

Сибирское отделение Российской Академии наук
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ им. Г.И. Будкера

А.В. Боженко, Д.А. Букин, В.Н. Иванченко,
А.А. Сальников, Э.К. Силагадзе

ПАКЕТ ПРОГРАММ
АППРОКСИМАЦИИ СЕЧЕНИЙ
В ЭКСПЕРИМЕНТЕ СНД

ИЯФ 99-103

НОВОСИБИРСК
1999

**Пакет программ
аппроксимации сечений в эксперименте СНД**

*А.В. Боженок, Д.А. Букин, В.Н. Иванченко,
А.А. Сальников, З.К. Силагадзе*

Институт ядерной физики СО РАН, 630090, Новосибирск, Россия

Аннотация

Сферический нейтральный детектор (СНД) предназначен для изучения процессов e^+e^- аннигиляции в области энергий до 1.4 GeV. Обработка эксперимента на детекторе завершается процедурой, концептуально одинаковой для различных изучаемых процессов – определением значений и ошибок параметров теоретической модели по полученным экспериментальным данным. Для решения этой задачи в эксперименте СНД был разработан универсальный пакет программ аппроксимации данных e^+e^- экспериментов, в котором используются как оригинальные программы, подготовленные для обработки эксперимента СНД, так и программа MINUIT, разработанная в CERN.

**Data approximation procedure
for SND experiment**

*A.V. Bozhenok, D.A. Bukin, V.N. Ivanchenko,
A.A. Salnikov, Z.K. Silagadze*

Abstract

Spherical Neutral Detector (SND) is intended to study e^+e^- annihilation processes at energy range up to 1.4 GeV. The studying of different processes is usually accomplished by conceptually similar procedure of estimation of parameters and their error values. To facilitate the task an universal package for approximation of experimental data was developed. The package is based on MINUIT minimization routine, written in CERN.

© *Институт ядерной физики им. Г.И. Будкера СО РАН, Россия*

Оглавление

1	Введение	4
2	Постановка задачи	5
3	Вычисление радиационных поправок	7
4	Учет разброса энергии	9
5	Ожидаемое число событий	10
6	Функция правдоподобия	11
7	Минимизация. Ограничение параметров модели. Штрафные функции	13
8	Эффективность	15
9	Учет фона. Совместная аппроксимация нескольких процессов	17
10	Логическая схема программы	19
11	Установка библиотек пакета	20
12	Подготовка задания	21
13	Типичный вид задания	30
14	Запуск задания	36
15	Представление результатов	38
16	Краткое описание библиотеки RADCOR	46
17	Графическое представление результатов	48
18	Рекомендуемое использование	50
19	Заключение	52

1 Введение

Одной из задач, часто встречающихся при обработке экспериментальных данных в физике высоких энергий, является описание измеренного сечения изучаемого процесса теоретической моделью. Задача решается, как правило, методом максимума правдоподобия, и, при наличии мощной программы минимизации, такой как, например, MINUIT [1], не является технически сложной. Однако, из-за существования многих дополнительных факторов, таких как радиационные поправки, разброс энергии в пучке, неточности измерения светимости и т.д., которые следует учитывать при нахождении параметров, задача становится достаточно трудоемкой.

Мы попытались выделить эти общие для конкретного случая e^+e^- взаимодействия факторы и оформить их в виде программы аппроксимации сечений, в которой на долю пользователя остается, на наш взгляд, самая интересная часть — выбор теоретической модели. Пользователь, естественно, должен иметь экспериментальные данные. В качестве дополнительного сервиса к программе аппроксимации предлагается библиотека моделей сечений и пакет визуализации полученных результатов.

Данная программа уже несколько лет с успехом используется при обработке данных с детектора СНД [2].

2 Постановка задачи

Типичный эксперимент на встречных e^+e^- пучках проводится методом сканирования. Коллайдер настраивается таким образом, чтобы среднее значение энергии в системе центра масс сталкивающихся пучков равнялось E_i . После настройки коллайдера в течении довольно продолжительного времени T_i детектором регистрируются события аннигиляции $e^+e^- \rightarrow X$ в различные конечные состояния X . Когда измерения в точке E_i завершены, коллайдер перенастраивается на следующую точку. Измерения в каждой энергетической точке проводятся независимо.

Во время обработки накопленных экспериментальных данных при помощи некоторых условий отбора из всей массы событий выбираются события интересующего нас процесса. Пусть для точки сканирования с энергией E_i количество отобранных событий составляет N_i . Для начала рассмотрим простой случай идеально-го отбора, когда среди отобранных событий отсутствуют события фона. Тогда можно определить физически наблюдаемое сечение в виде:

$$\sigma(E_i) = \frac{N_i}{\varepsilon_i \cdot L_i},$$

где коэффициенты пропорциональности: L_i – интеграл светимости и ε_i – эффективность отбора. Интеграл светимости является характеристикой эксперимента и его значение определяется по событиям хорошо изученных процессов с высокой статистикой, сечение которых можно вычислить теоретически с хорошей точностью. Как правило, это КЭД-процессы $e^+e^- \rightarrow e^+e^-$ и $e^+e^- \rightarrow \gamma\gamma$.

Эффективность ε_i , определяется акцептансом детектора и тем, как сильно условия отбора подавляют события искомого процесса. Обычно эффективность находят, применяя те же условия отбора к событиям моделирования изучаемого процесса.

Итак, в каждой энергетической точке имеется набор измерен-

ных величин: $E_i, L_i, T_i, \varepsilon_i, N_i$. Большинство этих величин измеряются с ошибками: $\Delta E_i, \Delta L_i, \Delta \varepsilon_i$.

Кроме того, из-за стохастических процессов в пучке величина энергии в системе центра масс для отдельного столкновения распределена с некоторым среднеквадратичным отклонением относительно среднего значения. В дальнейшем эта величина называется разбросом энергии δE_i .

Физически наблюдаемое сечение обычно представляют в виде:

$$\sigma(E) = \sigma_0(E)(1 + \delta(E, \sigma_0)), \quad (1)$$

где σ_0 - борновское сечение, т.е. сечение в первом порядке теории возмущений, а $\delta(E, \sigma_0)$ - радиационная поправка (РП), учитывающая изменение сечения за счет излучения радиационных фотонов. Эта радиоправка является функционалом сечения σ_0 .

Определение σ_0 - задача, которую ставит перед собой экспериментатор. Чтобы найти σ_0 нужно вычислить РП, а для этого необходимо знать зависимость σ_0 от энергии. Один из способов выхода из этого заколдованного круга — параметризация борновского сечения некоторой модельной функцией:

$$\sigma_0(E, \vec{\alpha}), \quad (2)$$

где $\vec{\alpha}$ - набор параметров, и проведение аппроксимации экспериментальных данных с использованием этой функции. В результате аппроксимации находится оптимальный набор параметров $\vec{\alpha}_{opt}$, и некоторый критерий согласия, например, значение χ^2 . Величина χ^2 позволяет понять, хорошо ли соответствует проверяемая модель сечения (2) наблюдаемым экспериментальным данным. Если экспериментатор не проверяет теоретическую модель, а просто хочет измерить борновское сечение, т.е. извлечь из формулы (1) величины радиоправок, то необходимо, чтобы выбранная параметризация хорошо описывала экспериментальные данные.

3 Вычисление радиационных поправок

Вычисление радиационных поправок (РП) проводится согласно работе [3]. В этой работе РП к начальным частицам рассчитываются, используя хорошо развитый в КХД метод ренормгрупп. Точность вычисления РП составляет около 0.1 %. Излучение конечными частицами нами не рассматривается. Наблюдаемое сечение представляется в следующем виде:

$$\begin{aligned} \sigma_r(E, \vec{\alpha}) &= \int_0^{1-\epsilon_1} dx_1 \int_0^{1-\epsilon_2} dx_2 \sigma_0(\sqrt{s(1-x_1)(1-x_2)}, \vec{\alpha}) \\ &\times D(x_1, s) D(x_2, s) K, \end{aligned} \quad (3)$$

где s - квадрат энергии в системе центра масс, $D(x, s)$ - плотность вероятности одной из начальных частиц потерять долю энергии x , $K = 1 + \frac{\alpha}{\pi}(\frac{\pi^2}{3} - \frac{1}{2})$. Анализ поведения $D(x, s)$ показывает, что при $x \rightarrow 0$ в формуле (4) имеется интегрируемая расходимость, в результате основная часть интеграла набирается при малых x . Поскольку интегрирование по формуле (4) требует больших затрат машинного времени, мы использовали приближенную формулу, получающуюся заменой $(1-x_1)(1-x_2) \sim (1-z)$:

$$\sigma_r(E, \vec{\alpha}) = \int_0^{\frac{2E_{max}}{\sqrt{s}}} dz \sigma_0(\sqrt{s(1-z)}, \vec{\alpha}) F(z, s). \quad (4)$$

Погрешность, получаемую при использовании такого приближения можно оценить, посчитав радиоправки обоими способами для заданного сечения. Например, для сечения 3π разница в сечениях с радиоправкой в максимуме ϕ мезона ($E = 1020$ MeV) составила 0.15%, а в подложке рядом с ϕ мезоном ($E = 1040$ MeV) — 3%, в-основном из-за излучения жестких фотонов со сбросом на ω мезон. То же сравнение для сечения $K_S K_L$ дало соответственно величины 0.06% и 0.2%.

Поляризация вакуума не включается в поправки по общепринятой договоренности. Вместо этого она учтена в сечении в относительной вероятности перехода в e^+e^- - пару. Проблема учета поляризации вакуума (вносить ли ее в сечение или добавлять в качестве дополнительного множителя в формулы (3), (4)) возникла при исследовании распадов Υ мезонов, где вклад поляризации вакуума значителен. Обсуждение этой проблемы, в результате которого и была выработана указанная договоренность, можно посмотреть, например, в работе [4]. В программе имеется возможность включить учет поляризации вакуума дополнительным множителем под интегралом в соотношении (4).

В формуле (4) значение интеграла зависит от выбора величины максимальной энергии E_{max} . В программе эту величину задает пользователь. Очевидно, что точно такое же ограничение на E_{max} должно присутствовать при моделировании событий изучаемого процесса для вычисления эффективности.

Часто желательно подобрать E_{max} так, чтобы при вычислении сечения в точке с энергией $E = \sqrt{s}$, интегрируемой оказалась область вплоть до E_{min} . Как легко заметить из (4) эти три энергии связывает соотношение:

$$E_{max} = \frac{\sqrt{s}}{2} \left(1 - \left(\frac{E_{min}}{\sqrt{s}} \right)^2 \right). \quad (5)$$

Выбор величины E_{max} – достаточно ответственная задача. Обычно пользователи программы указывают большое значение E_{max} , интеграл в (4) ограничивается в этом случае энергетическим поведением сечения, а именно, занулением сечения при малых энергиях, и, кроме того, кинематическим пределом $E_{max} < \sqrt{s}/2$.

В том случае, когда сечение быстро зануляется при уменьшении энергии, бессмысленно использовать большие значения E_{max} . Например, при изучении сечения $K_S K_L$ в области ϕ мезона (980 – 1060 MeV) из формулы (5) следует, что вполне достаточно установить $E_{max} = 77$ MeV.

