



# Санкт-Петербургский государственный университет

**С.НЕМНЮГИН**

## Участие научной группы Санкт-Петербургского государственного университета в Программе поддержки

- Немнюгин Сергей Андреевич (доцент).
- Машицин Константин Иванович (аспирант).
- Юсупова Анастасия Юрьевна (аспирант).
- Дрюк Андрей Викторович (аспирант).\*
- Козлов Илья Дмитриевич (студент магистратуры, 2 курс).

# **Анализ рождения пи-мезонов и положительных каонов на данных 7 и 8 сеансов эксперимента VM@N**

**Машицин Константин Иванович**

Для экспериментальных данных, полученных во время седьмого сеанса на установке VM@N, будет завершен анализ рождения пи-мезонов и положительных K-мезонов для взаимодействия пучка аргона с различными мишенями. Для экспериментальных данных, полученных во время восьмого сеанса на установке VM@N, будет выполнен анализ рождения положительных и отрицательных пи-мезонов и положительных K-мезонов в столкновениях пучка ксенона и мишени из йодида цезия.

Будет разработана методика идентификации положительных и отрицательных пи-мезонов и положительных K-мезонов по времени пролета, будут получены их спектры в фазовом пространстве. Будет выполнено сравнение данных эксперимента с результатами Монте-Карло моделирования, а также будут получены оценки систематических погрешностей и оценки сечения рождения рассматриваемых частиц.

## Краткое содержание работы

Цель работы - подготовка смоделированных и экспериментальных данных к оценке выходов частиц. Определение выходов – многоэтапная задача: необходимо выбрать надежные экспериментальные события, провести идентификацию, чтобы определить число частиц каждого типа, и нормировать полученный результат на эффективность реконструкции событий и триггерных детекторов.

- Получены оценки числа  $\pi$ -мезонов, родившихся в результате набора данных о взаимодействии пучка ионов аргона с медной мишенью при энергии 3.2 ГэВ на нуклон (7 сеанс).
- Для проведения этого анализа выполнена фильтрация экспериментальных событий для получения надежных экспериментальных данных.
- В процесс моделирования событий добавлены реалистичные эффекты, которые позволили добиться совпадения характеристик данных эксперимента и моделирования.
- Получены оценки эффективности реконструкции триггерной системы, трековых детекторов и восстановления событий.
- Выполнена нормировка числа идентифицированных в эксперименте частиц на эффективность.

Доклад на 10 совещании коллаборации BM@N (май 2023, Санкт-Петербург).

# Входные данные

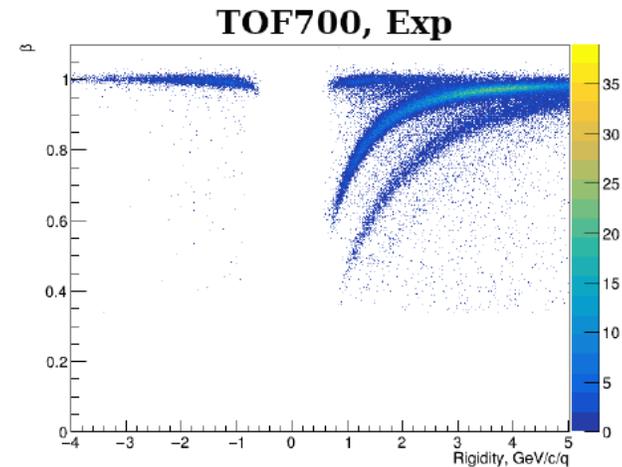
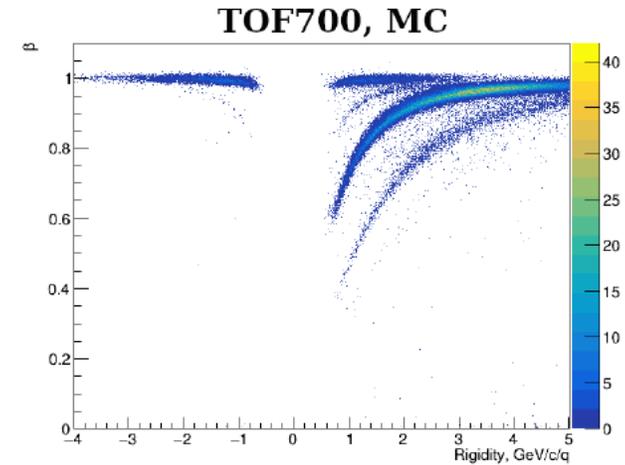
Run: 7 and Tracking: CellAuto

## Monte Carlo

- Generator: DCM-SMM
- System: Ar + Cu
- Energy: 3.2 AGeV
- Lorentz Shifts
- Dead strips, hits

## Exp data

- System: Ar + Cu
- Energy: 3.2 AGeV

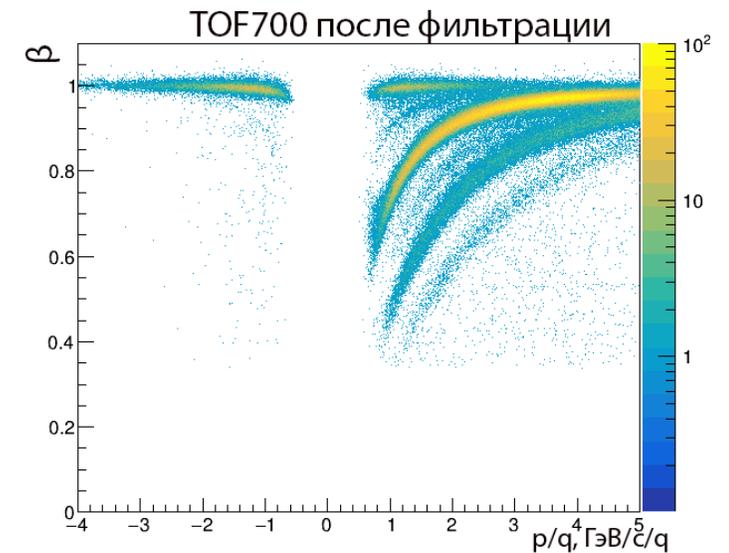
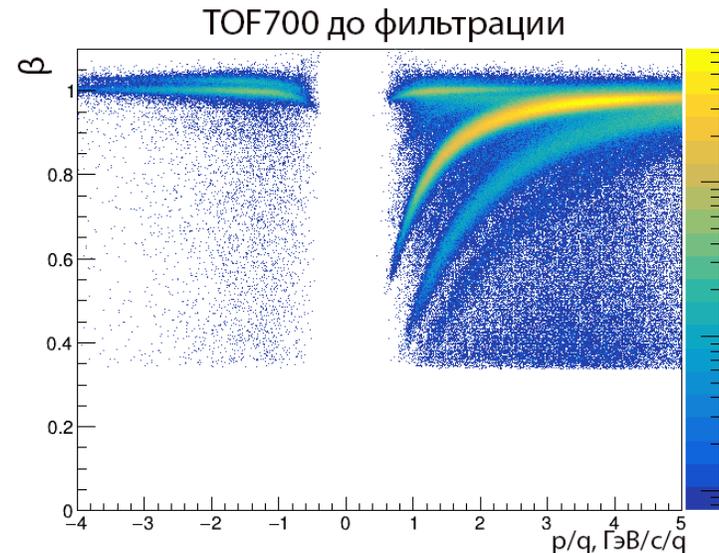
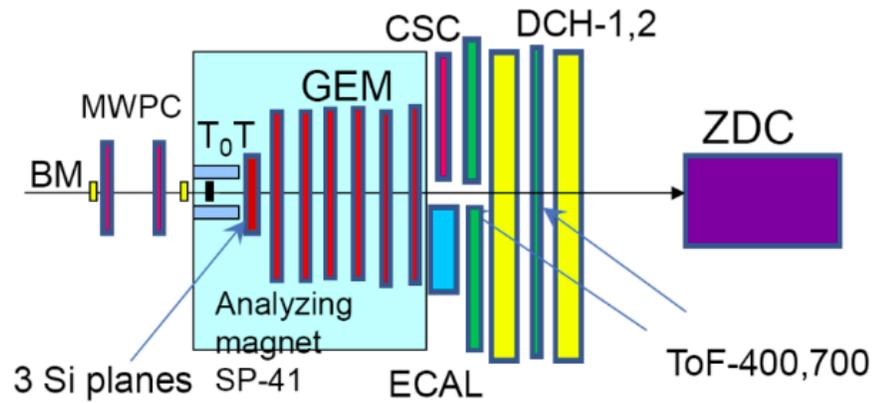


# Фильтрация экспериментальных данных

Критерии отбора надежных экспериментальных данных:

- Вершина находится в интервале  $V_x \in (-2, 4)$  см;  $V_y \in (-6, -1)$  см;  $V_z \in (-5, 5)$  см.  
Минимальное количество хитов в GEM: 4.
- Требования на минимальное количество хитов 2 в Si1 нет.
- Трек подтверждается в первой дрейфовой камере.

Для TOF700 трек ассоциируется с хитом, а для DCH1 с сегментом трека.



