Further development of a software for acoustic propagation modelling

To improving acoustics and oceanography research of the Arctic Ocean

Emilia Botnen Van den Bergh

Supervisors:

Kjell-Eivind Frøysa

Rogardt Heldal

Hanne Sagen

Espen Storheim

Master's thesis in Software Engineering at

Department of Computing, Mathematics and Physics,
Western Norway University of Applied Sciences

Department of Informatics,
University of Bergen

June 2020

Western Norway University of Applied Sciences

UNIVERSITAS BERGENSIS

Abstract

Simulation tools are important and helpful for researching, as it provides a way in which alternative designs and planes can be evaluated without having to do experiments on a real system. In the researching of the Arctic Ocean it is impossible to cover an experiment of the whole Arctic area. This means that it adds a great benefit to have a program that can design and help understand a larger area than what can be done experiment and observations on, in addition to be able to design and plan the experiment with the help of being able to run prediction simulation on the area of the experiment. This master thesis will present the implementation done to extend an existing program, the Arctic package, that supported simulation runs of two acoustic models. The aim of the project was to make the tool a greater resource in the research of the Arctic Ocean. The project resulted in a tool with a larger range of new functionalities of running several simulations models that are relevant to the research of acoustic in the Arctic Ocean and a more user friendly graphical user interface.

Acknowledgements

First, I would like to thank my supervisors at Western Norway University of Applied Sciences, Kjell-Eivind Frøysa and Rogardt Heldal, for their guidance and feedback and guidance through the writing of this thesis.

My gratetude also goes to my supervisors at the Nansen Environmental and Remote Sensing Center. Great thanks to Hanne Sagen for her engagement in the project. And a special thank to Espen Storheim for the continual help throughout this master with feedback on both the development and the thesis.

I would also like to express my gratitude to other people at the Nansen Environmental and Remote Sensing Center. Torill Hamre for welcomed us to the Nansen center at the beginning and providing us with the Arctic package. Then the secretaries that arranged a nice working desk to work on. I also appreciate that Helene Langehaug and Florian Geyer took time to participate in the usability test that gave useful results.

I am also very grateful to Frida Tryggestad Klockmann, for the good cooperation on the development of the program. It has been a pleasure to work together, and of great (and priceless) help by being able to discuss and get encouraging support when needed during project period.

Finally, I would like to thank my friends and family for their encouragement and support, and also some occasional but helpful feedback.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Observing and understanding the rapid changes taking place in the Arctic Ocean is crucial to assess the role of the ocean in climate change. Human activity in the area is increasing as the ice melting provides e.g. opportunities for new transport routes between the continents. This puts pressure on the vulnerable Arctic environment. In order to understand the impact of this, one needs to improve data from the sea-ice atmosphere in order to inform and enable sustainable development while preserving this fragile environment.

The Nansen Environmental and Remote Sensing Center (NERSC) is an independent non-profit research foundation conducting fundamental and applied environmental and climate research [1]. In their work, NERSC uses several acoustic networks to improve ocean observations in the Arctic Ocean. The Nansen Center is organized in seven thematic research groups, in which the project-based research activities are implemented. One of these groups is the Acoustics and Oceanography Group (AOG), that performs fundamental acoustic research for applications within sea ice research, oceanography, environmental monitoring, underwater communication and geo-positioning.

This research group uses today a program that supports two acoustic models that capture and visualize ocean data, but to gain deeper and better understanding of data, improvements in the program would be very helpful. The program has the functionality for comparing the observation data collected along different transects under the ice in the Arctic Ocean with the temperature data calculated by climate models. By this, the project expects to improve the climate models to predict the climate development in the Arctic ocean and provide a better estimate of the uncertainty in the model calculations. The program, called the *Arctic Package*, can easily run acoustic models on the paths between moorings with sound sources and receivers, which allows for comparison between measured data and model calculations on the same paths.

1.2 Problem Description

Modelling of the sound propagation in the ocean is important for several applications, such as noise estimation and and underwater communication. Over the years, several propagation models have been implemented in different programming languages. Some, developed in the early 1980's and 1990's, are still being used today [2]–[4], and are even being developed further [5]–[7]. The input files and formats can sometimes be complicated and not very well documented. It is also time consuming to learn how to operate the models, obtain the required input (such as sound speed and depth of the seafloor), and visualise the results once the simulations are complete. This was the motivation for the initial development of the Arctic Package.

For better and easier research, it is important to have good tools that are optimized to utilize the accessed data and further provide a good visualization of the result and the changes that occur. This master thesis is based on a project of further developing an existing program, Arctic package for acoustic propagation modelling. This program can use ocean data from several databases for acoustic modelling and is a tool for researching the Arctic Ocean and comparing it to real data collected by observations.

The main research questions in the present work are as follows:

Q1. How to expand and improve the Arctic Package, a tool for acoustic propagation modelling, for **easier** and **relevant** research in the fields of acoustics and oceanography?

Q2. How can the tool Arctic Package be more user friendly by increasing the **usability**?

The task will be to find out how to increase the functionality of this program, understand the user problems and determine the priority of the problems and find solutions to them. The goal is to improve the system so that it is better for the user, by improving design, expanding visualization capabilities, simplifying compilation and providing more functionalities. As the users of this program will primarily be researchers, it is important to look for what is relevant and will make the researching easier and better.

This master project in Software Engineering was done in close cooperation with Frida Tryggestad Klockmann [8]. Starting with understanding the program and establishing possible improvements in the aspect of the software. Continually evaluating and discussing the software solutions for developing different parts of the program, including functionalities and improving the graphical user interface.

1.3 Thesis Outline

The thesis is outlined in the following manner:

Introduction - Chapter 1 introduce the motivation, research problem and give an outline of the thesis.

Background - Chapter 2 provides a presentation of background information about the different aspects of the program further developed in this master and about data sets and acoustic models that will be added to the program in this project.

Related work - Chapter 3 introduce related work that are considered in the choices made in this master project.

Methodology - Chapter 4 gives an introduction to the Design Science Research Methodology and how it was used in this master, along with how the research methods and how the research data was collected.

Design and Implementation - Chapter 5 describes the design and implementation of the Arctic Package, the program further developed in this master.

Results - Chapter 6 presents the result of the extended program.

Usability Test - Chapter 7 presents the result of the usability tests performed on the original version and new version.

Discussion - Chapter 8 will provide interpretation and discussion on the implications and reflections on the results.

Conclusion - Chapter 9 provides the conclusion of the research questions.

Further work - Chapter 10 will present ideas for further work after this project.

Chapter 2

Background

In this chapter background information of some oceanography data models and databases related to this master project will be given introductory descriptions. Beginning with information about the specific databases that are or will be used in the program, this is different data sets with Arctic Ocean properties. Next will be a presentation of speed of sound and the acoustic models that were or will be supported by the package. Then an explanation of the program, The Arctic Package, that will be further developed on this master project, and its different aspects on graphical user interface, structure and visualizations.

2.1 Databases

For the program to perform the model calculations it needs some data. The program supports the use of several different data sets that can be used for the model prediction or making of the map. The map options are based on the data set and what type of parameter the map will be coloured by, which can be bathymetry, temperature or salinity. While model prediction will be based on data of sound speed.

2.1.1 Sound speed profiles

Sound speed profiles are some of the data the program calculates and uses to get predictions and information for the researching of climatic changes in the Arctic Ocean. There are two databases with sound speed data that the package support, this is the forth version of the Estimating the Circulation and Climate of the Ocean (ECCO) data set and the World Ocean Atlas data set. The "Estimating the Circulation and Climate of the Ocean" (ECCO) group has a goal to give the best possible estimates of ocean circulation and find it role in climate, the result is the ECCO data sets [9]. The World Ocean Atlas (WOA) is a data sets of the 'Ocean Climate Laboratory of the National Oceanographic Data Center' in USA [10]. In it is a climatology of fields of on-site ocean properties for the world ocean. Some of these properties are the sound speed, temperature, salinity and buoyancy frequency on different depths and coordinates. The fields are 3-dimentional, and in the older versions the data are interpolated onto 33 standardised vertical depths from the surface to the seafloor (5500 m). In the 2018 version this is increased to 102 depths. For this data set averaged fields are produced for annual, seasonal and monthly timescales.

Other data sets with data about the Arctic Ocean are the Ocean Model and a data set using the GECCO model. The Arctic Ocean state data set [11] contains estimations of a 10-year ocean synthesis (2007-2016) obtained by assimilating available observations of sea ice and ocean parameters into the GECCO model. The Ocean Model data set contains temperature and salinity data covering the the Fram Strait.

2.1.2 Bathymetry

Bathymetry is the measurement of depth of water in oceans, seas, or lakes. In the program the map can be coloured by the depth and be plotted in visualizations with SSP or model calculations. International Bathymetric Chart of the Arctic Ocean (IBCAO) is a database that has collected all available bathymetric data north of 64° North [12]. It is made for giving a detailed and accurate knowledge of the depth and shape of the Arctic seabed.

2.2 Acoustic propagation models

Acoustic propagation models are models that in different ways predict the sounds propagation in water. Most acoustic models rely on a limited set of input parameters, such as the depth and frequency of the acoustic source, the distance to the receiver, the bathymetry and sound speed profiles between the source and receiver. The sound speed profile can either be range-independent, where the sound speed profile $c(z)$ is constant along the transect, or range-dependent, where the sound speed varies with range, i.e. $c(r, z)$.

A significant amount of acoustic models are available on the Ocean Acoustic Library website [13], as well as relevant documentation. The models are typically divided into four main categories, by the underlying mathematical foundation: 1) Parabolic equations (PE); 2) normal modes; 3) raytracing, and 4) wavenumber integration. The mathematical derivations can be found in detail in [14].

2.2.1 Speed of sound

The speed of sound in the ocean varies from place to place, season to season, morning to evening and with water depth. Even though the variations are not large, they have important effects on how sound travels in the ocean. There are 3 factors that affects the sound speed, namely temperature, salinity and depth. The depth impacts the sound because of the ocean pressure due to the weight of the overlying the water. Since the speed of sound in water increases with the water temperature, salinity and pressure, data about the sound speed and their ray traces will give information that can be processed to give information about the 3 factors.

2.2.2 Eigenray

Eigenray [15] is a ray-tracing code written in FORTRAN which is based on the original RAY code by Bowlin [16]. Raytracing is a technique where an infinitesimally thin ray with a given launch angle, is propagated through the ocean towards the receiver depth. The ray path is the spatial path in range and depth between the source and receiver. An eigenray is a ray that connects a specific source with a receiver at a given depth. The model is using sophisticated logic in addition to exhaustive search to find eigenrays at long ranges accurately and efficiently.

There are two options under the Eigenray that decide the calculation and plotting that are done in a run of the model. After a run of Eigenray the top subplot will show plotting of turning points of each eigenray (the angle the ray arrives in) vs. the travel time. The middle panel will show depending on the run choice, Eigenray will show the bathymetry with the raytracing, Timefront will show the time front of the eigenrays, meaning the time and depth of the rays arrival. Each ray has a launch angle from the source, an arrival angle at the receiver, depth at the receiver, and travel-time.

2.2.3 Bellhop

Bellhop [3] is a ray tracing approach for predicting acoustic pressure fields in ocean environments. The model is made by Michael Porter and originally written in FORTRAN. Versions in MATLAB and Python also exists. There are variety of output that can be produced by the Bellhop model, it has both raytracing (eigenray or ray) and transmission loss at different coherence levels (incoherent / semi-coherent / full coherence). This model is not in the original program, but it is a goal to add support for it in the extension.

2.2.4 Range dependent Acoustic Model

The Range dependent Acoustic Model (RAM) uses a split-step Padé solution for the parabolic equation method (PE) method that is based on assuming that outgoing energy is dominates backscattered energy and factoring the operator in the frequency-domain using a rational function to obtain an outgoing wave equation [4]. There are two different PE-codes that have been implemented in the present work. The first is RAM [4], which calculates the transmission loss (TL) along the source-receiver path. The TL is a measure of how much the sound is reduced at a given point from a source. The second one is MPIRAM [17], which calculates the time front at a given receiver range as a function of frequency and bandwidth. Both of these are potential models to be added support of in the further development of the program extended in this master project.

2.2.5 KRAKEN

KRAKEN is a normal mode program where work began in the early 1980's by Michael Porter [2]. This technique is based on calculating a depth-dependent equation, which results in a series of normal modes. The contributions from each mode are in turn summed to construct the complete acoustic field [14].

The specific version of KRAKEN included in the Arctic Package is written in MATLAB by Brian Dushaw [18], including a MEX wrapper [19], which simplifies how the program is called from within MATLAB. The output from this version of KRAKEN is the normal modes and the time front, similar for the MPIRAM code discussed in Section 2.2.4.

2.3 The Arctic package

The program was created by Brian D. Dushaw, a former employee at NERSC and associate professor at Applied Physics Laboratory, University of Washington. The primary intention of the program was to simplify acoustic modelling, namely to easily choose source and receiver positions, then for so running one of the 2 acoustic propagation models supported by the package on the path between the source and receiver. Among other things, the Arctic Package has been used to simulate sound propagation in advance of acoustic experiments, i.e. during a project's design phase.

The program is written in the MATLAB programming language [20], and was developed on a Linux machine. It acts as a wrapper for interfacing sound speed and bathymetry data, setting up the input files and running the acoustic propagation models, as well as visualizing the results. Everything is accessed via a Graphical User Interface (GUI).

2.3.1 Graphical user interface

The graphical user interface (GUI) as it was, see Figure 2.1, had however a lot of potential for improvements. The setup of the graphical user interface was chaotic, with a lot of user interface controls (buttons, edit boxes, popup menus etc.) which was found to have disorganized positions in the start-up panel. For some of them it was also hard to (intuitively) understand their underlying functions.