4 Учет разброса энергии

Как уже было сказано выше, энергия в системе центра масс сталкивающихся электрона и позитрона распределена по нормальному закону со среднеквадратичным отклонением δE_i относительно среднего значения E_i :

$$N(E, E_i, \delta E_i) = \frac{1}{\sqrt{2\pi}\delta E_i} \exp\left(-\frac{1}{2}\left(\frac{E - E_i}{\delta E_i}\right)^2\right). \quad (6)$$

Разброс энергий приводит к тому, что при проведении эксперимента сечение усредняется по интервалу энергий. Для правильного учета этого эффекта необходимо подобным образом провести усреднение при вычислении сечения, а именно провести свертку сечения (4) и распределения энергии в системе центра масс (6):

$$\sigma_e(E_i, \vec{\alpha}) = \int_0^{+\infty} \sigma_r(E, \vec{\alpha}) N(E, E_i, \delta E_i) dE. \quad (7)$$

Программа изначально ориентировалась на обработку экспериментальных данных, записанных на коллайдере ВЭПП-2М, где величина разброса энергии составляет $\delta E_i \sim 0.2$ MeV. С другой стороны, ширины узких векторных резонансов в изучаемой области энергии составляют несколько MeV. Поэтому, для ускорения вычисления интеграла (7) в программе было сделано разложение подинтегральной функции в ряд Тейлора по малому параметру $(E - E_i)$ с отбрасыванием членов $\sim O((E - E_i)^3)$:

$$\begin{aligned} \sigma_e(E_i, \vec{\alpha}) &\simeq \int \left(\sigma_r(E_i, \vec{\alpha}) + \frac{d\sigma_r}{dE}(E_i, \vec{\alpha}) \cdot (E - E_i) \right. \\ &\quad \left. + \frac{1}{2} \frac{d^2\sigma_r}{dE^2}(E_i, \vec{\alpha}) \cdot (E - E_i)^2 \right) N(E, E_i, \delta E_i) dE \\ &= \sigma_r(E_i, \vec{\alpha}) + 0 + \frac{1}{2} \frac{d^2\sigma_r}{dE^2}(E_i, \vec{\alpha}) \cdot (\delta E_i)^2. \end{aligned} \quad (8)$$

Поправку, которую вносит разброс энергии, можно оценить, аппроксимировав вершину резонанса с массой m , шириной Γ и величиной сечения в максимуме H параболой $\sigma(E) \simeq -\frac{4H}{\Gamma^2}(E-m)^2 + H$. Тогда вторая производная от сечения в максимуме резонанса составит: $\frac{d^2\sigma}{dE^2}(m) \simeq -\frac{8H}{\Gamma^2}$, а относительная поправка к сечению $\frac{\Delta\sigma}{\sigma} \sim \frac{-4(\delta E_i)^2}{\Gamma^2}$. На ϕ мезоне, например, при разбросе энергии 0.2 MeV поправка оценивается в 1%.

5 Ожидаемое число событий

Таким образом, теперь для точки i имеется экспериментальный набор данных $E_i, N_i, L_i, T_i, \delta E_i, \Delta E_i, \Delta L_i$, а также вычисленная величина наблюдаемого сечения $\sigma_e(E_i, \vec{\alpha})$ с учтенными радипоправками и поправками на разброс энергии, и эффективность отбора ε_i , полученная по моделированию.

Ожидаемое число событий можно записать, умножив вычисленное сечение на эффективность и светимость:

$$M_i(\vec{\alpha}) = \sigma_v(E_i, \vec{\alpha}) L_i, \quad (9)$$

где

$$\sigma_v(E_i, \vec{\alpha}) = \sigma_e(E_i, \vec{\alpha}) \varepsilon_i. \quad (10)$$

Квадрат ошибки ожидаемого числа событий определяется точностями измерения энергии, светимости и эффективности:

$$\begin{aligned} (\Delta M_i(\vec{\alpha}))^2 &= (\Delta L_i)^2 \left(\sigma_v(E_i, \vec{\alpha}) \right)^2 + (\Delta E_i)^2 \left(\varepsilon_i \frac{d\sigma_v}{dE}(E_i, \vec{\alpha}) L_i \right)^2 \\ &+ (\Delta \varepsilon_i)^2 \left(\sigma_e(E_i, \vec{\alpha}) L_i \right)^2. \end{aligned} \quad (11)$$

6 Функция правдоподобия

После того, как получено ожидаемое число событий $M_i(\vec{\alpha})$ и его ошибка $\Delta M_i(\vec{\alpha})$, можно провести сравнение этого числа с наблюдаемым в эксперименте N_i .

Для этой цели некоторым образом должна быть построена логарифмическая функция правдоподобия

$$\mathcal{L}_i(M_i(\vec{\alpha}), \Delta M_i(\vec{\alpha}), N_i) \quad (12)$$

для точки i . Считая точки независимыми, можно получить логарифмическую функцию правдоподобия для всех I точек:

$$\mathcal{L}(\vec{\alpha}) = \sum_{i=1}^I \mathcal{L}_i(M_i(\vec{\alpha}), \Delta M_i(\vec{\alpha}), N_i). \quad (13)$$

Оптимальный набор параметров $\vec{\alpha}_{opt}$ находится минимизацией этой функции по параметрам $\vec{\alpha}$.

Конкретный вид функции правдоподобия (12) для отдельного измерения в точке i зависит от вида распределения, которому подчиняется число событий. Обычно этим распределением принято считать распределение Пуассона, т.е. вероятность получения N_i событий при среднем M_i запишется как

$$w_i = e^{-M_i} \frac{M_i^{N_i}}{N_i!}.$$

Логарифмическая функция правдоподобия в этом случае имеет вид:

$$\mathcal{L}_i = M_i - N_i \cdot \ln M_i + \ln N_i! \quad (14)$$

Как легко видеть, в функцию правдоподобия не вошла ошибка ожидаемого числа событий ΔM_i . На самом деле приведенная формула не совсем корректна, необходимо было учесть ошибку ожидаемого числа событий, которая появляется из-за неточного измерения энергии, светимости и эффективности. Приближение, использованное в формуле (14), будет справедливо, если относительная ошибка измеренного числа событий $\Delta N_i/N_i \sim 1/\sqrt{N_i}$ будет

много больше относительной ошибки ожидаемого числа событий $\Delta M_i/M_i$, то есть наблюдаемое число событий должно быть мало.

Противоположная ситуация реализуется, когда статистика достаточно велика. В этом случае распределение Пуассона переходит в распределение Гаусса:

$$w_i = \frac{1}{\sqrt{2\pi}\Delta_i} e^{-\frac{(M_i - N_i)^2}{2\Delta_i^2}}, \quad (15)$$

где

$$\Delta_i = \sqrt{(\Delta N_i)^2 + (\Delta M_i)^2} \quad (16)$$

– пересчитанная ошибка, которая включает в себя ошибку измерения числа событий и ошибки измерения других величин. Логарифмическая функция правдоподобия примет вид:

$$\mathcal{L}_i = \frac{1}{2} \left(\ln(2\pi\Delta_i^2) + \frac{(M_i - N_i)^2}{\Delta_i^2} \right).$$

Она совпадает с функцией χ^2 , с точностью до коэффициента $\frac{1}{2}$ и дополнительного слагаемого. Это дополнительное слагаемое слабо зависит от параметров минимизации $\vec{\alpha}$, поэтому перепишем функцию правдоподобия в следующем виде:

$$\mathcal{L}_i \simeq \frac{1}{2} \frac{(M_i - N_i)^2}{\Delta_i^2} = \frac{\chi^2}{2}. \quad (17)$$

Оба рассмотренных подхода описывали стандартное пуассоновское распределение для наблюдаемого числа событий N_i для разнотипной статистики. В случае не стандартного распределения, если пользователем заданы верхняя ΔN_i^+ и нижняя ΔN_i^- ошибки наблюдаемого числа событий, функция правдоподобия будет записываться в следующем виде, который условно называется асимметричным χ^2 :

$$\mathcal{L}_i = \frac{1}{2} \frac{(M_i - N_i)^2}{\tilde{\Delta}_i^2}, \quad (18)$$

где

$$\tilde{\Delta}_i^2 = \begin{cases} (\Delta M_i)^2 + (\Delta N_i^-)^2, & N_i < M_i \\ (\Delta M_i)^2 + (\Delta N_i^+)^2, & N_i \geq M_i \end{cases}. \quad (19)$$

7 Минимизация. Ограничение параметров модели. Штрафные функции

Набор параметров $\vec{\alpha}$ определяет поведение борновского сечения $\sigma_0(E, \vec{\alpha})$, заданного пользователем. Параметры $\vec{\alpha}$ могут быть любыми вещественными величинами. На практике обычно используют какие-либо характеристики изучаемых процессов, например, массы и ширины резонансов, относительные вероятности распадов, амплитуды и фазы интерференции, и т.д. Оптимальный набор параметров $\vec{\alpha}_{opt}$, то есть тот, при котором достигается наилучшее соответствие модели и измеренных значений, определяется нахождением минимума логарифмической функции правдоподобия. Минимизация проводится программой MINUIT [1] из библиотеки CERLIB [5].

Иногда возникает желание учесть при минимизации значение k -того параметра $\alpha_{k,0} \pm \Delta\alpha_{k,0}$, полученное в результате других экспериментов, например, среднемировое значение, взятое из таблиц [6]. Эта возможность в нашей программе реализована в виде штрафной функции:

$$\frac{1}{2} \frac{(\alpha_k - \alpha_{k,0})^2}{(\Delta\alpha_{k,0})^2}, \quad (20)$$

добавляемой к логарифмической функции правдоподобия. Параметр, на который накладывается подобное условие в дальнейшем будет называться привязанным к значению $\alpha_{k,0}$ с величиной привязки $\Delta\alpha_{k,0}$.

Другая необходимая возможность, которую должна предоставлять программа минимизации – ограничения, накладываемые на

значения параметров. В качестве простого примера, иллюстрирующего появление таких требований можно привести измерение ширины резонанса в эксперименте. Ясно, что при аппроксимации необходимо, чтобы ширина принимала только положительные значения. Ниже будет рассмотрено, какими средствами этого можно достичь в нашей программе.

MINUIT, на базе которой построена минимизация, позволяет ограничить k -тый параметр α_k на некотором отрезке $[a_k, b_k]$. Это достигается преобразованием этого отрезка на интервал $(-\infty, +\infty)$ и обратно следующими формулами:

$$\begin{aligned} \beta_k &= \arcsin\left(2\frac{\alpha_k - a_k}{b - a} - 1\right), & \alpha_k &\in [a, b] \\ \alpha_k &= a + \frac{b - a}{2}(\sin \beta_k + 1), & \beta_k &\in (-\infty, +\infty).. \end{aligned} \quad (21)$$

Минимизация проводится для преобразованного неограниченного параметра β_k , скрытого от пользователя.

Однако, такой метод ограничения имеет свои недостатки, в [1] приводятся некоторые из них, здесь будут отмечены только самые неприятные с точки зрения авторов. Во-первых, для стандартного ограничения параметра необходимо задавать одновременно нижнюю и верхнюю границу, а иногда требуется ограничение только с одной стороны. Во-вторых, из-за конечной точности вычислений параметр может выходить за границу, в некоторых случаях это недопустимо. По этим причинам, в частности, в том же описании MINUIT не рекомендуется пользоваться стандартным ограничением параметров.

