

# Curtin

UNIVERSITY OF TECHNOLOGY

---

Centre for Marine Science  
& Technology

AcTUP v2.2 $\ell$  <sup>$\alpha$</sup>

Acoustic Toolbox  
User-interface & Post-processor

Installation & User Guide

Amos L Maggi & Alec J Duncan

<sup>$\alpha$</sup>  the  $\ell$  designation indicates the limited or "lite" release of AcTUP

# Revision Sheet

<b>Organisation</b>	<b>Centre for Marine Science and Technology</b>
<b>Project Number</b>	
<b>Project Name</b>	<b>AcTUP Development (on-going)</b>
<b>Document Number</b>	
<b>Document Name</b>	<b>AcTUP – Installation &amp; User Guide</b>

<b>Revision No.</b>	<b>Issue Notes</b>	<b>Written By:</b>	<b>Checked By:</b>	<b>Approved By:</b>	<b>Date Issued</b>
A (L)	First draft	ALM & AJD	AJD		

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>5</b>
1.1	BACKGROUND.....	5
1.2	DOCUMENT DESCRIPTION.....	5
1.3	INTENDED AUDIENCE.....	5
<b>2</b>	<b>WHAT'S NEW.....</b>	<b>6</b>
<b>3</b>	<b>INSTALLING &amp; LAUNCHING ACTUP.....</b>	<b>7</b>
3.1	PREREQUISITES.....	7
3.1.1	<i>MATLAB</i> .....	7
3.1.2	<i>AT</i> .....	7
3.2	DOWNLOADING FILES.....	7
3.3	INSTALLING FILES.....	7
3.4	SETTING UP ACTUP.....	8
3.4.1	<i>if you used some other target directory</i> .....	8
3.5	RUNNING ACTUP.....	8
3.5.1	<i>end user</i> .....	8
3.5.2	<i>developer</i> .....	8
<b>4</b>	<b>USING ACTUP V2.2L.....</b>	<b>9</b>
4.1	MAIN MENU.....	9
4.1.1	<i>Configure Environment &amp; Propagation Models</i> .....	9
4.1.2	<i>Select Active Code ~ &lt;code&gt;</i> .....	9
4.1.3	<i>Run Current Model for ACTIVE Propagation code</i> .....	9
4.1.4	<i>Run Current Model for ALL Propagation codes</i> .....	9
4.1.5	<i>Run Model(s) from File(s)</i> .....	9
4.1.6	<i>View Log or Environment Files</i> .....	10
4.2	ENVIRONMENT & PROPAGATION DEFINITIONS.....	10
4.2.1	<i>Load DEFAULT Run Definition</i> .....	10
4.2.2	<i>Load Run Definition</i> .....	10
4.2.3	<i>Save Run Definition</i> .....	10
4.2.4	<i>Edit Environment</i> .....	11
4.2.5	<i>Edit Code-Independent Propagation Parameters</i> .....	11
4.2.6	<i>Edit Code-Dependent Propagation Parameters</i> .....	11
4.2.7	<i>Select Active Code ~ &lt;code&gt;</i> .....	11
4.3	PLOTTING TOOLS.....	12
4.3.1	<i>Amplitude-delay for each ray path</i> .....	12
4.3.2	<i>Bathymetry</i> .....	12
4.3.3	<i>Environment</i> .....	12
4.3.4	<i>Green's function</i> .....	12
4.3.5	<i>Mode locations</i> .....	12
4.3.6	<i>Mode shapes</i> .....	12
4.3.7	<i>Rays</i> .....	12
4.3.8	<i>Reflection coefficient</i> .....	12
4.3.9	<i>All Transmission loss plots</i> .....	12
<b>5</b>	<b>ACOUSTIC PROPAGATION CODES.....</b>	<b>13</b>
5.1	ACOUSTIC TOOLBOX (AT) CODES.....	13
5.1.1	<i>Bounce &amp; Bellhop</i> .....	13
5.1.2	<i>Kraken, KrakenC &amp; Field</i> .....	13
5.1.3	<i>Scooter &amp; Fields</i> .....	14
5.2	RAM.....	15
5.2.1	<i>RAMGeo (v1.5C00.03)</i> .....	15
5.2.2	<i>RAMSGeo (v0.5C01.01.01)</i> .....	15
5.2.3	<i>Additional modification and data handling</i> .....	15
<b>6</b>	<b>DATA PRODUCTS.....</b>	<b>16</b>
6.1	OUTPUT FILES.....	16

## **SUMMARY**

This document, Installation & User Guide, is a collection of notes intended to accelerate the new user's familiarisation with AcTUP. It is not intended as an exhaustive user manual.