## Учет динамических диапазонов

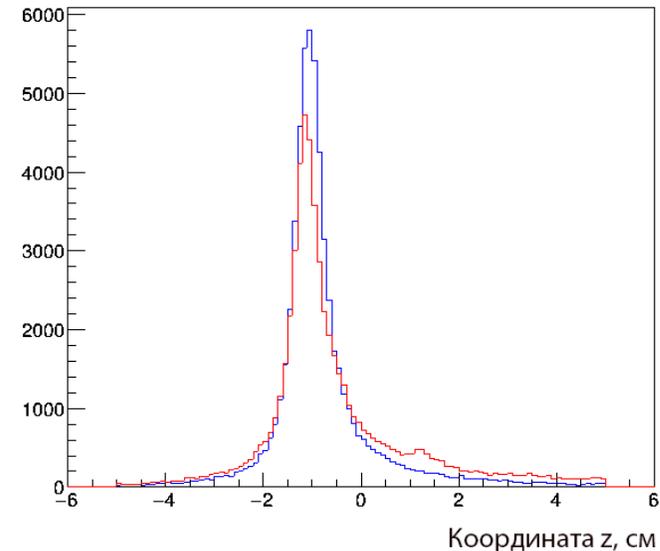
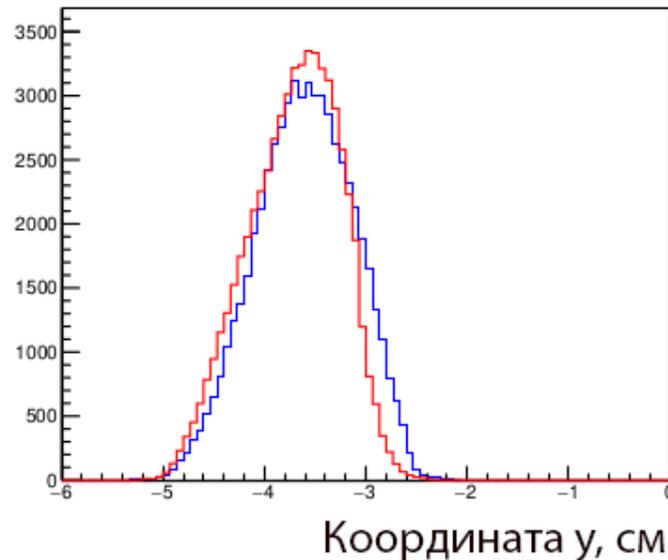
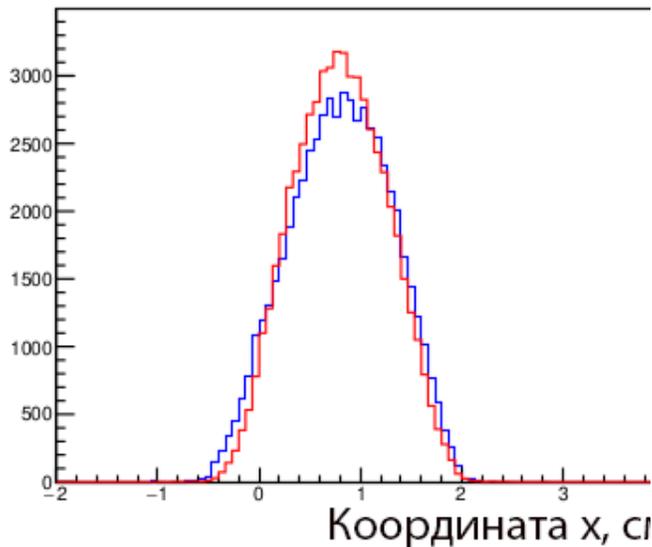
Процедура вычисления «динамических коридоров»:

- Трек экстраполируется до всех возможных хитов детектора.
- Вычисляются невязки, характеризующие отклонение траектории частицы от хита  $\Delta X = x_{\text{track}} - x_{\text{hit}}$ ;  $\Delta Y = y_{\text{track}} - y_{\text{hit}}$ .
- После аппроксимации гистограммы невязок функцией `gaus + pol2` с шагом 350 МэВ, извлекаются средние ( $\mu_x, \mu_y$ ) и стандартные отклонения ( $\sigma_x, \sigma_y$ ). Таким образом, эти параметры будут зависеть от импульса, что позволит более точно объединять треки в различных детекторах.

## Добавление реалистических эффектов

В моделирование включены реалистические эффекты:

- Размытие первичной вершины. Экспериментальные распределения первичной вершины по  $x$ ,  $y$ ,  $z$  координатам аппроксимированы нормальным распределением, получены их средние и стандартные отклонения. Эти параметры добавлены в процесс моделирования.
- Лоренцевский сдвиг электронов в GEM под действием магнитного поля.
- Учет неработающих каналов детектора. Из экспериментальных данных получен список неработающих стрипов. Для этого из данных было получено число откликов каждой считывающей полосы, если она не сработала ни разу, то этот стрип добавлялся в список. Затем, на этапе реконструкции смоделированных данных, сигнал с отмеченных стрипов не записывался.
- Настройка эффективности для детекторов трековой системы (GEM+кремниевые).

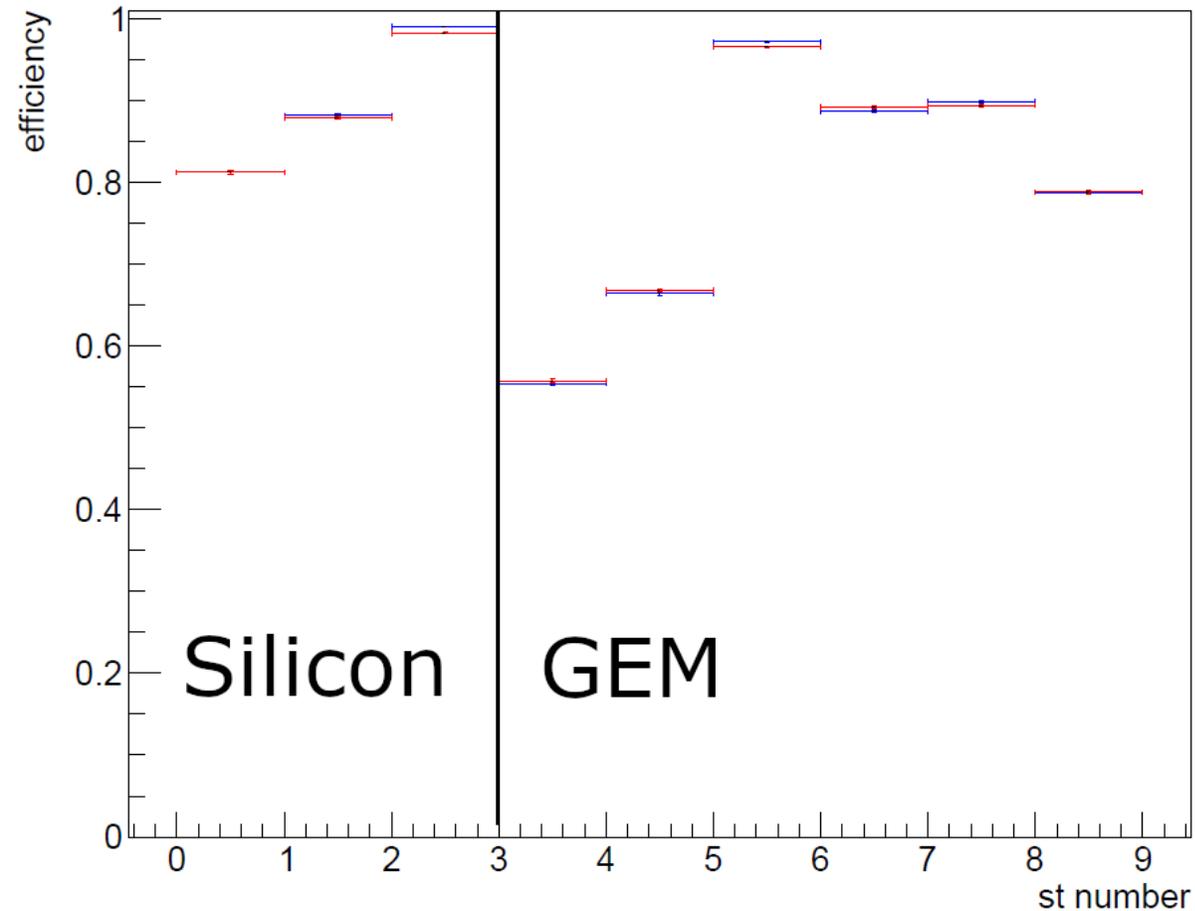


# Эффективность станций в магните

$$\text{Efficiency} = N_{\text{hit}} / N_{\text{acceptance}}$$

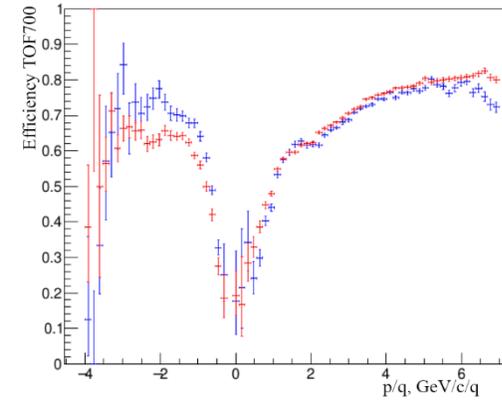
$N_{\text{hit}}$  – количество треков,  
оставивших ХИТ в станции;

$N_{\text{acceptance}}$  – общее  
количество треков,  
прошедших через рабочую  
область станции.

