Analysis

I. Hrivnacova, IPN Orsay

Geant4 ED PHENIICS Tutorial,
13 - 17 May 2019, Orsay

Outline

- Introduction
- Geant4 analysis tools
    - Using Geant4 analysis
    - Histograms, ntuples, analysis UI commands
    - Reader, Batch graphics
- Using external analysis tools
    - ROOT, Gnuplot, Excel, Open[Libre]Office, Softinex

Geant4 Analysis Tools

Geant4 Analysis Tools

- Provides code to write **histograms** and **“flat ntuples**” in several formats:
  - ROOT, XML AIDA format, CSV, HDF5 (since 10.4)
- Analysis category in Geant4 since December 2011
  - Before the analysis code in Geant4 examples used external tools (based on AIDA = Abstract Interfaces for Data Analysis) that had to be linked with Geant4 to produce histograms or ntuples
- Area of new developments and improvements: more features are added in each release
  - Example from the latest release: support for HDF5 output type
- Based on g4tools from inlib/exlib developed by G. Barrand (LAL): http://inexlib.lal.in2p3.fr/
  - “Pure header code” - all code is inlined, can be installed, besides Geant4 supported platforms, also on iOS, Android

Using Geant4 Analysis

Basic steps:

1) Create G4AnalysisManager
2) Book (create) your histograms, ntuples
3) Open a file
4) Fill values in histograms, ntuples
5) Write & close file
6) Delete G4AnalysisManager

Using Geant4 Analysis

Basic steps:

1)	Create G4AnalysisManager	… in RunAction constructor
2)	Book (create) your histograms, ntuples	… in RunAction constructor
3)	Open a file	… in BeginOfRunAction()
4)	Fill values in histograms, ntuples	… anywhere (during event processing)
5)	Write & close file	… in EndOfRunAction()
6)	Delete G4AnalysisManager	… in RunAction destructor

Performing the steps in the suggested class&method is not obligatory, but it guarantees correct functioning in multi-threaded mode

1) Create Analysis Manager

- The analysis manager is created with the first call to G4AnalysisManager::Instance() function

MyRunAction.cc

```
#include "MyAnalysis.hh"

MyRunAction::MyRunAction(const G4Run* run)
 : G4UserRunAction()
{
  // Create analysis manager
  G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();
  analysisManager->SetVerboseLevel(1);
}
```

2) Book (Create) Histograms

- Example of creating one-dimensional histograms

MyRunAction.cc

```
MyRunAction::MyRunAction(const G4Run* run)
 : G4UserRunAction()
{
  // Create or get analysis manager
  // ...

  // Create histograms
  analysisManager->CreateH1("EDep","Energy deposit", 100, 0., 800*MeV);
  analysisManager->CreateH1("TLen","Track length",   100, 0., 100*mm);
}
```

3) Open a File

- Example of opening a file

MyRunAction.cc

```
#include "MyAnalysis.hh"

void MyRunAction::BeginOfRunAction(const G4Run* run)
{
  // Get analysis manager
  G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

  // Open an output file
  analysisManager->OpenFile("MyApplication");
}
```

4) Fill Histograms

- Example of filling one-dimensional histograms

MyEventAction.cc

```
#include "MyAnalysis.hh"

void MyEventAction::EndOfEventAction(const G4Event* event)
{
  // Get analysis manager
  G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

  // Fill histograms
  analysisManager→FillH1(0, fEdep);
  analysisManager→FillH1(1, fTrackLength);
}
```

5) Write & Close a File

- Example of writing & closing a file

MyRunAction.cc

```
#include "MyAnalysis.hh"

void MyRunAction::EndOfRunAction(const G4Run* run)
{
  // Get analysis manager
  G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

  // Write and close the output file
  analysisManager->Write();
  analysisManager->CloseFile();
}
```