By opening the original version of the program, in additions to the GUI (Graphical User Interface), an empty window will appear. Here the map will appear when it is set, and sources and receivers are plotted on this map. This is illustrated in Figure 2.2 with bathymetry colouring and plots of sources and receivers. The user also has the possibility to plot the available source-receiver paths, illustrated

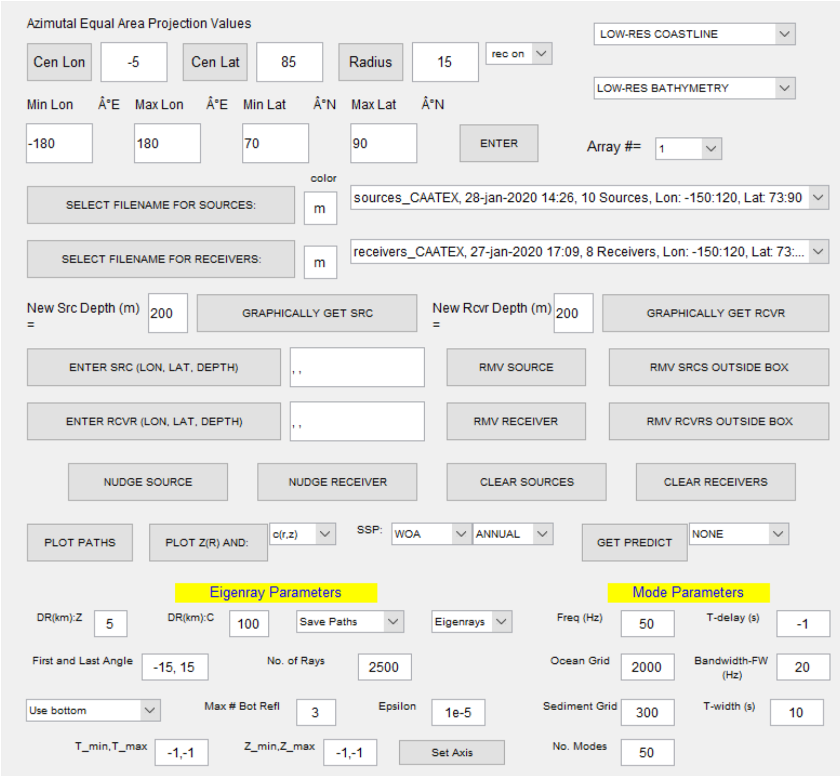
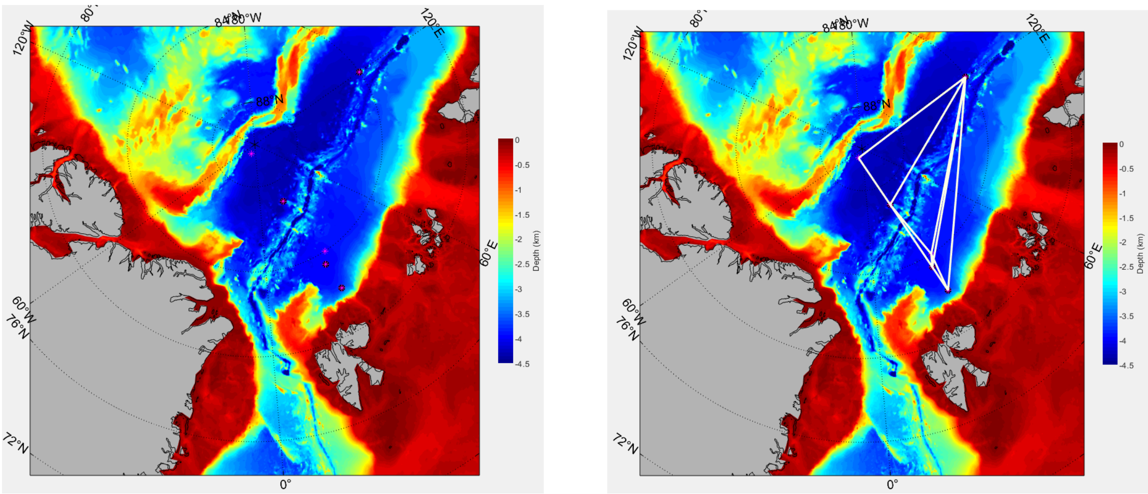


Figure 2.1: The main GUI of the original package.

by lines. To define the path for the model calculations, the source and receiver is selected directly on the map by clicking at or close to their actual plot. The program will automatically trace the source- and receiver-positions closest to the respectively clicks, and will run the chosen model along the path between them.



(a) (b)

Figure 2.2: The map shown in the Arctic Package with sources (pink stars) and receivers (black circles) plotted on (a), and with lines indicating the possible source-receiver paths (b). The colour plot shows the bathymetry (i.e. the depth of the sea floor).

2.3.2 Structure

Figure 2.3 shows a block diagram that illustrate the main functions of the original program, the Arctic Package. The blue blocks illustrate the databases with parameters for map colouring and the model prediction calculations. The diamonds illustrate the main functions of the program: set map, get sources/receivers and get prediction.

The first function that needs to be done is 'Set map', this make the map appear with colouring by bathymetry, temperature or salinity parameters from a database on the second window that appear when opening the program. This is needed to place possible sources and receivers that will give the options of paths to run a model prediction on.

The 'Get sources/receivers' function place the sources/receivers positions from the chosen files to the map, this makes it possible to pick the saved sources and receivers by clicking close to the plots on the map. 'Get prediction' is the running of a model prediction calculation that goes trough several steps before showing the visualization of the model prediction. First the choosing of source and receiver with the map, then getting the SSP and bathymetry from the databases, followed by the setup of input files, modelling, and finally a visualization of the results.

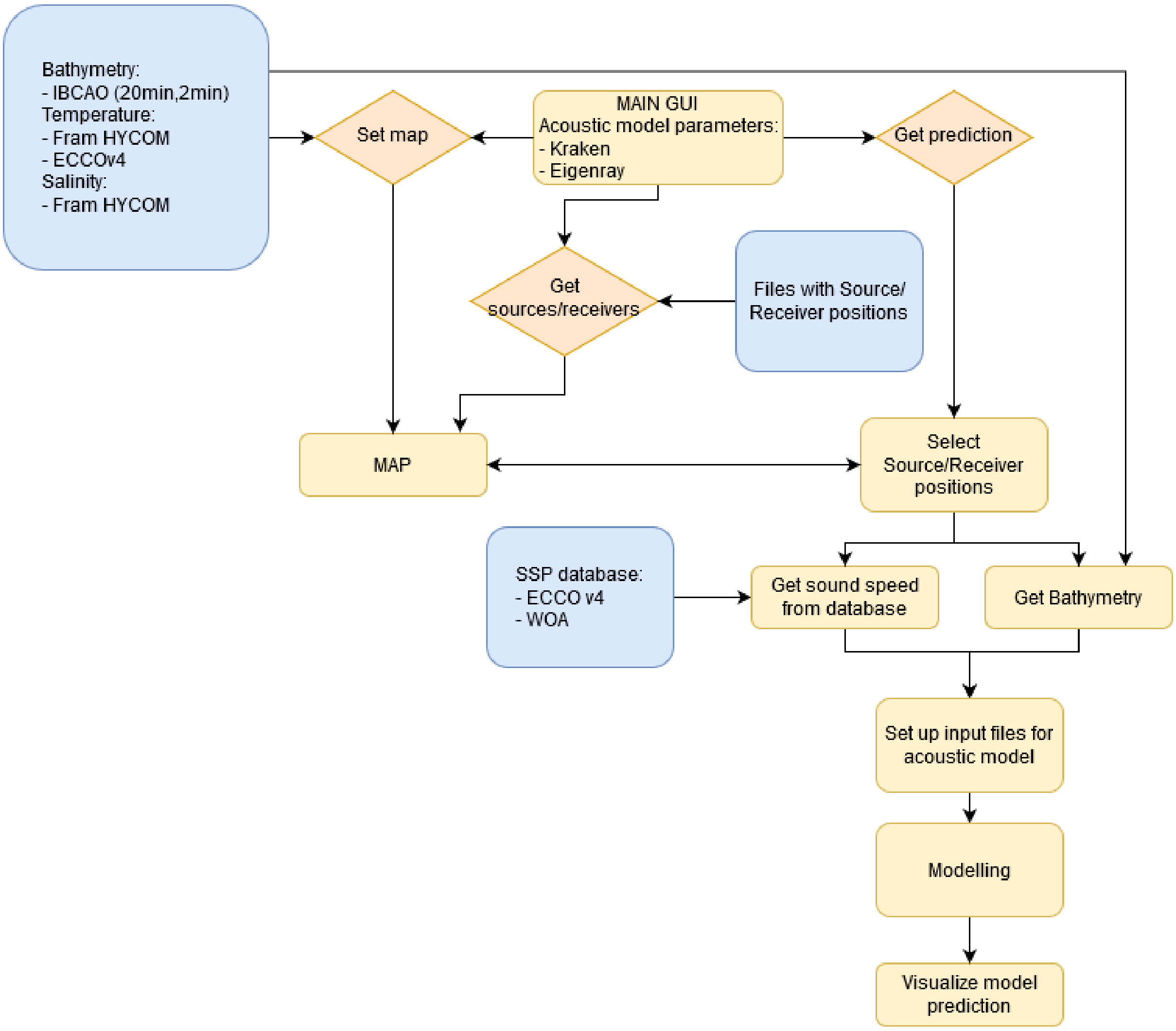


Figure 2.3: Block diagram illustrating the structure of the original package.

2.3.3 Visualizations

The Arctic package provides visualization to give an better understanding of the Arctic Ocean. This with a map of the Arctic Ocean and illustrations of the model prediction simulation results.

Map

The map give a overview of the Arctic Ocean by colouring the ocean by parameters of depth, temperature or salinity. This opens for a better understanding of the levels of these parameters. The temperature differs at different depths of the ocean, so the map can visualize different depths that there are temperature data on. The original program version can visualize three different depths (5 m, 95 m, 477 m) of the temperature parameter in ECCOv4, which shows the variation of the temperature in the Arctic Ocean. In addition to these, there are a HYCOM data sets of temperature and salinity at 300 m depth that also can colour the Fram Strait part of the map. This is shown in the Figure 2.4.

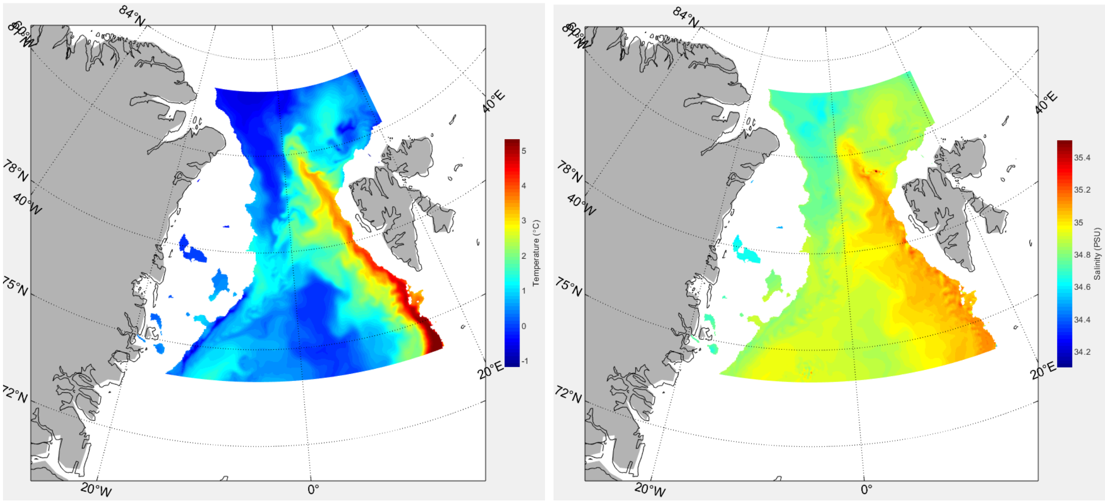


Figure 2.4: Examples of temperature (left) and salinity (right) maps in the original package.

Acoustic propagation models

In the program there are support for two acoustic models, these are visualized after a prediction run like shown in the Figure 2.5 and 2.6. The bottom panel of all the visualizations show the normalization of sound speed profile at 5 different ranges of the path (red lines), as well as the bathymetry (blue line). On the left, Figure 2.5 shows time fronts computed with Eigenray. On the right shows ray paths calculated with Eigenray. Figure 2.6 shows time fronts computed with KRAKEN.

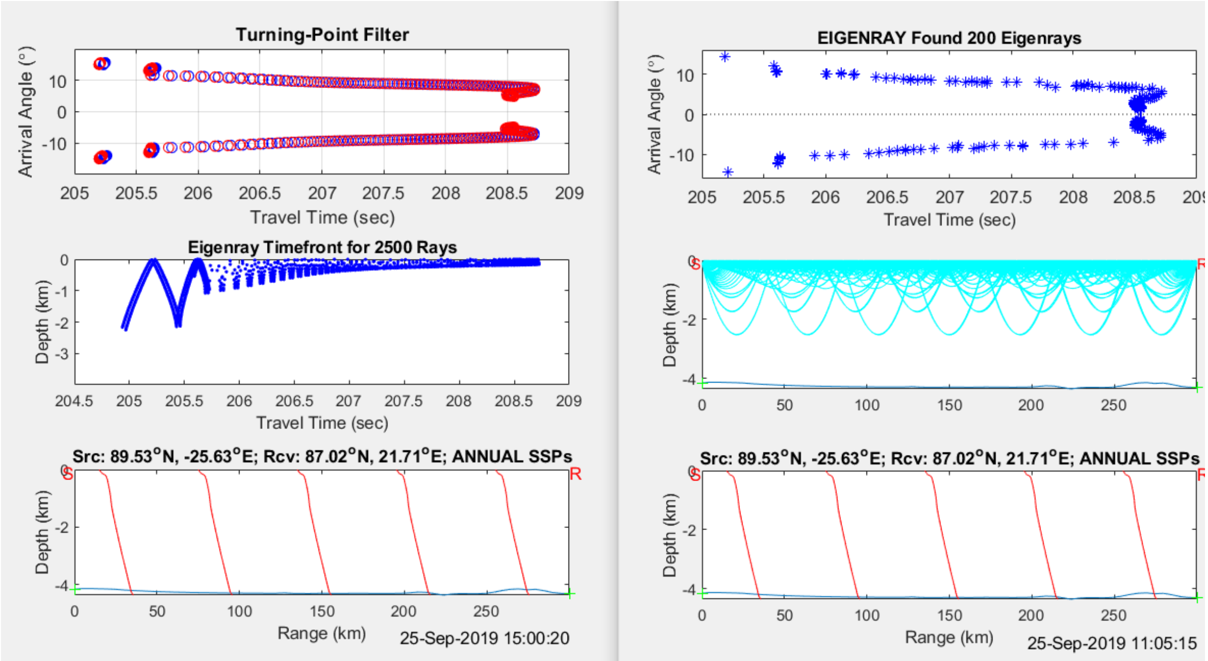


Figure 2.5: Examples of output from Eigenray in the original package.

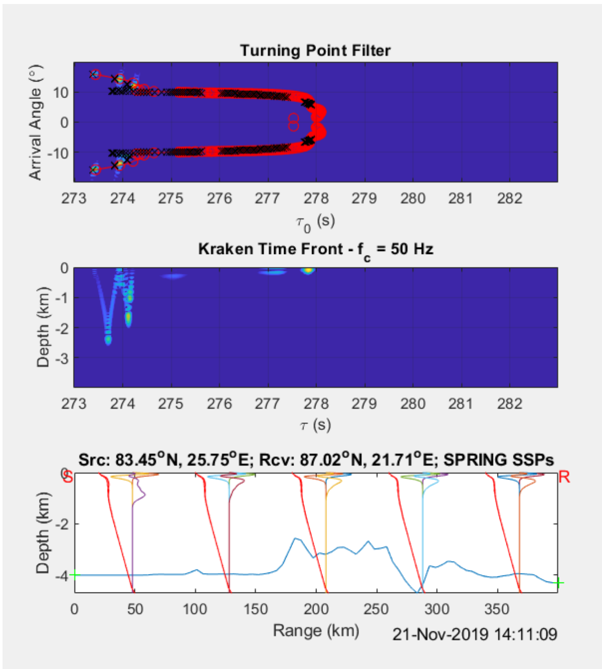


Figure 2.6: Examples of output from KRAKEN in the original package.

2.4 Summary of the Arctic package

The Arctic package functionalities before the work of this master project can be summarised to:

1. Visualize a map covering the Arctic Ocean coloured by specific data set parameters saved in a database containing:
   - Bathymetry data of two different resolution by International Bathymetric Chart of the Arctic Ocean (IBCAO)
   - Temperature data by ECCOv4
   - Temperature and Salinity data of Fram Strait by HYCOM
2. Plot sources and receivers on this map:
   - adding source/receiver plotting by getting positions from file
   - adding source/receiver plotting by coordinates of lat long and depth
   - adding source/receiver plotting by depth and choosing position directly on map
   - moving source/receiver plotting on the map
   - removing source/receivers plotting on the map
3. Being able to run normalized profiles on 5 different positions on the chosen path of sound speed, temperature, salinity or bouncy frequency taken from data sets:
   - ECCOv4
   - WOA
4. Being able to run the two supported models on paths between one of the sources and one of the receivers with same data sets as normalize profile plotting. The models running options being:
   - Eigenray
     - Eigenray
     - Time front
   - KRAKEN

Chapter 3

Related Work

In this chapter work related to this master will be presented. An other master project with similarities to this master is presented with comparing difference and similarities that lead to same or different choices in the project. Then a presentation of the CAATEX project that today are using the program, and improving the program will make it more convenient for the project's research matters.