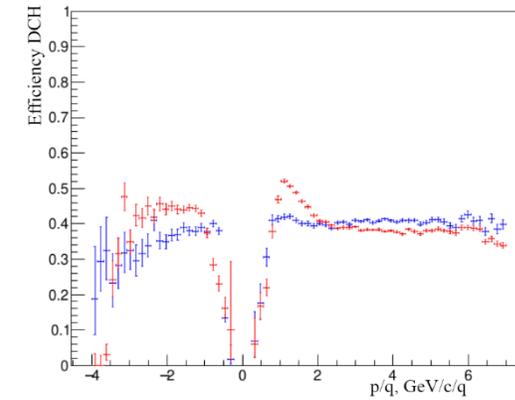
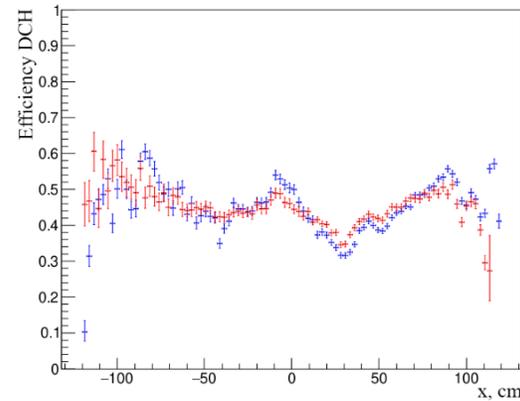


# Эффективность станций и триггеров

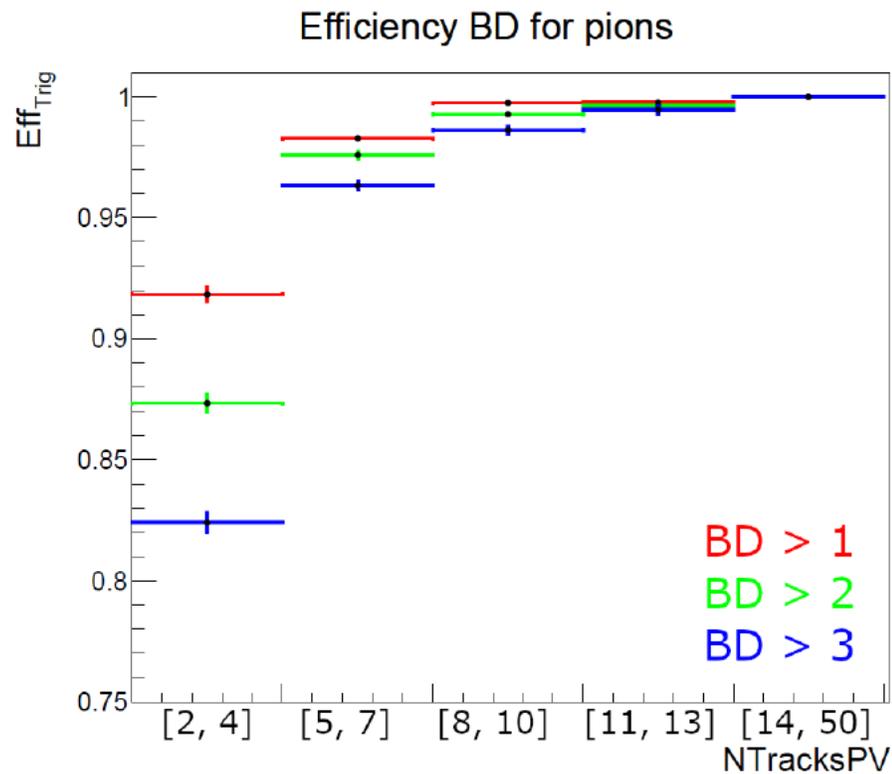
Коррекция эффективности TOF700.



Коррекция эффективности DCH.



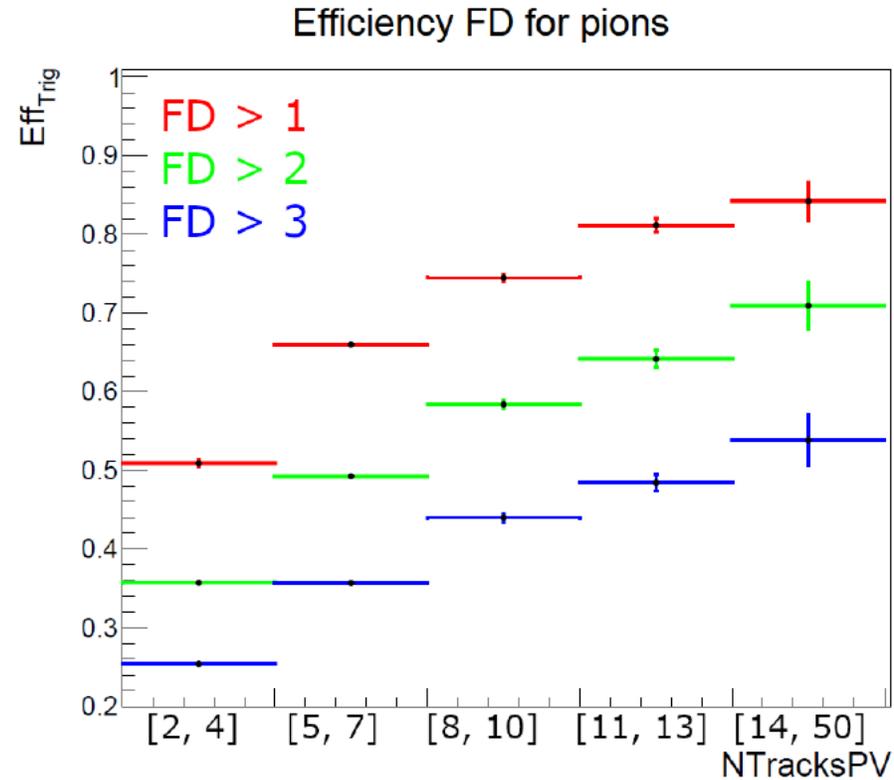
# Оценка эффективности триггера BD



$$\varepsilon(\text{BD} > k) = \frac{N_{\pi}(\text{BD} > k, \text{FD} > N, N_{\text{tr}})}{N_{\pi}(\text{FD} > N, N_{\text{tr}})},$$

$\pi$	C	Al	Cu	Sn	Pb
BD > 1	0.72	0.94	0.97	0.99	0.99
BD > 2	0.57	0.90	0.96	0.99	0.99
BD > 3	0.44	0.85	0.94	0.98	0.98

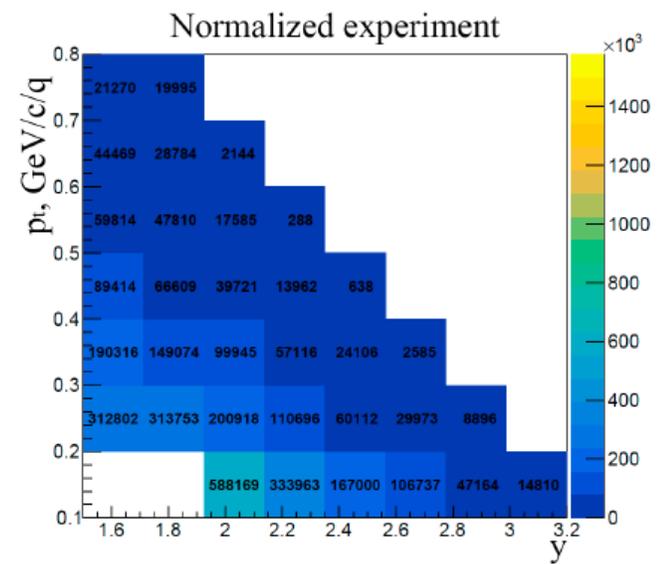
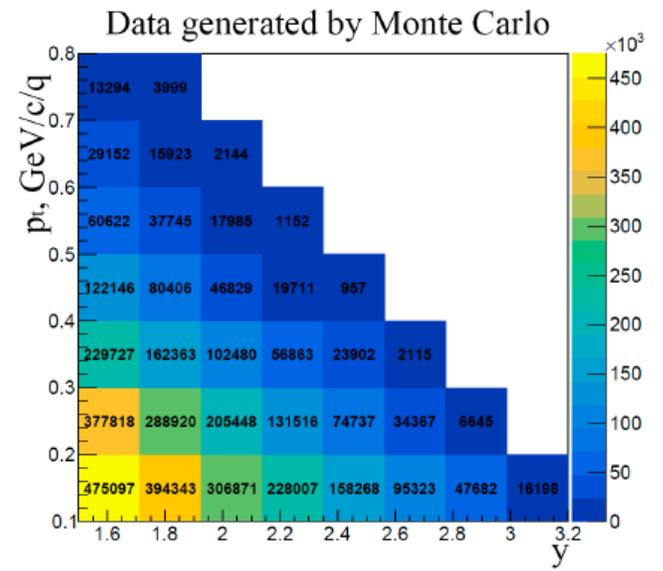
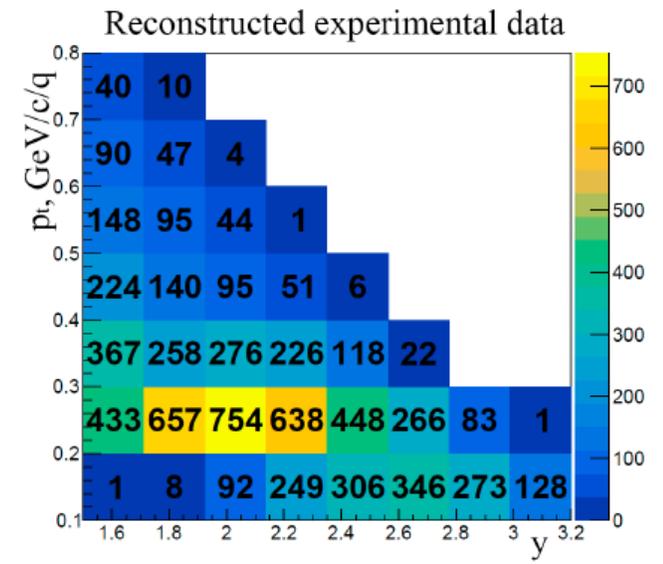
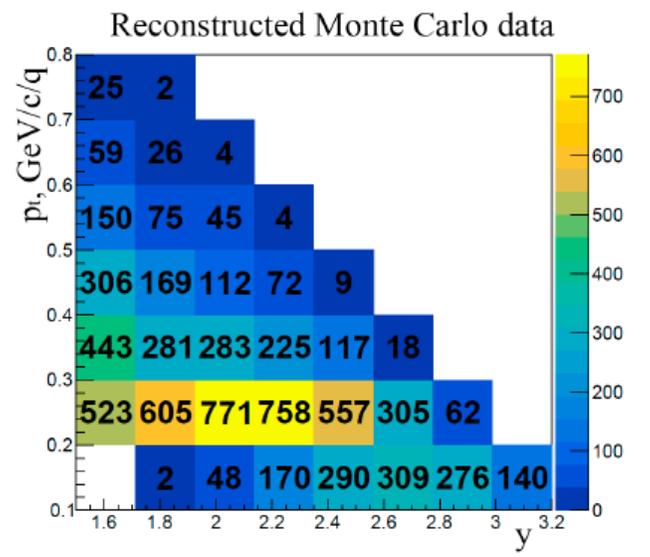
# Оценка эффективности триггера FD



