Zakład Fizyki Jądrowej
Uniwersytet Warszawski

Wydział Fizyki
Uniwersytet Warszawski

Computer Tools for Nuclear Physics

Initial state of collision within Glauber model

Krzysztof Piasecki


krzysztof.piasecki AT fuw.edu.pl

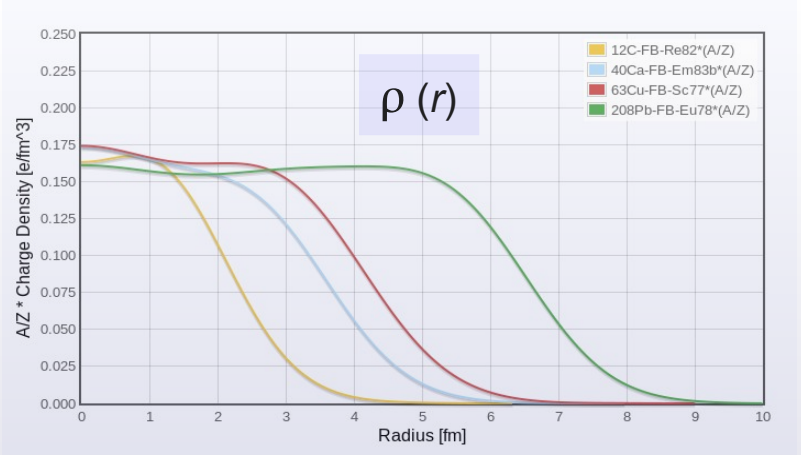
Basics of Glauber Model

- Glauber model: Relativistic Heavy-Ion collision
  - made of sum of independent NN collisions
  - density profile is diffused

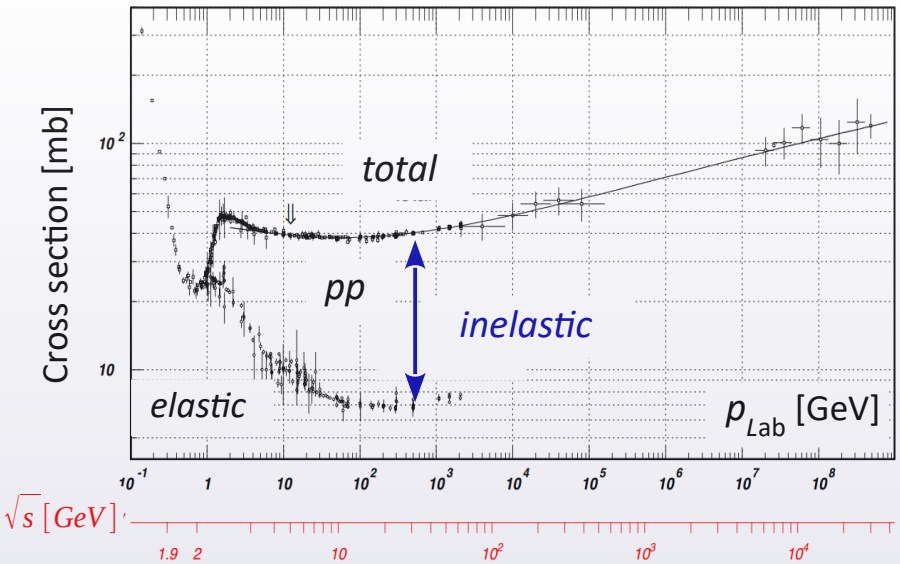
  Simplest density model: Fermi distribution

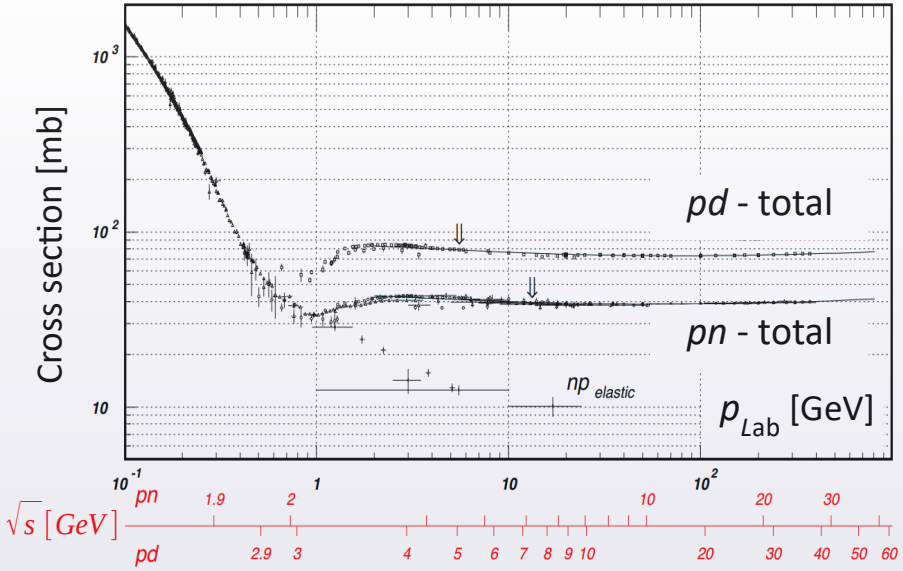
$$\rho(r) = \frac{\rho_0}{\exp\left(\frac{r - r_0}{d}\right) + 1}$$

- Cross section for NN collision is not constant.



journals.aps.org/prd/abstract/10.1103/PhysRevD.86.010001





$\sigma_{NN,\ inelastic}$: swing + wide plateau $\sigma_{NN} \approx 30 .. 50$ mb

Glauber Model: optical approach

- *Introductory paper:* M. Miller et al. Ann.Rev.Nucl.Part.Sci. 57, 205 (2007)
  Talk: J. Wilkinson Glauber modelling in hi-ener nucl. coll.

- **Optical**
  continuous ϱ(r) distribution
  Analytical formulae,
  Integrals are probed numerically.

  ⟵ **Glauber Model** ⟶

  Monte Carlo
  Granulation of A+B nucleons.
  Their centers are positioned according to ϱ(r).

Step 1: average chance for 1 NN collision = ?

