

Документация по коду, используемому для определения центральности по множественности рожденных частиц с помощью модели Глаубера

Илья Сегаль, НИЯУ МИФИ

Москва, Август 2019

1 Введение

Данная документация описывает порядок действий, которые необходимо проделать для того, чтобы определить центральность по множественности рожденных в процессе столкновения заряженных частиц.

Важнейшим этапом на пути к получению результата является аппроксимация распределения множественности, полученного из данных (модели или реального эксперимента) данными полученными с помощью модели Глаубера. Соответственно для того, чтобы пройти этот этап нужно иметь:

- 1 Распределение, построенное на данных. Получение данного распределения часто особая отдельная задача, которую мы не будем рассматривать в данном тексте.
- 2 Данные, полученные с помощью модели Глаубера. Эта часть является стандартной и она изложена в параграфе 2.

После того, как имеются два root-файла: один с гистограммой из данных, а другой – с моделью Глаубера, можно произвести фитирование данных с помощью модели Глаубера с наложением на неё Отрицательного Биномиального распределения (NBD). Произвести эту операцию можно с помощью кода `CentralityFramework`, о котором пойдет речь в параграфе 3.

Основным результатом предыдущего шага будет получение модельного распределения множественности рожденных заряженных частиц. Это распределение далее используется для определения классов центральности (см. параграф 4) и определения различных не наблюдаемых параметров процесса столкновения (прицельного параметра (далее – b), числа партисипант (далее – N_{part}), числа бинарных нуклон-нуклонных столкновений (далее – N_{coll}) (см параграф 5)

2 Модель Глаубера

В рамках данного анализа используется модель Глаубера, основанная на методе Монте-Карло (Glauber MC). Подробнее о ней, а также об ее отличиях от Оптической модели Глаубера можно прочитать в статье американских ученых по определению центральности в экспериментах на коллайдере RHIC.

Модель Глаубера предназначена, прежде всего, для того, чтобы смоделировать процесс столкновения. Пошагово процесс моделирования можно описать следующим образом:

- 1 В первую очередь прицельный параметр b данного столкновения выбирается случайным образом в соответствии с линейным распределением вероятности $P(b) \sim bdb$.
- 2 Для каждого из ядер нуклоны налетающего ядра и ядра мишени (N_{proj} и N_{targ}) располагаются случайным образом в пределах сфер радиусов R_{proj} и R_{targ} . Это достигается с помощью равномерного распределения вероятностей в азимутальном (ϕ) и полярном ($\cos\Theta$) углах и радиального распределения плотности, задаваемого, как $P(r) \sim r^2\rho(r)$, где $\rho(r)$ – распределение Вуд-Саксона(см. уравнение 1):

$$\rho(r) = \rho_0 \frac{1 + w \left(\frac{r}{R}\right)^2}{1 + \exp\left(\frac{r-R}{a}\right)}, \quad (1)$$

где ρ_0 отвечает за плотность в центре ядра, R – радиус ядра, a – глубина ядра, на которой второе слагаемое в знаменателе уменьшается в e раз, w – параметр, характеризующий отклонение формы ядра от сферической

Для ядер золота обычно выбирают следующие параметры: $R = 6,5541$ фм, $a = 0,523$ фм, $w=0$, основываясь на данных PDG.

3 Решение о том, происходит ли взаимодействие или нет основано на критерии, что расстояние между нуклонами i и j в поперечном сечении удовлетворяет условию $d_{ij}^2 \leq \frac{\sigma_{inel}^{NN}}{\pi}$, где σ_{inel}^{NN} – сечение неупругого рассеяния. Его можно определить с помощью различных параметризаций, доступных в трудах различных научных групп. Или еще более простым методом: проанализировав зависимости сечений нуклон-нуклонных столкновений от энергии столкновений, доступных на сайте PDG с помощью программы "xyscan". При этом стоит помнить, что в силу изотопической симметрии $\sigma_{inel}^{pn} = \sigma_{inel}^{np}$ и $\sigma_{inel}^{pp} = \sigma_{inel}^{nn}$, а также что $\sigma_{total}^{NN} = \sigma_{inel}^{NN} + \sigma_{elastic}^{NN}$

4 После этого проводится подсчет числа «партисипант» и бинарных столкновений.