$\pi$	C	Al	Cu	Sn	Pb
FD >1	0.37	0.49	0.63	0.73	0.80
FD >2	0.23	0.31	0.46	0.58	0.69
FD >3	0.15	0.19	0.33	0.44	0.58



# Распределение $\pi$ -мезонов в фазовом пространстве



# Анализ рождения ядерных фрагментов в 8 сеансе эксперимента BM@N

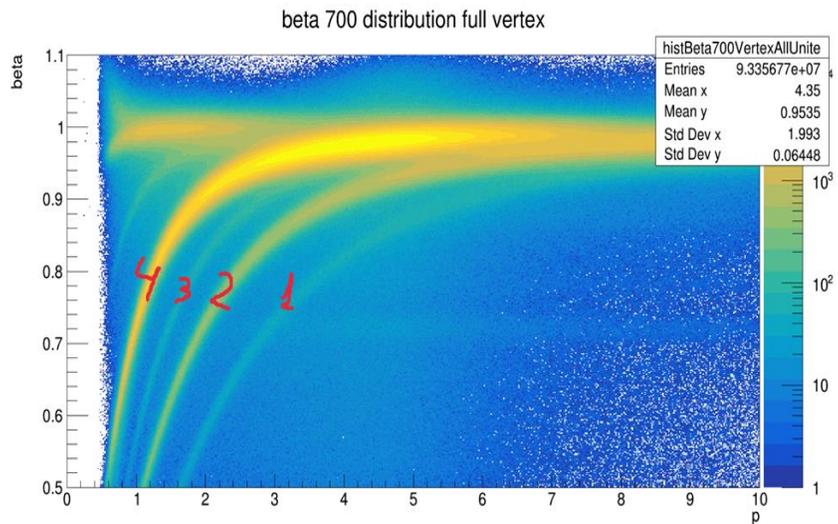
**Козлов Илья Дмитриевич**

Для экспериментальных данных, полученных во время восьмого сеанса на установке BM@N выполнить анализ рождения фрагментов гелия-3. Разработать методики реконструкции событий с рождением фрагментов гелия-3, получить оценки эффективности и других параметров.

Анализ будет включать отбор событий с рождением фрагментов гелия-3, сравнение событий с результатами моделирования и получение оценок сечений рождения фрагментов гелия-3.

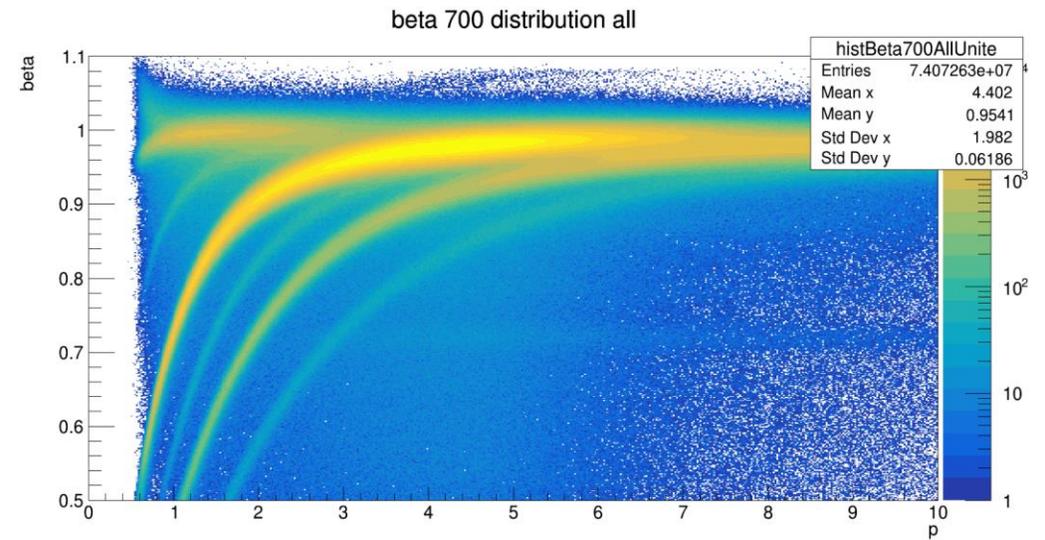
## Краткое содержание работы

- Исследовано влияние различных ограничений на дифференциацию данных по различным ядерным фрагментам: протоны, тритий, гелий-3, гелий-4:
  - геометрические ограничения;
  - по количеству хитов в разных детекторах (GEM, STS), по импульсу.
- Проанализированы треки частиц в пространстве быстроты и поперечного импульса.
- Разработаны макросы для обработки данных по трекам частиц, наложения ограничений на события и треки, построения зависимостей, а также оценки результатов.
- Обработано 30 миллионов событий.

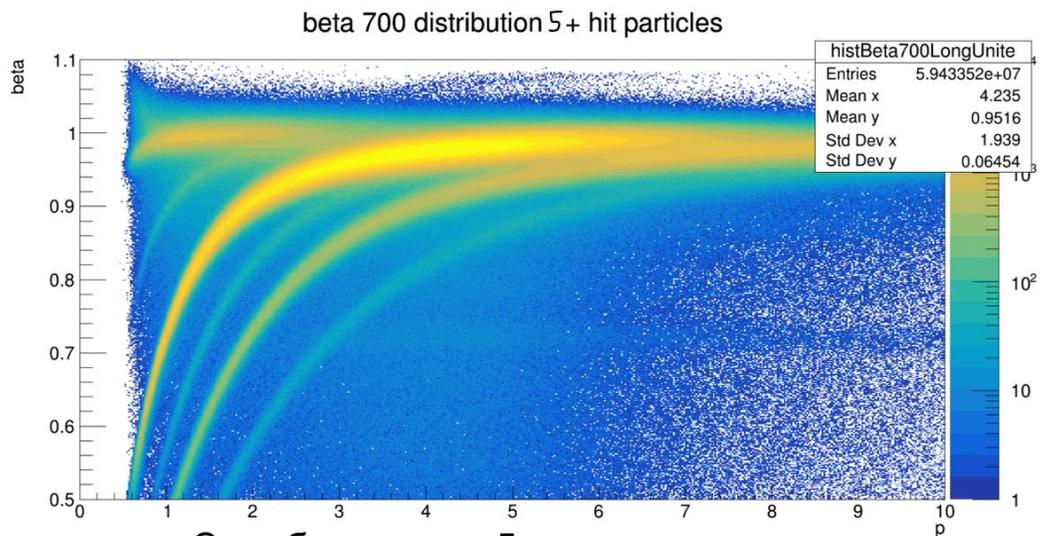


- 1 – тритий;
- 2 – гелий-4 ;
- 3 – гелий-3;
- 4 – протоны.

