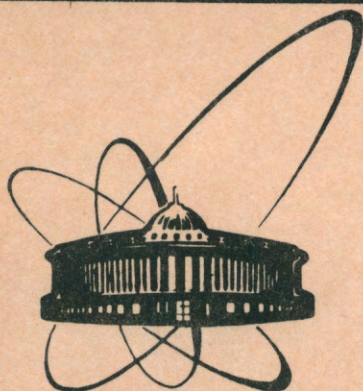


91-311



сообщения
объединенного
института
ядерных
исследований
дубна

P10-91-311

Ю.А.Будагов, Л.А.Кулюкина, Г.А.Ососков,
Л.А.Пермякова, С.Токар, Н.И.Чернов

АЛГОРИТМЫ ВОССТАНОВЛЕНИЯ КООРДИНАТ
ВЕРШИН РАСПАДА КОРОТКОЖИВУЩИХ ЧАСТИЦ
ПО ПРЯМОЛИНЕЙНЫМ ТРЕКАМ

1991

1. Введение

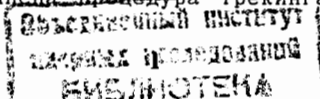
В последние годы большое внимание уделяется как теоретическому, так и экспериментальному изучению физики В-частиц, представляющей уникальную возможность либо проверить справедливость стандартной модели, либо получить альтернативные сведения о новой физике.

"В-физика" является ключевой задачей в экспериментах с фиксированной мишенью в ТЭВ-ных областях высоких энергий [1]. Для изучения основных ее проблем необходимо надежно и с высокой точностью фиксировать вершину распада В-частицы и продукты ее распада (D-мезоны). Это налагает достаточно жесткие требования как на вершинный детектор спектрометра, так и на соответствующую математическую обработку экспериментальной информации.

В данной работе предложен новый алгоритм определения координат вторичных вершин по данным измерений прямолинейных треков в микростриповом детекторе (MCD), являющемся основным элементом вершинной части спектрометра.

В работе [2] описана модель вершинного детектора в эксперименте по анализу адронных взаимодействий при энергиях порядка 3 ТэВ с образованием короткоживущих В- и D-мезонов. Характерными особенностями таких взаимодействий являются большая множественность вторичных частиц, малые углы разлета и малая длина пробега мезонов до распада ($\approx 100 \mu\text{m}$). Еще одной, не перечисленной выше особенностью, надо считать крайне низкое отношение сигнал/фон ($\approx 10^{-5}$) из-за малого сечения образования мезонов.

Возникающая при этом проблема быстрой, on-line фильтрации событий и выделение среди них полезных, многовершинных, решалась в работе [2]. В данной работе мы предполагаем, что уже решена эта проблема, а также выполнена процедура трекинга, т.е. мы располагаем



банком отфильтрованных многовершинных событий в виде наборов параметров треков по отдельным проекциям вместе с погрешностями этих параметров, определяемыми типом MCD, используемым в эксперименте.

Нашей задачей будет восстановление координат всех вершин в каждом событии и определение принадлежности треков к восстановленным вершинам. Предполагается, что задача будет решаться в off-line режиме, поэтому оптимизация алгоритмов будет вестись не по быстродействию, а по точности восстановления.

В работе обсуждаются различные подходы к проблеме восстановления вершин. Для этого мы привлекаем идеи теории распознавания образов, кластерного анализа, математической статистики и теории планирования эксперимента. Комбинации этих идей позволили создать алгоритмы, ориентированные на различные модели, и найти новые решения. Эта часть работы носит теоретический и методологический характер. Кроме того, мы описываем конкретный алгоритм восстановления вершин, который показал высокую эффективность и точность на модельных данных, соответствующих возможной структуре вершинного детектора для ТЭВ-ных энергий (см.[1]). Приводятся результаты численного эксперимента.

2. Общая математическая постановка задачи

Имеется K вершин $\bar{V}_1, \dots, \bar{V}_K$ с координатами $\bar{V}_j = (x_j, y_j, z_j)$, $1 \leq j \leq K$, (z -направление пучкового трека). Из этих вершин исходят N прямолинейных треков (каждый трек - из своей вершины). Треки измеряются в координатных детекторах, расположенных за вершинами в двух взаимно-перпендикулярных плоскостях (XZ, YZ). Каждый трек дает два набора измеренных точек: $(\bar{x}_i, \bar{z}_i) = \{(x_i^{(1)}, z_i^{(1)}) \dots (x_i^{(r)}, z_i^{(r)})\}$, ($i = \overline{1, N}$), на XZ -проекции и аналогичный набор (\bar{y}_i, \bar{z}_i) на YZ -проекции (здесь i - номер трека, r - число измеренных точек, определяемое числом установленных детекторов). Измеренные точки допускают некоторый случайный разброс со средним σ вокруг линии трека. Соответствие треков на XZ - и YZ -проекциях нам неизвестно и мы не ставим своей целью его определить. Наше предположение об идеальном трекинге заключается лишь в том, что измеренные точки "рассортированы" по трекам без ошибок и все шумы отсеяны. Координаты вершин и их число неизвестны и подлежат восстановлению. Геометрически мы имеем приблизительно следующую картину (рис. 1) - в переплетении прямолинейных треков надо выделить точки их схождения, в которых треки как бы собираются в пучки.

