Centrality questions & answers - III

V. Riabov

Small bug in the Glauber fit machinery

- Fitter.cpp
- Two histograms are compared to find the best fit:
  - ✓ Data: TH1F *hRefMult = new TH1F("hRefMultSTAR","hRefMultSTAR",2500,0,2500);
  - ✓ Glauber: fGlauberFitHisto = TH1F("glaub", "", fNbins*1.3, 0, 1.3*fMaxValue);

not an integer

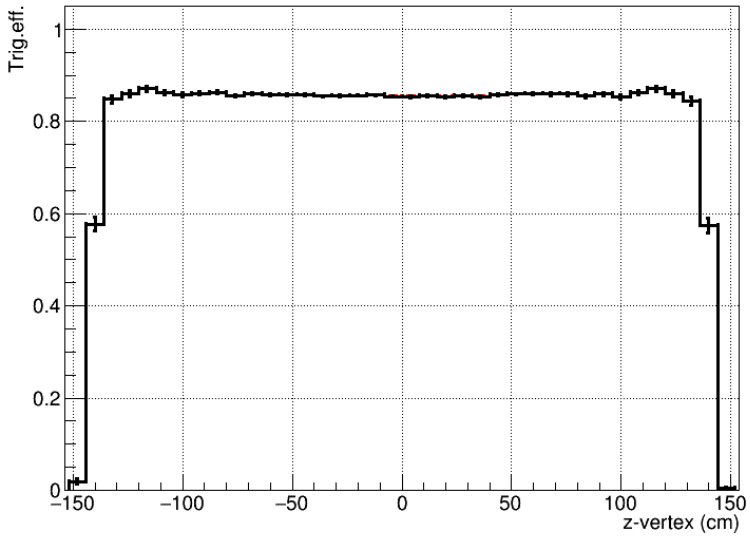
- Fit quality is evaluated using histograms with somewhat different binning (bin width)
- The easy fix:
  - ✓ Glauber: fGlauberFitHisto = TH1F("glaub", "", fNbins*1.3, 0, int(1.3*fMaxValue));
- Fit quality improves, modest change of the best fit parameters
- No dramatic changes

Centrality with the TPC

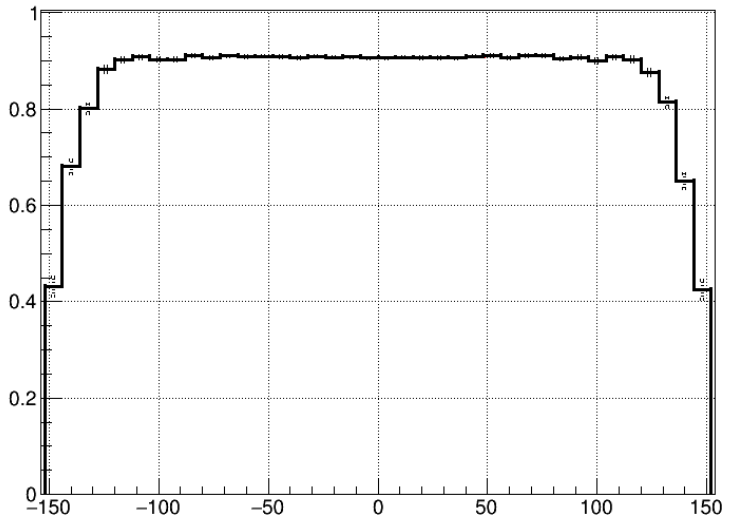
- Event selection:
  - ✓ FFD || FHCAL trigger (minimal bias)
  - ✓ reconstructed vertex: z-vertex !=0
  - ✓ reconstructed vertex is outside of the FFD: |z-vertex| < 130 cm
  - ✓ number of tracks: $N_{TPC} > 0$, track selections: nhits > 10; $p_T > 0.1$ GeV/c; DCA < 2.0 cm; $|\eta| < 0.5$
- Each of these conditions reduces the fraction of the total cross section
- By how much ???

Trigger efficiency: events with z_{vrtx} != 0 and N_{TPC} > 0

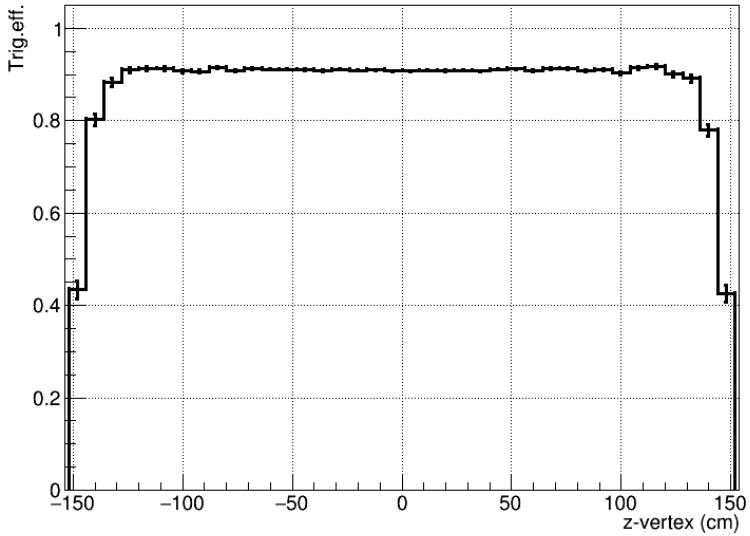
FFD + vertex + Ntr >0



FHCAL+ vertex + Ntr >0



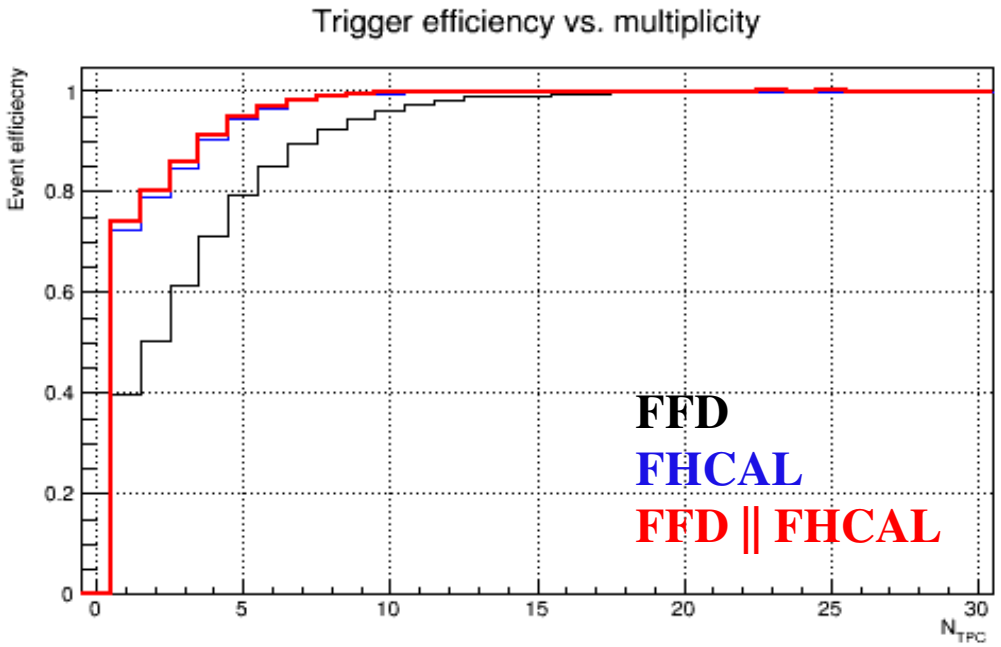
FFD||FHCAL+ vertex + Ntr >0



- Request 26 production, 1M DCM-QGSM-SMM
- Trigger efficiency is flat vs. z-vertex
- Trigger efficiency dropped by ~ 5% due extra req-s
- Proposal is to use FFD||FHCAL trigger selection for the large productions and ignore the fact that T_0 measurements are not available for !FFD events
- The resulting trigger efficiency is ~ 91%

