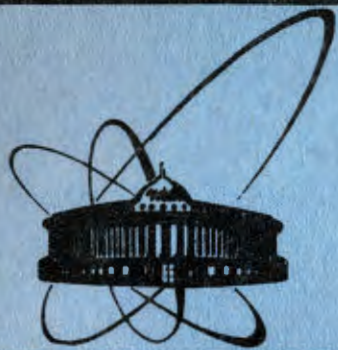


84-7

~~ЛЯП~~



ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

0-756

P5-84-7

1394/84

Г.А.Ососков, Н.И.Чернов

ЭФФЕКТИВНЫЕ АЛГОРИТМЫ
ПОДГОНКИ ОКРУЖНОСТИ

Направлено в журнал
"Computer Physics Communication"

1984

I. Во многих задачах обработки экспериментальных данных возникает необходимость аппроксимации их с помощью окружности. В некоторых случаях (снимки лунных кратеров^{1/1}, поверхности глазной роговицы^{2/2} и т.д.) вид окружности имеют сами обрабатываемые изображения. Однако чаще всего дуга окружности используется как аппроксимирующая кривая, обладающая рядом преимуществ в сравнении с другими кривыми, более простыми в подгонке, такими, как прямая или парабола. Рассмотрим в качестве примера актуальную для физики высоких энергий задачу автоматического распознавания траекторий элементарных частиц по данным оцифровки изображений взаимодействий, происходящих в экспериментальных установках.

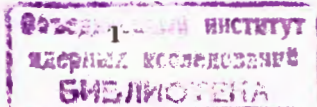
Эти данные поступают в ЭВМ в виде наборов координат точек, среди которых требуется выделить точки, относящиеся к трекам элементарных частиц, участвующих во взаимодействии. Для этого необходимо проследить треки в местах их разрывов, пересечений со следами других частиц или шумовыми образованиями.

Поскольку проекция траектории заряженной частицы в магнитном поле близка к окружности, естественно использовать уравнение последней в процессе прослеживания для аппроксимации участка траектории по уже найденным точкам. Это необходимо для проверки принадлежности точек к одной траектории, а также для экстраполяции траектории и определения района поиска новых точек, чтобы присоединить их к уже найденным. Помимо этого, окружность наиболее подходит для такой цели, так как сохраняет свою кривизну и поэтому дает лучший прогноз, чем парабола.

Следует указать на крайне жесткие требования к экономичности алгоритма подгонки в задачах прослеживания, вызванные необходимостью использования процедуры подгонки сотни раз на одно изображение при сотнях тысяч таких изображений в одном эксперименте.

Удовлетворительное решение задачи подгонки окружности практически отсутствовало из-за нелинейности этой задачи.

В настоящей работе после внимательного исследования различных алгоритмов подгонки предложены два новых алгоритма подгонки как полной окружности, так и дуги окружности. Сравнительные характеристики



(точность, число операций, время счета на ЭВМ) этих алгоритмов показывают их высокую эффективность и пригодность для исследования не только на больших ЭВМ, но и для быстрой on-line обработки на малых машинах.

2. Дадим точную формулировку задачи. Пусть заданы n точек на плоскости своими координатами (x_i, y_i) , $1 \leq i \leq n$. Требуется провести окружность

$$(x-a)^2 + (y-b)^2 = R^2 \quad (I)$$

минимизирующую сумму квадратов расстояний от точек (x_i, y_i) до проводимой окружности.

Для определения параметров этой окружности минимизируется функционал

$$L(a, b, R) = \sum_{i=1}^n (\sqrt{(x_i-a)^2 + (y_i-b)^2} - R)^2 \quad (2)$$

где справа выписана сумма квадратов расстояний

$$\rho_i = \sqrt{(x_i-a)^2 + (y_i-b)^2} - R \quad (3)$$

от точки (x_i, y_i) до окружности (I).

