their knowledge contribution. One can measure their contributions with respect to two dimensions:

- The maturity of the Application Domain
- The maturity of Solutions

The combination of these dimensions results in four quadrants. This is illustrated in Figure 4.2.

The Invention Quadrant describes research where both the Solution Maturity and Application Domain Maturity is low. DSR projects in which little un-

Solution Maturity	Low	improvement: Develop new solutions for known problems	Invention: Invent new solutions for new problems
	High	Routine design: Apply known solutions to known problems	Exaptation: Extend known solutions to new problems
		High	Low
		Application Domain Maturity	

Figure 4.2: The four quadrants in DSR.

derstanding of the problem context and no effective artifacts are available as solutions belongs to this quadrant. The newness of artifacts in this quadrant can make this research hard to publish.

The Exaptation Quadrant describes research where the Solution Maturity is high and Application Domain Maturity is low. DSR projects in which the problem context is not well known, but there exist mature artifacts in other fields that can be exapted as effective solutions belongs to this quadrant.

The Improvement Quadrant describes research where the Solution Maturity is low and Application Domain Maturity is high. DSR projects in which the problem context is mature, but there is a great need for more effective artifacts as solutions belong here. This is also where this project belongs. An improvement is a better artifact solution in the form of more efficient and effective products, processes, services, technologies, or ideas. In this case, both the product (The Arctic Package) and the process of analyzing acoustic measurements using the product is more efficient than before.

The Routine Design Quadrant describes cases where both Solution Maturity and Application Domain Maturity is high. The projects in belonging to this quadrant can be described as professional design or system building, and might not be research. This is not a part of DSR, but evolving or

best practices may be observed and documented in "Extractive case study" - work, and study of best practices in routine design may lead to empirical generalization.

Chapter 5

Design and Implementation

When expanding and improving an already existing program, there is a lot to take in to consideration. There are a lot of different ways to do it, with different pros and cons. Throughout this project I've worked in close collaboration with another master student. Her name is Emilia Botnen Van den Bergh. We have done a lot of the programming work together, and some things by our selves. This chapter will provide a description of the work that is done, who did what and explanations of why this way of doing it was chosen.

5.1 MATLAB

The Arctic Package was written in MATLAB, but the different models that was added to it was written by different people in other programming languages like Fortran and Python. These methods are added to the Arctic program in such a way that the programmer does not need to know how the programs work, just what and how the input should be, in order to generate the correct output.

In order to run the program, one is dependent on MATLAB, which comes with a huge cost. This is a disadvantage considering it makes it harder to distribute the program, and therefore possible less useful.

On the other hand MATLAB is very accurate when providing simulation results and there is a lot of nice looking features. There are various functions that you can use to plot data in MATLAB. Both three dimensional plots, surface and mesh plots.

MATLAB is an interpreter language, therefore it is greatly slower than for example C++, which is a multi-platform language. This might be difficult to quantify because it depends on the source code implementation and the action.

Even though there are a lot of good reasons to rewrite The Arctic Package in another programming language, the program was already written in MATLAB, and to change this would take a long time. This project had to be done within a year, as it is a master thesis, therefore, in order to actually expand the package, the choice to continue working on the MATLAB-version was taken. By doing so, it was possible to focus on the further development of the package, not just the routine work of rewriting the package in another programming language.

5.2 Architecture

To provide added functionality in form of saving the results, a better graphical user interface and more models and so on, there was need for some changes and additions to the architecture of the package. Figure 5.1 displays an block diagram of the original architecture of the package. As one can see in the figure, the bathymetry is extracted from IBACO (The International Bathymetry Chart of the Arctic Ocean)[34] while the SSPs are either taken from the WOA (World Ocean Atlas)[35] or ECCO v4 (Estimating the Circulation and Climate of the Ocean) database[36].

There is one main file that controls the GUI of the project. In this GUI the user can choose which database he or she wishes to extract the SSPs and bathymetry from, as well as other general, and more model specific, input. Before running the model, the user has to set up some fictional sources and receivers. These can be set from previously set databases with longitude, latitude and depth, or it can be done graphically by clicking where you want the nodes to be. It is also possible to set a source or receiver by writing the latitude, longitude and depth manually. There will be more information about the GUI in the next subsection.

When clicking the button “GET PREDICT” in the main GUI, the selection of source and receiver is done in the map. This is done in the plotZRT-file. If there is only one source and one receiver, these are chosen automatically. Still, in that same file, the program calculates the bathymetry on the path from the source to the receiver using the bathymetry map chosen. The sound

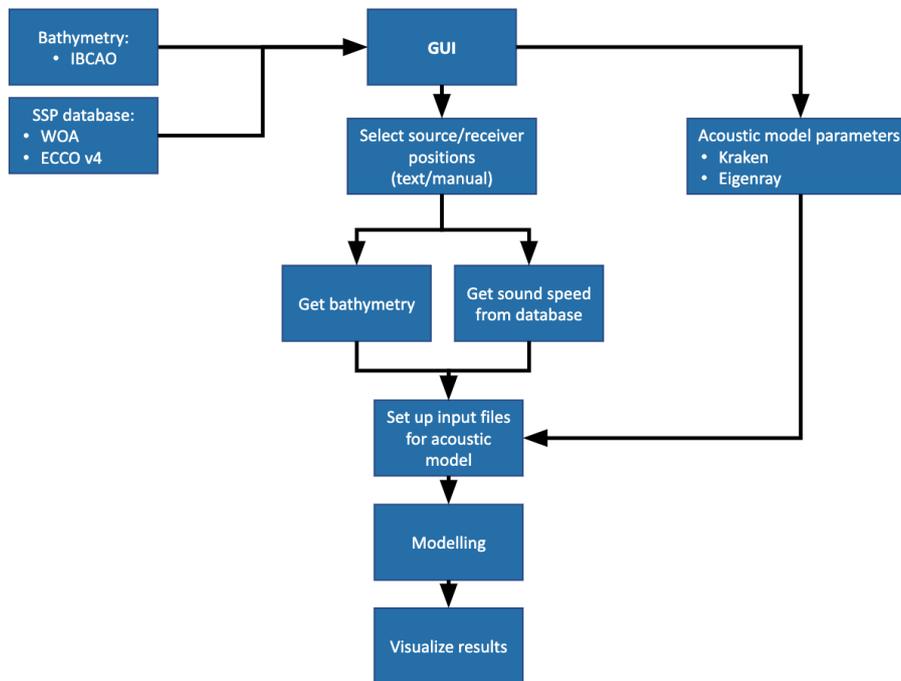


Figure 5.1: Block diagram of the original version of Arctic package.

speed profiles in the path is also extracted from the SSP database that is chosen in the GUI. These are plotted in the bottom third of a result-figure that pops up. Then the plotZRT-file checks which kind of model is chosen, and runs the program for the chosen model. These are the files runRam or runKraken, depending on the model choice. These files edits the input from the GUI and structures it so that it fits the model. It runs the model, and plots the results in the same window as the bathymetry and SSPs. These files works as adapters between the Arctic Package and the models.

The researchers at NERSC wanted more acoustic models, more environmental input (like other bathymetry and SSP inputs), a better GUI, simulating in time and space. In order to be able to have more bathymetry and SSP inputs, we kept the architecture in this part of the program very similar to what it already is. This architecture enables easy expanding in the form of adding new databases. However, bathymetry and SSP databases are not all in the same format, so there can be issues regarding this. To overcome this problem its smart to use the adapter technique. Adapter, Wrapper or Translator is a structural design pattern that convert the interface of a class into another interface that the clients expect. This lets classes work together that could not otherwise because of incompatible interfaces [37]. This makes

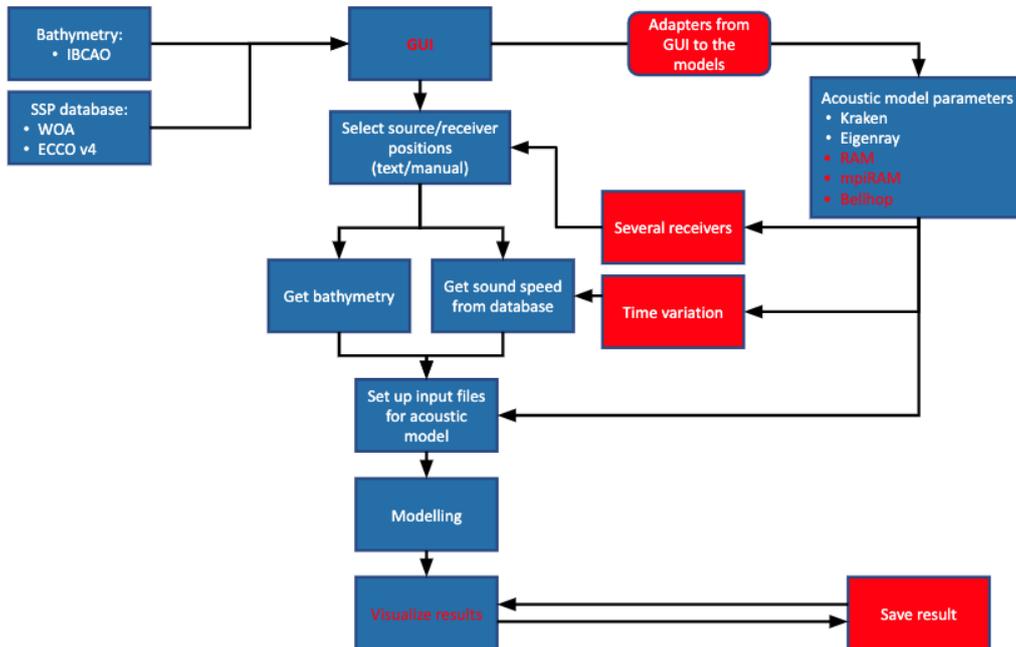


Figure 5.2: Block diagram of the Arctic package after adding more models, saving and other functionalities.

it possible to reuse the Bathymetry data, in stead of restructuring the entire data set to make it fit the package.

When adding new models, such as RAM and Bellhop, we did the same thing, where we created a class that works like an adapter class to connect the two interfaces that did not match completely. In addition to creating the adapter, we also had to make some changes in the GUI. The models need more than the general input like location of source an receiver, therefore, we needed to adjust the GUI to make it possible to adjust model specific input. Adding the RAM model was done in collaboration, while Emilia did most of the work on adding the MPIRAM and Bellhop.

The SSP data that we had access to, had different data for different time of the year, since the sound speed profiles changes with the season. It was possible to choose between the twelve months, the four seasons, or the average SSPs for the year, annual measurements. In the ocean water masses are constantly moving, which in turn affects sound propagation. Therefore, the researchers at NERSC wanted to be able to compare the results of the simulations on the different times of the year. They wanted this to be possible no matter what model is executed. In order to do this, two additional options

were added to the drop-down menu where one could choose which SSP data to use. Before it was possible to choose between the different months, seasons or the annual average. By adding the choices “Seasonally” and “Monthly”, the user can choose to model for several time steps. With the SSP-databases that is available in the system currently, these are the only way to simulate in time, but if a SSP-database that contains weekly or even daily measurements are added, one can simulate in more narrow time steps.

Architecturally, the time steps are simulated by adding a for-loop in the plotZRT-file. This loop is added after the longitude and latitude of both source and receiver is chosen, and after the bathymetry is calculated, as this will be the same for all of the time steps. This is visible in Figure 5.3. The loop loops through all the time steps, gets the SSPs from the database and runs the chosen model on this data. The results are displayed in a figure, the same way as it was in the old version, but there is buttons so that it’s possible to click through the different time steps and compare them. By saving the results, its possible to get other visualizations of the time-step results where it’s easier to compare. The work done to make simulating in time steps possible was done by Emilia.

The sound does not only propagate in a straight line from A to B. The researchers at NERSC also wanted to be able to analyze how the sound spreads not only on a single path, but in a general area from a source. To be able to analyze this, with the models that is made for simple paths from A to B, and not in three dimensions, functionality for plotting from one source to several receivers in a circle surrounding it was added. After the source location is selected on the map, if the choice of running the model on several receivers is made, a window asking for information about the receivers will pop up. Here, input about how many receivers is wanted and their distance from the source is added. When this is done, the receivers will be visible on the map. To run the models from the source to all the receivers, it’s also necessary to calculate the bathymetry on all of the different paths, therefore a for-loop is created in the plotZRT-file. The loop starts after the receiver input is chosen, and it does not end until the modeling and visualization of the model is done. For each receiver the bathymetry is calculated, and then, still in the loop, the code continues to the time-step-loop. This is visible in Figure 5.3, where an overview of the plotZRT-file is presented.