Work carried out by:

**Amos L Maggi & Alec J Duncan**

# 1 introduction

## 1.1 background

AcTUP was initially developed as the “Acoustics Toolbox Front-End” by Alec Duncan to facilitate the rapid application of, and performance comparisons between, the different acoustic propagation codes available in Mike Porter’s Acoustic Toolbox (AT).

This was achieved by providing the user with a unified environment model, automating the creation of code-specific initialisation files and by providing intelligent guesses and selection guidelines for code-specific parameters.

The cost of this kind of streamlining is some loss of generality, however users familiar with the individual propagation codes will find the ability to directly edit initialisation files from AcTUP goes some way to addressing this issue.

Since 2002, the software has steadily grown to include two non-“AT” propagation codes and a number of post-processing tools. The development of AcTUP from version 1.6 to the current version was largely funded by the Defence Science & Technology Organisation (DSTO).

## 1.2 document description

This Installation & User Guide is not intended as a complete user manual.

AcTUP is research level code and a little clunky at times but the menu driven format makes it mostly self-explanatory.

This guide provides:

- installation instructions;
- an outline of the steps required to generate principle data products where these are perhaps not obvious;
- limited guidance for parameter selection in order to successfully generate these products;
- warning of potential traps in the interface

## 1.3 intended audience

AcTUP Project Team

All potential AcTUP users – particularly new users

## 2 what's new

ActUP has grown considerably since v1.6 was released. Among the principle changes are:

- a Addition of two modified versions of Mike Collins' range-dependent PE code RAM. (The unmodified source code is available from <ftp://ftp.ccs.nrl.navy.mil/pub/ram/RAM/>).
- b Compatibility with the 2006 release of Mike Porter's Acoustic Toolbox.
- c A redesigned Run Definition class combining model environment and propagation parameters into single structure. This offers increased robustness, facilitating full documentation and recovery of all model parameters.
- d Ability to set up batch runs of one or more propagation codes using the same user friendly interface used for single runs.
- e A redesigned and more connected user interface to cope with the significantly expanded functionality.

## 3 installing & launching ActUP

### 3.1 prerequisites

#### 3.1.1 MATLAB

ActUP v2.2L requires only MATLAB version 6 or later. In addition to this the full release specification requires the MATLAB Signal Processing Toolbox

Note that version 2.2 was developed using MATLAB R2000a.

#### 3.1.2 AT

Mike Porter's Acoustic Toolbox and versions of RAM modified to work with ActUP are supplied with ActUP v2.2(L) to simplify installation procedure.

### 3.2 downloading files

All the files you require for installing ActUP v2.2L are available from the Centre for Marine Science & Technology website @

<http://www.cmst.curtin.edu.au/products/acttoolbox/index.html>

The appropriate link will allow you to download the following self-extracting zip file:

**ActUP v2.2 lite.exe**

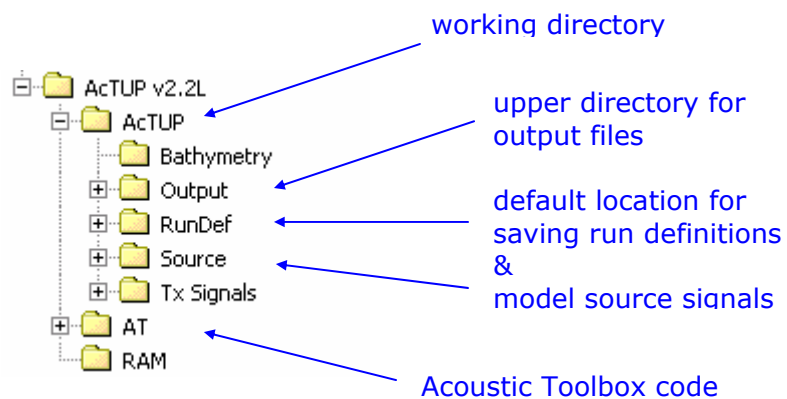
### 3.3 installing files

To install ActUP v2.2(L) simply run the executable above.

