A Consistent, User Friendly Interface for Running a Variety of Underwater Acoustic Propagation Codes

Alec J Duncan, Amos L Maggi

Centre for Marine Science and Technology, Curtin university of Technology

ABSTRACT

A number of underwater acoustic propagation modelling programs have been placed in the public domain by their authors. Different programs are required for different situations, but together they can perform the majority of common modelling tasks. All these programs are input file driven, but with different file formats, and each requires considerable time and effort to be expended in order to work out its specific requirements. The programs also generate output data in a number of different formats, making the setting up of software for visualisation or further processing a time consuming process. Several years ago the Centre for Marine Science and Technology publicly released a free Graphical User Interface (GUI) that provides a simple, consistent interface for running the various propagation routines in the Acoustic Toolbox, a collection of underwater acoustic propagation modelling programs written by Mike Porter from HLS Research. The GUI is written in Matlab, and also provides a variety of output data visualisation tools. It provides ready access to programs for acoustic field calculation by normal modes, wavenumber integration, and ray and beam tracing, and can also compute reflection coefficients for complicated layered seabeds. These codes are only suitable for modelling range-independent environments, with the exception of the ray code which can handle range-dependent bathymetry. An expanded version of this interface has recently been released and can be downloaded for free from CMST's web site. The primary enhancement is to provide fully range-dependent modelling through the inclusion of support for the RAM family of parabolic equation codes written by Mike Collins at the US Naval Research Laboratory. This paper describes the capabibilities of this new version of the GUI, and provides examples of its use.

INTRODUCTION

The complexity of underwater acoustic propagation makes numerical modelling an essential tool for researchers and an extremely desirable tool for educators, because in many cases modelling provides significant physical insight that isn't otherwise available. A number of different models have been developed over the years, each with its own range of applications, and many of them are freely available over the internet. (See, for example, www.hlsresearch.com/oalib/). Unfortunately these codes are invariably file driven, and only sketchily documented, making them a daunting prospect for a new user, and definitely not something you can let the average student loose on.

Several years ago the Centre for Marine Science and Technology publicly released a free Graphical User Interface (GUI) that provides a simple, consistent interface for running the various propagation routines in the Acoustic Toolbox, a collection of underwater acoustic propagation modelling programs written by Mike Porter from HLS Research. The GUI is known as the Acoustic Toolbox User Interface (AcTUI V1.6). It is written in Matlab, and also provides a variety of output data visualisation tools.

The programs supported by AcTUI V1.6 are listed in Table 1. They include a high-frequency propagation code (BELLHOP) that can handle range-dependent bathymetry (but not sound speed), and several low-frequency, range-independent codes (KRAKEN or KRAKENC together with FIELD, and SCOOTER together with FIELDS).

The biggest limitation of AcTUI V1.6 is the lack of support for a low-frequency, range-dependent code, although a fully range-dependent high-frequency code would also be useful. The latest public release of the GUI, which is known as the Acoustic Toolbox User interface and Post processor (AcTUP V2.2*l*), addresses the first of these problems through support for two customised versions of the Range-dependent Acoustic Model (RAM) written by Mike Collins from the US Naval Research Laboratory. RAM is an extremely efficient parabolic equation (PE) code that copes naturally with range-dependent environments and overcomes the principle limitation of the PE method: lack of accuracy for energy propagating at large angles to the horizontal (Collins 1992, Collins 1993, Collins 1999).

It is hoped to address the lack of a fully range-dependent high-frequency code in the near future through the provision of a fully range-dependent version of BELLHOP. This program is currently under development at CMST.

AcTUP V2.2*l* includes the following additional features:

- Compatibility with the 2006 release of Mike Porter's Acoustic Toolbox (Porter, 2005).
- A redesigned Run Definition class combining model environment and propagation parameters into a single structure. This offers increased robustness, facilitating full documentation and recovery of all model parameters.
- Ability to set up batch runs of one or more propagation codes using the same user friendly interface used for single runs.
- A redesigned and more connected user interface to cope with the significantly expanded functionality.

See Table 1 for a full list of the codes supported by AcTUP V2.2*l*.

Table 1 Propagation codes supported by AcTUI V1.6 and AcTUP V2.2*l*.