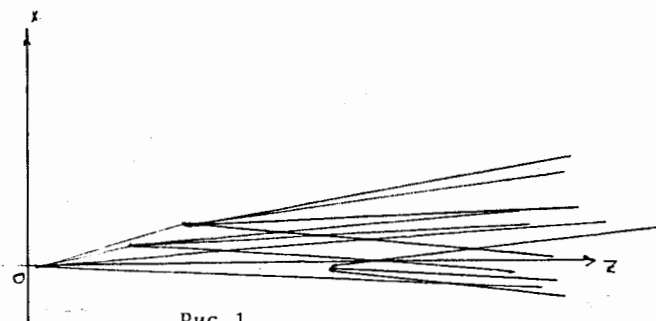


Рис. 1

Восстановление вершин в двух плоскостях (XZ и YZ) происходит независимо, и единственное, что их должно связывать, - это общие z -координаты вершин. Поэтому мы рассматриваем в основном "плоскую" задачу восстановления вершин и при необходимости "выходим в пространство".

Нашей целью будет нахождение минимального набора вершин, в которых "собираются в пучки" все треки (или максимальное их число, если число вершин ограничено). При этом вовсе не требуется, чтобы каждая вершина "притягивала" к себе больше треков. Более того, если мы будем располагать вершины в местах максимального скопления треков, как иногда делается [3], то легко допустить ошибку - рис. 2.

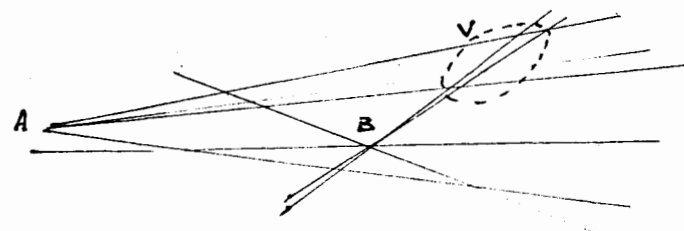


Рис. 2

На этом рисунке две вершины A и B испускают по четыре трека. Но если бы мы поставили одну "вершину" в область V , где проходит пять (!) треков, то для оставшихся трех нам потребовалось бы еще две вершины - всего получится три "вершины" вместо двух A и B .

3. Базовый алгоритм

В основу наших исследований легли идеи метода максимального правдоподобия (см., например, [4]) и теории планирования эксперимента [5], которые позволили сформулировать и решить задачу восстановления вершин в самом общем виде. Подход [5] был предложен доктором Л.Пронзато¹⁾.

Для каждого трека с измеренными точками (\bar{x}_i, \bar{z}_i) , $i=\overline{1, N}$ и вершиной \bar{v}_j ($j=\overline{1, K}$) введем величину "правдоподобия данного трека по отношению к данной вершине", основанную на стандартном предположении о нормальности отклонений треков от вершины:

$$f(\bar{x}_i, \bar{z}_i, \bar{v}_j) = \frac{1}{\sqrt{2\pi\sigma_1^2(\bar{v}_j)}} \exp(-d_i^2(\bar{v}_j)/2\sigma_1^2(\bar{v}_j)), \quad (1)$$

где $d_i(\bar{v}_j)$ - расстояние от линии i -го трека, проведенной по методу наименьших квадратов через точки (\bar{x}_i, \bar{z}_i) до вершины \bar{v}_j , а $\sigma_1^2(\bar{v}_j)$ - дисперсия этого расстояния (все формулы приведены в [2]). В целях повышения быстродействия могут использоваться и другие виды функций (1).

Пусть известна область D , в которой могут располагаться вершины, и на ней задано распределение вероятностей $\pi(\bar{v})$. Введем величины

$$L_i(\pi) = \int_D f(\bar{x}_i, \bar{z}_i, \bar{v}) d\pi(\bar{v}), \quad (i = \overline{1, N}), \quad (2)$$

которые можно рассматривать как правдоподобие i -го трека по отношению к распределению π . Из независимости измерений на различных треках следует, что

$$L(\pi) = \prod_{i=1}^N L_i(\pi) \quad (3)$$

является величиной правдоподобия всего набора треков по отношению к распределению π . Для того, чтобы связать проекции XZ и XY, надо перемножить величины $L(\pi)$, соответствующие обеим проекциям.