- 3D → 2D. **Thickness function T_A** [fm^{-2}] :

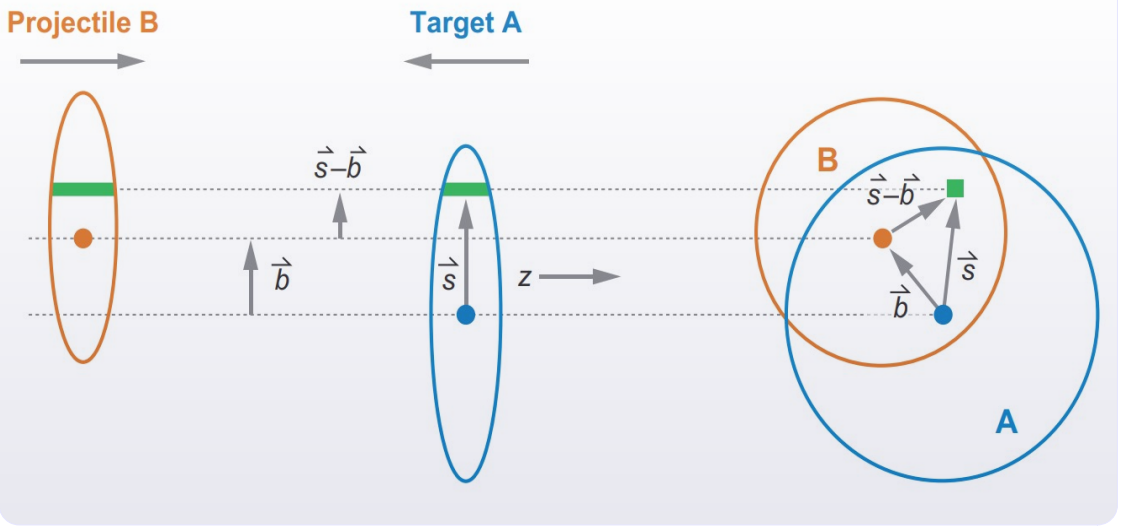
$$T_A(\vec{s}) = \int \rho_T\left(\sqrt{\vec{s}^2+z^2}\right)\, dz \qquad \int \rho_T\, d^3r = 1$$

(chance of finding 1 nucleon per 1 fm^2 of ⊥ area)

- **Overlap function T_{AB}**:

$$T_{AB}(\vec{b}) = \int d^2s\; T_A(\vec{s})\; T_B(\vec{s}-\vec{b})$$

(chance that at fixed *b*,
a nucleon from A meets a nucleon of B) [fm^{-2}]



⇒ $T_{AB}(b) \cdot \sigma_{NN,\,inel.}$ = chance for a *single* NN collision

Glauber Model: optical approach

- We set up two nuclei at the parameter *b*. Nucleons move forward and „try to collide", with success or failure.
  Average probability of *single* NN collision is : $T_{AB}(b) \cdot \sigma_{NN}$.
  Number of NN collisions = number of „successes" in $A \cdot B$ attempts $\leftrightarrow$ follows the binomial distribution:

$$P(n,b) = \binom{AB}{n} \left[T_{AB}\sigma_{NN,inel}\right]^n \left[1 - T_{AB}\sigma_{NN,inel}\right]^{AB-n}$$

- Total cross section for nucleus-nucleus (AA) collision. The AA collision occurs if at least 1 NN collision is done:

P (≥ 1 NN collision) = 1 – P (0 NN collisions) $\Rightarrow$ $$\sigma_{AB,inel} = 2\pi \int_0^\infty \left\{1 - \left[1 - T_{AB}(b)\sigma_{NN,inel}\right]^{AB}\right\} bdb$$

- Number of „binary" NN collisions:

$$N_{coll}(b) = \sum_{n=1}^{AB} nP(n,b) = \ldots = AB \cdot T_{AB}(b)\sigma_{NN,inel}$$

- Number of participant nucleons:

$$N_{part}(b) = A\int T_A(\vec{s}) \left\{1 - \left[1 - T_B(\vec{s}-\vec{b})\sigma_{NN,inel}\right]^B\right\} d^2s +$$
$$B\int T_B(\vec{s}-\vec{b}) \left\{1 - \left[1 - T_A(\vec{s})\sigma_{NN,inel}\right]^A\right\} d^2s$$

Inelastic NN cross sections

① σ(pp) is different from σ(pn) and σ(np)

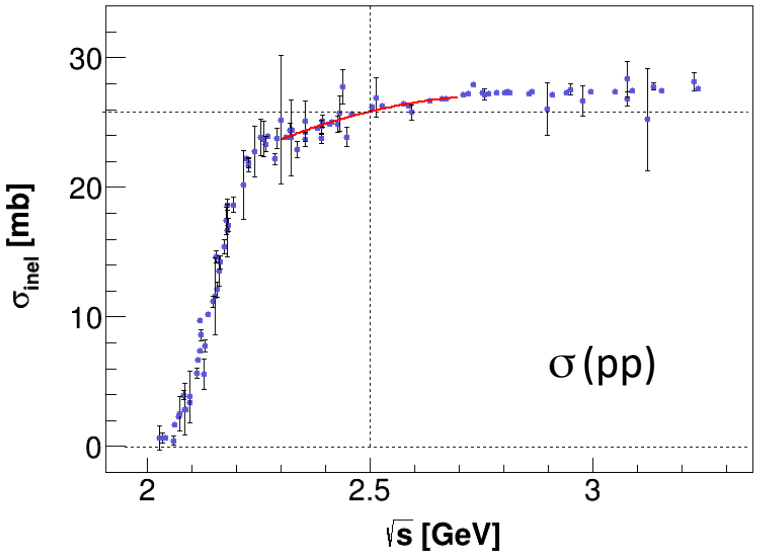
② Assumption: isospin symmetry [$\sigma_{nn} = \sigma_{pp}$]

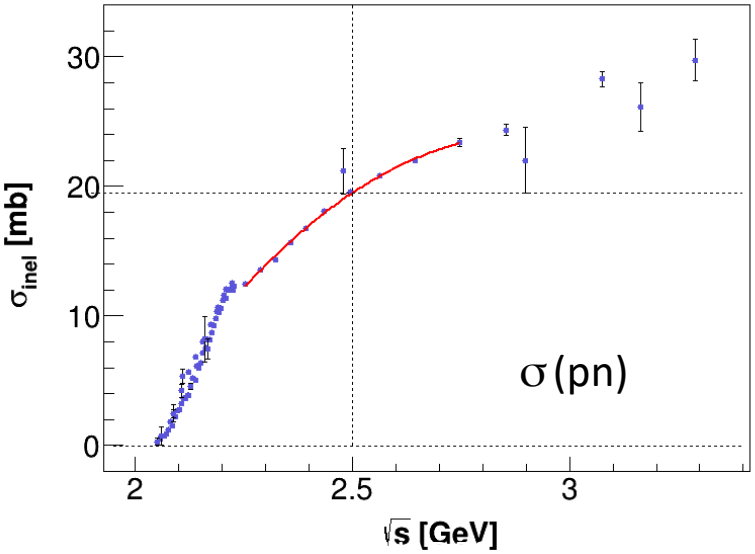
$$\sigma_{NN} = \frac{Z_p Z_t \sigma_{pp} + N_p N_t \sigma_{nn} + (Z_p N_t + N_p Z_t)\sigma_{np}}{A_p A_t}$$

