Heat, Light, and Sound Research, Inc.

Environmental Slicer: Getting Started

Michael B. Porter, John Peterson, and
Laurel J. Henderson
November 24, 2011

Heat, Light, and Sound Research, Inc. 3366 N. Torrey Pines Ct. Suite 310 La Jolla, CA 92037 **T** +001 858 457 0800
mikeporter@hlsresearch.com www.hlsresearch.com

Environmental Slicer: Getting Started

Introduction

The various modeling tools in the Acoustics Toolbox require input files describing the bathymetry, oceanography, and bottom-type along a bearing line from the source. The Environmental Slicer extracts this information from standard databases and sets up the propagation run. Thus it automates the process of calculating the field on a slice.

This package was originally developed as part of SimplePlan and SimpleSide which are Matlab GUIs for doing propagation modeling. In order to have some stand alone tools that could be incorporated in other applications, the core components have been separated to form the Slicer package.

Using the Slicer

The Slicer is called through a Matlab script which defines the vertical plane by the lat/long coordinates of a beginning and end point. An example is given in the script run_slicer_bellhop.m reproduced below. Working through the code we note that it starts with a 'slicer_init' call that defines basic configuration parameters such as the command used to execute an acoustic model. In addition, it reads in the bathymetry data which must be in the from of a standard 'xyz' file that can be downloaded from the National Geophysical Data Center (Design-a-Grid).

To plot the bathymetry, we first use the routine LoadBathymetry.m followed by writebty3d.m. These routines read the xyz format and write it out in a simpler rectangular grid format that is a standard we use for our 3D software. The resulting bathymetry data can then be plotted using plotbty3d.m to yield a plot as shown below. The Matlab functions are all included in the Acoustics Toolbox.

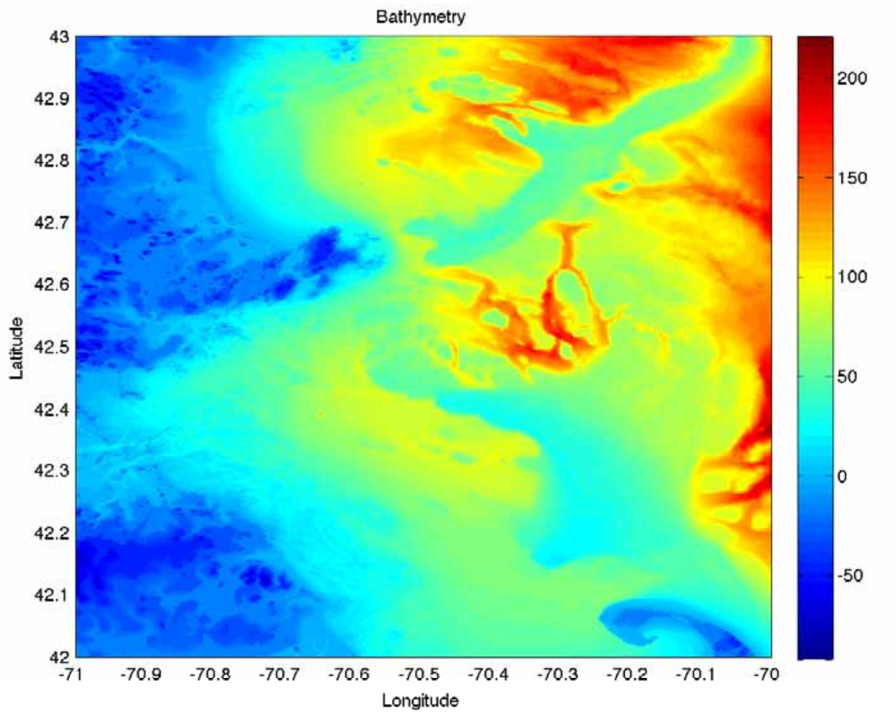


Fig. 1: Bathymetry for the Stellwagen Bank (degrees).