Emulation of trigger

- Trigger simulation is possible only for Request 26 production (DCM-QGSM-SMM)
- Use TPC multiplicity as a proxy for trigger efficiency for all productions

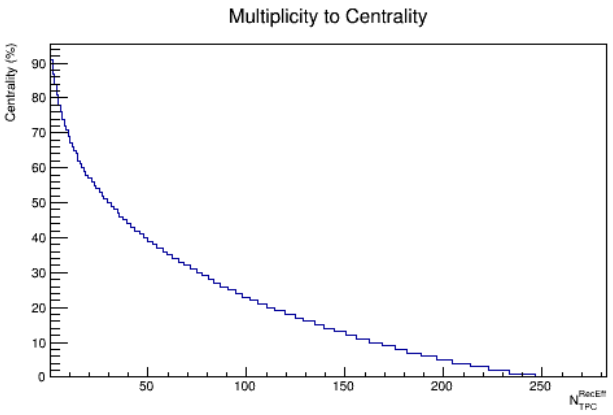


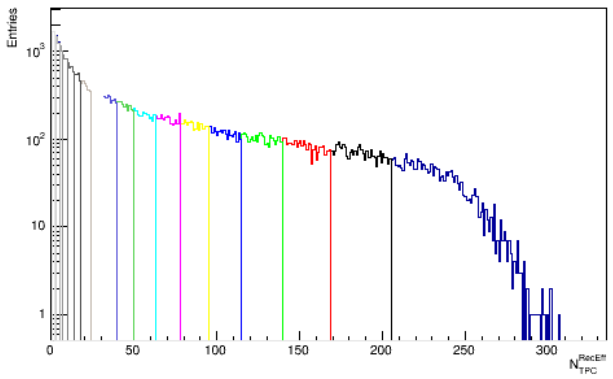
- At N_{TPC} > 0-10 the trigger efficiency saturates for FFD||FHCAL option
- Evaluated trigger efficiency is applicable only for default track/event selection cuts:

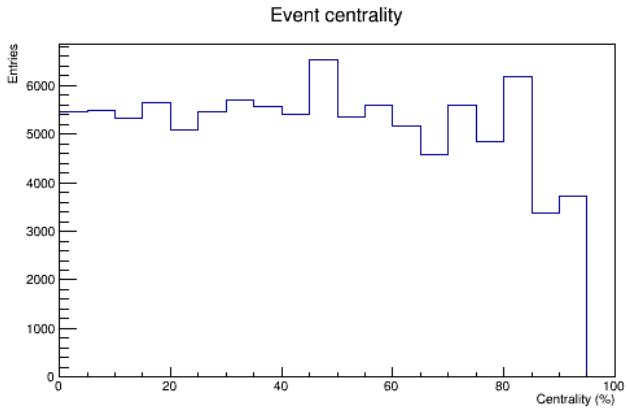
TrigEff[0] = 0;
TrigEff[1] = 0.739448;
TrigEff[2] = 0.802705;
TrigEff[3] = 0.85774;
TrigEff[4] = 0.910615;
TrigEff[5] = 0.949969;
TrigEff[6] = 0.967788;
TrigEff[7] = 0.982232;
TrigEff[8] = 0.990496;
TrigEff[9] = 0.994225;
TrigEff[10] = 0.99635;
TrigEff[11] = 0.997572;
TrigEff[12] = 0.998404;
TrigEff[13] = 0.99816;
TrigEff[14] = 0.99899;
TrigEff[15] = 0.998933;
TrigEff[16] = 0.998358;
TrigEff[17] = 0.999083;
TrigEff[18] = 0.998685;
TrigEff[19] = 0.998247;

Centrality by TPC multiplicity, DCM-QGSM-SMM

- Only for good events:
  - ✓ reconstructed vertex: z-vertex !=0
  - ✓ reconstructed vertex is outside of the FFD: |z-vertex| < 130 cm
  - ✓ number of tracks: $N_{TPC} > 0$, track selections: nhits > 10; $p_T > 0.1$ GeV/c; DCA < 2.0 cm; $|\eta| < 0.5$
  - ✓ Rndm() > TrigEff[N_{TPC}]
- Resulting multiplicity distribution samples ~91% of the total cross section
- Event multiplicity is calculated using weight for each track ~ 1/RecEff(z-vertex, η)
- Centrality is defined as percentile of the total multiplicity with maximum of 91%

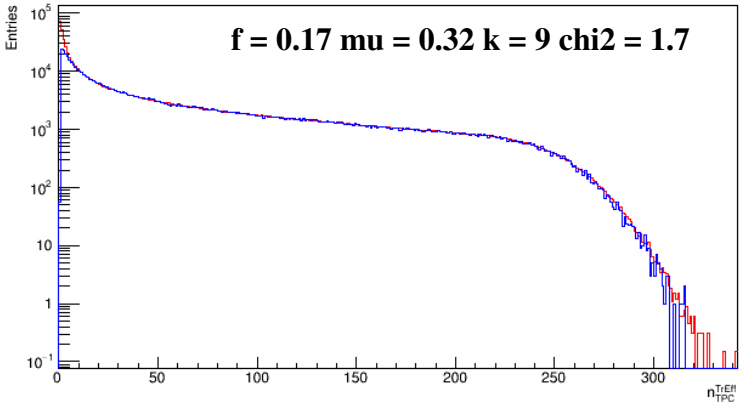


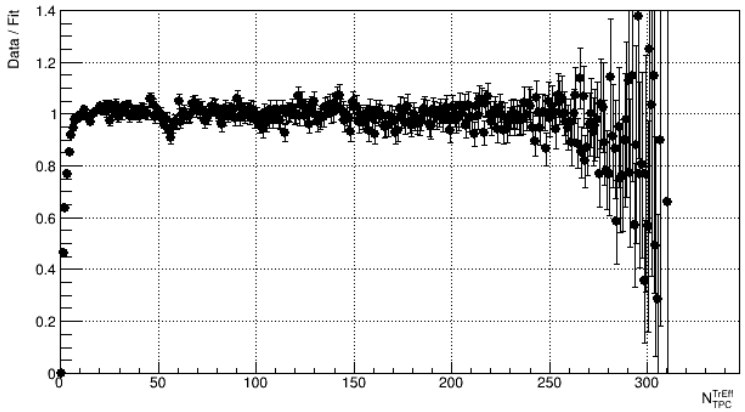




Glauber fit to N_{TPC}^{TrEff} distribution, DCM-QGSM-SMM

- Only for good events, isEventOk:
  - ✓ reconstructed vertex: z-vertex !=0
  - ✓ reconstructed vertex is outside of the FFD: |z-vertex| < 130 cm
  - ✓ number of tracks: $N_{TPC} > 0$, track selections: nhits > 10; $p_T > 0.1$ GeV/c; DCA < 2.0 cm; $|\eta| < 0.5$
  - ✓ Rndm() > TrigEff[N_{TPC}]
- Event multiplicity is calculated using weight for each track ~ 1/RecEff(z-vertex, η)
- Fit range: 9-308