3.1 SOLWEIG

SOlar and LongWave Environmental Irradiance Geometry is a radiation model that estimate climate and analyse the complex interaction between the thermal environment and urban design [21]. The model was developed by the Urban Climate Group of the Department of Earth Science at University of Gothenburg, Sweden. In the master thesis of Deepak Jesawni Dewan, "SOLWEIG A climate design tool"[21], the development of a graphical user interface to the model, as it before was a command-line based software. Like this master project the model was written in MATLAB and is for researching climate. Dewan developed the graphical user interface in Java and integrated the SOLWEIG model using MATLAB Runtime. The MATLAB Runtime is an independent set of shared libraries that makes it possible to execute compiled MATLAB applications or components [22].

In contrast to this master project, it was only the interface part of the program that was developed, as there were no expansion of the functionalities to the model. The aim of the project being to increasing usability and user friendliness of using the model by building a graphical user interface.

To establish the usability and user friendliness of the program, Dewan had users to test the program by performing tasks with the graphical user interface. A group of 17 testers performed the two task on the program to see if the testers were able to complete the tasks by following some specified steps. Then the testers gave feedback on if the result was useful, and reported on errors or fails to complete the tasks. Finally the testers rated the easiness, usability and gave additional comments.

The situation with Covid-19 made it hard to have the same amount of testers in this master, so the usability testing in this master covered two testers. These was observed while performing several tasks on the program in addition to giving feedback by both ranking different aspects and commenting on the original and the further developed version.

3.2 CAATEX

The Coordinated Arctic Acoustic Thermometry Experiment (CAATEX) is an ongoing project in the Acoustics and Oceanography group at NERSC [23]. This process addresses research in the central Arctic Ocean, especially the ocean climate change. Although the Arctic heats up relatively rapidly, the temperature under the sea ice in the central Arctic Ocean is poorly observed and remains largely unknown. The CAATEX project focus on the central Arctic Ocean, where they collect new ocean observations and obtain new knowledge about the decadal changes in the heat content there [24]. They also work on improving their understanding of the uncertainties in the heat content estimates from climate models.

The main measurement principle in CAATEX is acoustic thermometry, where acoustic signals are used to determine the range-depth average temperature along source-receiver transects. A sound signal is transmitted from a source mooring on a precise time schedule. The sound propagates in the water column to the receiver, and is recorded on a series of hydrophones (i.e. underwater microphones). The travel time is then measured, and in turn used to compute the average speed of sound, which can be inverted to obtained the average temperature. While this technique is certainly more complex than outlined before, acoustic modelling is still a critical component in the design of such experiments, as well as processing, analysis and interpretation of the results.

Chapter 4

Methodology

In this chapter a introduction to the Design Science Research Methodology will be presented along with how it was used in this master. Also how the research data was collected will be described.

4.1 Design Science Research Methodology

Working with this master project many elements of the systematic approach given by the Design Science Research Method were applied. The 3-cycled structure of the method is shown in Figure 4.1. The main elements of the process; environment, design science research and knowledge base, are tied together by three iterative circles[25].

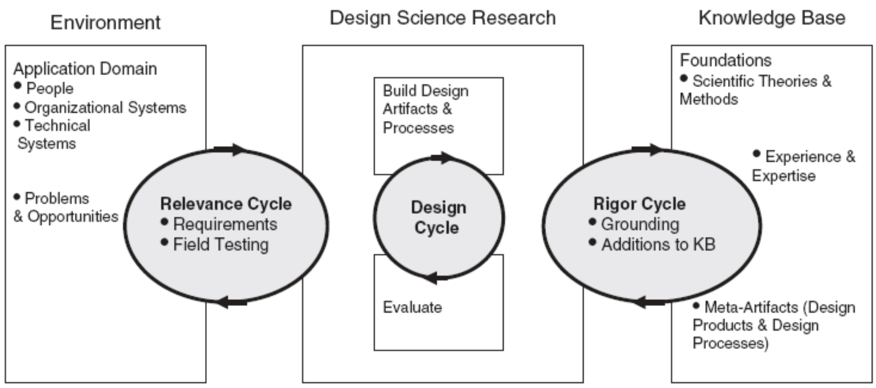


Figure 4.1: A Three Cycle View of Design Science Research [25]

The first cycle, named *Relevance Cycle*, connects the contextual environment(resources, problems and opportunities) of the research project with the design science research activities. The design science research is started with an application context that not only provides the input requirements for the research, but also defines acceptance criteria for the ultimate evaluation of the projects concluding delivery. This is followed in this master as the environment of NERSC both provides requirements and testing of the developed program.

The central *Design Cycle* iterates between building and evaluating the design artefacts and processes of the research. This is the main cycle which takes input coming from the environment and the knowledge base to developing and evaluating of the artifact. In this master project the development of the program have had input from both the environment of NERSC that have contributed with relevant information about systems and activity, and from general knowledge base of what makes a program better when it comes to usability and user friendliness.

The third, *Rigor Cycle*, links the design science activities with the knowledge base of scientific foundations, former experience and expertise contributing with information to the research project. Here in this project the developer's experience through her studies and the new knowledge gathered over the master project.

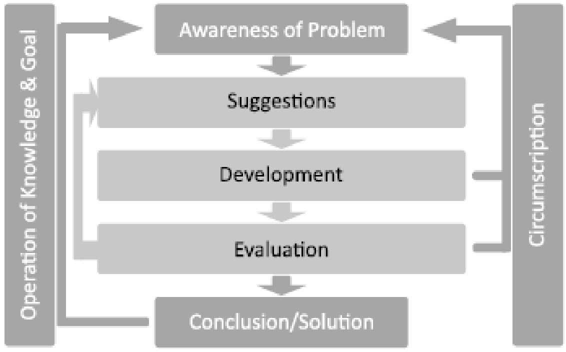


Figure 4.2: Reasoning on Design Cycle

The research in this master started by identifying and representing opportunities and problems in the actual application environment, which in Figure 4.2 is mentioned as the Awareness of Problem. The figure illustrate a kind of flow chart version of the *Design Cycle*. It is reasoning on the connections between the main elements in the designing process; Awareness of Problem, Suggestions, Development, Evaluation and Conclusion/Solution. Either it is case of designing a new program for new purposes, or revising/expanding an older program every cycle starts out with outlining the basic problems to be solved, and to become aware of operational problems and limits as seen at the actual stage/cycle of the design.

The Figure 4.2 shows how "awareness of problem" is the starting base for either initial or later revised cycle suggestions. The awareness will likely be due to iterative circumscription originating from problems realized during the Development and Evaluation steps. A conclusion or solution once found may be subject of needed revision due to new experience and knowledge, or more ambitious goals. And a new design process may be started with changed and extended awareness of problem.

The iterative cycle of Suggestion, Development and Evaluation will be performed frequently, in the search of finding more useful and more accurate solutions. At the development or evaluation level of the internal cycle a need for changes in Circumscriptions can arise, and take the process back to the position of Awareness of problem. In design research this is important as it outlines the importance of construction to gain understanding/knowledge. Development circumscription can be caused by limitations in the programming tool. While circumscription originating from the evaluation level can be that the outcome do not comply with the performing requirements and ambitions. Then there is conclusion/solution where the output of knowledge and the goal involves the *Relevance* and *Rigore cycle*. Operation of knowledge and goal can give back new problems and the cycle starts again.

Hevner *et al.* [26] have established seven guidelines to define the fundamental principles of design science research. In Table 4.1 these are presented with how they are defined in this master.

Table 4.1: Overview of general Design Guidelines with specific references to actual elements concerning in this thesis

Guidelines	In this Master Project
Design as an Artifact	Extended Arctic Package
Problem Relevance	Requirments from NERSC
Design Evaluation	Following foundations from knowledge base
Research Contributions	Tools: MATLAB Improvements: Usability & functionalities
Research Rigor	Following rules and disciplines
Design as a Search Process	Reasoning on Design cycle
Communication of Research	This Master Thesis

4.2 Research methods

Research data of this master was collected both from secondary by literature and primary with the usability testing. The literature about how to make programs more user friendly were gathered and used for evaluation in the process of the design cycle. The usability testing was performed on prototypes of the original and the new version of the program. The usability testing resulted in end users answers to several evaluation questions of the original and updated program.

There is a lot of literature about how a program becomes user friendly and which factors are important to consider when developing a program to be user friendly. Some of this literature are presented in Section 5.7, and are considered while developing the user interface of the program.

By final usability testing on prototypes both qualitative and quantitative methods were used to collect data for the research. The quantitative method of using surveys collected data of satisfaction ratings, rating of easiness to use, and users own evaluation of both the original and the further developed version. Participants performing different tasks on these prototypes were observed, and time lapses were noted. The task performing was observed/registered by remote screening which gathered qualitative data related to the participant chosen interface pathways and problems experienced. The testers conveyed comments and recommendations as written answers to open-ended questions or given orally to the observer.

Chapter 5

Design and Implementation

This chapter will give a description of both what are implemented and how the new structural design is formed. Starting with discussion the choice of programming language. Then listing up the main goals and then explaining them in more detail. Then presenting some usability design principles, the changes done to the graphical user interface. Then the new functional structure is presented, showing the main parts and changes. Some smaller additional functions are presented before some challenges in the project is described.

5.1 Programming language

The original package was made in MATLAB, and also a lot of the models that can be added have some codes already written in MATLAB. The expansion of functionalities will then be easier to do by continuing the development in MATLAB, as some parts are already made and just need to be connected to the existing version.

One of the limitations of MATLAB is graphical design, as the display of the user interface controls are depending of the operating system being used. This implies that the position and size of the interface control details/units will look different on a Mac compared to a Windows computer. So that limits the improvement of the user experience when it comes to what a better graphical design interface could give. Considering following the ”SOLWEIG a climate design tool” master project one could make the graphical user interface in another program and integrated the functions in MATLAB with the free MATLAB Compiler Runtime. But as the integration of new functions, models and environments are important, the time spent to code in different languages would be longer and would slow down the process of developing the program.

The decision to keep on using MATLAB was taken because this programming language has capacity to easily apply a various collection of functionalities. This property was given priority to optimizing processing time and more visually pleasing graphical presentation. The actual program will be used in natural scientific research, and a wider range of functionality will be more essential for that purpose. By changing the programming language the program could perhaps increase the running speed, and get a more fancy interface, but limitation in functionality could make it less beneficial for the researchers.

The potential of improving and adding functionalities in MATLAB is in this case regarded as most important and more relevant compared to other language alternatives. As indicated above MATLAB may have some limitations when it comes to graphical interface design, because the user interface controls are determined by the actual running operation system. So this characteristic put limits on the degree of improvements that can be implemented when it comes to better graphical design of the user interface.

5.2 Goals

Detecting the problems and making goals be in the first step of the design process - 'Awareness of the problem' -, as shown in Section 4.1. Together with the *Environment*, in this case NERSC, the possible extensions goals were settled by discussing what could be possible and would be of relevance for a researcher in the field of acoustic and oceanography. This resulted in the list below of elements that could improve the program.

- Save simulation results, as well as input data and other relevant metadata
- Add support for more acoustic models
- Improve the Graphical User Interface
- Simulate several time periods in one run, to so further compare the time-varying result
- Simulate in space, meaning running model prediction on several receivers surrounding the source in a circle.[1]
- Simulate several models for comparing
- Add more environment input

These goals were established, developed and accomplished in different times and design cycles, and were improved by going through the design research reasoning several times. Some were also done in several cycles as the development of other goals impacted what were needed to achieve the solutions. Examples of this is the improvement of GUI needed updates when simulations in time and space were added as it needed other inputs from the interface, and also the saving of simulations needed to get a updated set up for every new acoustic propagation model.

5.3 Save Model Simulation Result

The first goal was making it possible to save result with metadata after a run of a model on a path. This is important for 1) be able to reproduce runs, and 2) to further process the result. As today it needs to be run every time you want get calculations and visualizations. It it also critical for the goal of simulation in time and space.

As the acoustic models rely on different input parameters, and give out different result variables, every model needs its specific setup for what to be saved. A setup was first done for the two models that alreadxisted for the model. After adding new models, setups for these had to be made.

In the program all the models requires a sound speed profile, so this is saved for all runs. All the models will also have the information about the bathymetry, depth at the different range of the path chosen for the run. The models parameters, both input and output that are relevant, are placed in a 2-dimensional array after the modeling. The size of the array will depend on the amount of receivers and time steps, giving the array its size [*number of receivers* X *number of time steps*]. This array will be saved to a file when pushing the 'Save result'-button that is available on the visualization window that appear when running a model. All the general parameter that are not dependent of neither receiver or time steps are just saved in the first cell of the array, and not being repeated in the other cells as this would be the same and just use space if being saved in every cell. Also the parameter depending on either receiver or time steps will just be saved at only one position in the array.

[1] As the simulating in space was coded by the second student working on this program, Frida Tryggestad Klockmann [8], so this will not be deeply described in this thesis.

5.4 Adding Acoustic Models

There were added several acoustic models to the package, this was two versions of Range dependent Acoustic Model and Bellhop with several different simulation options. First among these is the RAM option that calculats the transmission loss along the path. The second is the RAM option that gives turning point filter [27] and time front. Finally, Bellhop with both have ray tracing and transmission loss calculations.

All the acoustic models needed a setup where there were done several steps to get the final visualisation of the modeling. When running a prediction the code file `plotZRT` first gets the run settings which are read from the GUI, then files with the path bathymetry or SSP are made. Then follows the data input files - depending on the model chosen - are written, and so the modelling that give out the new parameters that are plotted for visualization. These setups are coded in `runRAM`, `runMpiRAM` and `runBellhop`.

Both `runRAM` and `runBellhop` read the ssp and bathymetry variables from the `temp.ssp` and `temp.bth` files made before the running of the model. All the models also have parameters the user can set the value of in the user interface. All these model needs the user to choose frequency, both `runRAM` and `runBellhop` take also in the source's and receiver's depths. While `runMpiRAM` take in a Q-value (that determines the bandwidth of the signal) and the time window width from the parameter interface. `runRAM` will so write the input file ram.in before running the modeling that result in the tl.grid tl.line files that have the information about the transmission loss and are read to plot the visualization. `runBellhop` will write the input files `belltemp.ssp`, `belltemp.bty` and `belltemp.env` for the modelling to be able to produce the output files depending on the simulation type chosen, `belltemp.ray` for ray simulations and `belltemp.shd` for transmission loss simulations.

5.5 Adding Environmental Input

There exist several sets of different data from the Arctic Ocean, and NERSC would like to be able to look at more of these. The original program takes parameters from 3 databases, which are IBCAO, ECCOv4 and WOA that are presented in Chapter 2. In this master project, code was implemented for the program to get information from several other databases. This includes IBCAO with 30 arcsec resolution, WOA2018, GECCO and Ocean Model. The IBCAO with 30 arcsec resolution are used for a detailed visualisation of a map of the Arctic Ocean coloured by the the depth. While the other new data sets provides parameters used in the running of a model prediction.

The WOA2018 is just used for the sound speed profile (SSP) in the month of June, as this is the only data set from the WOA2018 version that is placed in the database. The setup makes the run take the WOA2018 data if it is in the database. So the remaining the periods data sets just need to be added to the database and named the same way as the SSP June file; "lev_jun2018", replacing 'june' with the time period it is and adding letter T, S or N after lev if it is for temperature (T), salinity (S) or buoyancy frequency (N).