Name	**Description**	**V1.6**	**V2.2*l***
BELLHOP	Beam tracing program	Y	Y
BOUNCE	Plane wave reflection coefficient for layered seabeds	Y	Y
FIELD	Used with KRAKEN or KRAKENC to compute transmission loss for range-independent environments.	Y*	Y
FIELDS	Used with SCOOTER to compute transmission loss for range-independent environments. (Wavenumber integration by fast-field method)	Y*	Y
KRAKEN	Real normal modes	Y	Y
KRAKENC	Complex normal modes	Y	Y
RAMGEO	Low frequency transmission loss in range-dependent environments - fluid seabeds.	N	Y
RAMSGEO	Low frequency transmission loss in range-dependent environments - elastic seabeds.	N	Y
SCOOTER	Computes depth dependent Greens function (horizontal wavenumber spectrum) for range-independent environments.	Y	Y

Y indicates specified code is included but is run automatically when one of the associated codes is run and doesn't appear explicitly on the menus.*

THEORY

This section is a very brief introduction to the theoretical basis of the various propagation models supported by AcTUP V2.2*l*. A detailed account can be found in Jensen et. al., (2000).

All acoustic propagation codes are ultimately based on the wave equation:

$$\rho(\mathbf{r},t)\nabla\cdot\left(\frac{1}{\rho(\mathbf{r},t)}\nabla p(\mathbf{r},t)\right)-\frac{1}{c(\mathbf{r})^2}\frac{\partial^2 p(\mathbf{r},t)}{\partial t^2}=f(\mathbf{r},t) \quad (1)$$

Here ρ is density, p is pressure, t is time, $\mathbf{r}$ is the position vector, ∇ is the spatial gradient operator, c is the sound speed in the medium, and f is a forcing term which represents the acoustic sources.

In most situations in ocean acoustics changes in density occur over relatively large spatial scales, apart from at boundaries between media (eg. at the seabed) where the density is effectively discontinuous. In this case, within a single medium, (1) reduces to the more familiar form:

$$\nabla^2 p(\mathbf{r},t)-\frac{1}{c(\mathbf{r})^2}\frac{\partial^2 p(\mathbf{r},t)}{\partial t^2}=f(\mathbf{r},t) \quad (2)$$

Changes in media can be handled by solving (2) independently in each medium, and using continuity conditions between media (usually continuity of pressure and normal velocity) to evaluate unknown coefficients.

Unfortunately (2) involves four independent variables (three spatial and time), and direct numerical solution using finite difference or finite element techniques is impractical except in situations where the computation domain is a relatively small number of acoustic wavelengths in size. Most problems of practical interest in underwater acoustics involve computational domains several orders of magnitude larger than this, so it is necessary to reduce the dimensionality of the problem, and to introduce assumptions that allow the use of more efficient numerical techniques.

All of the codes discussed here start by assuming a harmonic forcing term and pressure, i.e.: $f(\mathbf{r},t)=f(\mathbf{r})e^{-i\omega t}$ and $p(\mathbf{r},t)=p(\mathbf{r})e^{-i\omega t}$ where ω is the angular frequency. Substitution into (2) leads to the Helmholtz equation:

$$\nabla^2 p(\mathbf{r})+k(\mathbf{r})^2 p(\mathbf{r})=f(\mathbf{r}) \quad (3)$$

where $k(\mathbf{r})=\dfrac{\omega}{c(\mathbf{r})}$ is the acoustic wavenumber. Equation (3) has the advantage that it can be solved one frequency at a time, making it more tractable than (2), particularly if the solution is only required at a single frequency or over a small range of frequencies.

Acoustic propagation problems often involve point sources and have cylindrical symmetry, or at least have environments that vary slowly enough in azimuth that the azimuth dependence can be ignored when computing the transmission in a particular direction. In this case $\mathbf{r}=(r,z)$, where r is the horizontal range and z is depth, $\nabla^2=\dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial}{\partial r}\right)+\dfrac{\partial^2}{\partial z^2}$,.and (3) now has only two independent variables. However, for many underwater acoustic problems the solution of even this two-dimensional equation requires too much numerical effort, and further assumptions are necessary. There are a number of options:

Depth separated Helmholtz equation

If we assume that the environment is range-independent then the sound speed and boundary conditions are functions of z only. If we further assume a single point source at $z=z_s, r=0$, we can separate (3) into depth and range equations:

$$\frac{d^2 p(k_r,z)}{dz^2}+\left(k^2-k_r^2\right)p(k_r,z)=A\delta(z-z_s) \quad (4)$$

and

$$p(r,z)=\int_0^\infty p(k_r,z)J_0(k_r r)k_r\,dk_r \quad (5)$$

where k_r is a separation constant that turns out to be the horizontal component of the wave vector, and $p(k_r,z)$ is the solution of (4) for a given k_r. A plot of $p(k_r,z)$ against k_r for a particular depth is known as the horizontal wavenumber spectrum, and if $A=-1/2\pi$ it is called the depth dependent Greens function. A is a constant that depends on the amplitude of the source, δ is the Dirac delta function, and J_0 is the Bessel function of the first kind of zero order.