By having this double for-loop, it also enables the user to simulate both in space and time at the same time, by choosing to use several receivers and several time steps. The results of such simulations will be similar plots, where one can page through the time steps and receivers as one wish. By saving the

results, they can be visualized for better comparison later. The work done to make simulating with several receivers was done by me.

When it comes to saving the results this is done in a temporary two-dimensional array. A regular execution of a model with only one time-step and one receiver results in a one-by-one array containing the result-structure. However, if one chose to run with X receivers, where X is a number greater than 1, the results would be saved as a X -by-one array. Similarly, if the simulation was executed using Y time steps, the result would be saved as a one-by- Y array. And therefore, a simulation with X receivers and Y time steps would result in a X -by- Y array. This array is temporary saved as the program is running, but if the program is closed without saving, or if another model is being executed, one is at the risk of losing the data. The choice of saving these result-arrays as files on the computer was made with this in mind. The naming of the saved files contains the date and time of execution, as well as the name of the model executed, and whether or not there are several receivers or time steps. Emilia and I worked together on this part of the project.

5.3 The User Interface

One of the main wishes the researchers at NERSC had, was a better GUI. They expressed that they thought the old GUI was confusing and messy and that they wanted to lower the threshold for the user. With the GUI seen in Figure 5.4 everything is pressed into one main menu, and the font is so small that it's hard and maybe even impossible to read. When improving the GUI of the Arctic Package, we had Jakob Nielsen's list of 10 usability heuristics for user interface design in mind. These are called heuristics because they are general rules of thumb rather than strict and specific guidelines. The list of heuristics goes as follows [38].

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall

7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

The old system had one main menu that kept track of all of the inputs. Keeping this makes the first heuristic on the list pretty implicitly fulfilled, as it is easy to keep track of which inputs one has changed and need to change when seeing them all at the same time. However, the choice of moving the model specific input to another window was made. Because this window pops up whenever a model is chosen, the system status is visible here in the form of that the window pops up in front of the main GUI when it's time for setting these inputs and running the model. When the models are about to run, the console will tell the user to click on a source and a receiver. This makes it easy for the user to understand the status of the system.

The next heuristic on the list is all about making the system speak the users language and making the information appear in a natural and logical order [38]. To accomplish a natural and logical order of the tasks, grouping of the inputs were of importance. The first group contains inputs needed for creating the map, and the button for getting the map up on the figure. The next group contains inputs for setting the sources, and then a group for setting the receivers. After this run-settings can be set, where one can choose the model, get the separate window for model specific parameters and execute the model. All of these groups are on the same figure, and can be seen at the same time. This means that the order in which these tasks can be done is flexible, but the groups are placed in a logical and chronological order.

Heuristic number three "User control and freedom", is about supporting undoes and redoes, in case the user makes mistakes. When adding sources from a file, there is a button right next to the "Get-button" that says "Clear". This is an example of how one can undo the act of getting the sources on the map, if the choice of sources was made by a mistake.

Regrading consistency and standards, heuristic number four, the users should not wonder whether different words, situations and actions lead to the same thing or not [38]. Keeping the language simple and precise, and following platform conventions is of importance. As an example, the phrasing "Get predict" is used on all the buttons that activates the map where you choose source and receiver.

Design that prevents errors from occurring is good design. To prevent user mistakes, and following the 5Th heuristic, all of the inputs have default input. If the user is not certain what the input is supposed to be, or if it doesn't matter for a particular simulation, then he or she can run the program with the default settings. The program also disables the timestep(s)-drop down menu when the SSP chosen only contains one time step.

The 6Th heuristic "Recognition rather than recall" is about minimizing the user's memory load by making actions and options visible [38]. This is to prevent the user from having to remember information from one part of the dialogue to another. In the case of the work done to the arctic package regarding this heuristic, all of the windows can be visible at the same time. If the user needs to remember some of the input that he or she put into the model after the execution, this is still visible in the main GUI, or in the GUI window for the model specific input.

Moving on to the 7Th heuristic, regarding flexibility and efficiency of use, we chose to keep the GUI in such a way that most of the inputs was controlled by the main menu. This was done to keep the flexibility regarding order of the inputs. The programs nature itself is very flexible because it can do a lot of different things.

Heuristic number eight is about having a aesthetic and minimalist design. To achieve this we needed to prioritize the different buttons and input-sections that was visible in the old GUI. One of the first thing that was done with this in mind, was moving the model specific parameters to separate windows. There is no need for input sections for eigenray specific input when running eigenray. When doing this, it became clear that some of the input fields that looked like it only belonged to eigenray executions where general input that all of the models needed. This emphasizes how much the need for a new and improved user interface was. By talking to the researchers at NERSC, we also came to the conclusion that some of the less used features could be moved to a separate window. Therefore moving the inputs and buttons for setting the sources and receivers manually with longitude and latitude, as well as moving, setting or deleting singular sources or receivers were moved to this new window. To make the GUI more aesthetically pleasing, the placing of the input fields and buttons were set in more or less straight lines and columns, and less of the off set and messy nature of the old GUI.

The 9Th heuristic "Recognize, diagnose and recover from errors" states that error messages should be displayed in plain language. The user should be able to fix his own mistakes, and understand what went wrong without needing to know programming or how to code. An example of a error message the

user can meet is if he or she tries to click the “GET PREDICT”-button before sources is set. When this happens the error message states: “Must Define Source Location Before Plotting Bathymetry and SSPs”. This is a clear message that explains to the user what went wrong, and what he or she should do to fix the issue.

The 10th and last heuristic “Help and documentation” sets focus on that it may be necessary to provide help and documentation, even though it would be better if the system could be used without documentation [38]. This information should be easy to search and focused on the user’s task. By the time for writing this thesis, the documentation of the program is not finished.

Emilia and I collaborated on making the user interface. I developed a first draft, and set the model specific inputs, as well as the advanced source and receiver settings in separate dialogue boxes. During the development of the other features in the Arctic Package, we had to make some changes to the first draft. We discussed different approaches and developed the final result in collaboration. Because we chose to continue the programming in MATLAB, there was some limitations to design choices and the overall look of the GUI. With MATLAB code it’s not as easy to customize the UI-elements as it is in for example JavaScript with HTML and CSS. It is possible to do, but with limited experience using MATLAB, and because the results we got when customizing the UI-elements varied with the operating system, the choice of keeping it as close to the standard looks were made. This is also done to save time and focus on making it look good on all operating systems.

5.4 Visualizations

The researchers at NERSC want to be able to look at the previously saved results. The result visualizations that appears after executing a model does not necessary display all the information that the model produces. There could be other ways to visualize the results, that may be of great advantage for the researchers. All of the models that are saved has information on how the SSP’s looks like in the area, as well as the bathymetry.

One of the visualizations added to the Arctic Package is a display of the anomalies in the area of which the model is executed.

When looking at a previously saved result of an Eigenray execution, one can create a figure with the SSP displayed in as a color chart, with the bathymetry

and eigenrays plotted on top.

When running a RAM model with several receivers, one obtains information in a three dimensional area, but the standard result-output only displays a number of two dimensional plots one can page through. To take advantage of the three dimensional nature of this data, we added a visualization that saves all the data from the two dimensional propagation loss matrices in each of the result-structures in the result file, and concatenates this into one big three dimensional matrix. By looking at this matrix from another angle, one can observe how the sound propagates in several directions.

By saving an execution of the eigenray program with several time steps, its possible to get other visualizations than the standard output of this too. It is possible to compare the travel time, arrival angle and initial angle of the eigenrays at the different time steps.

The visualizations that compares the different time steps is added by Emilia, while I added visualizations for anomalies, eigenray on top of the SSP-chart and for several receivers. Emilia also added some visualizations for Bellhop and functionality for simulating mpi-RAM and Eigenray with time fronts at the same time and comparing them. I added visualizations for transmission losses for a given depth, and the average transmission loss on a path.

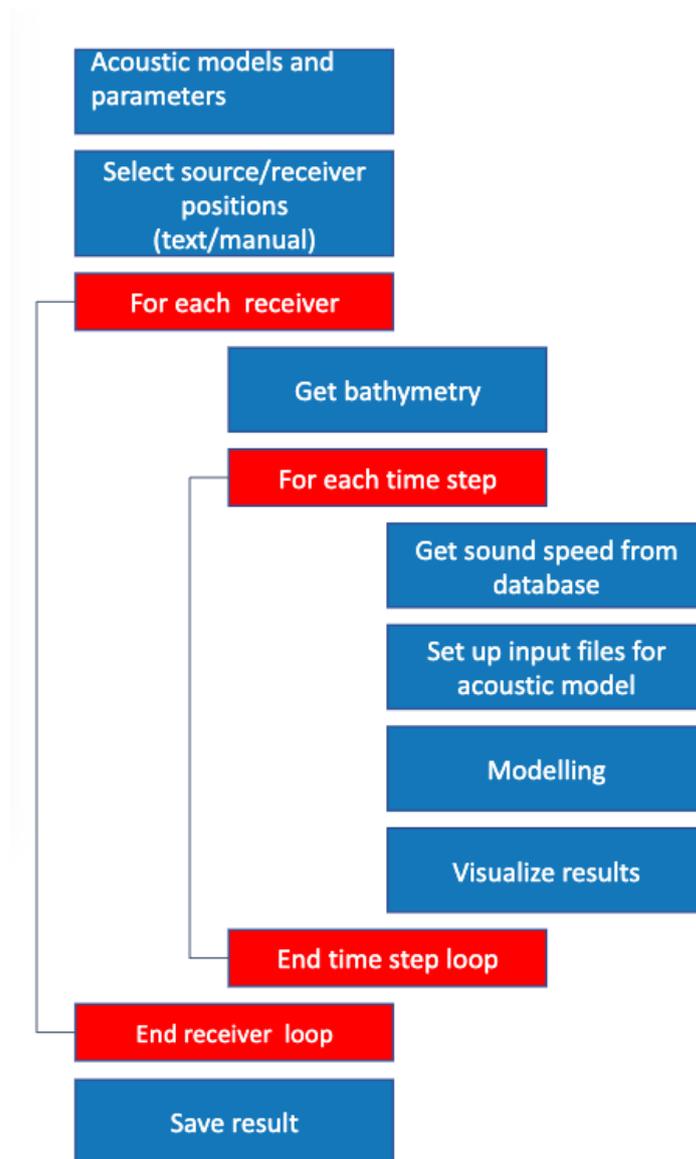


Figure 5.3: An overview of how the plotZRT-file handles several receivers and time steps.

Azimuthal Equal Area Projection Values

Cen Lon: -5 Cen Lat: 85 Radius: 15

Min Lon: -180 Max Lon: 180 Min Lat: 70 Max Lat: 90 ENTER

Array # =

SELECT FILENAME FOR SOURCES: sources_CAAATEX, 25May18 08:08, 8 Sources, Lon: -150:120, Lat: 73:90

SELECT FILENAME FOR RECEIVERS: receivers_CAAATEX, 25May18 08:08, 8 Receivers, Lon: -150:120, Lat: 73:90

New Src Depth (m) = 200 GRAPHICALLY GET SRC New Rcvr Depth (m) = 200 GRAPHICALLY GET RCVR

ENTER SRC (LON, LAT, DEPTH) .. RMV SOURCE RMV SRCS OUTSIDE BOX

ENTER RCVR (LON, LAT, DEPTH) .. RMV RECEIVER RMV RCVR OUTSIDE BOX

NUDGE SOURCE NUDGE RECEIVER CLEAR SOURCES CLEAR RECEIVERS

PLOT PATHS PLOT Z(R) AND: SSP: GET PREDICT NONE

Eigenray Parameters

DR(km):Z 5 DR(km):C 100 Save P... E...

First and Last Angle -15, 15 No. of Rays 2500

Use bottom... Max # Bot Ref 3 Epsilon 1e-5

T_min,T_max -1,-1 Z_min,Z_max -1,-1 Set Axis

Mode Parameters

Freq (Hz) 50 T-delay (s) -1

Ocean Grid 2000 Bandwidth-FW (Hz) 20

Sediment Grid 300 T-width (s) 10

No. Modes 50

Figure 5.4: Screenshot of the old GUI.

Chapter 6

Results and Interpretation

After expanding the Arctic Package and changing the GUI as well as interviewing users about the new version, there are a lot of new results to be presented. The results and interpretations of the results from the expansion of the Arctic package will be described in this chapter.

