DBDB-V and USML Software Development

Christopher Tannock

Group 103

August 14, 2014

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Outline

- Initial Work
- Digital Bathymetric Database (DBDB-V)
- Under Sea Modeling Library (USML)
- Implementation of Software
- Future Work

Initial Work

Learning the concepts behind…

- **Linux Operating System**
- **GNU Compiler Collection**
- **Emacs Text Editor**
- **Calling C and C++ functions from within MATLAB**
- **Digital Bathymetry Database**
- **Writing programs in C++**
- **Under Sea Modeling Library**

Outline

- Initial Work
- Digital Bathymetric Database (DBDB-V)
- Under Sea Modeling Library (USML)
- Implementation of Software
- Future Work

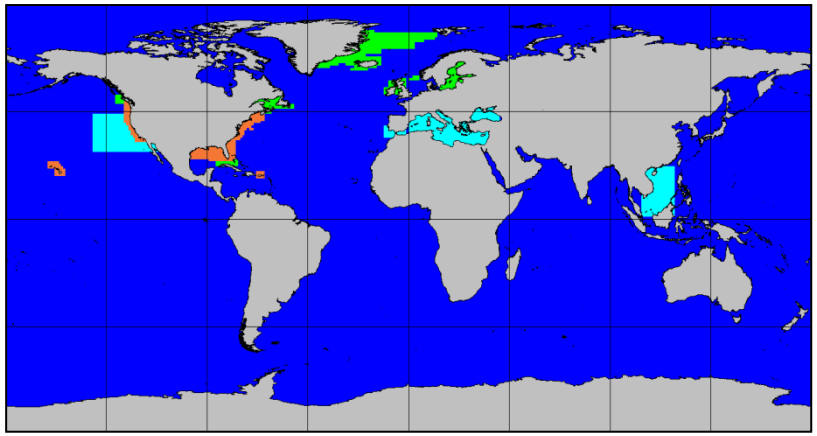
Digital Bathymetric Database – Variable Resolution (DBDB-V)

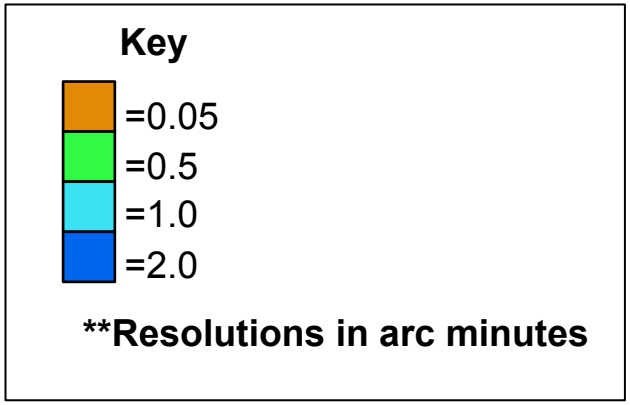
Overview:

- Variable resolution bathymetric database
- Open source with the API written in C
- Provided by Naval Oceanographic Office and is the most accurate / best bathymetric database for sonar applications

Goal:

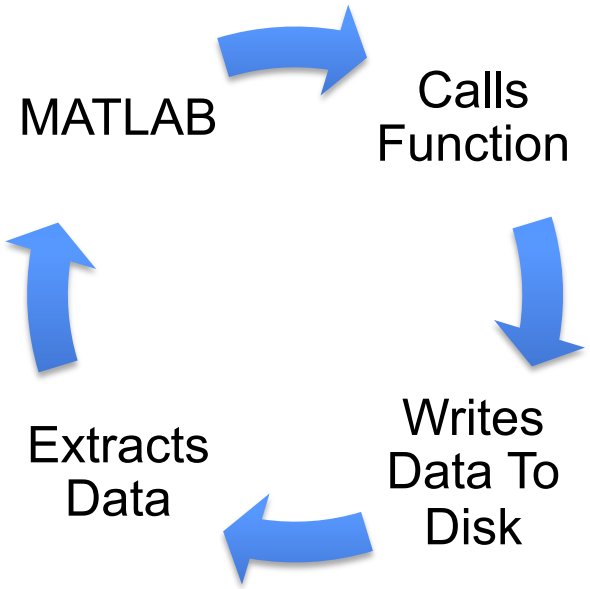
- To create executable functions using the provided API to increase runtime performance when extracting ocean depths within MATLAB