В нашей программе для ограничения параметров реализован дополнительный механизм, использующий штрафные функции. Если требуется, чтобы параметр α_k находился выше определенного предела $\alpha_{k,low}$, то вместо значения этого параметра во всех

вычислениях используется ограниченная величина:

$$\tilde{\alpha}_k = \begin{cases} \alpha_{k,low}, & \alpha_k < \alpha_{k,low} \\ \alpha_k, & \alpha_k \geq \alpha_{k,low} \end{cases}, \quad (22)$$

а к функции правдоподобия делается следующая добавка:

$$\begin{cases} \frac{1}{2} \frac{(\alpha_k - \alpha_{k,low})^2}{(\Delta\alpha_{k,low})^2}, & \alpha_k < \alpha_{k,low} \\ 0, & \alpha_k \geq \alpha_{k,low} \end{cases}. \quad (23)$$

Очевидно, что, во-первых, параметр $\tilde{\alpha}_k$, используемый в формулах вычисления сечения, не будет принимать значений ниже $\alpha_{k,low}$, а, во-вторых, функция минимизации при перемещении параметра в область ниже $\alpha_{k,low}$ начнет сразу же возрастать. Это возрастание функции правдоподобия будет тем быстрее, чем меньше будет величина $\Delta\alpha_{k,low}$. В дальнейшем изложении $\Delta\alpha_{k,low}$ называется величиной привязки параметра α_k к границе области $\alpha_{k,low}$, а параметр α_k – ограниченным снизу величиной $\alpha_{k,low}$. Аналогичным образом можно установить верхнее ограничение на значение параметра.

В приведенном выше примере с шириной резонанса, можно добиться необходимого эффекта, ограничив ширину снизу нулем, с величиной привязки порядка 0.1 MeV.

Надо отметить, что введение даже достаточно слабой привязки может иногда заметно улучшить сходимость, особенно в случае сильно коррелирующих параметров.

8 Эффективность

Эффективность обычно вычисляется с помощью моделирования процесса методом Монте-Карло. Как уже было сказано выше, при моделировании событий процесса желательно учитывать радиоправки также, как они учитываются при вычислении сечения в (4),

т.е. использовать одинаковое значение E_{max} и одинаковую плотность вероятности излучения фотона $F(z, s)$. В частности, при анализе данных СНД моделирование проводится с помощью того же пакета программ [7], что и при аппроксимации данных.

Эффективность ε_i , вычисленная по моделированию, имеет ошибку $\Delta\varepsilon_i$. Экспериментатор может считать эти ошибки независимыми и учитывать их таким образом, как это делается в (11). Однако, если имеется некоторое предположение о виде зависимости эффективности от энергии, т.е. экспериментатор может параметризовать эффективность функцией:

$$\varepsilon(E, \vec{\beta}), \quad (24)$$

где $\vec{\beta}$ - вектор параметров зависимости, то более правильно учесть эту зависимость в аппроксимации.

В этом случае ошибки эффективности $\Delta\varepsilon_i$ в формуле (11) полагаются равными нулю, а значения эффективности вычисляются по формуле (24).

В программе можно также задать режим нахождения оптимальных параметров $\vec{\beta}_{opt}$ зависимости (24). В этом режиме к функции правдоподобия будет сделана добавка вида:

$$\frac{1}{2} \sum_{i=1}^I \frac{(\varepsilon_i - \varepsilon(E_i, \vec{\beta}))^2}{(\Delta\varepsilon_i)^2}. \quad (25)$$

Вектор параметров аппроксимации теперь будет содержать параметры борновского сечения и параметры эффективности: $\vec{\alpha} \rightarrow (\vec{\alpha}, \vec{\beta})$. Оптимальные значения всех параметров, в частности, параметров зависимости (24) будут найдены в результате минимизации.

9 Учет фона. Совместная аппроксимация нескольких процессов

Выше был рассмотрен случай “идеального” отбора, когда экспериментатор располагает чистой выборкой событий изучаемого процесса. На практике, к сожалению, условия отбора с некоторой вероятностью пропускают фон других процессов. В общем случае неидеального отбора, в эксперименте измеряются те же величины, что и в случае идеального отбора, но предполагается, что в отобранное число событий дали вклад P процессов с индексами $p = 1, \dots, P$. Пользователь имеет параметризованные общим набором параметров $\vec{\alpha}$ модели борновских сечений для всех этих процессов $\sigma_{0,p}(E, \vec{\alpha})$, а также может определить эффективности отбора событий $\varepsilon_{i,p} \pm \Delta\varepsilon_{i,p}$ каждого процесса по соответствующему моделированию. Теперь ожидаемое число событий будет представляться суммой вкладов всех процессов:

$$\begin{aligned}\sigma_v(E_i, \vec{\alpha}) &= \sum_{p=1}^P \sigma_{\varepsilon,p}(E_i, \vec{\alpha}) \varepsilon_{i,p}, \\ M_i(\vec{\alpha}) &= \sigma_v(E_i, \vec{\alpha}) L_i,\end{aligned}\quad (26)$$

а квадрат ошибки ожидаемого числа событий — суммой квадратов ошибок разных вкладов:

$$\begin{aligned}(\Delta M_i(\vec{\alpha}))^2 &= (\Delta L_i)^2 \left(\sum_{p=1}^P \sigma_{v,p}(E_i, \vec{\alpha}) \right)^2 \\ &+ (\Delta E_i)^2 \left(\sum_{p=1}^P \varepsilon_{i,p} \frac{d\sigma_{r,p}}{dE}(E_i, \vec{\alpha}) L_i \right)^2 + L_i^2 \cdot \sum_{p=1}^P \left(\Delta\varepsilon_{i,p} \sigma_{\varepsilon,p}(E_i, \vec{\alpha}) \right)^2.\end{aligned}\quad (27)$$

Последующие действия по аппроксимации и нахождению оптимального набора параметров не отличаются от случая с идеальным отбором. В дальнейшем изложении сечение $\sigma_p(E)$ будет называть-

ся компонентой, а сумма таких сечений, описывающая один набор экспериментальных данных – процессом.

Кроме возможности использования составных процессов, по историческим причинам в программе сохранилась возможность подмешивания к сечению процесса $\sigma_v(E)$ сечения нерезонансного фона: $\sigma_L(E)$. Появление этого вклада приводит к добавлению в число ожидаемых событий слагаемого $\sigma_L(E_i) \cdot L_i$, а в квадрат ошибки числа событий слагаемого $(\Delta L_i \sigma_L(E_i))^2$. Эта особенность программы используется редко, так как не учитывает радиоправки и поправки на разброс энергии, то есть сечение $\sigma_L(E)$ не должно иметь резонансных структур, либо они должны быть учтены при вычислении $\sigma_L(E)$.

Другой источник дополнительных событий, который никак не был освещен ранее — это космический фон. В программе космический фон можно учесть при помощи добавки в число ожидаемых событий слагаемого вида $f(E_i; \vec{\alpha}) \cdot T_i$, где T_i – время набора статистики, а $f(E_i; \vec{\alpha})$ – функция, написанная пользователем, параметризирующая скорость счета космического фона. Обычно, если космический фон дает вклад в отобранные события, и его желательно учесть, то в простейшем случае эту функцию полагают равной одному из параметров минимизации: $f = \alpha_k$.

Иногда может потребоваться совместная аппроксимация S наборов экспериментальных данных с индексами $s = 1, \dots, S$, например, одновременная аппроксимация данных нескольких сканирований ϕ мезона, проведенных на одном детекторе. Считая данные сканирований независимыми, можно переопределить логарифмическую функцию правдоподобия как сумму логарифмических функций правдоподобия для отдельного набора данных:

$$\mathcal{L}(\vec{\alpha}) = \sum_{s=1}^S \mathcal{L}_s(\vec{\alpha}).$$

И наконец, наиболее общим случаем является совместное рас-

смотрение нескольких процессов:

$$\mathcal{L}(\vec{\alpha}) = \sum_{s=1}^S \sum_{p_s=1}^{P_s} \mathcal{L}_{p_s}(\vec{\alpha}) \quad (28)$$

некоторые из которых могут являться обычными процессами, т.е. $P_s = 1$. В дальнейшем, для удобства изложения будет введен сквозной индекс компоненты процесса $j = 1 \dots J$, где $J = \sum_{s=1}^S P_s$, а также сквозной индекс отдельного измерения $r = 1 \dots R$, где $R = \sum_{s=1}^S \sum_{p=1}^{P_s} I_{p_s}$. Естественно, для такого представления имеется однозначное соответствие между индексами $j \leftrightarrow (s, p)$ и $r \leftrightarrow (s, p, i)$.

10 Логическая схема программы

Структурно программа делится на 5 частей: пользовательская, библиотека `fit`, библиотека `radcor`, библиотека `kio` и библиотека `CERNLIB`, из которой для работы необходима программа минимизации `MINUIT`. Логическая схема работы программы представлена на рис. 1.

Программы библиотеки `fit` обеспечивают связь между пользовательской частью и `MINUIT`, а также осуществляют ввод/вывод. В библиотеке `radcor` содержится программа, рассчитывающая радиопоправки (`rcvcs`), а также набор программ для расчета стандартных сечений. Библиотека `kio` поддерживает независимые от операционной системы и архитектуры работу с данными, и низкоуровневый ввод/вывод.

Пользовательская часть включает в себя файл с программами, который рекомендуется написать на языке `FORTRAN` и файл с командами `MINUIT`. Правила подготовки задания излагаются в соответствующем разделе.

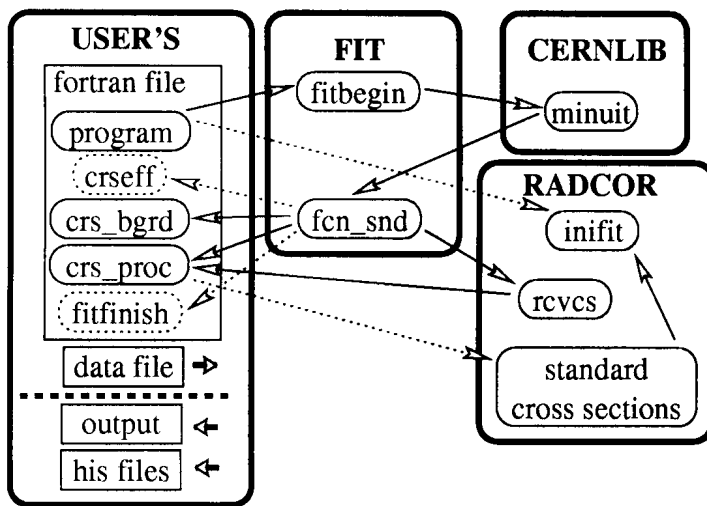


Рис. 1: Логическая схема работы программы. Точечными линиями показаны опциональные возможности. Пользовательская часть делится на входную часть (сверху) и результаты работы (снизу).

11 Установка библиотек пакета

В данном разделе будет описана последовательность установки библиотек пакета. Установка успешно производилась на операционной системе Linux RedHat-5.0, 6.0, с компилятором g77. При установке библиотек на других системах возможны различные сбои, в этом случае обратитесь к разработчикам пакета.

В кластере СНД библиотеки уже установлены, можете переходить к следующему разделу руководства.