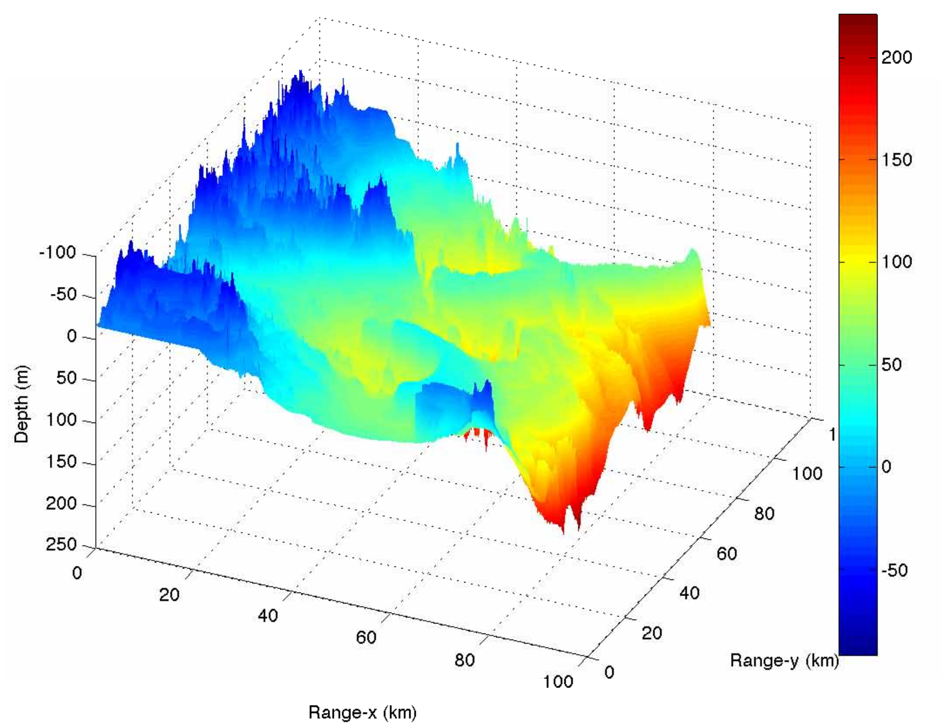


Fig. 2: Bathymetry for the Stellwagen Bank (km).

When the slicer runs it will extract the bathymetry for the vertical plane; however, it needs a lot of other information for the acoustic model such as the source frequency, ocean sound speed profile, bottom type. In addition it needs a definition of the plane of the slice. These parameters are all passed through a structure-variable called 'Slice'. The easiest way to initialize these is by reading an environmental file in the standard format used by models in the Acoustics Toolbox (KRAKEN, BELLHOP, SCOOTER, ...). The Matlab function 'read_env.m', which is also part of the Acoustics Toolbox, reads these files. The next set of lines defines the vertical slice in an obvious way. The sampling of the field in terms of the number of range and depth points is also specified.

Once this information is set up, 'slicer' is called and it returns the transmission loss (in dB) for subsequent plotting. The slicer can call either BELLHOP (ray/beam tracing) or RAM (PE) as selected by the variable 'Slice.acoustics_model'. In addition, the vertical plane can be specified either by its endpoints or by a source point and a bearing angle. An example of these latter features is given in run_slice_ram.m.

```
% Read in the local environmental slicer configuration file

[ ConfigParams, BathyData ] = slicer_init;

% Populate the relevant problem parameter fields of the Slice struct

Slice.acoustic_model = 'BELLHOP';
Slice.bathy_delta_km = 0.05;    % step size of interpolated bathymetry
Slice.bathy_interp   = 'C';
Slice.bathy_rd       = [];

Slice.SSP_range = [];
Slice.SSP_mat   = [];

%%
% The struct fields that follow are populated by calling read_env(),
which gets
% these values from an existing BELLHOP env file.

[ TitleEnv, freq, SSP, Bdry, Pos, Beam, cInt, RMax, fid ] = read_env(
'Stellwagen.env', 'BELLHOP' );
fclose( fid );

Slice.TitleEnv = TitleEnv;
Slice.freq     = 250.0;    % acoustic frequency in Hz
Slice.Bdry     = Bdry;     % air/sea and bottom interface parameters
Slice.Beam     = Beam;
```

```
Slice.cInt      = cInt;
Slice.Pos       = Pos;      % includes source depth
Slice.RMax      = RMax;
Slice.SSP       = SSP;      % sound speed profile

%%
% receiver geometry of the computational plane / slice

Slice.Plane.method     = 'ENDPOINTS';
Slice.Plane.src_lat    =  42.630;  % lat/lon of the source
Slice.Plane.src_lon    = -70.600;
Slice.Plane.rcv_lat    =  42.720;  % lat/lon of most distant receivers
Slice.Plane.rcv_lon    = -70.599;
Slice.Plane.max_d_m    = 300.000;  % depth of deepest receivers in me-
ters
Slice.Plane.nrr        = 500;      % number of receivers in range
Slice.Plane.nrd        = 200;      % number of receivers in depth

% Run the propagation model on the Slice

[ Slice_out, tl_db, rc ] = slicer( ConfigParams, Slice, BathyData );
%%

% plot TL

rcv_ranges = Slice_out.Pos.r.range;
rcv_depths = Slice_out.Pos.r.depth;

figure
imagesc( rcv_ranges, rcv_depths, tl_db )
title( 'BELLHOP example for Stellwagen Bank - Coherent TL' );
xlabel( 'Range (km)' )
ylabel( 'Depth (m)' )

tej = flipud( jet( 256 ) );  % 'jet' colormap reversed
colormap( tej )
caxisrev( [ 30 100 ] )
```