В силу нелинейной зависимости величин ρ_i от параметров окружности применение метода наименьших квадратов (м.н.к.) сопряжено с большим объемом вычислений. Обычно м.н.к. реализуется в виде итерационной процедуры с линеаризацией на каждом шаге и выбором начального приближения (a_0, b_0, R_0) - см., например, /3,4,5/. Соответствующая программа на ЭВМ требует слишком много машинного времени и поэтому ряд авторов /2,5/ отказывается от нее целиком или частично (используя ее только на самых последних этапах обработки).

В работах /2,5,7/ предлагается ряд упрощенных быстродействующих алгоритмов подгонки окружности, уступающих м.н.к. в точности оценок параметров. Опишем кратко два из них, т.к. это необходимо нам для дальнейшего.

3. Первый метод основан на аппроксимации точек полиномом второй степени. Вначале производится поворот координатных осей так, чтобы ось ox была параллельна прямой, проведенной через точки (x_i, y_i) по м.н.к. - см. /2/ или прямой, проходящей через крайние справа и слева точки - см. /5/. В новой системе координат по м.н.к. строится парабола

$$y = px^2 + qx + r \quad (4)$$

В качестве аппроксимирующей окружности берется окружность радиуса $\frac{1}{2p}$, касающаяся параболы (4) в ее вершине. В работе /5/ параметры этой окружности, также принимаются в качестве начального приближения для итерационной процедуры м.н.к. Данный метод подгонки окруж-

ности мы будем для краткости называть "параболическим" и обозначать м.п.

Описанный метод экономичнее, чем м.н.к., но существенно уступает ему в точности оценок. Даже в том случае, когда точки (x_i, y_i) лежат на одной окружности, параболический метод не дает точных результатов. В работе /7/ предлагается после получения оценок параболическим методом уточнять их с помощью различных процедур, что увеличивает время счета. Серьезным недостатком параболического метода является и то, что он применим лишь в случае, когда точки (x_i, y_i) располагаются вдоль небольшой дуги окружности (не более $30^\circ - 40^\circ$). В тех прикладных задачах, где точки (x_i, y_i) располагаются вдоль всей окружности, данный метод неприменим принципиально.

4. Второй метод основан на введении новой переменной в уравнение окружности (I), что позволяет свести его к уравнению линейной регрессии. Запишем уравнение (I) в виде

$$x^2 + y^2 = 2ax + 2by + R^2 - a^2 - b^2$$

и обозначим $z = x^2 + y^2$ (новая переменная), $\alpha = 2a$, $\beta = 2b$, $\gamma = R^2 - a^2 - b^2$. Получим

$$z = \alpha x + \beta y + \gamma$$

Это уравнение линейной регрессии переменных z, x, y с неизвестными параметрами α, β, γ . Оценивая по м.н.к. параметры α, β и γ , вычисляем искомые параметры a, b, R /8/.

Приведем соответствующие формулы, т.к. они понадобятся нам далее. Будем использовать обозначения, известные под названием "гауссовых скобок":

$$\sum_{i=1}^n x_i^p y_i^q = [x^p y^q]$$

и вообще для любой функции $f(x, y)$

$$\sum_{i=1}^n f(x_i, y_i) = [f(x, y)]$$

Чтобы упростить вычисления, удобно перенести начало координат в центр тяжести системы точек (x_i, y_i) . В такой системе координат $[x] = [y] = 0$ (здесь и далее $[]$ - "гауссовы скобки"). Мы будем предполагать, что соответствующий перенос уже сделан. Тогда система нормальных уравнений для параметров α, β и γ будет иметь вид

$$\begin{cases} \alpha [x^2] + \beta [xy] = [x(x^2 + y^2)] \\ \alpha [xy] + \beta [y^2] = [y(x^2 + y^2)] \\ n\gamma = [x^2 + y^2] \end{cases} \quad (5)$$

Вычисление параметров α, β, γ и a, b, R не представляет затруднений. Будем для краткости называть данный алгоритм методом сведения к линейной регрессии и обозначать м.л.р.

Данный метод, кроме быстродействия, привлекателен еще и тем, что дает совершенно точные результаты в случае, когда точки (x_i, y_i) лежат на одной окружности. Однако такой способ подгонки имеет весьма существенные недостатки, на которых необходимо остановиться.

