

Восстановление плоскости реакции с помощью BBC в SPD

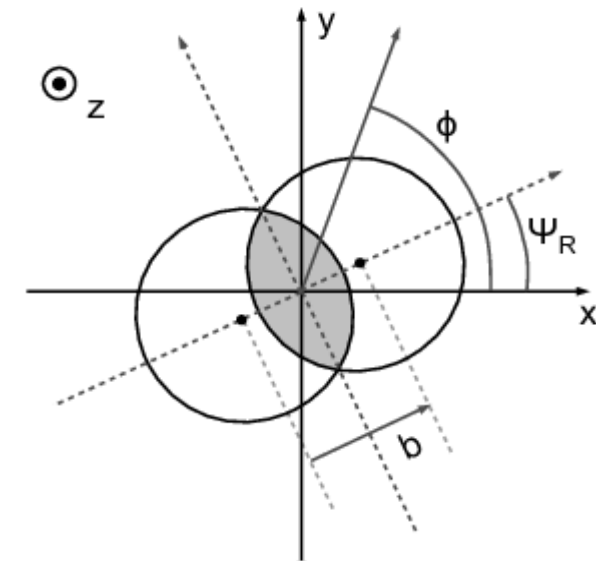
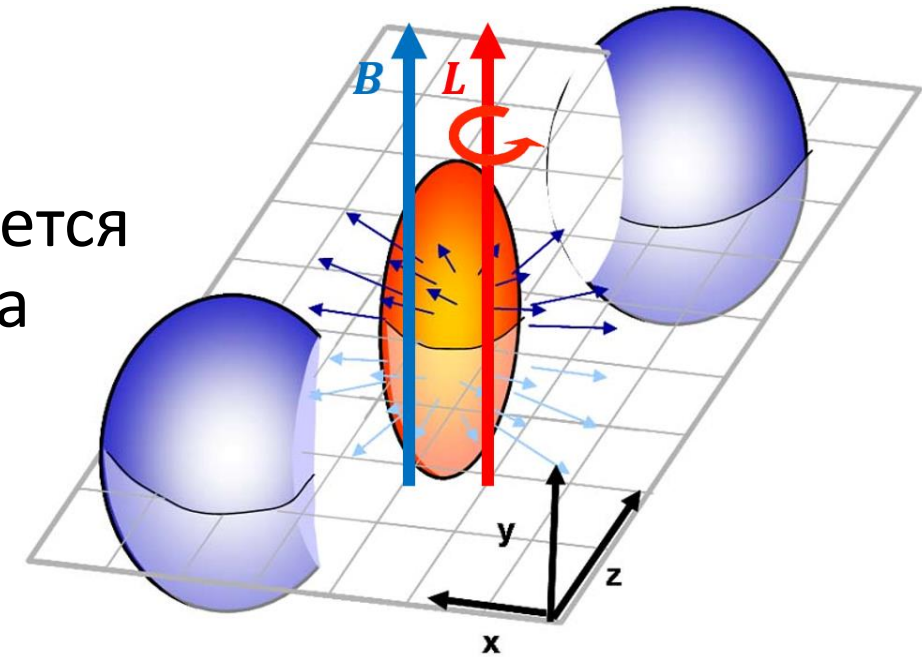
Алпатов Е.В.

