ReactorFsim Status

Matthew Worcester

Background and Simulation Meeting

December 11, 2004

Outline

- Overview
- Input
- Event simulation
- Detector simulation
- Output
- Summary

Overview

- C++ (includes CERNLIB Fortran routines)
- Top level: ReactorFsim
  - Defines detector(s) and ntuple/histograms
- ReactorEvent creates an event
- each event passed through detector simulation(s) (LightsOut) in ReactorDetector
- ntuple and histograms filled with event and detector info in ReactorNtuple

Input

- src/event_setup.txt
  - read by ReactorFsim and ReactorEvent
  - contains some job and event information
  - should be expanded to increase job control
- data/cf252_position.txt
  - read by MyCfSource
  - contains number and position of sources
  - can be expanded to include intensity, other source properties if necessary

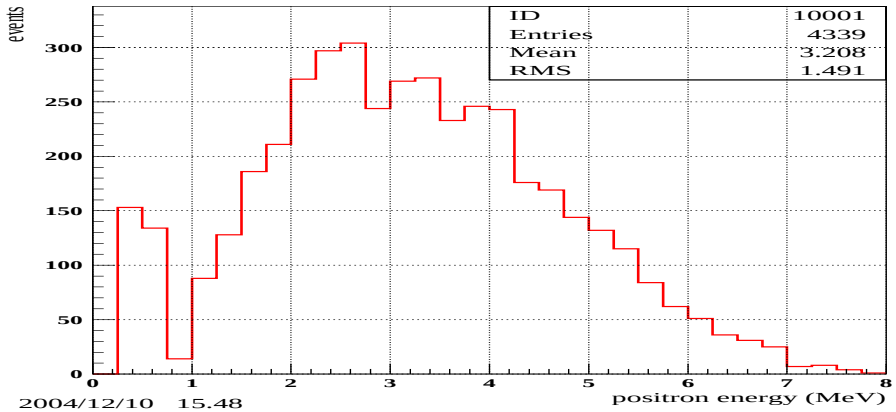
Input

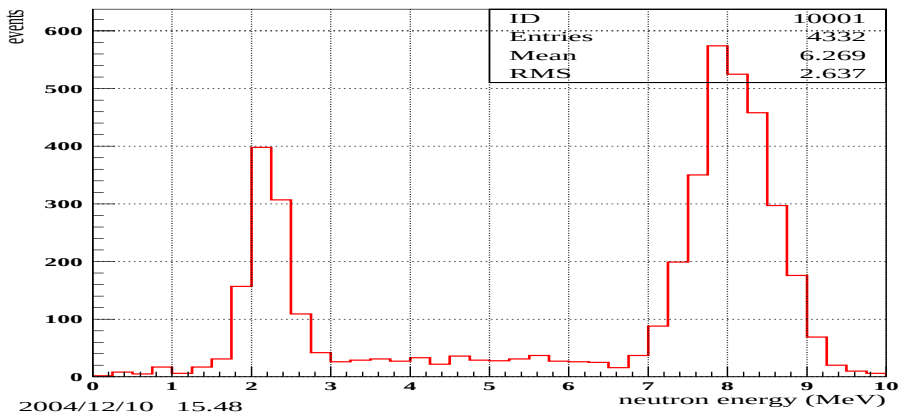
- data/tl_bi_decays.txt
  - read by MyTlSource and MyBiSource
  - contains ^{208,210}Tl and ^{212,214}Bi decay schemes
- data/Totb500a.dat
  - read by MuonPropagator
  - contains ~ 27k muons (energy, zenith angle) generated at 500 mwe by Martina and Jim's code
  - more muons and different depths easy to generate