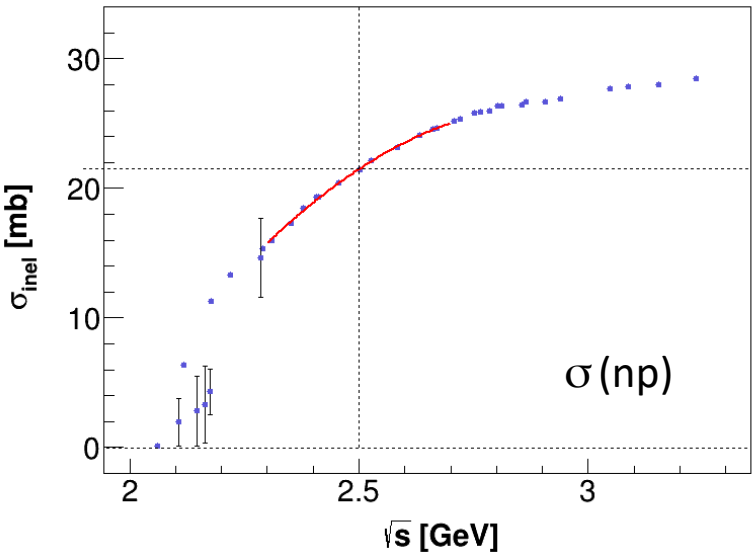
③ Experimentally, σ(pn) is not the same as σ(np)

④ σ(np) at low √s and σ(pn) at higher √s are rare

[③, ④] contribute to systematic errors





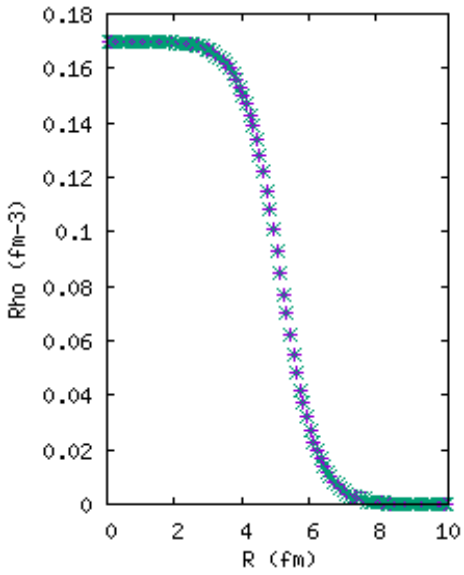


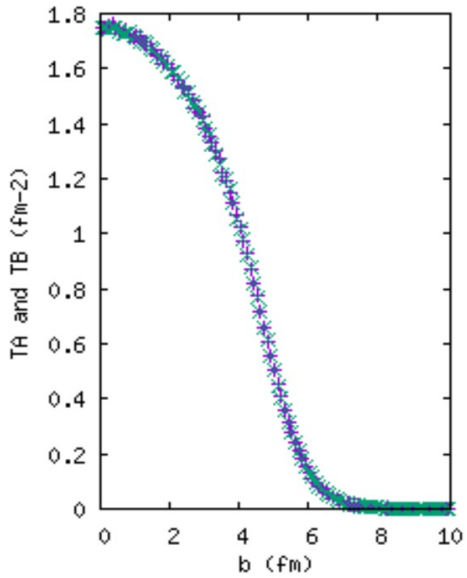
Refs. :

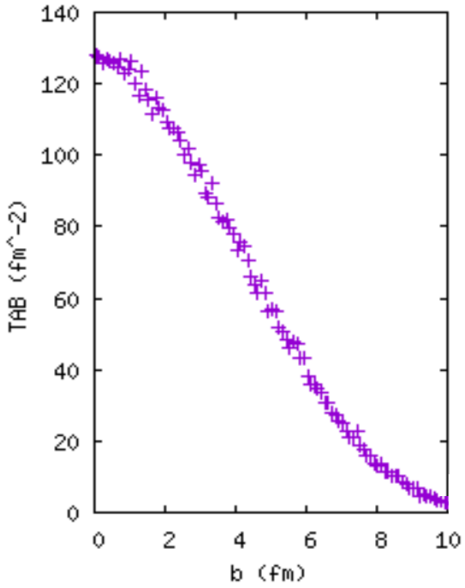
- PDG
- B. Kardan's Ms. th.

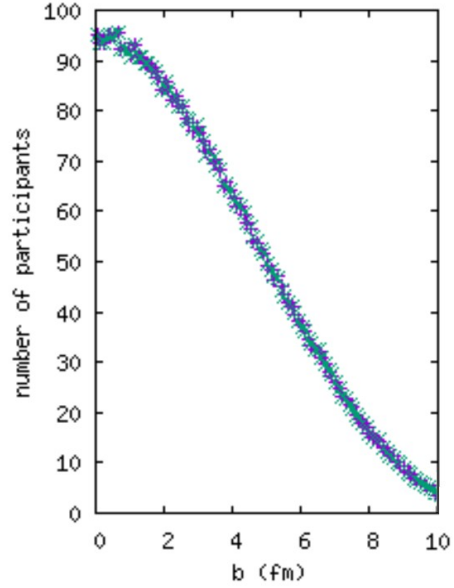
Glauber Model: optical approach

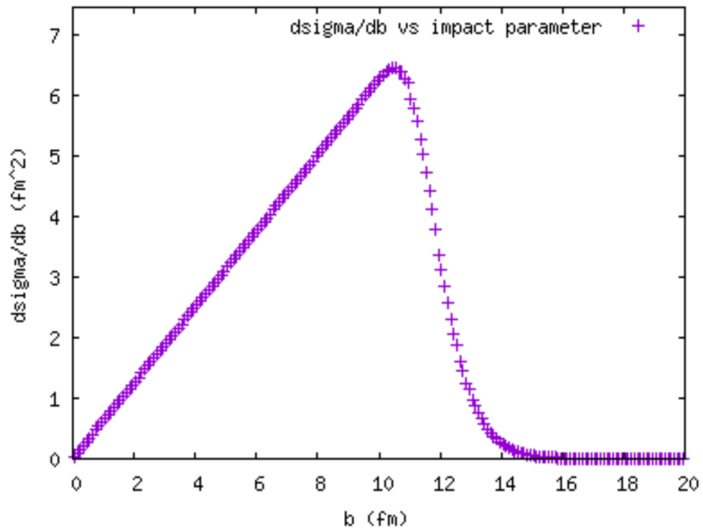
- **"Overlap":** optical Glauber model online **[click here]**. E.g. for ^{108}Ag + ^{108}Ag collision (let's assume $\sigma_{NN,\ inel.}$ = 25 mb)