Поставим задачей найти распределение $\pi(v)$, при котором $L(\pi)$ достигает максимума. Это типичная задача для теории планирования эксперимента [6,7]. Для ее решения разработана хорошая теоретическая база и ряд численных алгоритмов.

¹⁾ Во время его пребывания в Дубне в 1989 г.

Если область D выпукла, то множество Π_D всех вероятностных распределений на D выпукло (в симплексе всех вероятностных распределений в пространстве), а функция $L(\pi)$ вогнута на Π_D . Поэтому максимум $L(\pi)$ достигается на границе множества Π_D , в угловых точках, соответствующих дискретным распределениям.

Более того, имеет место следующая теорема [8,9], которую мы формулируем здесь в нашей терминологии:

ТЕОРЕМА 1. Максимум функции $L(\pi)$ достигается на дискретном распределении, сосредоточенном в K точках $(\bar{v}_1, \dots, \bar{v}_K)$ с вероятностями (p_1, \dots, p_K) , причем K не превышает числа треков: $K \leq N$.

Теорема придает найденному распределению $\pi(\bar{v})$ следующий смысл: точки $\bar{v}_1, \dots, \bar{v}_K$ - это искомые вершины, а величины p_1, \dots, p_K - оценки числа треков, выходящих из этих вершин. В нашей задаче ограничения $K \leq N$ может оказаться явно недостаточно, так как реально имеется 3-5 вершин, а число треков N доходит до нескольких десятков. Если число найденных вершин K окажется слишком большим, то можно применить процедуру кластеризации (слияния нескольких близлежащих вершин в одну с суммированием соответствующих вероятностей).

В теории планирования эксперимента предложен итеративный алгоритм поиска распределения $\pi(v)$, максимизирующего функцию $L(\pi)$ [8]. Поиск ведется только по дискретным распределениям, сосредоточенным в конечном числе точек. Мы изложим этот алгоритм применительно к нашей задаче.

Шаг 1. Выбираем начальное распределение, например, сосредоточенное в нескольких случайных точках.

Шаг 2. На n -й итерации к текущему распределению $\pi^{(n)}$ добавляем одну новую точку $\bar{v}_0 \in D$. Если $\pi_{\bar{v}_0}$ - означает распределение, сосредоточенное в точке \bar{v}_0 , то ищем новое распределение $\pi^{(n+1)}$ в виде $(1-\alpha)\pi^{(n)} + \alpha\pi_{\bar{v}_0}$ для некоторого $\alpha \in (0, 1)$. Точка \bar{v}_0 определяется как точка в D , в которой функция

$$d_n(\bar{v}, \pi) = \sum \frac{f(\bar{x}_i, \bar{z}_i, \bar{v})}{L_i(\pi^{(n)})} \quad (4)$$

достигает максимума. При этом если максимум оказался $\leq N$, то алгоритм заканчивает работу, так как в этом случае $\pi^{(n)}$ - ис-

комое распределение из теоремы 1. В противном случае нам достаточно найти такое \bar{v}_0 , что $d_n(\bar{v}_0, \pi) > N$. Параметр α определяется, например, из формулы

$$\alpha = \frac{1}{N} \frac{d_n(\bar{v}_0, \pi) - N}{d_n(\bar{v}_0, \pi) - 1}$$

(ряд других способов вычисления α приведен в [9,10,11]).

Шаг 3. Чтобы предотвратить чрезмерный рост числа точек, в которых сосредоточено распределение $\pi^{(n)}$, применяется принудительное его сокращение. Например, рассчитывается величина (4) для каждой из имеющихся точек $\bar{v} \in \text{supp}(\pi^{(n)})$ и точка с наименьшим значением d_n отбрасывается (с пропорциональным пересчетом вероятностей p_1, \dots, p_k).

После этого возвращаемся к шагу 2, так как там выполняется проверка на окончание алгоритма.

Если отказаться от шага 3, то число точек-носителей распределения $\pi^{(n)}$ растет до бесконечности, но в силу теоремы 1 оно в основном "садится" на $K \leq N$ точек, а остальные имеют суммарно малую вероятность. Это обстоятельство неудобно как для нашей задачи, так и с точки зрения вычислительной процедуры. Если же использовать шаг 3, то можно с самого начала ограничиться малым числом вершин. Правда, в этом случае сходимость алгоритма к точке максимума функции $L(\pi)$ не гарантирована (см. один пример ниже, в п.4).