Провести данное моделирование нетрудно с помощью кода, разработанного группой эксперимента PNOBOS. Также полезно почитать документацию к их коду.

Чтобы получить выборку модельных данных с помощью данного кода нужно:

1 В одну папку поместить макрос "runglauber_v3.2.C" (или любой другой необходимой версии) и файл rootlogon.C, в котором прописать "gROOT->LoadMacro("runglauber_v3.2.C+"); для создания соответствующий библиотеки при запуске root.

2 Запустить root. И прописать в командной строке: "runAndSaveNtuple(n,"sysA "sysB" в качестве n указать величину выборки (например, 10000), в качестве "sysA" и "sysB" указать сталкивающиеся системы (например, "Au3" и "Au3" для золота с указанными выше параметрами), в качестве signn указать σ_{inel}^{NN} , полученное указанным выше методом.

Таким образом, на выходе данный код дает дерево, в котором для каждого события прописаны b , N_{part} , N_{coll} и т.д.

3 Код Centrality Framework

Часто случается, что во время эксперимента по различным причинам (плохой аксептанс, неэффективный триггер) теряется информация о некоторых событиях, чаще всего периферических, что мешает узнать центральность в соответствии с ее определением, приведенным в параграфе 4. Поэтому очень важно,

чтобы при определении классов центральности распределение событий по множественности было полным.

Для этого достаточно получить "идеальное" модельное распределение. Сделать это можно, если аппроксимировать распределение, полученное из данных модельной зависимостью в той области, где потери информации минимальны, а значит отклонений от идеальной формы распределения практически нет.

Во-первых, для аппроксимации берутся результаты моделирования процесса столкновения (модель Глаубера). По этим результатам вычисляется число предков (ancestors), т.е. нуклонов, из которых возможно рождение новых частиц. Для множественности рожденных частиц часто используют двухкомпонентную модель:

$$N_{ancestors}(f) = fN_{part} + (1 - f)N_{coll}, \quad (2)$$

где f – параметр, который может принимать значения от 0 до 1.

Далее для каждого предка в событии разыгрывается число рожденных частиц. Данный процесс является случайным и, как показывает опыт описывается отрицательным биномиальным распределением (Negative Binomial Distribution – NBD):

$$P_{\mu,k}(n) = \frac{\Gamma(n+k)}{\Gamma(n+1)\Gamma(k)} \frac{\left(\frac{\mu}{k}\right)^n}{\left(\frac{\mu}{k} + 1\right)^{n+k}}, \quad (3)$$

где $P_{\mu,k}$ – вероятность рождения n новых частиц, k и μ – параметры распределения. Средним данного распределения является параметр μ , а дисперсией $\sigma^2 = \mu \left(\frac{\mu}{k} + 1\right)$.

В результате в каждом событии число рожденных частиц суммируется по всем предкам, что позволяет узнать множественность отдельного события. Повторяя этот процесс для каждого события можно заполнить гистограмму с модельным распределением.

Код для данных действий и дальнейшего анализа лежит по ссылке github.com/IlyaSegal/NICA

Для того, чтобы отфитировать модельное распределение можно использовать код, созданный сотрудниками университета ГСИ, Дармштадт, Ильей Селюженковым и Виктором Клочковым:

- 1 Зайти в директорию, в которую вы хотите сохранить код.