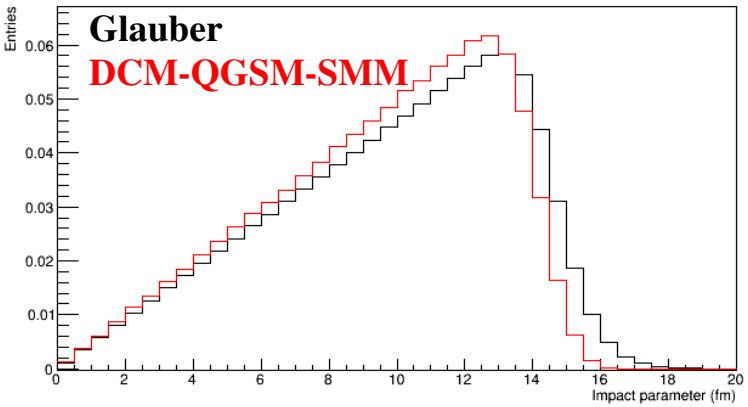


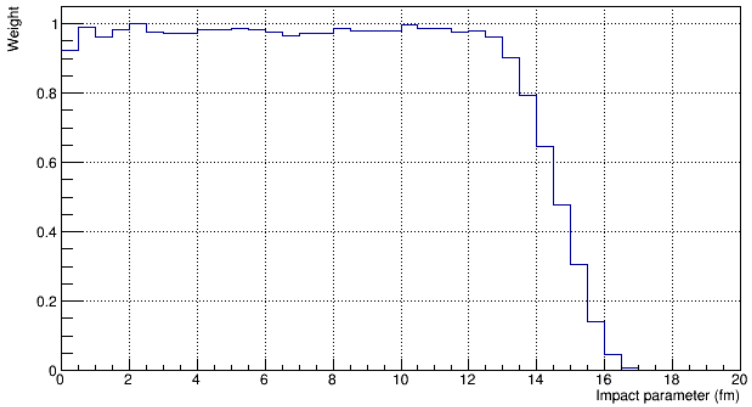


- Predicted trigger efficiency: Integral(data) / Integral (fit) = 83%
- Close but not quite the simulated 91% → why?

Impact parameter distributions, DCM-QGSM-SMM

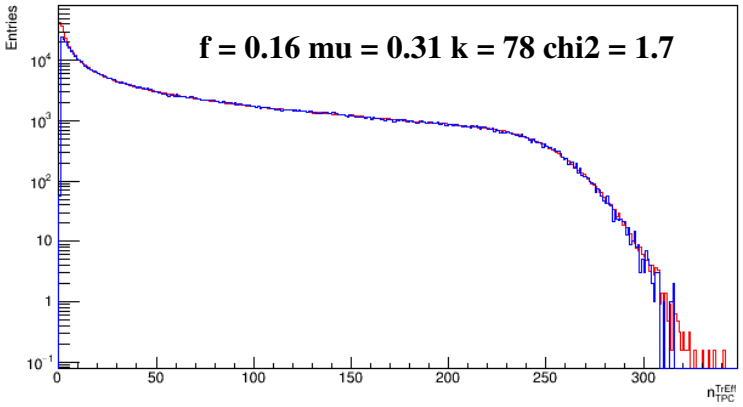
- Lets compare impact parameter distributions in Glauber and DCM-QGSM-SMM
- Distributions are different at b > 12 fm → different radii, definition of inelastic collisions ???
- Glauber can be reweighted to have the same b-distribution as in DCM-QGSM-SMM

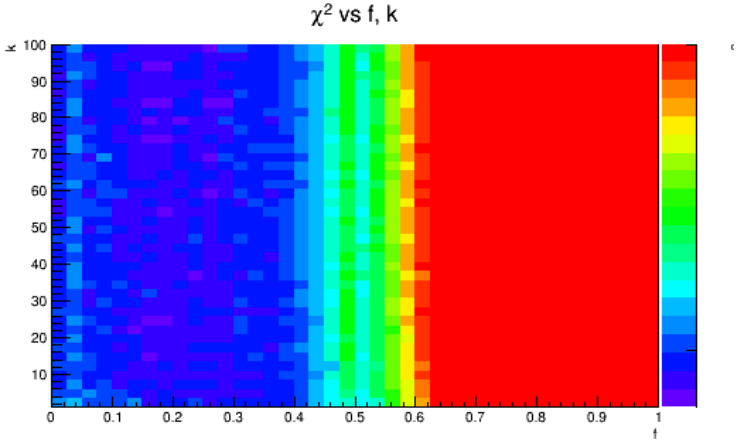


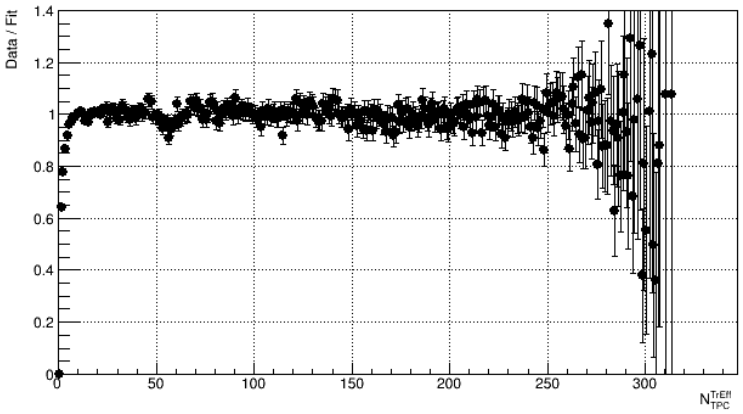


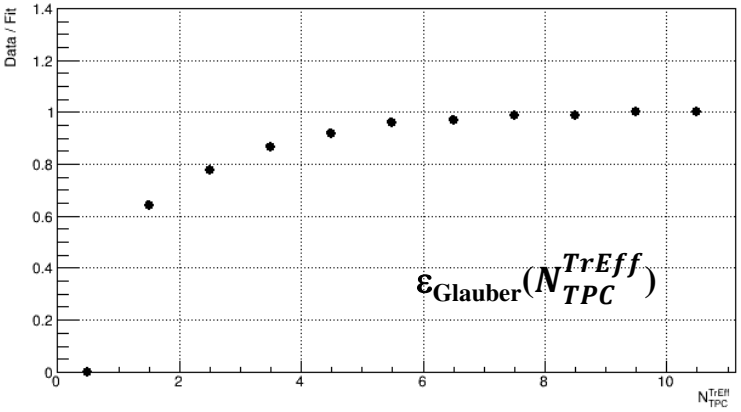
Weighted Glauber fit to N_{TPC}^{TrEff} distribution, DCM-QGSM-SMM