Установка библиотек CERN, которые необходимы для работы программы – отдельная задача, которая выходит за рамки данной публикации. Проверить, доступны ли вам библиотеки CERN, можно, выдав команду: `cernlib`. Если появится сообщение “command not found”, это означает, что либо библиотеки не установлены,

либо у вас не настроены переменные среды для доступа к этим библиотекам. В этом случае обратитесь к вашему системному администратору.

Для установки остальных библиотек необходимо:

1. получить исходные тексты библиотеки, которые можно получить по адресу http://sndfs0.inp.nsk.su/SND_Software/index.html в виде tar-файла `fit-<версия>.tar.gz`.
2. распаковать их командой `gzip -dc fit-<версия>.tar.gz | tar xvf -;`
3. перейти в поддиректорию `fit-<версия>`;
4. исполнить следующую команду:

```
gmake -f SMDROOT=<директория установки> SMDMKROOT='pwd'/snd-main \  
SMDLOCALS=-ALL- snd-main/Makefile.main \  
TOPDIRS='snd-main cocha/kio radcor fit' install
```

После этого в `<директории установки>` будут созданы поддиректории `lib`, `bin`, `include`, в которые и будут помещены необходимые библиотеки и исполняемые файлы.

12 Подготовка задания

Как уже было указано выше, для проведения аппроксимации экспериментальных данных пользователю необходимо подготовить программный файл и файл с командами MINUIT.

В программном файле должны быть представлены головная программа `program`, подпрограмма `crs_proc`, вычисляющая теоретическое сечение процессов, и подпрограмма `crs_bgrd`, вычисляющая расчетный нерезонансный фон. Если пользователь желает аппроксимировать эффективность, то зависимость эффективности от параметров минимизации и от энергии должна быть также реализована в этом файле в виде подпрограммы-функции `crseff`. Если пользователю необходимо произвести какое-либо дополнительное общение с `minuit` самостоятельно, по окончании или в процессе минимизации, то в этот файл надо также поместить подпрограмму `fitfinish`.

Описания вызовов подпрограмм в этом разделе будут придерживаться следующего соглашения: имена переменных будут начинаться с букв,

определяющих тип переменной: целые начинаются с ‘i’, переменные с двойной точностью начинаются с ‘d’, строковые – с ‘c’. Сразу за буквами, определяющими тип переменной, в имени переменной может присутствовать ‘_’, который является признаком того, что переменная будет изменяться в подпрограмме. Например, ‘d_crs’ - переменная с двойной точностью, которая будет изменена в подпрограмме. Кроме того, для описания массивов за именем переменной будут следовать скобки ‘()’, внутри которых будет помещена длина массива. В качестве длины в описании чаще всего будет использоваться имя целой переменной, которая тоже передается в подпрограмму.

В головной программе пользователю надо подготовить массивы данных, а затем передать их в программу `fitbegin`. Вызов в `fitbegin` выглядит следующим образом:

```
call fitbegin(
  inproc,
  initer,
  ifitmode(inproc),
  inener(inproc),
  iarrlen,
  dnev(iarrlen),
  decm(iarrlen),
  derre2(iarrlen),
  dsige2(iarrlen),
  dlum(iarrlen),
  derrlum(iarrlen),
  dlivtim(iarrlen),
  deff(iarrlen),
  derreff(iarrlen),
  derrnevup(iarrlen),
  derrnevdn(iarrlen),
  demaxrg(inproc),
  cminfile,
  chisfile )
```

Здесь переменные имеют следующее значение:

- `inproc` - суммарное количество J всех компонент.
- `initer` - задает период вычисления РП и производных от сечения для всех компонент. Пусть T - положительное число, тогда, если `initer = T`, то РП и производные вычисляются на каждом T -том

шаге минимизации. Если `initer = -T`, то производные вычисляются на каждом T -том шаге минимизации, РП не вычисляются. Если `initer = 0`, то ни РП, ни производные не вычисляются. На практике используются в-основном следующие значения:

- <10000 вычислить производные на первом шаге минимизации без РП;
- 1 вычислять производные на каждом шаге без РП;
- 0 не вычислять РП и производные;
- 1 вычислять РП и производные от сечения на каждом шаге минимизации;
- >10000 вычислить РП и производные только на первом шаге.

Примечание 1: *этот параметр позволяет отключать вычисление поправок для всех компонент одновременно. Если вам необходимо отключить поправки только для определенной компоненты, то используйте массив настроек минимизации `ifitmode`.*

Примечание 2: *Режим с вычислением производных с/без РП только на первом шаге позволяет проводить быструю минимизацию, так как наиболее трудоемкая часть вычислений выполняется только один раз. Однако, результат такой минимизации является некоторым приближением к искомому значению $\vec{\alpha}_{opt}$, и более того, ошибки параметров, скорее всего, будут определены неправильно. Тем не менее, данный режим оказывается полезен для быстрого приближения к окрестности минимума. Обычно поступают следующим образом: проводится быстрая минимизация из некоторой начальной точки $\vec{\alpha}_0$, находится оценка минимума в точке $\vec{\alpha}_1$. Затем проводится быстрая минимизация уже из начальной точки $\vec{\alpha}_1$, и так далее. Спуск в окрестность минимума занимает несколько (~ 5) итераций. Окончательное уточнение минимума и определение ошибок параметров рекомендуется проводить с `initer=1` или `-1`.*

- `ifitmode()` - массив, определяющий условия аппроксимации для конкретных компонент процессов. Для компоненты j в элемент массива `ifitmode(j)` записывается число вида $10000 \cdot l_4 + 1000 \cdot l_3 + 10 \cdot l_2 + l_1$, где l_1, l_2, l_3, l_4 - целые числа.

- Число l_1 задает вид функции правдоподобия и должно принимать одно из следующих значений:
 - 0 данные процесса не участвуют в аппроксимации;
 - 1 асимметричный χ^2 , ошибки числа событий должны быть указаны в массивах `derrnevup` и `derrnevdn` (см. ниже);
 - 3 пуассоновская функция правдоподобия, вида (14), подходит для процессов с малой статистикой;
 - 4 гауссова функция правдоподобия (17), подходит для процессов с большой статистикой;
 - 5 автоматический выбор между 3 и 4, по числу наблюдаемых событий в измерении. Граничное значение на число событий – 20;

Примечание: *методы 3,4,5 игнорируют значения, передаваемые в массивах `derrnevup` и `derrnevdn`.*

- Число l_2 предназначено для объединения компонент в процессы и может принимать значения 0..99. Для объединения необходимо, чтобы как минимум 2 компоненты имели это число одинаковым (не нулевым). Экспериментальные данные для одной энергетической точки у разных компонент одного процесса (см. ниже описание массивов данных) должны совпадать – это своеобразная проверка того, что пользователь понимает, что делает.
- С помощью значения l_3 можно отключить вычисление РП/производных для конкретной компоненты. Позволено использовать следующие значения:
 - 0 - стандартное вычисление РП и производных в соответствии со значением `initer`;
 - 1 - не вычислять РП, производные вычисляются;
 - 2 - не вычислять ни РП, ни производные.

– Число l_4 позволяет задать для конкретной компоненты режим использования функциональной зависимости эффективности от энергии. Возможные значения l_4 :

0 - стандартный режим. Значения и ошибки эффективности берутся из массивов `deff()` и `derreff()`.

1 - режим использования функциональной зависимости эффективности с определением оптимальных параметров этой зависимости. Пользователь должен написать функцию:

```
double precision function crseff(iproc,dener),
```

где `iproc` - индекс компоненты, `dener` - значение энергии в системе центра масс, и поместить ее во входной FORTRAN файл. Кроме того, в массивах `deff()` и `derreff()` должны быть переданы измеренные значения и ошибки эффективности данной компоненты. Эти значения будут использованы в добавке вида (25) к функции правдоподобия.

3 - то же самое, что и для $l_4 = 1$, но без добавки к функции правдоподобия.

- `inener()` – массив с количеством энергетических точек по каждой компоненте I_j , компоненты одного процесса должны иметь одинаковые значения;

- `iarrlen` – длина банков данных $R = \sum_{s=1}^S \sum_{p_s=1}^{P_s} I_{p_s} \equiv \sum_{j=1}^J I_j$. Сумма всех элементов массива `inener` должна равняться `iarrlen`.

- `dnev()`, `decn()`, `derre2()`, `dsige2()`, `dlum()`, `derrlum()`, `dlivtim()`, `deff()`, `derreff()` – банки данных по компонентам, содержащие соответственно N_r , E_r , ΔE_r , δE_r , L_r , ΔL_r , T_r , ε_r , $\Delta \varepsilon_r$. Длина каждого из этих массивов должна быть `iarrlen`. Данные располагаются в банках подряд;

- `derrnevup()`, `derrnevdn()` – банки данных по компонентам, содержащие верхнюю и нижнюю ошибки числа событий. Используются только если для конкретной компоненты задан способ аппроксимации асимметричным χ^2 ($l_1 = 1$ в соответствующем элементе массива `ifitmode`).
- `demaxrg()` – Массив величин E_{max} , ограничивающих интеграл (4), для каждой компоненты.
- `cminfile` – имя файла, содержащего команды MINUIT, без расширения. Для работы программы надо создать файл `<cminfile>.dat` на диске, и поместить в него определения параметров, настройки MINUIT, и команды на минимизацию. Пример содержимого файла будет приведен ниже.
- `chisfile` – имя выходных файлов, без расширения. В результате работы программы, кроме выдачи результатов в листинг, будут созданы файлы `<chisfile>_exp.his` и `<chisfile>_thr.his`, содержащие соответственно экспериментальные данные и протабулированное теоретическое сечение. Если имя пустое (' '), то файлы записаны не будут. Отключение записи выходных файлов особенно полезно в режиме быстрого поиска минимума, когда радиоправки/производные вычисляются только на первом шаге минимизации.

Для доступа к общим областям необходимо поместить в описательную часть программы следующую строку:

```
include 'fitdata1.inc'
```

При необходимости воспользоваться дополнительными возможностями программы, надо перед вызовом `fitbegin` заполнить соответствующие переменные общих областей:

- `errparmin0(ipar)` – массив, задающий ошибки минимизируемых параметров. По умолчанию этот массив заполнен нулями. Если занести в `errparmin0(k)` значение $\Delta\alpha_{k,0} > 0$, а в соответствующий элемент `parfix0(k)` значение $\alpha_{k,0}$, то параметр α_k будет привязан к значению $\alpha_{k,0}$ с ошибкой $\Delta\alpha_{k,0}$. Если же $\Delta\alpha_{k,0} < 0$, то данный параметр будет ограничен снизу значением 0, т.е. будет принимать строго не отрицательные значения. Величина $|\Delta\alpha_{k,0}|$ в обоих случаях является привязкой.
- `parfix0(ipar)` – массив значений привязки параметров, используется только если `errparmin0(k) > 0`.
- `parlimtype(ipar)`, `parlim0(2, ipar)`, `parlimerr0(2, ipar)` – массивы обеспечивающие ограничение параметров. Элементы массива `parlimtype` по умолчанию заполнены нулями, то есть ограничений на параметры не накладывается, если же `parlimtype(k) = 2` – то на параметр с индексом k накладывается ограничение, причем, в `parlim0(1, k)`, `parlimerr0(1, k)` – помещаются значения нижней границы и привязки к ней, в `parlim0(2, k)` и `parlimerr(2, k)` – верхней. Если привязка к какой-либо из границ занулена, то параметр с этой стороны ограничен не будет.
- `thanxtoandy` – логическая переменная, задает метод интегрирования радипоправок. Если `.false.`, то применяется стандартный метод вычисления интеграла (4), используя адаптивный метод интегрирования программой SIMPS1. Иначе, применяется разделение всего интервала интегрирования фиксированными точками. Для большинства процессов оба метода дают практически одинаковый результат. Иногда, при появлении плохой сходимости из-за того, что величина радипоправок, вычисленная адаптивным методом, сильно из-

меняется, может быть полезно переключиться на `.true..`

- `radcortype` – способ вычисления радпоправок. Поддерживаются следующие методы вычислений:

3 по формулам из статьи [3], без рождения реальных e^+e^- -пар;

4 по формулам из статьи [3], с рождением реальных e^+e^- -пар.