After extract the topmost ActUP directory will locate directly below the default target directory as shown in the figure below:



Below the target directory, your directory structure (no matter what target was selected) will look like:



You should AVOID editing or deleting installed directory names.

### 3.4 setting up ActUP

You only need to complete the following instructions if you DID NOT use the default target directory in the installation process.

#### 3.4.1 if you used some other target directory

Navigate to the folder

```
C:\Program Files\CMST Software\ActUP v2.2L\ActUP\Source
```

Open

```
GetAcDirectoryInfo.m
```

Edit the line ...

```
Top = 'C:\Program Files\CMST Software\ActUP v2.2L\';
```

... to reflect the correct location of the ActUP v2.2L directory.

Save and close this file.

Now point MATLAB at the working directory by

### 3.5 running ActUP

#### 3.5.1 end user

In order to run ActUP you must point MATLAB at the working directory ActUP. You must then run the top level m file `actup.m`.

The example below is for the default installation case.

```
>> cd('C:\Program Files\CMST Software\ActUP v2.2L\ActUP');  
>> actup
```

Note that if ActUP is used often the working directory will probably spend a lot of time in the current directory drop box at the top of the command window for rapid selection.

Alternatively, the two lines above can be inserted in to a MATLAB script or function somewhere on the MATLAB path allowing the `cd` and `launch` command to be executed in one line.

Note that it is not sufficient (and not recommended) that you place the working directory permanently on the MATLAB path.

#### 3.5.2 developer

This is NOT recommended for driving the application but while developing ActUP, it is more convenient to run from the source subdirectory.

```
>> cd('C:\Program Files\CMST Software\ActUP v2.2L\ActUP\Source');  
>> act
```



## 4 using ActUP v2.2f

The menu driven system is largely self-explanatory but some introductory notes are provided below to kick-start the familiarisation process and to help the user avoid pitfalls.

### 4.1 Main Menu

#### 4.1.1 Configure Environment & Propagation Models

Launches another menu which allows the user to define the model environment, set basic propagation parameters, to set parameters specific to the available propagation codes, to save all environment and propagation model parameters as a single Run Definition, and to load previously saved Run Definitions.

#### 4.1.2 Select Active Code ~ <code>

Selects which of the available codes is used in modelling propagation when the button below this one is selected (see 4.1.3).

#### 4.1.3 Run Current Model for ACTIVE Propagation code.

Runs the environment/propagation model currently in memory using the selected propagation code (see 4.1.2). Note ActUP does not check whether the code-specific parameters have been set for the selected propagation code.

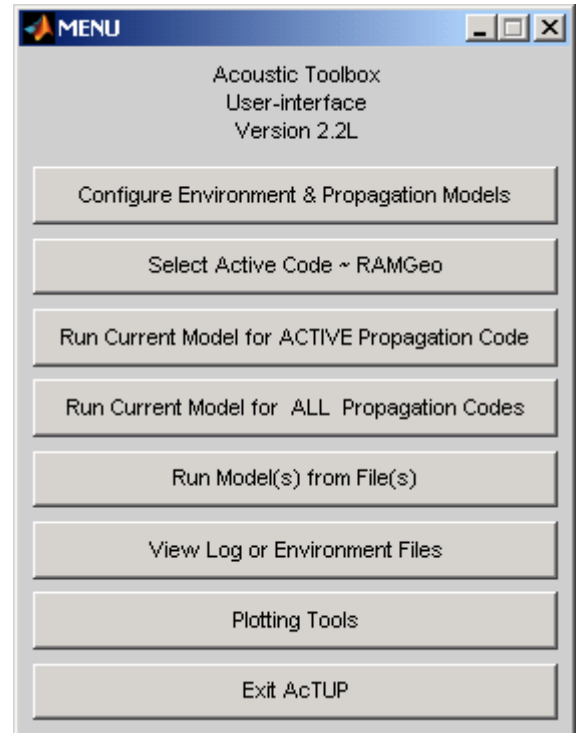
#### 4.1.4 Run Current Model for ALL Propagation codes.