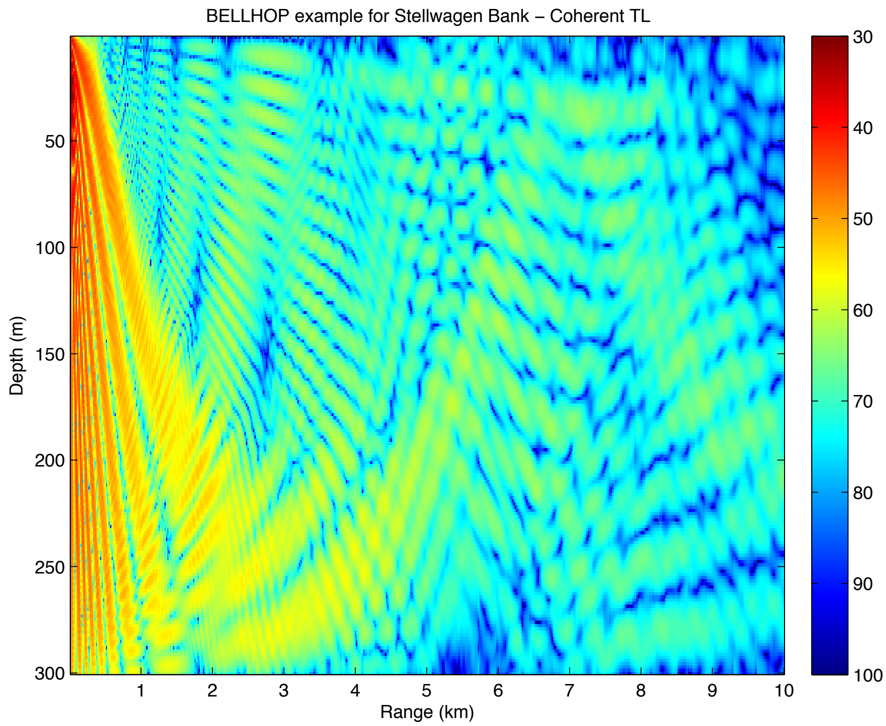


Fig. 3 : BELLHOP transmission loss as returned by the Slicer program.

