# NPTool: a Root and Geant4 framework for nuclear physics experiments

Adrien MATTA [a], Nicolas de Sereville [b], Marc Labiche [c]

[a]Department of Physics, University of Surrey
[b]Institut de Physique Nucleaire Orsay
[c]NPG, STFC Daresbury Laboratory

IPNOrsay,
Wednesday 26th November 2014

# Philosophy behind the project

## early 2009 : Starting point

- need for an universal simulation and analysis package
- all detector must be treated on an equal footing
- difficult balance of homogeneity/flexibility
- package seen as a tool box for experimentalist

# Philosophy behind the project

## early 2009 : Starting point

- need for an universal simulation and analysis package
- all detector must be treated on an equal footing
- difficult balance of homogeneity/flexibility
- package seen as a tool box for experimentalist

## User perspective

- Easy to use
- Dependable
- Readily available (easy to install, easy to run, fully supported)
- Common framework for the community
- Take full advantage of Root and Geant4
- Link simulation and analysis in a common package

# Structure

## NPLib

→ **Everything common to all experiment**

- Shared library to be used in Analysis, Simulation and macro
- Data class to store "Raw" data
- Physics class to store "Treated" data
- Event Generator
- Managers (Calibration, Root I/O,...)

# Structure

## NPLib

→ **Everything common to all experiment**

- Shared library to be used in Analysis, Simulation and macro
- Data class to store "Raw" data
- Physics class to store "Treated" data
- Event Generator
- Managers (Calibration, Root I/O,...)

## NPSimulation

→ **Everything about Simulation**

- Detector Manager
- Event Generator Manager
- One class per detector

# Structure

## NPLib

→ **Everything common to all experiment**

- Shared library to be used in Analysis, Simulation and macro
- Data class to store "Raw" data
- Physics class to store "Treated" data
- Event Generator
- Managers (Calibration, Root I/O,...)

## NPSimulation

→ **Everything about Simulation**

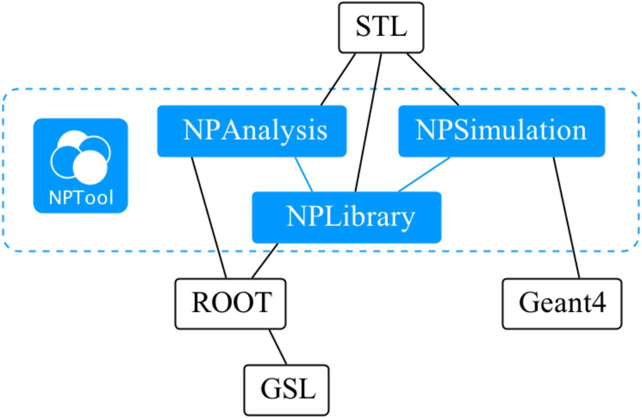
- Detector Manager
- Event Generator Manager
- One class per detector

## NPAnalysis

→ **Specific to an experiment**

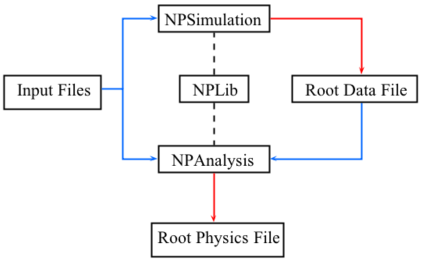
- One project per experiment
- User implementation
- High versatility

