PASSIVE SONAR PERFORMANCE ENGINE (SPE)

KD Heaney, OASIS Inc., Fairfax, Virginia, USA
RL Campbell, OASIS Inc., Fairfax, Virginia, USA

1 INTRODUCTION

This paper describes recent advances in computational acoustic forward modeling, via the Parabolic Equation (PE), and their inclusion into an integrated passive performance prediction algorithm. The industry standard PE implementation, RAM[1-3], was ported to C as a re-entrant subroutine and integrated with environmental input/output routines. The new PE code, CRAM, permits the computation of the full Nx2D field for an arbitrary number of sources (receivers via reciprocity) within a single execution call. With the CRAM code as a single C routine, multi-processor execution can be easily handled via compilation with any compiler supporting OpenMP. The Sonar Performance Engine (SPE) computes the signal excess (standard sonar equation) for a set of arbitrary receiver locations and capabilities within an environment by making a single call to CRAM. Optimization can then be performed by linking the SPE with a non-linear optimization algorithm (the Genetic Algorithm) for an efficient search of asset allocation choices. An example of the multi-sensor passive area coverage for two arbitrary arrays in the San Clemente basin is shown in figure 1. The environmental input for this run is the World Ocean Atlas (temperature/salinity), ETOPO1 bathymetry and a user specified geo-acoustics. The single CRAM run generates one-way TL from each geo-referenced "target" location to the user specified "receiver" locations. The figure shows the depth-averaged (over 0-100m) transmission loss for two sensors. Standard output is a KML plot for display in Google Earth.

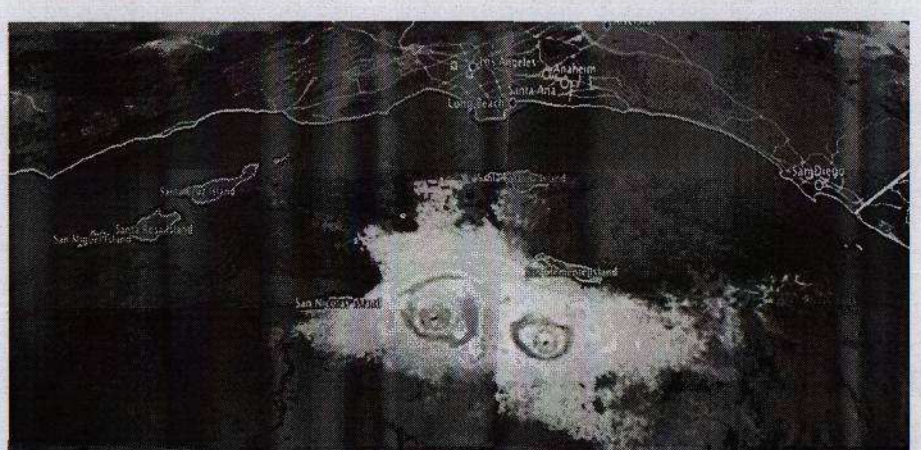

Figure 1. Example of Average TL output from Nx2D CRAM for two "receivers" from each location on a Cartesian grid in geographic coordinates.

2 PE CODE AND PASSIVE SONAR EQUATION

2.1 Nx2D CRAM

The program is invoked by supplying a text input file full of parameters, defining the geographic coordinates and depths of the sources and the receiver grid, and the paths of gridded environmental input files in netCDF format, as well as other control parameters. The program then sets up a set of 2D radials, samples the environmental files (NCOM or WOA for the sound speed profile, ETOP01 for the bathymetry, our own format for the bottom) as needed, and executes the separate 2D runs in parallel. The entire functionality is self-contained within a single executable - each radial is run as a function call to a highly modified, C-ported version of RAMGEO 2.0. No intermediate reading/writing to disk is necessary. Output is in the form of binary files in a very basic format readable by a simple Matlab function.

In the terminology of CRAM, a "source" is the point from which the the 2D PE run(s) are computed outward, and a "receiver" is a point on the output grid of each 2D run. Reciprocity permits these terms to be exchanged as the application requires, but for consistency we will use these terms in our literature.

The efficiency of CRAM is obtained by its unique handling of the multiple coordinate systems involved in the Nx2D problem. Propagation from a point source in the ocean is best expressed in cylindrical coordinates, where the environment is said to be dependent on range, depth, and azimuth, and the various azimuthal runs are computed by independent 2D PE runs (until azimuthal coupling is added, at which point the problem becomes a full 3D propagation model and the separate azimuth runs are not independent). However, the desired output for an Nx2D computation is often a lon/lat/depth grid in geographic coordinates. A conceptually easy method for handling this coordinate disparity would be to compute the Nx2D runs in a cylindrical region and then interpolate them onto the geographic grid as a post-processing step. This can be computationally intense and results in interpolation artifacts that are not acceptable when dealing with complex-valued fields.