Легко видеть, что м.л.р. заключается в минимизации функционала

$$M(a, b, R) = \sum_{i=1}^n ((x_i - a)^2 + (y_i - b)^2 - R^2)^2. \quad (6)$$

Заметим, что с учетом (3)

$$(x_i - a)^2 + (y_i - b)^2 = (R + \rho_i)^2,$$

поэтому

$$M(a, b, R) = \sum_{i=1}^n (2R\rho_i + \rho_i^2)^2 = 4R^2 \sum_{i=1}^n \rho_i^2 \left(1 + \frac{\rho_i}{2R}\right)^2.$$

В прикладных задачах обычно $\rho_i \ll R$, иначе вообще невозможно ожидать какой-либо точности оценок. Поэтому мы можем написать (см. (2)):

$$M(a, b, R) \approx 4R^2 \sum_{i=1}^n \rho_i^2 = 4R^2 L(a, b, R). \quad (7)$$

Таким образом, м.л.р. минимизирует не квадратическую ошибку

$L(a, b, R)$, а ее произведение на квадрат радиуса. Поэтому алгоритму для минимизации функционала $M(a, b, R)$ "бывает выгоднее" допустить увеличение квадратической ошибки ради уменьшения радиуса R . Типичный результат подгонки по м.л.р. представлен на рис. I.

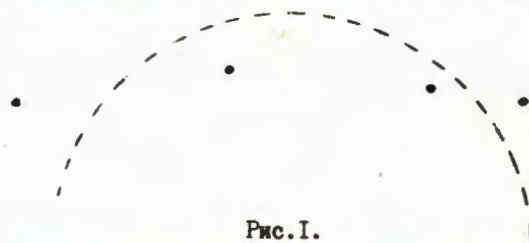


Рис. I.

Расчеты на ЭВМ показали, что даже при весьма небольшом случайном разбросе точек (x_i, y_i) *) точность оценок по м.л.р. ниже, чем по описанному в п.3 параболическому методу. Такая неустойчивость, излишняя чувствительность даже к небольшим случайным ошибкам - характерный

*) Для тестового расчета были взяты значения $a=b=0$, $R=1000$, а среднеквадратическое отклонение точек (x_i, y_i) было равно 1 см, также п.7.

недостаток данного алгоритма. Однако следует заметить, что при увеличении длины дуги, вдоль которой расположены точки (x_i, y_i) , точность м.л.р. - оценок возрастает и они практически совпадают с м.н.к. - оценками в случае, когда точки (x_i, y_i) расположены вдоль всей окружности.

Данный метод обобщается на случай подгонки любой алгебраической кривой вида

$$P(x, y) = 0, \quad (8)$$

где P - полином любой заданной степени. Например, уравнение кривой второго порядка можно записать в виде

$$y^2 = a_1 x^2 + a_2 xy + a_3 x + a_4 y + a_5$$

и введением новых переменных $y^2 = z$, $x^2 = u$, $xy = t$ получить уравнение линейной регрессии. После этого оценки параметров $a_1 - a_5$ находятся решением соответствующей системы линейных уравнений. Вариант такого метода для подгонки окружности мы предлагаем в п.6. Вопрос о точности оценок должен решаться в каждом отдельном случае, отметим лишь, что для замкнутых кривых высокая точность достигается при расположении точек (x_i, y_i) вдоль всей кривой (8).

5. Переходим к описанию предлагаемых нами алгоритмов подгонки окружности. Первый из них универсален, т.к. подходит для подгонки как полной окружности, так и дуги любого размера. Второй несколько превосходит его в быстродействии, но применим лишь для подгонки дуги малой величины. Он будет описан в п.6.

Первый из предлагаемых алгоритмов основан на идее описанного выше метода сведения к линейной регрессии, но он существенно улучшает точность оценок.