Если `radcortype < 0`, то метод вычисления радпоправок будет `|radcortype|`, с отключенным вкладом поляризации вакуума. Значение `radcortype = -3` берется по умолчанию, т.е. метод вычисления – работа [3], без рождения реальных e^+e^- -пар и без вклада поляризации вакуума.

В программах вычисления сечения необходимо на каждом шаге минимизации посчитать сечение с текущими значениями параметров минимизации. Текущие значения параметров минимизации находятся в общей области `parmin(ipar)`.

Результаты минимизации будут выведены в выходной файл, см. ниже. Кроме этого, пользователь может получить некоторую информацию в следующих общих областях:

- `minparmin(ipar)` - наилучшие значения параметров с начала задания;

- `minchi2` - наилучшее (минимальное) значение логарифмической функции правдоподобия, полученное при этих параметрах;

- `parcname(ipar)` - название параметра (`character*10`);

- `parvalue(ipar)` - текущее значение параметров;

- `paremat(ipar,ipar)` - текущее значение матрицы ошибок параметров;
- `pareplus(ipar)` - текущие значения верхних ошибок параметров, вычисленные `minos`;
- `pareminus(ipar)` - текущие значения нижних ошибок параметров, вычисленные `minos`;
- `pareparab(ipar)` - текущие значения параболических ошибок параметров;
- `pareglobcc(ipar)` - текущее значение глобального коэффициента корреляции данного параметра. Это коэффициент корреляции между данным параметром и линейной комбинацией всех остальных, такой, чтобы корреляция была максимальна;
- `curchi2` - текущее значение логарифмической функции правдоподобия;

Перечисленные общие области, за исключением `minparmin` и `minchi2`, заполняются при вызове функции `fcn_snd` с флагом 3. В частности, такой вызов делается в конце задания. Но пользователь, если желает, может заставить MINUIT сделать такой вызов, указав во входном файле MINUIT команду `call 3`.

Кроме того, при вызове с флагом 3, после заполнения указанных общих областей производится вызов подпрограммы `fitfinish`. Вариант этой подпрограммы по умолчанию не делает ничего. Однако, пользователь может заменить эту подпрограмму своей, в которой производятся какие-либо полезные действия. Например, выводятся значения параметров и их ошибок, полученные соответственно из общих областей `parvalue` и `pareparab`, с более высокой точностью, чем это делает `minuit`.

13 Типичный вид задания

В приведенном ниже задании проводится аппроксимация экспериментальных данных сечением процесса $e^+e^- \rightarrow \eta\gamma$. Для вычисления сечения используется подпрограмма `cretag` из библиотеки стандартных сечений `RADCOR`. Параметрами минимизации являются: масса ϕ мезона, ширина ϕ мезона и относительная вероятность перехода $\phi \rightarrow \eta\gamma$. На самом деле вместо массы ϕ мезона аппроксимируется значение массы минус 1000 MeV. Такая замена сделана намеренно, чтобы не потерять точность при выводе результатов программой `MINUIT`, которая делает это с точностью 6 значащих цифр.

```
      program test01
      implicit none
c --- this file is needed here to get access to some FIT variables
      include 'fitdata1.inc'
c In this example:
c approximation of Phi -> EtaGamma with fitting method: 5
c 3 parameters:
c   MsPhi-1000. (MeV)
c   GPhi      (MeV)
c   BrPhi2Eta
c initial values are defined in "test.dat"

c ===== fitbegin data =====
c --- a number of components
      integer inproc
      parameter (inproc = 1)
c --- auxiliary parameters: number of points per component
      integer in1
      parameter (in1 = 10)
c --- a period of radcor calculations, usual values:
c   < -10000 derivatives at the first pass only, no radcor
c   -1      derivatives at every step, no radcor
c   0       no radcor, no derivatives
c   1       radcor and derivatives at every step
c   > 10000 radcor and derivatives at the first pass
      integer initer
      parameter (initer = 1)
c --- per-component settings: (fitmode)+10*(superidx)+1000*(radtype)
```

```

c fitmode, now available:
c 0 - Process is not considered
c 1 - Asymmetric Chi2 (derrnevup/derrnevdn should be supplied)
c 2 - Pseudo-Poisson (disabled)
c 3 - Poisson (for small stat)
c 4 - Symmetric Chi2 (for high stat)
c 5 - Autoselection
c superidx - can be 0...99
c radtype, is one of the following:
c 0 - usual operation, according initer
c 1 - no radcor, only derivatives
c 2 - no radcor, no derivatives
integer ifitmode(inproc)
data ifitmode /5/
c --- number of points for each component
integer inener(inproc)
data inener /in1/
c --- the length of data array
integer iarrlen
parameter (iarrlen = in1)
c --- registered number of events
real*8 dnev(iarrlen)
data dnev / 2, 11, 60, 135, 362, 262, 77, 67, 15, 6 /
c --- energy in c.m.s., it will be filled from debeam array
real*8 decm(iarrlen)
c --- beam energies
real*8 debeam(iarrlen)
data debeam / 502.502, 505.432, 508.309, 509.291, 509.764, 510.246,
510.714, 511.181, 514.084, 516.974/
c --- error of decm, here it considered to be 50 keV
real*8 derre2(iarrlen)
data derre2 / iarrlen * 0.05 /
c --- spread of decm
real*8 dsige2(iarrlen)
data dsige2 / 0.250, 0.262, 0.265, 0.249, 0.271, 0.247, 0.267,
0.266, 0.253, 0.259/
c --- integrated luminosity and its error
real*8 dlum(iarrlen)
data dlum / 21.73, 29.68, 37.29, 38.89, 90.28, 83.24,
35.49, 38.20, 19.19, 21.79/
real*8 derrlum(iarrlen)
data derrlum / 0.09, 0.10, 0.11, 0.13, 0.18, 0.17, 0.11, 0.12,
0.08, 0.09/
c --- live time of measurement
real*8 dlivtim(iarrlen)

```

```

      data dlivtim / 9.0463d+4, 4.964d+4, 6.8389d+4, 6.9183d+4,
      & 2.13343d+5, 1.37486d+5, 5.4245d+4, 6.2844d+4, 3.2947d+4, 4.3345d+4/
c --- efficiency and its error (obtained from MC simulation)
      real*8 deff(iarrlen) /10*0.0893/
      real*8 derreff(iarrlen) /10*0.0012/
c --- arrays for user supplied errors of number of events
c   they are actually not used in this program
      real*8 derrnevup(iarrlen), derrnevdn(iarrlen)
c --- radcor cut-off energy, E_max, should be treated carefully
      real*8 demaxrg(inproc) /300./
c --- filename of minuit input command file
      character*(*) cminfile
      parameter ( cminfile = 'test' )
c --- filename of output files
      character*(*) chisfile
      parameter ( chisfile = 'test' )
c ===== end of fitbegin data

c --- Local variables
      integer ie

c --- call to inifit, initialization of auxiliary library RADCOR
      call inifit
c --- let us restrict width and branching (parameter 2 and 3)
c   in positive values
      errparmin0(2) = -0.01
      errparmin0(3) = -0.01
c --- Twice Ebeam to get Ecm
      do ie = 1, iarrlen
         decm(ie) = debeam(ie)*2.d0
      enddo
c --- The main call
      call fitbegin (inproc, initer, ifitmode, inener, iarrlen, dnev,
      & decm, derre2, dsige2,
      & dlum, derrlum, dlivtim, deff, derreff, derrnevup,
      & derrnevdn, demaxrg, cminfile, chisfile)
      end

c-----
c A subroutine to calculate a theoretical cross section.
c Should be written by user.
c Input parameters:
c   iproc - component index,
c   dener - energy in c.m.s.
c   parmin() - values of parameters, are transferred through common block.
c Output parameters:

```



```

c  d_crspr - calculated cross section
c
      subroutine crs_proc(iproc,dener,d_crspr)
      implicit none
c --- fitdata1.inc is needed to access current FIT parameters values, parmin()
      include 'fitdata1.inc'
c --- fitdata.inc is needed here to feed those parameters into standard library
c   for cross section calculation -- RADCOR
      include 'fitdata.inc'

      integer iproc
      real*8 dener, d_crspr
c --- a function from RADCOR to calculate cross section of Eta Gamma
      real*8 cretag

c --- parameters are transferred to RADCOR
      msphi   = parmin(1) + 1000.0d0
      gphi    = parmin(2)
      brphietag = parmin(3)
      if ( iproc.eq.1 ) then
         d_crspr = cretag(dener)
      else
         write (6,*) 'crs_proc general failure'
         stop
      endif
      end
end

c-----
c A subroutine to calculate background cross section.
c It has less use nowadays, so it just fills output with zeroes.
c Input parameters:
c  iproc - component index
c  dener - energy in c.m.s.
c  parmin() - minimization parameters, can be accessed through common block
c Output parameters:
c  d_crsbgl - non-resonant background cross section
c  d_crsbgt - cosmic background frequency
      subroutine crs_bgrd( iproc, dener, d_crsbgl, d_crsbgt )
      implicit none
      integer iproc
      real*8 dener, d_crsbgl, d_crsbgt

      if ( iproc.ne.1 ) then
         write (6,*) 'crs_bgrd unknown process id'
         stop
      endif
endif

```

```
d_crsgl=0.  
d_crsgt=0.  
end
```

Помимо этой программы необходимо создать файл с командами MINUIT. Имя файла указывается при вызове `fitbegin`. Ниже приводится пример файла для этого задания:

```
SET TITLE  
An example of Phi->EgaGamma cross section approximation  
PARAMETERS  
1 'msphi'      19.4  0.01  
2 'gphi'       4.4   0.01  
3 'brphietag' 0.013 0.01  
  
set err 0.5  
set eps 1e-8  
mini  
minos  
ret
```

Ниже для справки приведен список наиболее часто используемых команд MINUIT, некоторые из них были использованы в данном примере:

- `CONTOUR <par1> <par2>` - рисует контурные линии уровня функции правдоподобия в плоскости параметров `par1` и `par2`;
- `FIX <par>` - фиксирует величину параметра `<par>` на ее текущем значении;
- `HELP [SET|SHOW|<comm>]` - выдает список возможных команд MINUIT. Если дан аргумент, выдает краткую справку по указанной команде;
- `MINIMIZE [maxcalls]` - минимизирует заданную функцию. Параметр `maxcalls` ограничивает количество вызовов;
- `MINOS [maxcalls] [par]` - определяет асимметричные ошибки параметра `par`;

- `RELEASE <par>` - освобождает параметр `par`, ранее фиксированный командой `FIX`;
- `SCAN <par> <npts> <from> <to>` - сканирует значения функции правдоподобия по параметру `par`;
- `SET EPS <value>` - ограничение точности вычислений.