There is also a setup for getting data from the GECCO that covers 10 years of observations, and the Ocean Model covering the Fram Strait with data of both temperature and salinity. Using these data sets as input it is possible to calculate the sound speed, and by the set up in the program it is possible to plot normalized SSP on a transect with these data sets.

5.6 Simulate of Several Time steps, Receivers or/and Models

To be able to have more simulations in one run, there were made loops for running the parts that were needed to give the multiple simulations result. If the run includes several models the loop will return for making the input files needed for the other model. When having chosen to run several time periods, the loop will go back to get the sound speed from data set containing the next time period. As the run of an other model will be on the same path and in the same time period, the loop for this will go back to the set up of files before modelling.

The oceans water masses are in constant motion, which again affects the sound propagation. With the new setup that loops through time steps simulations and saves parameters from a every simulation to an array, time steps can be run consecutively. The result will then be visualized and can be browsed through the different time periods. The WOA data sets have seasonal and monthly data, this means that it is possible to get a seasonal and monthly run with this choice. The saved data can also be processed and visualized in different ways, being able to compare the different time periods. This is done by looping through the runs of calculating the different time periods parameters and resulting in a visualization like other runs, but with the possibility to go trough the different seasons or month.

5.7 Usability Design Principles

An important part of improving a program to become more user friendly is to increase the usability of the user interface. Generating significant levels of usability are conditioned by five quality components that should be achieved within the interface; learnability, memorability, efficiency, errors prevention, satisfaction [28].

Learnability being how easy the program is to learn, is decided by if the users can move quickly from not knowing the system to doing any work. **Memorability** meaning how easy to remember how to use the program, allowing the users with low usage rates to return after a period of inactivity without having to retrain everything. **Efficiency** means that the expert user is allowed a high level of productivity. **Errors Prevention**, so users do not make many mistakes and that these errors are not catastrophic (and that one can easily recover). **Satisfaction**, comfortable to use, users satisfy subjectively, so they like to use the system.

Norman [29] introduced several basic user interface design principles and concepts that are now considered critical for understanding what increase the usability of a program by increasing the quality components mentioned before. These principles includes visibility, feedback, constrains, consistency, affordance and mapping. The first principle is **visibility**, which means that it should be clear for the user what every interface element are supposed to do. Looking back on the usability quality components, visibility can approve on all of them, as it has a lot to say for how the user understand how the program works. Then there is **feedback** that show or gives the user information or indications of what has been done. Third is **constraints** that means that the user is stopped from doing wrong choices, and then preventing errors. One other is **consistency**, both intern and external, means that there should exist similar operations and these should have similar elements that perform the similar assignment. Also a factor is **affordance**, which is a visual attribute of an object or a control that gives the user clues as to how the object or control can be used or operated. The final principle is **mapping**, that is the relation between the controls and the effects, making it clear what the controls functions are.

5.8 Graphical User Interface Design

To design a new and better graphical user interface the design principles presented in Section 5.7 have been considered. The principles address factors that are important for good designing of the interface, which will contribute to an optimal user experience of the system. The factors are visibility, feedback, mapping, constraints, consistency and affordance, which are presented in Section 5.7. Through the developing of the system these factors were considered when evaluating how to design the new interface.

The setup of the graphical user interface is changed in order to be able to perform the new functions, and making it easier to understand and use. To increase the usability of the program, Norman's design principles have considered in the changes of the GUI. The development of the GUI was an ongoing cycle where things had to be changed when new controls had to be added.

Changes were made to simplifying the GUI to make it more understandable. This included moving advanced and less used functions to its own window. Also parameters connected to the different models were placed its own window that appears when choosing a model on the main GUI.

All the controls connected to the main functions were surrounded by panels with titles about what the content inside were about. This was done for improving especially two of Normans Design principles: visibility and mapping.

The advanced functions with regards to treating sources and receivers are, as mentioned, located in its own window that appears when pushing the 'Advanced Src&Rcvr Edit'-button. Here are also assembled other controls in several sub-panels with explaining titles on how the functions work. The source editing controls are all placed on the left side and receiver editing controls are set up in the same way on the right side making it consistence, which is one of Normans Design principles. Adding source/receiver by coordinates are changed from having 3 inputs in one edit-box to one edit-box for each input. This have an impact on the preventing of errors and making it easier for user to give the input the right way.

When it comes to Normans Design principle about constraints, which are mainly for the error prevention factor of usability, this is also integrated in the new GUI. This is done by enabling choices only when and if the settings are relevant to the set up. Examples on this is the new depth choice connected to the making of map for which parameters of temperature or salinity should be visualized if these parameters are chosen. This popup menu is not active when bathymetry or other data sets that is irrelevant or have no alternative choices of depth, as for colouring the map presentation. An other example of error prevention is the time steps option that are just a possibility for WOA data set, as the other sets would just have given the same variables for any time period chosen. So by constraining the user from choosing this for a data set that does not have it, make the user not misunderstand what period is run. And also preventing a run with four or twelve simulations of the same time period.

5.9 New Structure

The main functional structure of the new version are illustrated in Figure 5.1. The background colour, green or red, of elements indicates the changes in the program. Green boxes are new structure elements that represents added functionality or elements supporting them. The parts coloured red are modified elements. Blue representing the databases used by the program, addition to these databases are written in red text. Orange- and yellow-coloured elements are parts that are remained unchanged from the previous structure shown in Figure 2.3.

The main GUI is red for the reason of having been updated with a simpler setup, which is further explained in Section 5.8.

The setting of map can now be coloured by parameters for several new data sets; a higher resolution of bathymetry, and temperature or salinity parameters from the Ocean Model. The set up for getting sources or receivers from file is the same as before.

Several steps in the get prediction are red and changed in different ways. The **Acoustic model parameters** are decided after choosing to get prediction, and here are new model possibilities added. The step of **getting sound speed from database** is updated to be able take sound speed profile (SSP) from the 2018 version of WOA, if available, and calculate SSP from temperature and salinity data from GECCO or Ocean Model data sets. Continuing with the steps the of **setting up input files** and the **modelling** that are done before the result is visualized (**Visualize model prediction**), that are now possible for new models. As it is now possible to run simulations of several time steps, receivers or models different steps can be looped. The three loops have different starting points depending on what new information that needs to be gathered for the new simulations. New models just need the **set up of input files** and **modeling**, while time steps also need **get** new **sound speed data**, and new receivers need in addition **getting the bathymetry** of the new path.

It is also added the function of saving the prediction result (**Save result**) for later visualization and further processing. The **Save result** function will place the array with the result in a new database (**saved model predictions**). The result are saved and can be visualized the same way or other visualizations depending on the model, and if it has several receivers and/or time steps. A run of any model will temporarily save the needed parameter in every loop, this makes it possible to save this to a file if the result is of interest to save.

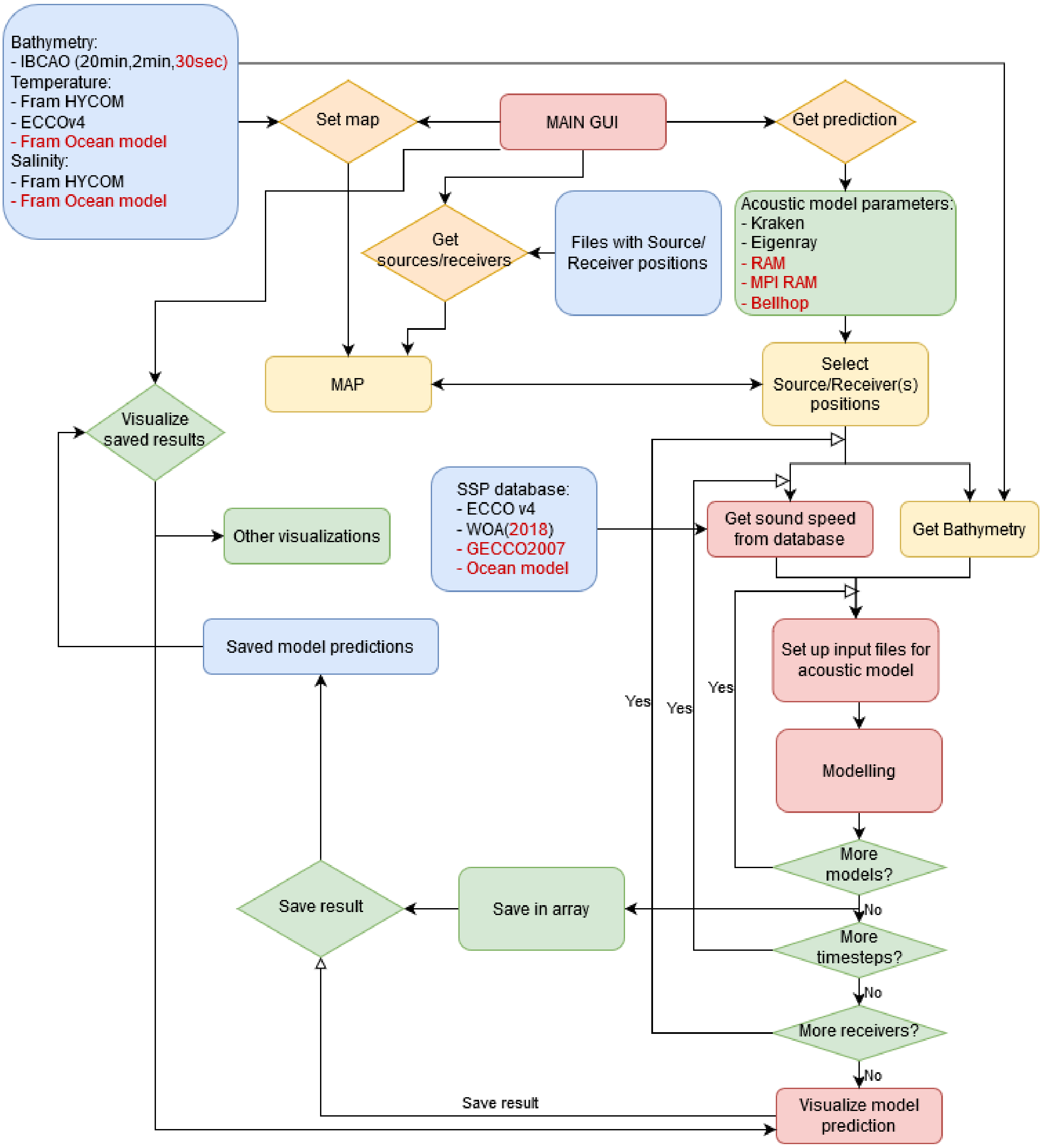


Figure 5.1: Block diagram illustrating the structure of the new package

5.10 Smaller additional functions

There are also made some smaller changes and added some functions to improve the user experience of the program. These are presented in the list below and followed by the reasons for having them and the properties added by using them.

1. Visualize the path of the model prediction on a map
2. Saving new sources and receiver positions to file with the graphical user interface
3. More error prevention when add source or receivers with coordinates
4. Plotting of possible path can be viewed and removed from the map, and are updated automatically when adding, moving or removing sources or receivers
5. Plotting of possible paths will just plot the paths and not plot the map colouring again

As the visualizations of previous run models can be opened, the need of being able to visualize the path the model is run on, is more important. The standard result visualization just give the source and receivers positions with the longitude and latitude values, which can be hard to interpret where is. In the new version it is added a button that makes it possible to get a map with the path plotted on.

If new interesting sources and receivers should be saved to a file for later use this is now possible with the program in the advanced setting of sources and receivers. The sources and receivers need to be added to the map, and by putting in the filename and then pushing the 'Save sources/receivers'-buttons.

Before the adding of source or receiver by coordinates needed the user to put in three values in one edit box, which make it easy for the user to get an error if the values are not put in the right way. To prevent these errors this was changed to be three edit-boxes with clear labels of where the longitude, latitude and depth value should be put in.

The 'Plot paths'-button is updated to also make the possible path be removed from the map after plotting it. Also if the possible paths are plotted, the action of adding, moving or removing a source or a receiver will make the possible paths be updated automatically to the possible paths of the new set of sources and receivers.

The plot paths functions have been change to just make the paths appear on the existing map and not loading the colouring of the map again. This makes the function more efficient as it do not do unnecessary work.

5.11 Challenges

Along the way of this project process, there were some challenges. First the compiling of the models that were supported gave some complications as it did not work correctly within the windows terminal. The solution was to use CYGWIN. This is a programming and runtime environment that allows the user to run programs designed for UNIX systems [30]. In this case, the FORTRAN compiler GFortran was installed and used to compile MPIRAM and Eigenray. It is also used to run these programs from within MATLAB.

Applying the data sets GECCO and Ocean Model to work on models run gave also problems as it had "NaN"-values that made the model runs fail. There were not enough time to look into this in order to get it work correctly.

The GECCO data can be directly loaded from a thread server at NERSC. As the (Covid-19) circumstances did not allowe us to stay longer at NERSC for testing and developing the support of getting it from this server, this was not developed. The provisional solution was to make a setup for reading an example of a GECCO data set that was saved in the database of the package.

Chapter 6

Results

There have been added new functions to the program to make it more useful for researchers at NERSC and to hopefully be understandable for students in the field of acoustics and oceanography. To accomplish usefulness the work has been closely followed up to what are relevant by working with researchers. The functions have been added with the aim of improving the program to make it easier to do research. In this chapter the result of the extended program will be presented.

Table 6.1 gives an overview of functions and the accompanying data sets that can be used by the functions in the original package, and the additions that have been made in the updated package. Several data sets are added and can be used in the new program for different functions, in addition to what were supported in the original package. The table takes up the three main functions that already existed: colouring map of Arctic Ocean by data set parameters, plotting of normalized profiles of sound speed, temperature, salinity or buoyancy frequency on 5 different places of a path and the acoustic model prediction run to visualizations.

Table 6.1: Overview of the available data sets for functions in the original package and the updated package used for main functions.

Functions	Database	Original package	Additions
Map	IBCAO	Resolution: 20 arcmin, 2 arcmin	Resolution: 30 arcsec
	HYCOM	Temperature, Salinity	
	ECCOv4	Temperature (5 m, 95 m, 477 m)	Temperature (50 m depths)
	Ocean Model	-	Temperature, Salinity (50 depths)
Normalized Profile Plot	ECCOv4	Sound Speed, Temperature Salinity, Buoyancy frequency	
	WOA	Sound Speed, Temperature Salinity, Buoyancy frequency	
	GECCO	-	Sound Speed[2], Temperature, Salinity
	Ocean Model	-	Sound Speed[2], Temperature, Salinity
Acoustic Models	ECCOv4	Eigenray, KRAKEN	RAM, MPIRAM, Bellhop
	WOA	Eigenray, KRAKEN	RAM, MPIRAM, Bellhop

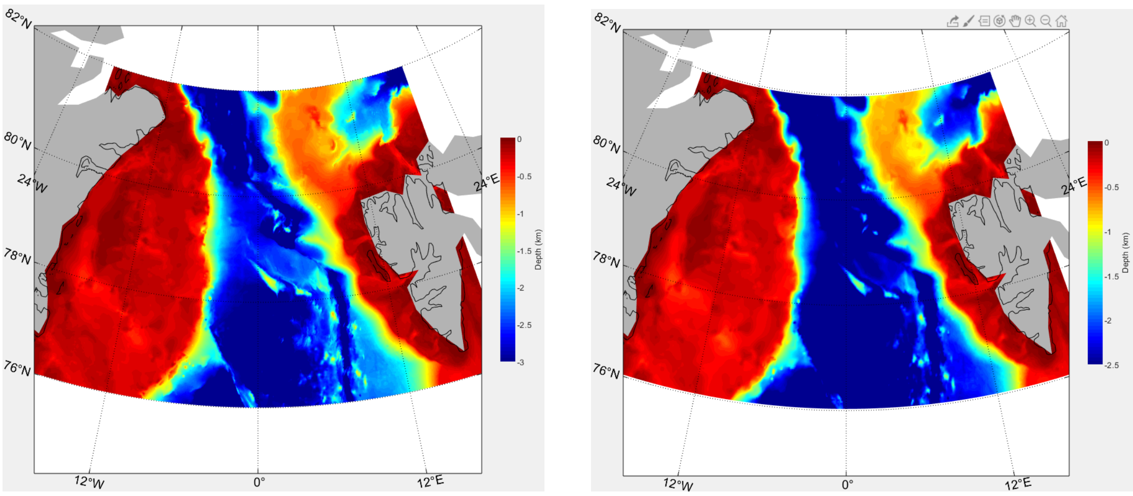
[2]Sound speed is calculated with the variables of temperature and salinity

6.1 Visualizations

The program is able to visualize a lot of the data it is given. Like the function of colouring map that shows the bathymetry, temperature or salinity. Then it is the visualization of the normalized sound speed profile at 5 places along the transect chosen. And finally the running of acoustic models that calculate and visualize results in different ways.