Формулу (7) перепишем в виде

$$\frac{M(a, b, R)}{R^2} \approx 4L(a, b, R).$$

На основании этого мы для минимизации функционала $L(a, b, R)$ будем минимизировать функционал

$$K(a, b, R) = \frac{M(a, b, R)}{R^2}$$

Учитывая (6), запишем

$$K(a, b, R) = \sum_{i=1}^n \left(\frac{x_i^2 + y_i^2}{R} - 2 \frac{a}{R} x_i - 2 \frac{b}{R} y_i + \frac{a^2 + b^2 - R^2}{R} \right)^2.$$

Для нахождения минимума данного функционала приравняем к нулю его частные производные по a, b и R . Получим систему уравнений

$$\begin{cases} a[3x^2+y^2]+2b[xy]+na(a^2+b^2-R^2)=[x(x^2+y^2)] \\ 2a[xy]+b[x^2+3y^2]+nb(a^2+b^2-R^2)=[y(x^2+y^2)] \\ 2a[x(x^2+y^2)]+2b[y(x^2+y^2)]+n(a^2+b^2-R^2)^2=[(x^2+y^2)^2] \end{cases}$$

Обозначим $F=\frac{1}{n}[3x^2+y^2]$, $G=\frac{1}{n}[x^2+3y^2]$, $H=\frac{2}{n}[xy]$, $P=\frac{1}{n}[x(x^2+y^2)]$,
 $Q=\frac{1}{n}[y(x^2+y^2)]$, $T=\frac{1}{n}[(x^2+y^2)^2]$.

Вводя, как и ранее, новый параметр $\gamma=R^2-a^2-b^2$, получаем систему уравнений для параметров a, b, γ

$$\begin{cases} Fa+Hb-a\gamma=P \\ Ha+Gb-b\gamma=Q \\ 2Pa+2Qb+\gamma^2=T \end{cases} \quad (9)$$

Это система трех уравнений второй степени. Она допускает исключение любой пары неизвестных, после чего сводится к уравнению 4-й степени для одного неизвестного. Наиболее удобным является исключение неизвестных a, b и решение уравнения 4-й степени относительно γ :

$$\gamma^4+A\gamma^3+B\gamma^2+C\gamma+D=0 \quad (10)$$

коэффициенты которого $A=-F-G$, $B=FG-T-H^2$, $C=T(F+G)-2(P^2+Q^2)$,
 $D=T(H^2-FG)+2(P^2G+Q^2F)-4PQH$.

Несмотря на кажущуюся сложность такого пути по сравнению с м.п. и м.л.р., удалось найти быстродействующую реализацию его на ЭВМ. Опишем способ численного решения уравнения (10). Оно имеет четыре корня, поэтому как для выбора искомого корня среди всех корней, так и для начального шага любой итерационной процедуры требуется знать приближенное значение искомого корня. Воспользуемся системой нормальных уравнений (5), дающей следующую оценку параметра γ :

$$\gamma_0 = \frac{1}{n} ([x^2] + [y^2])$$

Ее вычисление в данном случае не требует никаких затрат, так как величины $[x^2]$, $[y^2]$ вычислялись при нахождении коэффициентов уравнения (10). Значение γ_0 является хорошим начальным приближением искомого корня γ .

Для минимизации ошибок округления приведем коэффициенты уравнения (10) к одному порядку, разделив его на γ_0^4 . Получим уравнение относительно переменной $x=\gamma/\gamma_0$ с коэффициентами $A_0=A/\gamma_0^4$,
 $B_0=B/\gamma_0^3$, $C_0=C/\gamma_0^2$ и $D_0=D/\gamma_0$:

$$x^4+A_0x^3+B_0x^2+C_0x+D_0=0 \quad (11)$$

Нетрудно заметить, что всегда $A_0=-4$. Для остальных коэффициентов имеет место оценки

$$3-n \leq B_0 \leq 3; \quad 4-2n \leq C_0 \leq 4n; \quad -4 \leq D_0 \leq 2, 5n-3$$

вывод которых в силу громоздкости мы опускаем.

