Jupiter, Satellites and SimTools

Akiya Miyamoto
KEK
16-March-2005

Simulation Mini Workshop

Based on acfa-sim-j activity
Contents

- Jupiter
- Satellites
- SimTools
- Summary
Overview of our tools

- **lcbase**: configuration files
- **Leda**: Analysis tools (Kalman fitter, 4vector and jet finder utilities)
- **jsf**: Root-based framework
- **lclib**: QuickSim and other fortran based utilities
- **physsim**: Helas-based generator

- **Jupiter**: Full simulation based on Geant4
- **Uranus**: Data analysis packages
- **Satellites**: Data analysis packages for MC data

- We use only C++, except old fortran tools.
- Link to various tools at http://acfahep.kek.jp/subg/sim/soft
- All packages are kept in the CVS. Accessible from http://jlccvs.kek.jp/
Framework, QuickSim, Physsim

- Framework
  - JSF: Root based framework for physics and detector studies
  - Packages in JSF are reorganized to reduce dependences among codes.
  - Interfaces to StdHep and LCIO are implemented.

- Quick Simulator and Iclib
  - Detector parameter set for “GLD” configuration is prepared and tuning of the parameters are in progress

- Physsim
  - Collection of event generators based on Helas.
  - Anlib package (4 vector manipulation and jet clustering, etc.) is moved to Leda package
Jupiter/Satellites Concepts

Tools for simulation

- **JUPITER**
  - JLC Unified
  - Particle Interaction and Tracking Emulator
  - Geant4 based Simulator
  - MC truth generator

- **Satellites**
  - Input/Output module set
  - Monte-Carlo Exact hits To Intermediate Simulated output

- **METIS**
  - Unified Reconstruction and ANalysis Utility Set

For real data

- **URANUS**
  - Unified Reconstruction and ANalysis Utility Set

**JSF/ROOT based Framework**

- **LEDAs**
  - Library Extension for Data Analysis

- **JSF: the analysis flow controller based on ROOT**

  The release includes event generators, Quick Simulator, and simple event display

A.Miyamoto, Simulation Mini Workshop (16-17, March, 2005)
Geometry definition in Jupiter

- Geometry definition
  - Still hard coded in the source, but
    - Addition/Deletion of sub-detector components are easy in Jupiter
    - Geometry parameters are all defined in J4XXXParameterList classes, so easy to maintain parameters
  - We’ve started to develop XML interface to implement very complicated geometry, but this work does not complete yet.
Geometry in Jupiter

- Muon/Iron
- Solenoid
- Hadron Calorimeter
- Ele mag. Calorimeter
- TPC
- QC1
- IT
- VTX
- Forward Cal.
Calorimeter Geometry

- Default sensor size:
  - EM: 4cmx4cmx1mm, 38 layers
  - HD: 12cmx12cmx2mm, 130 layers
- Replica
  - Phi direction: Tower and mini-tower
- Sandwitch structure of X/Y scinti structure can be defined.

A. Miyamoto, Simulation Mini Workshop (16-17, March, 2005)
CPU time for Geometry initialization

- Originally, geometry definition of calorimeter takes quite long time.
  - For default configuration, ~2.5min, 1cmx1cm case > 1hour
  - Geant4.6.1 was used.

- Solution for speed up
  - For every geometry definition, G4 code checks name conflicts. To scan all existing geometry, takes are long time. CPU time was proportional to (Number of geometry)^2
  - To improve,
    - Remove name matching in G4HCTable::Register
    - Order of for loop in G4SDStructure::FindSubDirectory is reversed.

- As a result,
  - 20sec even with 1cmx1cm case.
Range cut for Calorimeter

- Geant4’s default range cut is 1mm, which is too large for our sampling thickness.
- Energy deposit and resolution depends on the range cut.
- Range cut <0.3µm is good for ΔE
- Simple MC results are always slightly better than beam test.
- Small range cut is memory consuming → using 1~10µm

![Energy Resolution vs. Range Cuts](graph1)

![Beamtest(T405/411) @KEK](graph2)

by M.C. Chang
StdHep interface for Jupiter

- StdHep interface was prepared as JSF module.
  - JSFReadHepStd reads StdHep data and saves in JSFGenerator

- By executing Jupiter as a module of JSF, JSFJ4, StdHep data is used as an input of Jupiter simulation.
Digitiation and Hit making

- Jupiter creates only Monte Carlo truth hit points. Smearing/Digitization is performed later in Satellites, because Geant4 simulation is the most time consuming part.
- Detector hit classes (J4xxxHit) are delivered from J4VHit
  - **Standalone Jupiter:**
    - J4xxxHit's are written to the ascii file.
    - Satellite job reads the file, and converted to S4xxxExactHit (inherited from ROOT)
  - **JSFJ4 = Jupiter in JSF environment** (recent standard)
    - J4xxxHit's are transferred to S4xxxExactHit and saved as a root file.
- We want to keep information of tracks which creates signal in calorimeter or muon detector → create a virtual detector to save information as XXX_Post hits.
Output class of Jupiter

- Jupiter Hit classes are derived from J4VHit

- **G4VHit**
- **J4VHit**
  - HitID, TrackID
  - P, Edep
  - Xpre, Xpost, ...

- **J4CALHit**
- **J4VTXHit**
- **J4TPCHit**
- **J4TPCPostHit**
  - LayerID
  - PostHitID

A.Miyamoto, Simulation Mini Workshop (16-17, March, 2005)
Current aim is to prepare a minimum set of Metis modules for studies of Particle Flow Algorithm.

Novice users will be able to do physics analysis using information of PFO classes.

As a first step, a cheated track finder and a cluster maker, etc are in preparation in order to know ultimate performance.

Each module is independent, thus shall be easy to implement different reconstruction algorithm according to interests.
Metis Analysis Flow

1. Jupiter result
2. Make smeared TPC hits from exact hit
3. Make tracks from TPC
4. Make hybrid tracks (TPC+IT+VTX)
5. Make smeared/merged CAL hits from exact hit
6. Make cluster from CAL hits
7. Make Particle Flow Objects
8. Jet clustering
9. Physics study
Cheated PFO analysis

ZH event at $E_{cm}=500$ GeV

- Exact hit points of TPC and CAL are displayed.
- Hits belong to the same PFO are shown with the same color.
- A framework of event display in JSF is used.

By K. Fujii (KEK), S. Yamamoto (GUAS), A. Yamaguchi (Tsukuba)
ROOT's X3d view of the same event
Same event, after a forced 4-jet clustering on PFOBjects
How to get/use our tools

- Our software tools are maintained in CVS server, jlccvs.kek.jp.
- At http://jlccvs.kek.jp/,
  - Description about how to download latest version.
  - Web interface to the CVS repository,
    - http://jlccvs.kek.jp/cgi-bin/cvsweb.cgi/
  - Snap shot of source codes.
    - http://jlccvs.kek.jp/snapshots/
- Recently, we prepared SimTools web page for easy use of our packages.
SimTools

- SimTools is a collection of pre-compiled libraries of jsf, lclib, Leda, Jupiter, Satellites, Uranus and lcbase.
  - Build conditions
    - Compiled on Redhat 9.0, gcc 2.2.2
    - Using Root 4.00.08
    - Ready to use if your system meets these conditions.
  - Includes several examples in using JSF/QuickSim, Jupiter, and Satellites.
  - Source codes are also included

- Web page
Summary

- We have developed software aiming to study detector performance based on full simulator, Geant4.
- Primary goal was to study detector performance in perfect conditions and to know critical factors which affect jet reconstruction.