- 2 Вбить в командную строку: "git clone <https://github.com/IlyaSegal/NICA>"

- 3 Вбить в командную строку: "cd centrality-master"
- 4 Вбить в командную строку: "mkdir build"
- 5 Вбить в командную строку: "cd build"
- 6 Вбить в командную строку: "source /path-to-root/thisroot.sh где path-to-root – путь до той версии root, с которой вы хотите собрать код"
- 7 Вбить в командную строку: "cmake .."
- 8 Вбить в командную строку: "make"
- 9 Далее в файле Fitter.cpp в функции "FitGlauber" на 225-ой строке указать количество итераций (на данный момент выставлено 2) и на следующей строке выставить такое же число после знака "+" для правильной нумерации итераций.
- 10 В файле main.cpp:
 - на строке 43 указать путь до файла с выборкой модели Глаубера
 - на строке 44 указать название дерева в этом файле
 - на строке 43 указать путь до файла с распределением, полученным из данных
 - на строке 44 указать название нужной гистограммы в этом файле
 - на строках 48 и 49 нужно выбрать пределы (минимальный и максимальный соответственно), в которых будет производиться фитирование. Сделать это нужно так, чтобы фитирование производилось только в той области, где распределение из данных считается идеальным, но и так, чтобы этот диапазон захватывал некоторые характерные черты распределения (хвост в области центральных событий, начало роста распределения в области периферических событий и т.д.)
 - на строке 63 можно выбрать объем данных, которые будут браться из модели Глаубера. В принципе, хорошего результата (вероятность ошибки 3,2%) можно добиться, если объем данной выборки будет в 10 раз больше того числа событий, которые попадают в диапазон фитирования. Для этого достаточно раскоммитить строчку 62 и закоммитить строчку 63

11 Далее, находясь в папке build вбить в командную строку: `"/glauber f1 f2 k1 k2 > Control.txt ult f1-f2` – диапазон, в котором будет производиться поиск параметра f , $k1-k2$ – диапазон, в котором будет производиться поиск параметров k , а `Control.txt` – это файл, в который будут заноситься результаты работы программы в текстовом виде.

4 Определение классов центральности

Центральность столкновений обычно определяется по величинам, которые называются предсказателями центральности (centrality estimators). Чаще всего в качестве предсказателей центральности выбирают величины, связанные с числом рожденных частиц (число зарегистрированных треков, число попаданий в какой-либо детектор и т.п.). Реже используют величины, связанные со «спектаторами», такие как, например, энергия зарегистрированная каким-либо калориметром, расположенным в направлении пучка. В данной работе для определения центральности используются оба предсказателя центральности.

Когда имеется неискаженное распределение событий по какому-либо предсказателю центральности, саму центральность C можно определить по следующей формуле:

$$C = \frac{\int_0^b \frac{d\sigma}{db'} db'}{\int_0^\infty \frac{d\sigma}{db'} db'} = \frac{1}{\sigma_{AA}} \int_0^b \frac{d\sigma}{db'} db', \quad (4)$$

где σ_{AA} – полное сечение или, другими словами, вся площадь распределения событий по предсказателю центральности, $\frac{d\sigma}{db'}$ – дифференциальное сечение или площадь распределения событий по предсказателю центральности, соответствующая конкретному значению центральности, b – прицельный параметр. В силу того, что прицельный параметр линейно связан с числом рожденных частиц (N) и числом «спектаторов» (N_{spec}) можно записать уравнение 4 в следующем виде:

$$C \approx \frac{1}{\sigma_{AA}} \int_{N_{thr}}^\infty \frac{d\sigma}{dN'} dN', \quad (5)$$

где N_{thr} – пороговое значение числа рожденных частиц, соответствующее искомому значению центральности.

Таким образом, можно заключить, что центральность равна процентному отношению площади части распределения событий по предсказателю централь-

ности от порогового значения до бесконечности ко всей площади распределения. При этом наиболее центральным событиям соответствует значение $C = 100\%$, а наиболее периферические $C = 0\%$.

Однако, стоит отметить, что обычно центральность не измеряют в конкретных значениях, а измеряют в классах центральности. Иначе говоря, разбивают распределение событий по предсказателям центральности на равные куски, называемые классами центральности (по 5%, 10%, 20% и т.д.), и для того, чтобы изучить зависимость какой-либо величины от центральности, ищут её значение отдельно в каждом классе центральности. Об этом речь пойдет в следующем параграфе.

5 Определение не наблюдаемых величин в различных классах центральности

Примером определения наблюдаемых величин в различных классах центральности является макрос `Pics.C`, расположенный в репозитории