There are two ways of solving these equations. The first is numerical solution of (4) for various values of k_r, followed by numerical integration of (5). This is the wavenumber integration method used by SCOOTER/FIELDS. SCOOTER solves (4), and FIELDS carries out the integration in (5) using an approximation, valid more than a few wavelengths from the source, that allows (5) to be evaluated by a Fast Fourier Transform. This is known as the fast-field method. The result is an accurate and reasonably efficient method of solving range-independent problems that can deal with seabeds comprising arbitrary fluid and elastic layers. The second method is to find the normal modes of (4). These are the values of $k_r = k_{rm}$ that result in non-trivial solutions of the homogeneous form of (4), i.e. (4) with a right hand side of zero, and correspond to the poles of the horizontal wavenumber spectrum. The corresponding solutions to (4), $\Psi_m(z) = p(k_{rm}, z)$, are known as mode functions. Under these conditions the integration in (5) reduces to a summation over the products of the mode functions evaluated at the source and receiver depths:

$$p(r,z) = \frac{i}{4\rho(z_s)} \sum_{m=1}^{\infty} \Psi_m(z_s) \Psi_m(z) H_0^{(1)}(k_{rm} r) \tag{6}$$

where $H_0^{(1)}$ is the Hankel function of the first kind of order zero. This results in an extremely efficient method of solution of range-independent problems because the mode functions only have to be evaluated once, and then the pressure can be evaluated at any range by using a simple summation. The main disadvantage of this method is that the normal modes only account for energy that is trapped within the waveguide by total internal reflection, so mode solutions tend to be inaccurate at short range where energy that is only partially reflected from the seabed is significant.

KRAKEN finds the normal modes and mode functions by looking for real values of k_{rm}, whereas KRAKENC searches for the required values of k_{rm} in the complex plane. The search along the real axis is a much easier numerical problem than the complex plane search, but limits the environments that KRAKEN can deal with to those without elastic seabed layers (an elastic halfspace is OK though). The complex plane search carried out by KRAKENC allows it to deal with elastic layers and also to account for some of the partially reflected energy through what are called leaky modes. Leaky modes correspond to values of k_{rm} that lie off the real axis even in the absence of seabed attenuation. However, KRAKENC doesn't always find all of the modes, which can lead to significant errors. The mode functions computed by either of these programs can be used by FIELD, which evaluates the pressure using (6).

It is also possible to extend the normal mode method to range-dependent problems by evaluating the modes at different ranges and then using one of two methods for determining how the mode amplitudes at one range depend on those at previous ranges. The adiabatic mode method assumes energy remains in a given mode, whereas the more computationally intensive coupled mode method allows energy to couple between modes. Range-dependent normal modes are not currently supported by AcTUP.

Ray and beam methods

These methods are based on assuming the following form for the pressure:

$$p(\mathbf{r}) = B(\mathbf{r}) e^{i\omega\tau(\mathbf{r})} \tag{7}$$

where $B(\mathbf{r})$ and $\tau(\mathbf{r})$ are slowly varying functions of position along a particular ray. This can be shown to be a high frequency assumption. Substitution into (3) and considerable mathematical manipulation (see Jensen et. al. 2000 for details), leads to the following vector equation, which can be numerically integrated to obtain the ray paths:

$$\frac{d}{ds}\left(\frac{1}{c}\frac{d\mathbf{r}}{ds}\right) = -\frac{1}{c^2}\nabla c \tag{8}$$

Here s is the distance along the ray.

In the ray approach the amplitude of the signal is determined by considering changes in the area delineated by adjacent rays, however this leads to a prediction of infinite amplitude when rays cross (i.e. at caustics), and zero amplitude in regions where no rays penetrate (shadow zones).

The beam approach, on the other hand, considers each ray to be the centre of a beam with a (usually) Gaussian intensity profile and computes the signal by summing the contributions of all beams with significant amplitude at the receiver location. Beam codes reduce problems with caustics and shadow zones and can be used at lower frequencies than ray codes. However, they are still inherently high frequency codes.

BELLHOP is a beam code of this type. It can be run using either beams with a Gaussian intensity profile, or beams with a triangular profile that simulate a standard ray code. BELLHOP requires a bottom reflection coefficient file, which AcTUP generates automatically by running another program from the Acoustic Toolbox, BOUNCE, which calculates the plane wave reflection coefficient of a layered seabed.