6.1.1 Map

The first task in the program is making a map, thereafter to get sources and receivers to choose the path with when running the model prediction. The map making makes it possible to visualize the arctic ocean with colouring by 3 different parameters: 1) bathymetry; 2) temperature; and 3) salinity. The color on the map can be matched with the colourbar on the adjacent side that tells which levels of the parameter the different colours mean. There has been added the function to visualize a bathymetry data set of higher resolution as the map colouring. Figure 6.1 shows an illustration of the highest available resolution of the bathymetry in the old and new version at the Fram Strait, respectively. On the left is the highest resolution with 30 arc seconds which is equivalent to 1/120 degrees that is added to the new version. The highest choosable resolution in the original program; with 2 arc min that are equivalent to 1/30 degrees, which remains as one of several an option in the new version is shown on the right.

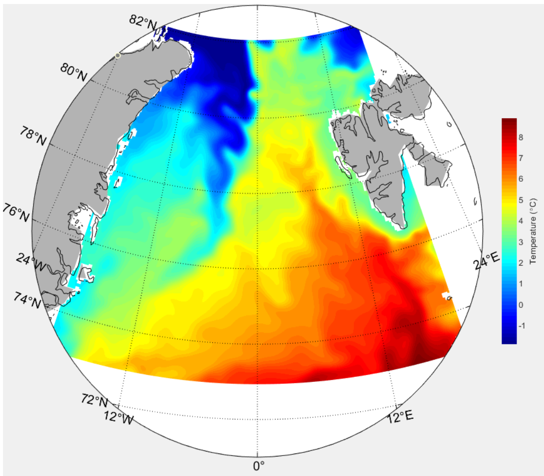


(a) (b)

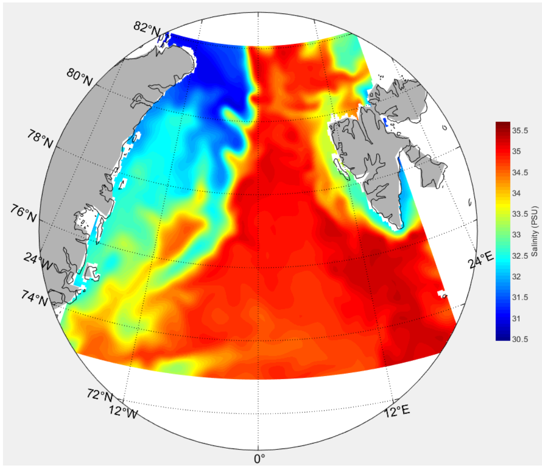
Figure 6.1: Bathymetry in the Fram Strait obtained from IBCAO with 30 arc sec resolution (a) and 2 arc min resolution (b).

ECCOv4 was also a map option for temperature colouring at the 3 depths: 5 m, 95 m and 477 m. Now this is expanded to 50 available depths that have the temperature data. In addition to the existing maps it is added a new option that takes data from a new database, the Ocean Model. This have both data for the temperature and the salinity at 50 depths. Like the ECCOv4 it is possible to choose which depth of the 50 depths that should colour the map. In Figure 6.2 the Fram Strait is shown with temperature (left) and salinity (right) at 5 m under the water surface from the Ocean Model data set that is added in the new package.

Table 6.2 gives an overview of the parameters and data sets the map can be coloured by in the original and the updated package. It shows which data sets have the different parameters that can be used to colour the map.



(a)



(b)

Figure 6.2: Ocean Model maps of temperature (a) and salinity (b) at 5 m depth in the Fram Strait region between Svalbard and East Greenland.

Table 6.2: Overview of the available data sets in the original package and the updated package used for making maps.

Database for Map	Original package	Additions
Bathymetry	IBCAO: Resolution: 20 min, 2 min	IBCAO: Resolution: 30 sec
Temperature	HYCOM Fram Strait(300 m) ECCOv4 (5 m, 95 m, 477 m)	ECCOv4 (50 depths) Ocean Model Fram Strait (50 depths)
Salinity	HYCOM Fram Strait (300 m)	Ocean Model Fram Strait (50 depths)

6.1.2 Output from acoustic models

In addition to the models supported in the original package, shown in Figure 2.5 and 2.6, there are added support for two types of RAM model, one that gives visualisation of transmission loss along path and one that gives visualization of the turning point filter and time front. Figure 6.3 and 6.4 shows examples of output visualisations of the two types of RAM model predictions that are supported in the new version.

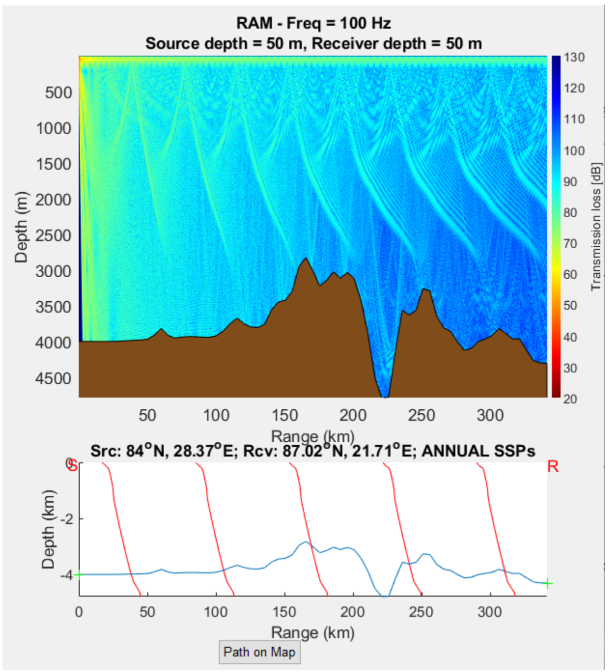


Figure 6.3: Upper panel: Transmission loss at 100 Hz as a function of range and depth, calculated with RAM. Lower panel: Bathymetry (blue line), and normalized sound speed profiles along the transect (red lines).

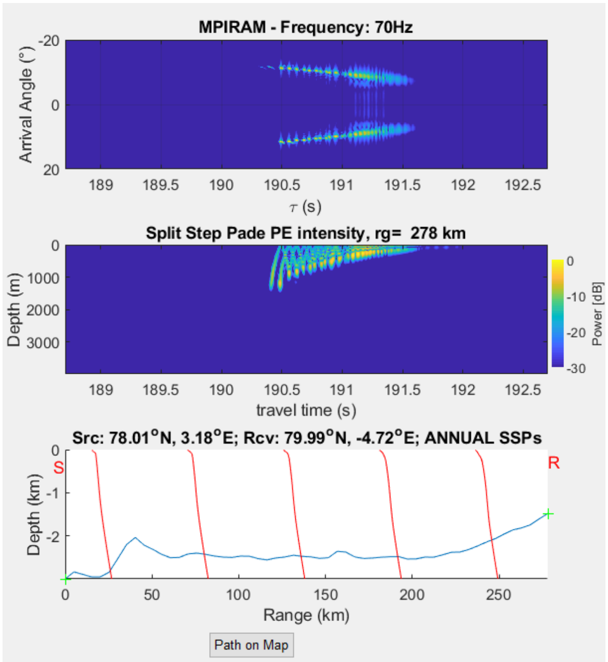
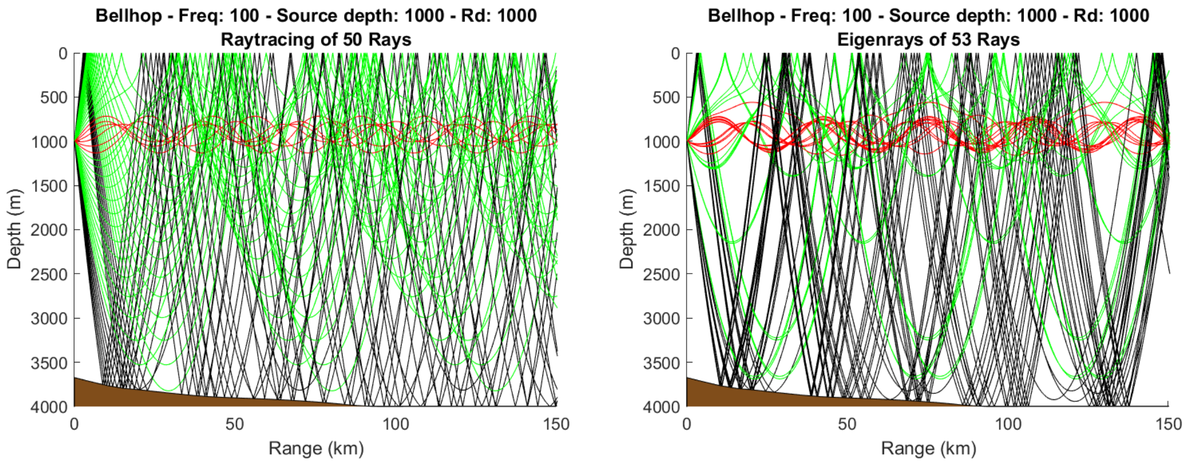


Figure 6.4: Upper panel: Turning-point filter at 70 Hz as a function of range and depth, calculated with RAM. Center panel: Time front at 70 Hz as a function of range and depth, calculated with RAM. Lower panel: Bathymetry (blue line), and normalized sound speed profiles along the transect (red lines).

Depending on choice of simulation type a running of the Bellhop model can lead to one of several alternative visualizations. The Raytracing and Eigenray visualize the calculations of the rays or eigenrays paths between source and receiver, as shown in Figure 6.5. Bellhop have a colour code for indicating if the ray hits any surfaces along the path. Red means that the ray have not hits neither the water surface or bottom, while green means it hits the water surface, blue would be if the path just hit the bottom, while black means it hits both bottom and water surface on its path. The Bellhop model also have different types of transmission loss, shown in Figure 6.6, based on different levels of coherence: incoherent, semi-coherent and full coherence. There are five different types of simulations, these can be run separately or combined, where one is a ray simulation and the other is a transmission loss simulation.



(a) (b)

Figure 6.5: Examples of Bellhop result when running the two types of ray tracing: paths of rays (a) and paths of eigenrays (b).

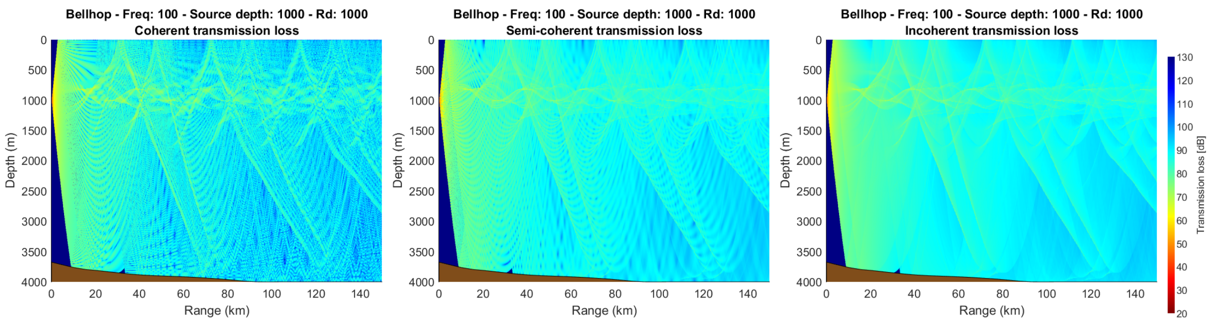


Figure 6.6: Visualisation examples of the different transmission loss run of Bellhop, with full coherence (left), semi-coherent (center) and incoherent (right)

Figure 6.7 illustrate the different visualisation option when a combination of a ray and a transmission simulation in Bellhop is done. Change of the visualisations options can be done by pushing a button. On the left the simulations are in different subplot and shown on its own, in the center the simulations result are overlapping in the same panel, while the last show only the transmission loss without the raytracing plot.

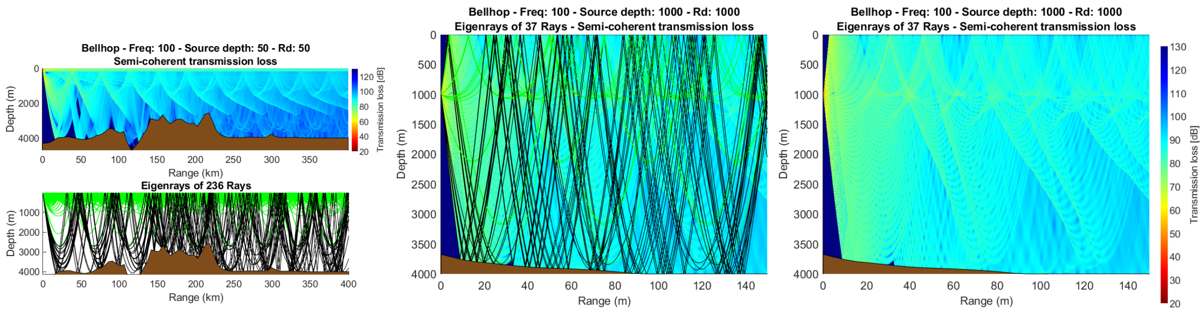


Figure 6.7: Example of visualizations of Bellhop result with eigenray and semi-coherent transmission loss simulations.

6.1.3 Comparison of results from different acoustic models

In addition to running one model calculations at the time, it is possible to run several model calculations and get a overlapping visualisation for comparing. This is possible for the different types of Eigenray and RAM. One of the options are RAM and the Eigenray's raytracing, where the RAM that shows the transmission loss on the path, while the Eigenray gives the ray paths. The ray paths can easily be removed, viewed or changed colouration for comparing the models by buttons or popup menus. Figure 6.8 shows some of the different visualization options when having run the models together.

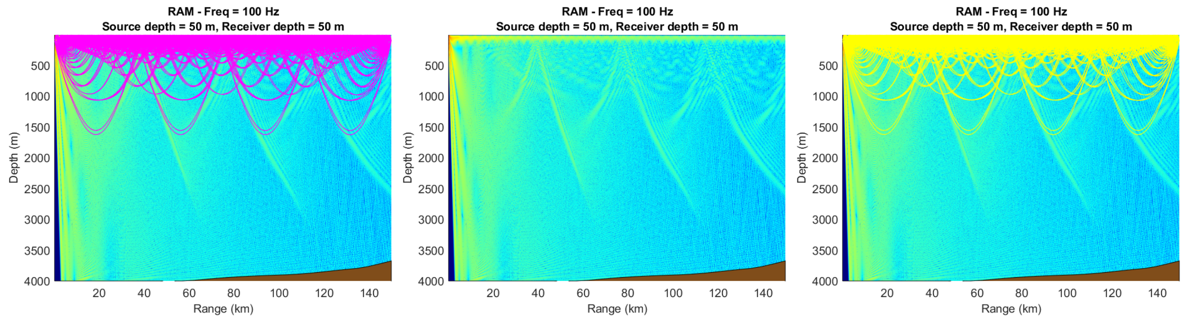


Figure 6.8: Different visualization possibilities for RAM and Eigenray. Transmission loss from RAM with eigenray paths in magenta (left), no eigenray paths (center), and eigenray paths in yellow (right).