Для решения уравнения (11) пригодна любая итерационная процедура с начальным приближением $x_0=1$. Мы воспользовались методом Ньютона, дающим за 2-5 итераций значение корня x с максимально допустимой на ЭВМ точностью ($\sim 10^{-13}$). Вычислив параметр $\gamma=\gamma_0 x$, мы находим a и b из системы (9) и

$$R = \sqrt{a^2+b^2+\gamma} \quad (12)$$

Таким образом, построенный алгоритм подгонки окружности состоит из пяти этапов:

I. Перенос начала координат в центр тяжести системы точек (x_i, y_i) .

II. Вычисление значений "гауссовых скобок"

$$[x^2], [y^2], [xy], [x(x^2+y^2)], [y(x^2+y^2)], [(x^2+y^2)^2]$$

III. Вычисление коэффициентов уравнения (11).

IV. Решение уравнения (11) итерационным методом Ньютона с начальным приближением $x_0=1$.

V. Вычисление параметров γ, a, b, R .

Ни один из этих этапов не требует трудоемких вычислений. Программа на языке ФОРТРАН (см. приложение) содержит около 90 операторов и имеет одно обращение к подпрограмме извлечения квадратного корня SQRT - при вычислении радиуса R по формуле (12). Следует иметь в виду, что в пороговых процедурах принятия решений, применяемых при прослеживании треков, достаточно использования R^2 , поэтому операция SQRT может быть опущена.

6. Второй предлагаемый нами алгоритм подгонки окружности предполагает, что точки расположены вдоль дуги небольшого размера (не более 40°).

Перенесем начало координат в центр тяжести системы точек (x_i, y_i) и повернем координатные оси так, чтобы ось OX совпала с прямой, проведенной по м.н.к. или стала параллельна прямой, проведенной через две крайние точки. В такой системе координат дуга становится почти горизонтальной (рис.2) и $|b| \approx R$, поэтому формулу (7) можно переписать в виде

$$\frac{M(a, b, R)}{b^2} \approx 4L(a, b, R)$$

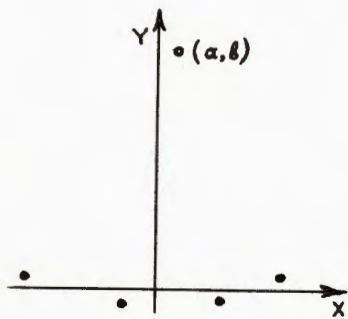


Рис. 2.

На основании этого соотношения мы будем минимизировать функционал

$$\tilde{K}(a, b, R) = \frac{M(a, b, R)}{b^2}$$

т.е.

$$\tilde{K}(a, b, R) = \sum_{i=1}^n \left(\frac{x_i^2 + y_i^2}{b} - 2 \frac{a}{b} x_i - 2 y_i + \frac{a^2 + b^2 - R^2}{b} \right)^2$$

Приравнявая к нулю частные производные по a, b, R , получим систему линейных уравнений

$$\begin{cases} 2aP + 2bQ + \gamma([x^2] + [y^2]) = T \\ 2a[x^2] + 2b[xy] = P \\ n\gamma = [x^2] + [y^2] \end{cases}$$

где использованы введенные ранее обозначения γ, P, Q, T . Далее вычисление параметров a, b, R не представляет затруднений.

В силу аналогии с системой (5) для м.л.р. будем далее называть этот метод модифицированным методом сведения к линейной регрессии (м.м.л.р.).

7. Характеристики описанных в п.5 и п.6 методов подгонки окружности были исследованы на ЭВМ CDC-6500. В качестве исходных данных (x_i, y_i) были смоделированы случайные точки, расположенные вдоль дуги окружности с центром $a=b=0$ и радиусом $R=1000$. Длина дуги была около 400 (от точки $(0, 1000)$ до точки $(400, 100\sqrt{64})$), а количество точек менялось от $n=4$ до $n=50$. Среднеквадратическое отклонение точек от окружности было равно 1.

Оценки параметров a, b, R и дисперсии σ^2 по предлагаемым нами методам сравнивались с оценками по м.н.к., м.п. и м.л.р.

Результаты сравнения точности оценок приведены в таблице I. Оценки по м.н.к. считались наиболее точными, для остальных приведена их относительная погрешность по сравнению с м.н.к. - оценками (описанный в п.5 метод обозначен и.м.л.р. - итерационный метод сведения к линейной регрессии).