6.1 Saving simulation results

Before the work on the Arctic Package, there was no simple way to save the results of a simulation. If one wanted to see the results of a simulation one had to recreate the exact same simulation. To fix this functionality to save results is added, There is added functionality to plot previously saved results, bathymetry, sound speed and anomalies. The goal was to save as little data as possible, but still being able to run the plot and reproduce the same results. The results needed to recreate the simulation is saved in a X-by-Y matrix with result-structures with date and time as a part of the filename. Where X is the number of receivers and Y is the number of time steps. Each slot in the matrix has information about the modeling done from the source to the receiver at that given time step. For simulations done with one receiver at one time step, the result-structure is saved in a one-by-one matrix.

For the user, saving the result is easy. They just click the "save result"-button located on the plotting. This is visible on Figure 6.1.

When clicking the result button, the MATLAB-file `saveResult.m` is executed. This file puts all the data that is in common for every model execution,

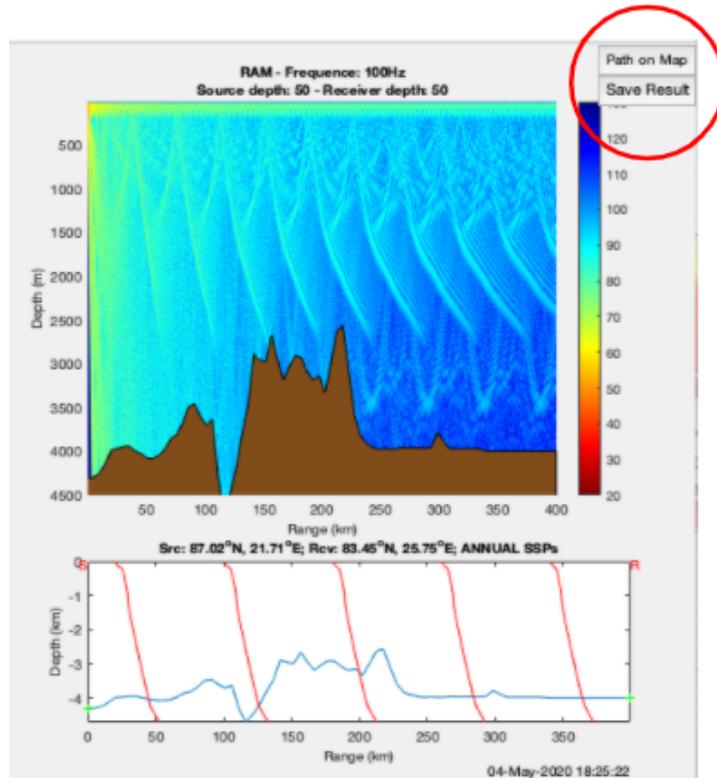


Figure 6.1: The button for saving the results. Here shown for a simulation of transmission loss with RAM.

such as source and receiver location, as well as the bathymetry and sound speed profile of that area in the result-structure. Then it checks which type of model simulation is executed, and saves the parameters and the results that are unique for that model in the same structure. When this is done, it saves this structure to a file with a name containing the type of model simulation that was executed, the date and time, whether the simulation was executed on a specific time period (Annual, a specific season or a month) or if it was executed on several time steps, and if the simulation was executed with several receivers.

6.2 User Interface

One of the things that was high up on NERSC wish list for the Arctic Package was a simplified graphical user interface (GUI). The old GUI (seen in Figure 5.4 and Figure 6.2), was both overwhelming, messy and confusing. It was

quite cumbersome to navigate through the interface.

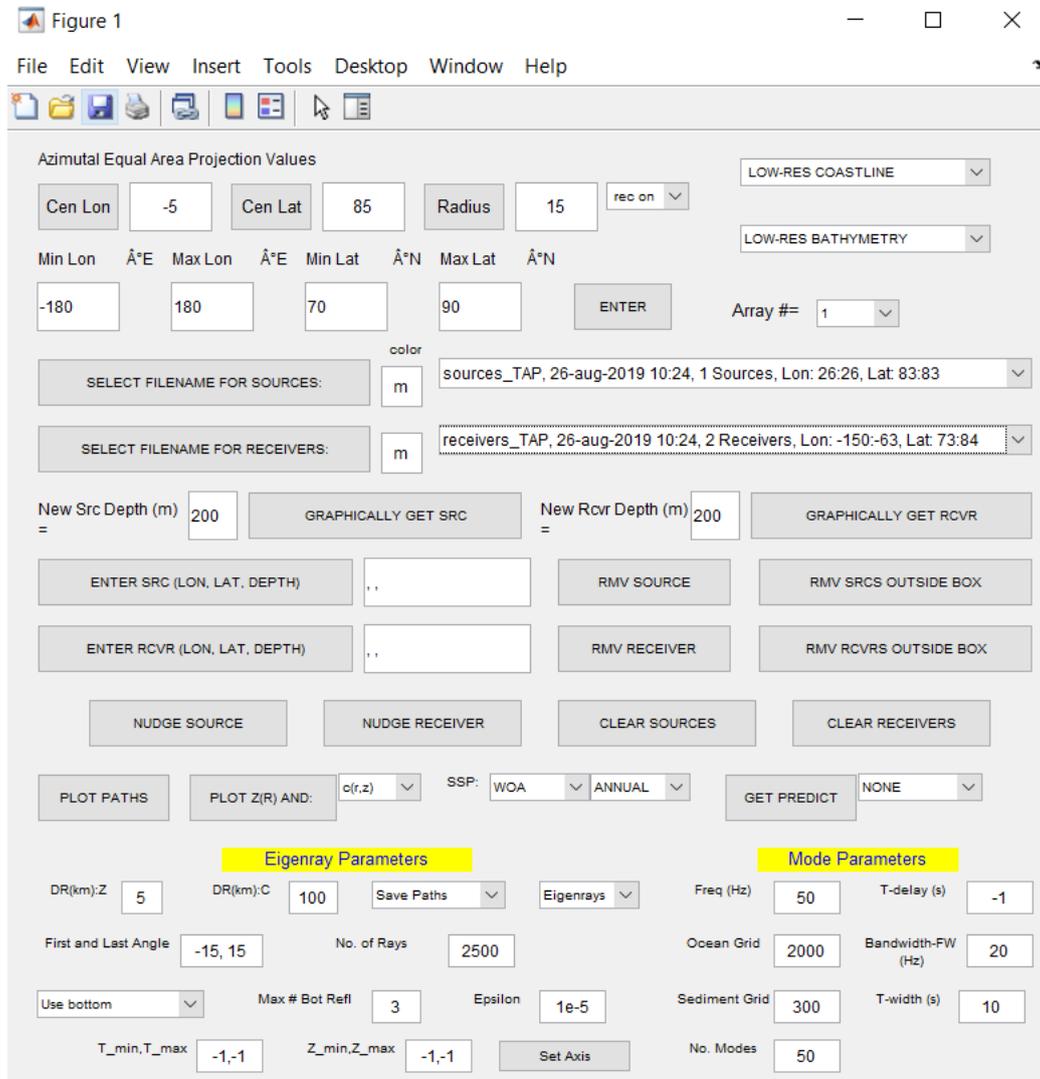


Figure 6.2: This is the old GUI on Windows.

The GUI looked different on different operation systems. On mac OS, the font was almost unreadable because it was so small. Figure 6.2 shows how the old GUI looked like on a Windows machine. Comparing Figure 5.4 and Figure 6.2, the difference is quite noticeable considering its the same program. This is because MATLAB uses the native definition of what a uicontrol should look like. A uicontrol is e.g. a push-button radio button or a drop down menu. MATLAB GUIs on a Mac machine will use mac OS uicontrols, while the same GUI on a Windows machine will use windows uicontrols. These

definitions may differ in appearance. It is also worth noticing that both on the mac OS and Windows there were some buttons that were just functioning as text. These were the three first buttons from the top left. “Cen Lon”, “Cen Lat” and “Radius” is just text on a button, and the button did not have any functionality.

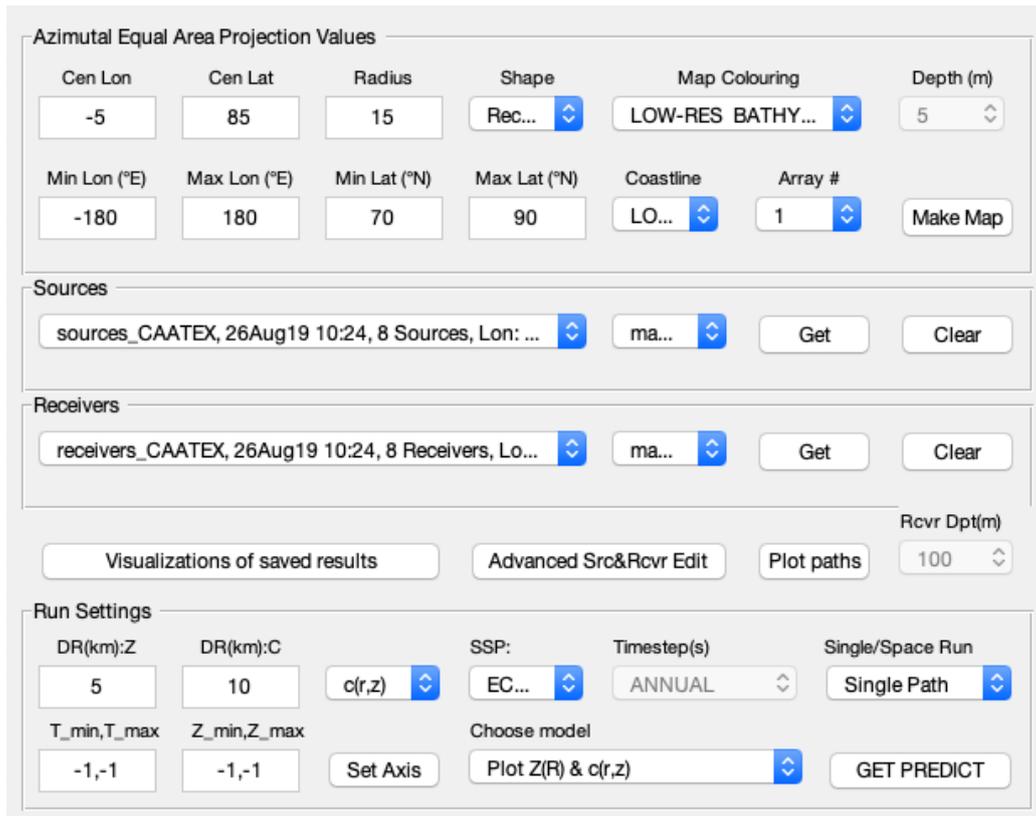


Figure 6.3: The new GUI on mac OS.

The new GUI looks more similar on both mac OS and Windows. Changing the non-functioning buttons to plain text and moving the text over the input fields made a more cohesive appearance. Grouping the inputs together based on what they did made the GUI easier to understand.

By clicking the “Advanced Src&Rcvr Edit”-button a separate window with advanced settings pops up. This window contains functionality for creating, moving or deleting sources or receivers graphically on the map and functions for setting new sources and receivers on the map by coordinates; longitude and latitude and the depth of the new node. This is shown in Figure 6.5

Azimuthal Equal Area Projection Values

Gen Lon	Gen Lat	Radius	Shape	Map Colouring	Depth (m)
-5	85	15	Rec... ▾	LOW-RES BATHY... ▾	5 ▾
Min Lon (°E)	Max Lon (°E)	Min Lat (°N)	Max Lat (°N)	Coastline	Array #
-180	180	70	90	LO... ▾	1 ▾

Make Map

Sources

sources_CAATEX, 26Aug19 10:24, 8 Sources, Lon: ... ▾ ma... ▾ Get Clear

Receivers

receivers_CAATEX, 26Aug19 10:24, 8 Receivers, Lo... ▾ ma... ▾ Get Clear

Rcvr Dpt(m)

Visualizations of saved results Advanced Src&Rcvr Edit Plot paths 100 ▾

Run Settings

DR(km):Z	DR(km):C	SSP:	Timestep(s)	Single/Space Run
5	10	c(r,z) ▾	EC... ▾	ANNUAL ▾
T_min,T_max	Z_min,Z_max	Choose model		
-1,-1	-1,-1	Plot Z(R) & c(r,z) ▾	GET PREDICT	

Figure 6.4: The new GUI on Windows.