The MPIRAM can be run with the Timefront of the Eigenray model. Like the RAM&ER option, the Eigenray plots are on top of the MPIRAM results, and the Eigenray models plots can be changed colour of and removed. Two subplots will be made from the model run, upper have a Turning Point filter that shows the travel time and arrival angle, then the RAM calculation of split step Padé PE intensity with the Eigenray Timefront of several rays. Similar to the RAM&ER visualisation the coloured can easily be changed by a popup menu and the Time front run plots can be change to being visible or not with the help of a button. These properties are shown in the Figure 6.9.

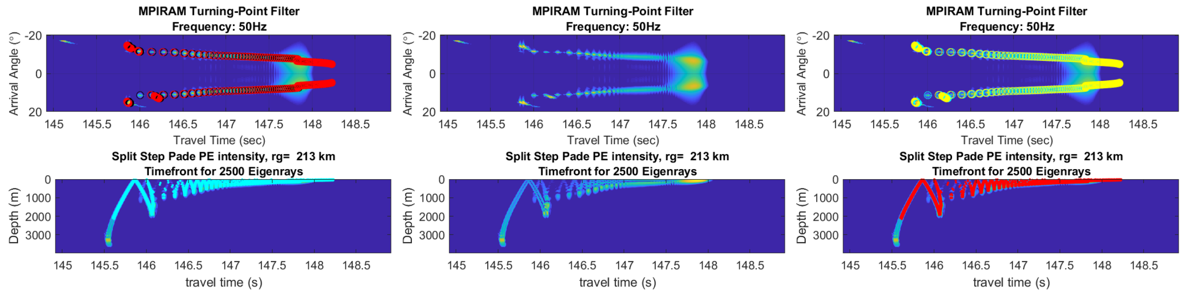
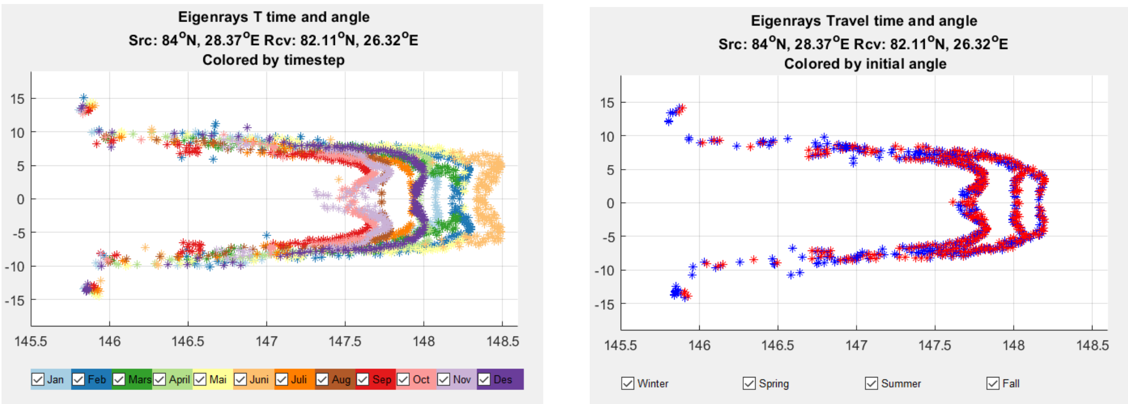


Figure 6.9: Different visualization possibilities for MPIRAM and Timefront Eigenray. Turning filter of MPIRAM and eigenray in black and red (left), no eigenray (center), and eigenray cyan and yellow (right).

6.1.4 Time variation - Example with World Ocean Atlas data

As the sound propagation is affected by the constant movement and the temperature of the water, the propagation will vary with time. By visualizing different time periods of the year, these changes can be studied. An update in the new version makes it possible to run seasonally or monthly time steps instead of just the average of a year, or a specific season or month.

The runs with several time steps can be saved and be visualized in other ways for better comparison. Here there are added two visualizations that compare the result of Eigenray at different time periods. Figure 6.10 shows one of these, the plots show the travel time and arrival angle of the eigenrays for every time step. On the left hand side it shows an example where each month is indicated by its own colour, while on the right is another example with seasonal time steps, which is coloured depending on the initial angle from source. The check boxes make it possible to choose which time period results should be visualized in the diagram.



(a) (b)

Figure 6.10: Comparison of the arrival angle (y-axis) and travel time (x-axis) of eigenrays for monthly data from WOA (a), and seasonal data (b).

The other time-comparison visualization is illustrated in Figure 6.11, it shows an example of a visualization that compares the travel time of the eigenrays in the different months. This was made to clearly show the variation of travel time of eigenrays at different time periods: months or seasons.

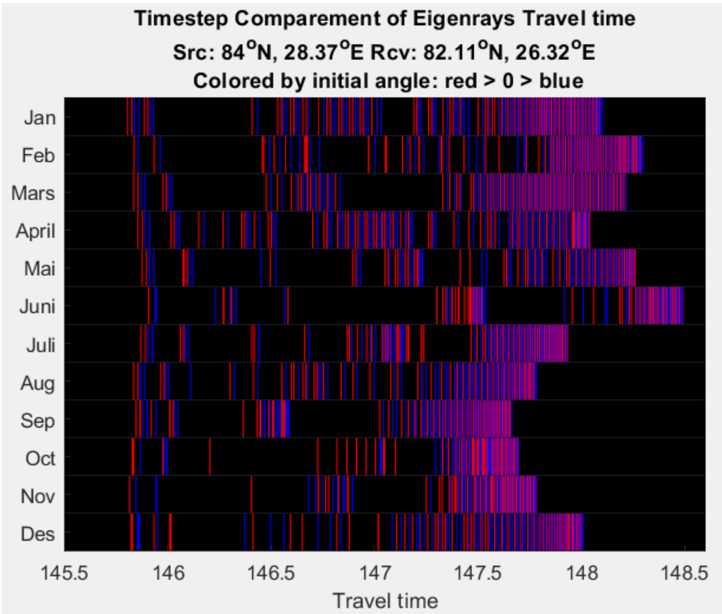


Figure 6.11: Comparison of the arrival angle (color: red positive, blue negative) and arrival time (x-axis) of eigenrays for montly data from WOA (y-axis).

6.2 Graphical user interface

The graphical user interface was modified considering the design principles for better user experience, and to comply with the additional introduced functions. The revised design is shown in Figure 6.12. There are less controls on the new main GUI, as advanced functions related to the source and receivers have its own sub menu, Figure 6.13.

Redesigning the main GUI aimed to separate parts that concerns different task and gather others which are linked to the different functions with panels.

The two first rows of the original GUI are for setting up the map, here there are minor changes. The popup menu with the options 'rec on' and 'rec off' has got an overlying heading 'Shape', and the two alternative options are renamed as 'Rectangle' and 'Circle'. In addition to all the controls that originally were in this section, a new popup menu was added for selecting the depth at which the value based coloured data are referred to. This is an attribute for map colouring option of Temperature and Salinity by the data sets ECCOv4 or Ocean Model, which has data values at 50 predefined depths. The 'Enter' button is renamed to the more specific "Set map". It is also moved to a concluding position following all the specifying parameters that impacts the making of map.

Further down on the control panel there are a setup for most frequently used functions related to sources and receivers; 1) getting and plotting the position of the moorings with sources or receivers that are saved in files and 2) clearing all the already chosen position plots entered in the actual map, in order to restart selection.

The option for manually alterations, as advanced adding and editing of the sources and receivers, are accessed by a menu that appears by pushing the 'Advanced Src&Rcvr Edit'-button

Below the panels for respective source and receiver it is a row of various buttons. The first is the 'Advanced Src&Rcvr Edit'-button which access a window with the options for manually alterations, as advanced adding and editing of the sources and receivers , shown in Figure 6.13. Next is the 'Plot Paths'-button that still sets the possible paths between the plotted sources and receivers on the map. But when this is done the text change to "Remove paths", as all the paths can be cleared from the map with a second click. The third is a new button for getting access to and visualize previously saved runs, according by the added possibility to save and further process a calculation in the new version of the program.

The bottom panel with two rows contains the run settings that the run of any model will be dependent of. Here some properties remain, even if some of them were initially assigned to be Eigenray parameters in the original version as they were placed under the 'Eigenray Parameters'-text. But as all model runs are using these inputs, they are kept on the main interface.

Figure 6.13 shows how the advanced node edit controls are in their separate menu in the new version. These were placed in line 5 to 8 in the earlier main GUI shown in Figure 2.1. In the new menu the left side is for the source editing, and on the right side are the controls for editing receivers. Both sources and receivers have the similar edit options positioned on the same rows.

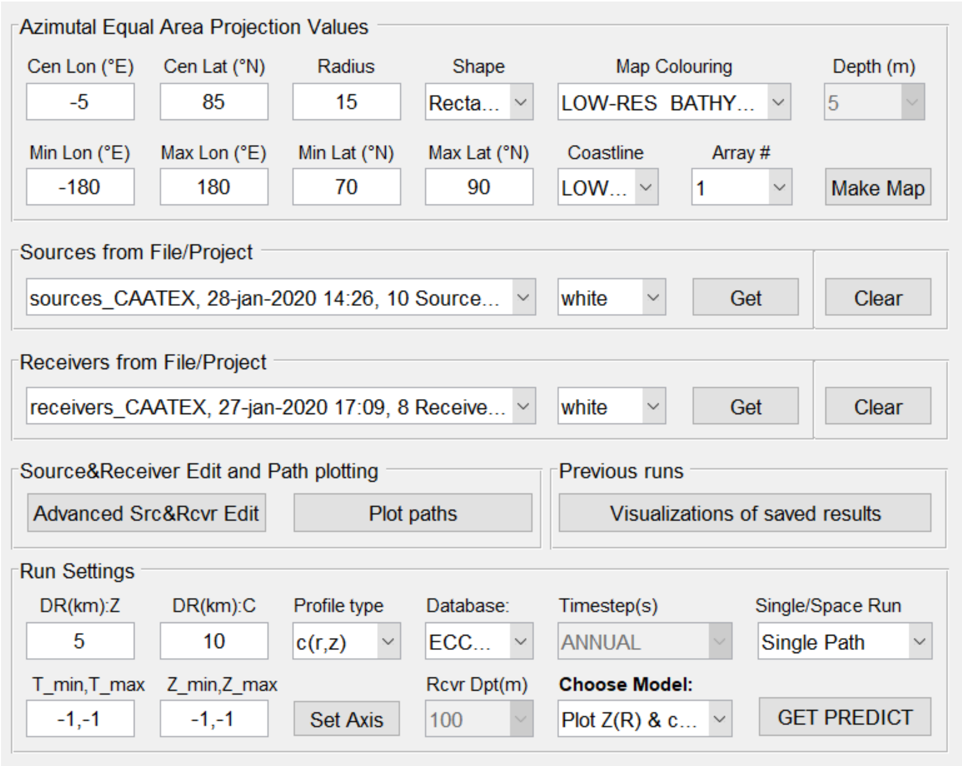


Figure 6.12: GUI of the new package.

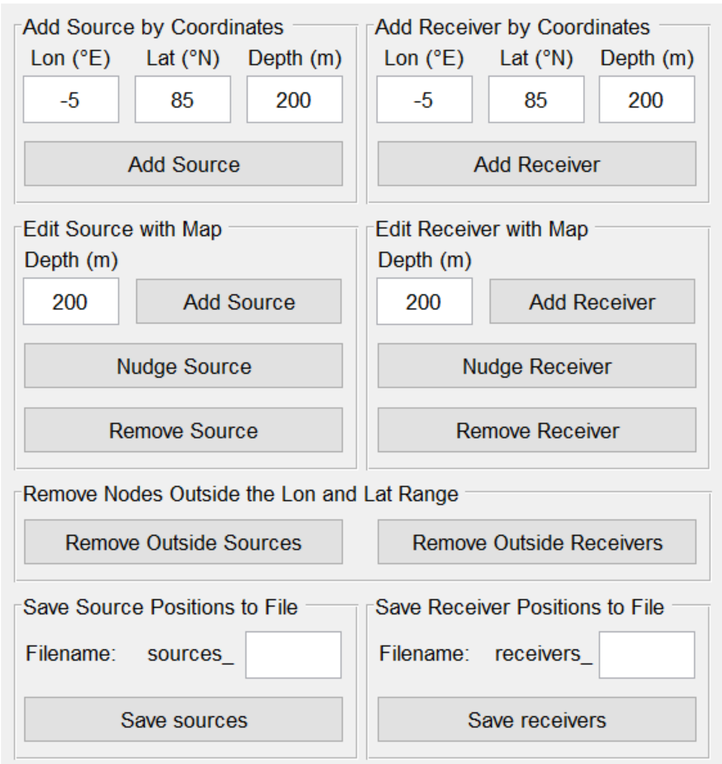


Figure 6.13: Advanced edit menu of sources and receiver.

Chapter 7

Usability Test

This chapter will present how an usability test were carried out on prototypes of both the original and the updated version of the program further developed in this master project and the result of this. This was done to be able to answer the second researching question (Q2) regarding the improvement of user friendliness and usability.

7.1 Tasks

In order to check to what extent the intended improvements of the GUI were gained, a set of test exercises were given to some test-pilots. They were asked to give a comparing evaluation of the user friendliness for the running set-up of the original and the new version of the program. There were build some prototypes of the original and the new further developed version that were possible to send to testers without needing to compile and install the acoustic models. Though, the tester needed MATLAB to run the prototype of the Arctic Package, like you would for the full program. The functions of running a model were replaced with a showing of previous saved runs of the chosen model so the tester still got the illusion of running the model prediction and got the visualisation that a model run would give.

There were done some usability testing on these prototypes. The participants got tasks that are possible to do with the versions and gave feedback on the experience of using the versions to perform the tasks. The aim of the test was to compare the usability and user experience of the original and new version and determent if the new was an improvement considering usability.

Before the testing, the participants had to answer some questions related to their background. This covered the field of their research, their use of software programs in their research, if they have used the original program before, and what operate system they used for the testing.

The test set contained 8 tasks; first 3 tasks using the original version, then similar 3 tasks with the new version, and finally 2 tasks for testing new functions incorporated in the new program.

The three task done on the original version:

1. Make map with the specified settings
2. Get sources and receivers from a specific file to be plotted on map and plot the paths between them
3. Run a KRAKEN model prediction with specified parameters.

Then the next five task done on the new version:

4. Make map with the specified settings

5. Get sources and receivers from a specific file to be plotted on map and plot the paths between them
6. Run a KRAKEN model prediction with specified parameters.
7. Run a Bellhop model prediction with both seasonal time steps and 3 receivers and other specified parameters.
8. Get several specified visualizations of a specific previously saved run of Eigenray with time steps

The three first tasks covering the main functions of the original package, that were performed by the participants, with following questions about the user's experience of performing the task. After these tasks the participants gave more feedback on the general experience of this version. The following tasks were on the prototype of the new version with following questions about the user experience. Three similar task, to what were done one the original, were done on the new version. Then there were two more tasks covering new functions of the new version of the program. After these the tester gave feedback on user's experience of this version.