Event simulation

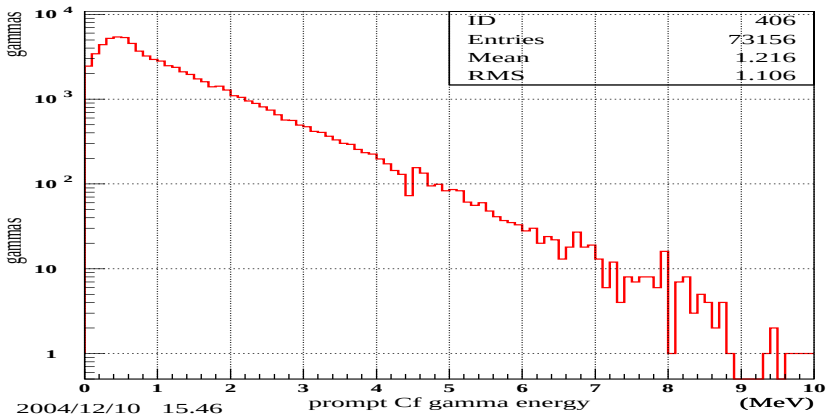
reactor

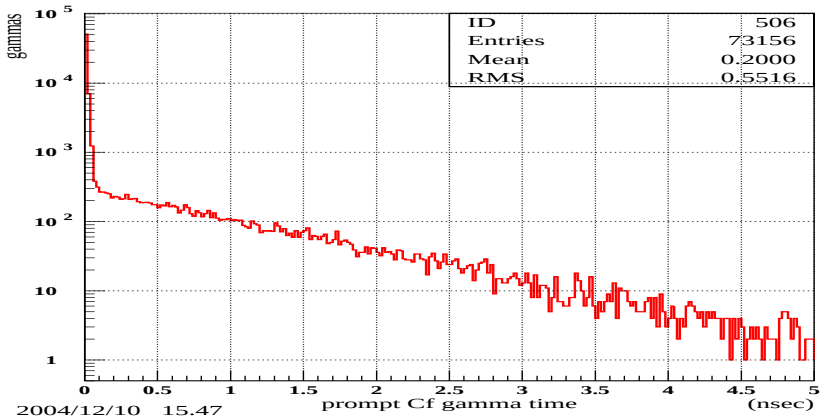
- inverse β-decay events
- neutrino energy and position selected randomly
- handles inverse β-decay kinematics
- fills momentum/energy (P) and position/time (R) arrays for one neutrino, positron, and neutron