В работе [5] приведен иллюстрирующий пример, выполненный Л.Пронзато, в котором описанный алгоритм позволил восстановить три вершины по 11 трекам и оценить число треков для каждой вершины.

4. Модификация базового алгоритма

Описанный в п.3 алгоритм имеет под собой серьезную теоретическую базу. Однако у него есть три недостатка.

Первый состоит в том, что целевая функция (3) не совсем точно отвечает нашим целям. Как было указано в п.2, мы ищем минимальный набор вершин, для которого каждый трек проходил бы рядом с какой-то вершиной, и если это так, то не важно, насколько близок он к другим вершинам. В формулах (2-3) правдоподобие трека все же возрастает, если он проходит вблизи сразу нескольких вершин. Это может приводить к ошибкам, проиллюстрированным на рис.2.

Естественный выход здесь - введение величины правдоподобия с помощью функции

$$L_1(\pi) = \max_{\bar{v} \in \text{supp} \pi} f(\bar{x}_1, \bar{z}_1, \bar{v}), \quad (i = \overline{1, N}), \quad (5)$$

где $\text{supp} \pi$ - носитель распределения π (для дискретных распределений - это набор точек, в которых оно сосредоточено).

Тогда целевая функция $L(\pi)$ записывается в виде

$$L(\pi) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma_1^2(\bar{v}(i))}} \exp\left[-\sum_{i=1}^N \frac{d_1^2(\bar{v}(i))}{2\sigma_1^2(\bar{v}(i))}\right], \quad (6)$$

где $\bar{v}(i)$ означает точку из $\text{supp} \pi$, на которой достигается максимум в (5). При распаде короткоживущих мезонов все вершины сосредоточены в малой области D (вокруг мишени) и потому $\sigma_1^2(\bar{v})$ почти постоянно для всех $\bar{v} \in D$ (см. подробнее в [2]). Учитывая это, максимум $L(\pi)$ в (6) можно искать через минимум функции

$$T(\pi) = \sum_{i=1}^N d_1^2(\bar{v}(i)) / \sigma_1^2(0), \quad (7)$$

где $0 \in D$ - центр мишени (см. [2]), а $\bar{v}(i)$ в этой формуле означает просто ближайшую к i -му треку точку из $\text{supp} \pi$. Выражение (7) выгодно отличается от (1-3) с точки зрения реализации на ЭВМ. Результаты же в большинстве случаев почти не отличаются, более того, в редких случаях (см. выше) замена (3) на (7) позволяет избежать грубых ошибок.

Отметим, что в (7) отсутствуют вероятности p_1, \dots, p_k . Это также упрощает вычисления, хотя и лишает нас дополнительной оценки числа треков $N_j \approx N p_j$. Впрочем, выражение (5) закрепляет каждый трек за некоторой вершиной, устанавливая таким образом кластеризацию и давая необходимую оценку числа треков у каждой вершины.

Второй недостаток базового алгоритма состоит в сложности поиска точки максимума функции (4) в области D . На этот счет теория не дает никаких общих рекомендаций. Однако в нашей задаче разумно искать новую точку \bar{v}_0 лишь в окрестностях линий треков (например, ширины $\pm 3\sigma_1(0)$) и, более того, в окрестностях только точек попарного пересечения треков. Выбрав таким образом координаты X и Z (или Y и Z) точки \bar{v}_0 на одной проекции, мы находим ее третью координату в окрестности одного из треков на другой проекции. При этом, естественно, предпочтение отдается трекам, не имеющим "своей" вершины (то есть для которых величина $L_1(\pi)$ из (5) еще очень мала).

Для ускорения поиска точки \bar{v}_0 , все точки попарных пересечений треков в области D находятся заранее и запоминаются. Если их оказывается слишком много, то можно применить кластеризацию - заменить группу близлежащих точек одной.

Наконец, третий недостаток базового алгоритма состоит в том, что при жестком ограничении на число точек в $\text{supp } \pi$ (когда оно намного меньше N) его сходимость к максимуму $L(\pi)$ не гарантирована. Например, если в ситуации на рис. 3 ищется $K = 2$ вершин

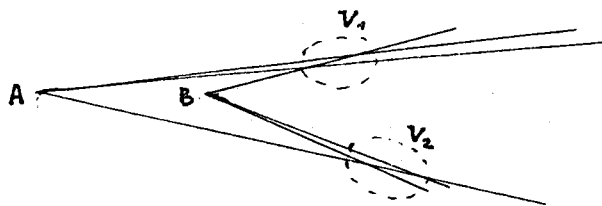


Рис. 3

и начальные точки \bar{v}_1 и \bar{v}_2 попали в области V_1 и V_2 , то процедура не сможет выйти из этого положения, так как даже при попадании новой точки \bar{v}_0 в A или B именно она будет отброшена на шаге 3, так как при отбрасывании одной из точек \bar{v}_1 и \bar{v}_2 появляются треки, очень далекие от двух оставшихся вершин. Чтобы выйти из этого "тупика", достаточно добавить в наш алгоритм дополнительный шаг - случайный разброс K точек v'_1, \dots, v'_k в области D (с учетом рекомендаций, данных выше), и если новый набор точек дал меньшее значение функции (7), то заменяем им старый. Этот шаг дает гарантию сходимости к минимуму функции (7), правда, по вероятности.