The testing was observed with remote screening of the testers screen while performing the tasks, help were given if the tester did not understand how to accomplish some of the specifications of the task.

The first task cover the making a map appear on Figure 2, the second window that appear when opening the program in addition to the GUI. The map plot is depended on the several parameters concerning the coordinates of the center, and the radius from this point the map should cover, which parameters in which data set should the map be coloured by, and area chosen by min and max latitude and longitude the that the colouring should cover. The task had specifications so the tester needed to change some of these parameters before making the map appear in the other window. A similar task was done on the new version on Task 4.

The second task was to plot some sources and receivers positions that are saved in files and plot the paths between them. The tester need to find the files and choose the colour of the plotting of the sources and receivers, then plot them before plotting the paths between them. The tester was given specifications on which file and colours to choose. Task 5 was a similar task done on the new version of the program.

The third and last task on the original version covers the running of a model prediction. Some specific parameters need to be changed before choosing the source and receiver the model should run on the path between. This is also tested on the new version in task 6.

The two last task covers some of the new functions added in the new version. Task 7 is an running prediction of a new model with both several time periods and receivers. This levels up the previous task by having to change more parameters and understanding other aspects of the interface.

The final task in the testing covers visualizing previously saved runs. This covers not only the standard plotting that one gets after a model prediction run, but also other visualisations that compares time periods or show different aspects of the run.

7.2 Task Experience Questions and Answers

The usability testing resulted in both some qualitative and quantitative data. Quantitative data were gathered with forms to record data of satisfaction ratings, and observations of task time. Since the task performing observed with remote screening there were also gathered qualitative data related to observations about the pathways participants took and problems experienced. Also comments and recommendations and answers to open-ended questions were given either in writing on the sheet or given orally to the observer.

There were two participants, both researcher, performing the usability testing. Participant 1 is a researcher in the field of oceanography and acoustics, and do both programming and data analysing in MATLAB. This participant has used the original program before, a good while back. Participant 1 performed the testing on a computer with operating system Microsoft Windows. Participant 2 research in the field of physical oceanography and use MATLAB to analyse output from global climate models. This participant has never used the program before and performed the testing on the operating system MacOS.

After every task the participant was asked to rate the satisfaction of using the version to perform the task and the easiness of performing it with the version. The following user questions were asked:

UQ1. How will you rate the satisfaction of using the program to complete this task?

UQ2. How easy was it to complete this task in this version?

UQ3. Did you have any difficulties to complete the task?

User question one and two were answered by ranking it on a scale from 1 ”Very unsatisfying/difficult” to 7 ”Very satisfying/easy”. The third question have the answers options ”No, had no problem completing the task” (N) and ”Yes, but was able to complete the task with some help” (Y). In Figures 7.1 and 7.2 the participants answers to the 3 questions after every task are shown, where task 1-3 are on the original and 4-8 are on the new version.

Table 7.1: Results from the usability test - Participant 1.

Task	Satisfaction (UQ1)	Easiness (UQ2)	Need of help (UQ3)
1	2	1	Y
2	4	5	N
3	2	2	Y
4	6	7	N
5	4	7	N
6	5	6	N
7	5	5	Y
8	7	7	N

Table 7.2: Results from the usability test - Participant 2.

Task	Satisfaction (UQ1)	Easiness (UQ2)	Need of help (UQ3)
1	3	3	Y
2	3	3	Y
3	3	3	Y
4	6	6	N
5	7	7	N
6	6	6	N
7	6	6	N
8	7	7	N

Table 7.1 show that the first tester needed help for two of the three task on the original and one of the five on the original. While Table 7.2 shows that the second tester just needed help on all the tasks on the original version.

The diagrams in Figure 7.1 illustrate the participants rating of satisfactions and easy to use of the program on the different tasks. The task 1-3 are done on the original, while the remaining were done on the new version. The diagrams clearly shows the increase of scoring when the participants use the new version.

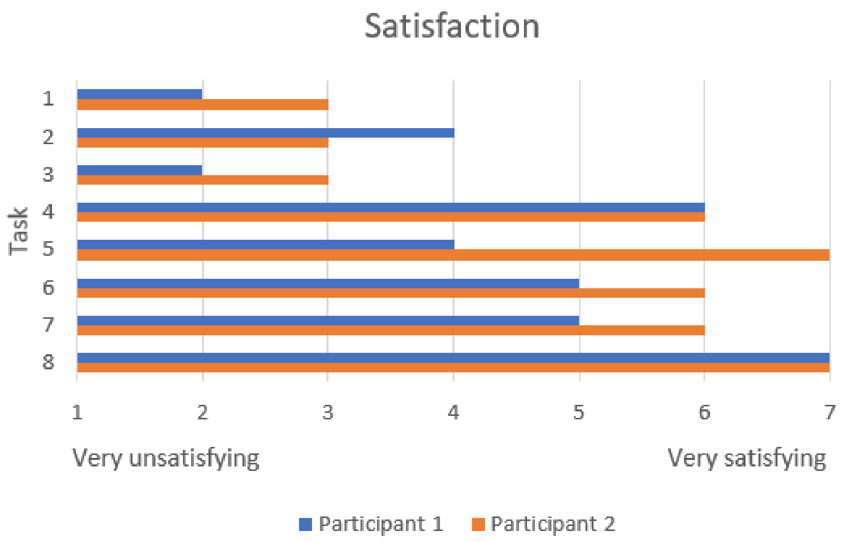


Figure 7.1: Participants answers to UQ1.

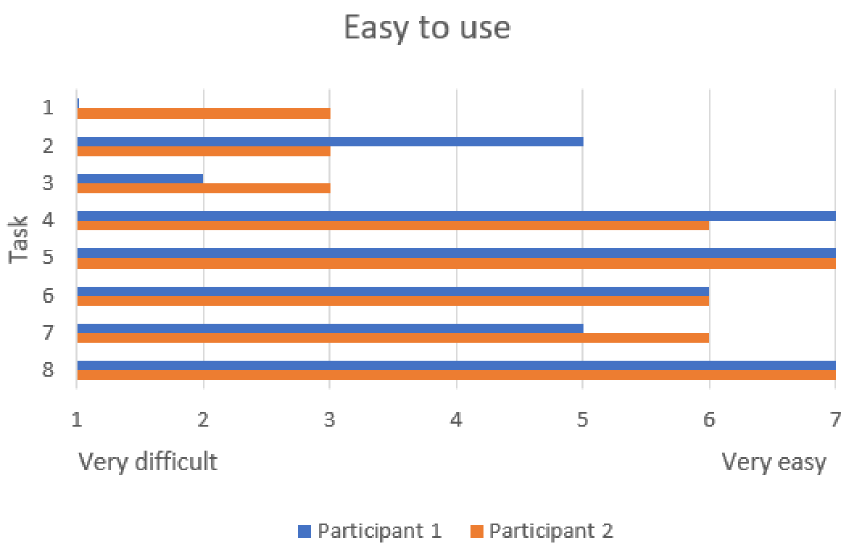


Figure 7.2: Participants answers to UQ2.

Figure 7.3 shows diagram that visualize the difference of the scoring of satisfaction on the similar task on the 2 versions. While the Figure 7.3 visualize the scoring of the how easy the two versions were to use for the similar task performed. The light blue bars represent the difference of the score by participant 1 while the yellow is the difference by participant 2. Except for the participant 2 satisfaction of the 'Get Src&Rcvr' task, the difference were at least 3, in the favour of the new version having a higher score. This means the scores increased mostly with at least half of the possible score range (7-1).

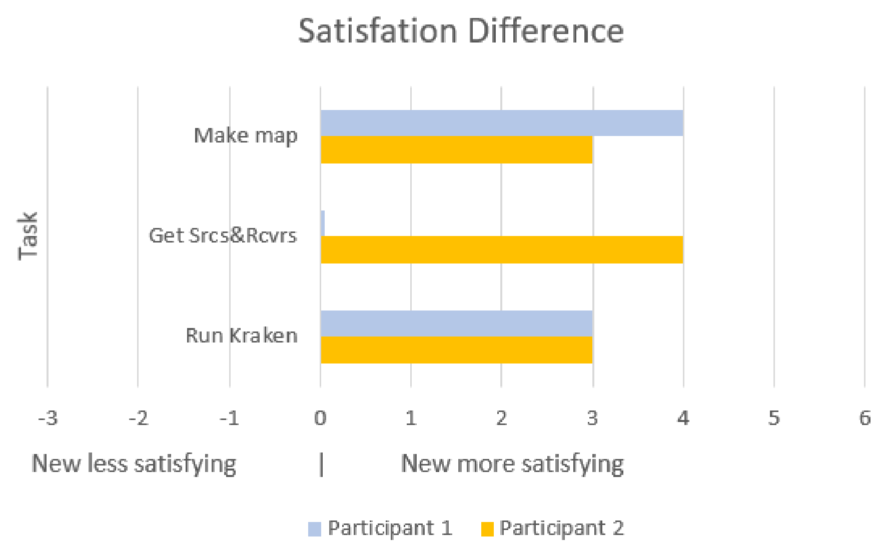


Figure 7.3: Differences on 'Satisfaction' score on original and new version (new version score - old version score)

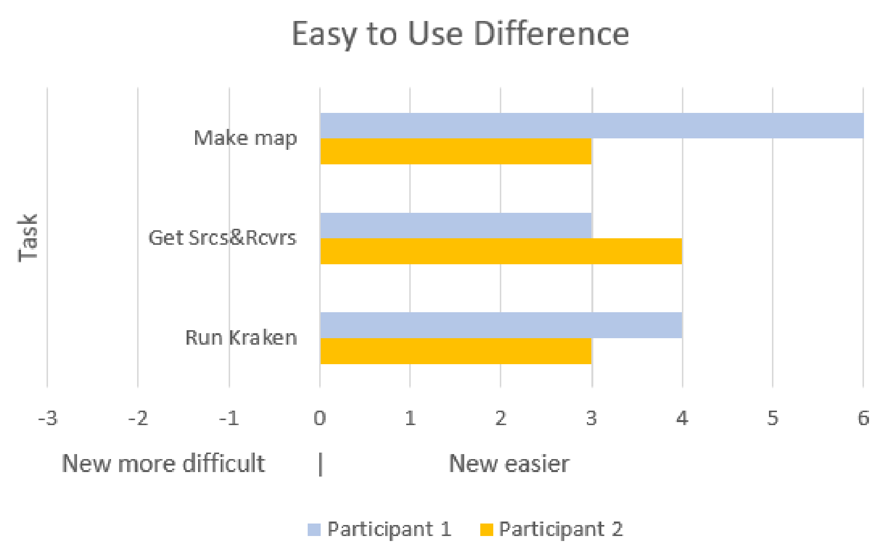


Figure 7.4: Differences on 'Easy to use' scores on original and new version (new version score - old version score)

7.3 User's Evaluations of the Versions

After the testing of each version the participants were asked to rate the user friendliness, how easy it was to use and how understandable the buttons was. Continuing with the scale from 1 to 7, where 1 is "Very unfriendly/difficult" and 7 is "Very friendly/easy". The participants rated the two versions that were tested with these evaluation questions:

EQ1. How would you rate the user friendliness this version?

EQ2. How easy to use would you say this version was?

EQ3. How easy would you say it was to understand what all the buttons would do?

Table 7.3 show the participants scores of the different questions on the different version. Participant 1 giving the low score of two to all the aspects on the original version, while participant 2 gave the score 3.

Table 7.3: User Evaluation of Original and New version

Version	Participant	User friendliness	Easy to use	Understanding of buttons
Original	1	2	2	2
	2	3	3	3
New	1	6	6	6
	2	6	6	6

The result is put in a bar graph shown in Figure 7.5 to illustrate the user evaluation on three different aspect of the original with gray bars and of the new version with green bars. Participant 1 is in a darker shade than Participant 2. The aspect that the participants rated were how user friendliness they would rate the version, how easy the versions was to use and how easy it was to understand what functions the different buttons on the GUI's were.

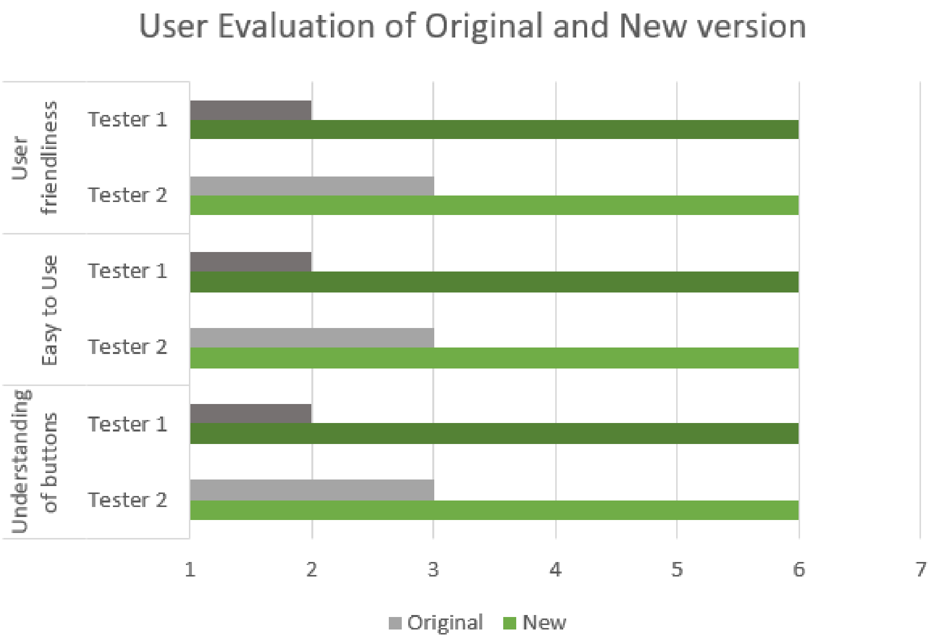


Figure 7.5: User ratings answering the Evaluations question EQ1, EQ2 and Eq3

7.4 Observations

Observations were done by remote screening of the participants performing the tasks. The different observations on the tasks will now be presented. On Task 1 being making the map on the original version both participant took around 4 minutes, but they both needed help finding how to get a round shaped map and to find where to choose the right database that would colour the map. One had also difficulties knowing how to make the map appear after having set the specifications. The similar task on the new version, Task 4, took around a minute for both participant, and there were no help needed to complete the task with all the specifications that were presented.

The second task on the original version covering the getting sources and receivers from file on the map that were visualized in the previous task, were complete in 5 minutes by both participants. One was able to do it without help but had some hesitations when it come to how to make them appear on the map. The other participant pushed a wrong button for plotting the sources, and was helped to find the right way. The same kind of task on the new version was done between 20 and 30 seconds by the participants, with no help given. Participant 1 expressed a bit of dissatisfaction with how the way sources and receivers that are outside the visualized map are placed at the edge of the map and how this mismatch with the path plotting that gives the line between the real placement of the source and receivers.

The last task on the original version was the running of KRAKEN model prediction. The participants were able to change all the parameters that were specified in the task, but could not understand how to decide the model and run it. The task was completed in 4 and 6 minutes with the help. When the task of running a model predictions was done on the new version in Task 6, the participants half their time with finishing in 1,5 and 3 minutes.

In the task where the participants were asked to do a more complicated and larger run of a new model, the participants finished in 2 and 2,5 minutes. While Participant 2 had no problems, Participant 1 overlooked one specifications as the prototype had not an updated menu window, as the text of a parameter box was cut and just said 'Number of' in stead' of 'Number of receivers'.

The final task covering the new function of opening saved previously run and visualizing these in different ways, the participants had no problem with the task and completed it within 2 minutes.