CRAM uses a better method, which results in an output produced without interpolation. The implementation of the 2D PE core in CRAM permits an irregular spacing of output ranges to be specified for each radial (this is accomplished by varying the PE range step size dynamically in order to make sure it is an integer factor of the distance to the next requested range). A set of radials to be calculated is decided upon, based on an acceptable cross-range error at the maximum range desired. For each of these radials, a separate list of output ranges is initialized. Then, for each output point on the grid, the radial bearing passing nearest to it is determined, and the exact range of the output point to the acoustic source is added to the nearest radial's list of desired output ranges. Finally, all of the radials are computed independently (an "embarrassingly parallel" problem which takes full advantage of multiprocessor architecture) and the field outputs are copied directly into their pre-assigned slots in the geographic coordinate grid. An illustration of the geometry is shown in Figure 2.

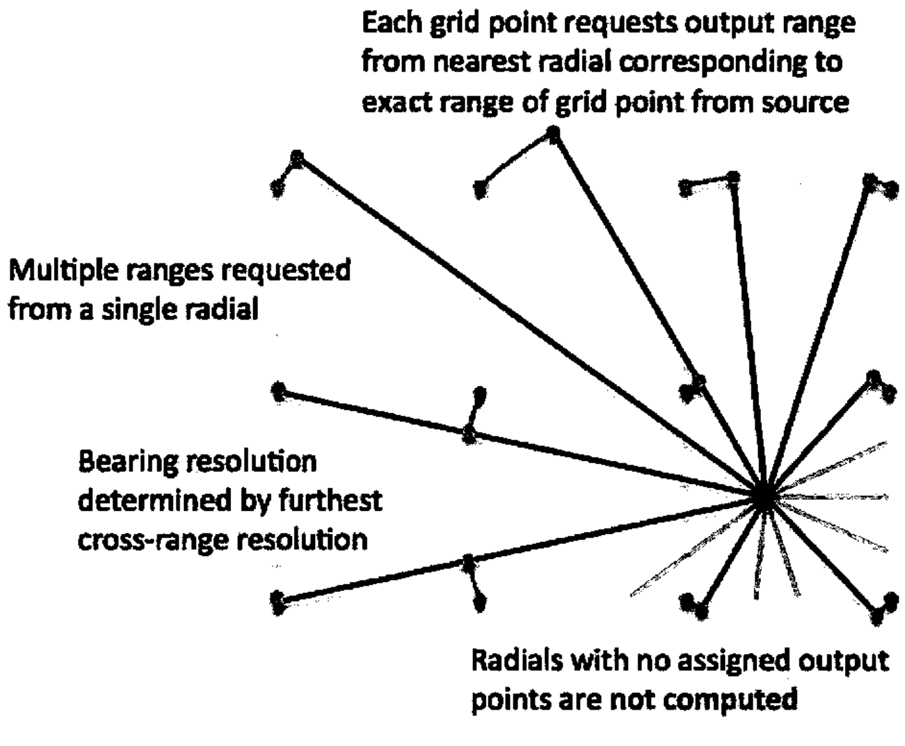
Each grid point requests output range from nearest radial corresponding to exact range of grid point from source
Multiple ranges requested from a single radial
Bearing resolution determined by furthest cross-range resolution
Radials with no assigned output points are not computed

Figure 2. Geometry of the Cartesian (geographic grid) mapping to Nx2D radials in CRAM.

The assumptions inherent in this method are that cross-range error in the cylindrical-to-geographic mapping is permitted within a tolerance that is on the scale of the output grid spacing, and that along-range error is not tolerated at all. Satisfying these criteria easily permits coherent field processing techniques such as beamforming of multiple sources in an arbitrary-shaped array, or time-domain synthesis via inverse Fourier transform.

Consider as well that the desired outputs need not be arranged on a grid. An arbitrary vessel track can be mapped onto the set of Nx2D radials via the same method, and take advantage of the same efficiency of mapping. This alternate output is enabled within CRAM by specifying individual receiver lon/lat/depth coordinates rather than grid parameters.

Another way of running the Nx2D problem is available in case cylindrical coordinate output is actually desired, and this is invoked by running a separate executable. This allows an output "grid" to be specified in depth/range/bearing, rather than depth/lon/lat, and the field will be computed from the source to each point in this grid. Internally, the same mapping technique is used, but for a single source this reduces to a 1:1 operation with no effect. If multiple sources are specified, the output coordinate system is centered on their mean longitude and latitude, permitting calculation of the field due to an arbitrary volumetric array. The chief application of this is for calculation of wind noise covariance matrices over a given array, and CRAM will automatically generate such a matrix as a quick post-processing step, if given multiple sources.