BELLHOP can also read in a bathymetry file that specifies the water depth as a function of range, which allows it to handle some range dependence.

Parabolic equation

The parabolic equation (PE) method assumes a solution in the form of an outgoing cylindrical wave:

$$p(\mathbf{r}) = \psi(r,z) H_0^{(1)}(k_0 r) \tag{9}$$

where $\psi(r,z)$ is assumed to be a slowly varying function of range, and k_0 is a reference wavenumber. Substituting into the homogeneous (zero r.h.s.) form of (3) and making the further assumption that energy is propagating at small angles to the horizontal (the paraxial approximation) leads to the standard parabolic equation:

$$2ik_0 \frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial z^2} + k_0^2 \left(\left[\frac{c_0}{c} \right]^2 - 1 \right) \psi = 0 \tag{10}$$

where c_0 is the reference sound speed corresponding to k_0. This equation is only first order in range and, given the field at one range, can be readily integrated to obtain the field at a subsequent range. This leads to an efficient "range marching" algorithm. Although (10) is only accurate for propagation at angles close to the horizontal, recent developments have led to PE algorithms that relax this restriction to the point where it is no longer an impediment.

The Range-dependent Acoustic Model (RAM) developed by Mike Collins (Collins 1993) implements an extremely efficient algorithm that allows a trade-off between the range of propagation angles that can be accurately modelled, and computation speed. This trade-off is implemented through the user specifying the number of terms to use in a Padé expansion that appears in the equations. More terms give accuracy over a wider range of angles but result in longer computation times.

RAM comes in several variants:

- RAM - fluid seabeds, seabed layering specified relative to the water surface. (Strata are horizontal.)
- RAMGeo - fluid seabeds, seabed layering specified relative to the seabed. (Strata follow bathymetry.)
- RAMS - As for RAM but can handle elastic seabeds.
- RAMSurf - as for RAMGeo, but also inputs a file specifying the height of the top boundary of the water column as a function of range.

RAMGeo is more consistent than RAM with the way in which BELLHOP handles range-dependent bathymetry, and was therefore selected for use with AcTUP. The version supplied with AcTUP has been modified by CMST to produce an additional output file that AcTUP can convert to the standard "Shade file" format used by the acoustic toolbox routines.

To allow for modelling situations where shear wave propagation in the seabed is important, CMST developed a modified version of RAMS that uses the same seabed layer specification method as RAMGeo. This is known as RAMSGeo. This program required significant modifications to be made to the original code and, although it has been tested successfully on a number of standard problems, should be used with a certain degree of caution.

ACOUSTIC TOOLBOX USER INTERFACE AND POST-PROCESSOR (ACTUP)

Principle of operation

A simplified block diagram of the way AcTUP works is shown in Figure 1.

The environment and code parameter editor allows the user to load, modify, and save the environment specification (sound speed profiles, seabed characteristics etc.) and other parameters relating to the propagation calculation (source and receiver depths, receiver ranges etc.). This block also allows the user to specify the propagation code that will be run, and any parameters that are specific to that propagation code. All of this information is saved in the run definition file.

The code engine uses the information in the run definition to write the input file that is required by the propagation code that it has been told to run. In some cases the run definition will require several codes to be run in sequence (eg. BOUNCE then BELLHOP, or KRAKEN followed by FIELD), in which case several input files will be required.

The next step is for the code engine to run the propagation codes in the required sequence. The codes are compiled .exe files and are run using Matlab's ! command. The propagation code(s) will produce one or more output files.

The post processor includes an extensive set of routines for reading the output files and plotting the results.

User interface

AcTUP is menu driven, and most parameters are entered via dialog boxes. The exceptions to this are sound speed profiles, which can either be entered into a dialog box or read from a file, and bathymetry, which is always read from a file.