# Structure


STL
NPTool
NPAnalysis
NPSimulation
NPLibrary
ROOT
Geant4
GSL

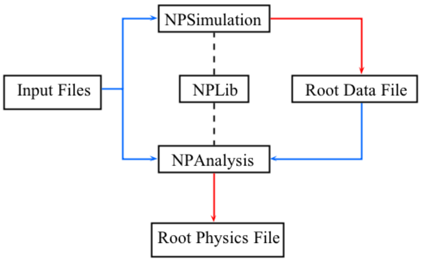
# Way to use it offline

## Simulation of an experiment

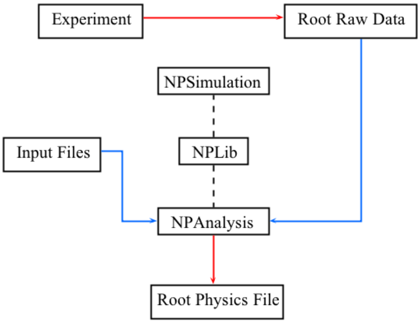


# Way to use it offline

## Simulation of an experiment

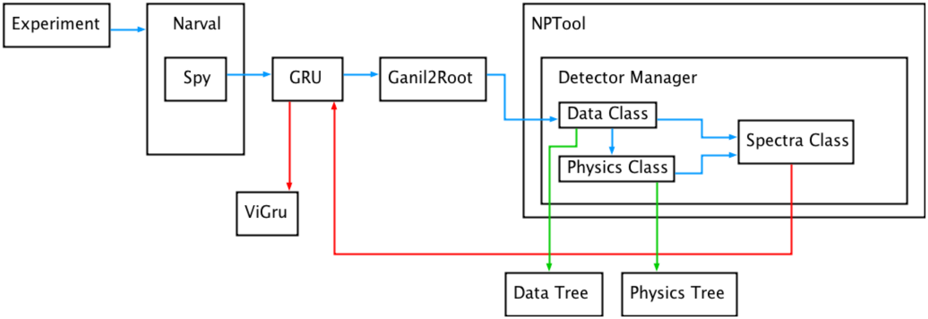



## Analysis of an experiment



# Way to use it online

## Online integration at GANIL



# Detectors

## Available detectors

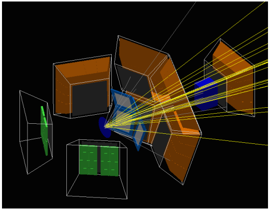
- Must2
- Sharc
- Tiara
- Gaspard
- Paris
- "Helios" (UK)
- Hyde
- Various Silicon (S1,W1,...)

## Work in progress:

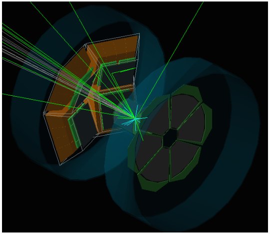
- Exogam
- Tigress
- Spice
- Eden
- T-Rex
- Miniball

# Detectors

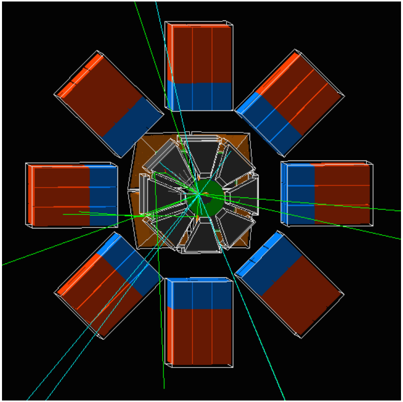
## RRC66 (Riken):



## e628 (Ganil):



## Paris-Gaspard-Must2 (Ganil):



# Adding new detectors

## in NPL

Describe the Root Data format

- TDetectorData class → Use of std::vector
- TDetectorPhysics class → Use of std::vector

# Adding new detectors

## in NPL

Describe the Root Data format

- TDetectorData class → Use of std::vector
- TDetectorPhysics class → Use of std::vector

## in NPS

Inherited from VDetector class:

- ReadConfiguration
- ConstructDetector
- ReadSensitive
- InitializeRootOutput
- InitializeScorers

The DetectorManager:

# Adding new detectors

## in NPL

Describe the Root Data format

- TDetectorData class → Use of std::vector
- TDetectorPhysics class → Use of std::vector

## in NPS

Inherited from VDetector class:

- ReadConfiguration
- ConstructDetector
- ReadSensitive
- InitializeRootOutput
- InitializeScorers

The DetectorManager:

- Declare your detector
- Manage a std::vector<VDetector>
- Take care of calling methods

# Event Generators

## Physics Case

- Beam
- Source
- Gamma decay
- Particle decay
- Two body reaction
- Phase space reaction

# Event Generators

## Physics Case

- Beam
- Source
- Gamma decay
- Particle decay
- Two body reaction
- Phase space reaction

## Versatility

- User cross section
- Beam emmitance or exp. distr.
- Daisy chain

# Event Generators

## Physics Case

- Beam
- Source
- Gamma decay
- Particle decay
- Two body reaction
- Phase space reaction

## Versatility

- User cross section
- Beam emmitance or exp. distr.
- Daisy chain

## Example

```
Beam
   Particle= 20Ne
   Energy= 50
   .
   .
   .
%%%%%%%%%%%%%%%
TwoBodyReaction
   Beam= 20Ne
   Target= 12C
   Light= 4He
   Heavy= 28Si
   ExcitationEnergyLight= 0.0
   ExcitationEnergyHeavy= 10.0
   CrossSectionPath= flat.txt CS10He
   ShootLight= 1
   ShootHeavy= 1
%%%%%%%%%%%%%%%%
ParticleDecay 28Si
   Daughter= 24Mg 4He
   ExcitationEnergy= 0.5 0
   DifferentialCrossSection= TGenPhaseSpace particle28Si
   shoot= 1 1
%%%%%%%%%%%%%%%%
GammaDecay 24Mg
   Cascade
      BranchingRatio= 70
      Energies= 0.2 0.3
   Cascade
      BranchingRatio= 30
      DifferentialCrossSection= Gamma.txt gamma24Mg
      Energies= 0.5
```

# Adding new event generators

## in NPS

Inherited from VEventGenerator class:

- ReadConfiguration
- GenerateEvent
- InitializeRootOutput

The EventGeneratorManager:

# Adding new event generators

## in NPS

Inherited from VEventGenerator class:

- ReadConfiguration
- GenerateEvent
- InitializeRootOutput

The EventGeneratorManager:

- Declare your event generator
- Manage a std::vector<VEventGenerator>
- Take care of calling methods

# Simulation and Analysis

- Input File saved in the Output
- Calibration manager
- Root Input/Output manager
- Custom Threshold
- Custom channel desabling

# Simulation and Analysis

- Input File saved in the Output
- Calibration manager
- Root Input/Output manager
- Custom Threshold
- Custom channel desabling

## Configuration file example for NPA

ConfigMust2
  MAX_STRIP_MULTIPLICITY 3
  STRIP_ENERGY_MATCHING_NUMBER_OF_SIGMA 2
  STRIP_ENERGY_MATCHING_SIGMA 0.02
  DISABLE_CHANNEL MM1STRY12
  DISABLE_ALL MM5

# Installing NPTool

## Requierement

- GSL
- Root
- Geant4 (4.9.6)

# Installing NPTool

## Requierement

- GSL
- Root
- Geant4 (4.9.6)

## Downloading the source

- web site : https://github.com/adrien-matta/nptool
- direct download
- git clone

# Installing NPTool

## Requierement

- GSL
- Root
- Geant4 (4.9.6)

## Downloading the source

- web site : https://github.com/adrien-matta/nptool
- direct download
- git clone

## A few easy steps (to be found on the website)

- Defining environment variable
- Building NPL
- Building NPS
- Creating your NPA project