Event simulation





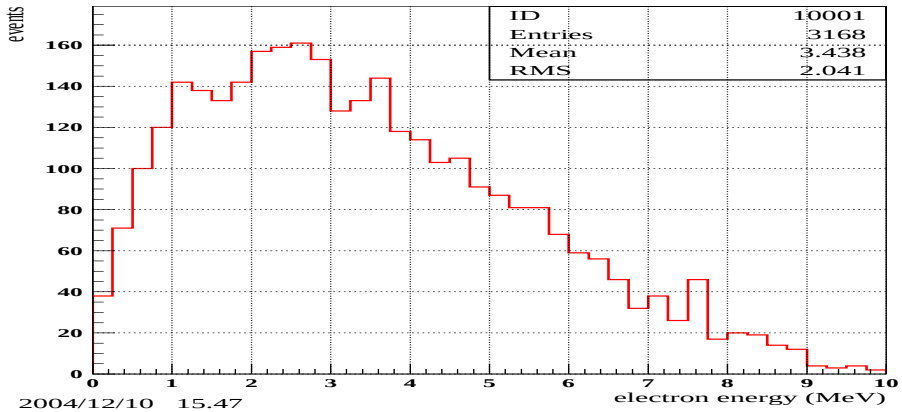
252cf

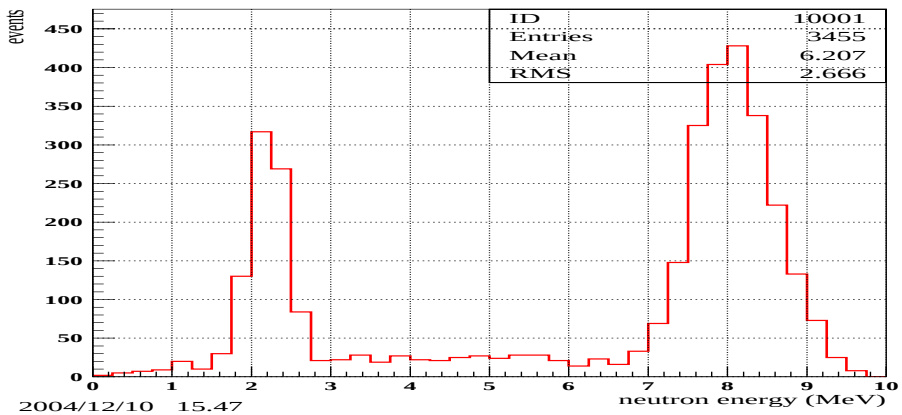
- calls MyCfSource to handle source physics
- Compton scatters the prompt photons
- fills P and R arrays for each neutron and photon

Events

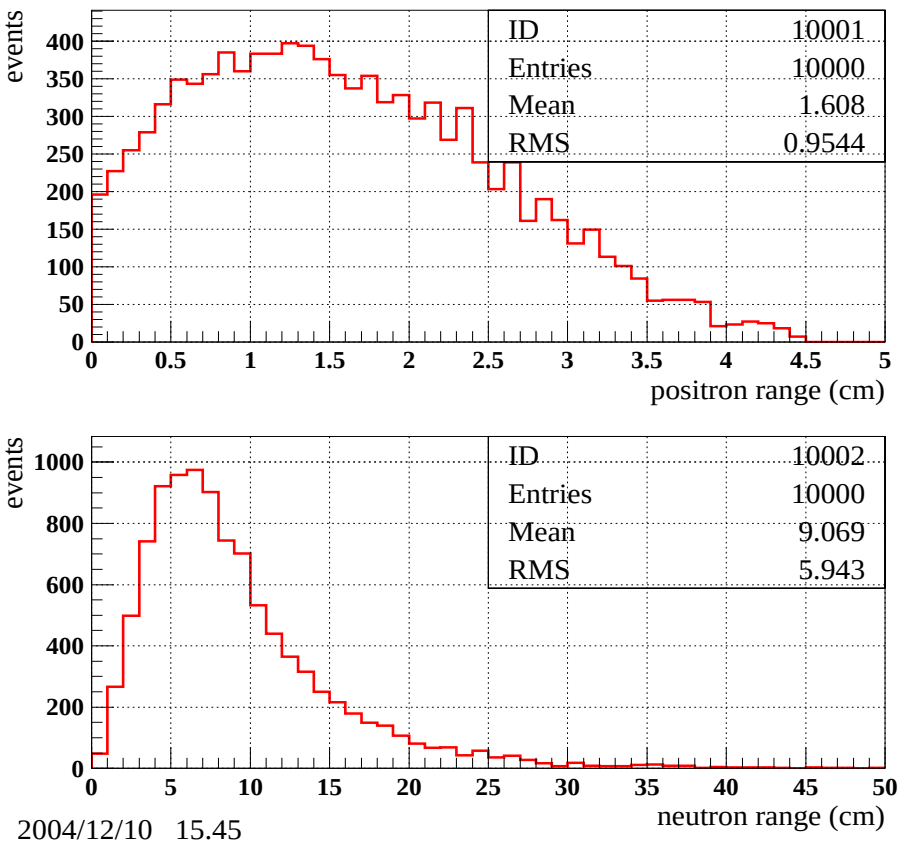
muon

- calls MuonPropagator to handle muon and $^{9}Li/^{8}He$ production and decay ($\rightarrow e^{-}$, n)
- keeps muon -dE/dx in virtual “photons” (not Compton scattered) along the muon track
- fills P and R arrays for one electron, one neutron and each “photon”





Detector simulation (LightsOut)