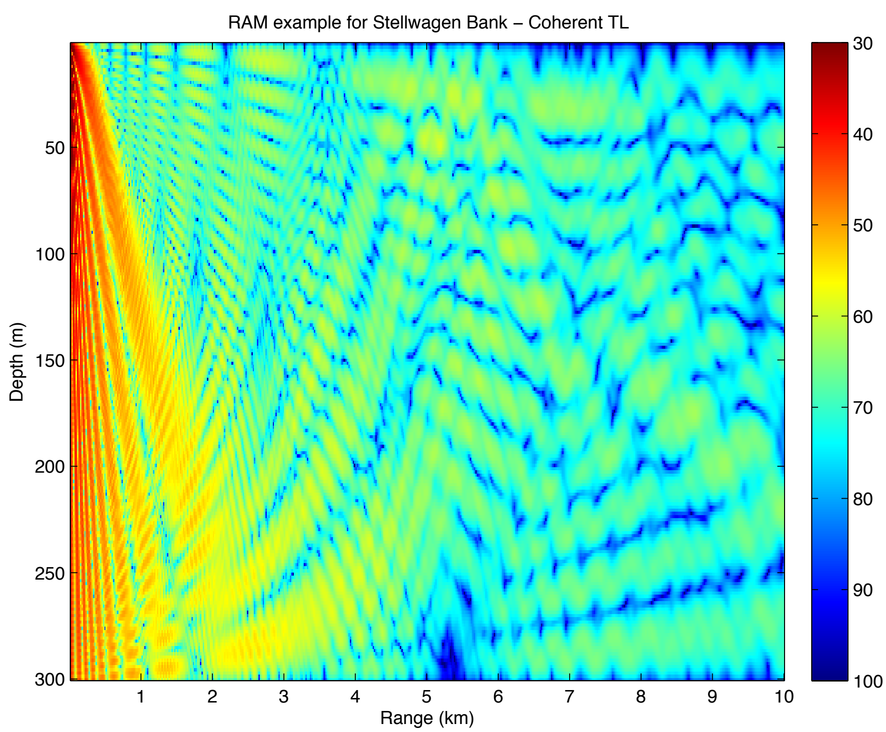


Fig. 4 : RAM transmission loss as returned by the Slicer program.

Using the Slicer for an x-y Grid

Often one wants the TL on a horizontal plane rather than a vertical one. This is fundamentally a more involved calculation since ocean acoustic models generally work intrinsically on a vertical slice. Thus to produce the field on a horizontal slice, the acoustic model needs to be run on a sequence of bearings. Further, if the desired TL grid is rectangular in the x-y plane, then the resulting field needs to be interpolated from the polar grid onto that rectangular grid. These functions are automated by another Matlab function called 'tlgrid.m'.

An example calling script for tl_grid is shown below. Note that tl_grid also extracts the sound speed profile automatically from the World Ocean Atlas. Therefore it does not need to call read_env.m to read an existing environmental file with that information. The World Ocean Atlas provides an ocean climatology based on monthly, seasonal, or annual averages. Here we use the monthly averages with the specific month selected by the variable Params.SSP_month.

An additional feature for tl_grid is that we can specify a bandwidth for frequency averaging. This is treated in a approximate way by doing range-averaging using formulas suggested by Harrison.

```
% This script provides an example of how to call the tl_grid() function

Params.BathyFile = 'Stellwagen.xyz'; % NGDC bathymetry for Stellwagen Bank
Params.acoustic_model = 'BELLHOP';   % either 'BELLHOP' or 'RAM'

Params.src_lat   =  ( 42.5 );        % source latitude  in decimal degrees
Params.src_lon   = -( 70.6 );        % source longitude in decimal degrees
Params.src_depth = 15.0;             % source depth in meters
Params.src_freq  = 100.0;            % source frequency in Hz
Params.src_bw    = 0.075;      % percent bandwidth, [max(f)-min(f)] / mean(f)

Params.SSP_month = 6;   % month of year for WOA SSP, 1=Jan, 2=Feb ...

% Sediments in Stellwagen Bank are generally Terrigenous Gravel, Sand.
%
% geo-acoustic parameters for grain size of phi = 3, fine/vf sand (Wentworth)
Params.Bot.alphaR = 1593.0;
Params.Bot.betaR  = 0.0;
Params.Bot.rho    = 1.339;
Params.Bot.alphaI = 0.592;
Params.Bot.betaI  = 0.0;

% The following fields control the lat, lon grid where TL is interpolated.
% The dimensions of each 'axis' of the cube and the number of points is
% controlled here. Each is specified by providing the start and end values,
% and the number of grid points in that direction. Southern latitudes and
% Western longitudes are negative numbers (in decimal degrees).
```

```
Params.grid_lat = [ 42.5,  42.8, 251 ];
Params.grid_lon = [-70.8, -70.2, 201 ];
Params.grid_depths = [ 10, 50, 5 ];     % depths: 10, 20, 30, 40, 50 m

% The following parameters control the resolution of the underlying polar
% grid where TL is calculated.
% This is a fan of vertical planes. Using lower resolution increases the
% accuracy of the interpolation (at the expense of more computer time).

Params.bearing_resolution  = 2.0;    % resolution of polar angles (degrees)
Params.range_resolution_km = 0.05;   % resolution of range receivers (km)

% Run the tl_grid function, it returns TLmat( nlat, nlon, ndepths )

TLmat = tl_grid( Params );
%%

% Example plot

lat_deg = linspace( Params.grid_lat( 1 ), Params.grid_lat( 2 ),
Params.grid_lat( 3 ) );
lon_deg = linspace( Params.grid_lon( 1 ), Params.grid_lon( 2 ),
Params.grid_lon( 3 ) );

figure
imagesc( lon_deg, lat_deg, TLmat( :, :, 2 ) )
axis xy
title( 'Stellwagen Bank Example - Receiver depth = 20 m' );
xlabel( 'Longitude (degrees)' );
ylabel( 'Latitude (degrees)' );

tej = flipud( jet( 256 ) );  % 'jet' colormap reversed
colormap( tej )
caxisrev( [ 50 110 ] )
```