Runs the environment/propagation model currently in memory using all available propagation codes. Note that in this mode Bounce and Fields are only run in conjunction with Bellhop and Scooter respectively.

Note ActUP does not check whether the all code-specific parameters have been set. This would be rather difficult to do anyway since Run Definitions are rarely built from scratch. They are often built from previously saved Definitions. It is impractical and perhaps impossible to track environment or propagation model edit and from these determine algorithmically whether previously defined code-specific parameters are still appropriate.

#### 4.1.5 Run Model(s) from File(s)

This options enables batch mode processing. The user is prompted to select a directory containing one or more Run Definitions. ActUP will sequentially load and run each Run Definition file. For each Run Definition file, the propagation code applied is which was active (4.1.2) at the time it was last saved to disk. Currently only one active propagation code is selectable at any one time so that in order to run multiple propagation codes for the same Run Definition from file, the user must save a copy of same the Run Definition for each propagation code required.



#### 4.1.6 View Log or Environment Files.

Allows user to open and view log (.prn) and environment definition (.env) files.

The log files are text files which contain a log of the command line instructions issued by ActUP and any errors generated by them. (Note in some cases MATLAB commands `copyfile` and `movefile` have been used to replace command line `copy` and `move` and so do not generate entries in the log file).

The environment files are somewhat of a misnomer and an artefact from early versions of ActUP.

These are in fact (renamed) copies of the configuration files accessed by the propagation codes and contain propagation model parameters as well as environment model parameters.

Additionally, .env does not describe a single format. One of the motivations for ActUP is that each propagation code uses quite data and data formats. These are written to the .env files and then copied to the standard configuration file name for the active propagation code just prior to execution. The .env files are stored with the other output

## 4.2 Environment & Propagation Definitions

### 4.2.1 Load DEFAULT Run Definition

The default configuration has preloaded code-specific parameters for each available propagation code, and can be run as a first test after installation.

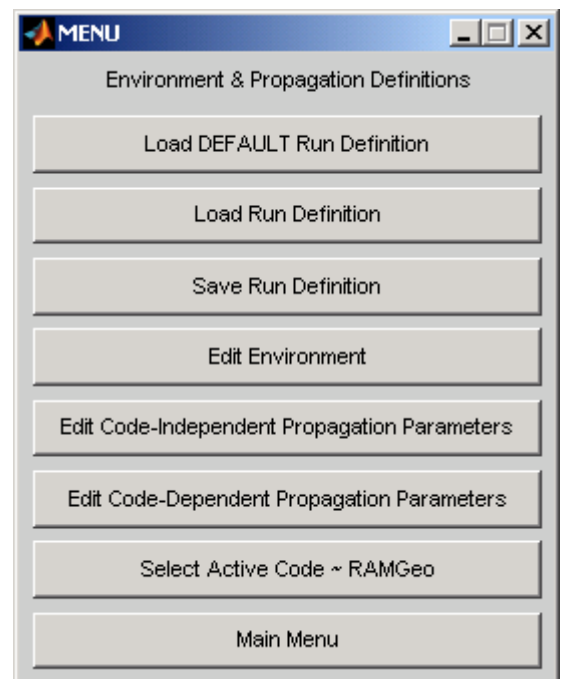
### 4.2.2 Load Run Definition

To date there is no specific filename extension identifying Run Definition objects and instead we use the default .mat extension denoting the format in which the Run Definition is saved. This and the way environment and propagation models are handled may change in future versions.

Note that backward compatibility has been maintained. ActUP 2.2 will seamlessly load Run Definitions saved by previous versions then create and fill missing fields with default values. While the converted Run Definition may run with previously unavailable code options, results may not be satisfactory.

