Quick tutorial for centrality determination using direct impact parameter reconstruction in MPD (NICA)

Dim Idrisov, Petr Parfenov, Vinh Ba Luong and Arkadiy Taranenko

March 2021

1 The direct impact parameter reconstruction

1.1 Installation

Below is a detailed description of the program used to reconstruct the distribution of the impact parameter. To start the fitting process, dowland script from https://github.com/Dim23/GammaFit.git

.

1.2 Fitting model data

In the case of fitting model data, it is sufficient to have only a histogram with a multiplicity distribution. Use following commands to run `GammaFit.C` :

```
root -b GammaFit.C(fileadres , current_mult , outadres , minNch)
```

Where the arguments are:

- `fileadres` - input root file.
- `current_mult` - Histogram of multiplicity.
- `outadres` - output file from fitting process.
- `minNch` - lower value of the fitting area (‘20‘ by default).

to set the value of the cross section in line 17 set the desired value of the variable `sigma`.

1.3 Fitting reconstracted data

When fitting the reconstructed data, it is necessary to take into account the efficiency of the detectors. To take these features into account, normalization is made to non-reconstructed model data.

To start the fitting process, run `GammaFit.C` with next option:

```
root -b GammaFit.C(fileadres , current_mult , outadres , minNch , efficiencyFit ,f
```

Where the arguments are:

- `fileadres` - input root file with multiplisity of the reconstructed data.
- `current_mult` - Histogram of multiplicity.
- `outadres` - output file from fitting process.
- `minNch` - lower value of the fitting area (‘20‘ by default).
- `efficiencyFit` - should be set ‘true‘ (‘false‘ by default).
- `fileadres2` - input root file with multiplisity of non-reconstructed model data.
- `current_mult2` - Histogram with multiplisity of non-reconstructed model data.

2 OUTPUT

Resulting file `outadres` will contain TCanvas with fit results and data-to-fit ratio - `Canvas_Of_fit_result`. Where the - `fit_func` is the resulting fit function of the multiplicity distribution . `FitResult` - TTree with fit parameters of fit function. `Result` - TTree with min and max percent of centrality and also the boundaries of the centrality classes.

`TTree` contains the following information about each centrality class:

- `MinPercent` – minimum value of centrality in the given centrality class
- `MaxPercent` – maximum value of centrality in the given centrality class
- `MinBorder` – lower cut on multiplicity for the given centrality class
- `MaxBorder` – upper cut on multiplicity for the given centrality class.

`fit_B_Mean` - TGraphErrors of impact parametr as a function of centrality .

`ImpactParametDist_CENT*_*` - histograms with the distribution of the impact parameter in centrality class.

2.1 Using centrality classes provided from the framework in the analysis

The file `outadres` have all needed information about centrality class. Use macro `printFinal.C` to display this information in a simple and readable way:

```
root -l -b -q printFinal.C'("path-to-FINAL.root")
```

This will print out all needed information for each centrality class. This macro also can save output information in latex and csv tables format. Example of `printFinal.C` saving in latex table:

```
root -l -b -q printFinal.C'("path-to-FINAL.root","./example.tex")
```

Example of `printFinal.C` saving in csv table (compatible with LibreOffice and MS Excel):

```
root -l -b -q printFinal.C'("path-to-FINAL.root","./example.csv")
```

Example of `printFinal.C` saving in C++ code:

```
root -l -b -q printFinal.C'("path-to-FINAL.root","./example.C")'
```

After `printFinal.C` generates output C++ code, one can use `Float_t GetCentMult(Int_t)` as a function which returns centrality percent based on input multiplicity value.