The resulting TL is shown below. Note that the transmission loss plot follows the contours of the land. This particular run was done using BELLHOP at 100 Hz and took about 90 seconds on an iMac. Because the grid is computed using many independent runs on different bearings, the run times are longer than for a single slice.

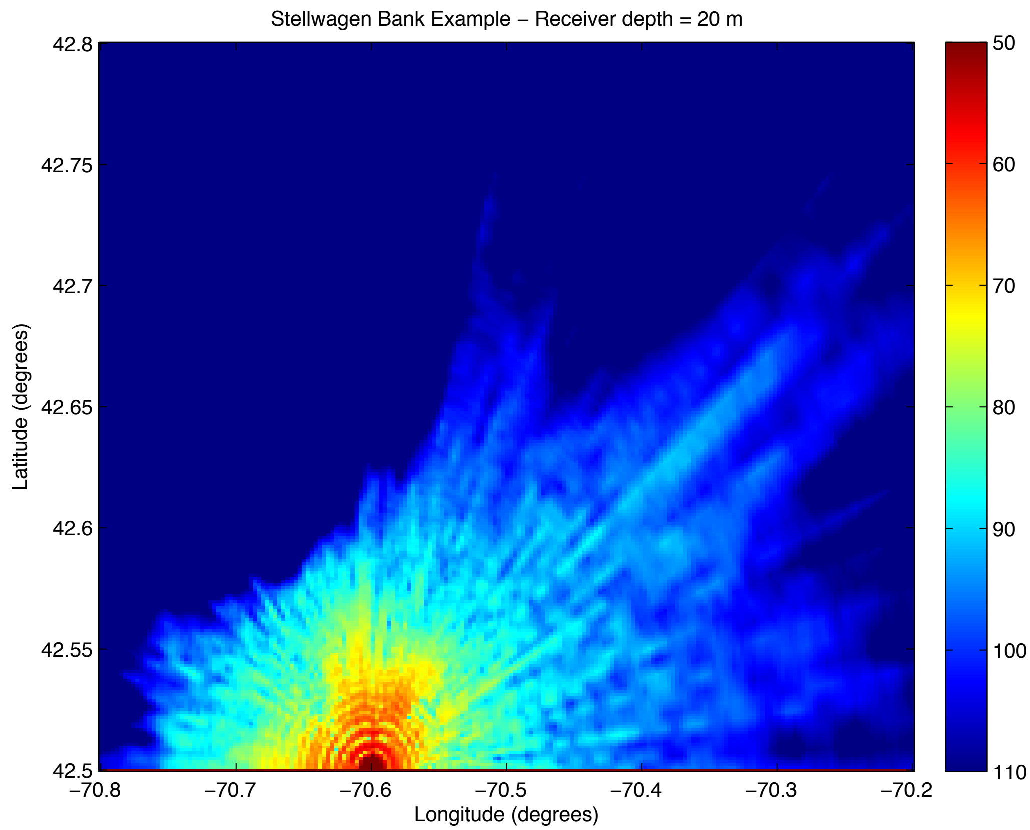
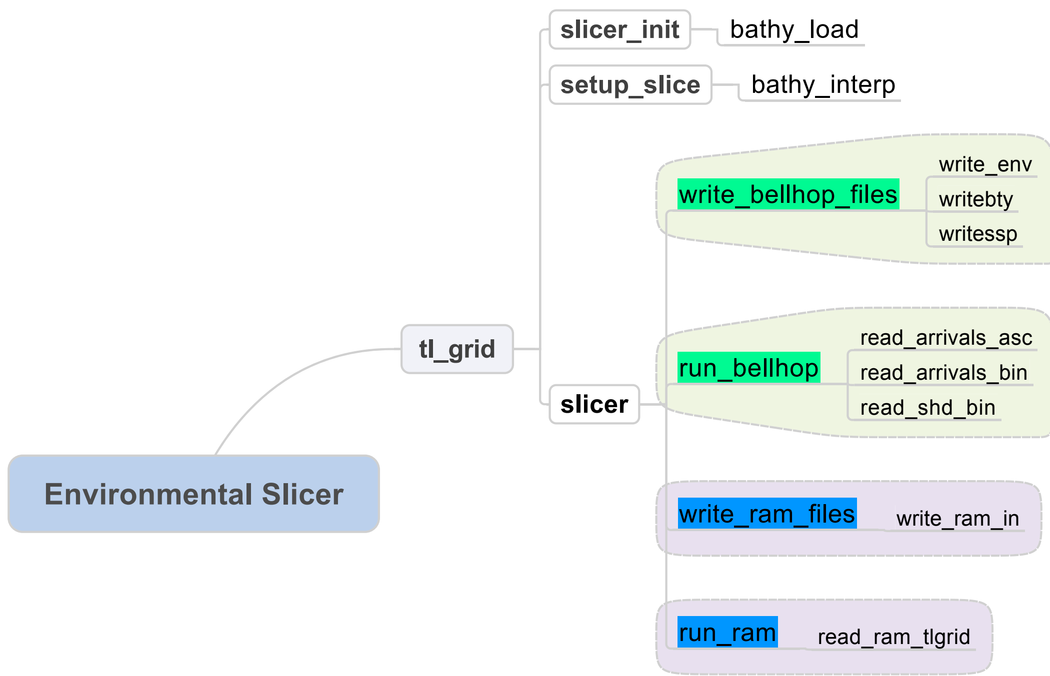


Fig. 5: TL plot for the Stellwagen Bank produced by TL_grid.

Organization of the Slicer Package

The Slicer package is a directory of Matlab files that should be installed in the Acoustics Toolbox in the directory at/Matlab/Slicer. The structure of the package is shown below. The main routine is 'slicer.m' which calls routines to write the model input files and run the acoustic model (write_bellhop_files, run_bellhop, write_ram_files, run_ram). The slicer_init routine needs to be called before slider to set up the commands to run various models and to load the bathymetry. Note that Stellwagen.xyz is currently hardwired in.

The Slicer package takes advantage of (and requires) existing routines in the Acoustics Toolbox to read and write various input and output files for BELLHOP.



Summary and Future Work

This report documents a preliminary version of the Slicer. It attempts to unify software that has been used and developed by different people at HLS; however, further work is needed. The routines for reading and writing RAM files are not unified in the sense that the Acoustics Toolbox also has similar tools. In addition the bathy_load command is essentially duplicated by LoadBathymetry in the Acoustics Toolbox. Further, we have found issues in setting up the RAM files to obtain precisely the desired output grid.

The slicer_init code should be called automatically by slicer on the first call so that the user does not have to think about that. In addition, features in TL_grid (the World Ocean Atlas and frequency averaging should also be implemented when the Slicer is called directly). Finally, there is currently no error checking if the slice is not fully contained within the box where the bathymetry has been supplied. In such cases, the slice run will silently fail and produce erroneous results.

These issues will be addressed in a future release; however, the existing version should be a useful tool.