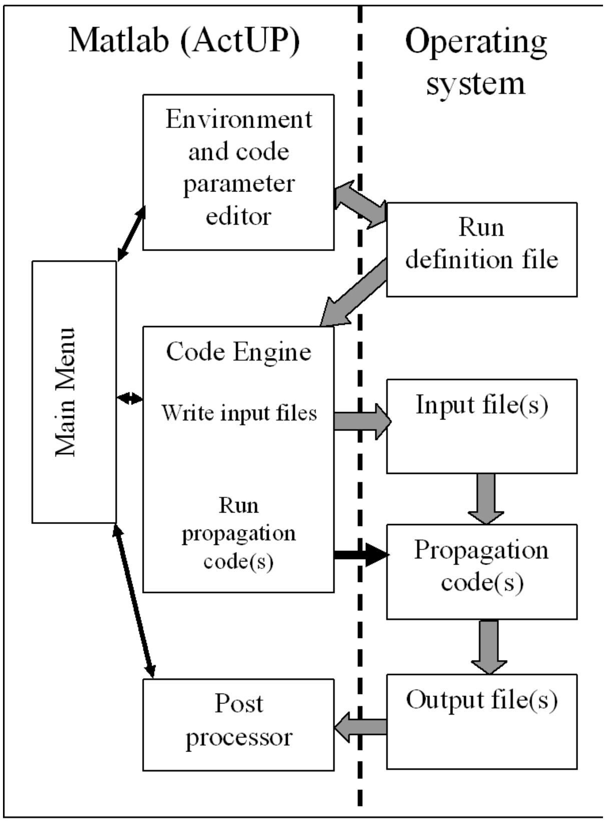

Figure 1. Simplified block diagram of the operation of AcTUP

Care has been taken to use detailed prompts in an attempt to make the operation of the program as intuitive as possible.

The first item on the main menu shown in Figure 2 provides access to the environment and code parameter editor. The editor has changed substantially since AcTUI V1.6, partly to support range-dependent environments and partly to logically separate editing of the environment (sound speed profiles, water depth and seabed geoacoustic properties) from editing of code-independent propagation run parameters (source depth, receiver depths, ranges, etc.). Code-dependent parameters are now also specified by way of this editor, rather than at run-time as was the case in the earlier version. All parameters are now saved as part of the run definition, allowing much improved reproducibility and documentation of runs compared to the earlier version, which didn't save the code-dependent parameters.

Where multiple parameter values are allowed they can be specified using any valid Matlab vector specification. For example, specifying the frequency as 10:2.5:500 would run the propagation code(s) at 2.5 Hz intervals from 10 Hz to 500 Hz. The same method can be used to specify receiver depths.

Fully range-dependent environments are handled through the introduction of an environment array, which is simply a number of environment specifications, each corresponding to a particular range. The environment array editor window is shown in Figure 3 and includes a list of currently specified environments and controls for editing, inserting, deleting and moving individual environments. Three environments have been specified in the example shown here, corresponding to ranges of 0m, 1000m, and 5000m. AcTUP can automatically linearly interpolate between these environments and output interpolated water column and seabed sound speed profiles to the propagation code. This can significantly improve the accuracy of results calculated with the RAM variants because they assume step changes in water column and seabed properties at the ranges at which they are specified. In the example shown here the profiles will be interpolated and output to the propagation code at a 100m range interval.

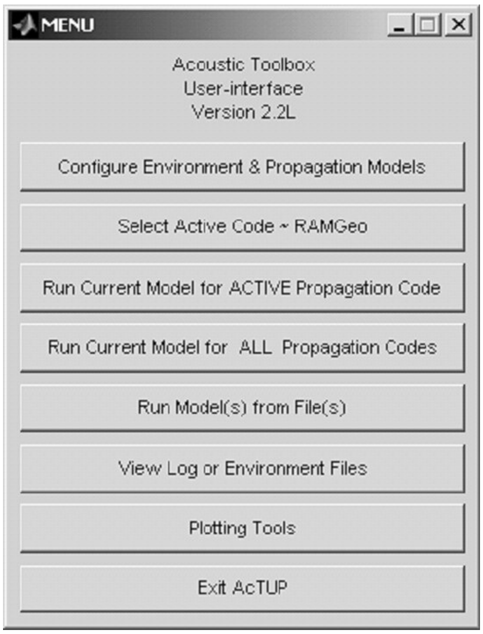
MENU
Acoustic Toolbox
User-interface
Version 2.2L
Configure Environment & Propagation Models
Select Active Code ~ RAMGeo
Run Current Model for ACTIVE Propagation Code
Run Current Model for ALL Propagation Codes
Run Model(s) from File(s)
View Log or Environment Files
Plotting Tools
Exit AcTUP