Совместное совещание групп МИФИ-ЛФВЭ в МИФИ

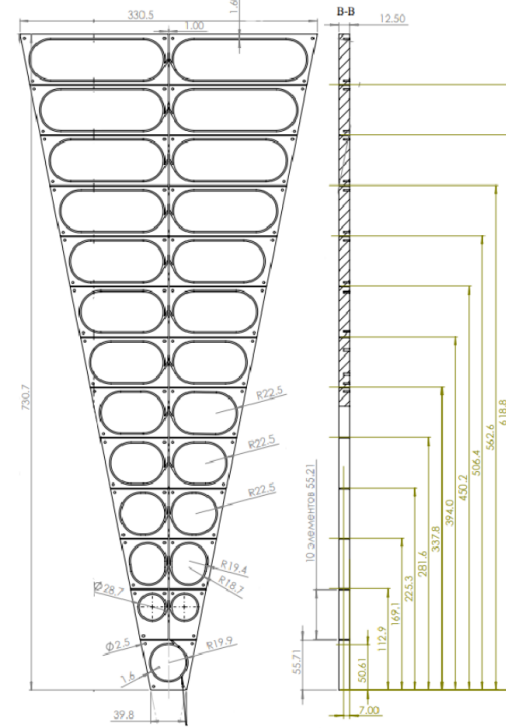
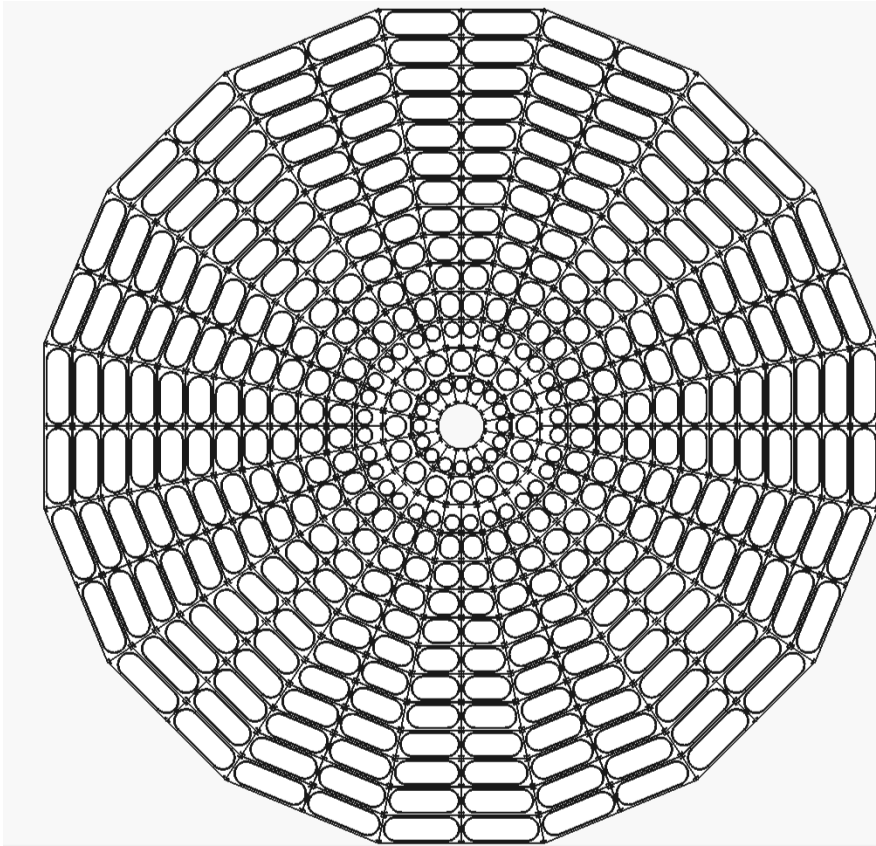
14 октября 2025

Мотивация

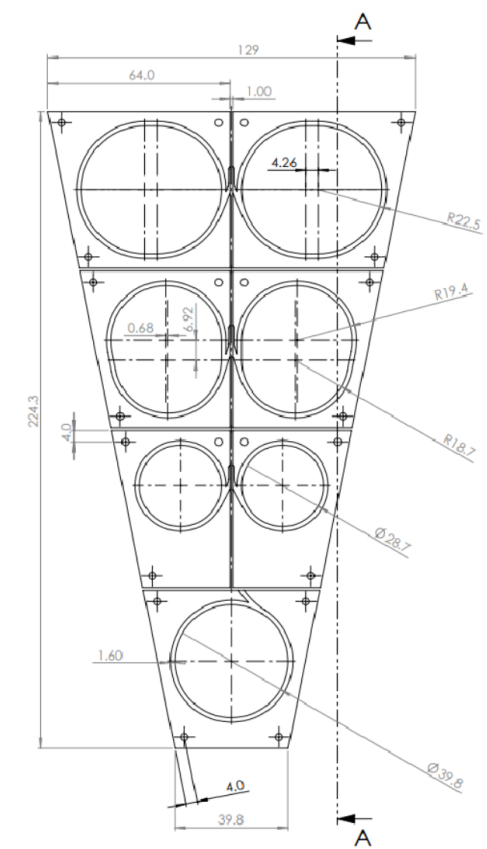
- Плоскость реакции в столкновениях ионов задается прицельным параметром и направлением пучка
 - Азимутальный угол плоскости - Ψ_{RP}
- Азимутальное распределение частиц, разложенное в ряд Фурье:
 - $E \frac{d^3N}{d^3p} = \frac{1}{2\pi} \frac{d^2N}{p_T dp_T dy} (1 + \sum_{n=1}^{\infty} 2v_n \cos(n(\varphi - \Psi_{RP})))$
 - $v_n = \langle \cos[n(\varphi_i - \Psi_{RP})] \rangle$
- Глобальную поляризацию гиперонов считают с помощью плоскости реакции
 - $P_H = \frac{8}{\pi\alpha_H} \langle \sin(\psi_{RP} - \varphi_b^*) \rangle$
- Фемтоскопические исследования, СМЕ и т.д.