- Same conditions as in slide 7, but with weights for Glauber events by b-value





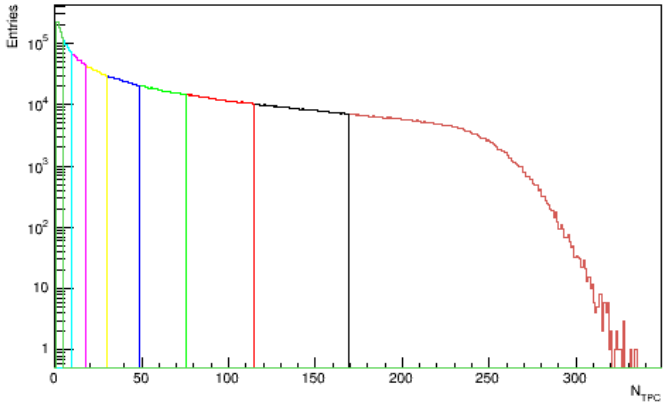


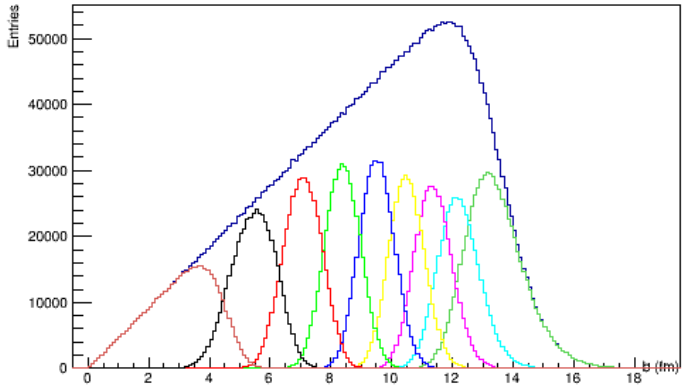


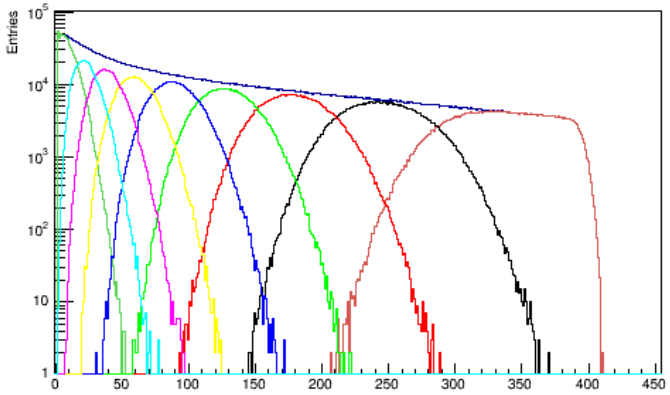
- Predicted trigger efficiency: Integral(data) / Integral (fit) = 90% ≅ simulated 91%
- Turn on curve is very similar to the simulated one (it should not be identical)

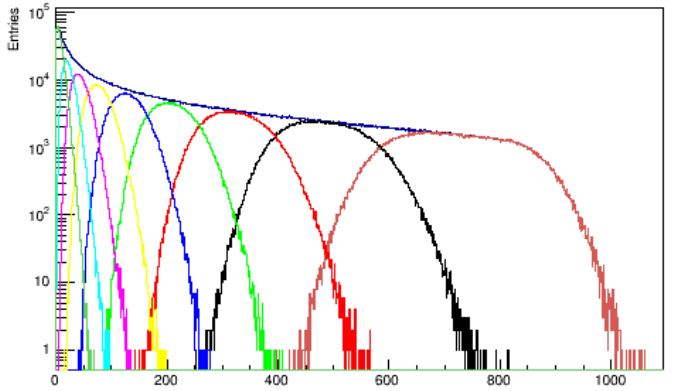
Centrality bins in Glauber, DCM-QGSM-SMM

- $P(N_{TPC}^{TrEff}) = \varepsilon_{\text{Glauber}}(N_{TPC}^{TrEff}) \cdot \sum_{N_a} NBD(\mu N_a, kN_a) \times MCG(N_a)$
- Fraction of the cross section seen = 90%
- Percentile multiplicity splitting: 0-10%, 10-20%, … 80-90%



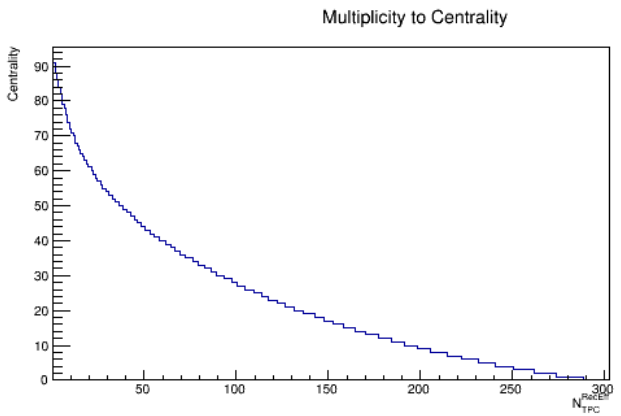


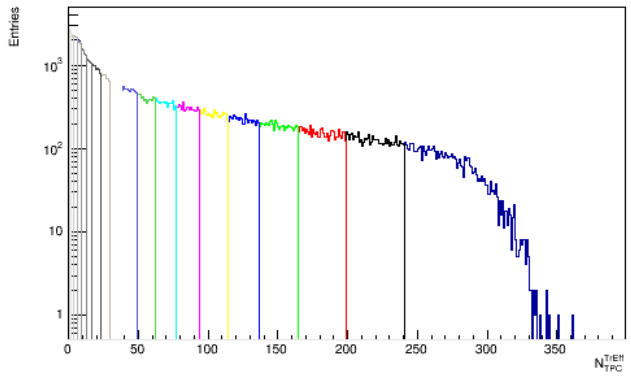


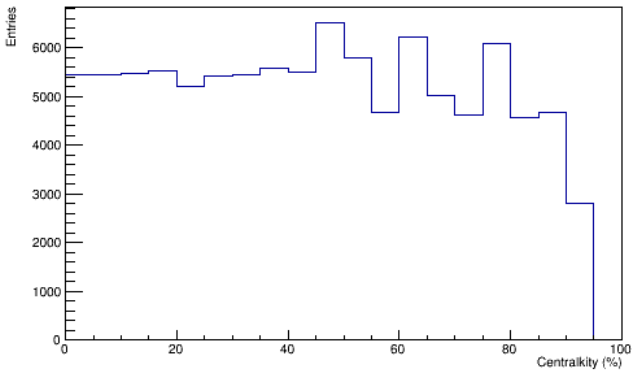


Centrality by TPC multiplicity, UrQMD

- Only for good events, isEventOk:
  - ✓ event not empty: $N_{generated\_primary} > 2 * 209$
  - ✓ reconstructed vertex: z-vertex !=0
  - ✓ reconstructed vertex is outside of the FFD: |z-vertex| < 130 cm
  - ✓ number of tracks: $N_{TPC} > 0$, track selections: nhits > 10; $p_T > 0.1$ GeV/c; DCA < 2.0 cm; $|\eta| < 0.5$
  - ✓ Rndm() > TrigEff[N_{TPC}]
- Assume that multiplicity distribution samples 91% of the total cross section (~ 88%)
- Event multiplicity is calculated using weight for each track ~ 1/RecEff(z-vertex, η)
- Centrality is defined as percentile of the total multiplicity with maximum of 91%

