AraSimQC and Public Simulation Sets

Brian Clark
23 July 2015
ARA Collaboration Meeting

Supported by NSF CAREER Award 1255557
and BigData Grant 1250720
Purpose of AraSimQC

- Allow a quick visual comparison between:
  - Different versions and releases of AraSim
  - Different detector configurations and station number
- Allow the user to easily browse all histograms of variables from all classes
- Automatically generate general purpose simulation sets for each new version/ release of AraSim
- Create website for easy viewing of AraSim output (plots, simulation sets, etc)
Purpose of AraSimQC (cont)

• AraSimQC:
  o Is a collection of interlocking bash scripts
  o Checks for updates to AraSim (releases and trunks)
  o Generates a set of output plots when changes are found

• AraSimQC automatically posts these to a website for viewing

• AraSimQC saves the simulation output so it can be downloaded and re-used later for general purposes

Work closely with Kaeli Hughes, who writes IcemcQC, the software package for the ANITA (Antarctic Impulse Transient Antenna) Experiments. Huge thanks to Dr. Carl Pfendner for mentorship and support during the development of both AraSimQC and IcemcQC.
Website: Feel Free to Follow Along

- Website home: http://ara.physics.ohio-state.edu/arasimqc
- All internal pages require the general collaboration username and password
Goals of AraSimQC

• Runs automatically—minimize human supervision
• Modular design
• Resource efficient: utilizes PBS/qsub for batch processing of simulations
• Multiple checks for errors throughout; thorough logging of errors and progress
• Easy to update as collaboration needs evolve
History of AraSimQC

• Born out of a need to keep track of the rapid changes to the AraSim code
  o ARA Collaboration has 10+ global partners
  o AraSim generally sees 2-3 trunk updates per month

• May-August 2014: initial code designed and written by graduate Khalida Hendricks May-August 2014 at OSU

• September 2014-present: further modifications and development maintained by Brian Clark and Kaeli Hughes (IcemcQC) also at OSU
If Code Updates are Found:

• AraSimQC download and runs the latest versions of AraSim (both releases and trunks)
• It makes a standard set of plots
• It uploads these plots to a website for viewing
If Code Updates are NOT Found:

- AraSimQC checks to see if a new plotting program has been added since the last run
- If yes, new plots are generated for all existing trunks and releases
- These updates are appended to the website
Total Automation

- Procedure is totally automatic
- Run by an automated account on OSU computing cluster
- Performed once per week by a chron job

- Manual mode possible: useful if you want to make a bunch of plots for a selected revision
Website Features: Plot Comparison

- Home -> AraSimQC Internal -> Plots Comparison

- Is the primary end product of AraSimQC
- Allows easy comparison of a plot across revisions, station configurations, and energies of AraSim
- Useful “All Plots” feature, which will show you all of the plots associated with a single class
- Can easily download a PNG or PDF copy of each plot
First, select the parameters that will be shared between the two plots.

Select Class: Primaries
Select Plot: posnu

The description of the plotted variable will appear below (if available):

Now, select the parameters for the first plot.

Select ARA Configuration: ARA 1
Select Energy (eV): 10e18
Select Revision Number: Release 1.0

Finally, select the parameters for the second plot.

Select ARA Configuration: ARA 7
Select Energy (eV): 10e19
Select Revision Number: Trunk 510

Compare!
Doxygen Comment Support

- Doxygen is a tool available for generating program documentation from annotated C++ sources (also does PHP, Java, Python, etc.)
- Parses source files (header files, .cc files, etc), figures out class structure and code comments, and outputs them in easy formats
- The full doxygen documentation for AraSim is available: Home -> AraSim, link at bottom of page
Here are the classes, structs, unions and interfaces with brief descriptions:

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antenna</td>
<td></td>
</tr>
<tr>
<td>Antenna_r</td>
<td></td>
</tr>
<tr>
<td>Antenna_string</td>
<td></td>
</tr>
<tr>
<td>ARA_station</td>
<td></td>
</tr>
<tr>
<td>RayTrace::attenuationModel</td>
<td>A class providing a description of the attenuation length for radio signals in the ice</td>
</tr>
<tr>
<td>basicAttenuationModel</td>
<td></td>
</tr>
<tr>
<td>constantRefractiveIndex</td>
<td></td>
</tr>
<tr>
<td>Counting</td>
<td></td>
</tr>
<tr>
<td>Detector</td>
<td></td>
</tr>
<tr>
<td>EarthModel</td>
<td></td>
</tr>
<tr>
<td>Efficiencies</td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td></td>
</tr>
<tr>
<td>exponentialRefractiveIndex</td>
<td>Implements an exponential parameterization for the ice index of refraction</td>
</tr>
<tr>
<td>RayTrace::fullRayPosition</td>
<td>A structure for tracking intermediate coordinates during a ray-trace</td>
</tr>
<tr>
<td>IceModel</td>
<td></td>
</tr>
<tr>
<td>IdealStation</td>
<td></td>
</tr>
<tr>
<td>RayTrace::IndexOfRefractionModel</td>
<td>A class providing a description of the index of refraction of the ice</td>
</tr>
<tr>
<td>InstalledStation</td>
<td></td>
</tr>
<tr>
<td>Interaction</td>
<td></td>
</tr>
<tr>
<td>inverseExponentialRefractiveIndex</td>
<td></td>
</tr>
<tr>
<td>RayTrace::minimalRayPosition</td>
<td>A structure for tracking intermediate coordinates during a ray-trace</td>
</tr>
<tr>
<td>negligibleAttenuationModel</td>
<td>Implements an attenuation model with no attenuation</td>
</tr>
<tr>
<td>Parameters</td>
<td></td>
</tr>
<tr>
<td>pathPrinter&lt; positionType &gt;</td>
<td></td>
</tr>
<tr>
<td>RayTrace::pathRecorder&lt; positionType &gt;</td>
<td></td>
</tr>
<tr>
<td>pathStore_test&lt; positionType &gt;</td>
<td></td>
</tr>
<tr>
<td>pathStore_vector&lt; positionType &gt;</td>
<td></td>
</tr>
<tr>
<td>Position</td>
<td></td>
</tr>
<tr>
<td>RayTrace::positionRecordingWrapper&lt; positionType &gt;</td>
<td>For internal use by TraceFinder</td>
</tr>
<tr>
<td>Primaries</td>
<td></td>
</tr>
<tr>
<td>quadraticRefractiveIndex</td>
<td></td>
</tr>
<tr>
<td>Ray</td>
<td></td>
</tr>
<tr>
<td>RayTrace::IndexOfRefractionModel::RayEstimate</td>
<td></td>
</tr>
<tr>
<td>RayTrace::rayPosition</td>
<td>A structure for tracking intermediate coordinates during a ray-trace</td>
</tr>
<tr>
<td>RaySolver</td>
<td></td>
</tr>
</tbody>
</table>
Doxygen Comment Support (cont)

• The AraSimQC website displays “hint boxes” that give the meaning of plotted variables
• These definitions are generated by the doxygen comments found in the AraSim code itself
• An AraSimQC module parses the class header for the doxygen flags, outputs comments in method friendly to jQuery processing
The description of the plotted variable will appear below (if available):

position where neutrino enters the earth
Adding Plots

- Adding new plots is straightforward; super-user places plotting programs in a specified directory—Done!
- All plotting programs must be C++ compilable programs
- Website supports the submission of plots by the collaboration—please help us make AraSimQC better!
- New plots submissions must include:
  1. List of all plotting programs
  2. List of all plots (can have more than one plot per program)
  3. Attach plotting programs and makefiles
- Link and detailed instructions are on the “plots request” page (Internal Home -> Plots Request)
Step 1: List Plotting Programs (.cc Files)

Below, list the names of the program files you are submitting. List one script per line.


Step 2: List Plot Titles

Below, list the titles of the plots you are submitting. List one plot per line.


Step 3: Attach Plotting Scripts and Makefiles

Below, attach the plotting program and corresponding makefiles. Please submit them as a single tarball or zip file.

Drop files or click here to upload
Website Features: Public Simulation Sets

- Simulations are resource intensive—need lots of time and computing power
- The AraSimQC output will serve as “publicly available” simulation sets (available to collaboration only)
  - Several energies (10e17 through 10e21), option for input spectrum if requested
  - Several configurations (ARA1, ARA7)
- Coming soon: the simulation sets can be accessed via the OSU ARA website
Next Steps

• Addition of more classes and plots; your help is welcome and requested!
• Adding plots is a great project for a new student!
Please head to the website, try it out, and offer us comments on further improvements!

---

Thank you!

Questions? Comments?
Back-Up Slides
The AraSimQC Software Outline

1. AraSimQC and AraSimQCParameters.txt
2. AraSimQC Updater
3. AraSimQC Version
4. AraSimQC Setup
5. AraSimQC Submit
6. runAraSim
7. AraSimQC Plotter
8. AraSimQC Sorter
9. AraSimQC Website*
10. ArasimQC Comments*
11. AraSimQC Closeout

* = Auxiliary Modules
AraSimQC Software Flowchart

1. AraSimQC
   - AraSimQC Updater
     - If releases has been updated...
       - If trunk has been updated...
         - AraSimQC Version
         - AraSimQC Setup
         - AraSimQC Submit
         - AraSimQC Plotter
           - AraSimQC Sorter
           - AraSimQC Website
           - AraSimQC Closeout

23 July 2015

AraSimQC and Public Simulation Sets