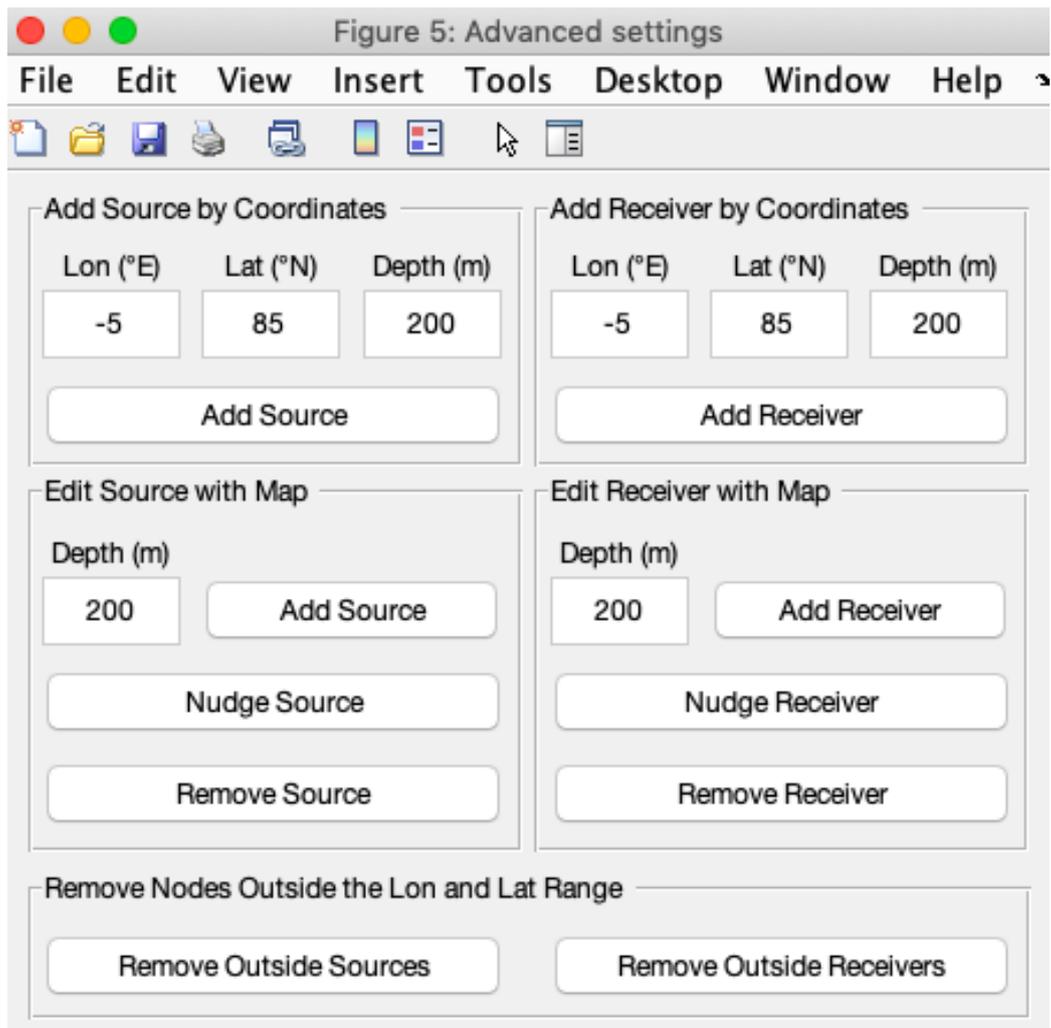


Figure 6.5: Window for advanced settings for sources and receivers.

As seen on the Figures 6.3 and 6.4, we've removed the model specific input fields to separate windows. Two examples of this can be seen in Figure 6.6. These never appear at the same time. When a new model is chosen in the "Choose model"-drop down menu, this window changes to the corresponding model. Some of the input fields that previously was placed under the "igenray Parameters" heading in the old GUI, was not in fact specific to the Eigenray model, but had to be set for running the program no matter which model is chosen. These were kept in the new version of the GUI.

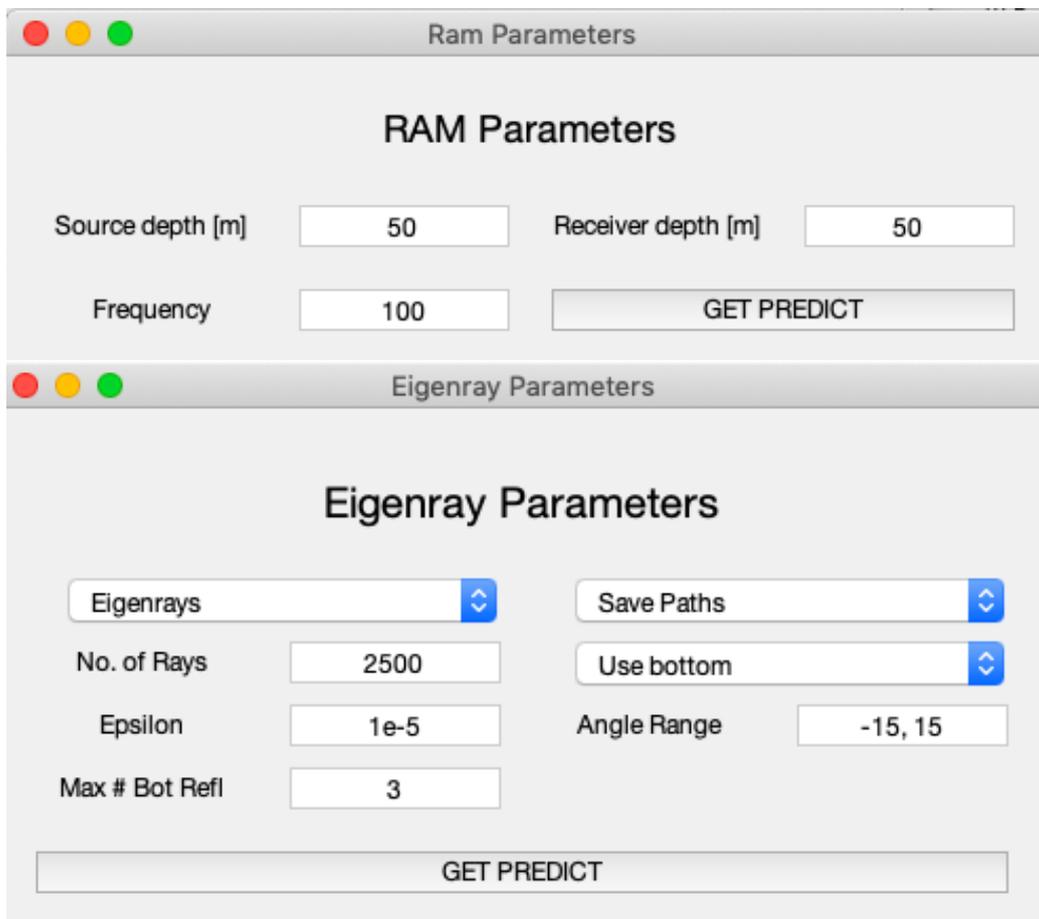


Figure 6.6: Separate windows for model specific input for Eigenray and RAM.

6.3 The new models

Several new acoustic models are added to the program; RAM, MPIRAM and Bellhop. All of these models were written by someone else already, but we

had to add them to the program and write some adapters as well as making a menu for the model specific input for these.

The MPIRAM model has three model specific parameters that can be set by the user in the MPIRAM Parameters window. The results after running the MPIRAM model is shown in Figure 6.7.

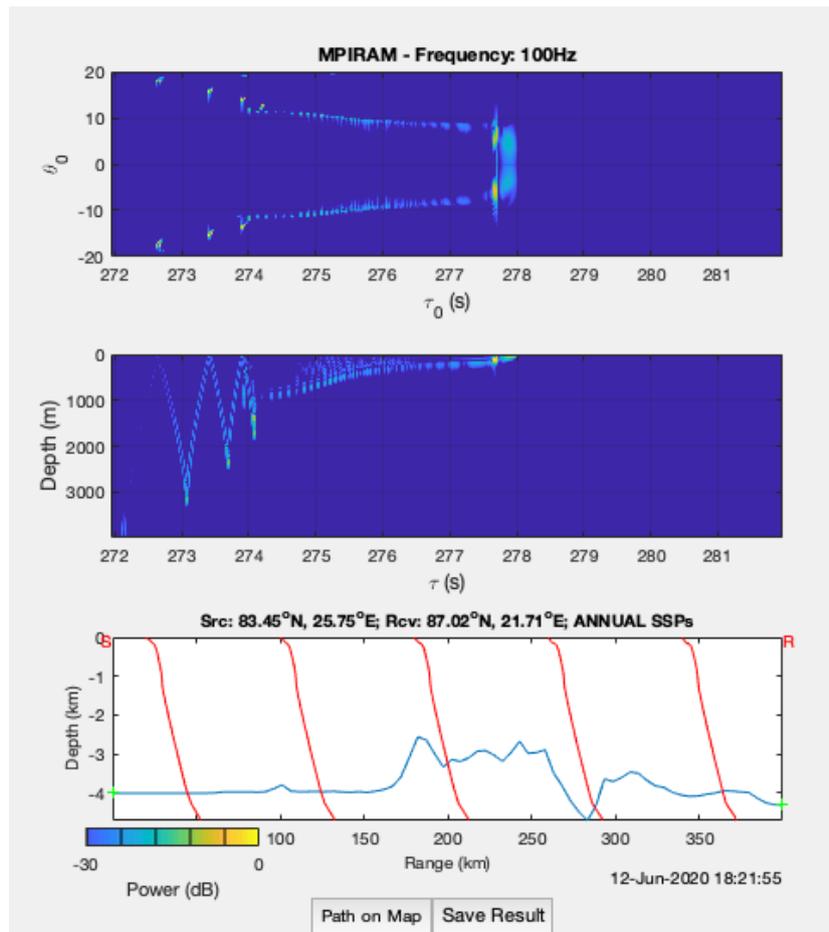


Figure 6.7: Example of a result when running MPIRAM

RAM is shown in Figure 6.8. As with all of the other models, it can be executed from the main GUI. The model itself is better described in Chapter 2.

The Bellhop model is also new, and added to the Arctic Package. This model is quite diverse. It can produce both ray tracing, eigenray, coherent transmission loss, semi-coherent transmission loss and incoherent transmission loss. It can also produce combinations of ray tracing and eigenray with the different

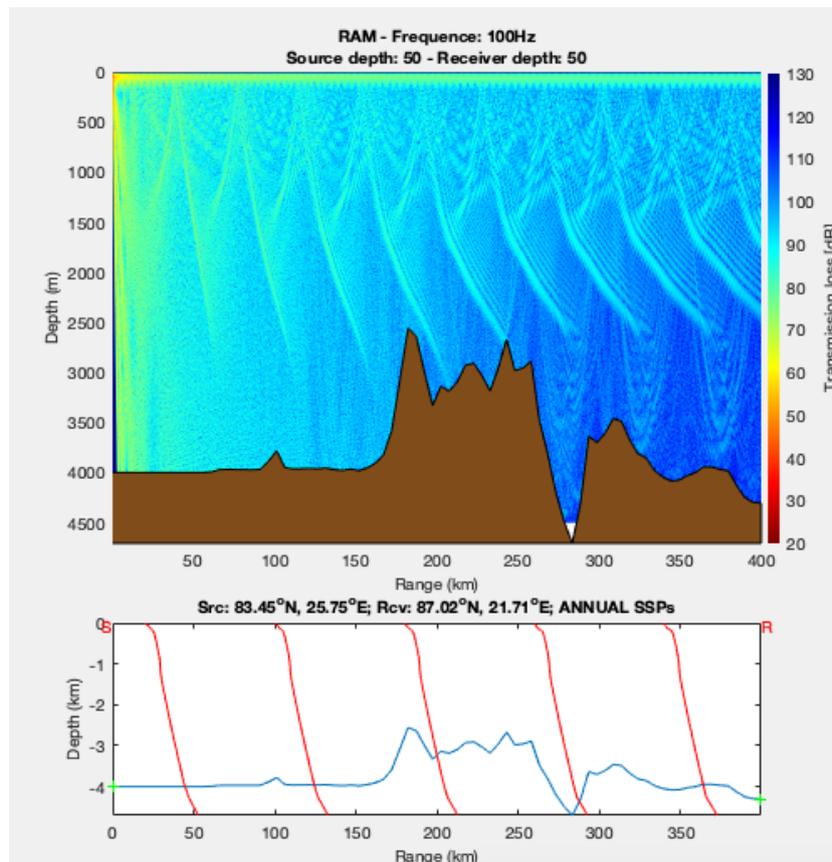


Figure 6.8: Example of a result when running RAM

transmission losses.

The Arctic Package could only run two models before. The new models contribute to the package and means that the researchers now can run many different models with the same program, and they don't need to do any programming themselves in order to get these models up and going. This means that they for example can run different models on the same path and compare the results. To assemble many models into one program also makes it easy for students to study how the different models work, without having to download a number of different tools in order to run the models.