BBC в SPD



(a)

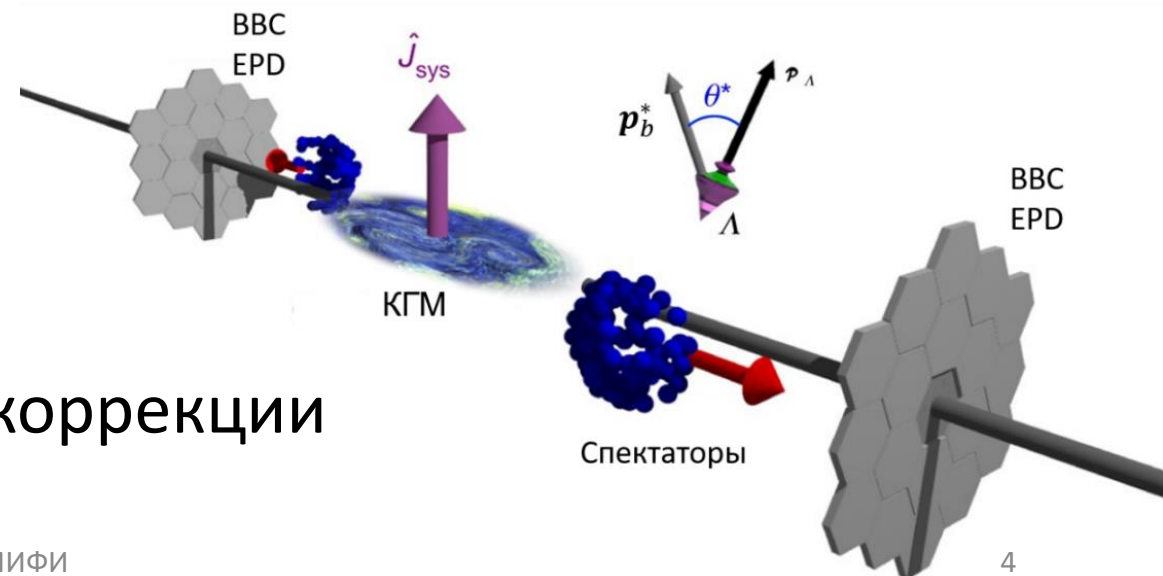


(b)

- **Задача:** написание программного пакета для восстановления плоскости реакции с помощью детектора BBC

