Journal of Physics G: Nuclear and Particle Physics

PAPER

NPTool: a simulation and analysis framework for low-energy nuclear physics experiments

To cite this article: A Matta *et al* 2016 *J. Phys. G: Nucl. Part. Phys.* **43** 045113

IOP Publishing Journal of Physics G: Nuclear and Particle Physics

J. Phys. G: Nucl. Part. Phys. **43** (2016) 045113 (15pp) doi:10.1088/0954-3899/43/4/045113

NPTool: a simulation and analysis framework for low-energy nuclear physics experiments

A Matta[1,6], P Morfouace[2], N de Séréville[3], F Flavigny[3], M Labiche[4] and R Shearman[1,5]

[1] Department of Physics, University of Surrey, Guildford, Surrey, GU2 7XH, UK
[2] National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, USA
[3] Institut de Physique Nucléaire, CNRS-IN2P3, Univ. Paris-Sud, Université Paris-Saclay, 91406 Orsay Cedex, France
[4] Science and Technology Facilities Council, Daresbury Laboratory, Sci-Tech Daresbury, Daresbury, Warrington, WA4 4AD
[5] National Physical Laboratory, Teddington, Middlesex, TW11 0LW, UK

E-mail: a.matta@surrey.ac.uk, morfouac@nscl.msu.edu, deserevi@ipno.in2p3.fr, flavigny@ipno.in2p3.fr, marc.labiche@stfc.ac.uk and r.shearman@surrey.ac.uk

Received 7 January 2016
Accepted for publication 29 January 2016
Published 7 March 2016

Abstract

The Nuclear Physics Tool (NPTool) is an open source data analysis and Monte Carlo simulation framework that has been developed for low-energy nuclear physics experiments with an emphasis on radioactive beam experiments. The NPTool offers a unified framework for designing, preparing and analyzing complex experiments employing multiple detectors, each of which may comprise some hundreds of channels. The framework has been successfully used for the analysis and simulation of experiments at facilities including GANIL, RIKEN, ALTO and TRIUMF, using both stable and radioactive beams. This paper details the NPTool philosophy together with an overview of the workflow. The framework has been benchmarked through the comparison of simulated and experimental data for a variety of detectors used in charged particle and gamma-ray spectroscopy.

Keywords: data analysis, Monte Carlo Simulation, nuclear physics, nuclear reaction

(Some figures may appear in colour only in the online journal)

[6] Author to whom any correspondence should be addressed.

1. Introduction

With the advent of radioactive ion beam facilities, the last two decades have seen a steep rise in the complexity of low-energy nuclear physics experimental setups. This is mainly due to the large increase of electronic channels together with the use of multiple detection systems in order to perform the most complete study of a nuclear reaction in a single experiment. Indeed, it is more and more common to find setups where charged particle detectors are coupled to γ-ray and neutron detectors while the recoil nucleus is detected by a complex detection system at the focal plane of a magnetic spectrometer. In most cases a beam tracker system is also needed to reconstruct the impact position and incident direction of the beam particle at the target position.

This complexity leads to a longer time spent on developing dedicated analysis and simulation tools. Nowadays, experiments are typically analyzed by a handful of people (two or three) looking at different reaction channels from the same data set. Each person tends to develop their own analysis code, often starting from scratch. This approach has the clear benefit of developing a code exactly suited for the physics case under study, and to be able to cross check the analysis results if several persons analyze the same subset of data. However, this is done at the expense of efficiency, since the same problems may have to be solved several times by different people. Similar issues arise with Monte Carlo simulations of a setup, which are usually very specific both in terms of geometry and physics case. Having a large community working together on a single tool guarantees obtaining a more reliable one within a shorter development time.

Currently there are numerous projects focusing either on the simulation or the analysis of a specific detector, with no link between these two aspects. Only a few are progressively trying to fill this gap by defining a general framework where both the analysis and the simulation of a detector (or set of detectors) is provided in a consistent way. The FairRoot framework [1], which started at about the same time as the NPTool, is a good example of such a common approach but is somewhat more focused toward facility and experiments having needs closer to intermediate/high-energy physics than low-energy physics. The code is designed to deal with more or less rigid setups, whereas the NPTool is designed to be a modular platform, allowing the combination of different detectors at will. Each experiment in the low-energy nuclear physics community uses a different setup, recombining existing detector systems at different positions around the target. The NPTool was designed with that need of flexibility in mind, and changing a setup and mixing different detector systems is done simply by changing a few lines in the input file. KaliVeda [2] is another case where an important effort has been made in obtaining a common framework for simulating and analysing the INDRA [3] and FAZIA [4] charged particle arrays. Those arrays, being again relatively fixed setups, mean that the KaliVeda package has a limited focus on flexibility. To our knowledge, there is no existing framework adapted to low-energy physics where it is possible to: (i) couple existing analysis and simulation codes; (ii) add support for new detectors easily; (iii) be able to combine all existing detectors at will.

Clearly, analysis and Monte Carlo simulation projects which are flexible enough to adapt to the large variety of detectors used in the low-energy nuclear physics field, but rigid enough to bring consistency and re-usability, are needed. Furthermore, the need is to have a project which has specific event generators (two-body and n-body cases, γ-ray cascades, etc) and tools, such as a calibration manager facility, adapted to the low-energy community. Object-oriented technology appeared to be the best choice to ensure high modularity and re-usability for such an analysis and a Monte Carlo simulation framework. Therefore we selected the

Geant4 [5] Monte Carlo simulation toolkit and the ROOT [6] data analysis framework, both developed in C++, on top of which the NPTool is based.