Extracting Data From DBDB-V
Formerly

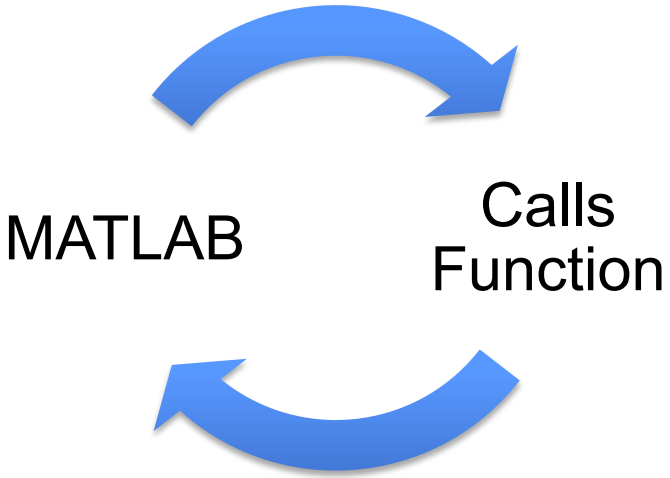
- Extract Data:
  - DBDB-V GUI or Commands (writes to .txt)
  - MATLAB Wrapper (reads out .txt file)
- Drawback:




```
Please enter/choose a source database:

For future use, you may set global environmental variables
  to point to the databases, avoiding the need to type in
  the path every time you access this program.  See the
  Database Description Document (Section 4.1.1.1) for details.

If you are opening a database other than level 0, be sure to
  also open the level0 database if you would like to extract from
  the unclassified, public release database.

Please enter the path and name of an additional database, if used.
  Press <enter> if no additional databases are to be opened.
>  /home/tannockc/tools/DBDB-V/data/dbdbv5_level0c_0.h5

  The database was successfully opened.

Please enter the path and name of an additional database, if used.
  Press <enter> if no additional databases are to be opened.
>

Please choose one of the following tasks:
  1 - Extract a single point
  2 - Extract a trackline
  3 - Extract a set of radials
  4 - Extract an area-of-interest (minimum bounding rectangle)
  5 - View available resolutions
  6 - View available security key details
  7 - View the databases' metadata
  8 - Exit
(hit <enter> for 1)>
```

Extracting Data From DBDB-V
Presently

- Extract Data:
  - Within MATLAB
    - dbv5_get_point()
    - dbv5_get_line()
    - dbv5_get_area()
    - build_dbv5_commands()