Такое ограничение необходимо, так как в противном случае функция правдоподобия имеет "скачущий" рельеф, что сильно сказывается на сходимости метода MIGRAD. Значение `<value>` задает масштаб изменений рельефа, при котором значения функции считаются одинаковыми. Вообще говоря, величину неровностей рельефа определяют, в основном, функции, считающие сечение и радиоправки. Эта величина выбирается опытным путем, хотя в большинстве случаев можно использовать значение 10^{-8} .

- `SET PAR <number> <value>` - присвоить параметру `<number>` значение `<value>`;
- `SET ERR <value>` - установить уровень определения ошибок. Так как в пакете `fit` используется логарифмическая функция правдоподобия, то уровень 0.5 соответствует одному стандартному отклонению. *Примечание: если не поместить в задание строку `set err 0.5`, то ошибки будут определяться на уровне 1;*
- `SET TITLE <title>` - определить название задания, которое будет выводиться в таблицах `MINUIT`;
- `PARAMETERS` - строка, начинающая блок определения параметров. Обычно этот блок располагается в начале файла. Определение параметра состоит из одной строки в этом блоке, в которой перечислены: номер параметра, название параметра, начальное значение параметра, начальный шаг параметра. Если пользователь желает ограничить параметр

средствами MINUIT, то далее в строке должны следовать нижняя и верхняя граница области. Если начальный шаг равен нулю, то параметр будет константой. Блок определения параметров заканчивается пустой строкой.

14 Запуск задания

Так как задание представляет собой обычную FORTRAN программу, то достаточно скомпилировать ее и собрать с необходимым набором библиотек `fit`, `radcor`, `CERNLIB`. Эти операции можно проделать вручную. Но для удобства пользователей и для возможности использования системы пакетных заданий, были написаны две запускающие процедуры. Первая (FIT) предназначена для запуска задания в интерактивном режиме, вторая (BATF) – в пакетном. Синтаксис вызова этих процедур следующий:

```
FIT <имя задания> [опции1]
```

```
BATF <имя задания> [<имя листинга>|-|. [опции2]]
```

где <имя задания> – имя входного FORTRAN файла (расширение `.for`), <имя листинга> – имя выходного файла (расширение `.lif`), если не указано, или вместо имени указано ‘-’ или ‘.’, то будет использовано имя входного файла. [опции] – список опций, перечисленных через запятую. В вызове FIT позволено использовать следующие опции: `debug` - режим отладки задания, `norun` - скомпилировать задание, но не исполнять. В вызове BATF позволено использовать следующие опции: `class <имя очереди>` - задает очередь для исполнения задания. В кластере СНД существуют очереди `short`, `medium` и `long`.

На поведение этих процедур при компиляции и сборке могут повлиять следующие переменные среды:

- `SNDROOT` - имя директории с установленными ресурсами. Должна быть определена для нормальной работы процедуры;
- `FITMAKE` - полное имя сценария сборки задания, вместо стандартного
`$$SNDROOT/fit.mk`;
- `FITDEPS` - дополнительные зависимости, т.е. имена ресурсов, которые необходимо использовать при сборке задания. Стандартный набор ресурсов состоит из следующих элементов:
`'snd-main/fortran.mk radcor/fitdata.inc
fit/fitdata1.inc fit/libfit.a radcor/libradcor.a
cocha/kio/libkio.a'`;
- `FITLIBS` - добавочный список библиотек при сборке задания;
- `FITINCS` - добавочный список включаемых файлов при компиляции задания;
- `SNDHOME` - имя директории пользователя, в которой он размещает свои локальные пакеты. Локальные пакеты могут понадобиться пользователям, которые ведут собственные разработки/модификацию тех ресурсов, которые необходимы заданию для сборки;
- `SNDLOCALS` - список имен локальных пакетов пользователя. Если этот список содержит имена некоторых пакетов и для сборки задания необходимы ресурсы из этих пакетов, то эти ресурсы будут браться из локальных директорий пользователя. Например, если `SNDLOCALS=radcor`, то для сборки задания, в частности, будут использованы включаемый файл `$$SNDHOME/radcor/fitdata.inc` и библиотека `$$SNDHOME/radcor/libradcor.a` вместо стандартных `$$SNDROOT/include/fitdata.inc` и `$$SNDHOME/lib/libradcor.a`.

15 Представление результатов

Во время исполнения задания создается файл с выводом задания. В конце задания в этот файл выводятся результаты аппроксимации. Ниже приведена часть выходного файла вышеприведенного примера:

```
[skipped]
FBEGIN 1999/11/24 05:33 by VI,ZS,AB,DB Cross section fit
FBEGIN number of processes          1
FBEGIN period of radcor/derivatives calculation is          1
FBEGIN process # 1 cut-off energy  300.00
FBEGIN process # 1 autoselection between poisson and gauss
[skipped]
FCNSWD initial parameters values:
# 2 (gphi      ) is positive with fix 0.100000E-01
# 3 (brphietag) is positive with fix 0.100000E-01
[skipped]
FCM= 11.42573 FROM MIGRAD STATUS=CONVERGED 64 CALLS 66 TOTAL
      EDM= 0.18E-06 STRATEGY= 1 ERROR MATRIX ACCURATE

EXT PARAMETER          STEP          FIRST
NO.  NAME      VALUE      ERROR      SIZE      DERIVATIVE
  1  msphi      19.560      0.14706      0.15648E-01  -0.63385E-03
  2  gphi        4.6844      0.43182      0.26909E-01  -0.31322E-03
  3  brphietag  0.13770E-01  0.77441E-03  0.45219E-04  -0.71200
      ERR DEF= 0.500

EXTERNAL ERROR MATRIX.  NDIM= 50  NPAR= 3  ERR DEF= 0.500
0.216E-01 0.191E-01-0.508E-04
0.191E-01 0.186E+00-0.257E-03
-0.508E-04-0.257E-03 0.600E-06

PARAMETER CORRELATION COEFFICIENTS
NO.  GLOBAL  1  2  3
  1  0.45141  1.000 0.300-0.446
  2  0.77053  0.300 1.000-0.769
  3  0.80153 -0.446-0.769 1.000
*****
**  4 **MINOS
*****

MINUIT TASK: An example of Phi->EgaGamma cross section approxim
```

FCW= 11.42573 FROM MINOS STATUS=SUCCESSFUL 168 CALLS 234 TOTAL
 EDM= 0.18E-06 STRATEGY= 1 ERROR MATRIX ACCURATE

EXT NO.	PARAMETER NAME	VALUE	PARABOLIC	MINOS ERRORS	
			ERROR	NEGATIVE	POSITIVE
1	msphi	19.560	0.14706	-0.14431	0.14928
2	gphi	4.6844	0.43182	-0.40575	0.45447
3	brphietag	0.13770E-01	0.77441E-03	-0.75270E-03	0.78393E-03

ERR DEF= 0.500

** 5 **RET

CALL TO USER FUNCTION WITH IFLAG = 3

.....MINUIT TERMINATED AND RETURNS TO USER PROGRAM.

Number of fitting runs = 236

N	NegLog	Chi2
1	11.4	5.7

Table 1. Fitted cross-section

Total arrays len. = 10 Neglog.likelihood = 11.43 Chi2 = 5.73

	NP	ME	Ecm	ErrE	CrExp	ECrU	ECrD	CRE	CR	C	Cexp-CRE	dChi**2
			Mev	Mev	nb	nb	nb	nb	nb	nb	EcrU	
101	1	1005.00	50.m	92.039m	.1214	59.5m	.152544	.000000	.000000	-.4983	.2483	
101	2	1010.86	50.m	.370620	.1489	.1102	.348787	.000000	.000000	.1466	21.5m	
101	3	1016.62	50.m	1.60901	.2394	.2113	1.74659	.000000	.000000	-.5748	.3304	
101	4	1018.58	50.m	3.47133	.3316	.3054	3.52958	.000000	.000000	-.1756	30.8m	
101	5	1019.53	50.m	4.00975	.2284	.2173	3.86564	.000000	.000000	.6310	.3982	
101	6	1020.49	50.m	3.14753	.2168	.2050	3.25562	.000000	.000000	-.4987	.2487	
101	7	1021.43	50.m	2.16963	.2818	.2526	2.37851	.000000	.000000	-.7413	.5495	
101	8	1022.36	50.m	1.75393	.2445	.2171	1.69821	.000000	.000000	.2279	52.0m	
101	9	1028.17	50.m	.781657	.2585	.1997	.415250	.000000	.000000	1.417	2.009	
101	10	1033.95	50.m	.275356	.1645	.1093	.204899	.000000	.000000	.4283	.1834	
1	1	1005.00	50.m	92.039m	.1214	59.5m	.152544	1.70822	2.10650	-.4983	.2483	
1	2	1010.86	50.m	.370620	.1489	.1102	.348787	3.90578	5.03570	.1466	21.5m	
1	3	1016.62	50.m	1.60901	.2394	.2113	1.74659	19.5587	26.6996	-.5748	.3304	
1	4	1018.58	50.m	3.47133	.3316	.3054	3.52958	39.5250	54.6298	-.1756	30.8m	
1	5	1019.53	50.m	4.00975	.2284	.2173	3.86564	43.2882	58.2959	.6310	.3982	
1	6	1020.49	50.m	3.14753	.2168	.2050	3.25562	36.4571	45.6251	-.4987	.2487	
1	7	1021.43	50.m	2.16963	.2818	.2526	2.37851	26.6351	30.5928	-.7413	.5495	
1	8	1022.36	50.m	1.75393	.2445	.2171	1.69821	19.0169	20.0423	.2279	52.0m	
1	9	1028.17	50.m	.781657	.2585	.1997	.415250	4.65005	3.02063	1.417	2.009	
1	10	1033.95	50.m	.275356	.1645	.1093	.204899	2.29450	.989465	.4283	.1834	

[skipped]

В этом файле сначала располагается подтверждение полученных данных – количество компонент, настройки вычисления РП, условия, накладываемые на параметры минимизации. После выполнения команды `MINI`, `MINUIT` выводит стандартную таблицу со статусом минимизации, значениями параметров и их ошибок в точке минимума, а также матрицу ошибок и корреляционную матрицу. После выполнения `MINOS` выводятся асимметричные ошибки параметров, которые в некоторых случаях могут сильно отличаться от симметричных ошибок, найденных `MINI`. В конце работы выводится количество вызовов минимизируемой функции, а также значение этой функции и χ^2 в минимуме для каждой компоненты. Затем следует таблица в которой сравниваются экспериментальные данные и значение аппроксимирующей функций.

В строке таблицы перечислены:

- индекс компоненты или процесса. Если этот индекс < 100 , то это компонента, если > 100 – процесс. Так как отдельный процесс является и компонентой и процессом, то для него в таблице будет две строки (одинаковые);
- номер энергетической точки в этой компоненте;
- точность энергии;
- величина экспериментально наблюдаемого сечения (1);
- величина верхней ошибки этого сечения;
- величина нижней ошибки этого сечения;
- величина вычисленного аппроксимирующего сечения с РП, домноженного на эффективность;
- величина вычисленного аппроксимирующего сечения с РП. Для процесса невозможно определить эту величину, поэтому здесь будет помещен 0;
- величина вычисленного аппроксимирующего сечения (борновского);

- разность экспериментального и теоретического сечения, выраженная в единицах верхней ошибки экспериментального сечения;
- вклад в χ^2 от данной точки.

