QUICK INTRODUCTION TO USING THE NAVAL RESEARCH LAB'S RAM PARABOLIC EQUATION (PE) CODE THAT INCLUDES BOTTOM LOSS

January 11, 2006

Dr. David C. Calvo
calvo@ccs.nrl.navy.mil
Acoustics Division, Code 7145
Naval Research Laboratory
Washington, DC 20375
202-404-4800
ftp site for Acoustics Branch (other codes available):
ftp://ftp.ccs.nrl.navy.mil/pub/ram/RAM

Contents

Two parabolic equation codes are included named RAM and RAMS.
RAM stands for Range-dependent Acoustic Model, RAMS is the same but accounts for shear waves in the ocean bottom. Both codes support propagation into the bottom (bottom loss), but RAM only handles compression waves in the ocean bottom whereas RAMS handles both compression and shear waves in the bottom. Generally, shear waves will lead to more loss and can affect transmission loss in the water column. A third computer code is provided to create a color transmission loss plot (in postscript form) of the outputs of RAM or RAMS. RAMS is more physically accurate than RAM, but is more sensitive to numerical parameters as described at the end of the document. A proper choice of numerical parameters is learned through experience, but the examples provided can be used as a starting point. *It is recommended to make backups of the .f files and the .in files to keep as reference examples. That way one can modify the files that do not have backup in their names and keep the backed up files as reference points.*

Files

Three source code files written in FORTRAN77 need to be compiled:

ram1.5.f (RAM)
rams0.5.f (RAMS)
ramclr.f (CODE TO VIEW OUTPUT IN .PS FORM)

Compiling

(On a unix platform, for example, determine if FORTRAN77 is available by typing man f77 or man f90 or man g77)

compiler strings:
f77 –O3 –o ram ram1.5.f

f77 –O3 -o rams rams0.5.f
f77 –O3 –o ramclr ramclr.f

The –o option names the executable file (the file you invoke when you want to run the code) the name after this flag. The –O3 option optimizes the code to the platform it is being compiled on. The code can run significantly faster this way.

Output Files (produced by RAM and RAMS):

tl.line the transmission loss versus range at a specified receiver depth (ASCII file)
tl.grid the entire range-depth transmission loss field (binary file)

Output Files (produced by ramclr.f)
ramclr.ps (a color post script that displays transmission loss)

Input Files:
ram.in (the input to RAM)
rams.in (the input to RAMS)
tl.grid (the input to ramclr)

Description of input file to RAM

(Note: UNITS ARE SPECIFIED IN **MKS** format i.e. meters/second, meters, g/cm^3 (density))

Row 1: header
Row 2: frequency in Hz, source depth, receiver depth (data at this depth goes into tl.line)
Row 3: maximum range (rmax), range step (delta r or dr), ndr (writes output every ndr range steps)
Row 4: maximum depth of the whole computational domain (zmax), depth step (dz), ndz (output data at every ndz vertical point at a given range), vertical extent of the domain to output in tl.grid (zmplt).

To keep output file sizes down (since ramclr may have problems with big tl.grid files) use ndr and ndz to control size. Also zmplt is maximum depth that will be outputted. This must be less than zmax which is the actual depth of the entire computational domain. Note that the accuracy of the solution will depend on dr and dz and the number of Pade terms (np).

Row 5: reference sound speed, number of Pade terms (np), # of stability constraints (ns), radius of stability constraint (rs)

Note: 1600 m/s should be adequate for first quantity (there is a separate

area to actually put in the sound profile to be used, see below, but 1600 m/s is just a representative sound speed that the code uses). Number of Pade terms controls the accuracy of the computation along with dr and dz. The larger the # of Pade terms, the more accurate the computation but the slower it takes to run. The number of stability constraints should be increased if the code keeps producing garbage results for any combination of np, dr and dz. For radius of stability constraint, leave at 0. More details can be found in the attached *Journal of the Acoustical Society of America* paper by Collins.

Row 6 and onward:
0.0 200 (range=0 bathymetry=200 meters)
40000.0 400.0 (range 40000 meters, bathymetry = 400 meters)

in the above example, RAM will linearly interpolate the bathymetry between 0 and 40000 and then maintain 400 meter depth until rmax = 50000 is reached.
One can create a more continuous bottom profile by adding in more lines between 0 and 40000. See ramclr.ps to view what the example bathymetry looks like. It is indicated by a black line.

Additional input (separated by -1 -1 lines)

Depth	Sound Speed in Water
0	1400.0
100.0	1520.0
400.0	1530.0

(In practice one can take the output of the XBT after computing sound speed and stick it in here. Again, sound speed will be linearly interpolated between values. If a sound speed calculator is available that outputs two ASCII columns, depth and speed, then simply cut and paste into the input file.)

Depth (referenced from bottom)	Sound speed in bottom (user can input profile like above)
0	1700.0

Depth	Density of Bottom
0	1.5 (1.5 times the density of water)

Depth	Attenuation of Bottom (in dB/wavelength)
900	0.5
1000	10.0

Note: the above attenuation (which becomes extremely large at 1000 meters) is used to damp out waves which reach the bottom of the computational domain at zmax. It is a buffer region that starts at 900 meters. In practice one should put in attenuation at bottom referenced depth 0 if a value is known for the bottom. Be cautious with units, some

tables use dB/wavelength (=dB/lambda) but you may only have a value in units of dB/(m kHz). In that case multiply the value in dB/(m kHz) by the speed of sound in its medium and divide by 1000.

Line that says 25000.0 (a new profile at 25000.0)
Input is same pattern as above but is now updated at 25000 meters.

General rules for using RAM:

As a rule of thumb make (dz = dr/10) . You will have to make dr smaller as the frequency increases (relative to the example in provided input files). This means your output file will have more data points so use ndr and ndz to keep file sizes manageable.

Occasionally you will see output that looks like the computation stopped halfway or the data will be blank. This likely means there was an instability in the calculation. The fix is to vary dr until you get a reasonable looking solution. If you can't find a good dr value, try increasing ns above 1.

It is good to refine your grid (make dr and dz smaller) and increase the number of Pade terms so you make sure your solution has converged on a solid answer. Experience will guide you in deciding numerical parameters given a frequency and geoacoustic environment.

Note: you should make zmax about 3 or 4 times the depth of the ocean bottom. Ideally, you should make it bigger if processing time is not a problem.

Guidelines for using RAMS:

RAMS is more physically accurate (includes shear effects) but requires more care in choosing the parameters dr, dz, np (# of Pade terms) and theta (a number varing between 0 and 90 that the code uses to avoid numerical instabilities that cause garbage results).

Also, in the input file you need an extra area for shear wave speed and shear attenuation. Compare the example file inputs to RAM and RAMS to see precisely how those quanitites are entered.

Starting with a test case that you know works, you should increase the frequency and decrease dr. If the output is garbage, start increasing the parameter theta until you have a reasonable looking answer. Then continue to refine dr and dz adjusting theta as necessary.

The larger the number of Pade terms, the more accurate your answer will be, but you'll

find you'll need to increase theta more to avoid getting garbage results. As a general rule, only increase theta to the point that you don't see NaN (not-a-number) flashing by you as the output displays after running the executable.

Guidelines for using ramclr.f

The example postscripts outputs ramclrSHEAR100Hz.ps and ramclrSIMPLEBOTTOM.ps have been created using the following inputs to ramclr.f

1
0
1
25
20
200
45
150
10