Chapter 8

Discussion

This chapter will summarize the key result, present interpretations and discuss the implications. With Chapter 6 results of the additional functions added to the existing Arctic Ocean package, and Chapter 7 results of a usability test, this chapter will interpret the result and discuss if the master project aims are accomplished.

8.1 Saving results

As there have been made possible to save result, researchers will save a lot of time by not having to run the same simulation again. The saved result opens the opportunity to further process and visualized for better understanding and researching of the Arctic Ocean. There are other ways to visualize the output data from the models with focus on other aspect of the result than the standard visualizations that are illustrated after a model prediction run.

The storing of results allows the user to easily revisit previous simulations without redoing the simulation (in which case the simulation parameters may be different) or save the entire workspace to a single file. This would store a lot of additional parameters that are not required, e.g. those used to set up the GUI.

An additional aspect of saving results are that the analysis of results can be simplified, e.g. by verifying the simulation parameters and even redo a simulation with different parameters as a sensitivity analysis. As the result files have timestamps etc., archiving results are easier by default than before.

8.2 Support of new Models

The expansion of making the program being able to run several other models than the two supported before gives a wider use of the program. This benefit the researcher that can then use one program to get the different model predictions on the same path. So gathering several models in a program makes it much more relevant and easier for a researcher to use in research. As the researcher now can learn to use one program's graphical user interface to be able to run all these models instead of learning how to set up the multiple different input files and how to plot the different model predictions.

8.3 Comparing Time steps and/or Models

The function of saving result temporary in an 2-dimensional-array after every modeling opened the possibility of running several simulation in one run. This includes simulation of several time steps, receivers and models. This opens for the options to compare and study the differences of time steps and model calculations on the same paths.

As the water in the ocean is constantly moving, the sound propagation is changing, so to be able to compare time variations can give the researcher a better understanding of the changes. Comparison of different time periods are important as the ocean is a rapid changing environment. Ocean currents, eddies, the sea-ice cover, greatly affect the acoustic propagation. One example here is the acoustic thermometry experiments, e.g. CAATEX, which are typically deployed for one year. The acoustic signals will change over the year, and modelling with sound speed profiles from different time periods are very useful in the planning and analysis of the results.

The visualisation combining different models gives a view on if the models compared gives similar predictions. With the ability to remove and view one of the model predictions plots gives a better visualization on the similarities of the models result. With the colouring change of the overlaying plot one can decide if one like to see the different with contrast or a similar colour for a softer change between the viewing and removing of the plots. One example is time fronts from MPIRAM and Eigenray 6.9. The output from MPIRAM gives a better picture of the intensity of the sound vs depth and time. However, these results do not provide information about where in the water column the sound has travelled from the source to the receiver. This is given by Eigenray, along with the arrival angle and arrival time.

8.4 Get area overview

The several receivers run can give a more spacial overview of a specific area. This expand the study of just one path to several paths surrounding a source, which leads to being able to concentrate on an area in the researching. Sound does not propagate along a single transect from a source to a receiver. The sound radiated by a sound source will rely on many parameters, e.g. the geometry and frequency. By automatically modelling the sound along a series of transects, the time spent setting up these simulations are greatly reduced compared to a manual approach. The horizontal maps of the spatial sound propagation can also reveal regions that are optimal for receiver locations. Another example is modelling of man-made noise and potential influence on marine mammal habitats and breeding grounds.

8.5 Usability

From the result of the usability test it is a distinct increase in the score of the new version from the original version. Ratings of the original one is mainly in the unsatisfying and difficult side of the scale, as the new version have scores all in the higher part of the scale telling it is satisfying and easy.

By interpreting the result in of the usability test illustrated in the tables and figures in Chapter 7 one get an overview of the usability of the original and new version. The result of the user question about rate of satisfaction after every task, written in column 2 of Table 7.1 and 7.2, shows that all except one task done on the original version are rated under the neutral score 4. The exception has the middle score 4 that represent neither satisfied nor unsatisfied. This shows that the scoring on the original version are on the unsatisfied side of the scale. The same tables also show the scores of satisfaction of using the new version after the next five tasks. Here all the scores except one middle score of 4 are in the satisfied side of the scale. Participant 2 giving the two highest scores after all the task done on the new version, that correspond to pretty and very satisfied, while Participant 1 rank more variation between 4 and 7.

Figure 7.1 illustrate the satisfaction scoring of after the task, this the clear show the satisfaction of doing the different task. Figure 7.3 illustrate the difference of the scoring of satisfaction on the similar task done on the different version. Here the positive scores will say that the newer version gives better score than the original on similar task. Five of the six tasks have increased with 3 or 4 for the different participant. This is a great difference as the highest possible is 6, that is just possible if the participants went from the lowest score meaning 'very unsatisfied' to the highest standing for 'very satisfied'. The only one that do not increase is the get source and receiver task rank by Participant 1 that expressed the dissatisfaction of the result visualisation the function made on the map. As this function was not change on this aspect the participant naturally scored the same satisfaction score.

In the third column in Table 7.1 and 7.2 the rank of easiness the two participants gave after every task. In five of the six rankings of the easiness of use the original version to complete the task, the score is under 4 meaning it was difficult to complete the task. This is reflects with the need of help in the same tasks, the participant was clear on not being able to complete the task with the specified settings. Participant 1 did manage to complete the Task 2 without problems and help, as this might be the simplest task to do.

The result of the participants original version evaluation ranking, shown in Table 7.3, are on the lower side of the scale meaning it was not user friendly, was difficult to use and the buttons were difficult to understand. While on the new version both participants gives the second highest score on all the ratings, meaning it was user friendly and easy to both use and understand the buttons. The increase of 3 and 4 of the rank gives a good indication that the new version has improved in the aspects of user friendliness and usability.

A validation question might be raised as some similar tasks are done first on the original and then later on the newer version, because it might make it easier in it self, and there are not a completely neutral ground for testing the new version. This might have an impact, but as both participants were aware of this case and there also were done new task on the new version make it possible to make assumptions on if the new version was more user friendly and easier to use.

As the usability testing result in several ratings and observations, there are a foundations to interpret the improvement of the further development on the program. Of course a larger amount of tester would be beneficial for a more reliable result. The tester could then be divided in to two groups, that tested and evaluate each versions or the groups could start with different version.

Chapter 9

Conclusions

In order to conclude if the aims of this master project have been accomplished, one has to look at the two main questions posed as its starting point.

Q1. How to expand and improve the Arctic Package, a tool for acoustic propagation modelling, for **easier** and **relevant** research in the fields of acoustics and oceanography?

This first question (Q1) asks for a relevant expansion of the existing tool Arctic Package for the research field Polar Acoustics and Oceanography. As the goals of the development were discussed with experienced researchers in this field, a set of relevant and convenient updates were identified, and found suitable for program development.

As part of this master project we managed to implement expansion of the original Arctic Package, by adding functions being potentially relevant and useful to researchers in the field. The expanded Arctic Package is now able to support several acoustic models and thus giving the possibility of using parameters from several other available data sets. The new developed program has been given a capacity of storing prediction results from already executed calculation runs, and in that manner allowing to process and visualize comparison of different models and consecutive time steps.

Expanding the functionality of original program turns it to be a more flexible program; one single program is able to run an increased number of different models. This capacity make it easier for researchers, as this program satisfy the need for running different models within one programs framework. This means that the researcher do not need to download and learn to use several programs, each of them adapted to a single or limited number models.

Q2. How can the tool Arctic Package be more user friendly by increasing the **usability**?

The second research question (Q2) raises the subject of need for a more user friendly program. The usefulness and actual use of the program will not only depend of the programs performance strength of giving out fast and reliable results. It is considered crucial to a programs usability to have an user-friendly interface, both for easy and logical initial setting of parameters and data, but also for flexible and relevant presentation of overall or detailed combination of results.

An interface usability test was performed on the original and the new revised version of the program to get an comparative user evaluation. The result from the usability tests shows that a higher level of users satisfaction was gained by using the new interface version. Both the qualitative and quantitative test data clearly indicates this, in spite of the fact that the revised program version were had to include specification of several new added functions. That is likely to imply that the new program version is more likely to be used, not only by experienced full time researchers, but also by students for more part time use for limited exploration tasks.

Altogether my conclusion is that the aims of this master project have been accomplished.

Chapter 10

Further work

There will always be wishes and needs for extending already developed program packages by means of optimizing processing speed and capacity, including more functional tasks and increase flexibility, and improve interface to make it more user friendly.

As demonstrated in the present work, the modularity of the Arctic Package allows for adding additional acoustic propagation models. One modelling technique that has not yet been included in the Arctic Package is wavenumber integration. The most popular code here is OASES [31], which has been used e.g. in [32] to model acoustic communication signals under a sea-ice cover. This code has fairly complicated input files, where the user would benefit from having a tool such as the Arctic Package to set up the simulations. In addition, this code is computationally intensive and is typically to be run on a server or super computer. Such an interface would however seems to be possible to implement within the Arctic Package.

It is made a setup to process the data that GECCO have, as a testfile of it is processed in the new version, this should be replaced by getting the GECCO data directly from the server at NERSC. This will provide daily reports from the ten years that are estimated by assimilating available observations of sea ice and ocean parameters.

Making the program be possible to run with MATLAB Runtime, could also be a beneficial upgrade, as that will make it not need the licences required for MATLAB. Another task could be to follow the way Dewan did in his master project by integrating a model written Matlab[21] to a graphical user interface made outside MATLAB.

References

[1] NERSC, *Home Page of Nansen Environmental and Remote Sensing Center*, 2019. [Online]. Available: https://www.nersc.no/.

[2] M. Porter, "The KRAKEN normal mode program," Naval Research Laboratory, Technical Report, May 1992. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a252409.pdf.

[3] ——, "The BELLHOP manual and user's guide: Preliminary draft," Jan. 2011.

[4] M. D. Collins, "A split-step padé solution for the parabolic equation method," *The Journal of the Acoustical Society of America*, vol. 93, no. 4, pp. 1736–1742, 1993. DOI: 10.1121/1.406739. eprint: https://doi.org/10.1121/1.406739. [Online]. Available: https://doi.org/10.1121/1.406739.

[5] M. Porter, "Bellhop3d user guide," *Heat, Light, and Sound Research, Inc., La Jolla, CA*, 2016.

[6] M. B. Porter, "Beam tracing for two- and three-dimensional problems in ocean acoustics," *J. Acoust. Soc. Am.*, vol. 146, no. 3, pp. 2016–2029, 2019. DOI: 10.1121/1.5125262.

[7] M. Collins, A. Turgut, R. Menis, and J. A. Schindall, "Acoustic recordings and modeling under seasonally varying sea ice," *Scientific Reports*, vol. 9, p. 11, 2019. DOI: 10.1038/s41598-019-44707-0.

[8] F. T. Klockmann, "Optimization of software for modeling acoustic monitoring systems. Monitoring polar areas for climate change," Master's thesis, Department of Computing, Mathematics and Physics at the Western Norwegian University of Applied Science, and Department of Informatics at the University of Bergen, Bergen, Norway, Jun. 2020.

[9] NASA, *Homepage of ecco*. [Online]. Available: https://ecco.jpl.nasa.gov/.

[10] *World ocean atlas*, Last accessed: 2020-06-06. [Online]. Available: https://en.wikipedia.org/wiki/World_Ocean_Atlas.

[11] G. Luy, *Arctic Ocean state estimates for 2007 using the GECCO model*, 2020. [Online]. Available: https://zenodo.org/record/3696686#.XqgASy2B3UI.

[12] M. Jakobsson, L. Mayer, B. Coakley, J. Dowdeswell, S. Forbes, B. Fridman, H. Hodnesdal, R. Noormets, R. Pedersen, M. Rebesco, H. W. Schenke, Y. Zarayskaya, D. Accettella, A. Armstrong, R. Anderson, P. Bienhoff, A. Camerlenghi, I. Church, M. Edwards, and P. Weatherall, "The International Bathymetric Chart of the Arctic Ocean (IBCAO) version 3.0," *Geophysical Research Letters*, vol. 39, Jun. 2012. DOI: 10.1029/2012GL052219.

[13] Woods Hole Oceanographic Institution, *Ocean Acoustics Library - OALIB*, Last accessed: 2020-06-06. [Online]. Available: https://oalib-acoustics.org.

[14] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*, 2nd. Springer Publishing Company, Incorporated, 2011, ISBN: 1441986774.

[15] J. Hovem and H. Dong, "Understanding ocean acoustics by eigenray analysis," *Journal of Marine Science and Engineering*, vol. 7, p. 118, Apr. 2019. DOI: 10.3390/jmse7040118.

[16] J. Bowlin, J. Spiesberger, T. Duda, and L. Freitag, "Ocean Acoustical Ray-Tracing Software RAY," Woods Hole Oceanographic Institution, Technical Report, Oct. 1992.

[17] Brian D. Dushaw, *MPIRAM: The RAM Parabolic Equation Code in Fortran 95*, Last accessed: 2020-06-06. [Online]. Available: `http://staff.washington.edu/dushaw/AcousticsCode/RamFortranCode.html`.

[18] B. D. Dushaw, *MATLAB KRAKEN*, Last accessed: 2020-06-06, 2010. [Online]. Available: `http://oalib.hlsresearch.com/Modes/matlab.kraken.tgz`.

[19] The MathWorks Inc., *MEX file functions*, Last accessed: 2020-06-06. [Online]. Available: `https://www.mathworks.com/help/matlab/call-mex-file-functions.html`.

[20] MATLAB, *R2019b*. Natick, Massachusetts: The MathWorks Inc., 2019.

[21] D. J. Dewan, "SOLWEIG – A climate design tool," 2009.

[22] The MathWorks, Inc., *MATLAB Runtime*, Last accessed: 2020-06-06. [Online]. Available: `https://se.mathworks.com/products/compiler/matlab-runtime.html`.

[23] NERSC, *CAATEX: Coordinated Arctic Acoustic Experiment*, 2020. [Online]. Available: `https://www.nersc.no/project/caatex`.

[24] H. Sagen, *Måle havtemperatur med lyd*, 2019. [Online]. Available: `https://caatex.nersc.no/node/18`.

[25] A. Hevner, "A three cycle view of design science research," *Scandinavian Journal of Information Systems*, vol. 19, Jan. 2007.

[26] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *Management Information Systems Quarterly*, vol. 28, pp. 75–105, Mar. 2004.

[27] M. A. Dzieciuch, P. F. Worcester, and W. H. Munk, "Turning point filters: Analysis of sound propagation on a gyre-scale," *J. Acoust. Soc. Am.*, vol. 110, no. 1, pp. 135–149, 2001. DOI: `10.1121/1.1377869`.

[28] J. Nielsen, *Usability 101: Introduction to usability*, 2012. [Online]. Available: `https://www.nngroup.com/articles/usability-101-introduction-to-usability/`.

[29] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*. New York: Wiley, 2002. [Online]. Available: `https://www.csun.edu/science/courses/671/bibliography/preece.html`.

[30] Cygwin authors, *CYGWIN*, Last accessed: 2020-06-06. [Online]. Available: `https://cygwin.com/index.html`.

[31] H. Schmidt, *OASES - Version 3.1 - User guide and reference manual*, Last accessed: 2020-06-06, 2004. [Online]. Available: `http://acoustics.mit.edu/faculty/henrik/oases.pdf`.

[32] G. Hope and H. Sagen and E. Storheim and H. Hobæk and L. Freitag, "Measured and modeled acoustic propagation underneath the rough arctic sea-ice," *J. Acoust. Soc. Am.*, vol. 142, no. 3, pp. 1619–1633, 2017. DOI: `10.1121/1.5003786`.