### 4.2.3 Save Run Definition

While you may use a different extension from .mat it is advised that you stick with this format. If you save both input and output files in an areas designated for ActUP modelling then there is no need to distinguish these .mat files from any others since they are the only ones used by ActUP.



#### 4.2.4 Edit Environment

This launches an interface to edit the model environment. This consists of an array of range-independent Environment objects, each defined at a specific range. The range-independent codes (all codes, except RAM\* in this version) only use the first element of the Environment array.

Each Environment object is a collection of range independent layers of acoustic media. The first layer is always the water column and the second layer is always the upper-most layer of the substrate. Each layer consists of lists of depth- (z-) dependent properties which must be specified.

Bathymetry is currently specified in the propagation model. It is used to specify the depth at which the upper interface of the 2<sup>nd</sup> acoustic medium layer occurs relative to the upper surface of the first (water) layer.

Currently the Environment object array is limited in such a way that each component Environment object must contain the same number of layers.

#### 4.2.5 Edit Code-Independent Propagation Parameters

Basic parameters defining the propagation scenario are specified here. Along with the environmental model, these parameters form a basis for configuring runs by the different propagation codes, and in some cases (e.g. Bounce, Kraken, Scooter) completely specify the run.

#### 4.2.6 Edit Code-Dependent Propagation Parameters

Information additional to that provided above is required for some codes (i.e. Bellhop, KrakenC, RAMGeo and RAMSGeo). Here the user can set appropriate parameters values for each code. Note that most of the values for RAMGeo and RAMSGeo are common and editing a parameter in one of these effectively edits that parameter for both codes. (The interface design has been compromised for the sake of development time and contains artefacts of earlier [unreleased] versions. Some streamlining may occur in subsequent versions).

Note that when a Run Definition is saved, all the code-dependent parameters are saved with it irrespective of which code is selected as active.

#### 4.2.7 Select Active Code ~ <code>

See 4.1.2.

### 4.3 Plotting Tools

These are fairly self explanatory and the following is merely a brief description of which data files are accessed and which codes produce those files.

#### 4.3.1 Amplitude-delay for each ray path.

Plot the data from .arr files output by Bellhop in "A" mode

#### 4.3.2 Bathymetry.

Plots bathymetry data from bathymetry files in the input format required by ActUP

ascii format:

row #	format	comments
1	%d	number of points
2 to N+1	%f %f	r [km] z [m]

#### 4.3.3 Environment

Plots water and substrate physical and acoustic parameter profiles for the currently loaded environment model.

#### 4.3.4 Green's function.

Plots the results generated by Scooter (.grn files).

#### 4.3.5 Mode locations.

Uses output from Kraken & KrakenC.

#### 4.3.6 Mode shapes

Uses output from Kraken & KrakenC.

#### 4.3.7 Rays

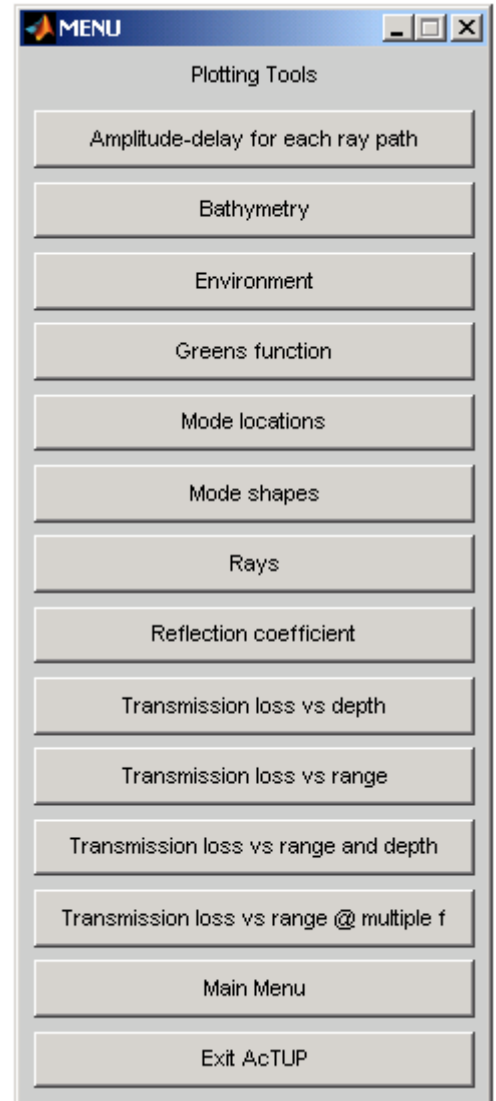
Plots ray paths generated by Bellhop in "R" mode (.ray files)