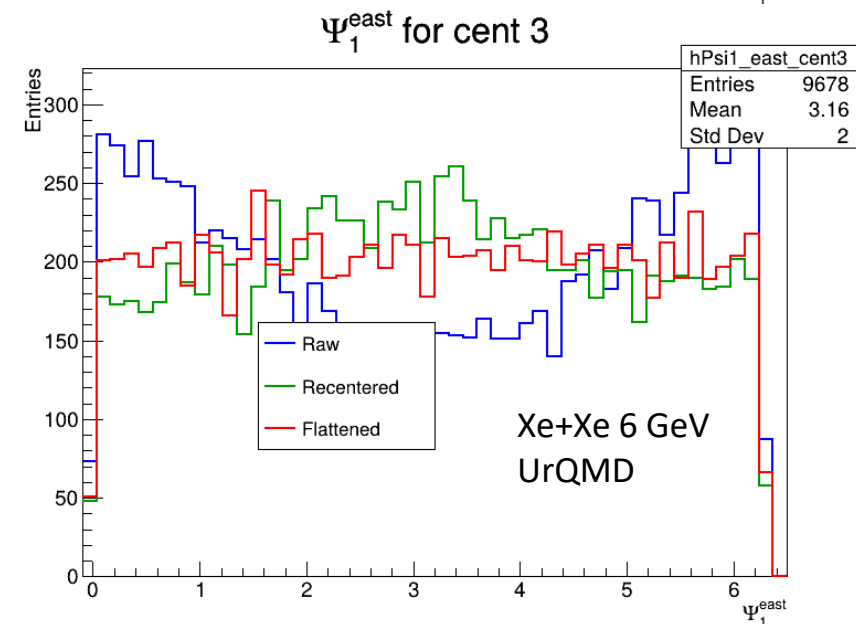
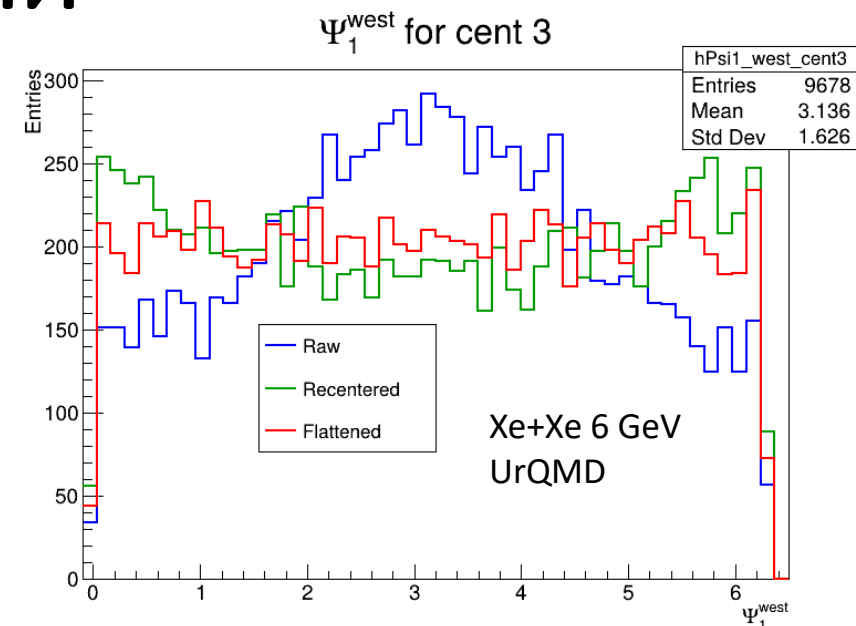
Как восстанавливается плоскость?

- Метод плоскости событий:
 - $Q_{n,x} = \sum_i w_i \cos(n\varphi_i) = Q_n \cos(n\Psi_n)$
 - $Q_{n,y} = \sum_i w_i \sin(n\varphi_i) = Q_n \sin(n\Psi_n)$
 - $\Psi_n = \arctan^2(Q_{n,y}, Q_{n,x})/n$
 - $v_n^{\text{obs}}(p_T, y) = \langle \cos[n(\varphi_i - \Psi_n)] \rangle$
- Плоскость реакции $\sim$ плоскость событий первого порядка
- Ψ_1 восстанавливается по спектаторам (непрореагировавшие нуклоны)
- **BBC в SPD можно использовать для восстановления Ψ_1**
- Ограниченный аксептанс $\rightarrow$ требуются коррекции