6.4 Time Variation

The constant movement of the water masses in the sea affects how the sound propagates over time. Because of this, functionality for handling time variation is added, so that one can do simulations from a source to a receiver, and see how the results would vary over time. The SSPs that is available in the Arctic Package had data for each month and the four seasons. By running the models several times on the different SSPs for the months or the seasons, the program produces different results due to the difference in the SSPs. Instead of choosing one of the months, seasons or the annual average, it is now possible to choose to do simulations for monthly or seasonally SSPs. The drop down menu in which the user can choose to do this is visible in Figure 6.9.

After the model is executed for all of the time steps, the result window will contain a similar result as it does when a regular mode is executed, but there will be buttons on each side of the figures so that the user can step between the time steps. By saving the results and clicking the "Visualization of saved results"-button, it is possible for the user to get some nice plots that compare the different time steps to each other. Figure 6.10 displays a visualization of the arrival time of eigenrays executed on the different months in a year.

The time variation addition to the package makes it possible to study how the sound propagation in the ocean changes over time. This part of the project has been Emilia's focus.

The time variation is of interest for the researchers because the water in the ocean is constantly moving, in which changes the sound propagation. Running the same model over the same path over time makes it possible to compare and get a better understanding of the pattern in this behavior.

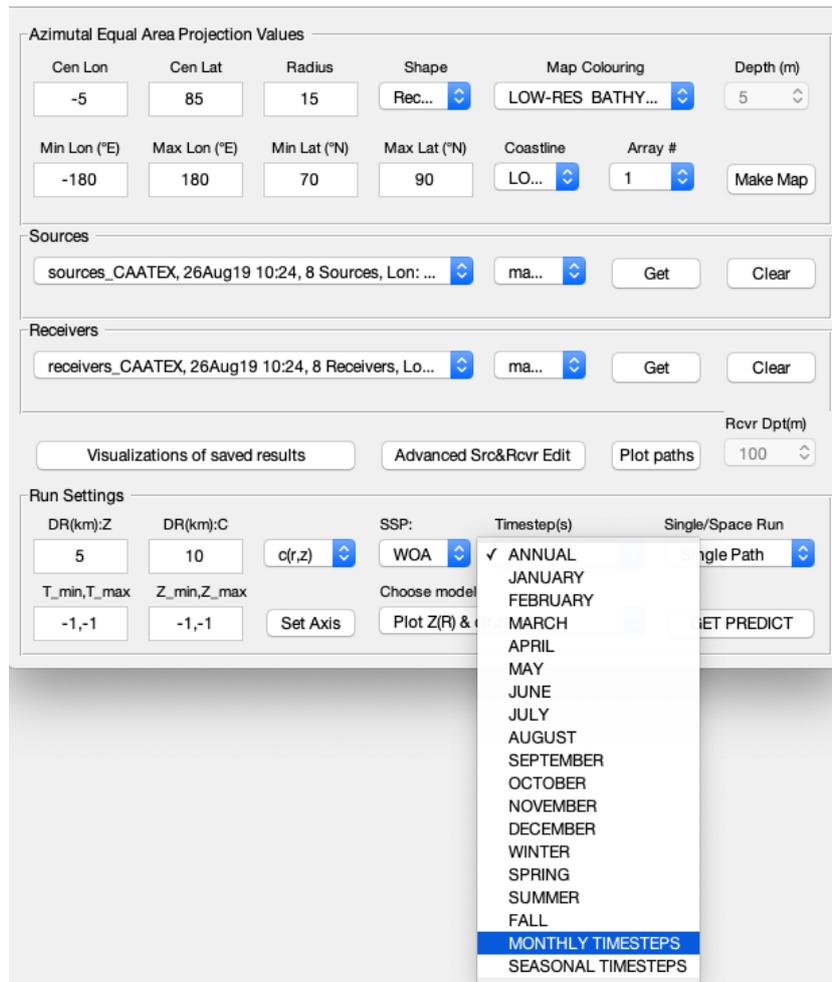


Figure 6.9: The drop down where the user can choose to run the model over several time steps

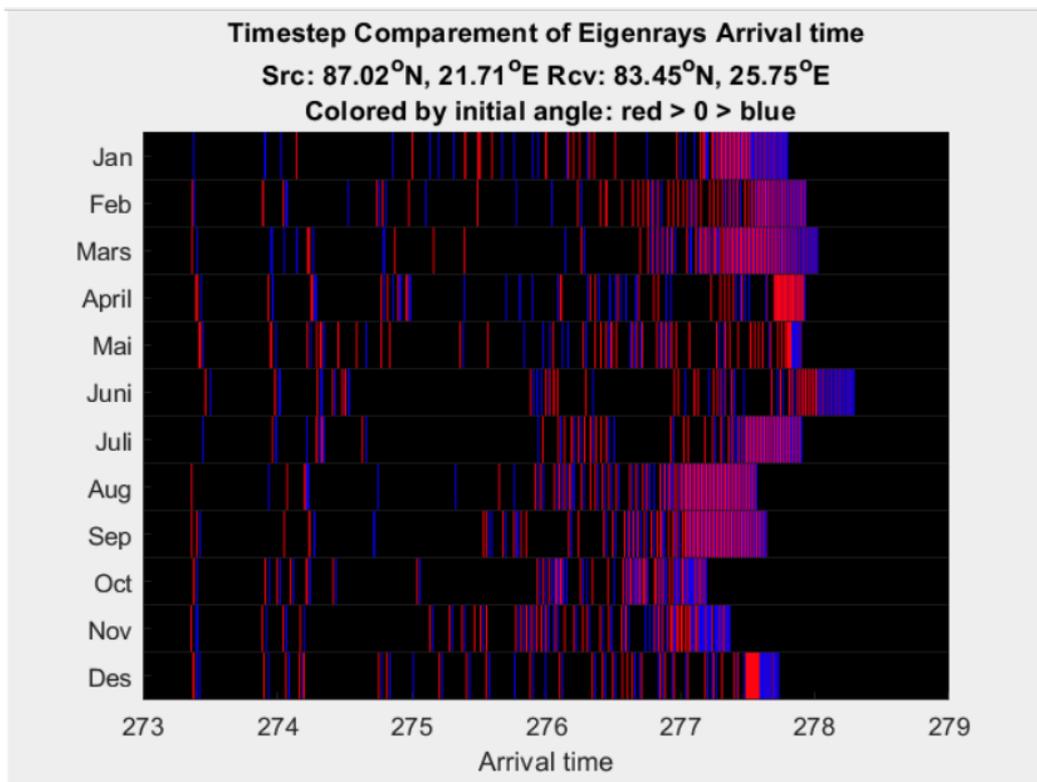


Figure 6.10: A visualization of Eigenrays Arrival time for the different months.

6.5 Simulating in space

There is also added functionality to handle receiver variation, so that one can do simulations in a three-dimensional space. In the new GUI seen on Figure 6.3 and 6.4, there is a drop-down menu with the heading “Single/Space Run”, where the user can choose between running the program on a singular path from a source to a receiver, or if the user wants to run the program from one source to many receivers. If this drop-down menu is set to “Space Simulation” when the “GET PREDICT”-button is pressed, the user only has to chose the source from the map. When the source is chosen, a separate window for setting the receivers pops up. This window can be seen in Figure 6.11.

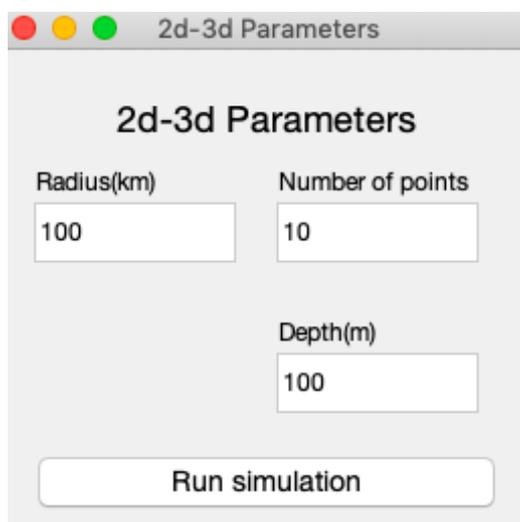


Figure 6.11: The window that controls information about receivers when running a model on several receivers

In this window the user decides how far away from the source the receivers should be, the depth of them and how many receivers it should be. More receivers will result in a more detailed result, but it will also affect the run time. Simulating any models with several receivers will result in the same standard results as shown in the Figures 2.3, 2.7 and 2.6, but there will be buttons on each side of the plots, for clicking through each of the paths. On Figure 6.12 one can see how the receivers surround the source on the map if the radius is set to 75km the number of points is set to 100.

When this is executed and saved with RAM, its possible to visualize the results from another angle. The rosette plot, is a plot where the user can see

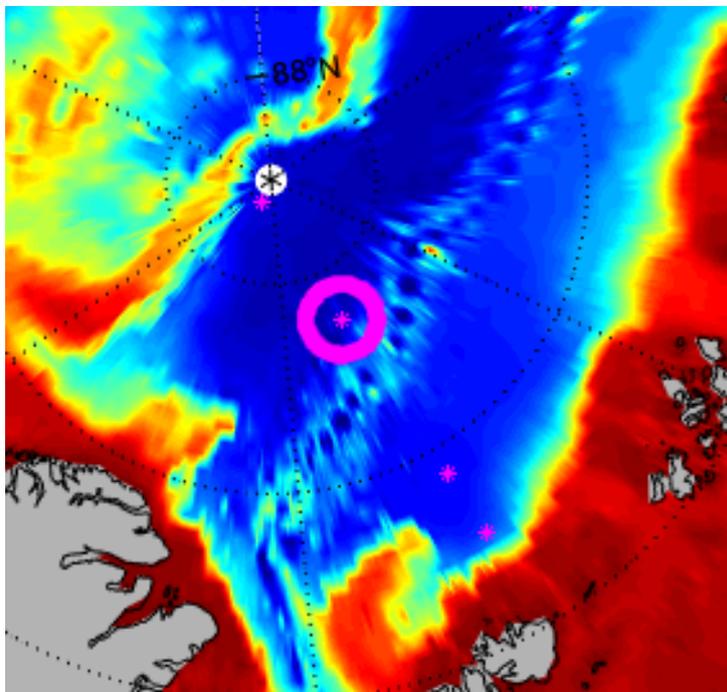


Figure 6.12: The map with 100 receivers surrounding a source with a radius of 75 km.

the propagation loss from the source to all the receivers surrounding it in a circle at a given depth. The user can choose any depth he or she wants, and look at how the sound propagates in several directions at that given depth, and analyze where the sound is louder. Figure 6.13 displays the transmission loss from a source to 100 receivers surrounding it with a distance of 75 km at a 2700 meters depth. If this simulation was done with different time steps as well, the user would be able to step between the different time steps.