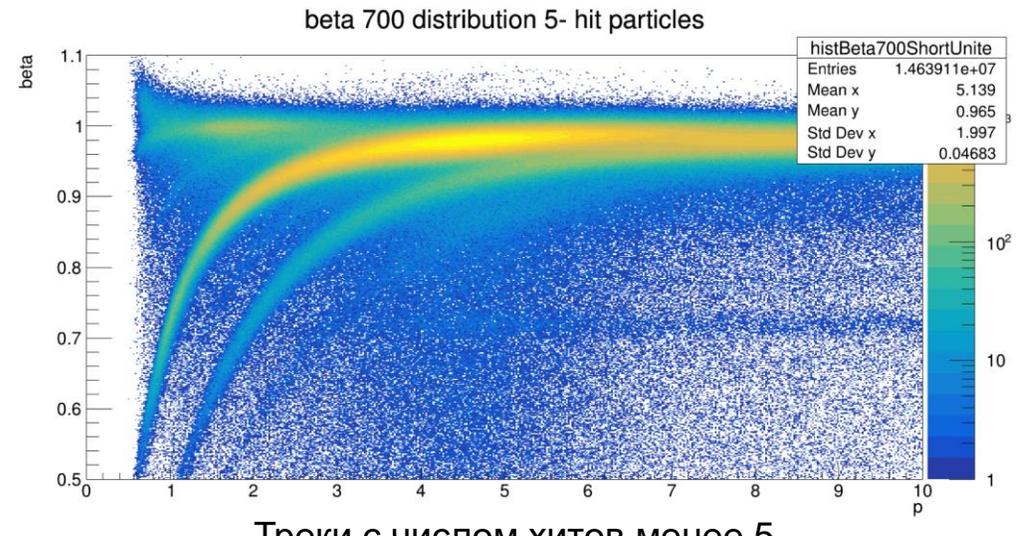
Все треки из TOF-700 без ограничений  
 $93 \times 10^6$  треков



С ограничением на первичную вершину и расстояние до нее

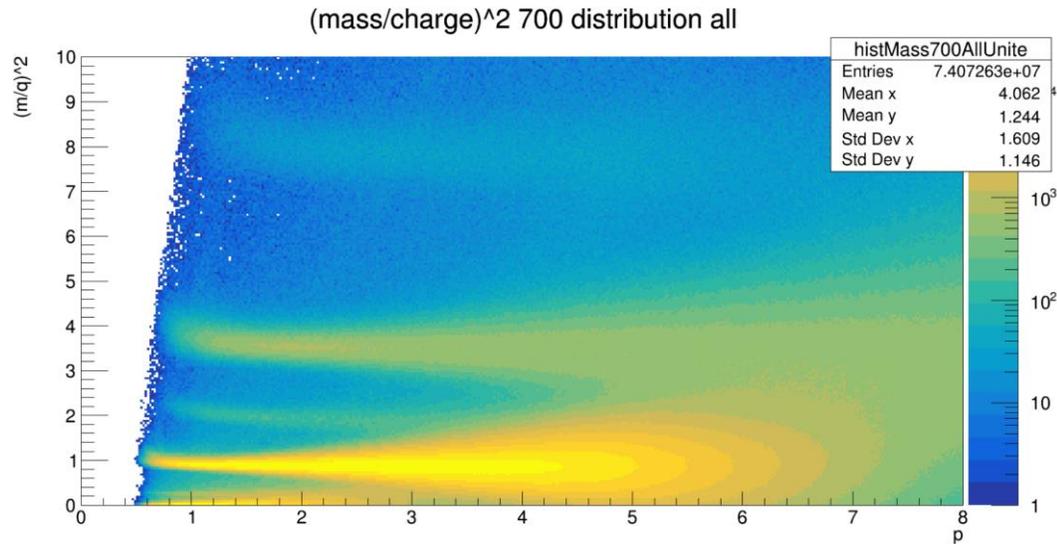


С требованием в 5 хитов минимум

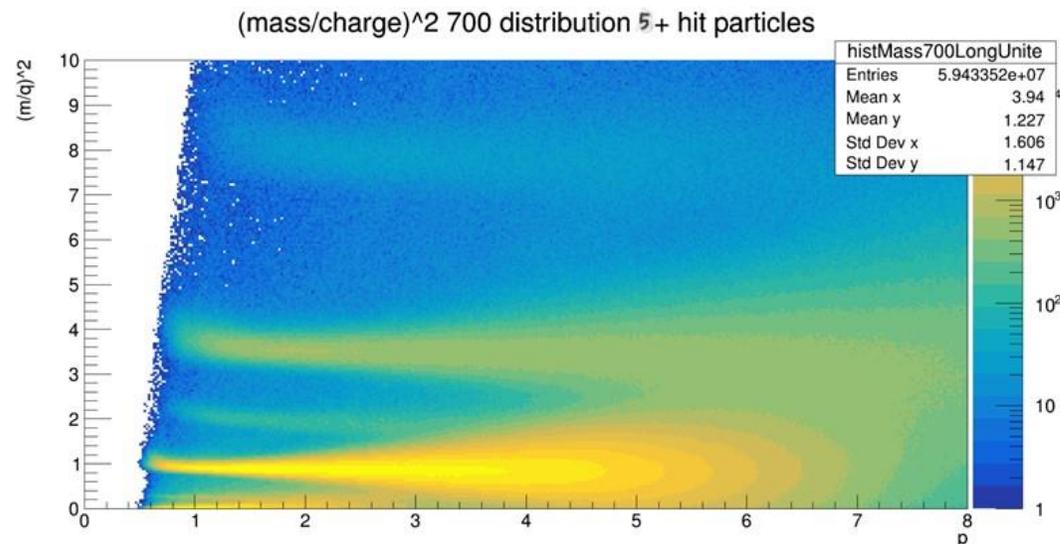


Треки с числом хитов менее 5

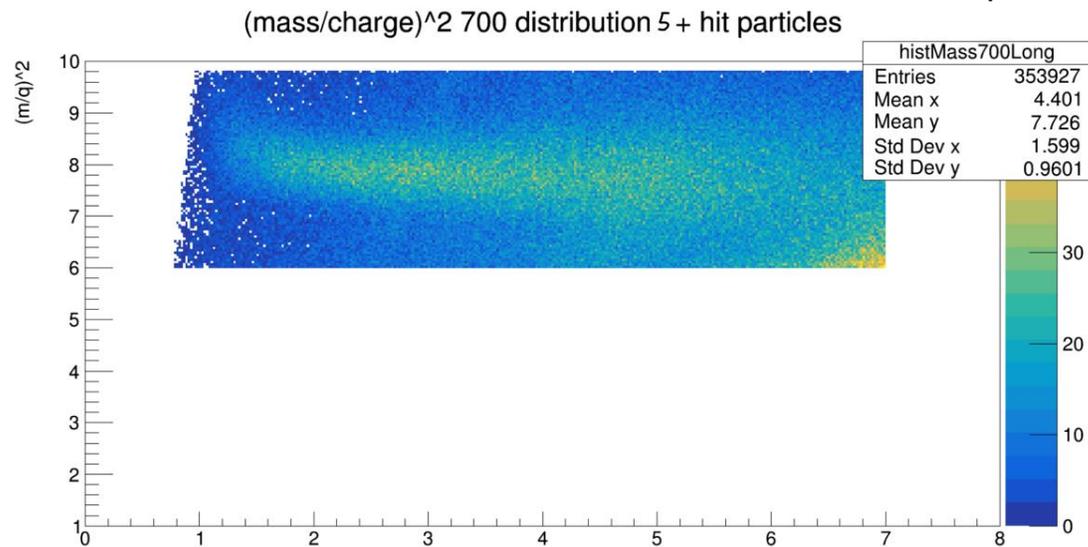
# Зависимость $(\text{масса/заряд})^2$ от импульса