6) Delete Analysis Manager

- The analysis manager is deleted in RunAction destructor

MyRunAction.cc

```
#include "MyAnalysis.hh"

MyRunAction::~MyRunAction()
{
  // Delete analysis manager
  delete G4AnalysisManager::Instance();
}
```

Selection of the Output Type

- For a simplicity of use, G4AnalysisManager provides the complete access to all interfaced functions for all output formats: **ROOT, CSV, AIDA XML, HDF5**
  - Though it is implemented via a more complex design.
  - The real type is different for each output type: G4CsvAnalysisManger, G4RootAnalysisManger, G4XmlAnalysisManger
- The generic types are defined in dedicated header files for each output type:
  - g4root.hh, g4csv.hh, g4xml.hh, g4hdf5.hh
  - Using **namespaces** and **typedefs**
- It is recommended to add the selected include in an extra header file MyAnalysis.hh and include this header file in all classes which use g4analysis

MyAnalysis.hh

```
#ifndef MyAnalysis_h
#define MyAnalysis_h 1

#include "g4root.hh"
//#include "g4csv.hh"
//#include "g4xml.hh"
#endif
```

Histograms

- 1D, 2D, 3D histograms and 1D, 2D profile histograms available
- Histogram identifiers
  - The histogram identifier (an integer value) is automatically generated when a histogram is created by G4AnalysisManager::CreateH1(), and its value is returned from this function.
    - It is used eg. in G4AnalysisManager::FillH1(id, value)
  - The default start value 0 can be changed (eg. to 1) with the G4AnalysisManager::SetFirstH1Id(G4int) method.
  - The 1D, 2D and 3D histograms IDs are defined independently
- Histogram objects
  - It is also possible to access directly the histogram by G4AnalysisManager::GetH1(G4int id). The concrete histogram type is hidden behind a selected namespace

```
G4cout << "Print histograms statistic \n" << G4endl;
G4cout << " EAbs : mean = " << analysisManager->GetH1(1)->mean()
       << " rms = " << analysisManager->GetH1(1)->rms() << G4endl;
```

Histogram Options

- The properties, additional to those defined in g4tools, can be added to histograms via G4AnalysisManager
  - Unit: if defined, all filled values are automatically converted to this defined unit
  - Function: if defined, the function is automatically executed on the filled values (can be log, log10, exp)
  - Binning scheme: users can define a non-equidistant binning scheme (passing a vector of bin edges)
  - ASCII option: if activated the histogram is also printed in an ASCII file when G4AnalysisManager::Write() function is called.
  - Activation: users can activate/inactivate selected histograms

Book (Create) an Ntuple

- Example of creating an ntuple

MyRunAction.cc

```
MyRunAction::MyRunAction(const G4Run* run)
 : G4UserRunAction()
{
  // Create or get analysis manager
  // ...
  // Create ntuple
  analysisManager->CreateNtuple("MyNtuple", "Edep and TrackLength");
  analysisManager->CreateNtupleDColumn("Eabs");
  analysisManager->CreateNtupleDColumn("Labs");
  analysisManager->FinishNtuple();
}
```

Fill an Ntuple

- Example of filling an ntuple

MyEventAction.cc

```
void MyEventAction::EndOfEventAction(const G4Event* event)
{
  // Get analysis manager
  G4AnalysisManager* analysisManager = G4AnalysisManager::Instance();

  // Fill ntuple
  analysisManager->FillNtupleDColumn(0, fEnergyAbs);
  analysisManager->FillNtupleDColumn(1, fTrackLAbs);
  analysisManager->AddNtupleRow();
}
```

Ntuples

- Ntuple and Ntuple Column Identifiers
  - Automatically generated when the ntuple or ntuple column is created by G4AnalysisManager::CreateNtuple() or G4AnalysisManager::CreateNtupleTColumn() and its value is returned from this function.
  - The default start value 0 can be again changed with the G4AnalysisManager::SetFirstNtupleId(G4int) and G4AnalysisManager::SetFirstNtupleColumnId(G4int)methods.
  - In a similar way as for histogram ID
- The ntuple column ID is not specific to the ntuple column type
- Available column types:
  - Integer (I), float (F), double (D), string (S)
  - std::vector of integer (I), float (F), double (D) types