Both executables (Cartesian and radials) allow for multiple frequencies, source locations, and receiver depths to be run all at once. The output is a [Z x R x B x F x N] matrix for the radials code, and a [Z x X x Y x F x N] for the Cartesian code, where Z is depth, X and Y are longitude and latitude, R and B are range and bearing, F is frequency, and N is source index.

The inclusion of the environmental look-up software within the Nx2D CRAM run leads to a significant paradigm shift regarding code structure for integrated Sonar Tactical Decision Aids (STDA). Historically, via RAM and other propagation models, the Transmission Loss (TL) is computed by generating selecting a radial, generating the 2D environmental field input file, running the propagation model and then reading in the TL field via a file. This process requires a substantial amount of I/O even when efficiently coded using RAM-disks. For CRAM, however, the entire TL process occurs within memory with a simple set of calls to the NETCDF reader (an efficient operation in itself). This leads to significant improvements in ease of coding, operational speed and system i/o requirements. There are substantial advantages (interpolation, coherent field computations, etc) associated with having the full Nx2D field within memory as well.

2.1.1 Sonar Performance Engine

With the field computed from each position on a geographic grid to a set of asset locations, we are in a position to compute the multi-platform passive sonar performance. The standard narrowband passive sonar equation is used[4] to compute the signal excess:

$$SE = SL + TL - (AN - AG) + 10\log T + DT \qquad (1)$$

where SL is the source level (dB/uPa^2/Hz), TL is the transmission loss (dB), AN is the directional ambient noise (dB/uPa^2/Hz), AG is the directional array gain (dB), T is the integration time(s) relative to one second and DT is the detection threshold (dB) set at +6 in this example. Ambient noise can be defined as a location dependent (lat/lon) noise directionality. For a specified array orientation, the beam noise is computed from the array to each location on the geographic grid. The beam noise level (BN = AN - AG) is then used as the 3rd term on the right hand side of the sonar equation. Ambient noise can be computed using a standard US Navy model (Dynamic Ambient Noise Model), computed separately using CRAM, read from a table, or given as a single omni-noise value for the entire domain. CRAM is ideally suited for computing the environmentally dependent noise directionality. For noise directionality computations, the Nx2D field is run from each receiver location out in range along each of a sufficiently dense set of bearings. Surface wind noise is generated by integrating the associated patch size ($rd\theta$) with the dipole radiation source pattern using a quarter-wavelength source depth. Each patch in range is added incoherently, generating a bearing-dependent noise directionality. Beam noise can be computed by convolving the array beam pattern with the noise-directionality field. This does not include full-field propagation effects beyond TL in the radial direction. To include full propagation in the beam-noise computation, an array can be used within CRAM. The field from each phone is then computed to each patch of the ocean. The noise cross-spectral density matrix (CSDM) is computed by summing the outer-products of the array pressure response vector for each patch. Beam responses are determined by applying the beamformer to the wind CSDM.

For optimal lay-down computations a cost function must be defined. Various tactical options are available (area-coverage, search rate, minimal probability of false-alarm, time-to-detect). For the multi-asset problem several more options of the cost function are available. These include total area covered and area where multiple detections occur. For illustration purposes, here we present a cost function based upon the area covered with a 50% probability of detection (Pd). For a given DT (defining the Probability of False Alarms, Pfa) the 50% Pd is when the Signal Excess is greater than 0. The cost-function is then the integral of all area where SE is greater than zero. With the cost function and the available assets defined, a non-linear optimization algorithm can be used to search the parameter space. We have integrated the SPE (with CRAM) into a Genetic Algorithm[5, 6] search which seeks to determine asset locations and orientations that maximize total area coverage.

3 PHILIPPINE SEA EXAMPLE

As an example, optimal passive sonar performance computation is presented based upon the ONR Philippine Sea 2009 experiment. During this experiment a source and receiver ship were towed in the Philippine Sea and recordings of TL vs range were made, observing both Convergence Zone (CZ) propagation and bottom bounce (BB) paths. Recordings were made of narrowband tones from 25-400 Hz. For the optimization example, we seek to deploy two 64-element horizontal arrays with 3m spacing, at a depth of 100m, and optimize the detection of passive targets within a depth region of 0-200m. The frequency is set to 50 Hz. The ambient noise field is omnidirectional and is set to 70 dB and a 10 second integration window is applied. The source level is 130 dB. The GA was run with 10 individuals for 10 generations. In this paper, we make no claims that this is an exhaustive search or that the system has converged on the optimal solution. We simply state that in the given time, the GA has found an allocation of resources that are an improvement over those of a random laydown. The results of a Genetic Algorithm search for a two-receiver laydown (two towed horizontal line arrays) is shown in Figure 3.