Только геометрические ограничения

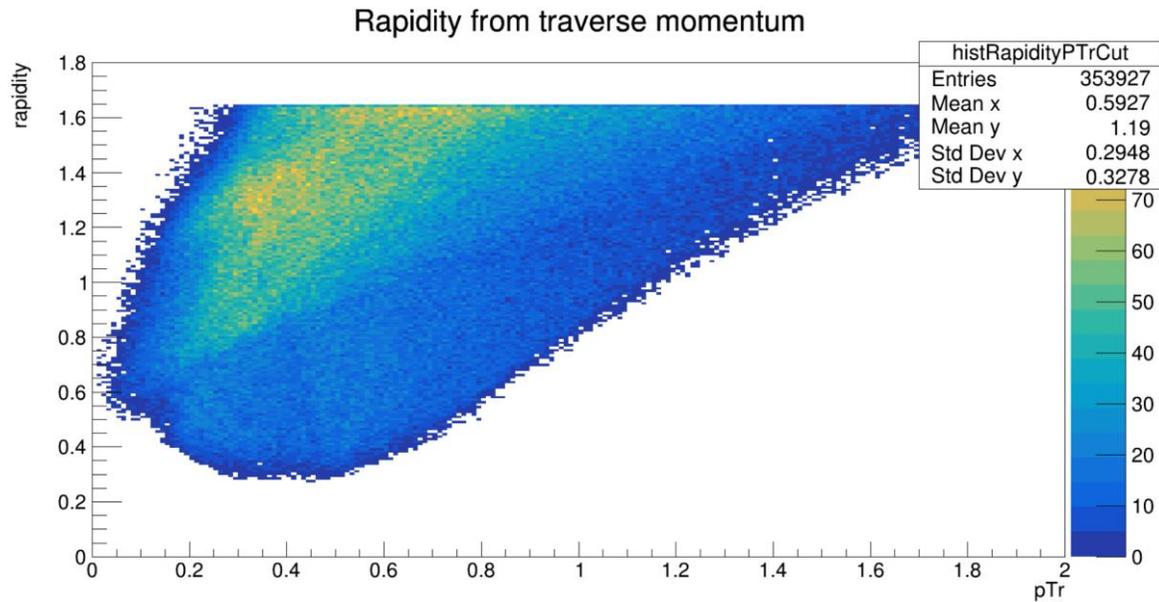


Ещё и ограничение на минимум 5 хитов

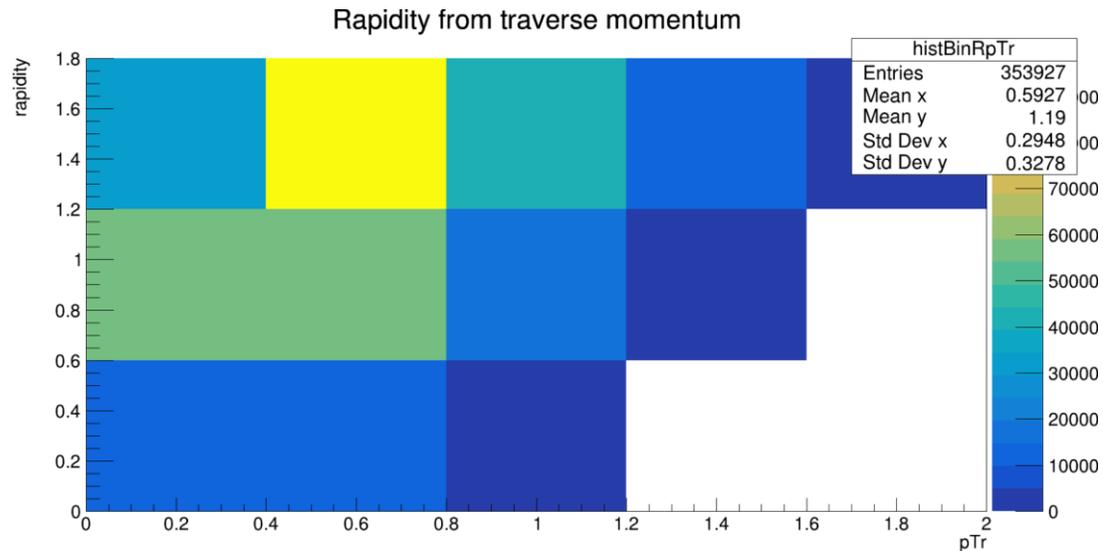


Оставлена только линия для трития

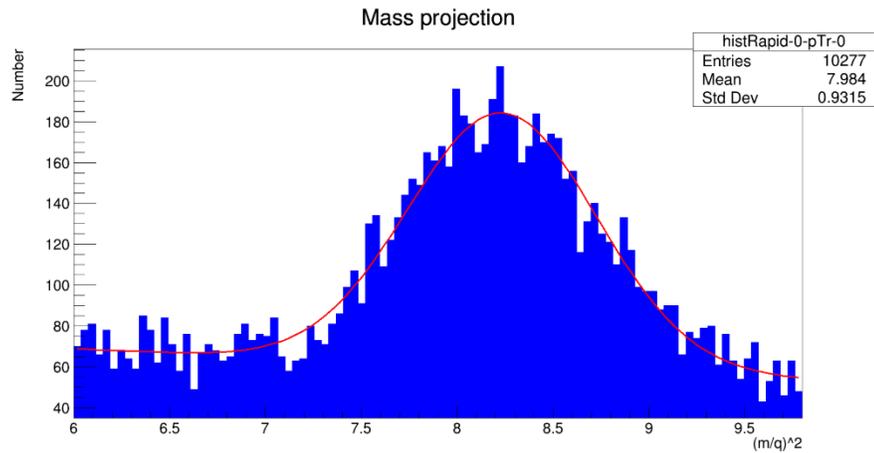
# Зависимость быстроты от поперечного импульса



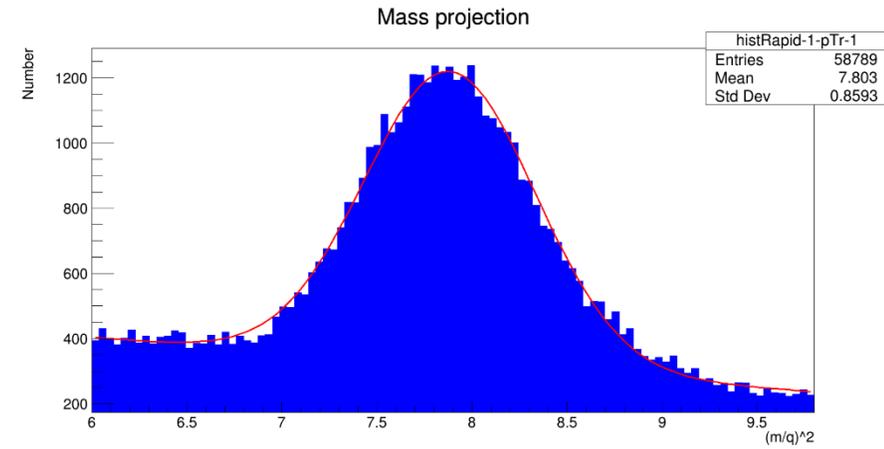
$$\text{Быстрота: } \theta = \frac{1}{2} \ln\left(\frac{E + p}{E - p}\right)$$



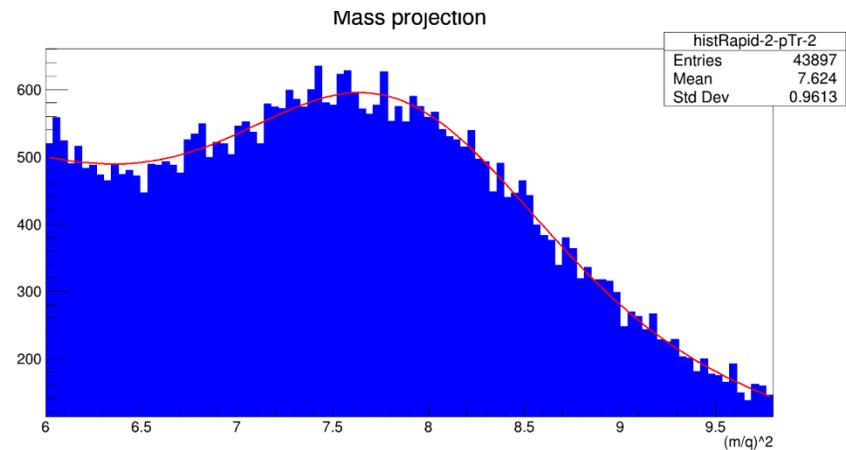
# Гистограммы зависимости быстроты от поперечного импульса с аппроксимацией по Гауссу



Распределение  $m^2/q^2$  для  $p_{tr}$  от 0,0 до 0,4 и  $\theta$  от 0,0 до 0,6



Распределение  $m^2/q^2$  для  $p_{tr}$  от 0,4 до 0,8 и  $\theta$  от 0,6 до 1,2



Распределение  $m^2/q^2$  для  $p_{tr}$  от 0,8 до 1,2 и  $\theta$  от 1,2 до 1,8

# **Развитие функционала программного обеспечения визуализации моделированных и экспериментальных данных эксперимента VM@N на основе технологий виртуальной и дополненной реальности**

**Юсупова Анастасия Юрьевна**

Дальнейшее развитие функционала прототипа программного обеспечения для визуализации установки эксперимента VM@N, а также результатов моделирования и экспериментальных данных, на основе технологий виртуальной и дополненной реальности. Подготовка выпуска первой версии программы для общего доступа.