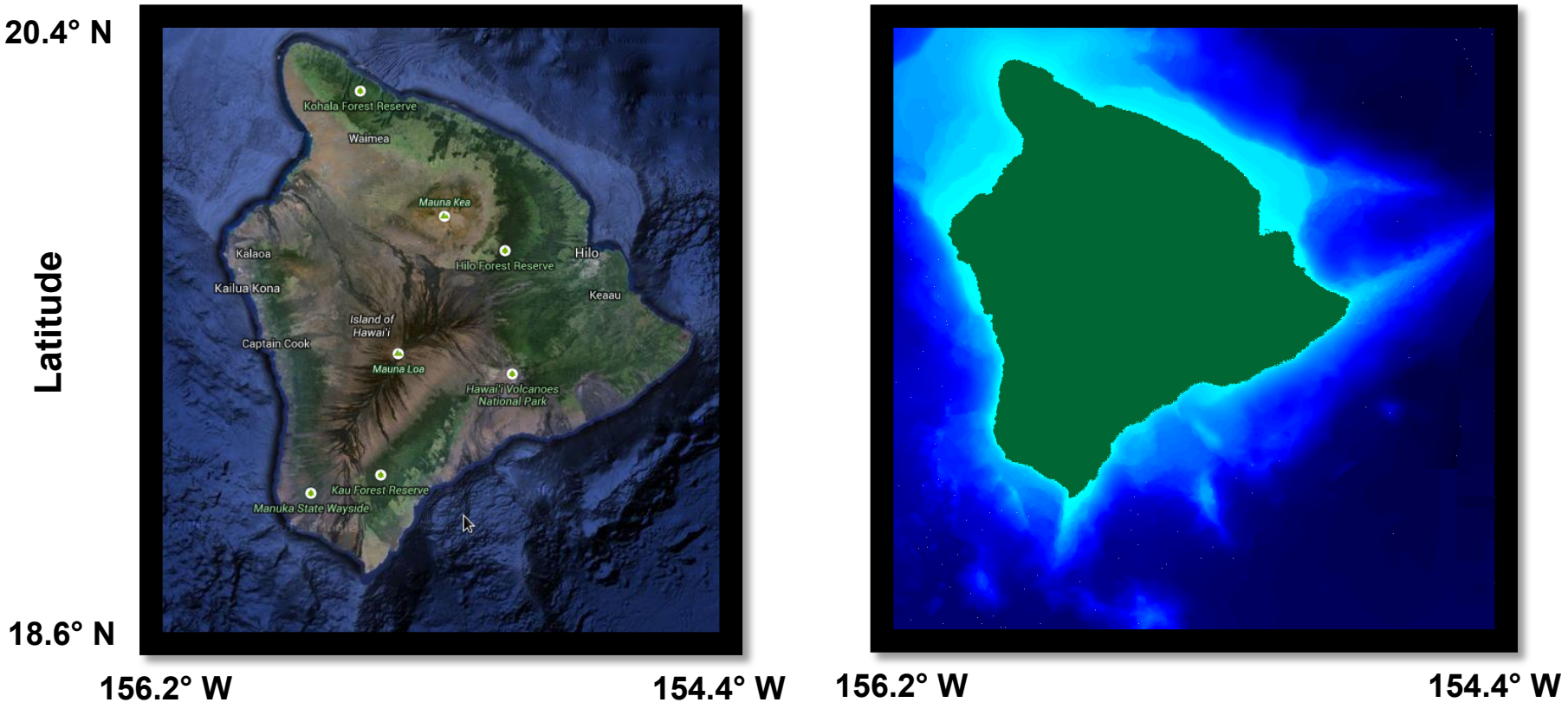


Point Inputs	Line Inputs	Area Inputs
Database	Database	Database
Coord. System	Coord. System	Coord. System
Latitude	Starting_Lat	South
Longitude	Starting_Lon	West
	Ending_Lat	North
	Ending_Lon	East
	Spacing	Spacing
	Tracktype	