- Let's check how settings can influence $\langle A_{part} \rangle$ and $\langle N_{Coll} \rangle$:

	Uniform balls	Fermi distrib. σ_{NN} = 20 mb	Fermi distrib. σ_{NN} = 30 mb
$\langle A_{part} \rangle_b$	55	44.5	48.5
$\langle N_{coll} \rangle_b$	92	65	76

Glauber Monte Carlo: TGlauberMC

- TGlauberMC: ROOT-based simulator of initial conditions of AA collision within the Glauber MC approach.
  Authors: from PHOBOS @ RHIC

Homepage:	tglaubermc.hepforge.org
Documentation:	https://tglaubermc.web.cern.ch/html/index.html
Download:	www.hepforge.org/downloads/tglaubermc
Prerequisites:	Root ≥ 4
Main papers:	[No. 1 @ 2008] [No. 2 @ 2019] [No. 3 @ 2019] [No. 4 @ 2021]
User guide:	Best description in paper 2.
Applicability:	(authors:) $\sqrt{s_{NN}} \in$ [200 GeV … 10 TeV] , but used also for $\sqrt{s_{NN}}$ = 2.5 GeV.

- Installing TGlauberMC on your account of NPD's training computer:

```
$  mkdir tglaubermc ; cd tglaubermc
$  cp -p ~kpiasecki/soft/TGlauberMC/tglaubermc_install.sh .
$  ./tglaubermc_install.sh
```

- Each time you open a new Terminal, do `cd tglaubermc` and then :

```
$  . ./tglaubermc_start.sh
```

(*note*: dot at the beginning)

Now if you just launch Root in this directory, you have all the TGlauberMC functions connected.

Glauber Monte Carlo: TGlauberMC

- Currently implemented nuclei: p d t ^{3,4}He C O Ne Al Si S Ar Ca Ni Cu Nb Zr Ru Ag Sn Xe W Au Pb U

Basic form of parametrization of nuclear shape:

$$\rho(r) = \rho_0 \cdot \frac{1 + w\left(\frac{r}{R}\right)^2}{1 + \exp\left[\frac{r - r_0}{a}\right]}$$

Shape deformation is an option for:
Ne, Al, Si, Cu, Xe, Au, U

$$\rho(r) = \rho_0 \cdot \frac{1}{1 + \exp\left[\frac{r - R\left(1 + \beta_2 Y_{20} + \beta_4 Y_{40}\right)}{a}\right]}$$

Possible nuclei and profiles can be checked or updated in the `TGlauNucleus::Lookup` method.

- First, *b* parameter is pulled randomly from ◿ distribution. Centers of nuclei are defined as [-*b*/2, 0, 0] and [*b*/2, 0, 0] within frame as shown on this Figure.

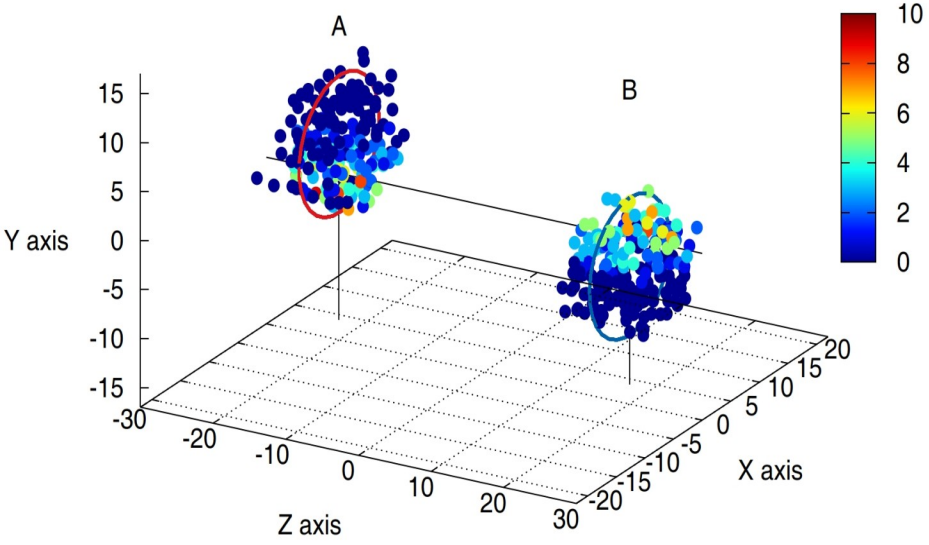
  Positions of nucleons are pulled from shape distributions. A minimum distance between N-N centroids is required (d_{min} = 0.4 fm by default).

- Deformed nuclei are rotated randomly by φ and ϑ.

  Nucleons are 'traversing' head-on ('eikonal' approximation) and undergo collisions independently.

- A single NN collisions is counted if the distance *d* between centroids is less than:

$$d < \sqrt{\frac{\sigma_{NN}}{\pi}}$$



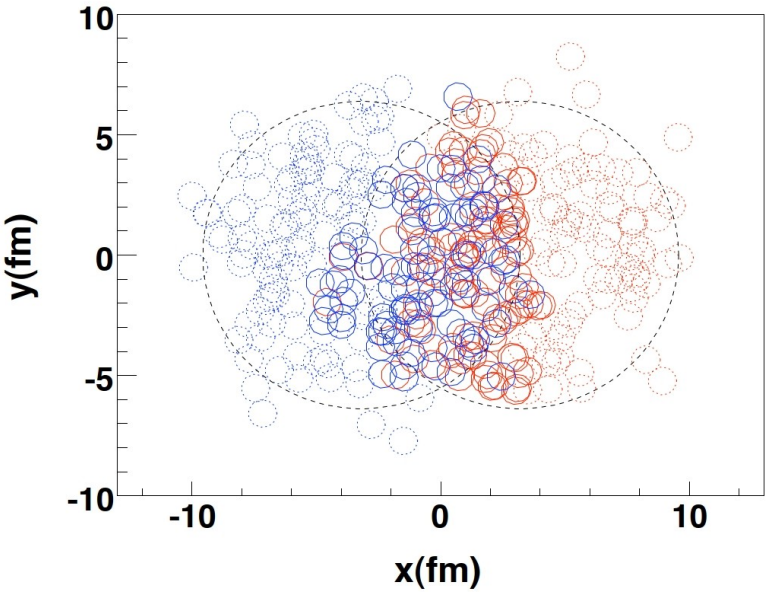
J.Adamczewski-Musch et al., Eur. Phys. A 54, 85 (2018)