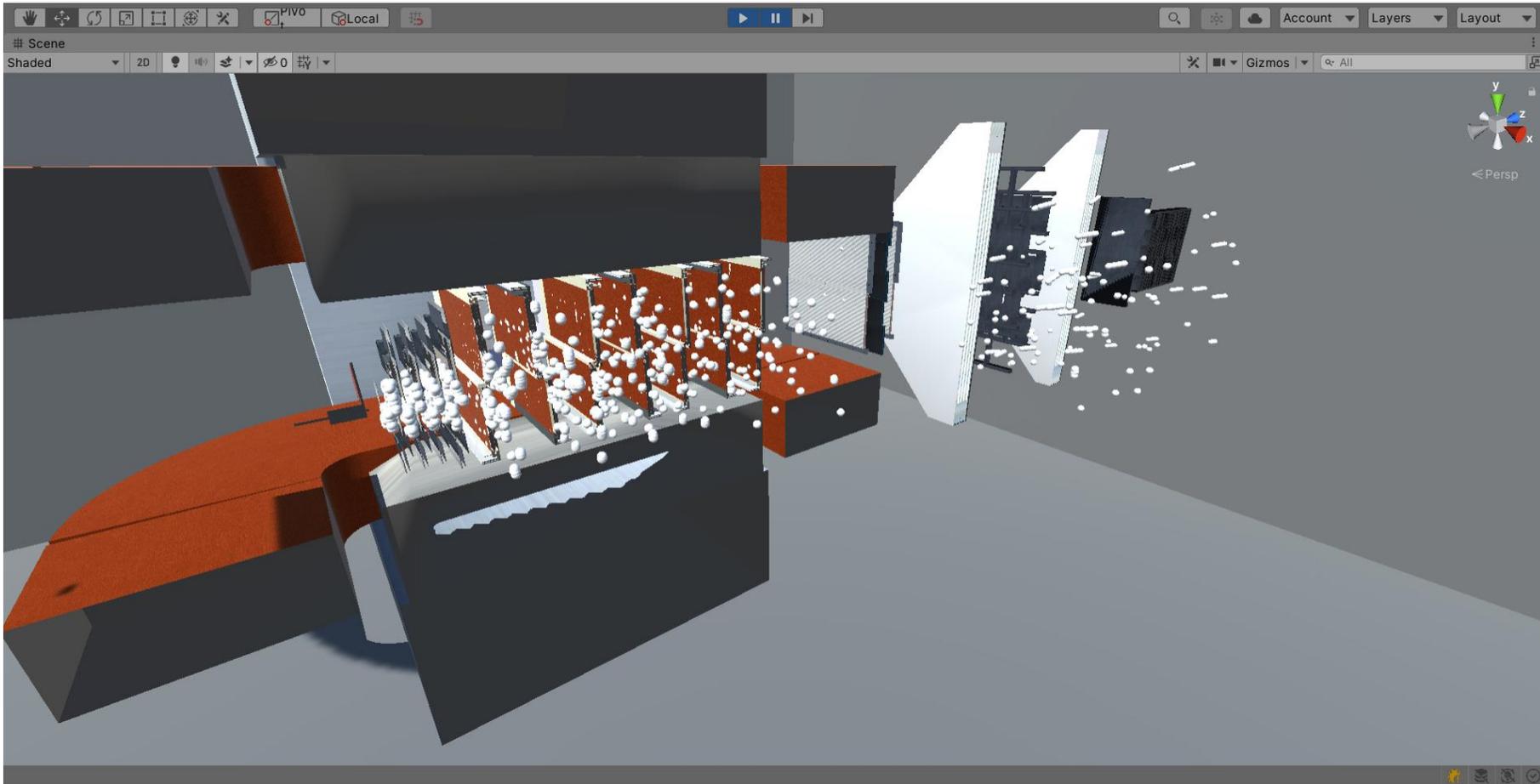
## Краткое содержание работы

- В разработанный ранее прототип программы визуализации эксперимента VM@N добавлено отображение треков и хитов с возможностью настройки изображения (размера, прозрачность).
- Написан макрос для считывания хитов, которые отображаются в виде 3D объектов-сфер. Треки отображаются отдельно для каждого детектора или для всех вместе. Данные импортируются из VmnRoot в формате JSON.
- Для лучшей визуализации геометрии проделана работа над шейдерами для отображения материалов (outline shader, transparent shader).
- Ключевым фактором текущей работы является формат импорта хитов и реконструкция треков. В настоящее время для треков используется их линейная интерполяция, так как производительности UNITY не хватает для реконструкции треков так, как это сделано в VmnRoot. Ведется работа над более реалистичным отображением треков.

Подготовлена и направлена для публикации статья.

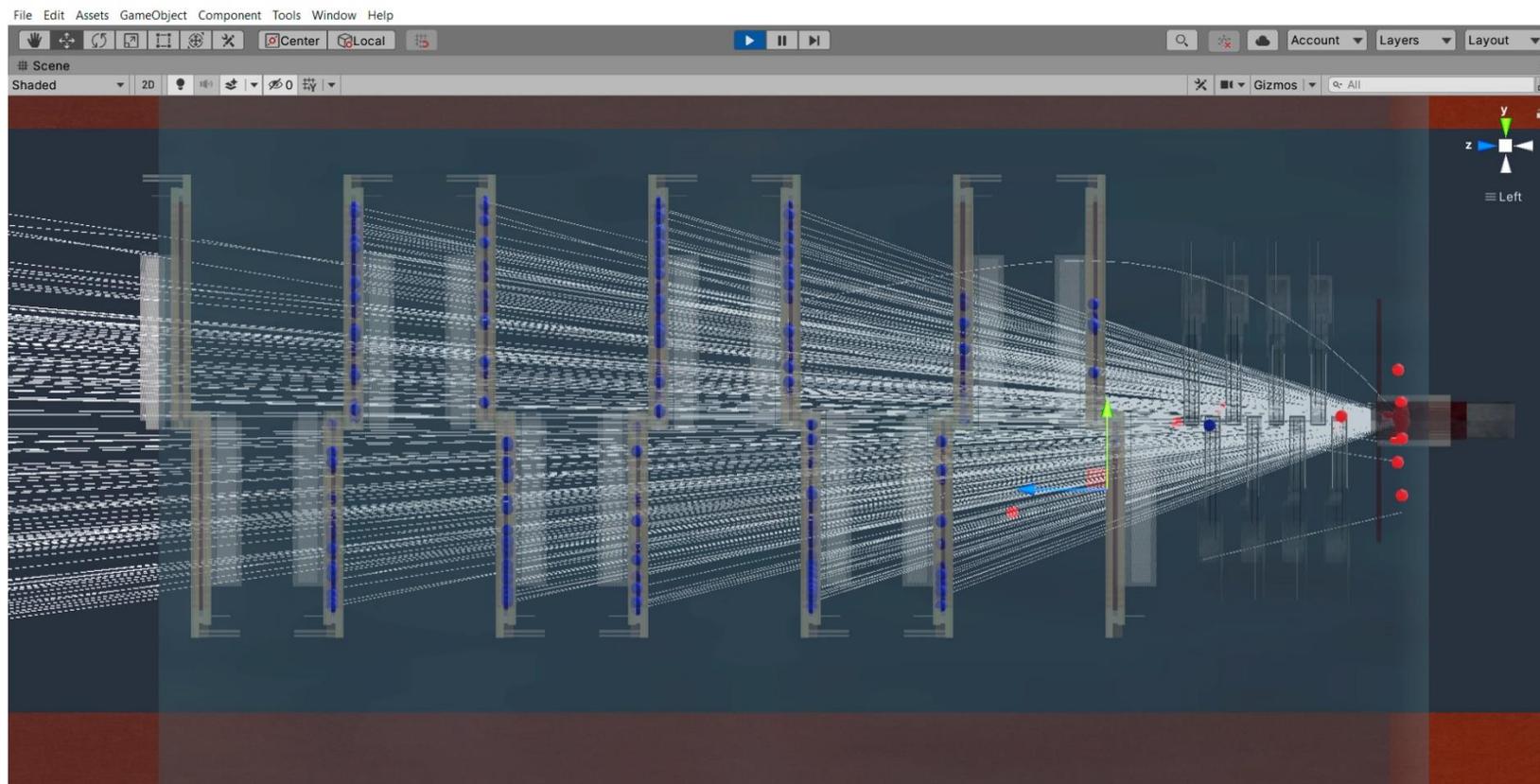


Меню настройки отображения частиц (размер, прозрачность)



Написан макрос для считывания хитов.

Хиты отображаются в виде 3D объектов-сфер,



Треки отображаются отдельно для каждого детектора (скриншот для GEM детектора) или для всех вместе. Данные импортируются из VmnRoot в формате JSON.

# **Оптимизация модулей программного пакета VmnRoot с целью повышения производительности моделирования и реконструкции событий**

**Немнюгин Сергей Андреевич**

Выполнить анализ и локализовать фрагменты программного кода пакета текущей версии VmnRoot, используемого для анализа экспериментальных данных 8-го сеанса эксперимента VM@N, которые следует оптимизировать. Определить возможность их оптимизации с применением различных инструментов повышения производительности. Особое внимание должно быть уделено оптимизации «векторного» трекинга, а также операций с картой магнитного поля, поскольку они являются одной из основных причин низкой производительности ряда модулей пакета VmnRoot.

## Краткое содержание работы

- Выполнен «hotspot» анализ текущей версии VmnRoot, выявлены участки кода, проблемные с точки зрения производительности.
- Выполнена оптимизация исходного кода метода интерполяции магнитного поля. Получено увеличение производительности на моделированных данных.
- Выполнена оценка возможного ускорения реконструкции при частичном отказе от использования векторных контейнеров.
- Выполнена адаптация одного из методов VmnRoot для гибридной архитектуры. Получены оценки эффективности гибридного кода.
- Выполнена частичная векторизация программного кода с помощью векторных интринсиков.

Доклад на 10 совещании коллаборации VM@N (май 2023, Санкт-Петербург).

# Производительность и методы оптимизации

## Тайминг двух методов реконструкции

- L1 ~0.5 сек/событие.
- Vector Finder (VF) ~2 сек/событие.

## Тестовый стенд

ASUS Vivobook, CPU Intel Core i7-11370H (4 ядра, 2x гипертрединг, AVX2), 16 Гб RAM.

## Фокус оптимизации

- BmnNewFieldMap.cxx / BmnNewFieldMap.h

## Методы оптимизации

1. Оптимизация исходного кода.
2. Оценка возможной эффективности гибридизации программного обеспечения.

# Field map

Результат анализа реконструкции с VF.

Список наиболее важных «ХОТСПОТОВ» и пример одного ИЗ НИХ.