5. Дополнения к базовому алгоритму

Описанные в п.4 модификации базового алгоритма существенно улучшают и ускоряют его работу. Кроме того, мы еще более усилили его эффективность, внося ряд дополнений, основанных на современных методах математической статистики и кластерного анализа.

Предположим, что какой-то трек далеко отстоит от всех точек $\bar{v}_1, \dots, \bar{v}_k \in \text{supp } \pi$. Тогда вклад этого трека в целевую функцию (7), равный $d_1^2(\bar{v}(i)) / \sigma_1^2(0)$, очень велик, что может иметь нежелательные эффекты (например, два трека на расстоянии $\approx 10 \sigma$

от всех вершин оказываются более предпочтительными, чем один трек на расстоянии $\approx 100 \sigma$ от всех вершин, хотя все три трека одинаково "потеряны" с точки зрения конечного результата). Для подавления этих эффектов используются так называемые "робастные" варианты функции (7):

$$\hat{T}(\pi) = \sum_{i=1}^N \hat{f}(d_i(\bar{v}(i)) / \sigma_i(0)), \quad (8)$$

где $y = \hat{f}(x)$ растет не столь быстро, как $y = x^2$. Термин "робастный" означает: устойчивый к появлению больших отклонений. Мы используем "урезанную" функцию

$$\hat{f}(x) = \begin{cases} x^2 & \text{при } |x| \leq c \\ c^2 & \text{при } |x| > c. \end{cases} \quad (9)$$

Второе дополнение заключается в уточнении положений вершин по методу наименьших квадратов. Согласно (5-7), все треки разбиваются на кластеры, соответствующие вершинам. Кроме того, робастный вариант (8-9) определяет еще один кластер "потерянных" треков, которые отстоят от всех вершин на расстояния $> c$. После этого внутри каждого кластера (кроме последнего, состоящего из "потерянных" треков) можно уточнить положение вершины с помощью метода наименьших квадратов (подробные формулы приведены в [2]). При этом функция (7) (или (8)) наверняка уменьшится. Найденные положения вершин несколько отличаются от старых, и, применив к ним формулы (5-9), мы можем получить новое разбиение треков на кластеры и, соответственно, вновь уточнить положения вершин и т.д. Эту процедуру, названную нами методом динамической кластеризации, следует проводить до тех пор, пока кластеры не стабилизируются. Ее преимущество состоит в монотонном уменьшении функции (7) (или (8)). Численные расчеты показывают, что в среднем происходит 2-3 преобразования кластеров, но часто оно доходит до 7-10. Процедуру динамической кластеризации можно применять не только на последнем шаге работы алгоритма, но и на каждой его итерации.

6. Рабочая версия алгоритма

Суммируя все сказанное в пп.3-5, мы разработали одну версию

алгоритма, результаты применения которой приводятся ниже в п.7.

Шаг 1. Так же, как и в п.3, $\pi(0)$ выбираем из множества точек пересечения прямых, определяемых параметрами треков.

Шаг 2. Случайный переход к шагу 3 или шагу 4 (с равными вероятностями = 1/2).

Шаг 3. Случайный выбор K вершин на одной из случайно выбранных проекций (с учетом рекомендаций из п.4), к которым применяется метод динамической кластеризации до сходимости. Если в результате значение функции (8) уменьшится, то заменяем этим набором вершин старый. Переходим к шагу 2.

Шаг 4. Случайный выбор одной новой вершины \bar{v}_0 на одной из случайно выбранных проекций с добавлением ее к старому набору. Затем на обеих проекциях исключаем одну из старых вершин (ту, при которой значение функции (8) для оставшихся K вершин минимально), и к оставшимся вершинам применяем метод динамической кластеризации до сходимости. Переходим к шагу 2.

Этот итерационный процесс продолжается до тех пор, пока число "холостых" оборотов (не приведших к замене старого набора вершин на новый, т.е. уменьшению (8)) не превысит заданного порога. После этого - остановка.