NPTool development started in 2009 for the analysis of the MUST2 charged particle detector array closely followed by Monte Carlo simulations for the next generation GASPARD array. It has since then been used successfully for the analysis and simulation of complex experiments at GANIL, RIKEN and TRIUMF. A public release of the NPTool licensed under the GNU Public License 2.0 can be downloaded at https://github.com/adrien-matta/nptool . Comprehensive documentation on how to install and use the framework is also available from the project website http://nptool.org.

This paper presents a detailed description of the design and development of an analysis and Monte Carlo simulation framework by the NPTool collaboration which satisfies the requirements mentioned above. This article is based on NPTool framework version 2.

2. NPTool basics

The NPTool was designed with several goals in mind. First, the analysis of data obtained from experiments or generated by Monte Carlo simulations are treated in exactly the same way in the NPTool. This ensures that no potential systematic bias appear in the analysis when comparing experimental and simulation results. Second, the NPTool has been designed as a modular toolbox for the low-energy physics community, making a large set of tools available, a flexible workflow and a well structured framework with which to work. Last, the NPTool should be modular so that it can evolve as new applications are envisioned.

2.1. Philosophy

Usability and modularity have been the driving concepts for developing the NPTool framework. This approach allows us to propose a common workflow to most of the experiments. The amount of time needed to obtain results from an experiment is then significantly reduced since the end user can focus mostly on the physical treatment and specificity of the analyzed data. Since the beginning, efforts have been made to ensure that the NPTool runs optimally on desktop and laptop machines since the low-energy nuclear physics community does not have an easy access to larger computer platforms.

Having usability in mind, the NPTool has been designed to run with only two input files. They describe both the geometry of the setup and the physics of a given experiment, using human readable ASCII files formatted with tokens. These files contain all the important details to run a Monte Carlo simulation and a full analysis, without the need to go in depth into the NPTool internal structure.

The NPTool's high modularity makes it very convenient to add functionalities seamlessly both from the developper and end user point of view. This aspect is particularly important when an end user is facing the situation when a new detector must be included in the framework, when a new analysis should be added or when an event generator should be developed. This modularity allows any end user to participate and to be integrated in NPTool's development, taking benefit of the work of others, while sharing its own development and improvement. This also prevents the framework from becoming a black box.

In order to reach these two goals, software design pattern was used generalising as much code as possible to make it more reliable, readable and easier to maintain. The C++ language and its object-oriented approach appeared as a natural choice to meet the NPTool's needs. To ensure the reliability of the framework, the NPTool is developed and tested on Mac OS X and on various Linux flavours including Ubuntu and Debian. Every time an update, a bug fix or a

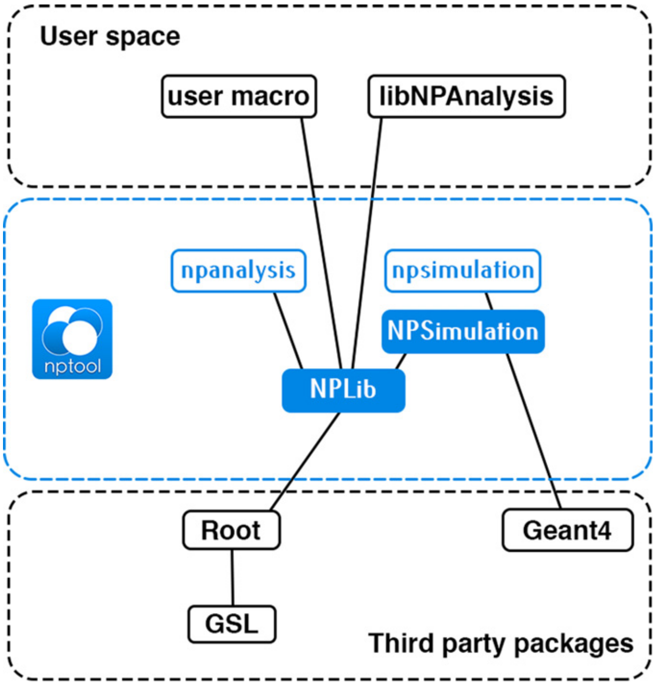

Figure 1. Schematic view of the NPTool framework dependencies (black line). *NPLib* and *NPSimulation* are whole project regrouping several librairies. *npanalysis* and *npsimulation* are executables.

new feature is committed to the framework, a set of standard tests is run automatically detecting any build and run time problems.

2.2. Architecture

The NPTool framework has a layered architecture which is schematically represented in figure 1 where the dependences between the main components are shown. While the core layer (NPLib) defines the essential tools and features of the NPTool framework, the application layer encompasses the *npsimulation* and *npanalysis* programs using both the core functionalities. On top of this application layer is the user layer, where end users analyze data by means of ROOT macros or the *libNPAnalysis* library in the case of more elaborate analysis.