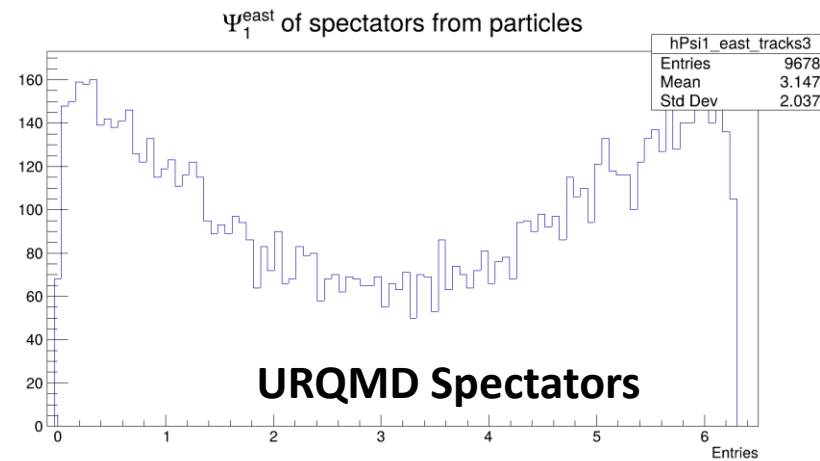
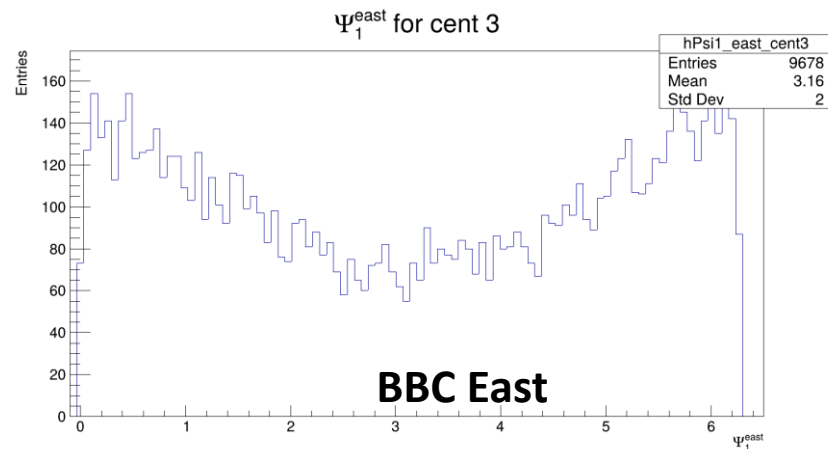
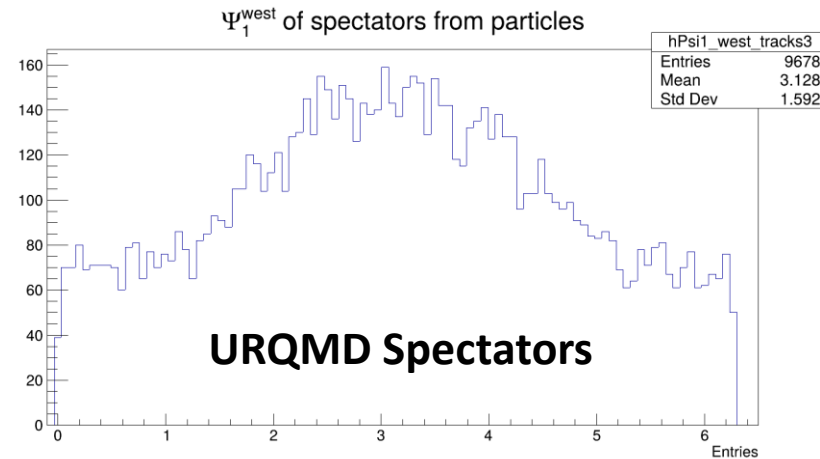
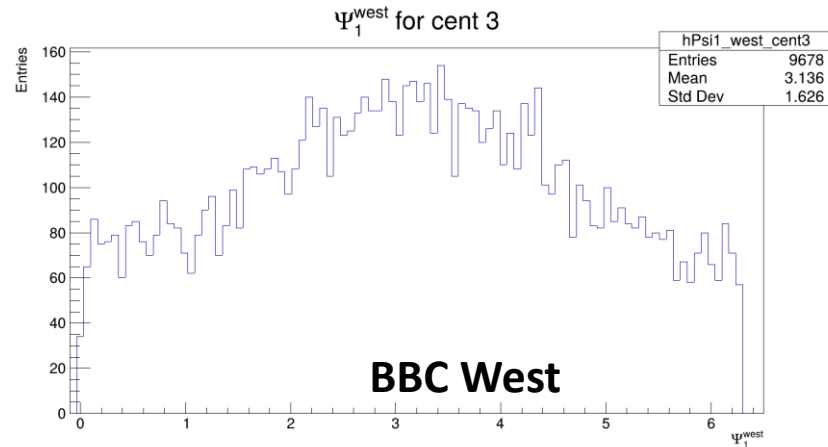