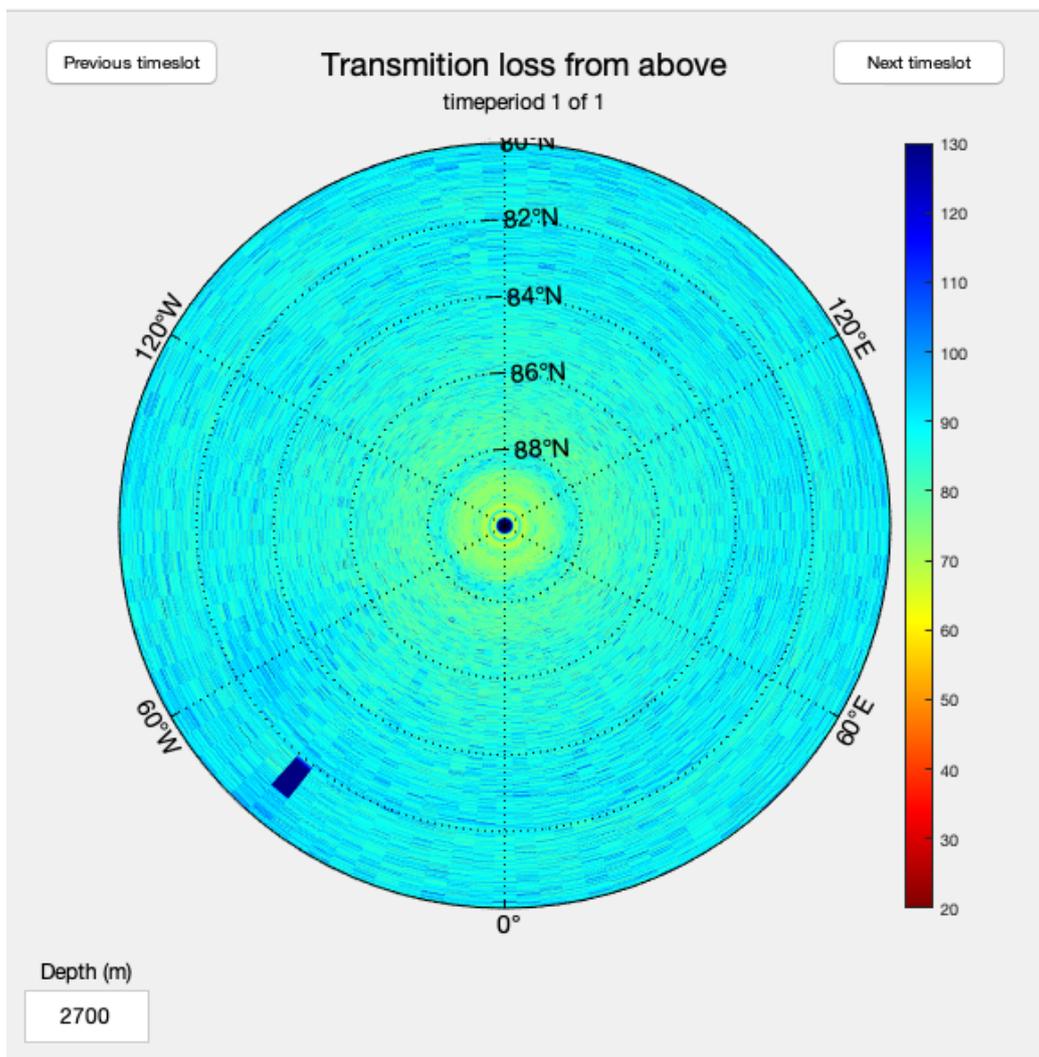


Figure 6.13: The transmission loss from a source to several receivers in a circle surrounding it.

Since the researchers at NERSC would like to look at the sound propagation in space towards noise problems, the simulation of sound in space is really

useful. This can also be very useful for when it comes to making environmental considerations from where to do and not do things that makes noise. If one were to know in which depths the animals have their residence, it is possible to do simulations to check where one should avoid making noise because of the animal habituates.

The visualizations for simulating in space is currently best for RAM, as it has a top down view where the user can see how the sound propagates from the source to all of the receivers at the same time (at a given depth). For the other models, the user currently receive the same results as with only one receiver, but he or she has the ability to click trough the receivers, and study how the models are changing along with the receiver and the new bathymetry and SSP. This view is making it harder to see the bigger picture, than the top down approach.

6.6 Visualization of saved results

By saving the results after the user has executed a model, the opportunity for visualizing the results further than what is done in the default result window arises.

Figure 6.14 shows the anomalies on the sound speed over the range and depth of the measurements. The figure displays the results after the Bellhop model is executed on both several time steps and several receivers. However, the anomaly result is retrieved from the SSPs, so the model choice does not matter. The fact that the model i executed on several receivers and on several time steps is not important in the terms of showing the anomaly for a certain path between the source and one of the receivers at a certain time step. The user can step between the result for the different receivers and time steps by clicking the buttons at the bottom of the window shown in Figure 6.14.

If the user has executed and saved an Eigenray model, he or she can visualize the eigenrays on top of a SSP color map, along with the bathymetry. This is shown in Figure 6.15. The bathymetry and SSPs are available for the results for all models, but in order to be able to put the eigenrays on top, it has to be a result from the Eigenray model.

After executing and saving RAM, it is possible for the user to view the transmission loss from a given depth in a simple plot. This is visible in Figure 6.16.

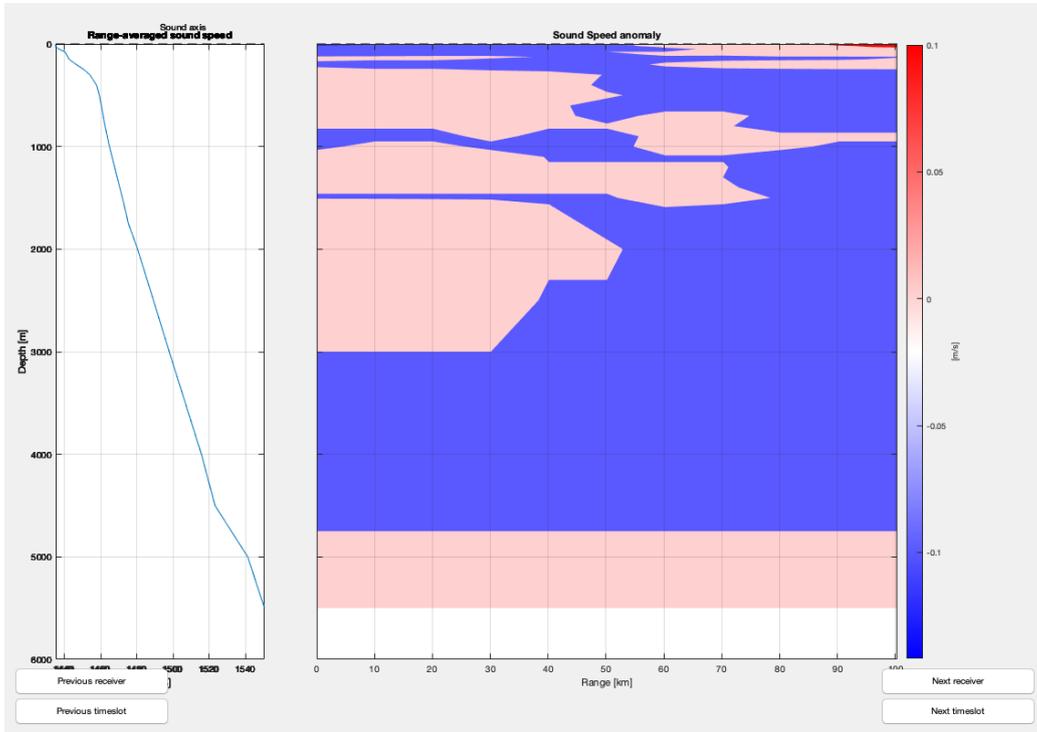


Figure 6.14: The transmission loss from a source to several receivers in a circle surrounding it.

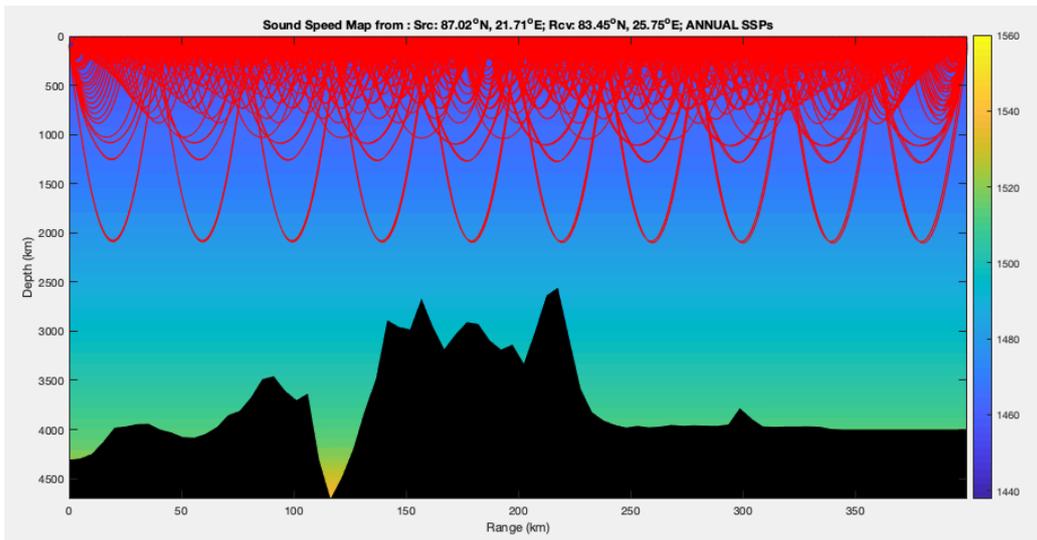


Figure 6.15: The eigenrays plotted on top of the sound speed profile. Bathymetry shown in black.

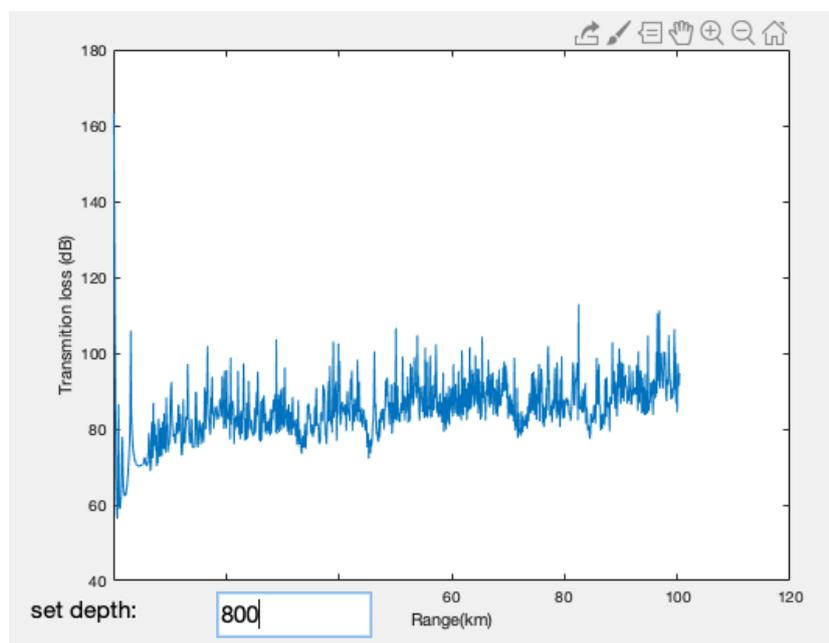


Figure 6.16: The transmission loss on a given depth as a function of range.

In addition to this, it is also possible for the user to view the average loss from the surface to 4500 m depth. Because the depth usually is less than 4500 m, and because the transmission loss is very high when the sound reaches the bottom, the plot of the average loss from the surface to 4500 m depth often reflects the bathymetry on the path. This might not be that interesting for the researchers. To solve this we added functionality for plotting the average from the surface to the shallowest point on the path. We also made it possible for the user to see the average transmission loss from the surface to a given depth. This is all visible on Figure

Saving the results has been of great benefit and it was done for two reasons. To be able to reproduce executions and visualizations and to be able to post process results. Post processing the results can be useful for making statistics and maybe even use the data in for example machine learning contexts.

The new visualizations enables the researchers to look at the data from new and different perspectives. This can be useful to discover patterns that might not be as visible in other visualizations. Visualization of data with many dimensions can be complicated, and different approaches can give very different results. The researchers working within the acoustic field will have great use of the new visualizations not only for analyzing reasons but also to create figures that can be used in research reports and in presentation slides.

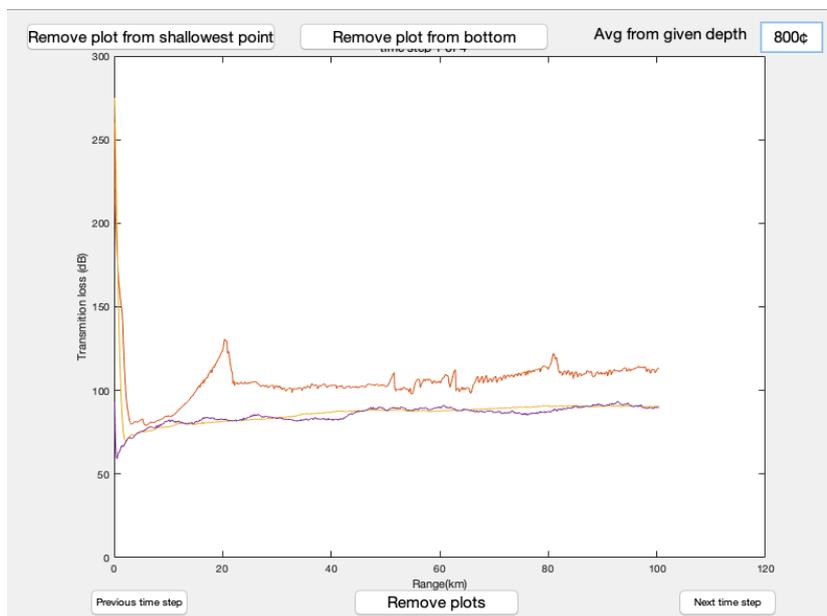


Figure 6.17: The average transmission loss on a path from the surface to a given depth (here 800 m), to the shallowest point on the path and to 4500 m.