Glauber Monte Carlo: TGlauberMC

- Code offers a few classes and macros.
  Class `TGlauNucleon` represents a single nucleon: position, type (p/n), origin (nucleus A/B), No. of collisions,...
  Class `TGlauberMC` is the main simulation manager.

- ```
  root[1] TGlauberMC gmc ("Pb","Pb",30.)
  root[2] gmc.NextEvent ( 5. )
  root[3] gmc.Draw ()
  ```

  Initialize Pb+Pb collision with σ_{NN} = 30 mb
  Simulate 1 collision at *b* = 5 fm. For -1 , random *b*.
  Visualize of the collision (wounded/participants)



- Btw. you can apply range of *b* pars:

```
gmc.SetBmin ( 4. );
gmc.SetBmax ( 7. );
```

Get (inelastic) collision cross section based on this event:

```
root[4] gmc.GetTotXSect ()
12.57
```

Getting No. of participants and collisions:

```
root[5] cout << gmc.GetNpart() <<'\t'<<
                gmc.GetNcoll() << endl;
370 786
```

Get array of nucleons:

```
root[6] gmc.GetNucleons ()
```

Simulate e.g. 1000 collisions at once.

```
root[7] gmc.Run (1000)
```

Retrieve event-wise ntuple (can be saved in Root file)

```
root[8] TNtuple* ntu = gmc.GetNtuple ()
```
```

Glauber Monte Carlo: TGlauberMC

- Macro `runAndSaveNtuple` simulates and stores the resulting event-wise ntuple in file:

`root [0] runAndSaveNtuple (100, "Ni", "Ni", 25.)` simulate 100 Ni+Ni collisions with σ_{NN} = 25 mb

- Variables available in the ntuple include:

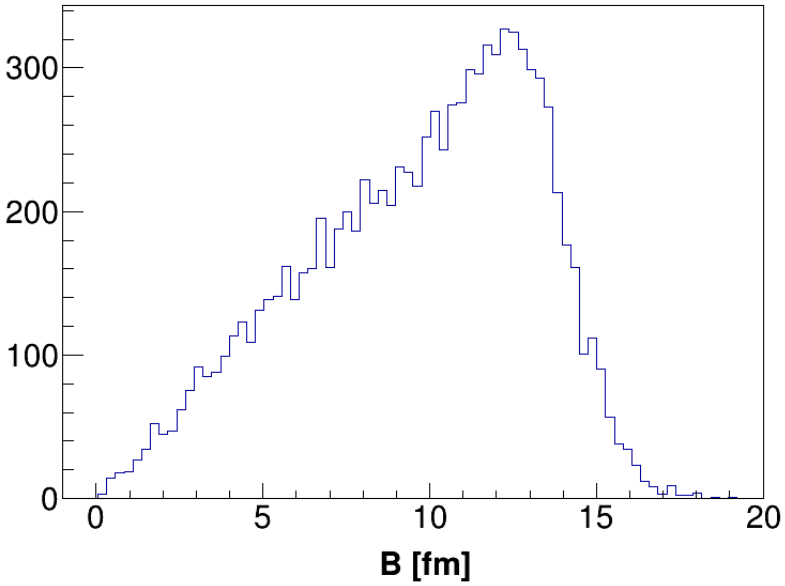
`Npart`	No. of participants
`Ncoll`	No. of collisions
`Nhard`	No. of hard-core collisions
`B`	Impact parameter
`BNN`	Average NN impact parameter
`Ncollpp`	No. of pp collisions
`Ncollpn`	No. of pn collisions
`Ncollnn`	No. of nn collisions
`VarX`	Variance of X of wounded nucleons
`VarY`	Variance of Y of wounded nucleons
`VarXY`	Covar. between X and Y of wounded nucleons
`NpartA`	No. of wounded nucleons in nucleus A
`NpartB`	No. of wounded nucleons in nucleus B
`Npart0`	No. of singly-wounded nucleons
`AreaW`	Area defined by width of participants
`PsiN`	Event plane angle of n-th harmonic
`EccN`	Participant eccentricity for n-th harmonic

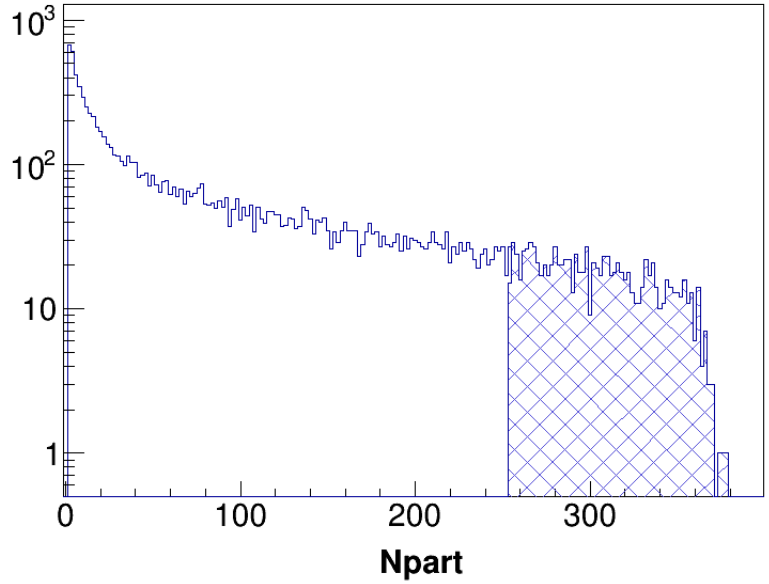
`AreaA`	Area defined by ”and” of participants
`AreaO`	Area defined by ”or” of participants
`MeanX`	$\langle x \rangle$ of wounded nucleons
`MeanY`	$\langle y \rangle$ of wounded nucleons
`MeanX2`	$\langle x^2 \rangle$ of wounded nucleons
`MeanY2`	$\langle y^2 \rangle$ of wounded nucleons
`MeanXY`	$\langle xy \rangle$ of wounded nucleons
`MeanXSystem`	$\langle x \rangle$ of all nucleons
`MeanYSystem`	$\langle y \rangle$ of all nucleons
`MeanXA`	$\langle x \rangle$ of nucleons in nucleus A
`MeanYA`	$\langle y \rangle$ of nucleons in nucleus A
`MeanXB`	$\langle x \rangle$ of nucleons in nucleus B
`MeanYB`	$\langle y \rangle$ of nucleons in nucleus B
`PhiA`	ϕ angle nucleus A (applied if deformed)
`ThetaA`	ϑ angle nucleus A (applied if deformed)
`PhiB`	ϕ angle nucleus B (applied if deformed)