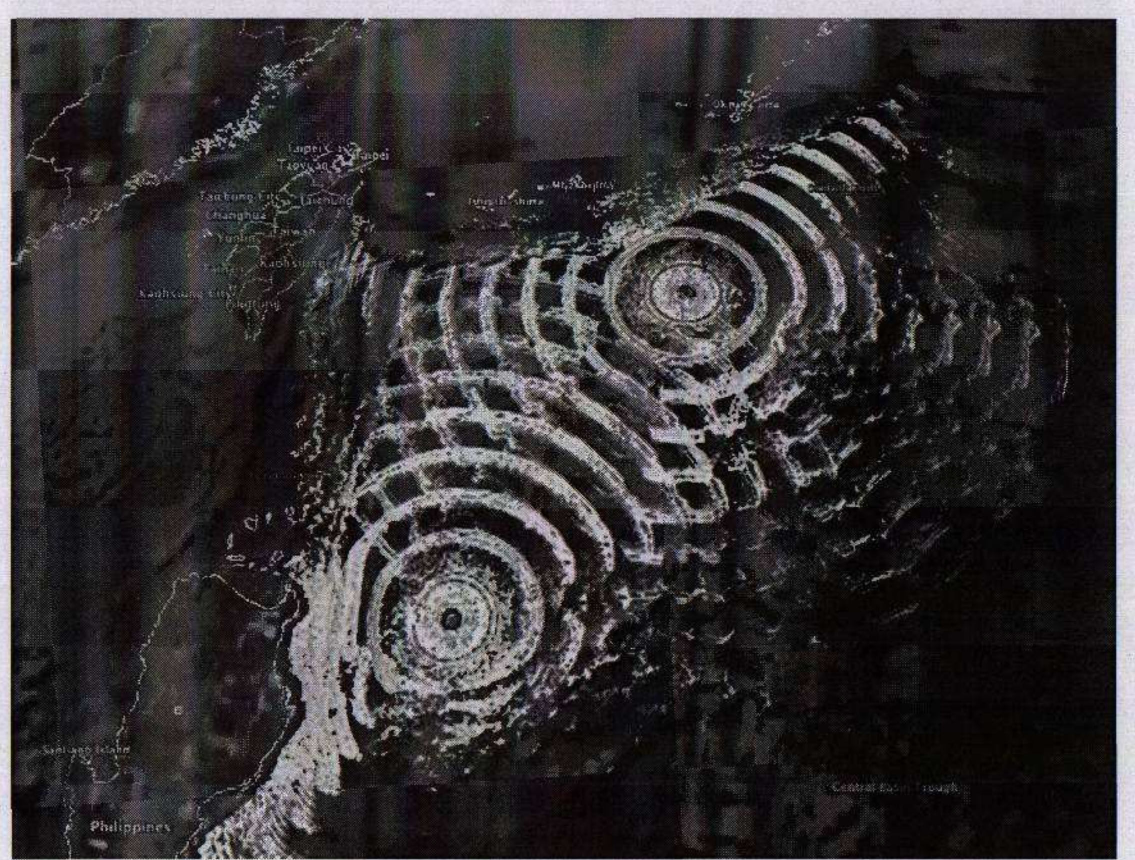

Figure 3. Average SNR over depth for 2-platform laydown using the combined Genetic Algorithm and Sonar Performance Engine approach. The dynamic range is 0-30 dB.

4 REFERENCES

1. Michael Collins, *A split-step Pade solution for the parabolic equation method*,Journal of the Acoustical Society of America **93** (4), 1736-1742 (1993).
2. Michael D. Collins and Evan K. Westwood, *A higher-order energy conserving parabolic equation for range-dependent ocean depth, sound speed and density*,Journal of the Acoustical Society of America **89** (3), 1068-1076 (1991).
3. Michael D. Collins, *A higher-order parabolic equation for wave propagation in an ocean overlying an elastic bottom*,Journal of the Acoustical Society of America **86** (4), 1459-1464 (1989).
4. Robert J. Urick, *Principles of Underwater Sound*, 3 ed. (Peninsula Publishing, 1995).
5. David E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. (Addison-Wesley, Boston, 1989).
6. J. Holland, *Adaptation in natural and artificial Systems*. (University of Michigan Press, 1975).