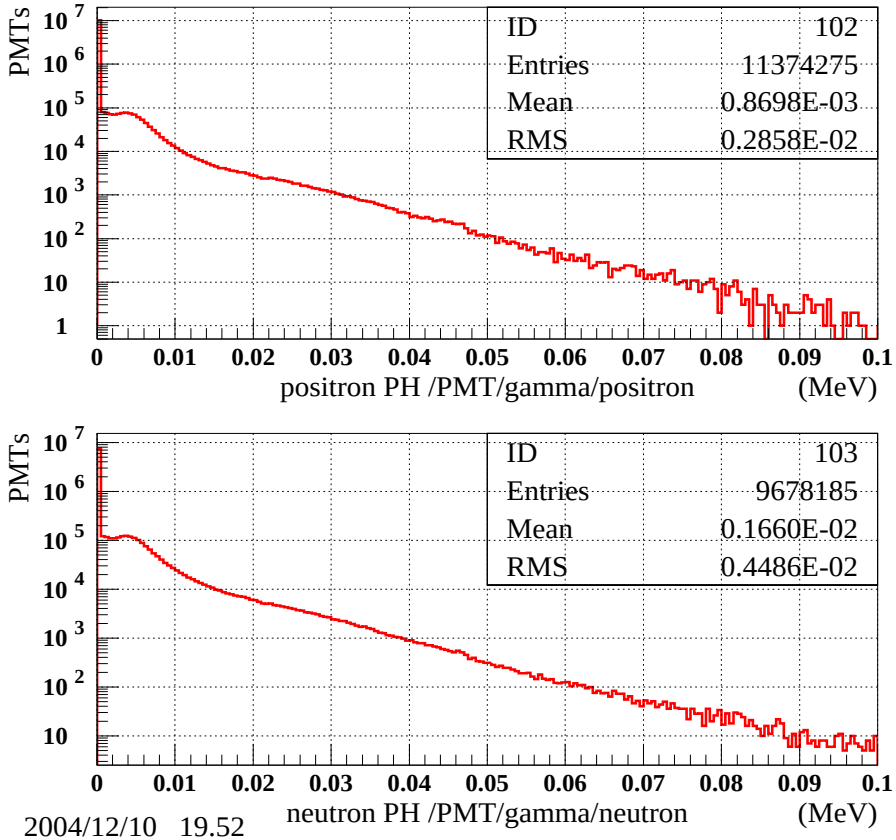
- GenerateSecondaries:
- propagates e^+ (PositronRange) and neutron(s) (NeutronPropagator) from P and R arrays
- does nothing for e^- (needs work)
- refills P and R arrays with new position, energy, time, etc.

Detector simulation (LightsOut)

- GenerateGammas:
- fills energy (E), position (R), time (T), and parent particle (O) arrays for each energy deposit (called a “gamma”) in the scintillator
- checks Event for any photon P and R arrays and passes information directly to E, R, T, and O
- generates PMT radiation photons and Compton scatters them into the scintillator
- each e^+ generates a two Compton-scattered annihilation photons and one “gamma” ($E_\gamma = KE_{e+}$)
- each e^- has one “gamma” ($E_\gamma = KE_{e-}$)
- each neutron generates Compton-scattered photons from H or Gd capture

Detector simulation (LightsOut)

- GenerateResponse:
- Loop over “gamma” information in E, R, T and models PMT response for each “gamma” type
- Saves data for each PMT:
  - generated and smeared PH
  - generated and smeared time
  - attenuation and solid angle
  - number of PE and Q
- Saves data summed and averaged over all PMT



Detector simulation (LightsOut)

- CalculateQPosition and ImproveQPosition do position reconstruction based on PMT quantities
- CalculateEnergy does energy reconstruction based on PMT quantities and reconstructed position
- ReactorTrigger simulation shell (does nothing) exists, called by ReactorFsim

Output

- PAW ntuple and histograms
  - all event and detector quantities
  - filled in ReactorNtuple
- Text written to screen
  - should be dumped to a log file

Summary

- Flexible structure exists to add new event types and particles
- Lots of big and small projects available (see question 3)
- Fastest development will occur with external input from other sources (as done for muons)