Figure 2. Main menu

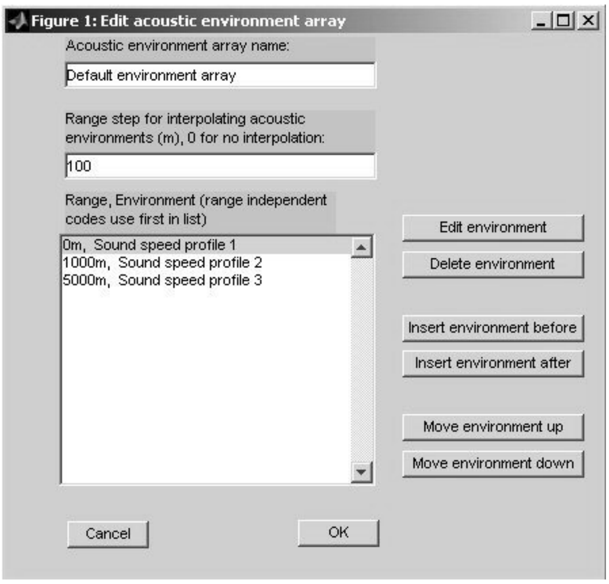
Figure 1: Edit acoustic environment array

Acoustic environment array name: Default environment array

Range step for interpolating acoustic environments (m), 0 for no interpolation: 100

Range, Environment (range independent codes use first in list)

0m, Sound speed profile 1
1000m, Sound speed profile 2
5000m, Sound speed profile 3

Figure 3. Environment array editor window

Individual environments are specified as a series of layers in the same way as in earlier versions. An example of the environment editor window is shown in Figure 4, and has a very similar layout and set of controls to the environment array editor window. The properties of each layer are specified as a function of depth from the top of that layer. The water depth is taken from a bathymetry file if one is specified, otherwise the maximum depth given in the first layer is used.

The main menu shown in Figure 2 includes three run options that utilise the code engine shown in the block diagram in Figure 1. The "Run Current model for ACTIVE Propagation Code" option is the most commonly used and runs the currently specified propagation code using the parameters in the current run definition. The "Run Current Model for ALL Propagation Codes" option runs all propagation codes using the parameters in the current run definition and is mainly included for testing the operation of the interface. The "Run Model(s) from File(s)" option runs the models and parameters specified by all the run definitions in a user specified folder. This provides a simple way of setting up batch runs of arbitrary combinations of codes and parameters and is useful, for example, for running a series of time consuming calculations overnight. Note that it is not necessary to use this option if frequency is the only parameter to be varied, as this is handled more simply by specifying a vector of frequencies in the run definition.

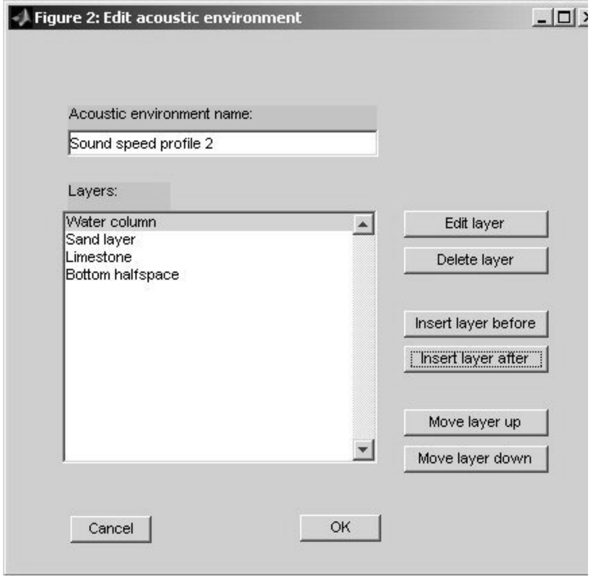
Figure 2: Edit acoustic environment

Acoustic environment name: Sound speed profile 2

Layers:

Water column
Sand layer
Limestone
Bottom halfspace

Figure 4. Environment editor menu

AcTUP includes an extensive set of plotting tools, which are accessed via the plotting tools menu shown in Figure 5. These tools provide a simple and very flexible way of visualising the results of propagation code calculations, including transmission loss plots and plots of intermediate results such as mode shapes, depth-dependent Greens functions, and seabed reflection coefficients.

EXAMPLE

AcTUP can be used for a wide variety of acoustic propagation problems, so the choice of an example to demonstrate its capabilities is rather arbitrary. The example chosen for this paper is based on the sound speed field and bathymetry shown in Figure 6. The sound speed field corresponds to a notional warm core eddy in the upper 500m of the water column.

The transmission loss plots shown in figures 7 and 8 were calculated using RAMGeo for a frequency of 60 Hz, a source depth of 100m and a fluid seabed with a sound speed of 1749 m.s^{-1}, a density of 1941 kg.m^{-3} and no absorption. Figure 7 was calculated for the range-independent sound speed by applying the 0m range sound speed profile at all ranges. Figure 8 is the fully range-dependent result. Figure 9 is a ray trace computed using BELLHOP for the range-independent sound speed case (BELLHOP can't deal with range-dependent sound speed).