Две последние колонки помещены в эту таблицу для удобства поиска точек, которые дают наибольший вклад в χ^2 . Следом выводится таблица(ы), по которым можно проверить правильность введенных экспериментальных величин. Так как эти таблицы не предназначены для считывания программными средствами, а исключительно для просмотра, то в них для построения более ровных колонок (помещающихся на экран) при выводе чисел используются распространенные обозначения для кратных и дольных величин, то есть: р-пико, и-микро, м-мили, к-кило, М-мега, и т.д.

Следующий пример показывает аппроксимацию данных в случае нескольких процессов. Экспериментальные данные, использованные в этом задании, являются результатом обработки процесса $e^+e^- \rightarrow K_S K_L$ на детекторе КМД [10], и были предоставлены нам командой этого детектора. В задании участвуют данные 4 сканирований: 94, 96-2, 96-3, 96-4.

```

c+++ KsKl@CMD approximation
      program run0
      implicit none
      include 'fitdata1.inc'

c === fitbegin data
      integer inproc                                ! number of processes
      parameter ( inproc = 4 )
      integer in1, in2, in3, in4                    ! number of points per scan
      parameter ( in1 = 9, in2 = 12, in3 = 12, in4 = 11 )
      integer initer                                 ! a period of RADCOR calculation
      parameter ( initer = 1 )
      integer ifitmode(inproc) /1,1,1,1/          ! fit setting
      integer inener(inproc) /in1,in2,in3,in4/    ! number of points
      integer iarrlen                                ! total length of dataarrays
      parameter ( iarrlen = in1+in2+in3+in4 )
      real*8 dnev(iarrlen) /                        ! number of events

```

```

& 135, 8487, 4697, 5347, 7610, 9372, 3725, 1900, 2645, !94
& 77, 216, 2265, 6552, 8214, 15755, 18403, 10855, 2800, 862, 492, 285, !9602
& 39, 640, 4570, 7541, 17836, 27070, 25681, 18433, 8685, 3337, 845, 487, !9603
& 46, 382, 2478, 6735, 14703, 21322, 22823, 8800, 6237, 4072, 961 / !9604
real*8 decm(iarrlen) / ! e.c.m. energy
& 1010.272, 1017.086, 1018.136, 1018.956, 1019.214, 1019.986, 1020.128,
& 1021.850, 1023.972,
& 1004.25, 1010.53, 1016.36, 1017.19, 1018.06, 1018.99, 1020.00, 1020.96,
& 1021.88, 1027.69, 1033.63, 1039.48,
& 1004.64, 1011.30, 1015.99, 1016.93, 1017.91, 1019.04, 1019.95, 1020.86,
& 1021.74, 1022.67, 1028.36, 1034.06,
& 1004.19, 1011.30, 1015.91, 1016.94, 1017.92, 1018.76, 1019.68, 1020.68,
& 1021.60, 1022.59, 1028.41/
real*8 derre2(iarrlen) / ! energy error
& .03, .02, .02, .02, .02, .02, .02, .02, .02,
& .36, .11, .04, .03, .03, .02, .02, .02, .05, .09, .36, .36,
& .36, .06, .03, .02, .02, .02, .02, .02, .03, .05, .09, .36,
& .36, .07, .03, .02, .02, .02, .02, .03, .03, .04, .07 /
real*8 dsige2(iarrlen) /iarrlen*.18/ ! energy spread
real*8 dlum(iarrlen) / ! luminosity
& 16.98, 80.50, 27.68, 25.59, 27.23, 35.44, 16.00, 17.34, 48.16,
& 31.23, 21.62, 29.13, 55.72, 47.04, 63.79, 71.46, 52.13, 19.71, 21.33,
& 28.76, 22.89,
& 42.60, 55.15, 61.59, 67.59, 100.02, 103.52, 98.95, 91.79, 58.36, 30.51,
& 25.13, 28.44,
& 25.08, 32.12, 33.16, 61.84, 79.37, 84.61, 86.89, 40.75, 41.57, 38.15, 30.77/
real*8 derrlum(iarrlen) / ! luminosity error
& .25, .80, .36, .34, 1.23, .39, .20, .27, .45,
& .30, .27, .33, .40, .36, .42, .45, .38, .24, .29, .29, .26,
& .45, .90, .41, .43, .52, .53, .52, .51, .41, .35, .27, .28,
& .26, .30, .30, .41, .47, .48, .49, .34, .34, .32, .31/
real*8 dlivtim(iarrlen) /iarrlen*0./ ! live time
real*8 deff(iarrlen) / ! efficiency
& .2524, .2357, .2581, .2182, .2907, .2626, .2643, .2080, .2648,
& .2848, .2796, .2731, .2729, .2710, .2651, .2752, .2844, .2721, .2518,
& .2309, .2271,
& .2770, .2734, .2733, .2724, .2705, .2690, .2673, .2661, .2642, .2622,
& .2516, .2395,
& .2791, .2567, .2662, .2538, .2625, .2724, .2622, .2601, .2557, .2543,
& .2446/
real*8 derreff(iarrlen) / ! efficiency error
& .1760E-01, .1404E-01, .3103E-01, .1241E-01, .7192E-02, .1257E-01,
& .2690E-01, .2249E-01, .4518E-01,
& .2226E-01, .1415E-01, .1395E-01, .2494E-01, .7607E-02, .5601E-02,
& .4541E-02, .5765E-02, .1516E-01, .6727E-02, .7860E-02, .1022E-01,

```

```

& .2307E-01, .7670E-02, .3698E-02, .3221E-02, .2741E-02, .2647E-02,
& .2659E-02, .2789E-02, .3141E-02, .3931E-02, .6579E-02, .8064E-02,
& .1999E-01, .2372E-01, .1566E-01, .1262E-01, .7623E-02, .4827E-02,
& .4477E-02, .7962E-02, .9425E-02, .1007E-01, .2008E-01/
real*8 derrnev(iarrlen) /          ! error of number of events
& 13, 95, 78, 74, 89, 103, 62, 46, 90,
& 37, 19, 50, 86, 94, 127, 137, 108, 55, 70, 84, 19,
& 28, 45, 86, 106, 164, 190, 183, 158, 110, 69, 44, 41,
& 10, 37, 61, 95, 140, 163, 167, 105, 91, 74, 45/
real*8 demaxrg(4) /inproc*200./ ! radcor cut-off
character*(*) cminfile          ! a file with minuit command (input)
parameter (cminfile = 'testcmd')
character*(*) chisfile          ! an output filename
parameter (chisfile = 'testcmd')
c === end of fitbegin data

c --- initialization of RADCOR library
call inifit
c --- use of fixed nodes for rad.cor. integration
thanxtoandy = .true.
c --- limit some parameters to positive values
errparmin0(1) = -0.1          ! cross section at phi for scan94
errparmin0(2) = 0.           ! phi mass for scan94
errparmin0(3) = -0.1          ! phi width (common)
errparmin0(4) = -0.1          ! background cross section (common)
errparmin0(5) = -0.1          ! cross section at phi for scan9602
errparmin0(6) = 0.           ! phi mass for scan9602
errparmin0(7) = -0.1          ! cross section at phi for scan9603
errparmin0(8) = 0.           ! phi mass for scan9603
errparmin0(9) = -0.1          ! cross section at phi for scan9604
errparmin0(10) = 0.          ! phi mass for scan9604
c --- main call
call fitbegin(inproc, initer, ifitmode, inener, iarrlen, dnev, decm,
& derre2, dsige2, dlum, derrlum, dlivtim, deff, derreff, derrnev,
& derrnev, demaxrg, cminfile, chisfile )
end

c --- cross section of ee->KsKl
c use standard routine from RADCOR library
subroutine crs_proc( iproc, dener, d_crs )
implicit none
include 'fitdata.inc'
include 'fitdata1.inc'
integer iproc
real*8 dener, d_crs

```

```

      real*8 crkskl ! routine from RADCOR library
      real*8 h
c --- transfer parameters into RADCOR library common block
      if ( iproc.eq.1 ) then
          h = parmin(1)
          msphi = parmin(2) + 1000.0d0
      else if ( iproc.eq.2 ) then
          h = parmin(5)
          msphi = parmin(6) + 1000.0d0
      else if ( iproc.eq.3 ) then
          h = parmin(7)
          msphi = parmin(8) + 1000.0d0
      else if ( iproc.eq.4 ) then
          h = parmin(9)
          msphi = parmin(10) + 1000.0d0
      else
          write (6,*) 'something goes wrong'
          stop
      endif
      gphi = parmin(3)
c --- cross section
      d_crs = h * crkskl(dener)
      end

c --- background cross section
      subroutine crs_bgrd( iproc, dener, d_crl, d_crt )
      implicit none
      include 'fitdata.inc'
      include 'fitdata1.inc'
      integer iproc
      real*8 dener
      real*8 d_crl, d_crt
      d_crl = parmin(4)
      d_crt = 0.
      end

```

Файл с командами MINUIT приведен ниже:

```

SET TITLE
KsKl @ CMD
PARAMETERS
1 'h1'      1.0  0.01
2 'mphi1'  19.47 0.1
3 'gphi'   4.4  0.1
4 'sigb'   0.   0.

```

```

5 'h2'      1.0  0.01
6 'mphi2'   19.47 0.1
7 'h3'      1.0  0.01
8 'mphi3'   19.47 0.1
9 'h4'      1.0  0.01
10 'mphi4'  19.47 0.1

```

```

set err 0.5
set eps 1e-8
set print 1
mini
ret

```

Результаты совместной аппроксимации этих экспериментов приведены в следующей выдаче:

[skipped]

```

FCN= 12.19898 FROM MIGRAD STATUS=CONVERGED 276 CALLS 278 TOTAL
      EDM= 0.17E-05 STRATEGY= 1 ERROR MATRIX ACCURATE

```

EXT PARAMETER			STEP		FIRST
NO.	NAME	VALUE	ERROR	SIZE	DERIVATIVE
1	h1	0.94728	0.25484E-01	0.23776E-02	0.88183E-02
2	mphi1	19.449	0.98025E-01	0.15559E-01	-0.61647E-02
3	gphi	4.4322	0.51752E-01	0.35457E-02	0.15068E-02
4	sigb	0.0000	constant		
5	h2	0.92169	0.11384E-01	0.10502E-02	0.13079E-02
6	mphi2	19.515	0.41908E-01	0.15612E-01	-0.57061E-02
7	h3	0.94784	0.85115E-02	0.75827E-03	0.19775E-01
8	mphi3	19.415	0.21139E-01	0.15532E-01	-0.27647E-01
9	h4	0.96149	0.11668E-01	0.10492E-02	-0.31844E-02
10	mphi4	19.362	0.39635E-01	0.15490E-01	-0.23490E-01

[skipped]

N	NegLog	Chi2
1	2.9	5.7
2	3.4	6.8
3	3.7	7.4
4	2.2	4.5

[skipped]

```

Total arrays len. = 44 Neglog.likelihood = 12.20 Chi2 = 24.40

```

[skipped]

16 Краткое описание библиотеки RADCOR

Для упрощения написания программы вычисления сечения, пользователю предлагается набор программ из библиотеки RADCOR. Имена программ и переменных общих областей, используемых этой библиотекой, стандартизованы, и состояются из следующих слов:

- Ms** масса резонанса;
- G** ширина резонанса;
- Wd** парциальная ширина;
- Br** относительная вероятность процесса;
- Ph** фаза;
- Cr** сечение;
- Dr** обратный пропагатор;
- Cs** фактор, учитывающий зависимость констант связи от энергии;
- Pi, Pi0, Eta, Etap, Ks, Kl, Kc, Rho, Om, Phi** — заряженный пион, нейтральный пион, η , η' , K_S , K_L , заряженный K , ρ , ω , ϕ мезоны соответственно;
- Ee** состояние e^+e^- ;
- G** фотон

Например, масса ρ мезона находится в переменной **MsRho**, относительная вероятность перехода ω мезона в e^+e^- задается значением **BrOmEe**, ширина распада ϕ мезона в 3 пиона вычисляется функцией **WdPhi3Pi**, сечение процесса $e^+e^- \rightarrow K_S K_L$ — функцией **CrKsKl**. Все переменные и программы библиотеки вычисляются с двойной точностью. Параметры библиотеки, определяющие поведение сечения, располагаются в общих областях, доступ к которым возможен, если включить в описательную часть программы пользователя строку:

```
include 'fitdata.inc'
```

При первом обращении к любой из программ вычисления сечения, делается вызов в подпрограмму `inifit`, которая при первом вызове заполняет переменные общих областей среднемировыми значениями. Поэтому, если пользователь хочет поменять какое-либо значение, то он должен вызвать `inifit`, а затем занести значение в соответствующую переменную.

