



Simulation in IceTray

(cascades and flashers anyway)

Tom McCauley

LBNL

Flasher/Cascade workshop

July 25, 2005



Details, misc.



- Latest version: V01-00-03, released July 23
- <http://code.icecube.wisc.edu/svn/meta-projects/simulation/releases/V01-00-03>
- Documentation: <http://code.icecube.wisc.edu>
- Serialization now included: i.e. Can output offline data format to binary, xml files; binary can be read back in
- Steering files are commonly ROOT scripts
- Python interface to IceTray: ithon project
- XML input format included with ithon



Configuration/sources



- In IceTray, data is broken up into streams
- Data resides in the Physics stream
- Configuration data is found in the Calibration, DetectorStatus, and Geometry streams
- Geometry: The positions of the DOMs (of course)
- Calibration: e.g. ATWD gains, pedestals, etc. (i.e. information one would find in domcal*.xml)
- DetectorStatus: What DOMs are on, what channels are used, LC modes, trigger modes, etc. (i.e. information one would find in data XML steering files)



Simulation configuration issues



- There are a lot of “knobs” in the simulation: how can they be handled?
- Each stream is handled by a Source class
- Sources: Set by hand (in the steering file), ASCII file, XML file, F2k file, SQL Db
- How are all these accommodated?
- Highest- level configuration: given a time (like an event time), input the time to the SQL database and construct the detector as it was then
- Lowest- level configuration: make my own custom detector by hand
- Many permutations in- between
- Q: what's needed for accurate simulations?



Cascade simulation



- Simple cascade generator: cascade-generator project included w/ simulation
- See Sourav's talk for more details on results
- `/cascade-generator/resources/scripts/TestCascade.C`
(.py, .xml)



Flasher data (and simulation)



- Why? 1) Data could provide a “real” cascade in which to test cascade reconstruction algorithms 2) A comparison of flasher data to simulation can provide an end- to- end verification of the simulation
- No show- stoppers that would prevent doing flasher simulation in IceTray (?)
- What's needed: photonics tables
- PSInterface needs level 1 and level 2 tables; flashers are a “level 1” problem: a simple hack should work around this(?)



Conclusions



- Recent release of simulation is “good-to-go”
- Try it out: more users -> more feedback -> better simulation