7. Описание численного эксперимента

Для проверки работы нашего алгоритма был использован банк из 500 3-ТЭВ-ных событий, смоделированных на ЭВМ VAX-8350.

В моделировании событий можно выделить два основных момента:

а) моделирование первичного взаимодействия (ПВ), которое является предметом изучения; б) моделирование отклика детекторов установки на это событие.

Сначала был создан банк первичных событий при помощи программы RUTH, основанный на пакете Pythia [12], и затем на основе этого банка была осуществлена собственная трассировка частиц в микростриповом детекторе (MCD). Второй этап был реализован программой VDET, основанной на пакете Geant 3.18 [13]. Результатом работы программы VDET является банк модельных событий, имитирующих 3-ТЭВ-ные события через MCD.

Для того чтобы изучить эффективность восстановления В-вершины, мы смоделировали два банка событий: банк "фоновых" событий и банк "В-событий". Первый представляет собой набор событий, возникающих естест-

венным образом при взаимодействии 3-ТЭВ-ной частицы. Второй банк - набор событий, отобранных по наличию в них В-частиц.

Рассмотрим подробнее вышеуказанные этапы моделирования.

а) В рамках генератора Pythia 5.3 адроны, вступающие во взаимодействие, - это пучки протонов, а само взаимодействие делится на две фазы: "жесткую" и "мягкую". Первая включает генерирование жесткого процесса рассеяния с использованием матричных элементов, структурных функций ($2 \rightarrow 2$ процессы: $2 \rightarrow 2$, $2 \rightarrow 1$), радиационных поправок (в начальном и конечном состояниях) и реализацией партонных струй. В рамках мягкой фазы осуществляется адронизация партонных струй на основе известного струнного механизма [14].

Удобством генератора Pythia является то, что он позволяет "отбрасывать" интересующие нас subprocesses в каждом событии (напр., можно обеспечить возникновение кварков).

С помощью генератора Pythia было смоделировано взаимодействие протон-протон при 3 ТЭВ и создано два банка ПВ соответственно для фоновых событий и событий с В-частицами. Основные характеристики смоделированных взаимодействий приведены в таблице 1.

Таблица 1

Характеристики смоделированных событий

	$N_{ch} \pm \sigma_{ch}$	$N_{tot} \pm \sigma_{tot}$	GeV/c	
			\bar{P}_T	\bar{P}
фон			0.55	
В-события	14.7 \pm 5.	26.2 \pm 7.7	1.85	3.30

Здесь: $N_{ch}(N_{tot})$ - средняя множественность всех заряженных частиц, возникающих в первичном взаимодействии; $\sigma_{ch}(\sigma_{tot})$ - соответствующие $N_{ch}(N_{tot})$ стандартные отклонения; \bar{P}_T и \bar{P} - средний поперечный и полный импульсы вторичных частиц ПВ (в случае В-событий, величины \bar{P}_T и \bar{P} относятся только к В частицам).

б) Программа VDET трассировки частиц в МСД основана на пакете GEANT 3.18 [13]. Этот пакет позволяет определить установку как набор объемов, заполненных детекторами. Все части установки, включая мишень, и регистрирующие плоскости размещены в главном объеме. Схема МСД (расположение плоскостей и мишени) приведена на рис.4б.

Параметры вершинного детектора предполагаются следующими. Взаимодействие происходит в алюминиевой мишени толщиной 3 мм (с центром в точке $x = y = z = 0$). Пучок движется вдоль оси Z в положительном направлении. Координатные детекторы (кремниевые пластинки размером $7,5 \times 7,5 \text{ см}^2$ и толщиной 300 мкм) располагаются в плоскостях XZ и YZ, перпендикулярно пучку, по шесть пластин в каждой плоскости. Пластинки расставлены на равных расстояниях друг от друга в чередующемся порядке на расстоянии от 0,5 см до 15,5 см от мишени. Шесть пар таких X-Y координатных пластин имеют в центральной области ($2 \times 2 \text{ см}^2$) разрешение 25 мкм, а вне этой области - 50 мкм.

Предполагалось, что эффективность регистрации в каждой пластине - 100 %, шумовые измерения отсутствовали. Средняя длина пробега мезонов до распада $\approx 2 \text{ см}$, но достигала и 10 см. События содержали также посторонние вершины и фоновые треки.

Трассировка частиц проводилась от точки ПВ до границы главного объема. При этом учитывалась возможность распада частиц, все электромагнитные процессы (многократное кулоновское рассеяние, образование пар e^+, e^- , комптон-эффект и др.), а также адронное упругое и неупругое взаимодействия. Последние осуществлялись с помощью программы GHEISHA [15] (часть GEANT). Частицы, которые являлись продуктом упомянутых физических процессов, тоже трассировались.