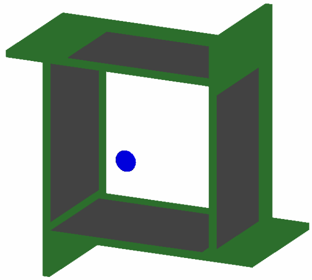
# Running a Simulation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Target
  Thickness= 0.001
  Radius= 5
  Material= CD2
  Angle= 0
  X= 0
  Y= 0
  Z= 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Upstream Box
SharcBOX
  Z= -30
  ThicknessDector1= 100
  ThicknessDector2= 100
  ThicknessDector3= 100
  ThicknessDector4= 100
```

# Running a Simulation

```
%%%%%%%%%%%%%%%%%%%
Target
   Thickness= 0.001
   Radius= 5
   Material= CD2
   Angle= 0
   X= 0
   Y= 0
   Z= 0
%%%%%%%%%%%%%%%%%%%
%Upstream Box
SharcBOX
   Z= -30
   ThicknessDector1= 100
   ThicknessDector2= 100
   ThicknessDector3= 100
   ThicknessDector4= 100
```

## Geant4 Geometry

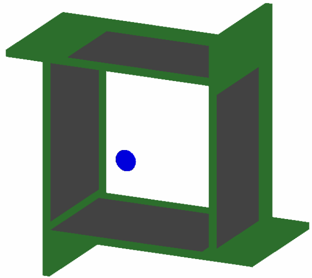


# Running a Simulation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Target
   Thickness= 0.001
   Radius= 5
   Material= CD2
   Angle= 0
   X= 0
   Y= 0
   Z= 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Upstream Box
SharcBOX
   Z= -30
   ThicknessDector1= 100
   ThicknessDector2= 100
   ThicknessDector3= 100
   ThicknessDector4= 100
```

```
Beam
   Particle= 11Li
   Energy= 550
   EnergyProfilePath= Energy.root Energy
   XThetaXProfilePath= XThetaX.root XThetaX
   YPhiYProfilePath= YPhiY.root YPhiY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TwoBodyReaction
   Beam= 11Li
   Target= 2H
   Light= 3He
   Heavy= 10He
   ExcitationEnergyLight= 0.0
   ExcitationEnergyHeavy= 0.0
   CrossSectionPath= flat.txt CS10He
   ShootLight= 0
   ShootHeavy= 1
```

## Geant4 Geometry

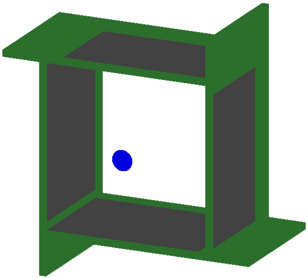


# Running a Simulation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Target
   Thickness= 0.001
   Radius= 5
   Material= CD2
   Angle= 0
   X= 0
   Y= 0
   Z= 0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Upstream Box
SharcBOX
   Z= -30
   ThicknessDector1= 100
   ThicknessDector2= 100
   ThicknessDector3= 100
   ThicknessDector4= 100
```