Список переменных в общих областях библиотеки с их начальными значениями (на момент выхода препринта):

<code>MsPi = 139.56995D0</code>	<code>MsPi0= 134.9764D0</code>
<code>MsEta= 547.30D0</code>	<code>MsEta=957.78D0</code>
<code>MsKs = 497.672D0</code>	<code>MsKc=493.677D0</code>
<code>MsRho= 770.0D0</code>	<code>MsOm = 781.94D0</code>
<code>MsFO = 980.0D0</code>	<code>MsAO = 983.4D0</code>
<code>MsPhi= 1019.413D0</code>	<code>MsK892 = 891.66D0</code>
<code>MsK0892 = 896.10D0</code>	<code>GRho = 150.7D0</code>
<code>G0m = 8.41D0</code>	<code>GPhi = 4.43D0</code>
<code>GK892 = 50.8D0</code>	<code>GK0892 = 50.5D0</code>
<code>PhRho = 0.D0</code>	<code>PhOm = 0.D0</code>
<code>PhPhi= 155.D0</code>	<code>PhRhop = 186.d0</code>
<code>BrRhoPiG = 4.5D-4</code>	<code>BrRhoPiOG= 6.8D-4</code>
<code>BrRhoEtaG= 2.4D-4</code>	<code>BrRhoEe = 4.49D-5</code>
<code>BrOm3Pi = 0.888D0</code>	<code>BrOmPiOG= 0.085</code>
<code>BrOmEtaG= 6.5D-4</code>	<code>BrOm2Pi = 0.0221D0</code>
<code>PhOm2Pi = 90.D0</code>	<code>BrOmEe = 7.07D-5</code>
<code>BrPhi2Kc = 0.491D0</code>	<code>BrPhiKsKl= 0.341D0</code>
<code>BrPhi3Pi = 0.155D0</code>	<code>BrPhiPiOG= 1.31D-3</code>
<code>BrPhiEtaG= 1.26D-2</code>	<code>BrPhi2Pi = 8.D-5</code>
<code>PhPhi2Pi = -20.D0</code>	<code>BrPhiEe = 2.99D-4</code>
<code>MeVnb = 3.8938D+11</code>	<code>Alpha = 1.D0/137.036D0</code>

В указанном списке переменная `MeVnb`, которая не подчиняется стандарту наименований, используется для перевода сечений из MeV^{-2} в нанобарны. Наиболее интересные программы из библиотеки, которые могут быть вызваны пользователем, приведены в следующем списке:

<code>Cr2Kc(E)</code>	<code>CrKsKl(E)</code>
<code>Cr2Pi(E)</code>	<code>Cr3Pi(E)</code>

CrEtaG(E)
inifit

CrPiOG(E)

Здесь E – энергия в системе центра масс $E = \sqrt{s}$. Более подробное описание данной библиотеки выходит за рамки данной публикации.

17 Графическое представление результатов

Кроме выходного файла, в котором приводятся значения параметров, полученные при аппроксимации, создаются выходные файлы `<chisfile>_exp.his` с экспериментальными данными и `<chisfile>_thr.his` с протабулированным теоретическим сечением, где вместо `<chisfile>` будет подставлено значение одноименного параметра из вызова `fitbegin`.

Формат этих файлов следующий: строки начинающиеся с символа '#' являются комментариями, иначе строка представляет информацию для одной энергетической точки. В файле `<chisfile>_exp.his` в строке записываются:

- индекс компоненты или процесса. Если это число меньше < 100 , то это компонента, если > 100 , то процесс;
- энергия в системе центра масс;
- ошибка энергии;
- экспериментальное сечение;
- верхняя ошибка экспериментального сечения;
- нижняя ошибка экспериментального сечения;
- отношение наблюдаемого сечения к радиоправленному сечению ($1/\varepsilon$);
- отношение радиоправленного сечения к борновскому ($1 + \delta$), вычисленная по теоретической зависимости;

Файл `<chisfile>_thr.his` содержит протабулированные точки теоретической зависимости сечения от энергии. Шаг таблицы выбирается автоматически, в зависимости от поведения сечения. Гарантируется присутствие в таблице точек со всеми экспериментальными энергиями. В одной строке приводятся следующие величины:

- индекс компоненты или процесса, аналогичный тому, который присутствовал в экспериментальном файле;
- значение энергии в системе центра масс;
- величина наблюдаемого сечения, т.е. сечения с учтенной радиопоправкой и эффективностью;
- величина радиоправленного сечения;
- величина борновского сечения;
- величина сечения нерезонансного фона;

Помимо простого разглядывания содержимого этих файлов, пользователь может воспользоваться макросом `crsfit.kumac` для отрисовки полученных сечений в PAW [9], Работа с этим макросом состоит из двух этапов – загрузка файлов с диска в вектор и отрисовка информации из вектора. Загрузку можно осуществить командой:

```
crsfit load vectorname <chisfile>.
```

Отрисовка загруженной информации выполняется по команде:

```
crsfit vectorname procindex options,
```

где *vectorname* - имя загруженного вектора, *procindex* - индекс компоненты или суперпроцесса, *options* - строка с опциями для отрисовки. Отдельная опция представляется символом из следующего списка:

- s - `superimpose`, наложить на тот же рисунок;
- e - `no experimental data`, не рисовать экспериментальные точки;
- t - `no theoretical curve`, не рисовать теоретическую кривую;
- v - `visible cross section`, рисовать наблюдаемое сечение, с учтенной радпоправкой и эффективностью;
- r - `radcor cross section`, рисовать радпоправленное сечение;
- c - `born cross section`, рисовать борновское сечение;
- f - `efficiency`, рисовать эффективность;
- o - `radcor`, рисовать радпоправку $1 + \delta$;
- l - рисовать нерезонансный фон;
- 0 - гарантировать появление на рисунке оси абсцисс;
- 1 - рисовать засечки на конце ошибках экспериментальных точек;
- a - убрать ограничения на отрисовку суперпроцесса/компоненту;
- C - соединять точки кривой сплайном;

По умолчанию набор опций состоит из строки ‘‘v1’’. Внешний вид рисунка можно настроить, изменяя следующие установки PAW: `ksiz`, `mtyp`, `pmci`, `plci`, `lwid`, `ltyp`, `hcol`, `fcoll`. На рис. 2 приводится типичный вид рисунка, создаваемого макросом `crsfit`.

18 Рекомендуемое использование

Использование данного пакета для аппроксимации сечений в основном ограничивается временем счета задания. Это время зависит от следующих факторов:

- период вычисления радпоправок;
- количество энергетических точек в задании;
- сложность функции сечения;

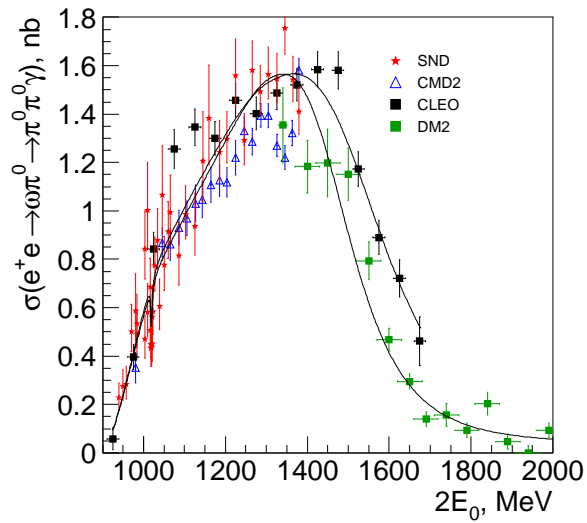


Рис. 2: Типичный вид рисунка, создаваемый макросом `crsfit`.

- корреляции параметров;
- требуется ли в задании определить асимметричные ошибки параметров при помощи `MINOS`;
- величина максимальной энергии при вычислении радпоправок;
- полный интервал энергии, на котором проводится аппроксимация;

Например, время выполнения задания по аппроксимации сечения $K_S K_L$ по экспериментальным данным детектора КМД на машине типа Pentium 100 MHz составило:

- 13 минут 40 секунд с вычислением радпоправок на каждом шаге;
- 1 минуту с вычислением радпоправок в начале задания и при выводе окончательного сечения;

19 Заключение

Был разработан универсальный пакет аппроксимации данных экспериментов на встречных e^+e^- пучках. Пакет успешно применялся и продолжает применяться в группе детектора СНД. С использованием данного пакета были получены практически все результаты СНД, в том числе по процессам $e^+e^- \rightarrow \omega\pi^0$, $e^+e^- \rightarrow \eta\gamma$, $e^+e^- \rightarrow \pi\pi$, $e^+e^- \rightarrow \mu\mu \dots$.

Литература

- [1] *F.James, M.Roos*. MINUIT. Function minimization and error analysis. CERN Program Library entry D506, 1989.
- [2] *M.N.Achasov et al.* Препринт ИЯФ им. Будкера 99-16, Новосибирск, 1999, e-print <http://xxx.lanl.gov/abs/hep-ex/9909015>.
- [3] *Кураев Э.А., Фадин В.С.* Ядерная физика, 41 (1985) 733.
- [4] *Z.Jakubowski et al.* Z. Phys. C 40, (1988), 49.
- [5] CERNLIB long writeup.
- [6] Review of Particle Physics, Eur. Phys. J. C 3, (1998).
- [7] *Бердюгин А.В.* Магистерский диплом, НГУ (1997).
- [8] *G.Bonneau, F.Martin* Nucl. Phys. B27(1971), 381.
- [9] PAW. Physics Analysis Workstation Package, CERN Program Library entry Q121, 1999.
- [10] Препринт ИЯФ им. Будкера 99-11, Новосибирск, 1999.