Средняя множественность треков $\approx 20-30$ на событие. Число вершин (всех, включая посторонние) от 2 до 6, из них вершин распада мезонов - от 2 до 3.

На рис. 4 показано смоделированное событие $pp \rightarrow B\bar{V}X$ в проекции на плоскость XY (4а) и то же самое событие в проекции на плоскость YZ (4б).

Замечание. Наш подход является идеализированным в том смысле, что заменяет адрон-ядерное взаимодействие адрон-адронным и, значит, не учитывает процессы испарения остаточного ядра. Эти процессы являются источником низкоэнергетических частиц, из которых прежде всего нейтроны представляют опасность, т.к. могут неупруго взаимодействовать с кремниевыми пластинками и тем самым имитировать вторичные вершины. Однако, с другой стороны, взаимодействие в пластине вызывает скачок сигнала, что может быть использовано для его обнаружения в других трековых детекторах. Поэтому наш подход можно считать вполне реалистичным.

8. Результаты численного эксперимента

Алгоритм, описанный в п.6, был реализован на языке FORTRAN на ЭВМ VAX-8350. Эта программа прочла 500 событий из созданного банка (п.7) и обработала их за 30 минут. В подавляющем большинстве событий все вершины найдены правильно с очень малыми ошибками. Результаты работы подробно приводятся ниже.

Для сравнения найденных вершин с модельными для каждой найденной вершины вычислялось расстояние до ближайшей к ней модельной вершины и наоборот. Если эти расстояния превышали 4 мм, то вершина считалась ошибочной (или потерянной). Таких вершин оказалось менее 10 %. Для остальных расстояний рассчитанное среднее оказалось равным $\bar{d} \approx 0,33 \text{ мм}$ для отклонений модельных вершин от найденных и $\bar{d} \approx 0,25 \text{ мм}$ - наоборот.

Здесь следует отметить, что определенное число "потерянных" вершин распада мезонов объяснялось тем, что либо из них выходило всего 1-2 трека, попавших в детектор, либо все выходящие треки оказались в узком конусе (вероятность появления таких вершин - до 5%, см. [2]).

Более подробный анализ был проведен для вершин распада мезонов. Для каждой из них было вычислено расстояние до ближайшей к ней найденной вершине по X- и Y-координатам ("поперечное смещение"). Потерянных вершин, для которых смещение $\geq 1 \text{ мм}$, оказалось, как и ранее, менее 10 %. Для остальных средняя величина смещения равнялась $\bar{d} \approx 0,028 \text{ мм}$ (т.е. 28 мкм, что сравнимо со средней ошибкой измерений в детекторах).

Был проделан также анализ относительных ошибок по z-координате. Для каждой вершины распада мезонов было вычислено отклонение Δz (по z-координате) от ближайшей к ней из найденных вершин и относительные ошибки $\Delta z / z_{np}$ (в процентах) - гистограммированы (z_{np} - длина пробега мезона до распада).

При этом учитывались только мезоны с $z_{np} > 5 \text{ мм}$, так как при малых z_{np} дробь z / z_{np} возрастает из-за малого знаменателя. Распределение ошибок $\Delta z / z_{np}$ приведено в гистограмме 1 на рис.5.

Мы видим, что половина вершин (254 из 470) были восстановлены с относительной ошибкой $\leq 1\%$.

Аналогичная гистограмма 2 представляет распределение величин $\Delta z / \langle z_{np} \rangle$ для тех же вершин распада мезонов, где $\langle z_{np} \rangle$ - средняя длина пробега мезона, рассчитанная по формуле

$$\langle z_{np} \rangle = \frac{c \tau}{m} \sqrt{p^2 + m^2},$$

где p и m - импульс и масса мезона, τ - время жизни, c - скорость света.