#### 4.3.8 Reflection coefficient

Plot the data from .brc files output by Bounce.

#### 4.3.9 All Transmission loss plots

Plots output from .shd files generated by Bellhop (C, S & I modes), Kraken, KrakenC, Fields, and RAM\* (see 5.2).



## 5 Acoustic Propagation Codes

### 5.1 Acoustic Toolbox (AT) Codes

The following components of Mike Porter's Acoustic Toolbox are supported and directly available from HLS Research.

The version supplied and tested with ActUP v2.2 is that released by Mike Porter in early 2006.

For further details, please refer to the HLS Research site:

<http://www.hlsresearch.com/oalib/Modes/AcousticsToolbox/>

For an excellent reference that covers the theoretical background of all the propagation codes used by ActUP, see:

Jensen et al. (1994) *Computational Ocean Acoustics*, AIP Press, New York.

#### 5.1.1 Bounce & Bellhop

Bounce calculates reflection coefficients at the lower water column interface. In ActUP it can be run independently or in tandem with Bellhop which uses the reflection coefficient output to model either:

- Beam/Ray paths
- Amplitude and Delay of arrivals at specified field points, or
- Coherent, semicoherent or incoherent transmission loss

Bellhop is a beam-tracing program that can include range dependent bathymetry (but not sound speed). Beam tracing is similar in principle to ray tracing but traces the paths of finite width beams rather than infinitesimal width rays. This reduces problems caused by ray theory artefacts such as caustics and shadow zones. Bellhop can use beams with a Gaussian intensity profile, or geometric beams which produce the same result as a standard ray trace. Bellhop is inherently a high frequency code, and although its useful frequency range extends lower than standard ray trace programs, it should be used with extreme caution in situations where the water depth or the size of any significant feature in the sound speed profile is less than 20 wavelengths.

#### 5.1.2 Kraken, KrakenC & Field

Kraken finds the normal modes for the model propagation environment using real arithmetic and estimates the attenuation by a perturbation technique. This method works well for layered fluid seabeds and can handle an elastic lower halfspace (i.e. a halfspace with a significant shear speed). It can't cope with elastic intermediate layers. The normal modes only account for energy that is trapped in the waveguide and therefore this method is inaccurate at short range where the effect of untrapped energy is significant.

KrakenC finds the normal modes in the complex wavenumber plane, which allows it to deal with elastic seabed layers, and to include the effects of leaky modes, which account for some of the untrapped energy. This makes it more accurate than Kraken at short range. However, finding modes in the complex plane is much more difficult than finding them on the real axis which makes KrakenC prone to missing some of the modes, leading to inaccurate results. Use with caution!

Field uses the normal modes calculated by either of these programs to determine the transmission loss at the specified field points.

ActUP automatically runs Field immediately following either Kraken or KrakenC. This is a very efficient way of calculating low frequency transmission loss at long

ranges, as the modes only have to be calculated once, after which the transmission loss can easily be computed at any desired range. However, the number of modes increases with increasing frequency, making normal mode methods unsuitable for modelling propagation in situations where the waveguide is many acoustic wavelengths deep.

In future versions Field may be offered as a separate option to make use of existing mode files.

Range dependent problems can be handled by computing the normal modes at different ranges and then using adiabatic or coupled mode theory to calculate how the modes at one range couple to the modes at the next range. This capability is not currently supported by ActUP

### 5.1.3 Scooter & Fields

Scooter calculates the depth-dependent Green's function for the model environment.