The NPTool *core layer* is called NPLib and has three main functions. First, it provides all the most generic aspect of the NPTool framework. This includes all the abstract base classes and their associated facility such as the *VDetector* and the detector manager. Second, it provides all the detector libraries needed to (i) store the data, (ii) apply the physics analysis associated to each independent detector and (iii) generate associated spectra. Last, the *core layer* provides some physics tool such as relativistic kinematics or energy loss calculations.

The NPTool *application layer* provides the two main NPTool executable files: *npsimulation* and *npanalysis*. While the latter only depends on NPLib, the former also strongly depends on the Geant4 toolkit. All Geant4-related files needed to define the geometry of each detector as well as the event generator are defined in the NPSimulation directory of the distribution.

The *user layer* is the place where the end user analyzes the Monte Carlo simulation files generated by *npsimulation* or data files from an experiment. An important aspect of the NPTool is that since these two kind of files use the same classes as data storage, they are treated in the exact same way by *npanalysis*. Depending on the complexity of the analysis, the end user can decide to use ROOT macros or to build the *libNPAnalysis* library.

The NPTool is designed and optimised to run locally on desktop and laptop machines, with typical data sets up to a few Tb, as is usual in the low-energy nuclear physics community. With this in mind a type of functional multithreading has been included in *npanalysis* where each detector is analyzed in parallel within a single event. This type of parallelisation provides a good middle ground for machines using a single hard drive to readout and write the data and keep the data race problem easier to solve.

2.3. Workflow

When starting a new analysis, good practice would be to create a new directory from where both the *npsimulation* and *npanalysis* utilities will be called. As already mentioned, these utilities use the same two input files for describing the geometrical setup and the kind of physics (reaction, decay, source) at play. This is mandatory to allow a full consistency between the analysis and the simulation. For traceability purposes, these two input files are stored inside the output ROOT file generated by either program.

2.3.1. Running a simulation. Using *npsimulation* is as easy as running the following single command line:

```
$ npsimulation -D geometry_filename.detector
               -E generator_filename.reaction
               -O output_rootfilename
```

where the -D and -E flags specify the geometry and event generator files, respectively. Examples of these files can be found in the *Inputs* directory of the NPTool framework. The -O flag specifies the ROOT output file name. The *npsimulation* utility will launch a customised QT interface if Geant4 was compiled with QT support, otherwise it will run with the usual command line interface. In both cases it is possible to modify and reload the geometry and the event generator files on the fly. OpenGL and VRML visualisations are available and exporting the geometry to the GDML format is available. Modular physics lists are supported and their descriptions will be also saved in the output ROOT file for further reference.

2.3.2. Running an analysis. Running the *npanalysis* utility is as easy as for the *npsimulation* utility. The following single command line could be used:

```
$ npanalysis -D geometry_filename.detector
             -E generator_filename.reaction
             -C calibration_filename.txt
             -R runstotreat_filename.txt
             -O output_rootfilename
```

where the -C and -R flags specify the list of calibration files and the list of files to analyse, respectively. Other flags have the same meaning as before. In the case of the analysis of data files generated with *npsimulation*, the first three options of the previous command line can be omitted since (i) the data are already calibrated and (ii) the geometry and event generator files saved in the ROOT file generated by *npsimulation* will be taken as default. The output ROOT file generated by *npanalysis* will contain a ROOT TTree, called *PhysicsTree*, with a branch for each detector including calibrated and treated data.

If a shared library named *libNPAnalysis* is present in the directory where *npanalysis* is run, additional physical quantities requiring information from several detectors can be computed. For example, one can first get the reconstructed position of the beam on target from a beam tracking detector system, then get the position of interaction and energy of a charged particle from a silicon detector array and finally compute laboratory emission angle, laboratory energy, center-of-mass angle and excitation energy assuming two-body kinematic. All this additional information can be stored in the ROOT output file. The *libNPAnalysis* library is generated when the *Analysis.*{*h,cxx*} files are present; examples can be found in the *Examples* directory of the NPTool distribution.

2.4. Adding a new detector

Due to the NPTool's high modularity, it is straightforward to add a new detector using the *nptool-wizard* facility. Once the executable launches, the user will be prompted to provide authorship, contact email and a detector name. The tool will automatically create the folders and files associated to the new detector both in NPLib and NPSimulation. The new detector classes are registered to the *DetectorManager* class which ensures that the new detector is perfectly functional and integrated in the framework. There is no need to tweak any other files than those created to make the new detector fully operational.

For a new detector called *NewDetector*, the *TNewDetectorData*, *TNewDetectorPhysics* and *TNewDetectorSpectra* classes are created in NPLib. While *TNewDetectorData* is in charge of storing the raw data (uncalibrated) from experiments or Monte Carlo simulations, *TNewDetectorPhysics* performs the calibrations, applies thresholds and performs the physical treatment of the detector. For example, this treatment could be to select events that have the same energy deposit in the two sides of a double sided silicon striped detector (DSSSD), or to compute the centroid of a charge distribution in a position sensitive detector. *TNewDetectorSpectra* is an optional class which generates control spectra that can be used for online applications for example.

In NPSimulation, the *NewDetector* class is created and is in charge of describing the detector geometry and defining the sensitive elements of the geometry where a particle information is recorded. The geometry, compliant with the Geant4 standard, can be either hard coded, which gives excellent start up performances, or loaded from a GDML file, which gives lower startup performances. The GDML support is very convenient for interchanging existing geometries within different projects.