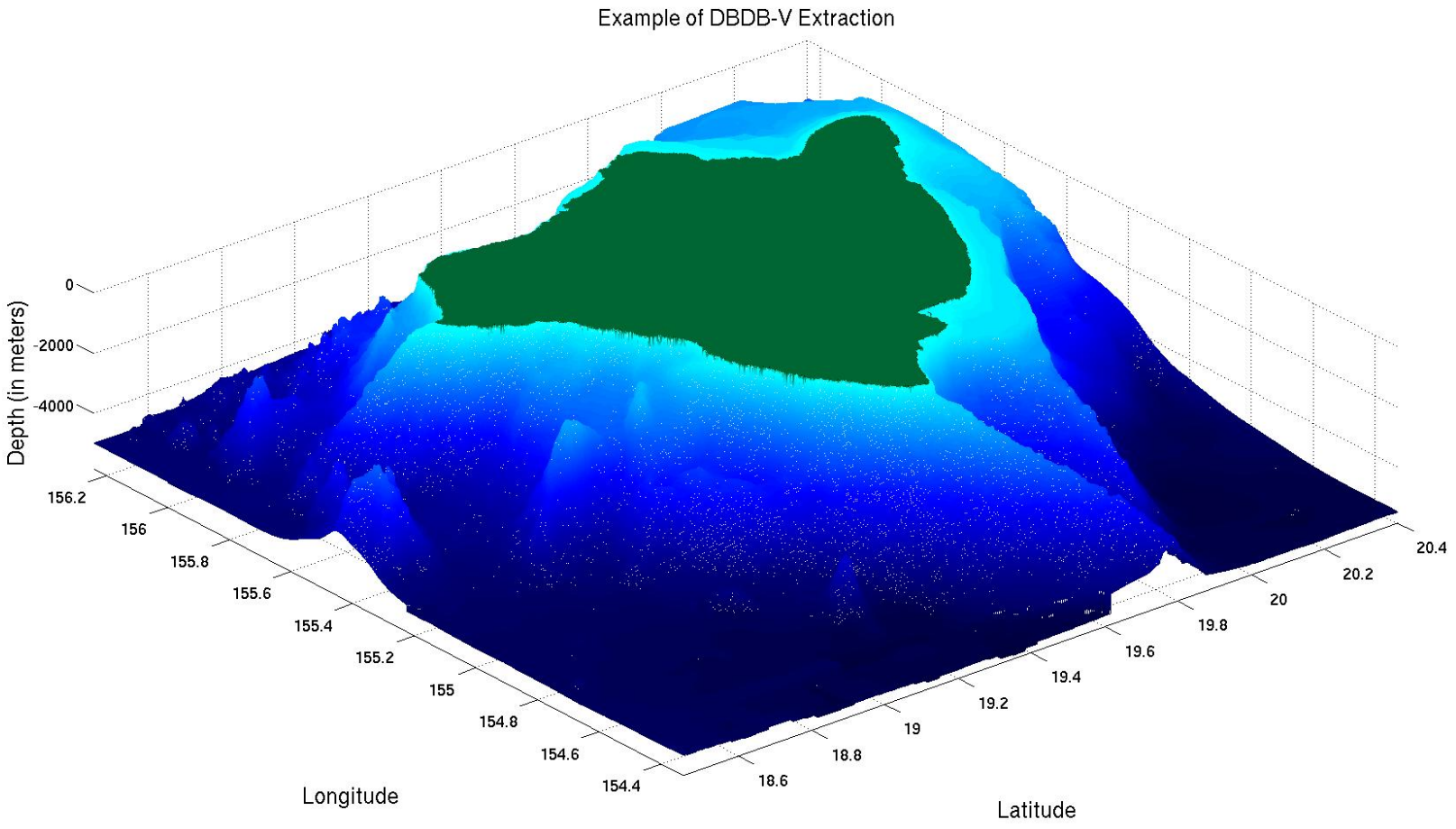
Example of Extraction Over Hawaii

Google Maps

DBDB-V



Example of Extraction Over Hawaii (3D)



Outline

- Initial Work
- Digital Bathymetric Database (DBDB-V)
- Under Sea Modeling Library (USML)
- Implementation of Software
- Future Work

Under Sea Modeling Library (USML)

Overview:

- Collection of open source C++ software which is used for sonar modeling and simulation
- Unique from other under sea ray-tracing modules because USML provides a convenient package for 3D modeling implemented in a modern programming language

Goal:

- Get the Under Sea Modeling Library up and running
- Learn how to run a basic Eigenray test model
- Implement reflection test models with gridded bathymetry bottom from DBDB-V

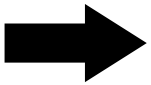
Basic Eigenray Test

Overview:

- The original Eigenray test was created to get a basic model running within USML
- Projected to be later built upon to add variable inputs:
  - Gridded bathymetry
  - Sound velocity profiles

Created Eigenray Test Directory:

- eigenray_simple.cpp
- eigenray_simple.h
- eigenray_main.cpp
- makefile
  - **eigenray** (executable)



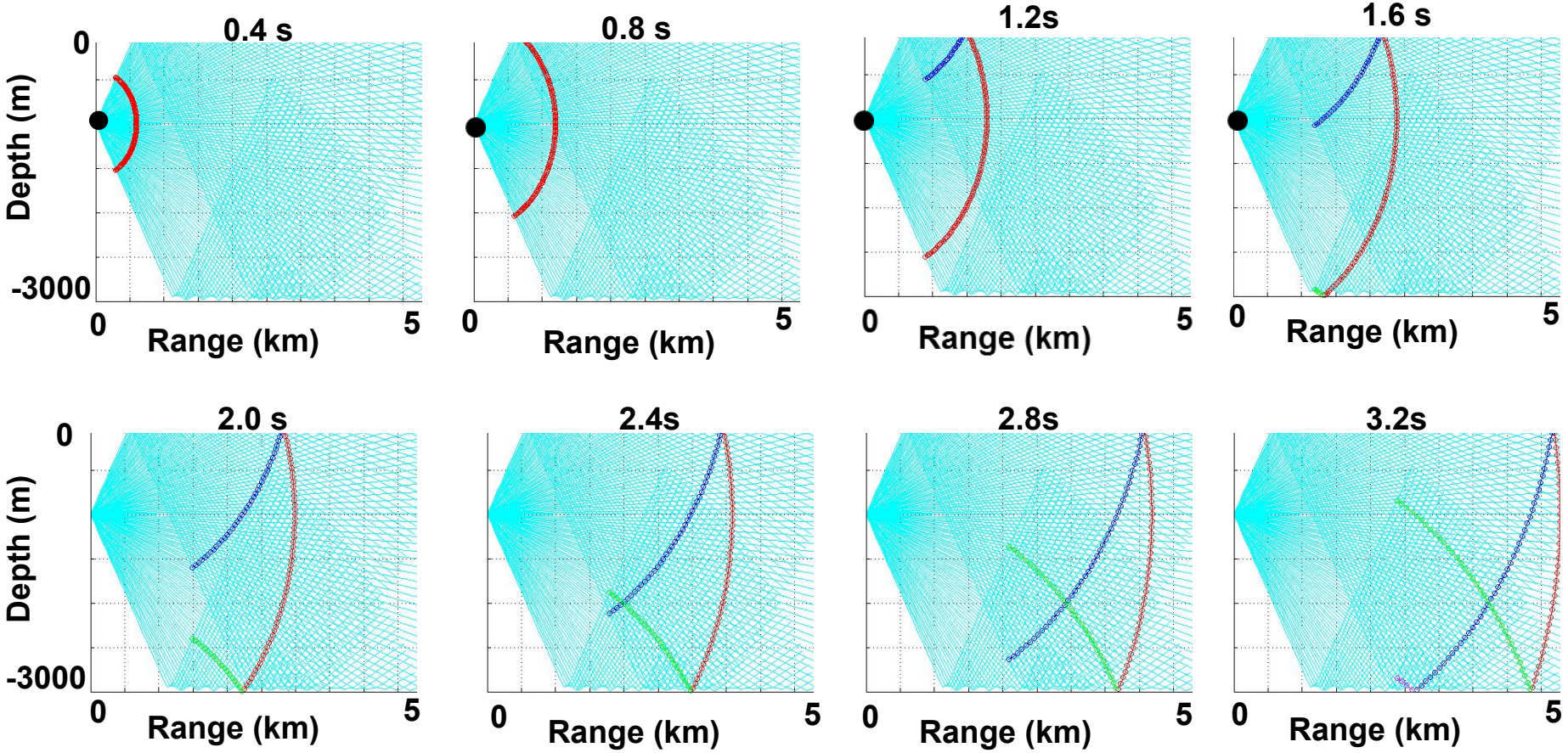
Outputs:

- Writes Wavefronts (.nc)
- Writes Propagation Loss (.nc)
- Writes Data Tables (.csv)

Eigenray Test (Flat Bathymetry)

Time Steps: 0.4 seconds
Time Max: 3.5 seconds
Source Location: (45, -45)

Source Depth: 1000 m
Speed of Sound: 1500 m/s
Depth of Ocean: 3000 m



Pros / Cons of Eigenray Test

Pros:

- First successful use of USML
- USML provides GUI for visualization
- User Can Input Data:
  - Time Steps
  - Time Max
  - Source / Target Locations
  - Sound Speed
  - Ocean Depth

Cons:

- Only works with flat bathymetry
- Uses constant sound velocity
- Currently not designed to work within MATLAB

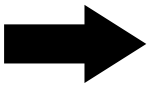
Eigenray Reflection Test

Overview:

- The Eigenray Reflection Test was created to build upon the Basic Eigenray Test
- Goal of incorporating the gridded bathymetry provided by DBDB-V

Created Reflection Test Directory:

- reflection_simple.cpp
- reflection_simple.h
- reflection_main.cpp
- makefile
  - ❖ **reflection** (executable)
- reflection_simple_mex.cpp



Outputs:

- Writes Reflection Grid Test (.nc)
- Writes Data Tables (.csv)

Reflection Test (Gridded Bathymetry)

Example Data From USML

Location:
Mediterranean Sea
(South East of Sicily)