6.7 Answer to Research Question 1

The first research question was “How well can existing tools in the Arctic package written in MATLAB be improved regarding usability and extendibility?”. From the results presented above, we can conclude that the tool has been improved both its usability and extendibility. By making the GUI less cumbersome, more clean and easier to understand the usability has improved. Both users that is used to the old version of the Arctic Package and new users can now easier understand how the program is supposed to be used. By grouping the parameter inputs that are similar and related to the same things together in the main GUI, it is easier for the user to understand what parameters are affecting what. By moving the model specific parameters to separate dialogue windows, it prevents the user from entering unnecessary parameters for models that are not being executed. This makes the user less confused and saves time, because the user do not need to spend time wondering what to write in the input fields that are not even getting used.

By moving the model specific parameters to separate dialogue windows, it also improves the extendibility of the tool. It is way easier to add new acoustic

models that requires some new model specific input now that one do not need to change the entire main GUI. Now, the only thing that has to be done in the GUI when adding a new model, is adding it to a drop-down menu where the models are chosen, and creating the new dialogue window. Saving the results also helped a lot when it comes to improving the extendibility. When having the previously executed simulations available, it is possible to add a lot more visualizations of the data. Some of the possible visualizations is already added, but it is still possible to add more, now that the data can be saved.

Chapter 7

Feedback from NERSC

To validate the results and claims made in the thesis, interviews with two users of the Arctic package was held. The results from the interviews with the researchers at NERSC will be presented in this section. The interviews were semi-structured, which is a combination of structured and unstructured interviews. That means that the questions and themes were prepared in advance of the interview, but that one is still flexible and able to follow topical trajectories in the conversation that may stray from the guide when this is appropriate.

7.1 The Interviews

1. On a scale from 1 to 10, where 1 is a lot harder, 5 is no change and 10 is a lot easier. How easy or hard is it to make the map visible on the new version compared to the old one?

Espen: “7”

Florian: “10. However, I do not know what Array# means under the plotting of the map.”

2. On a scale from 1 to 10, where 1 is a lot harder, 5 is no change and 10 is a lot easier. How easy or hard is it to add sources and receivers on the new version compared to the old one?

Espen: “8”

Florian: “9. It took me some seconds to find the placement to where you can add sources under “Source&Receiver Edit and Path Plotting””

3. On a scale from 1 to 10, where 1 is a lot harder, 5 is no change and 10 is a lot easier. How easy or hard is it to adjust the model-specific parameters on the models before running it on the new version compared to the old version?

Espen: “9 or 10”

Florian: “8. I made a Eigenray modeling in the Fram Strait. It looked like one of the rays went through the sea floor, despite the fact that the setting “use bottom” was set. I was wondering if this was due to low horizontal resolution, so I tried setting DR[km]:Z from 5 to 1. I got the comment “Please set DZ(Z) to be a positive number, e.g. 5(km)”. I tried again with 2 instead of 1. It worked, but the ray still went through the sea floor. Other things, like changing the number of rays or the angle range was very easy. Unfortunately, I get some error messages when using the RAM-model too, regardless of the settings. ”

4. How difficult is reading characters on the screen on the new version compared to the old one?

Espen: “The GUI was nice. I had a problem with the degree symbol on mine. Also for example the font on the label on the color bar on the map is a little small.”

Florian: “There is no problems reading the font now. For use in publications some of the fonts might still be too small (colorbar), but it is very common when we make plot-scripts ourselves that we have to adjust the font sizes at the end to make it fit the figure we have made. There is nothing to fault here.”

5. How can you make use of the time variation on the new version of the package compared to what you could do in the old version?

Espen: “It is a big step forward, where you can simulate for all months, so this is something that will be used if you get to run the models on all of the different ones. It is difficult to give a good answer, but it is at least a valuable contribution. ”

Florian: “It was easy to choose an ocean model which has data for different months, e.g “Ocean model” or “WOA” and then plot for e.g. May. Unfortunately I get error messages when I tried this with “Ocean model”, with “WOA” it works just fine. The ray still goes through the ocean floor.”

6. How can you make use of the several receivers on the new version of the package compared to what you could do in the old version?

Espen: “That is twofold. One thing is that you can simulate with Eigenray for example, but now you also have the functionality of RAM - transmission loss and possibly Bellhop. Both for use in the design phase of experiments and analysis of results.”

Florian: “It was easy to choose a space simulation with two receivers. I was a bit surprised that there were only one figure with buttons “prev. receiver” and “next receiver” to step through the receivers. Because of latency it is hard to handle these. It would have been more userfriendly”

7. How does saving the results help you as a user of the Arctic Package?

Espen: “It helps me to run different simulations in one sequence. Especially to look at data that has already been simulated, so that you do not have to repeat the simulation, but look at model specific settings, to validate, quality check and possibly execute the model again with different parameters.”

Florian: “Saving of results are very important, because this is a program for testing experiment setup, where settings are chosen by hand. If we cannot save the results, it’s easy to loose it is easy to lose track of what is done.”

8. How do you think the program has improved or worsen compared to how it was?

Espen: “I would say that you have take a huge step in the right direction. The functionality has expanded significantly, the GUI is better, not only for us that can already use the Arctic package, but eventually also for new users of the system.”

Florian: “The old version was more or less useless to everyone except those who had made it and those who had spent a lot of time with the package. Now one quickly understand how to use the program.”

9. Do you have any other comments?

Espen: “No, not at the moment”

Florian: “You should include a README-file in the package that explain which script to run in order to start the program, who made the program

and when, and maybe even an explanation of how one should cite if one was to use the package in publications.”

7.2 Interpretation of interview answers

The results from the interviews highly suggest that the changes done to the arctic package was for the better. This means that the researchers think that the Arctic Package is easier to use now than before. It can lead to faster and easier modeling, leaving the researchers with time to spend on more researching. This can improve the way of working at NERSC and the new tool is improved in the terms of usability.

The first four questions are questions about the GUI, and how this has made it easier or harder to do some of the tasks. The answers from the interviewees confirms that the new GUI has mad it easier to do these tasks. Both Espen and Florian suggested that the label on the color bar on the map was too small, but that the look was overall good. Florian had some unexpected issues when running the models, this was not due to the GUI, but to the model settings that had some bugs. The reason why the issues did not get detected before may have been because he was running the program om a Windows machine, while the program that was sent to him was mostly tested on mac OS X.

Question 5 to 7 is about some of the new functionality, and how this has added value to the Arctic Package. The answers to question 5, about time steps, states that the time step simulation will be used and that the users think it is valuable. They thought it was easy to find and use, but there was some issues with one of the models. When it comes to the 6Th question they answer that they will make use of the several receivers with Eigenray, but also with the RAM model for transmission loss. It can be used both in the design phase and in the analysis of the results. When running the model with only two receivers, one of them stated that it would be more user friendly to have a separate figure for each receiver. However, running with several receivers is mostly intended for those cases where one has a higher number of receivers, and getting more than 20 separate figures would make it almost impossible to keep track of everything at once. The interviewees also states that saving the results comes in handy when they want to run several different simulations in one sequence in addition to looking at models that has already been simulated and that it is valuable when it comes to twerking the parameters.

The last two questions are more general. Question 8 is about the improvement of the Arctic Package as a whole. The users states that the package has been improved both in terms of expanding the functionality, but also with a better GUI, so that both old and new users of the system can use and understand the tool.

7.3 Discussion

The interviews were held in Norwegian and translated to be understandable for the non-Norwegian speakers reading this thesis. Therefore the interview question presented is not exactly the same as the ones in the interviews, and the answers are also translated to English. However the translations are pretty accurate, so the over all meaning is the same.

Due to the outbreak of Covid-19 there was some limitations to whom was able to test and validate the system. The circumstances made it so only two people was available for interviews after the testing. One of those persons was also an important person in the means of helping with the requirements, the programming and also the writing of this thesis. Also, both persons were using the old version of the tool before, so they were familiar with the functionality and the old GUI. Therefore there might be some bias to the results, and the results from the interviews cannot tell us how a researcher that never had seen the program before would react to the new version.

The original plan was to have some in depth interviews with three or four of the users of the Arctic Package, where two of them were users of the old version of the package, while the other(s) were new users. I would use this interview in combination with a questionnaire about the user interface, where I would like to get informant that don't work in either software nor acoustics. In the questionnaire I would have questions with screenshot examples and ask the informants which they prefer.

Still the results I have gotten from the interviews that I did have is promising. The interviews were held with actual stakeholders, and the fact that they are happy with the results is a good sign, and one might argue that this implies that other users would have similar responses. The two people that was interviewed had different operation systems. One used MAC OS x, while the other used Windows 10. This is an advantage, because the results then are based on the look and feel form two different operative systems.

7.4 Answer to Research Question 2

The second question was “How well can the way of working at NERSC be improved by improving the tools in the Arctic package?”. The result presented above proves that the Arctic package has been improved both in terms of expanding functionality and a better GUI. The functionality is the most important part when it comes to improving the way of working at NERSC.

Saving the results makes it possible to look at previously executed models, and easier to compare the results from different simulations. It also makes it possible to create new types of visualizations of the data the results gives. This makes the way of working at NERSC easier, because they do not need to reproduce the same results several times. The new models is also an important contribution to the package, and makes it easier for the researchers at NERSC to deal with several models when they’re all tucked inn to one program. The simulation in time and space is also very valuable, designing experiments, for comparison and for analysis. The work that is done has a lot of potential in the future as well.

Chapter 8

Threats to Validity

When doing a case study, it is important to keep in mind that there may be things in your study that leads to bias results. There are several types of validity, and Runeson and Höst has a scheme that distinguishes between four aspects of the validity [39]. This chapter will go through four types to classify potential threats to validity in this study.

8.1 Construct validity

Construct validity refers to the degree to which inferences can be made from the operationalizations in the study to theoretical constructs on which those operationalizations were based [40]. Operationalizations is the process that defines fuzzy concepts and allows them to be measured, empirically and quantitatively [41].

Runeson and Höst has an example where the constructs discussed in the interview questions are not interpreted in the same way by the researcher and the interviewed persons. In such a case there is a threat to construct validity. In this case, since the interviewed persons were very familiar with the old system and with acoustic research in general, there is little chance that they interpreted the questions in a different way than me.

8.2 Internal validity

Internal validity is of concern of when casual relations are examined [39]. Runeson and Höst explains that when the researcher is investigating whether one factor affects an investigated factor there is a risk that the investigated factor is also affected by a third factor. There is a threat to the internal validity if the researcher is not aware of the third factor or if he or she does not know to what extent it affects the investigated factor [39].

In the case of this study there are some places this could be an issue. The interview questions might have been leading even though I tried to make the questions as neutral as possible, moreover the interviewed persons where both using the old system before, and wanted a new and improved system. One of the interviewed persons was also involved in the development process. This might have lead to some bias.

8.3 External validity

External validity refers to how well it is possible to generalize the findings, and to what extent the findings are of interest to other people outside the investigated case [39]. In other words, how generalizable the findings are. Do they apply to other people, situations or time periods?

Because there were only two persons interviewed and both of them were familiar with the old version of the system, the findings may not apply to those whom have never used the system before. Generalizability is inherently limited for case studies. However, the project has brought an example of how one can improve and extend a program by using existing models and combine them into one tool, using MATLAB.

8.4 Reliability validity

The reliability aspect of validity is concerned with to what extent the data and the analysis are dependent on the specific researchers [39]. If the same study was conducted later on with another researcher, the result should be the same. If it is not clear how to code collected data or if the questionnaires or interview questions are unclear, this would be a threat to reliability validity[39].

In the case of this study, the first questions in the interviews were very clear, and not a threat to the reliability validity. However, question 5 to 8 is a bit more open. This was done in order to not miss out on any important information from the interviews, as there was only one round of them. The last question is very open, and the answers to this question would probably vary.

When it comes to the result regarding the GUI, other researchers might have other preferences. Even though we tried to stay neutral when studying and coming up with a new GUI, there is a chance that our preferences has affected the GUI. Adding the new models, saving the results, adding functionality for time steps and several receivers and new ocean models would probably be very similar regardless of the researcher.

More work has to be done both with the Arctic Package, but also in the field of expanding MATLAB programs. This thesis is hopefully a step in that direction

Chapter 9

Conclusion

In this thesis, a new and improved version of the Arctic Package has been presented. Computer technology has been used to establish and develop a tool that allows the user to analyze existing ocean models effectively and obtain more information. The new version has an updated GUI, several new acoustic models, and ways to model them over time, and with several receivers. It also provides the users with several different visualizations of the results. The new version of the tool makes it possible for researchers and students study acoustic models in a simpler and more efficient way. The tool is written in MATLAB, with the acoustic models written in other programming languages like Fortran, C and Python.