Коррекции плоскости событий

- Дополнительное взвешивание w_i
- Отцентровка (Recentering)
 - $Q_{x,y}^{recentered} = Q_{x,y} - \langle Q_{x,y} \rangle$
- Выполяживание (Flattening, Shift)
 - $\psi^{flat} = \psi + \Delta\psi$
 - $\Delta\psi = \sum_n [A_n \cos(n\psi) + B_n \sin(n\psi)]$
 - $A_n = -\frac{2}{n} \langle \sin(n\psi) \rangle$; $B_n = \frac{2}{n} \langle \cos(n\psi) \rangle$
 - $\psi = \tan^{-1} \left(\frac{\sin(\psi)}{\cos(\psi)} \right)$



URQMD: описывают ли спектаторы RP?



- В URQMD угол плоскости реакции должен быть равен нулю
- Ожидаем ярко выраженные пики
- Результат – сильное размытие

Особенности программы

- Состоит из 3-х классов
 - SpdBbcErFinder – основной класс для восстановления плоскостей. Позволяет восстанавливать несколько плоскостей до заданного порядка, сохранять и использовать коррекции, обрабатывать события и получать результаты
 - SpdErFinderResult – контейнер с результатами, хранит плоскости события и Q-вектора для требуемых плоскостей и всех этапов коррекций
 - SpdBbcGeom – вспомогательный класс, описывающий геометрию детектора. Задается количество рядов, количество тайлов в рядах, их длина, внутренний радиус детектора. Позволяет найти координату пересечения с детектором и по координате – ряд и номер тайла.

Пример использования

```
1  #include "SpdBbcEpFinder/SpdBbcGeom.h"
2  #include "SpdBbcEpFinder/SpdEpFinder.h"
3
4  //Load library libSpdBbc.so
5  gSystem->Load("SpdBbcEpFinder/libSpdBbc.so")
6
7  //Histograms and class initialization
8  SpdBbcGeom *geom = new SpdBbcGeom();
9  SpdEpFinder *epFinder = new SpdEpFinder(nEventBins, OutputCorrectionFileName, InputCorrectionFileName)
10 // Set epFinder variables
11 for (int iRow = 1; iRow <= geom->NumRows(); iRow++) {epFinder->EnableRow(iRow, true)}
12 epFinder->SetAdcMinThreshold(adcMin);
13 epFinder->SetMaxTileWeight(maxTileWeight);
14
15 //Event Loop: Fill bbc data, calculate corrections and measure Psi
16 ...
17 for (int iRow = 1; iRow < geom->NumRows(); iRow++) {
18     for (int iTile = 0; iTile < geom->TilesInRow(iRow); iTile++) {
19         epFinder->SetBbcAdcW(iRow, iTile, bbcAdcW[iRow][iTile]);
20         epFinder->SetBbcAdcE(iRow, iTile, bbcAdcE[iRow][iTile]);
21     }
22 }
23 SpdEpFinderResult epResult = epFinder->Compute(eventBin);
24 TVector2 Qe = epResult.Qe(order, correction);
25 double PsiE = epResult.PsiE(order, correction);
26 ...
27 //End of analysis: Save correction histograms
28 epFinder->Finish();
```

```
31 float bbcAdcW[rows][tiles], bbcAdcE[rows][tiles];
32 //Track loop: find intersections and tile data
33 ...
34 double hitX, hitY, hitZ;
35 int detectorSide;
36 bool intersects = geom->CalculateIntersection(track.x, track.y, track.z,
37     track.px, track.py, track.pz,
38     hitX, hitY, hitZ, detectorSide);
39 if intersects {
40     bool tileGet = geom->GetTileRowByIntersection(hitX, hitY, row, tile);
41     if tileGet {
42         if detectorSide > 0 {bbcAdcW[row][tile] += track.Energy;}
43         else {bbcAdcE[row][tile] += track.adcE;}
44     }
45 }
```


Заключение

- Подготовлен программный пакет для восстановления плоскостей событий с помощью BBC в SPD
- Проведены первые тесты на данных Xe+Xe 6 ГэВ URQMD

Планы

- Выложить код в открытый доступ
- Для тестов получить ν_1 относительно $\Psi_{1,BBC}$
- Проверка на моделировании Geant + UrQMD
- Учет формата хранения данных с BBC в SPD

Дополнительные слайды

Распределение спектаторов в событии в ВВС