Таблица I.

	оценки a, b, R	оценка σ^2
м.п.	1%	2%
м.л.р.	2%	3%
м.м.л.р.	0,05%	0,005%
и.м.л.р.	0,01-0,02%	$\leq 0,001\%$

Здесь оценка дисперсии вычислялась по формуле

$$\hat{\sigma}^2 = \frac{1}{n-3} \sum_{i=1}^n \left(\sqrt{(x_i - a)^2 + (y_i - b)^2} - R \right)^2$$

Заметим, что в быстрых алгоритмах подгонки окружности, требующих оценки дисперсии, используется приближенная формула, вытекающая из (3) и (7) (см. также [6])

$$\hat{\sigma}^2 = \frac{1}{4R^2(n-3)} \sum_{i=1}^n \left((x_i - a)^2 + (y_i - b)^2 - R^2 \right)^2 \quad (13)$$

позволяющая избежать операции SQRT.

Точность алгоритма м.м.л.р. существенно зависит от величины дуги. При малых дугах его точность близка к и.м.л.р., при больших резко падает.

Продолжительность работы каждого алгоритма была оценена с помощью счетчика времени SECOND на ЭВМ CDC-6500. Результаты приведены в таблице 2, где указано время в миллисекундах (10^{-3} сек).

Таблица 2.

	n=4	n=10	n=50
м.н.к.	8	17	75
м.п.	1,1	2,0	8,1
м.л.р.	0,7	1,5	5,9
м.м.л.р.	0,8	1,6	6,0
и.м.л.р.	1,2	2,0	6,6

Из таблицы видно, что и.м.л.р. не уступает в быстродействии параболическому методу, что означает бесперспективность различных уточняющих процедур для м.п. - оценок, которые упоминались в п.3.

В таблице 3 приведено число операций, требуемых упомянутыми алгоритмами. Здесь общее число операций вычисляется по общепринятому правилу: сложение и вычитание - одна операция, умножение и деление - три операции, извлечение квадратного корня - 25 операций.

Таблица 3.

	+, -	*, /	$\sqrt{\quad}$	общее
М.Н.К.	97n+169	149n+245	7n+1	719n+537
М.П.	13n+ 29	9n+ 49	1	40n+201
М.Л.Р.	11n+ 8	6n+ 20	1	29n+ 93
М.М.Л.Р.	11n+ 14	6n+ 39	1	29n+156
И.М.Л.Р.	11n+16i+21	6n+7i+42	1	29n+37i+172

Здесь i - число итераций метода Ньютона при решении уравнения (II). Для определения этой величины, а также для проверки сходимости итерационной процедуры был проделан численный эксперимент, в ходе которого в единичный квадрат 10000 раз бросались наборы из 10 случайных точек. Случаев расходимости итерационной процедуры зарегистрировано не было, среднее число итераций равно 5. Необходимо заметить, что если исходные данные (x_i, y_i) располагаются вдоль дуги окружности (как в описанной выше тестовой модели), то среднее число итераций сокращается и становится равным 3.

В ряде задач отслеживания треков заряженных частиц в физике высоких энергий приходится работать только с треками, имеющими вид горизонтально расположенных дуг большого радиуса. В этом случае можно применять м.м.л.р., опуская поворот координатных осей. Время работы такой процедуры совпадает с м.л.р. (табл.2), а точность оценок близка к м.м.л.р. (табл.1). Если в соответствующих пороговых процедурах принятия решения использовать величину R^2 , а оценку дисперсии вычислять по формуле (13), то мы получим экономичный вариант алгоритма подгонки для использования в быстрой on-line обработке на малых ЭВМ.