```
Beam
   Particle= 11Li
   Energy= 550
   EnergyProfilePath= Energy.root Energy
   XThetaXProfilePath= XThetaX.root XThetaX
   YPhiYProfilePath= YPhiY.root YPhiY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TwoBodyReaction
   Beam= 11Li
   Target= 2H
   Light= 3He
   Heavy= 10He
   ExcitationEnergyLight= 0.0
   ExcitationEnergyHeavy= 0.0
   CrossSectionPath= flat.txt CS10He
   ShootLight= 0
   ShootHeavy= 1
```

## Geant4 Geometry



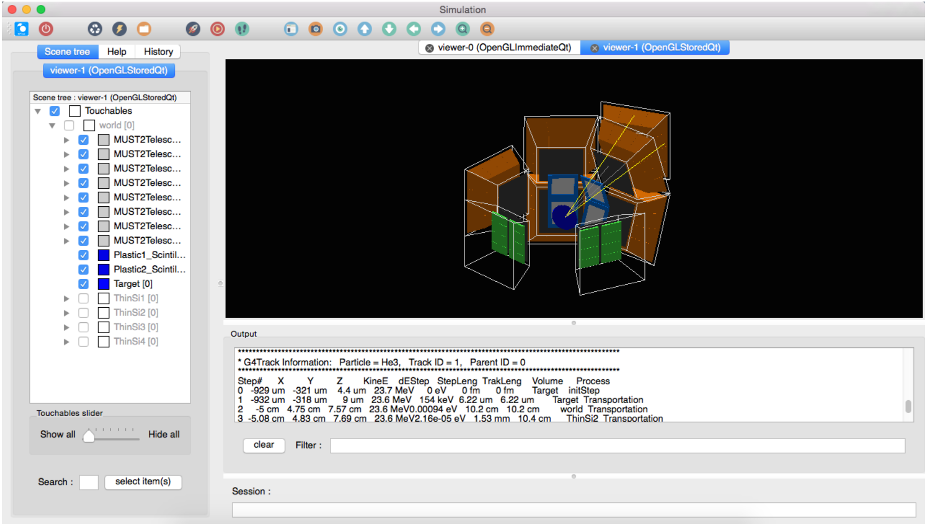
## A Single command line

Simulation -D Detector.txt -E Reaction.txt

→ Launch the custom Qt interface

# Running a Simulation

## Geant4 Qt Interface



# Running an Analysis

## A few ingredients

- Same detector file
- Same reaction file
- List of run
- Optional calibrations file

# Running an Analysis

## A few ingredients

- Same detector file
- Same reaction file
- List of run
- Optional calibrations file

## What it does

- Automatic root file reading
- Automatic detector treatment
- Automatic calibration reading

# Running an Analysis

## A few ingredients

- Same detector file
- Same reaction file
- List of run
- Optional calibrations file

## What it does

- Automatic root file reading
- Automatic detector treatment
- Automatic calibration reading

## What you have to do

Using the detector informations:

- Energy loss correction
- Doppler correction
- Excitation energy calculation
- ...

All this is done using NPLib objects
Energy loss table from NPSimulation

# Running an Analysis

## A few ingredients

- Same detector file
- Same reaction file
- List of run
- Optional calibrations file

## What it does

- Automatic root file reading
- Automatic detector treatment
- Automatic calibration reading

## What you have to do

Using the detector informations:

- Energy loss correction
- Doppler correction
- Excitation energy calculation
- ...

All this is done using NPLib objects
Energy loss table from NPSimulation

## A Single command line

Analysis -D Detector.txt -E Reaction.txt -R RunToTreat.txt -C Calibration.txt

→ Stand alone program, output a ROOT Tree and/or Histo

# Recent improvements

## Recent improvements

- Simpler Scorer
- Detailed geometry
- Selective compilation
- Selective library load
- Material manager
- Improved energy loss table generation
- Improved Qt interface
- Detector cloner for faster implementation

# Perspectives

## NPL

- More physics case
- General class library for calibration
- Strengthen the Online capabilities

# Perspectives

## NPL

- More physics case
- General class library for calibration
- Strengthen the Online capabilities

## NPS

- More Detector
- Simplified code
- Modular physics list

# Perspectives

## NPL

- More physics case
- General class library for calibration
- Strengthen the Online capabilities

## NPS

- More Detector
- Simplified code
- Modular physics list

## NPA

- Derived from TSelector
- Parallelised analysis with pROOF

# Thank you

## They are using it

- University of Surrey (UK)
- IPNOrsay (France)
- NPG STFC Daresbury (UK)
- CEA Saclay (France)
- GANIL (France)
- Riken (Japan)
- Triumf (Canada)
- Universidad de Santiago (Spain)
- Hanoi University of Technology (Vietnam)

## Numbers:

- 881 commit since 2009
- $\sim$ 40k lines of code
- 85% C++ / 12% bash
- 10 GitHub contributor
- 13 detectors NPSimulation
- 29 detectors in NPLib
- 9 PhDs