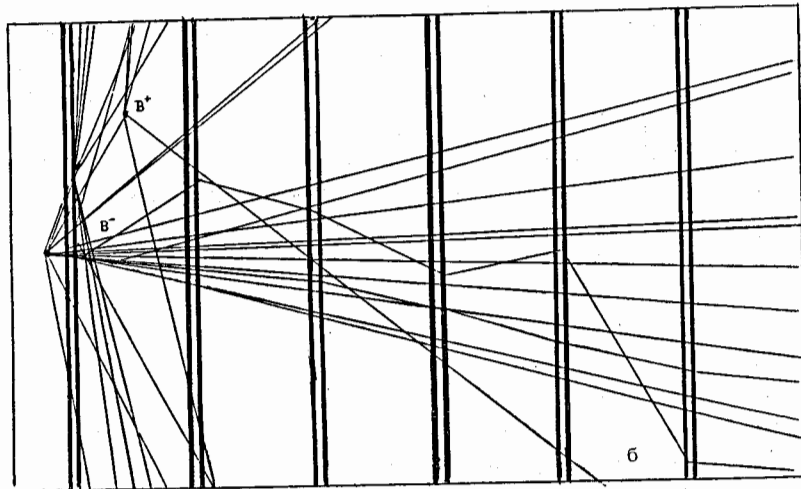
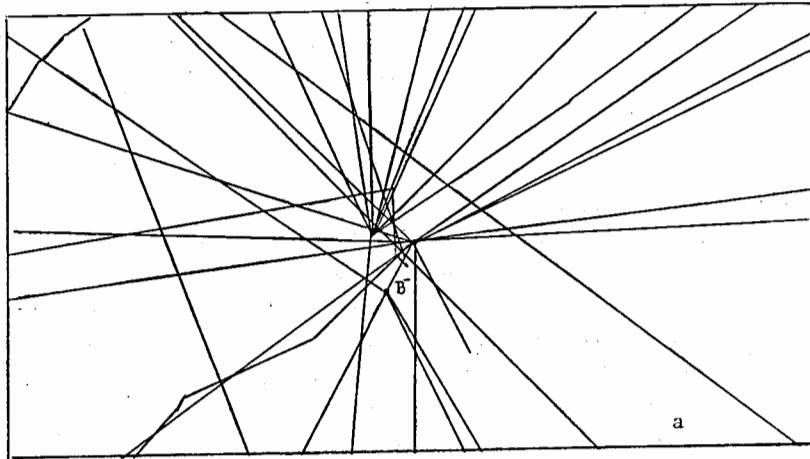


Рис. 4

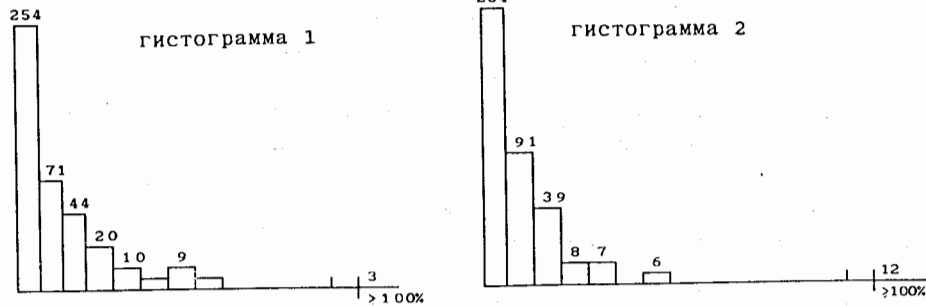


Рис. 5

Полученные результаты свидетельствуют о хорошем соответствии координат вершин, вычисленных с помощью изложенной методики, и данных, полученных при моделировании отклика установки на модельные события.

Л И Т Е Р А Т У Р А

1. L.Rossi, CERN/EP 88-63, 1988.
- В.Сох, in: Annals of the N.-Y. Acad. of Sciences, V.535,1988,224.
- Зайцев А.М. в кн.: Материалы рабочего совещания "Физика на УНК", Протвино, ИФВЭ, 1989, с.27.
2. Ососков Г.А. и др. ОИЯИ Р10-90-367, Дубна, 1990.
3. Yatsunenکو Yn. JINR E1-88-907, Dubna, 1988.
4. Иоселиани Ц.И. и др. ОИЯИ Р10-86-666, Дубна, 1986.
5. Chernov N.I. et al., Aplikace Matematiky (1991), in press.
6. Федоров В.В. Теория оптимального эксперимента. М., Наука, 1971.
7. Pazman A. Foundations of Optimum Experimental Design, VEDA, Bratislava, 1986.
8. Mallet A. Biometrika 73 (1986),n.3, 645-656.
9. Böhning D. Biometrika 76 (1989),n.2, 375-383.
10. Lindsay B.G. Annals of Stat. 11(1983),n.1, 86-94.
11. Tornsay B. A moment inequality and monotonicity of an algorithm, in: Semi-Infinite Programming and Applications, A.V.Fiacco and K.O.Kotranek(Eds), Springer-Verlag, Berlin, 249-260.
12. Bengtsson H.V., Sjöstrand T. Comp.Phys.Comm. 46(1987),43-42.
13. Brun R. et al, GEANT3, CERN DD/EE/ 84-1 (1987).
14. Sjöstrand T. Comp.Phys.Comm. 39 (1986) 347.
15. Fesefeld H. Nucl.Instr.& Meth. A263 (1988), 114; A292(1990),279.

Рукопись поступила в издательский отдел
5 июля 1991 года.