Simulation Tutorial @ SpinFest

Charles F. Maguire
Vanderbilt University

Tutorial Assumptions

- **Audience is grad students or post-docs new in PHENIX (we have several examples recently in our group at VU)**
  - Posses basic programming skills
  - Essentially unfamiliar with PHENIX software system
  - Tutorials will last ~15 minutes

- **Much documentation already exists (~14 years worth!)**
  - Unfortunately documentation is scattered
    There is no one binder of documents that we can give new people
  - Worse, old documentation becomes obsolete and wrong
  - Learning the software system is often trial and error
  - Wikipedia looks like a good idea (as long as it is maintained/updated)

The Basic Building Blocks

what I tell our new people (~2-3 week learning curve)

- **The programming language is (largely) C++**
  - If you don't know C++ , you better get a book and learn it
    Recent text: learn C++ in 24 hours (??, used to be 21 days)
  - Simulation system still has a FORTRAN77 component
    which is a legacy of using GEANT3 (like having to know Latin)

- **Software is stored in a repository called CVS**
  - You can probably learn enough CVS in 24 hours
  - Only a small number of commands are typically needed
  - Software libraries are compiled and built every ~24 hours

- **Principle user interface to software is ROOT**
  - ROOT "macros" control execution of the software
  - ROOT has excellent graphical capabilities
  - Mammoth set of centralized documentation http://root.cern.ch

So where do you start?

- **For all things in PHENIX you can start here**
  - http://www.phenix.bnl.gov/internal.html
  - Main source of information for any major category in PHENIX
  - Look at “Computing” and “Simulation” subsystem links

- **For all software information use Computing link**
  - http://www.phenix.bnl.gov/WWW/offline
  - Look especially at the tutorials menu item
  - This computing page also has a link to the main Simulation page

- **For specific simulation information go first to**
  - http://vpac17.phy.vanderbilt.edu/index.html
  - This main page also has a tutorials menu item

What do you see first at Simulation page?

- ***People* menu button**
  - Mug shots of Simulation group members
    (Out of date, new people not yet included, previous members not removed)
  - Principle support members
    - Ivan Danchev (new post-doc)
    - Hugo Valle (senior graduate student)
  - *If you have a problem in simulation, best to e-mail all three of us*

- ***Simulation* button**
  - Gives a 4 paragraph overview of how simulations are done
  - Contains other buttons for more (to be done) descriptions

- ***Tutorials* button**
  - First link tells how to get started with simulations today in PHENIX
  - Other buttons tell you how previous years simulations were done

The Four Steps in Simulation

- ***Generate*** **event files with separate programs (EXODUS, HIJING)**
  - There is old (and new) event generator documentation

- ***Track*** **events through PISA program to generate "Hits files"**
  - PISA is PHENIX's implementation of the GEANT3 simulation libraries
  - GEANT3 is a third generation simulation package from CERN
  - There is now a fourth generation (pure C++) GEANT since 1999

- ***Reconstruct*** **hits information into data summary files (DSTs)**
  - Done using the ROOT/Fun4All framework in PHENIX

- ***Analyze*** **DSTs into physics results**
  - Typically user written libraries for specifics physics topics
    Done also in ROOT/Fun4All framework
  - Corresponds to the "Analysis Train" phase for the real data processing

Getting started with the Four Steps in Simulation
Using the "one stop shopping" method

- **"One stop shopping" method**
  - A single WWW page gives you all the instructions for each step
  - For two steps all necessary files are obtained with one command
  - Assumes the user will be working at RCF
- **Instructions for one stop shopping are at tutorial link**
  http://www.phenix.bnl.gov/phenix/WWW/simulation/pisaHead.html
- **To generate PISA hits output files**
  - source /afs/rhic/phenix/software/simulation/head/pisaLinker.csh
    pisa < pisa.input >& pisa.out &
- **To reconstruct PISA hits files into DST output files**
  - source /afs/rhic/phenix/software/simulation/head/pisaToDSTLinker.csh
    root < pisaToDST.input >& pisaToDST.out &
- **To analyze simulated CNT nanoDSTs**
  - CVS check out of the (new) offline/analysis/simCNTCheck area
    Just written yesterday to work with the newest library version

Three of the Steps Under the Microscope

- **PISA hits file generation**
  - Main output is a ROOT format file called PISAEvent.root
    pisa.out ASCII log file generally not used unless there was an error
  - Main input control file is pisa.kumac file
    kumac is like a FORTRAN predecessor to ROOT's C++ macros
  - Only change typically needed in pisa.kumac file is for magnetic field sign
    This is done in the MAGF control line (see WWW page documentation)

- **Reconstruction of the PISA hits file into DST files**
  - Three flavors of DSTs can be produced (DST, nanoDST, HWG) be default
    simDST.root, simCNT.root, simHWG.root
  - Main input control file is ROOT macro called pisaToDST.C
    You should read this macro and all of its comments
  - Subsidiary input control file is pisaToDST_IOManager.C
    Controls contents of the output files

- **Analysis of the CNT files (as an example)**
  - CVS check out of the (new) offline/analysis/simCNTCheck area
  - Follow this by a standard build of the simCNTAnalyze library
  - Follow the build by using the analyze_simCNT.C macro in the macro area

Three of the Steps More Under the Microscope

- **PISA hits file generation**
  - Main output is a ROOT format file called PISAEvent.root
  - PISAEvent.root file can be “scanned” with pisaRootRead binary
  - pisaRootRead binary produces ancestor NTUPLEs: ancXxx.root
    For example ancdch.root has the hits in the Drift Chamber
  - Besides the hits information, one also gets the track ancestry information

- **Reconstruction of the PISA hits file into DST files**
  - There is a special evaluator class EvaSimreco
  - Evaluator class is not typically used in production work
  - Evaluator class output file is evaSim.root containing several NTUPLEs
  - The evaSim.root contains reconstruction and ancestry information
    for several central arm components: EMC, Pad, Cgl, Reaction Plane
    There is also an NTUPLE for pair mass reconstruction with diagnostics

- **Analysis of the CNT files (as an example)**
  - Follow the build by using the analyze_simCNT.C macro in the macro area
  - Output file is simCNTAnalyzer.root
  - Also an NTUPLE file like one of the evalSim.root files
    except that there is no diagnostic tracking information
  - For another example, look at Tatsuya Chujo’s code (real and simulated DSTs)
    offline/analysis/HWGana/CuCu_Fun4All

Major Simulation Projects

- Consult simulation home page RunServer for details
  http://vpac17.phy.vanderbilt.edu/index.html

  - *RunServer software managed now by Hugo Valle (took over from Debsankar)*

- Special simulations projects request link
  Requires password (phnxsim03)

- Large project requests need big remote site farms (hundreds of Gbytes output)
  So far it seems that only the Vanderbilt ACCRE farm will run these large projects.
  - *Working with M. McCumber on Grid-based submission (~2 months away?)*
  - *Major simulation projects typically take 2-3 weeks of testing and production*

Summary

- **Much documentation does exist, most of it accurate but some of it obsolete**
- **Major effort in the next 3 months to have a good centralized package of simulation documentation**
  - Task delegated to our new post-doc who will be its first beneficiary
  - We have several new graduate student who will test drive the package too