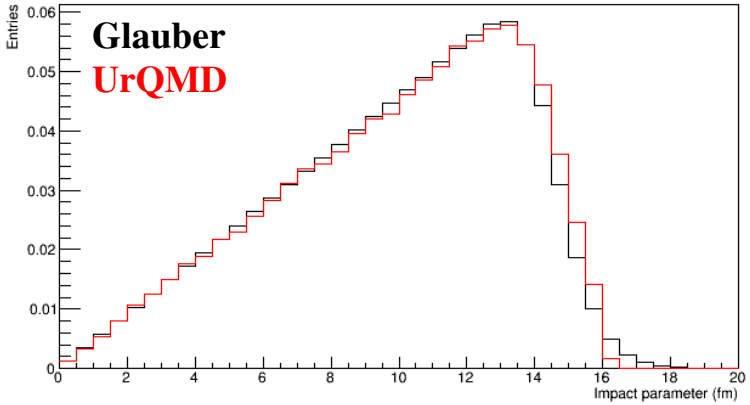


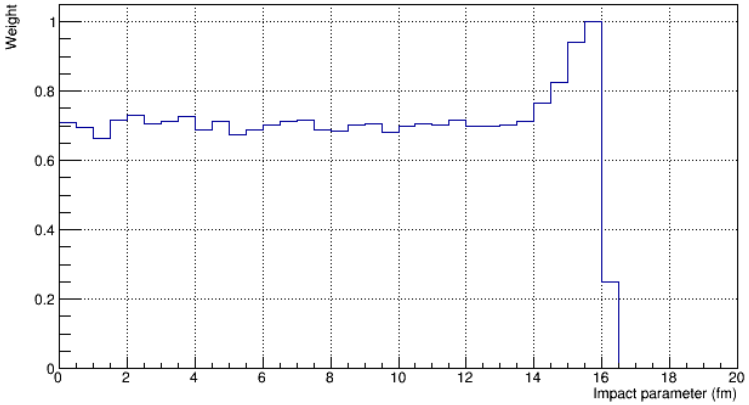




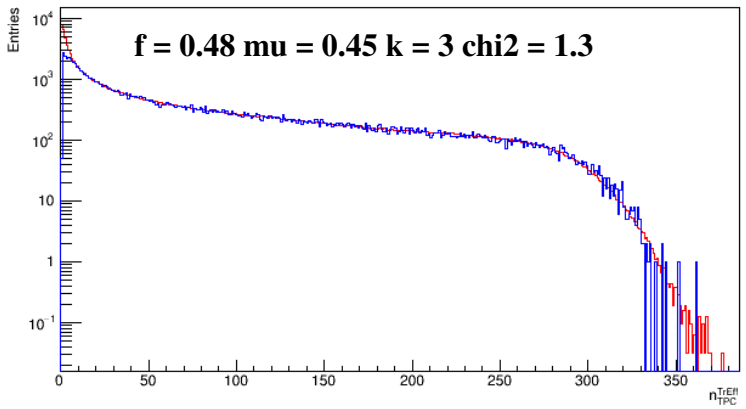
Impact parameter distributions, UrQMD

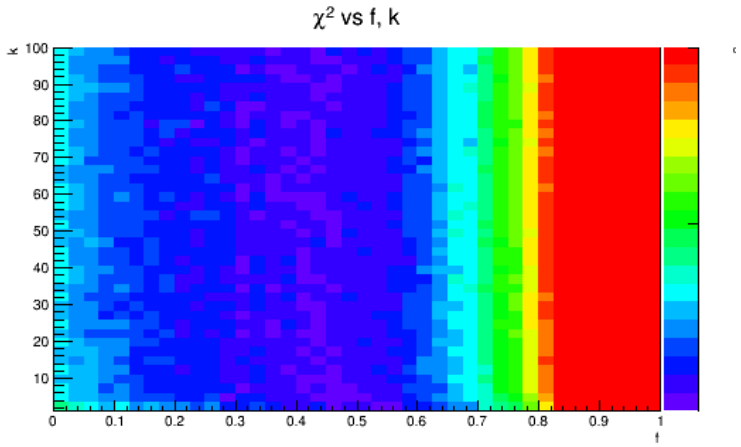
- Lets compare impact parameter distributions in Glauber and UrQMD
- Distributions are different at b > 12 fm → different radii, definition of inelastic collisions ???
- Glauber can be reweighted to have the same b-distribution as in UrQMD