Time Steps:
0.1 seconds

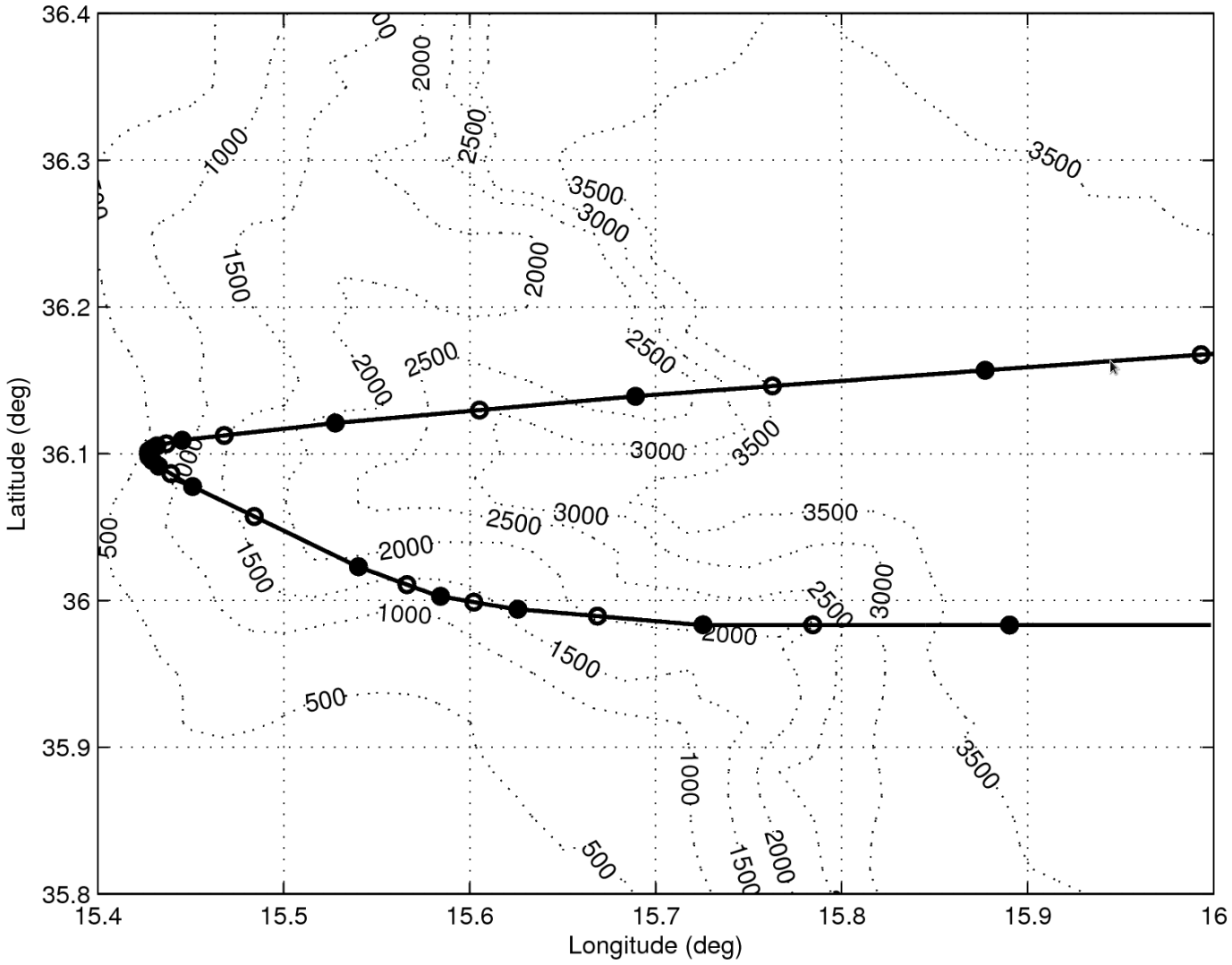
Time Max:
80.0 seconds

Sound Speed:
1500 m/s

Database:
ETOPO1

Key

● = Reflection off floor
◯ = Reflection off surface



Pros / Cons of Reflection Test

Pros:

- Builds on Basic Eigenray Test
  - Gridded bathymetry
- USML provides GUI for visualization
- User Can Input Data:
  - Time Steps
  - Time Max
  - Source / Target Locations
  - Sound Speed
  - Ocean Depth
  - Gridded Bathymetry
- MATLAB Capability

Cons:

- Uses constant sound velocity
- Gridded bathymetry currently only works with ETOPO databases
  - Working on adding DBDB-V

Outline

- Initial Work
- Digital Bathymetric Database (DBDB-V)
- Under Sea Modeling Library (USML)
- Implementation of Software
- Future Work

Implementation of Software

- Transferred all files and folders from upstairs computer to sonar lab
- Successfully built and used DBDB-V in the sonar lab with the improved extraction functionality
- Currently transferring RPMs to sonar lab to help build USML

Outline

- Initial Work
- Digital Bathymetric Database (DBDB-V)
- Under Sea Modeling Library (USML)
- Implementation of Software
- Future Work

Future Work

- Successfully build USML in the sonar lab
- Fix memory allocation leaks in Eigenray Reflection Test code to use DBDB-V gridded bathymetry within test models
- Add functionality for sound velocity profiles within Eigenray Reflection Test
- Add wrappers around DBDB-V to set default parameters and limit user inputs
- Acquire updated version of DBDB-V

Acknowledgements

- Ben Evans
- Nick Greene
- Evan Hanusa
- Dereck Padden
- Jordan Rosenthal