Fields (not Field) integrates the resultant Green's function results (in .grn files) to determine the transmission loss by using an FFT. This is the fast field technique.

This method of calculating transmission loss accounts for all the acoustic energy, whether trapped or not, and does not rely on the less than 100% reliable process of finding normal modes. It can also cope quite happily with layered elastic seabeds, and even model boundary interface waves. It is therefore both more accurate and more reliable than the normal mode method and is, far and away, the preferred technique for low frequency, short range, range independent problems. It is, however, much more computationally intensive than normal modes for long-range problems, and the computational load increases rapidly with increasing frequency.

In ActUP Scooter and Fields can be run in sequence automatically (Scooter+Fields) or manually. The later is useful when high-resolution range data  $\Delta r$  is only required about small regions  $\Delta r$  about a number of ranges  $\{r_i\}$  where  $|r_{i+1} - r_i| \gg \Delta r$ , while maintaining workable file sizes.

When the sequence is run automatically, it is done so for each specified frequency in turn. Field uses the .grn file just created by Scooter and the field points specified by the receiver location in the Code-Independent parameter set.

Care needs to be taken when Fields is run manually because Fields needs very little information. In fact it can be run without the same Run Definition (environment + propagation parameters) previously used by Scooter.

When run manually, the user must first launch "Edit Code-Dependent Parameters" for Fields. Here the user specifies a group of target .grn files by choosing one of the group interactively. ActUP will then locate all .grn files in the same folder that match the base filename prefix excluding the frequency tag. The user is then prompted to specify the field point grid for the TL calculation. The suggested parameters appearing on the Fields dialog box will be based on the .grn file contents and the receiver locations in the Code-Independent parameters. It is recommended although not absolutely necessary that the Run Definition used to generate Scooter be loaded for before setting up launching Fields.

## 5.2 RAM

Two modified versions of Mike Collins' Range-dependent Acoustic Model (RAM) have been integrated into the ActUI framework. RAM is a parabolic equation (PE) code that uses a split-step Padé algorithm to achieve high efficiency and the ability to model propagation at large angles from the horizontal (the usual limitation of PE codes). There is a trade-off between the angular range and the speed of computation that is governed by the number of terms the user specifies for the Padé approximation – the more terms, the wider the angle, but the slower the code runs.

RAM is capable of modelling low frequency propagation in fully range dependent environments (i.e. range dependent bathymetry and sound speed), and this capability is fully supported by ActUP.

### 5.2.1 RAMGeo (v1.5C00.03)

This is a CMST version based on Mike Collins' RAMGeo version 1.5.

It has been modified to output complex transmission loss data as well as the standard magnitude only files. Additional files output are:

tl.pgrid      Complex pressure analogue to tl.grid

tl.pline      Complex pressure analogue to tl.line

Additional minor modifications include some error flags which enable ActUI and the user to better identify some run time errors.

### 5.2.2 RAMSGeo (v0.5C01.01.01)

This is a CMST version based on Mike Collins' RAMS version 0.5 and the RAMGeo version discussed in the previous section.

Mike Collins' elastic substrate version of RAM has been modified so that it uses the same (bathymetry datum) substrate profile specification model as RAMGeo.

### 5.2.3 Additional modification and data handling

When these two versions are run under ActUP, the .pgrid files are translated and written to shade format files (.shd) used by the other (Act) models.

Code for conversion and reading shade files in ActUP has been modified to cope with the rare but conceivable case where the (complex) TL record length can exceed twice the length (in number not necessarily bytes) of the (real) range vector record. Generally only a risk in model propagation planes with unusually high aspect ratios.

## 6 Data Products

### 6.1 Output Files

extension	description	format	produced by
.arr	Amplitude and time-of-flight data as well as other ray parameters for arrivals at field point.	ascii	Bellhop
.brc	Bottom reflection coefficient as a function of incident angle	ascii	Bounce
.grn	Depth-dependent Green's function solutions	binary	Scooter
.mod	Mode solutions	binary	Kraken KrakenC
.ray	Ray trace output		Bellhop
.shd	Complex transmission loss as a function of depth and range	binary	Bellhop Field Fields ActUP