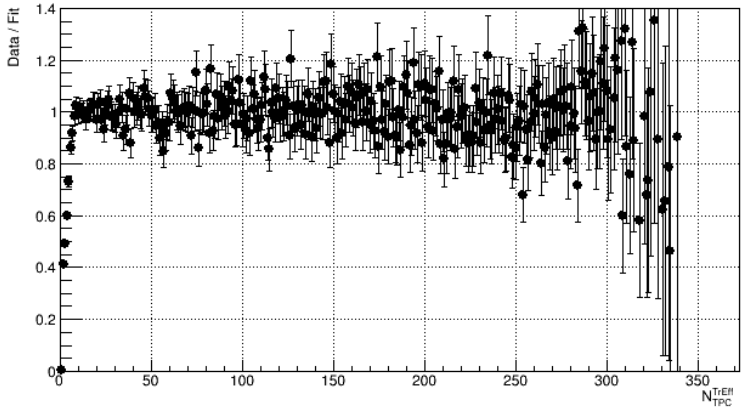


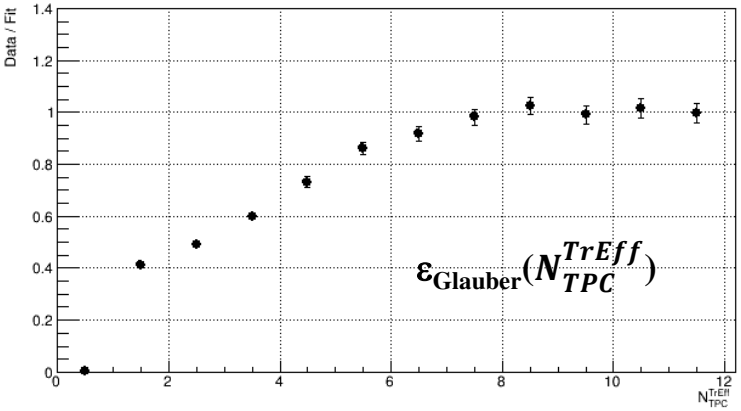


Weighted Glauber fit to N_{TPC}^{TrEff} distribution, UrQMD





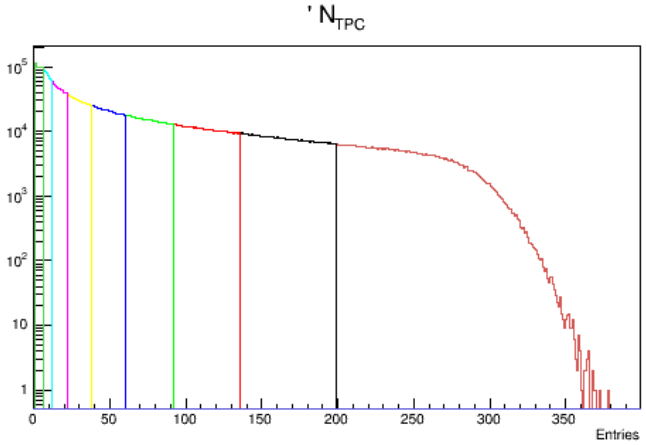


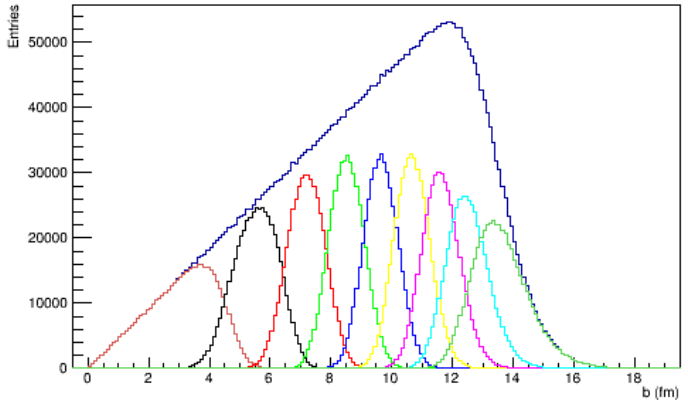


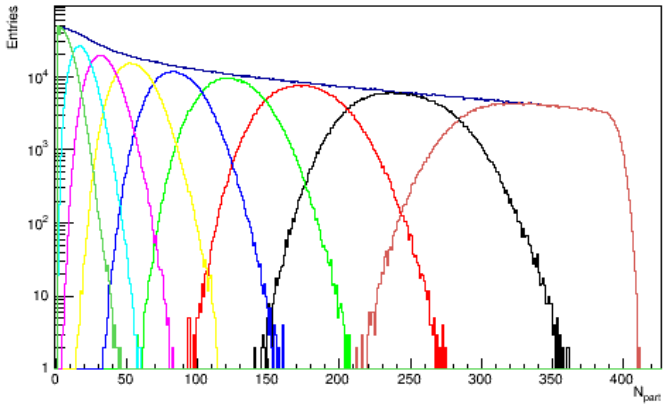
- Predicted trigger efficiency: Integral(data) / Integlal (fit) = 85% ≅ simulated 91 (88)%
- Turn on curve is very similar to the simulated one (it should not be identical)

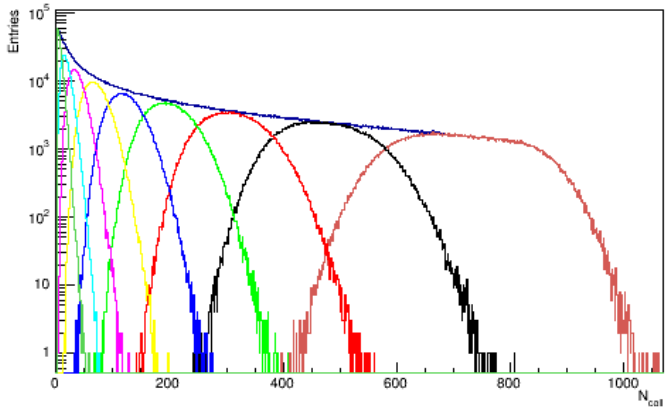
Centrality bins in Glauber, UrQMD

- $P(N_{TPC}^{TrEff}) = \varepsilon_{\text{Glauber}}(N_{TPC}^{TrEff}) \cdot \sum_{N_a} NBD(\mu N_a, k N_a) \times MCG(N_a)$
- Fraction of the cross section seen = 90%
- Percentile multiplicity splitting: 0-10%, 10-20%, … 80-90%



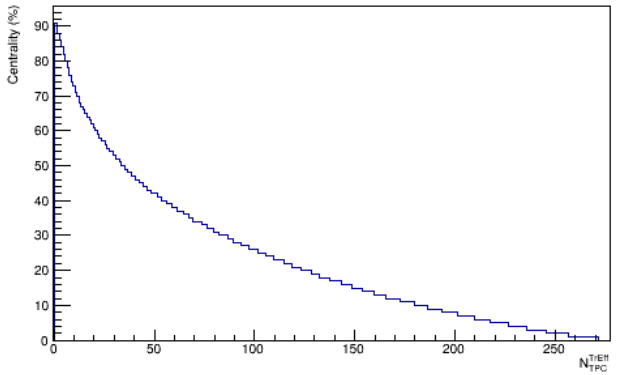


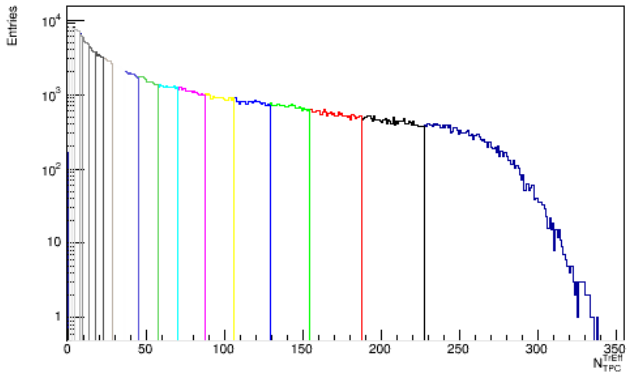


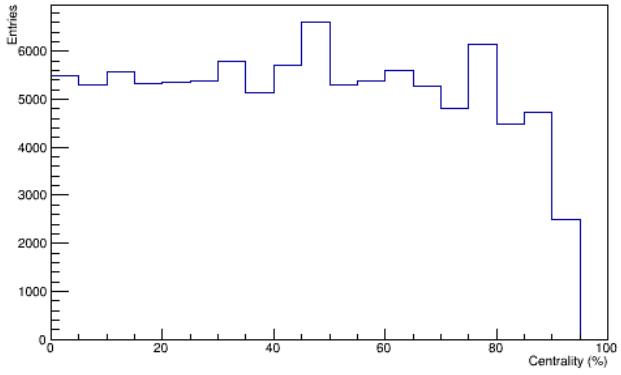


Centrality by TPC multiplicity, PHSD

- Only for good events, isEventOk:
  - ✓ event not empty: $N_{generated\_primary} > 2 * 209$
  - ✓ reconstructed vertex: z-vertex !=0
  - ✓ reconstructed vertex is outside of the FFD: |z-vertex| < 130 cm
  - ✓ number of tracks: $N_{TPC} > 0$, track selections: nhits > 10; $p_T > 0.1$ GeV/c; DCA < 2.0 cm; $|\eta| < 0.5$
  - ✓ Rndm() > TrigEff[N_{TPC}]
- Assume that multiplicity distribution samples 91% of the total cross section (~ 90%)
- Event multiplicity is calculated using weight for each track ~ 1/RecEff(z-vertex, η)
- Centrality is defined as percentile of the total multiplicity with maximum of 91%