3. Physics and event generators

In the Geant4 framework primary particles are generated by the user via an event generator. These primary particles propagate through the experimental set-up in which they further interact depending on the physics list selected by the user. Depending on the selected physical processes, secondary particles are created and tracked by Geant4. The specificity of the NPTool's physics list and event generators is now presented.

3.1. Physics list

The NPTool uses a modular physics list in a Geant4 fashion. The default reference list is the Geant4 ElectroMagnetic one with Option 4 (*emstandard_opt4*) using the most accurate standard and low-energy models. The user can adjust the secondary particle cut off as well as replacing *emstandard_opt4* by either *livermore* or *penelope* low-energy electromagnetic models. In addition, one can activate hadronic and optical processes from the most relevant reference physics list, including the *Nuclear decay*, *Ion binary cascade*, *Stopping physics*, *Electromagnetic extra physics*, *Optical physics*, *Hadron elastic physics* and *Hadron QGSP BIC HP physics*.

In order to guarantee a full consistency between *npsimulation* and *npanalysis*, an energy loss table file is produced by the former for each couple of charged particles defined in the event generator and material used in the geometry. To analyze real data, the *NPEnergyLoss* facility can use the aforementioned generated table, SRIM [7] tables or LISE++ [8] tables.

3.2. Event generators

The NPTool includes dedicated event generators that have been developed to fulfil the needs of the low-energy nuclear physics community. All NPTool event generators can be daisy chained without restriction so that the output of an event generator can be used as the initial condition for another one. The particle properties (momentum and direction) are calculated prior to the event generator so that all particles are emitted at the beginning of the event.

Due to the NPTool's modularity design, it is straightforward to add a new event generator if a specific physics case is not covered by the existing event generators. A few examples of available event generators in the NPTool are now presented.

3.2.1. Beam. The *Beam* event generator is one of the key NPTool event generators since it is needed as soon as an ion beam interacts in a target material. This event generator allows us to fully characterise the beam emittance and energy profile. This information can either be given by a Gaussian analytical model where its mean and standard deviation are passed as arguments, or using distribution files either in ASCII or ROOT format. While the former approach is well suited for easy simulation of the experimental setup prior to the actual experiment, the latter case is very useful in order to get an exact response of the setup once the beam emittance and energy distribution have been determined during the course of the experiment. In both cases a beam energy, a beam direction and impact position at the entrance of the target are generated event by event. The beam is then slowed down in the target using the Geant4 energy loss tables to an interaction layer which is randomly chosen according to a flat distribution within the target thickness.

3.2.2. Two-body reactions. The *TwoBodyReaction* event generator is used to describe nuclear reactions between an ion beam and a target nucleus where two nuclei are present in the exit channel. The ion beam energy comes from the result of the *Beam* event generator and the emitted nuclei can be produced either in their ground state or any excitation energy. Any two-body reaction is valid as long the mass and charge conservation is respected. The kinematic calculation is fully relativistic and uses tabulated masses [9] to calculate the reaction Q-value. The starting point of the calculation is a center-of-mass angle, then the relevant kinematical information (energy and emission angle with respect to the beam incident direction) of the emitted nuclei is derived after applying the laws of momentum and energy conservation. The emission angle of each outgoing nucleus is then transformed in the laboratory frame.

In order to describe the reaction mechanism (direct reaction, compound nucleus, etc) of the reaction under study, a differential cross-section file as a function of the angle in the center of mass can be specified. In the case of a two-body reaction with a large beam energy spread, it is also possible to specify a double differential cross-section with respect to the center of mass angle and the beam energy. In order to characterise the efficiency of the experimental setup, a simulation can be run using an isotropic differential cross-section file.

3.2.3. Particle and γ-ray decays. Particle and γ-ray decays are simulated using the *ParticleDecay* and *GammaDecay* event generators, respectively. Note that since all the particles are emitted at the beginning of the event, only prompt γ-ray or particle decays are allowed. However, in-flight decay taking into account the lifetime of the decaying particle is taken into account through the physics list. For both particle and γ-ray decays, all calculations are performed in the reference frame of the decaying nucleus and then boosted to the laboratory frame. This procedure implicitly takes into account the relativistic Doppler shift effect in the case of γ-ray decays.

When a single particle or γ-ray is emitted there is the possibility of specifying an angular distribution pattern in the decaying nucleus frame reflecting the transition to the final state.

For γ-ray decays, the case of cascades is fully supported and complex decay schemes can be simulated. The branching ratio of each branch of the cascade should be given together with a list of the γ-ray transition energies.

The case of multi-particle emission (not including γ-rays) is also fully supported. A Monte Carlo phase space calculation is performed using the ROOT phase space algorithm generating an n-body event with a constant cross-section.

4. Benchmarks

Benchmarks play an important role to check the installation or upgrade integrity of the NPTool framework. They are also useful for comparing CPU performances on different computer platforms So far, two benchmarks are included in the NPTool framework. The first one analyzes experimental data from a beam tracker detector using the *npanalysis* facility, while the second one runs a silicon array simulation using the *npsimulation* facility and displays some basics control spectra. Each benchmark produces figures that can be compared to the reference figures provided in the NPTool. These two benchmarks cover all the core functionalities of the NPTool's framework.