While developing the new version of the tool a methodology called design science was followed. The goal of design science research is to produce prescriptive knowledge for professionals in a discipline and to share empirical insights gained from investigations of the prescriptions applied in context. Design science has eight guidelines that was followed trough the whole process of development.

The thesis has presented how the design and implementation of the new version of the Arctic Package was carried out. Both in terms of why we chose to stick to MATLAB as a programming language, the choices that were taken with regard to the architecture and the GUI, as well as the visualizations of the results. The results has been presented and interpreted.

Interviews with two of the stakeholders has been held, where they got to express their opinion on the work that has been done to the tool. Their feedback has validated that the Arctic Package has in fact been improved.

With this thesis I wanted to answer the research questions presented in the introduction.

1. How well can existing tools in the Arctic package written in MATLAB be improved regarding usability and extendibility?
2. How well can the way of working at NERSC be improved by improving the tools in the Arctic package?

The existing tools in the Arctic package written in MATLAB has been improved in regards to usability and extendibility. The new GUI makes it easier to use, and because the model specific parameters are in separate windows, the main user interface does not have to be changed in order to add new models. This, among other things, has improved the usability and extendibility of the Arctic Package.

Throughout the thesis we've gotten significant proof that the improvement Arctic package in fact can improve the way of working at NERSC both because it is easier to understand and use the new GUI, but also because the new version has a lot of new functionalities. The researchers and students do not need an excessive number of tools in order to run many different models. The interviews held with the researchers confirmed that the way of working at NERSC has been improved.

Chapter 10

Further Work

When developing programs like the Arctic Package, there is almost never a time where one is a 100% satisfied with the product. Even if everything works smoothly and as intended, there is almost always some functionality that can be added, or small things that could be improved. This chapter will contain thoughts about what could be done with the Arctic Package if someone were to continue working on the program.

10.1 MATLAB

The Arctic Package is written in MATLAB, for better or worse. MATLAB is a proprietary product of The MathWorks[42]. The users are dependent on a vendor for products and services, in this case they're dependent on MathWorks. This means that the user must have a valid license of MATLAB installed on his/her computer in order to run the program. Since the license is not freeware, it might provoke the user's negligence of buying one and therefore also provoke the user to not use the tool. MATLAB is not free, and the Arctic Package is dependent on MATLAB to run. In the long run it would make sense to rewrite the package in another programming language, for easier distribution of the package to others without costs.

Even though MATLAB offers a set of toolboxes that allow the developer to create windows-based applications, there might be more powerful graphical libraries belonging to other programming languages that offer better options and more possibilities to developers. These programming languages would maybe make it easier to customize and achieve a user-friendly interface.

10.2 GUI

Even though we've improved the old GUI to be more user-friendly and easier to understand, there is still room for improvement here. With the limitations that MATLAB had, the end result of the GUI is good, but a program that does not consist of many small dialogue windows might be better. It would look less cluttered without the many dialogue windows.

It would be nice to have a more visually pleasing design in terms of more colors and button design. The GUI could also still be improved in terms of flexibility when using the program.

10.3 Machine Learning

Machine Learning(ML) is a hot topic today. ML studies algorithms that improve with and learn from experience[43]. It is a broad family of statistical techniques for automatically detecting and utilizing patterns in data.

ML is quickly growing in acoustics. The rapid development has convincing results that indicate a promising future in the area [44]. In the article "Machine learning in acoustics: theory and applications" five acoustics research areas is highlighted. One of those are source localization in ocean acoustics [44]. ML can discover complex relationships between features if its given sufficient training data. For further work on the Arctic Package it would be very interesting to investigate how one could make use of ML in this context.

10.4 Visualizations

If someone were to continue working with the Arctic Package there would also be a lot of potential in the terms of visualizations. Much of the data saved from the executed models are still not visualized, and there is a lot of potential for making new and useful visualizations.

When executing a model with several receivers, it would be possible to visualize the data in a three dimensional model, so it would be possible to observe how the sound behaves in a three dimensional area. It would be interesting to see how one could visualize for example eigenrays in a three dimensional area.

When running a model with several time steps it would be interesting to see the results changing automatically, as an animation. This would be even more interesting if the time steps were smaller, for example that each day had it's own SSP data.

When running a model with both several receivers and with several time steps it would be possible to combine the two visualizations mentioned. A three dimensional animation would be interesting, to see how the sound behaves over time in a three dimensional area.

There might also lay some potential in combining models as well. There is already possible to run MPIRAM and the Eigenray model with time fronts at the same time, but then the program runs both the models before plotting the results. Being able to find the results that has been executed on the same paths and combining the visualizations of these would be very interesting.

10.5 More

The executions of the models results in a lot of information, some of which is harder to visualize than others. The researchers at NERSC have said that they want to keep statistics in the area. To make this useful it is important to know what kind of information is important and of interest for the researchers and others with access to the data. The work on finding out which information to extract and how to save and make this data useful in statistics could be of great benefit for NERSC.

To combine acoustic models with ocean models can also be beneficial for the researchers at NERSC. Continuing to make this possible would make the Arctic Package even more useful.

Bibliography

- [1] University of Rhode Island and Inner Space Center. How is sound used to measure temperature in the ocean? <https://dosits.org/people-and-sound/research-ocean-physics/how-is-sound-used-to-measure-temperature-in-the-ocean/>. Accessed: 01.04.2019.
- [2] WWF. Climate change. <https://www.wwf.org.uk/what-we-do/area-of-work/climate-change-and-energy>. Accessed: 15.01.2020.
- [3] Former Foreign minister Borge Brende. The Arctic: Important for Norway, Important for the world. https://www.regjeringen.no/no/aktuelt/arctic_harvard/id2406903/. Accessed: 15.01.2020.
- [4] Holly Shaftel. The effects of climate change. <https://climate.nasa.gov/effects/>. Accessed: 19.02.2020.
- [5] Norwegian Polar Institute. Climate change in the arctic. <https://www.npolar.no/en/themes/climate-change-in-the-arctic/#toggle-id-1>. Accessed: 19.02.2020.
- [6] WWF. THE ARCTIC. <https://www.wwf.org.uk/where-we-work/places/arctic>. Accessed: 15.01.2020.
- [7] Jens Hovem and Hefeng Dong. Understanding ocean acoustics by eigenray analysis. *Journal of Marine Science and Engineering*, 7:118, 04 2019.
- [8] Nansen Environmental and Remote Sensing Center. Nansen environmental and remote sensing center. <https://www.nersc.no/>. Accessed: 27.02.2020.
- [9] Nansen Environmental and Remote Sensing Center. About us. <https://www.nersc.no/about>. Accessed: 03.02.2020.

- [10] Norsk polarinstitutt. Klimaendringer i arktis: status og framtid. <https://www.npolar.no/tema/klimaendringer-arktis/>.
- [11] Ellen Øseth. *Klimaendringer i norsk Arktis Konsekvenser for livet i nord*. Norsk Polarinstitutt Rapportserie 136. Norsk Polarinstitutt, 2010.
- [12] Emilia Botnen Van den Bergh. Further development of a software for acoustic propagation modelling. Technical report, University of Bergen, Western Norway University of Applied sciences, 2020. To be published.
- [13] University of Rhode Island and Inner Space Center. How does sound move? <https://dosits.org/science/movement/how-does-sound-move/>. Accessed: 21.04.2020.
- [14] University of Rhode Island and Inner Space Center. Refraction. <https://dosits.org/science/movement/how-does-sound-move/refraction/>. Accessed: 21.04.2020.
- [15] James Bowlin, John Spiesberger, Timothy Duda, and Lee Freitag. Ocean acoustical ray-tracing software ray, 10 1993.
- [16] Bruce Howe Chris Eggen and Brian Dushaw. A MATLAB GUI for Ocean Acoustic Propagation. Technical report, Applied Physics Laboratory University of Washington, 1992. Accessed: 27.04.2020.
- [17] University of Rhode Island and Inner Space Center. Tutorial: Sound channel. <https://dosits.org/tutorials/science/tutorial-sound-channel/>. Accessed: 08.06.2020.
- [18] National Oceanic and Atmospheric Administration. What is bathymetry? <https://oceanservice.noaa.gov/facts/bathymetry.html>, 06 2018.
- [19] Matthew Dzieciuch, Peter Worcester, and Walter Munk. Turning point filters: Analysis of sound propagation on a gyre-scale. *The Journal of the Acoustical Society of America*, 110(1):135–149, 2001.
- [20] Michael Porter and Edward L. Reiss. A numerical method for oceanacoustic normal modes. *The Journal of the Acoustical Society of America*, 76(1):244–252, 1984.
- [21] Michael D Collins. Users guide for ram versions 1.0 and 1.0p. Technical report, Naval Research Laboratory, 2015. Accessed: 04.05.2020.
- [22] Michael B. Porter. The bellhop manual and users guide: Preliminary draft. Technical report, Heat, Light, and Sound Research, Inc., 2011. Accessed: 04.05.2020.

- [23] Jens M. Hovem. Ray trace modeling of underwater sound propagation. In Marco G. Beghi, editor, *Modeling and Measurement Methods for Acoustic Waves and for Acoustic Microdevices*, chapter 23. IntechOpen, Rijeka, 2013.
- [24] F. Lindberg, B. Holmer, and S. Thorsson. Solweig 1.0 modelling spatial variations of 3d radiant fluxes and mean radiant temperature in complex urban settings. *International Journal of Biometeorology*, 52(697713), 2008.
- [25] Deepak Jeswani Dewan. Solweig a climate design tool. Master’s thesis, University of Gothenburg, 2009.
- [26] S. Rowan, S. Tygier, R. Appleby, and R. Barlow. M1.4 milestone report: Profiling merlin. Technical report, University of Huddersfield, University of Manchester, 2017. Accessed: 06.06.2020.
- [27] Nansen Environmental and Remote Sensing Center. Polar acoustics and oceanography. <https://www.nersc.no/group/polar-acoustics-and-oceanography>. Accessed: 03.02.2020.
- [28] NERSC. About CAATEX. <https://caatex.nersc.no/node/1>. Accessed: 2020-05-20.
- [29] Nansen Environmental and Remote Sensing Center. CAATEX: Coordinated Arctic Acoustic Experiment. <https://www.nersc.no/project/caatex>. Accessed: 03.02.2020.
- [30] R.R.R. Reubens. To craft, by design, for sustainability: Towards holistic sustainability design for developing-country enterprises, 2016.
- [31] Alan Hevner, Alan R, Salvatore March, Salvatore T, Park, Jinsoo Park, Ram, and Sudha. Design science in information systems research. *Management Information Systems Quarterly*, 28:75–, 03 2004.
- [32] Cambridge University Pres. Cambridge business english dictionary. In *Cambridge Business English Dictionary*. Cambridge University Press 2020, 2020.
- [33] Jorge Allende. Rigor the essence of scientific work. *Electronic Journal of Biotechnology*, 7(1), 2012.
- [34] NOAA/NGDC&WDC. The international bathymetry chart of the arctic ocean. <https://www.ibcao.org/>. Accessed: June 12, 2020.

- [35] National Centers for Environmental Information. World ocean atlas. <https://www.ncei.noaa.gov/products/world-ocean-atlas>. Accessed: June 12, 2020.
- [36] NASA ECCO group. Estimating the circulation and climate of the ocean. <https://www.ecco-group.org/>.
- [37] Richard; Johnson Ralph; Vlissides John Gamma, Erich; Helm. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994.
- [38] Jakob Nielsen. 10 usability heuristics for user interface design. *Nielsen Norman Group*, 1(1), 1995.
- [39] Per. Runeson and Martin. Hst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131 –, 2008.
- [40] W. Trochim. Research methods knowledge base. <https://conjointly.com/kb/>. Accessed: June 15, 2020.
- [41] Martyn Shuttleworth. Operationalization. <https://explorable.com/operationalization>. Accessed: June 15, 2020.
- [42] MathWorks. Mathworks. <https://se.mathworks.com/>. Accessed: 2020-05-27.
- [43] Vikramjit Mitra, Hosung Nam, Carol Y Espy-Wilson, Elliot Saltzman, and Louis Goldstein. Articulatory information for noise robust speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):1913–1924, 2010.
- [44] Michael Bianco, Peter Gerstoft, James Traer, Emma Ozanich, Marie Roch, Sharon Gannot, Charles Deledalle, and Weichang Li. Machine learning in acoustics: a review, 05 2019.