Output File(s)

- Depending on selected file format, multiple output files can be produced
- ROOT, HDF5
  - All histograms, profiles and ntuples are written in one file
- XML
  - The histograms and profiles are written in one file
  - Each ntuple is written in a separate file
- CSV
  - Each histogram, profile and ntuple are written in a separate file
- File names are generated automatically:
  - fileName[_objectName].ext
    - where ext = xml, csv

Analysis UI Commands

- General options

```
# Set verbose level
/analysis/verbose level
# Set activation option
/analysis/setActivation true|false
```

- Handling output files and general options

```
# Set name for the histograms and ntuple file
/analysis/setFileName name
# Set name for the histograms/ntuple directory
/analysis/setHistoDirName name
/analysis/setNtupleDirName name
```

Analysis UI Commands (2)

- Commands to create or define 1D histogram:

```
# Create 1D histogram
/analysis/h1/create name title [nbin min max] [unit] [fcn]
# Set histogram parameters
/analysis/h1/set id nbin min max [unit] [fcn]
```

- Example of a macro gammaSpectrum.mac in TestEm5 example:

```
/analysis/setFileName gammaSpectrum
/analysis/h1/set  3 200 0.01 10 MeV       #gamma: energy at vertex
/analysis/h1/set  5 200 0.01 10 MeV log10 #gamma: energy at vertex (log10)
/analysis/h1/set 20 200 0 6 MeV           #gamma: energy at exit
/analysis/h1/set 40 200 0 6 MeV           #gamma: energy at back
```

- Analogous commands are available for 2D and 3D histograms and 1D and 2D profiles under h2, h3, p1 and p2 directories

Analysis UI Commands (3)

- The commands for 1D histogram control:

```
# Activate printing 1D histogram on ASCII file
/analysis/h1/setAscii id true|false
# Set title for the 1D histogram
/analysis/h1/setTitle id title
# Set x-axis, y-axis title for the 1D histogram
/analysis/h1/setXaxis id title
/analysis/h1/setYaxis id title
# Set activation for the 1D histogram
/analysis/h1/setActivation id true|false
# Set activation to all 1D histograms
/analysis/h1/setActivationToAll  true|false
```

- The same sets of commands are available for 2D and 3D histograms and 1D and 2D profiles, under h2, h3, p1 and p2 directories

Analysis Reader

- Since Version 10.1
- Allow to read in g4analysis objects from the files generated by the analysis manager(s) during processing Geant4 application.
- Available for each supported output format
- Generic types defined in the same way as for analysis managers:
  - G4AnalysisReader: the public reader interface
- The histograms and profiles objects handled by analysis reader are of the same type as those handled by analysis manager, the ntuple objects are of different types

Batch Graphics

- Since Version 10.2
- Users can activate plotting of selected histograms and profiles using G4AnalysisManager functions:

```
// Activate plotting of 1D histogram
analysisManager->SetH1Plotting(id, true);
   // etc for H2, H3, P1, P2
```

- Or via UI command

```
/analysis/h1/setPlotting id true|false
/analysis/h1/setPlottingToAll  true|false
   ## etc for h2, h3, p1, p2
```

- The selected objects will be plotted in a single postscript file with the page size fixed to A4 format

Plotting Style

- Set plotting style

```
/analysis/plot/setStyle  styleName
```

  - ROOT_default (default), hippodraw: high resolution fonts
  - inlib_default: low resolution fonts
  - High resolution fonts are available only if Geant4 libraries are built with the support for Freetype font rendering

- The page layout

```
/analysis/plot/setLayout  columns rows
```

  - The number columns and the number of rows in a page.
  - The maximum number of plots is limited according to selected style.