4.1. CATS: a multi-wire low-pressure tracking system

CATS consists of two low pressure multi-wire proportional chambers with one plane of anode wires placed between two cathode planes, respectively segmented into 28 vertical or horizontal strips [10]. It was designed to provide event-by-event particle tracking in experiments with radioactive beams at GANIL.

Experimental data for the present benchmark comes from the interaction of a ^{58}Ni ion beam at 75 MeV u^{-1} in the CATS detector. A mask with a known pattern was inserted in front of CATS in order to calibrate the position reconstruction. Results that should be obtained when running the benchmark are shown in figure 2. The mask pattern including circular holes of different diameters is clearly observed. Aside from testing the position reconstruction for the CATS detector, this benchmark validates the calibration manager facility and the *npanalysis* utility.

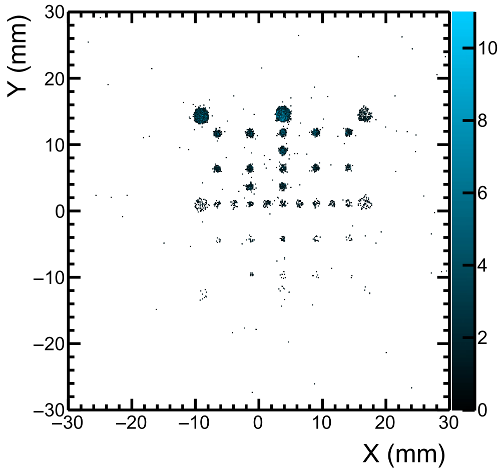

Figure 2. Pattern reconstructed by the CATS detector when a mask is inserted in front of the detector.

4.2. GASPARD: a silicon array for particle spectroscopy

GASPARD [11] aims to be a next generation silicon array with a coverage close to 4π which can be completely integrated with new generation γ-ray detectors such as AGATA [12] or PARIS [13]. While the physics case tackled by GASPARD is broad, it has been designed with the study of nuclear structure by direct reactions in mind.

The benchmark performs a Monte Carlo simulation of the d(^{132}Sn,p)^{133}Sn$_{g.s.}$ reaction where a ^{132}Sn ion beam at 10 MeV u^{-1} interacts in a 5 μm CD_2 target. The protons are generated according to a DWBA differential cross-section and detected in the close to final GASPARD geometry presented in figure 3 (top left). The geometry includes (i) a barrel of eight square silicon telescopes, (ii) two end-caps of eight trapezoid telescopes and (iii) two annular detectors at forward and backward angles. Each telescope has three layers, the first one being a DSSSD with 128 strips on each side. A selection of the results that should be obtained with the analysis macro is displayed in figure 3. This includes the geometrical efficiency of the setup as a function of the angle in the laboratory frame (top right). The beam spatial distribution at the target position (bottom left) is represented together with the target frame materialised by the dotted circle. Last, the number of emitted and detected protons as a function of the center-of-mass angle are represented (bottom right). This benchmark mainly tests the generated and detected information stored event-by-event in the *InitialConditions* and *InteractionCoordinates* classes. These classes are extremely useful for quickly controlling any Monte Carlo simulation.

5. NPTool validation

In this section we present representative cases of analysis performed using the NPTool aiming at a comparison between experimental and simulated data. These examples provide a validation of the physics list used in the NPTool for low energy nuclear physics.