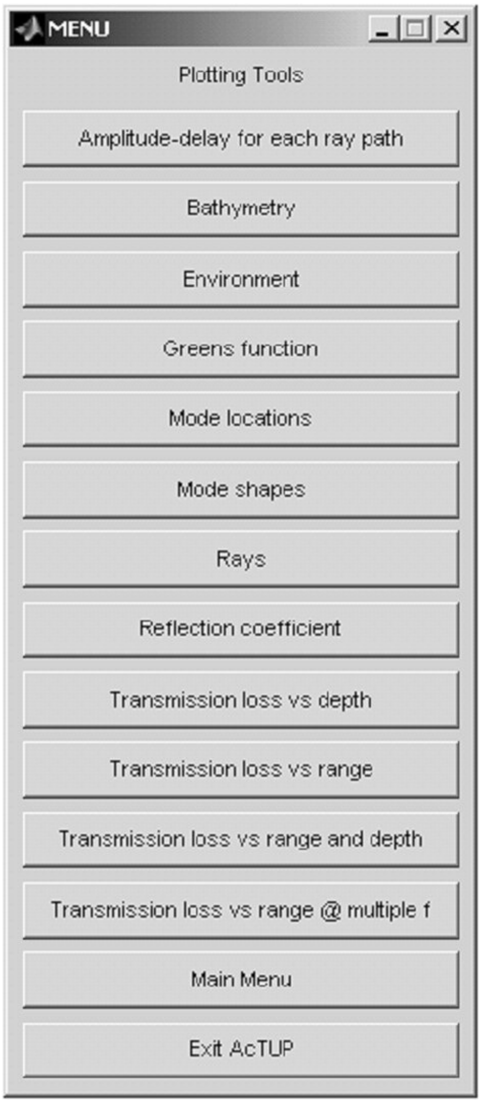
MENU
Plotting Tools
Amplitude-delay for each ray path
Bathymetry
Environment
Greens function
Mode locations
Mode shapes
Rays
Reflection coefficient
Transmission loss vs depth
Transmission loss vs range
Transmission loss vs range and depth
Transmission loss vs range @ multiple f
Main Menu
Exit AcTUP

Figure 5. Post processing menu

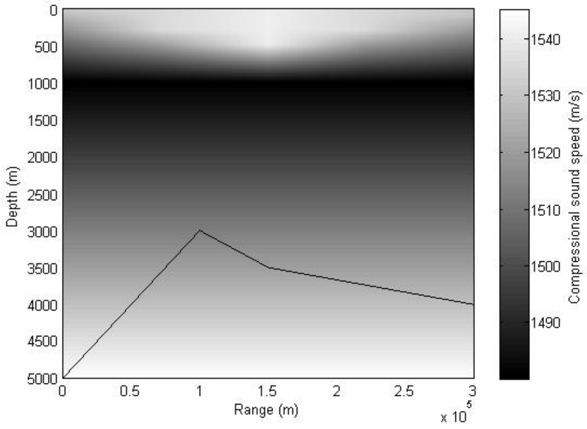

Figure 6. Sound speed and bathymetry for range-dependent example. Shading represents sound speed, line represents bathymetry.

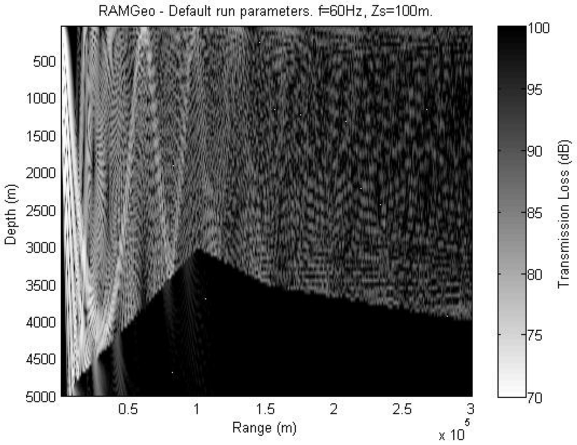

Figure 7. Transmission loss at 60 Hz calculated using RAMGeo for range-independent sound speed profile.