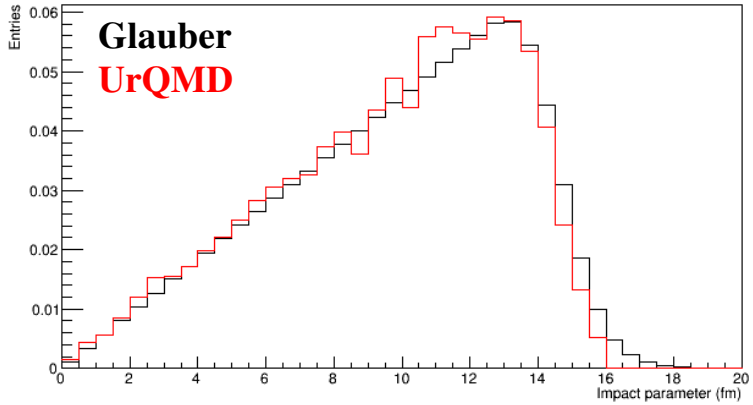


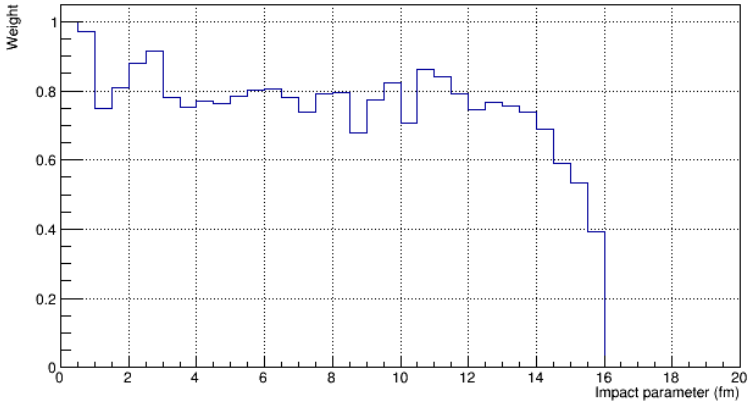




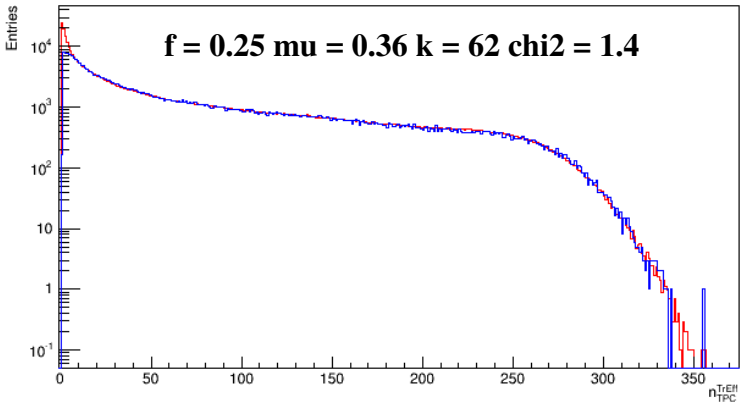
Impact parameter distributions, PHSD

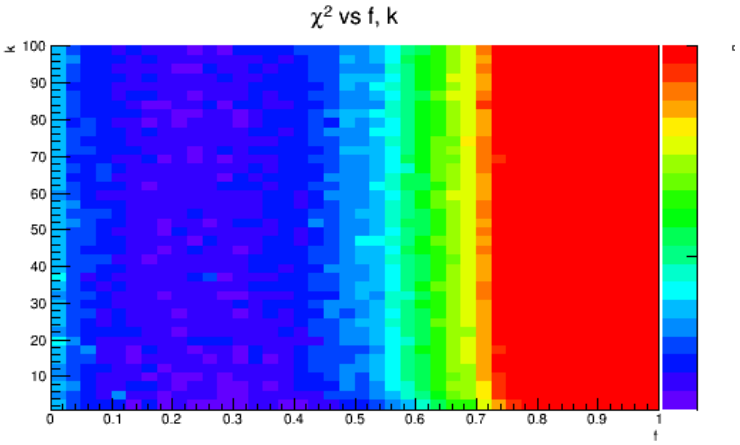
- Lets compare impact parameter distributions in Glauber and UrQMD
- Distributions are different at b > 12 fm → different radii, definition of inelastic collisions ???
- Glauber can be reweighted to have the same b-distribution as in UrQMD

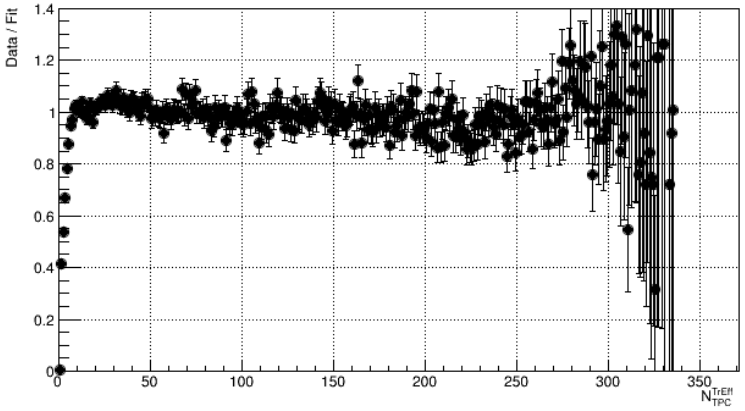


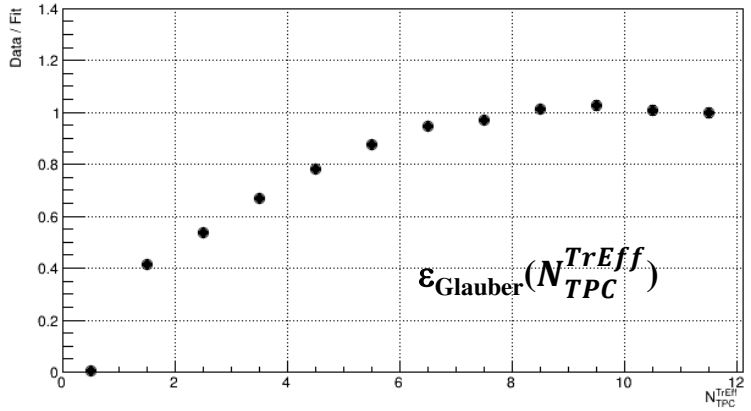


Weighted Glauber fit to N_{TPC}^{TrEff} distribution, PHSD





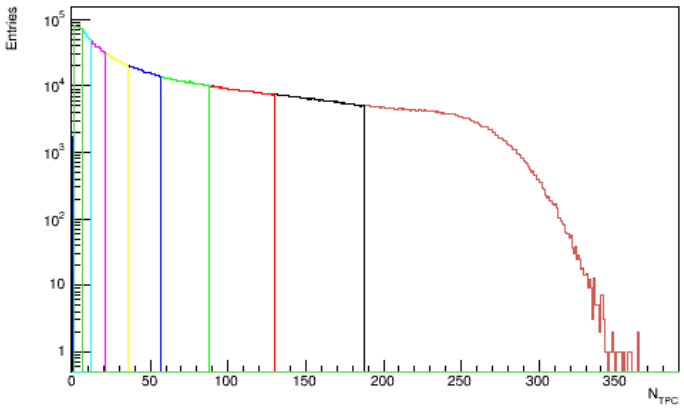


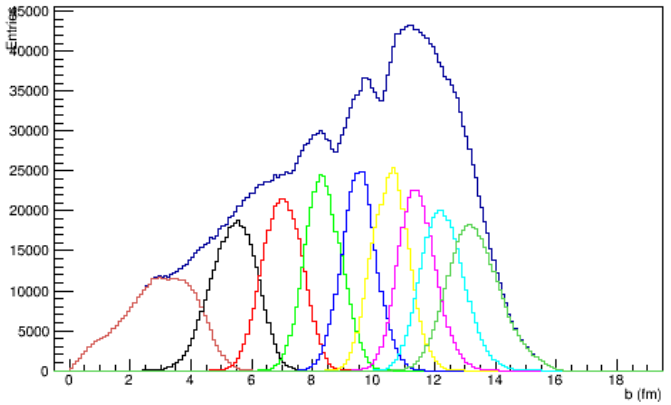


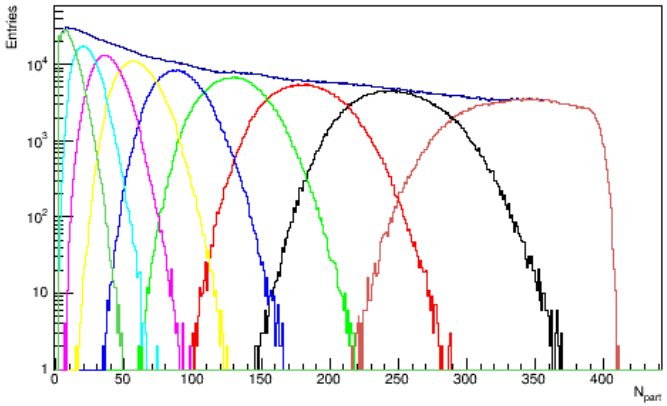
- Predicted trigger efficiency: Integral(data) / Integlal (fit) = 87% ≅ simulated 91 (90)%
- Turn on curve is very similar to the simulated one (it should not be identical)