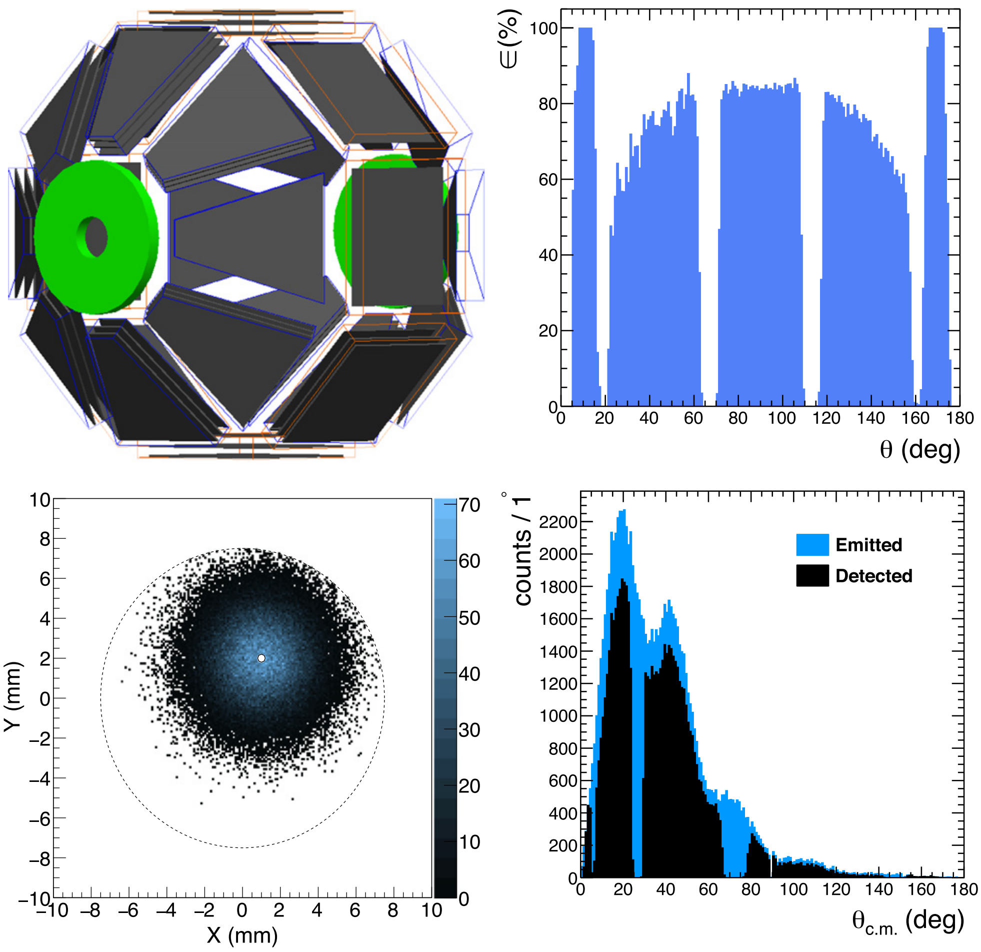

Figure 3. Geometry of the GASPARD silicon array simulated in NPTool (top left), together with the corresponding efficiency (top right). The beam spatial distribution at the target position is also represented (bottom left) while the emitted and detected protons are shown in the bottom right pannel.

5.1. Charged particle identification

The data presented in this example were accumulated on the RIPS [14] beam line at Riken during an experiment studying spectroscopic properties of ^{8}He and ^{10}He nuclei [15, 16]. The experimental particle identification plot presented in figure 4 was obtained using a ^{9}Li beam at 50 MeV u^{-1} impinging a 1.8 mg cm^{-2} CD_2 target. The light charged particles were detected with four MUST2 [17] telescopes equipped with 20 μm thick silicon detectors as an additional first stage. The simulated spectrum was generated using isotropic sources of protons, deuterons, tritons, ^{3,4,6}He, and ^{7,8}Li ions with flat energy distribution. The simulation manages to reproduce correctly the punch-through and separation in energy of each particle group. To obtain such an agreement the incident angle of each particle is used to correct its energy loss in the 20 μm silicon detectors.

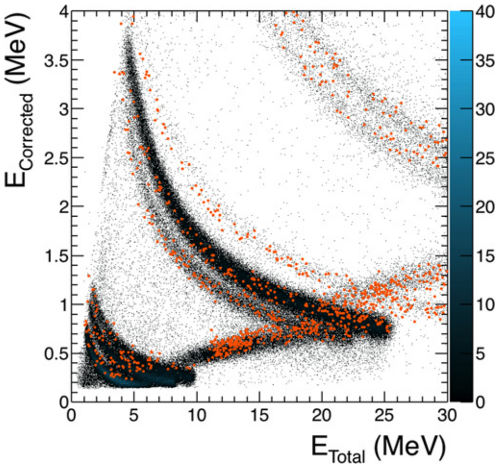

Figure 4. $\Delta E - E$ particle identification from the energy loss in the 20 μm silicon detector versus the total energy deposited in the telescope. The blue histogram represents experimental data, scattered orange dots are randomly generated particles (see text for the species).

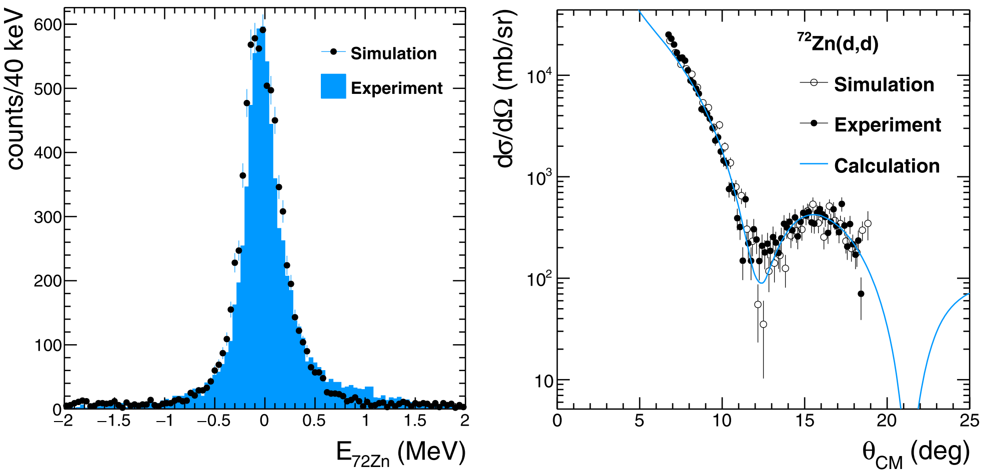

Figure 5. Left: excitation energy spectrum of ^{72}Zn derived from elastically scattered deuterons. Solid histogram shows the experimental restults to be compared to the simulated solid dots. Right: measured (solid dots) and simulated (hollow dots) elastic differential cross section. The solid line represents the theoretical differential cross section using the Daehnick's optical potential parametrisation [19].