Viewing/Processing Resulted Files

- The analysis tool allow to fill histograms and/or ntuples and save them in files of supported formats:
    - ROOT, XML, CSV, HDF5
- Users Geant4 application need not to be linked with the external analysis tools in order to use the Geant4 analysis tool and produce the file(s) with histograms and/or ntuples
- Since version 10.2 it is also possible to activate batch graphics
- *The analysis tools have to be installed on the users machines in order to view interactively or process the analysis of the data in the generated files*

Analysis of Generated Files
With External Tools

Plotting ROOT files
... with ROOT

ROOT
https://root.cern.ch/

ROOT is a powerful analysis tools which provides

- histogramming and graphing to view and analyze distributions and functions
- curve fitting (regression analysis) and minimization of functionals, statistics tools used for data analysis,
- matrix algebra, four-vector computations, standard mathematical functions, multivariate data analysis, e.g. using neural networks,
- persistence and serialization of objects, which can cope with changes in class definitions of persistent data, creating files in various graphics formats, like PostScript, PNG, SVG
- 3D visualizations (geometry), image manipulation, used, for instance, to analyze astronomical pictures
- access to distributed data (in the context of the Grid), distributed computing, to parallelize data analyses, access to databases,
- ...

Viewing ROOT Files

- Start ROOT session

```
$> root
```

- Open a ROOT browser in the ROOT interactive shell

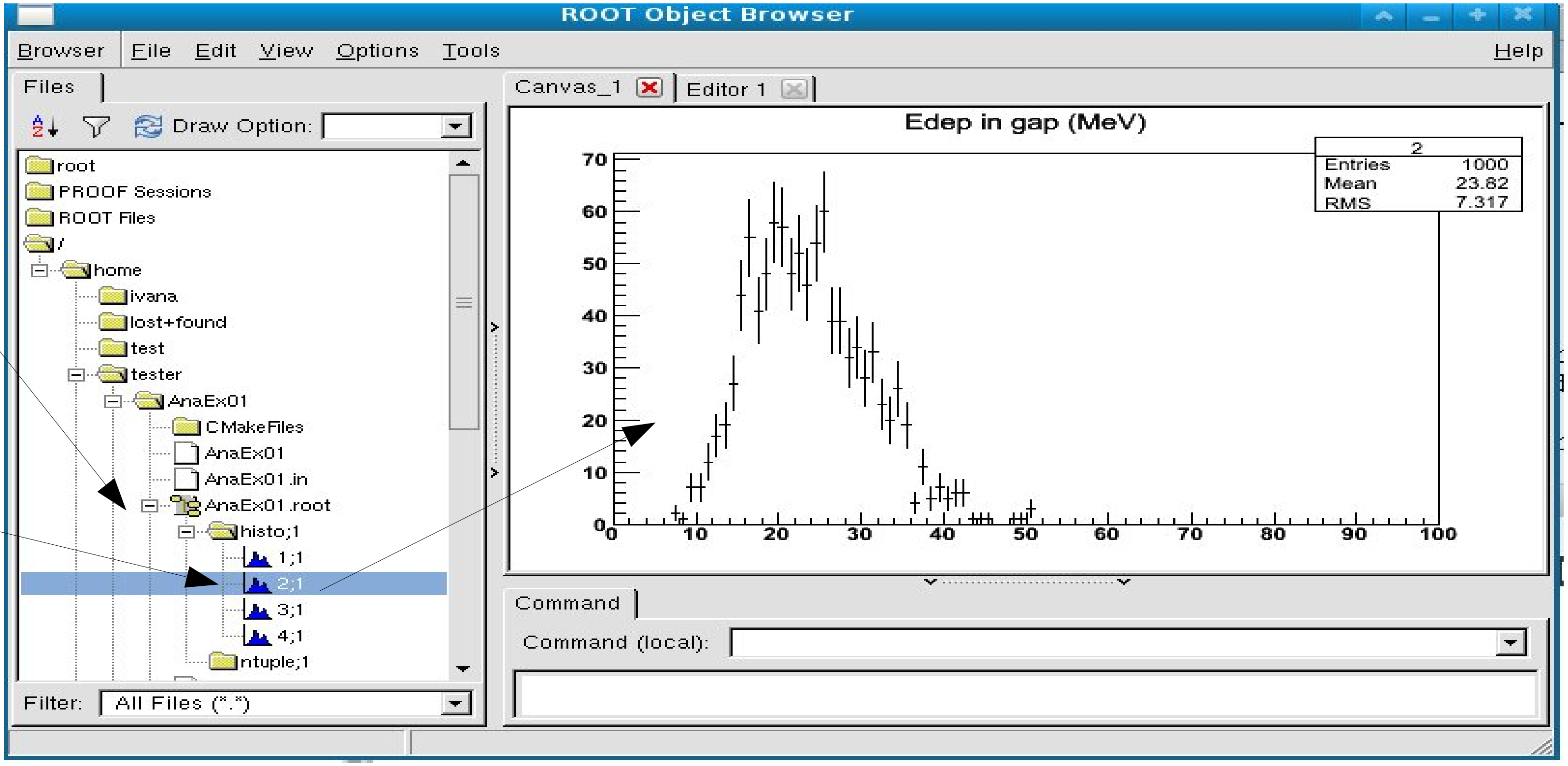
```
root [0] TBrowser b;
```

- See ROOT documentation
  - How to edit histogram properties
  - How to open Fit panel
  - How to write ROOT macros

Viewing ROOT Files (2)

File generated In Geant4 simulation

Selected H1 is automatically drawn in the canvas



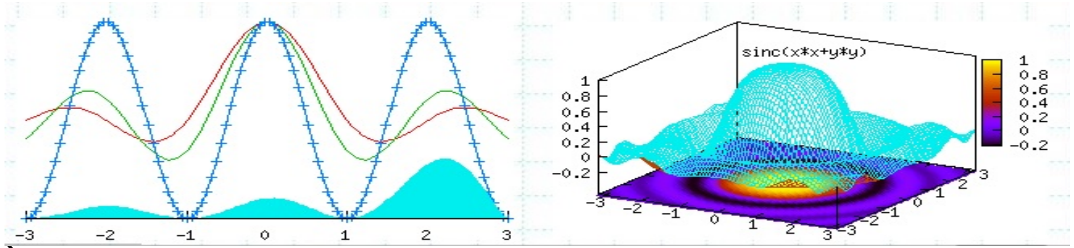
2	
Entries	1000
Mean	23.82
RMS	7.317

Analysis of Generated Files With External Tools

Plotting CSV Files GNUplot, Excel, Open[Libre]Office

Plotting CSV Files

- Gnuplot is a portable command-line driven interactive data and function plotting utility
  - http://www.gnuplot.info/
- Excel:
  - The .cvs file can be imported as a text file and then processed as the data in spreadsheet
- Open[Libre]Office
  - The .csv file can be open from the "File" menu as "Text CSV" file
  - http://www.openoffice.org/
  - http://www.libreoffice.org



Examples of plots from Gnuplot

Thanks to J. Perl, M. Kelsey (SLAC)