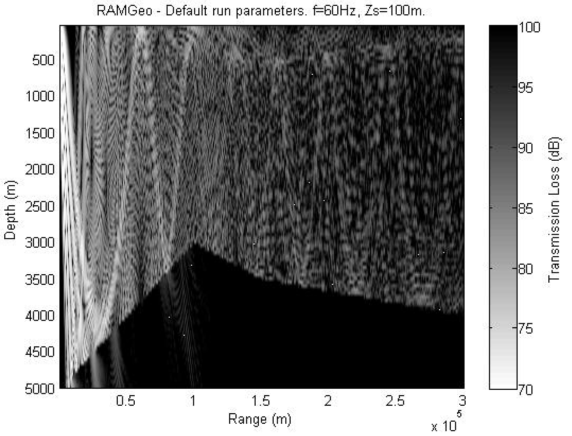

Figure 8. Transmission loss at 60 Hz calculated using RAMGeo for range-dependent sound speed profile shown in Figure 6.

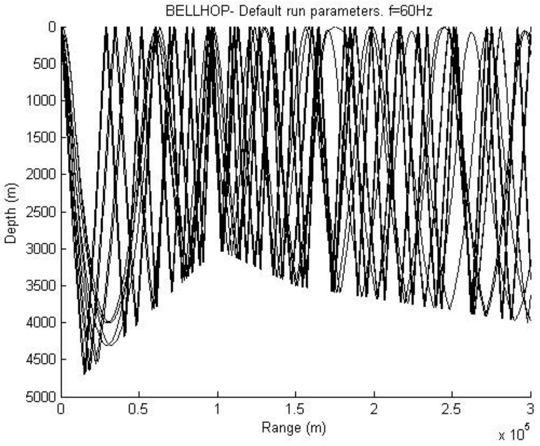

Figure 9. Ray trace computed using BELLHOP for the range-independent sound speed profile.

Comparing figures 7 and 8 shows that the eddy significantly increases the transmission loss in the upper 500 m of the water column for ranges between 140 km and 240 km, but reduces the transmission loss at slightly deeper depths. This is because the upper turning points of the ray paths are depressed by the warmer waters of the eddy. Range-independent codes, and codes that can only handle range-dependent bathymetry, would not show this effect.

CONCLUSIONS

The release of AcTUP V2.2*l* adds fully range-dependent low frequency underwater acoustic propagation modelling to the array of modelling tools provided by its predecessor, AcTUI V1.6. The improved interface greatly simplifies the task of running acoustic propagation models, especially when runs at multiple frequencies are required or where it is desirable to apply several different propagation codes to the same problem.

As was the case with its predecessor, the interface will be extremely useful as an educational tool as it allows students and instructors to quickly and easily run a variety of propagation codes on problems designed to highlight different aspects of the physics of underwater acoustic propagation.

The interface will also be useful for researchers in underwater acoustics, particularly because it provides a quick and easy way of obtaining a first look at a propagation problem. In many cases it will be the only tool required, but it is not a panacea: the desire to make the interface as straightforward as possible for the average user has resulted in some loss of flexibility, and there will be situations where specialist modellers will need to write their own interface code in order to carry out a specific task.

There is a substantial "wish list" of potential improvements to AcTUP that will be implemented over time. Chief of these are:

- Support for a fully range-dependent high frequency code. It is hoped to have a fully range-dependent version of BELLHOP developed and incorporated into the interface early in 2007.
- Support for coupled and adiabatic range-dependent normal mode codes. FIELD has the capability to compute transmission loss in range-dependent environments, by either of these methods, using modes computed by KRAKEN or KRAKENC at a number of different ranges. However, this capability is not supported by AcTUP V2.2*l.*
- Pre-processing functions to read environment data from standard data bases and from standard field data file formats.

AcTUP V2.2*l* can be downloaded from the Centre for Marine Science and technology web site:

http://www.cmst.curtin.edu.au/products/actoolbox/

ACKNOWLEDGEMENTS

The development of AcTUP V2.2*l* would not have been possible without financial support from the Defence Science and Technology Organisation (DSTO).

REFERENCES

Collins, M. D. 1992, *A self-starter for the parabolic equation method*, J. Acoust. Soc. Am., 92 (4), October 1992, pp 2069-2074.

Collins, M. D. 1993, *A split-step Padé solution for the parabolic equation method*, J. Acoust. Soc. Am., 93 (4), April 1993, pp 1736-1742.

Collins, M. D. 1999, *Users Guide for RAM versions 1.0 and 1.0p.* ftp://ftp.ccs.nrl.navy.mil/pub/ram/RAM/ram.ps

Jensen, F. B., Kuperman, W. A., Porter, M. B., and Schmidt, H. 2000, *Computational Ocean Acoustics*, Springer-Verlag, New York, ISBN 1-56396-209-8.

Porter, M. B., 2005, *Acoustic Toolbox*, http://www.hlsresearch.com/oalib/Modes/AcousticsToolbox/