5.2. Two-body kinematics and angular distributions

The ^{72}Zn(d,^{3}He)^{71}Cu proton pick-up reaction was performed at GANIL to study shell evolution in neutron rich Cu isotopes [18]. The ^{72}Zn secondary beam was produced after fragmentation of a ^{76}Ge primary beam and transported to a CD_2 target of 0.26 mg cm^{-2} through the LISE spectrometer. In order to obtain the optical potential parameters needed for

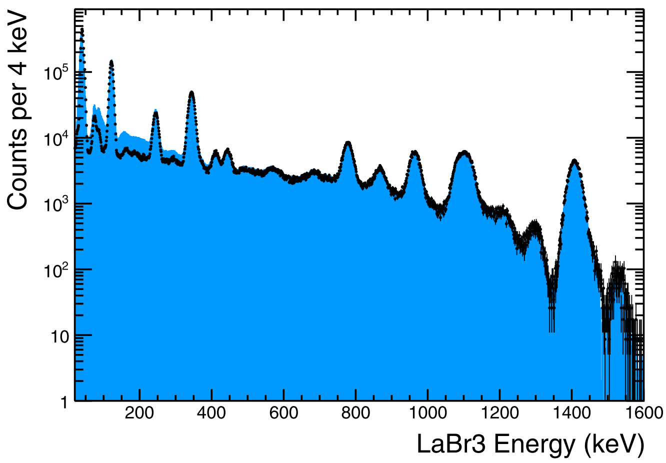

Figure 6. Measured (solid blue histogram) and simulated (solid orange dot) γ-ray spectrum from an ^{152}Eu radioactive source. The photopeak efficiency is correctly reproduced by the simulation on the whole range of energy.

analyzing the experimental differential cross section the deuteron elastic scattering was measured with two MUST2 telescopes placed around 90° in the laboratory frame. The excitation energy spectrum of scattered ^{72}Zn ions was reconstructed with the missing-mass method from the detected deuterons. Figure 5 compares experimental and simulated results and excellent agreement between the two can be observed. The Monte Carlo simulation took the measured beam emittance and energy spread as an input. A realistic cross-section calculation of the elastic scattering process was also used in the simulation. This way the angular distribution of the emitted deuterons can be reconstructed consitently in both simulated and experimental data.

5.3. γ-ray spectrometry

The NAtional Nuclear Array (NANA) is currently being designed and constructed for use at the UK's National Physical Laboratory (NPL). The purpose of this array is to perform direct measurements and metrological standardisation of nuclear decay activities. In the first generation the NANA array will consist of 12 individual scintillator detectors placed in a high efficiency geometry around the source position. NPTool simulations are used to define the design specification of the array, and especially its performance depending on the material chosen for scintillation: either $LaBr_3$(Ce) or $CeBr_3$ crystals. Once the array is constructed and operational, Monte Carlo simulations will help to validate the characterisation of environmental samples.

Figure 6 compares experimental and simulated data obtained with a $LaBr_3$ crystal detecting the γ-rays from a ^{152}Eu source. The experimental spectrum was obtained after 1 hours of counting with a 128 kBq source while the simulated spectrum is obtained by generating the corresponding number of ^{152}Eu decays through the Geant4 nuclear decay process (using *G4NuclearDecay4.3* and *Geant4.10.02*), with no further normalisation. The key feature reproduced here is the photo-peak efficiency of the detector, over the whole energy range, from a few tens of keV up to 1.6 MeV. Minor discrepancies in the background below 300 keV remain despite a very accurate geometry.

6. Work in progress and future development

The NPTool is an active project under continuous development. This activity is reflected by the nearly 1500 changes made to the source repository since the start of the project in 2009. An ongoing movement to add more detectors, both to analysis and simulation, is complemented by the constant thrust toward a faster, easier to use and more reliable framework. This section will detail some of the key features we plan to include in the near future.

6.1. Online analysis

Coupling the existing analysis with online analysis during experiments is a key feature for the future of the project. The ability to develop online analysis and test it with simulation before the scheduling of an experiment is a long-standing problem in the community. The NPTool already has most of the tools required to perform those tasks, and has been successfully coupled to the GANIL online package GRU [20] in a 2014 experimental campaign. However, having its own, independent, online utility is still a key feature for future versions of the NPTool. This would give the ability, for example, to perform an online analysis of a running simulation, and to make dynamical changes to the simulated geometry and events generated. A working prototype is already present in the framework but needs further improvement to be user friendly.

6.2. Cross-section bias

Simulation of nuclear fusion evaporation or fission is performed using the Geant4 reference physics list. One of the main limitation is that, due to the small cross section of these processes, one needs to simulate tens of million of beam events to obtain a useful spectrum. To overcome this limitation, we are actively working towards implementing cross section bias, allowing the end user to increase the cross section of a given process.

7. Conclusion

The NPTool is a flexible, adaptable and reliable framework, based on ROOT and Geant4. This framework, developed specifically for the low-energy nuclear physics community, fulfills the needs for analyzing and simulating heterogeneous detection systems with modular geometries.

While the NPTool requires some knowledge of C++ to be used to its full extent, it greatly simplifies the analysis and simulation processes so that it is easily usable by novice users. The source code has been distributed openly since the beginning of the project and the collaboration open to anyone wishing to contribute. The framework has been used for several PhD theses [15, 21–26] and peer review papers [16, 18, 27], leading to an extensive validation of the framework. Documentation is available from our website and full support ensured via email.