`ThetaB`	ϑ angle nucleus B (applied if deformed)

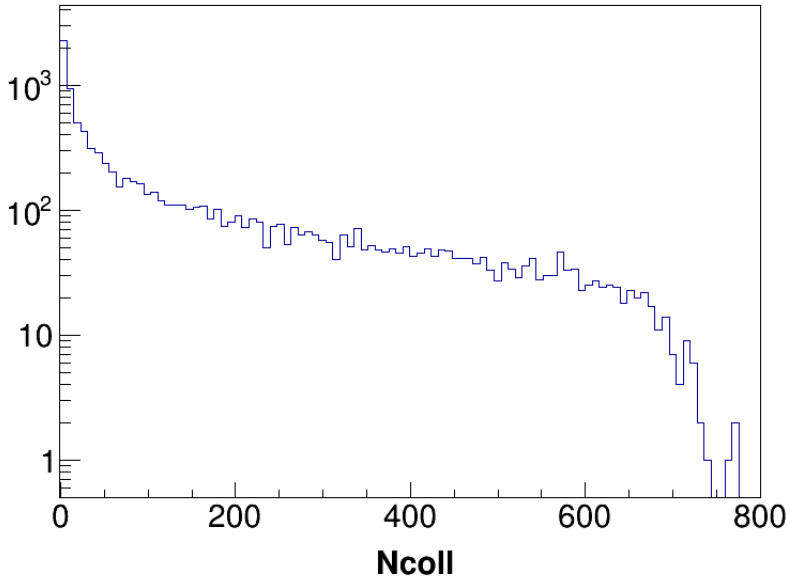
- *Caution*: `PsiN` and `EccN` are constructed from weighted positions of wounded nucleons.

Glauber Monte Carlo: TGlauberMC

- Simulation: Au+Au, σ_{NN} = 23.7 mb .

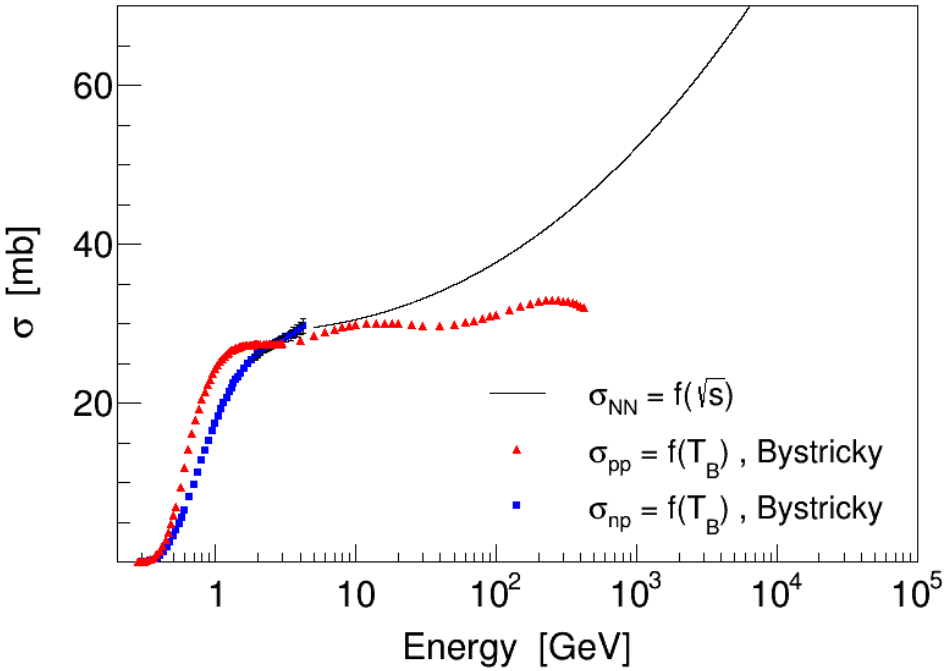






Glauber Monte Carlo: TGlauberMC

- Helper functions for parametrizations of σ_{NN}, σ_{np} and σ_{pp} .
  - `Double_t TGlauberMC::GetSigmaNN ( √sNN )` valid within $\sqrt{s_{NN}} \in$ [10 GeV ... 100 TeV]
    `TF1 TGlauberMC::GetSigmaNNvsEnergy ()`
  - `Double_t TGlauberMC::GetSigmaNP_Bystricky( T Beam )` valid within $T_{Beam} \in$ [0.3 GeV ... 4.2 GeV]
    `TGraph TGlauberMC::GetSigmaNPvsEnergy_Bystricky ()`
  - `Double_t TGlauberMC::GetSigmaPP_Bystricky( T Beam )` valid within $T_{Beam} \in$ [0.3 GeV ... 425 GeV]
    `TGraph TGlauberMC::GetSigmaNPvsEnergy_Bystricky ()`

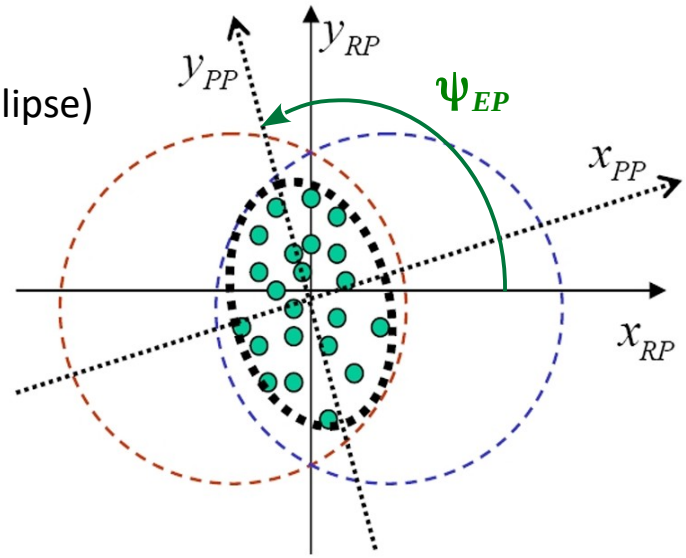


Geometry of wounded nucleons

- Centroids of nuclei are initially fixed to the XZ plane (see Fig. on p. 8) .

  Ideally:
  - area occupied by wounded nucleons looks like a vertical almond (~ellipse)
  - angle ψ between the longer axis and X axis: $\psi_{EP} = 90°$
  - shape has given eccentricity ε

  However, in an event, positions of nucleons are pulled randomly.
  - shape deviates from an "ideal" almond
  - angle ψ_{EP} differs from 90°.
    A de-facto plane is called "**event plane**" or "**participant plane**"
  - shape has given **eccentricity ε** (deviated from ideal ε)



[S. Voloshin et al.]

ψ_{EP} is found from positions (x_i, y_i) by:

$$\psi_{EP} \equiv atan\left(\frac{\langle \sin\varphi \rangle}{\langle \cos\varphi \rangle}\right) + \pi$$

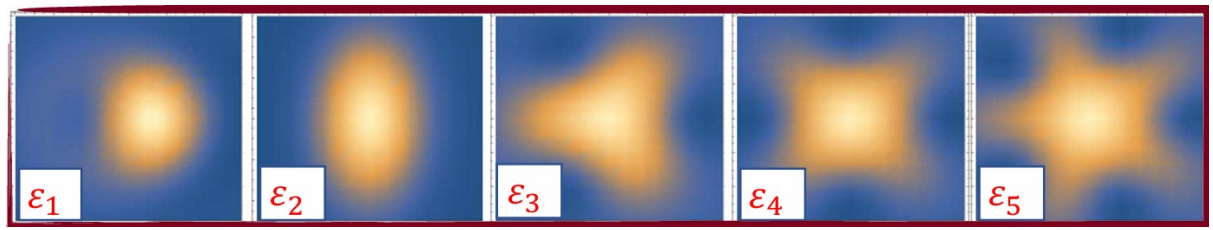
ε is also found from wounded nucleons:

$$\varepsilon \equiv \frac{\sqrt{\langle r^2\cos(2\varphi)\rangle^2 + \langle r^2\sin(2\varphi)\rangle^2}}{\langle r^2 \rangle}$$

- For more refined analyses of shape and fluctuations, ψ and ε of higher harmonics is available.

$$\varepsilon_n \equiv \frac{\sqrt{\langle r^n\cos(n\varphi)\rangle^2 + \langle r^n\sin(n\varphi)\rangle^2}}{\langle r^n \rangle}$$

$$\psi_n \equiv \frac{1}{n}\, atan\left(\frac{\sum r^n\cos(n\phi)}{\sum r^n\sin(n\phi)}\right)$$



[Source: M. Stefaniak]

Glauber Monte Carlo: TGlauberMC

- Let's focus on `gmc.GetNucleons()`. It returns the address to the array of `TGlauNucleon` objects.

- What does a single `TGlauNucleon` object contain ?

  - **position**: You can `SetXYZ` and `GetX/Y/Z`, but also `RotateXYZ(_3D)` it.
    Also it knows if it `IsInNucleusA/B`.

  - **type**: `IsProton, IsNeutron, Get/SetType`. Also energy, but it has no effect.

  - **collision status**: if it `IsWounded` and how badly: `GetNcoll`.
    Alternatively, if it `IsSpectator`.
    You can also "make" it `Collide` or heal fully: `Reset`.

Glauber Monte Carlo: TGlauberMC

- Looping over nucleons in an event.