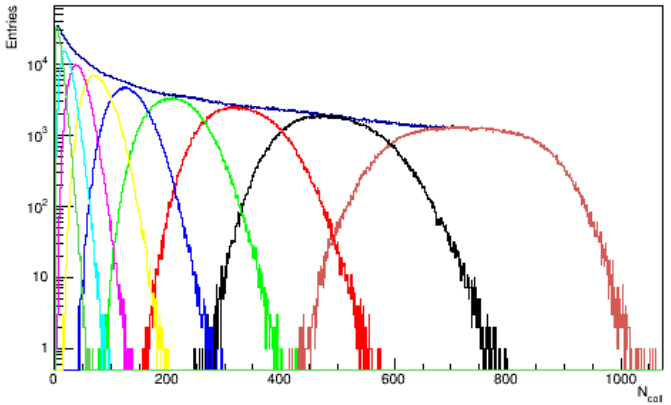
Centrality bins in Glauber, PHSD

- $P(N_{TPC}^{TrEff}) = \varepsilon_{\text{Glauber}}(N_{TPC}^{TrEff}) \cdot \sum_{N_a} NBD(\mu N_a, kN_a) \times MCG(N_a)$
- Fraction of the cross section seen = 90%
- Percentile multiplicity splitting: 0-10%, 10-20%, … 80-90%

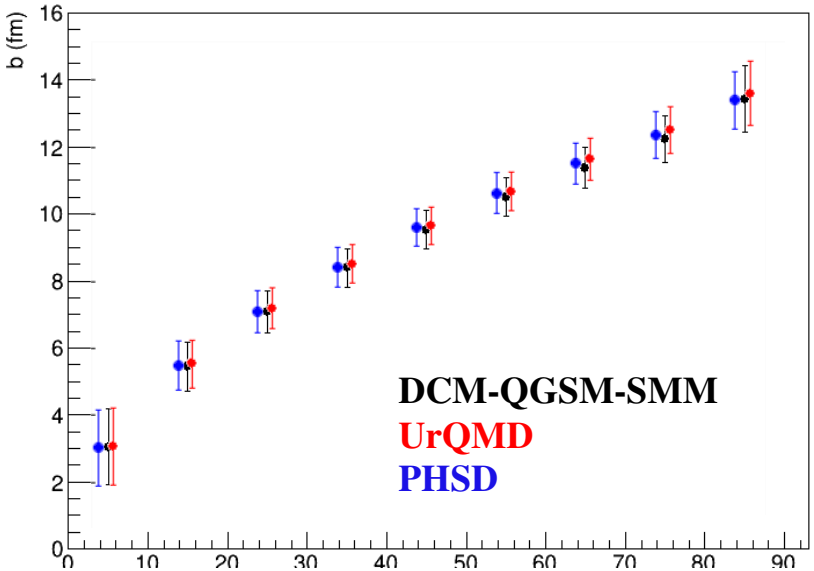


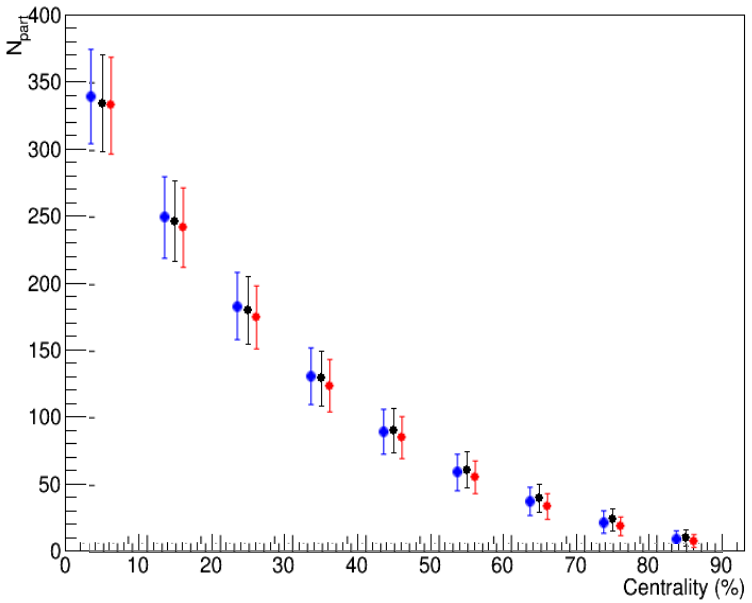


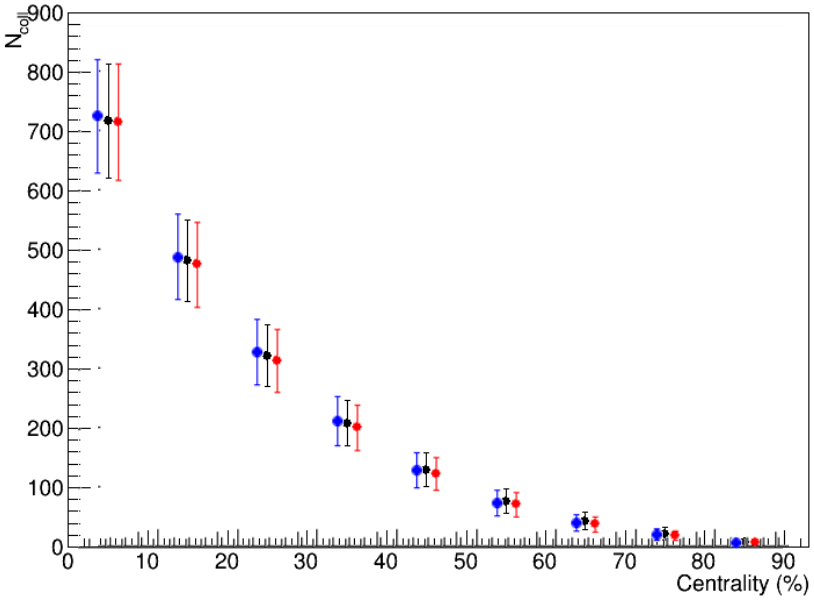




Glauber, model comparison







- Glauber fit results are very close for the two event generators

Conclusions

- Centrality procedure with TPC multiplicity has been developed
- Centrality by TPC reduces effective trigger efficiency by ~ 5%
  - ✓ ~ 90% with FFD||FHCAL trigger
  - ✓ ~ 83% with FFD trigger