The increasing complexity of detector design in our field of physics calls for a unified framework to perform both simulation and analysis of experiments. We hope that the NPTool will continue to fulfil many of these needs while the NPTool collaboration will continue to improve this comprehensive toolbox for the low-energy nuclear physics community.

Acknowledgments

A Matta gratefully acknowledges the support from the UK STFC ST/L005743/1 grant.

References

[1] Al-Turany M, Bertini D, Karabowicz R, Kresan D, Malzacher P, Stockmanns T and Uhlig F 2012 The FairRoot framework *J. Phys.: Conf. Ser.* **396** 022001
[2] Frankland J D, Bonnet E, Gruyer D *et al KaliVeda data analysis toolkit* (http://indra.in2p3.fr/KaliVedaDoc)
[3] Pouthas J *et al* 1995 INDRA, a 4π charged product detection array at GANIL *Nucl. Instrum. Methods Phys. Res.* A **357** 418–42
[4] Bougault R *et al* 2014 The FAZIA project in Europe: R&D phase *Eur. Phys. J.* A **50** 47
[5] Agostinelli S *et al* 2003 Geant4 a simulation toolkit *Nucl. Instrum. Methods Phys. Res.* A **506** 250–303
[6] Brun R and Rademakers F 1997 ROOT an object oriented data analysis framework *Nucl. Instrum. Methods Phys. Res.* A **389** 81–6
[7] Ziegler J F, Ziegler M D and Biersack J P 2010 SRIM: the stopping and range of ions in matter *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* **268** 1818–23
[8] Tarasov O B and Bazin D 2008 LISE++: Radioactive beam production with in-flight separators *Nucl. Instrum. Methods Phys. Res.* B **266** 4657–64
[9] Audi G, Wapstra A H and Thibault C 2003 The ame2003 atomic mass evaluation: (ii). tables, graphs and references *Nucl. Phys.* A **729** 337–676
[10] Ottini-Hustache S *et al* 1999 CATS, a low pressure multiwire proportionnal chamber for secondary beam tracking at GANIL *Nucl. Instrum. Methods Phys. Res.* A **431** 476–84
[11] Beaumel D 2013 The GASPARD project *Nucl. Instrum. Methods Phys. Res.* B **317** 661–3
[12] Simpson J and AGATA collaboration 2015 The AGATA spectrometer: next generation gamma-ray spectroscopy *J. Phys.: Conf. Ser.* **606** 012017
[13] Maj A *et al* 2009 The Paris Project *Acta Phys. Pol.* B **40** 565 (www.actaphys.uj.edu.pl/fulltext?series=Reg&vol=40&page=565)
[14] Kubo T, Ishihara M, Inabe N, Kumagai H, Tanihata I, Yoshida K, Nakamura T, Okuno H, Shimoura S and Asahi K 1992 The RIKEN radioactive beam facility *Nucl. Instrum. Methods Phys. Res.* B **70** 309–19
[15] Matta A 2012 Study of the very neutron-rich ^{10}He by one-proton transfer reaction ^{11}Li(d, ^{3}He) *PhD Thesis* Université Paris-Sud XI
[16] Matta A *et al* 2015 New findings on structure and production of ^{10}He from ^{11}Li with the (d, ^{3}He) reaction *Phys. Rev.* C **92** 041302
[17] Pollacco E *et al* 2005 MUST2: A new generation array for direct reaction studies *Euro. Phys. J.* A **25** 287–8
[18] Morfouace P *et al* 2015 Evolution of single-particle strength in neutron-rich ^{71}Ccu *Phys. Lett.* B **751** 306–10
[19] Daehnick W W, Childs J D and Vrcelj Z 1980 Global optical model potential for elastic deuteron scattering from 12 to 90 mev *Phys. Rev.* C **21** 2253
[20] Legeard L 2015 *GRU* GANIL ROOT utilities (http://wiki.ganil.fr/gap/wiki/documentation/software/gru)
[21] Giron S 2011 Study of the reaction of astrophysical interest ^{60}Fe(n, γ)^{61}Fe via (d, p γ) transfer reaction *PhD Thesis* Université Paris-Sud XI
[22] Flavigny F 2011 Extraction of absolute spectroscopic factors from knockout and transfer reactions *PhD Thesis* Université Paris-Sud XI
[23] Moukaddam M Evolution of the shell structure in medium-mass nuclei : search for the 2d5/2 neutron orbital in ^{61}Ni *PhD Thesis* Université de Strasbourg

[24] Boissinot S 2013 Structure and spectroscopy of the Oxygen-24 drip-line nucleus from elastic and inelastic proton scattering using MUST2 detectors at RIKEN *PhD Thesis* Université Paris-Sud XI

[25] Lefebvre L 2013 Study of α clusters in ^{40}Ca and ^{40}Ar through nuclear break-up *PhD Thesis* Université Paris-Sud XI

[26] Morfouace P 2012 Single-particle states in neutron-rich ^{69}Cu and ^{71}Cu by means of the (d,^{3}He) transfer reaction *PhD Thesis* Université Paris-Sud XI

[27] Flavigny F *et al* 2013 Limited asymmetry dependence of correlations from single nucleon transfer *Phys. Rev. Lett.* **110** 122503