```
R__LOAD_LIBRARY (libMathMore)
R__LOAD_LIBRARY (runglauber_v3.2_C)

int nucloop () {
 TGlauberMC gmc       ( "Pb" , "Pb" , 25. );
 TH1F       hNucColl ("hnuccoll", "", 13, -0.5, 12.5);
 TObjArray*    nucArray ;
 TGlauNucleon* nuc ;

 for (int iEvent = 0; iEvent < 100 ; iEvent++)
 {
  gmc.NextEvent ( -1 );
  nucArray = gmc.GetNucleons ();
  for (int iNuc = 0; iNuc < nucArray->GetEntries(); iNuc++)
  {
   nuc = (TGlauNucleon*) nucArray->At (iNuc) ;
   hNucColl->Fill ( nuc->GetNColl() );
  }
 }
 hNucColl->Draw ();
 return 0;
}
```

Create the simulator of Pb+Pb collision @ σ_{NN} = 25 mb

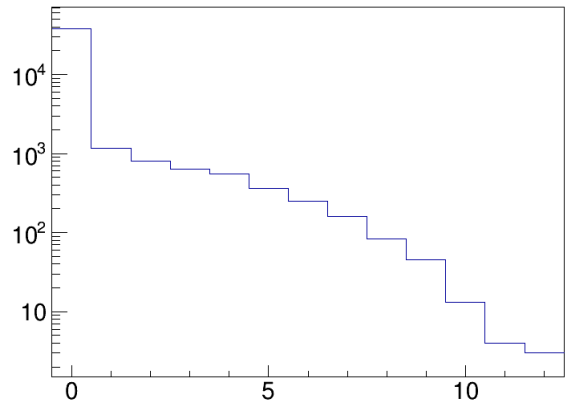
Loop over events

Simulate 1 coll. at random *b*

Get array of nucleons

Loop over nucleons

Get given nucleon



Glauber Model vs experimental data

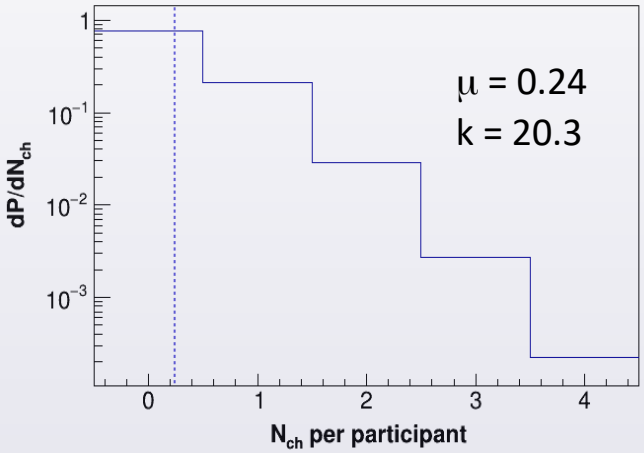
- Obtained A_{part} distribution is not yet a place of comparison with experiment:
  1. usually only charged particles are measured (protons)
  2. detector acceptance is limited → will measure only part of particles
  3. even if particle gets inside: setup has some inefficiency of track measurement (efficiency ≡ ε) .

- N_{part} distribution from TGlauberMC simulation has to be projected to N_{ch} measured particles, Accounting for fluctuations. Each participant is associated with „*P* of creating registered particle".

- One usually applies the **negative binomial distribution, NBD** with mean μ and width parameter *k*:

$$P_{\mu,k}(n) = \frac{\Gamma(n+k)}{\Gamma(n+1)\,\Gamma(k)} \cdot \frac{(\mu/k)^n}{(\mu/k+1)^{n+k}}$$

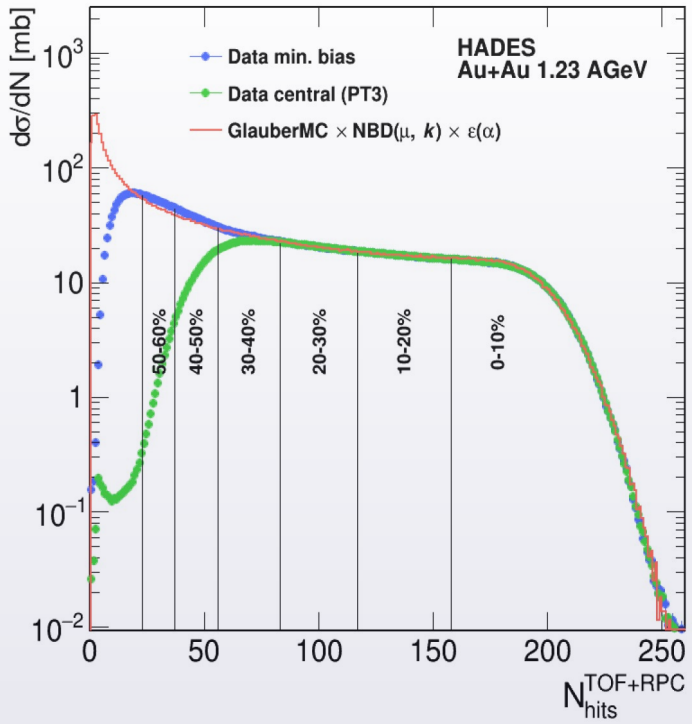
Example for Au+Au @ 1.2A GeV (HADES):



- Plot shows the distrib. of multiplicity of charged particles.
  - Glauber MC
  - Experiment, "minimum bias"
  - Experiment "central trigger"

Good agreement between model and experiment.

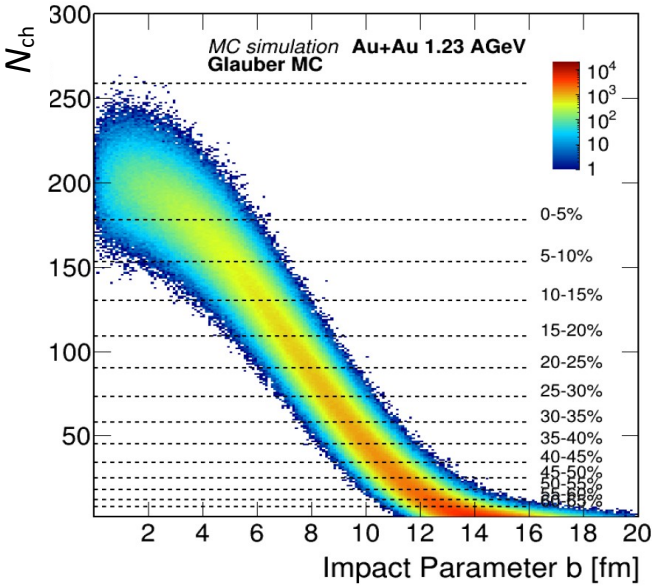
Experimentally one defines the „**centrality classes**" on this plot.

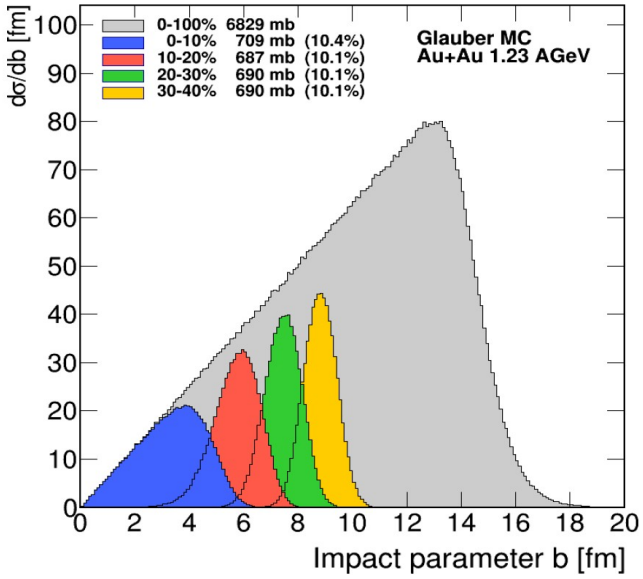


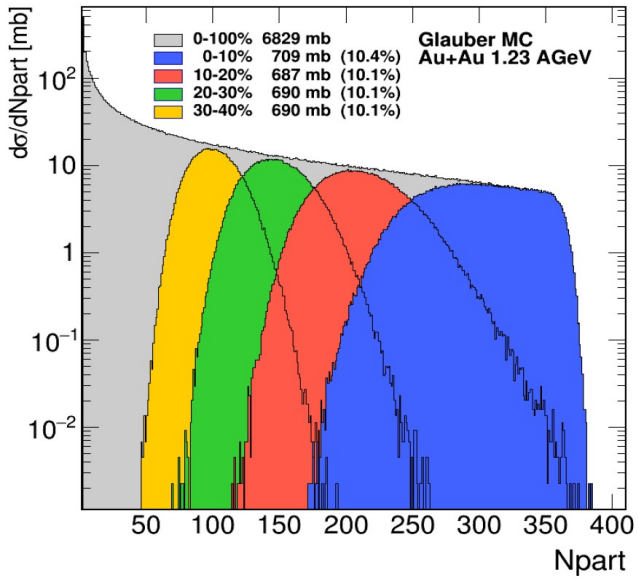
J. Adamczewski-Musch et al. (HADES), EPJA 54, 85 (2018)

Glauber Model vs experimental data

- We now have N_{ch} distribution from the Glauber model - tuned to the experimental one .
  In real experiment, to assess the centrality class, one usually slices the N_{ch} distribution.
  Question: how much such slices correlate with selecting b_{par} ranges and N_{part} ranges?







→ Strong anticorrelation between N_{ch} and b_{par} :
In subsequent bins of N_{ch} (**centrality classes**) – an average b moves.

But caution: in every centrality class b always exhibits a distribution.
Even a choice of a narrow N_{ch} range → N_{part} distribution is diffused. Although always $\langle N_{ch} \rangle \sim \langle A_{part} \rangle$.
Basing on Glauber model, in every centrality class, one can obtain $\langle A_{part} \rangle_b$.

J. Adamczewski-Musch et al. (HADES), EPJA 54, 85 (2018)