## Elapsed Time: 236.642s

CPU Time:	220.952s
Instructions Retired:	2,847,827,400,000
Microarchitecture Usage:	99.7% of Pipeline Slots
Total Thread Count:	49
Paused Time:	0s

## Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time	% of CPU Time
BmnNewFieldMap::FieldInterpolate	libBmnField.so.0.0	26.672s	12.1%
TArrayF::At	libBmnField.so.0.0	23.547s	10.7%
TArray::BoundsOk	libBmnBase.so.0.0	19.911s	9.0%
CbmKFMath::multQtSQ	libKF.so.0.0.0	16.942s	7.7%
BmnNewFieldMap::IsInside	libBmnField.so.0.0	14.888s	6.7%
[Others]	N/A*	118.993s	53.9%

\*N/A is applied to non-summable metrics.

574			
575	Double_t BmnFieldMap::Interpolate(Double_t dx, Double_t dy, Double_t dz) {	0.2%	0.656s
576	// Interpolate in x coordinate		
577	fHb[0][0] = fHa[0][0][0] + (fHa[1][0][0] - fHa[0][0][0]) * dx;	0.5%	1.392s
578	fHb[1][0] = fHa[0][1][0] + (fHa[1][1][0] - fHa[0][1][0]) * dx;	0.4%	1.112s
579	fHb[0][1] = fHa[0][0][1] + (fHa[1][0][1] - fHa[0][0][1]) * dx;	0.4%	1.140s
580	fHb[1][1] = fHa[0][1][1] + (fHa[1][1][1] - fHa[0][1][1]) * dx;	0.4%	1.258s
581			
582	// Interpolate in y coordinate		
583	fHc[0] = fHb[0][0] + (fHb[1][0] - fHb[0][0]) * dy;	0.9%	2.672s
584	fHc[1] = fHb[0][1] + (fHb[1][1] - fHb[0][1]) * dy;	0.7%	2.228s
585			
586	// Interpolate in z coordinate		
587	return fHc[0] + (fHc[1] - fHc[0]) * dz;	1.8%	5.296s

# Оптимизация кода

```
...  
#pragma GCC target("avx2")  
#pragma GCC optimize("O3")  
...  
Double_t BmnNewFieldMap::FieldInterpolate(TArrayF* fcomp, Double_t x,  
Double_t y, Double_t z) {  
    Int_t ix = 0;  
    Int_t iy = 0;  
    Int_t iz = 0;  
    Double_t dx = 0.;  
    Double_t dy = 0.;  
    Double_t dz = 0.;  
  
    ...  
    if(x < fXmin || x >= fXmax || y < fYmin || y >= fYmax || z < fZmin || z >= fZmax)  
        {return kFALSE; }  
    {Double_t xx, yy, zz;  
    xx = (x - fXmin) / fXstep;  
    yy = (y - fYmin) / fYstep;  
    zz = (z - fZmin) / fZstep;  
    ...
```

Тест производительности.

500 моделированных событий  
(DCM-SMM).

До оптимизации: 1043 сек.

После оптимизации: 950 сек.

Выигрыш ~100 сек/500 событий.

# Оптимизация кода

```
#include <vector>
...
vector<Double_t> xIn;
xIn[0] = par->GetX();
xIn[1] = par->GetY();
xIn[2] = par->GetTx();
xIn[3] = par->GetTy();
xIn[4] = par->GetQp();
vector<Double_t> xOut(5, 0.);
vector<Double_t> F1(25, 0.);
...
```

1. Отказ от векторных контейнеров в тех частях кода, которые могут быть критичными для производительности, в пользу «обычных» статических массивов.

2. При использовании векторных контейнеров:

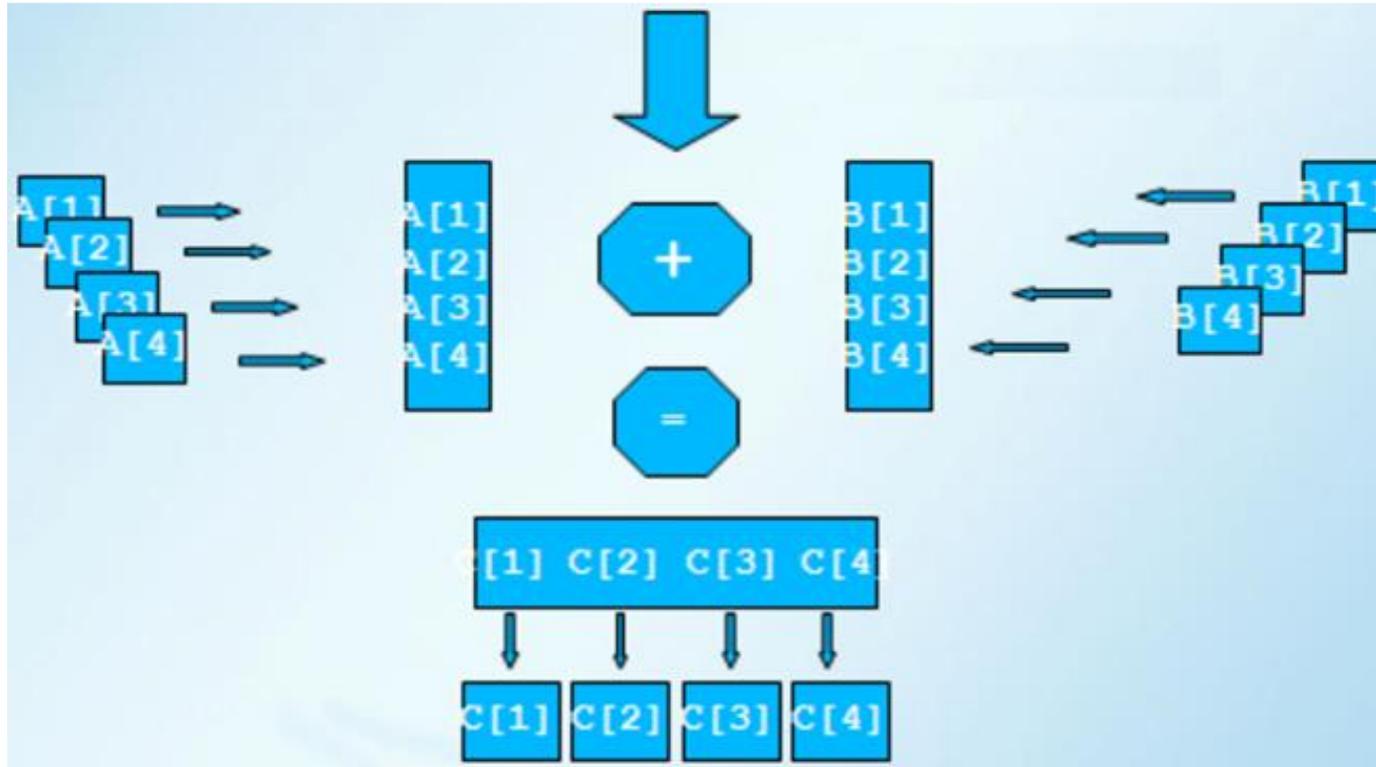
- Использовать более эффективные методы подстановки элементов (с меньшим количеством транзакций с памятью);
- *a priori* резервирование предполагаемого количества элементов.

```
vector<Double_t> prevPredX;
prevPredX.push_back(prevNode->GetPredictedParam()->GetX());
...
```

```
vector<Double_t> prevPredX;
prevPredX.reserve(5);
prevPredX.emplace_back(prevNode->GetPredictedParam()->GetX());
...
```

Оценка увеличения  
производительности  
~10 % и больше. 

# Векторизация FieldMar и программной реализации фильтра Калмана с помощью векторных интринсиков



Векторизация применяется к циклам обработки массивов.

Выполнена частичная векторизация с помощью `xmmintrin` векторных интринсиков.

```
...
#pragma GCC target("avx2")
#pragma GCC optimize("O3")
#include <x86intrin.h>
...
__m256d s;
__m256d *cInxx, *cIn_tmpxx;
cInxx = (__m256d*) cIn;
cIn_tmpxx0 = (__m256d*) cIn_tmp;
...
s = _mm256_broadcast_pd(0);
...
```

# Гибридизация кода

```
#include "BmnStripData.cuh«
...
__global__ void IsInsideKernel(Double_t* XPoints, Double_t* YPoints, int NPoints, \
Double_t x, Double_t y, bool* check_flag)
{
int tid = blockIdx.x * blockDim.x + threadIdx.x;
int j = NPoints - 1;
bool local_check_flag = false;
while(tid < NPoints)
{
int i = tid;
...
cudaMalloc(&d_check_flag, sizeof(bool));
cudaMemcpy(d_check_flag, &check_flag, sizeof(bool), cudaMemcpyHostToDevice);
int threadsPerBlock = 256;
int blocksPerGrid = (NPoints + threadsPerBlock - 1) / threadsPerBlock;
cudaEvent_t start, stop;
cudaEventCreate(&start);
cudaEventCreate(&stop);
cudaEventRecord(start);
IsInsideKernel<<<blocksPerGrid, threadsPerBlock>>>(XPoints, YPoints, \
NPoints, x, y, d_check_flag);
cudaEventRecord(stop);
cudaEventSynchronize(stop);
...

```

Время выполнения работы макроса run_sim_bmn.cxx		
	Без оптимизации, сек	С оптимизацией CUDA, сек
DCMSMM 200	4097.30	4056.14
	4071.04	4014.10
	4058.99	3988.31