Analysis of Generated Files With External Tools

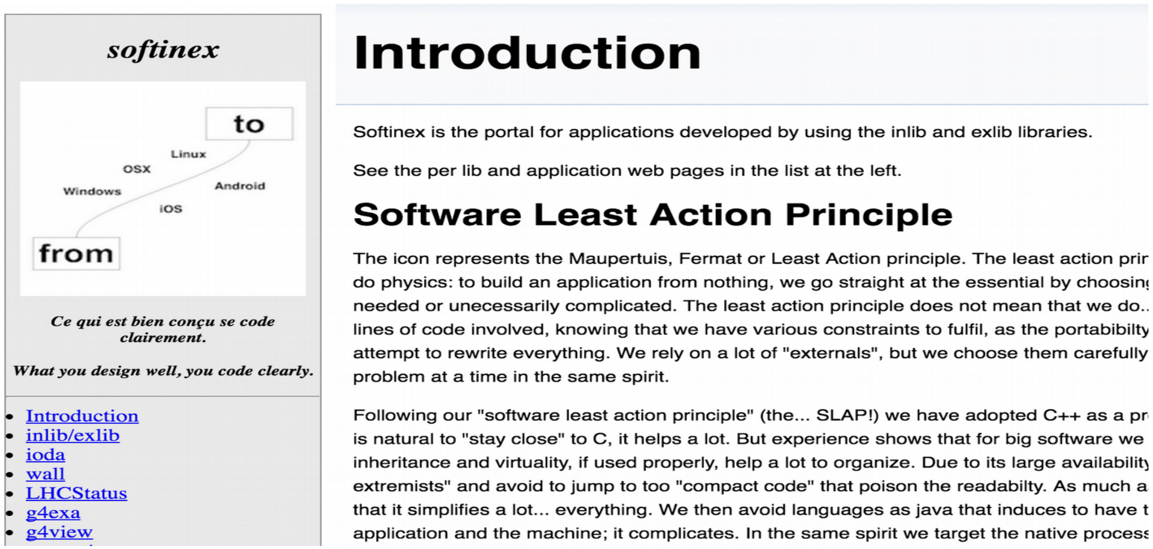
Plotting XML Files SoftInex

Using External Analysis Tools In a Geant4 Application

ROOT, AIDA

Using softinex

- Detailed guidance on integrating with Geant4 available at http://softinex.lal.in2p3.fr/



softinex

Ce qui est bien conçu se code clairement.

What you design well, you code clearly.

- Introduction
- inlib/exlib
- ioda
- wall
- LHCStatus
- g4exa
- g4view

Introduction

Softinex is the portal for applications developed by using the inlib and exlib libraries.

See the per lib and application web pages in the list at the left.

Software Least Action Principle

The icon represents the Maupertuis, Fermat or Least Action principle. The least action prir
do physics: to build an application from nothing, we go straight at the essential by choosin
needed or unecessarily complicated. The least action principle does not mean that we do..
lines of code involved, knowing that we have various constraints to fulfil, as the portabibilty
attempt to rewrite everything. We rely on a lot of "externals", but we choose them carefully
problem at a time in the same spirit.

Following our "software least action principle" (the... SLAP!) we have adopted C++ as a pr
is natural to "stay close" to C, it helps a lot. But experience shows that for big software we
inheritance and virtuality, if used properly, help a lot to organize. Due to its large availability
extremists" and avoid to jump to too "compact code" that poison the readabilty. As much a
that it simplifies a lot... everything. We then avoid languages as java that induces to have t
application and the machine; it complicates. In the same spirit we target the native process

- including support for using in Windows
- A successor of OpenScientist

Examples

- In examples/extended/analysis
- analysis/AnaEx01, 02, 03 – examples to demonstrate how to make histograms and ntuples
  - http://geant4.web.cern.ch/geant4/UserDocumentation/Doxygen/examples_doc/html/Examples_analysis.html (link)
  - AnaEx01 – use of **Geant4 analysis tools**
  - AnaEx02 – use of **ROOT** classes, requires linking with Root libraries
  - AnaEx02 – use of **AIDA** interface classes, requires linking with an AIDA compliant tools, eg. OpenScientist

Summary

- Geant4 provides a lightweight analysis tools as part of distribution
- The Geant4 analysis is now used in all basic, extended and most of advanced examples
- Users can also choose to use an external package and link their application against its libraries