Литература

1. Астафьев В.А. и др. В кн.: "Тезисы Первого Всесоюзного семинара по автоматизации научных исследований в ядерной физике и смежных областях". Изд-во "Дониш", Душанбе, 1980, с.267.
2. Ососков Г.А. ОИЯИ, Р10-83-187, Дубна, 1983.
3. Ван-дер-Варден Р., Математическая статистика. ИЛ, М., 1960.
4. Соколов С.Н., Силин И.Н. ОИЯИ, Д-810, Дубна, 1961.
5. Липкин И.М. и др. Препринт ИТЭФ, № 47, М., 1975.
6. Hall D. UCRL-18544, Berkeley, California, 1968.
7. Говорун Н.Н. и др. ОИЯИ, 1101, Дубна, 1962.
8. Говорун Н.Н. и др. ОИЯИ, 2036, Дубна, 1965.

ПРИЛОЖЕНИЕ.

```

SUBROUTINE CIRCLE(NP,X,Y,XC,YC,R)
DIMENSION X(NP),Y(NP)
C
C***** PROGRAM OF CIRCLE FITTING *****
C INPUT DATA: NP-NUMBER OF POINTS, X,Y-THEIR COORDINATES
C OUTPUT DATA: XC,YC,R-ESTIMATIONS OF PARAMETERS
C
C
N=NP
RN=1./N
C
C CENTRE OF MASSES
C
XM=YM=0.
DO 1 I=1,N
  XM=XM+X(I)
  YM=YM+Y(I)
1  XM=XM*RN
  YM=YM*RN
C
C GAUSSIAN BRANCHES
C
X2=Y2=XY=XD=YD=D2=0.
DO 2 I=1,N
  XI=X(I)-XM
  YI=Y(I)-YM
  XX=XI*XI
  YY=YI*YI
  DD=XX+YY
  X2=X2+XX
  Y2=Y2+YY
  XY=XY+XI*YI
  XD=XD+XI*DD
  YD=YD+YI*DD
  D2=D2+DD*DD
2  X2=X2*RN
  Y2=Y2*RN
  XY=XY*RN
  XD=XD*RN
  YD=YD*RN
  D2=D2*RN
C
F=3.*X2+Y2
G=3.*Y2+X2
H=XY+XY
H2=H*H
P2=XD*XD
Q2=YD*YD
FG=F*G
C
C COEFFICIENTS OF EQUATIONS
C
GAM0=X2+Y2
FACT=GAM0*GAM0
A2=(FG-H2-D2)/FACT
FACT=FACT*GAM0
A1=(D2*(F+G)-2.*(P2+Q2))/FACT
FACT=FACT*GAM0
A0=(D2*(H2-FG)+2.*(P2*G+Q2*F)-4.*XD*YD*H)/FACT

```

C
C
C

ITERATIONAL PROCEDURE BY NEWTONS METHOD

```
A22=A2+A2
ITMAX=20
EPS=1.E-10
YB=1.E50
ITER=0
XA=1.
3 YA=A0+XA*(A1+XA*(A2+XA*(XA-4.)))
IF (ABS(YA).GT.ABS(YB)) GO TO 4
IF (ITER.GE.ITMAX) GO TO 4
DY=A1+XA*(A22+XA*(4.*XA-12.))
XB=XA-YA/DY
IF (ABS(XA-XB).LT.EPS) GO TO 5
XA=XB
YB=YA
ITER=ITER+1
GO TO 3
4 PRINT 99,ITER
XB=1.
C
C
C
PARAMETERS OF CIRCLE
5 GAM=GAM0*XB
F1=F-GAM
G1=G-GAM
DET=F1*G1-H2
XC=(XD*G1-YD*H)/DET
YC=(YD*F1-XD*H)/DET
R=SQRT(XC*XC+YC*YC+GAM)
XC=XC+XM
YC=YC+YM
RETURN
99 FORMAT (/10X+++++ NO CONVERGENCE AFTER*I4*ITERATIONS*/)
END
```

Примечание. В случае использования программы на ЭВМ с меньшей разрядностью сетки для вещественных чисел (например, на ЭВМ серии ЕС) следует описывать переменные

X2, Y2, XY, XD, YD, D2, F, G, H, F1, G1, H2, P2, Q2, FG, GAM0, GAM, FACT, DET, A0, A1, A2, A22, XA, XB, YA, YB, DY

с двойной точностью.

Рукопись поступила в издательский отдел
5 января 1984 года.

Ососков Г.А., Чернов Н.И.

P5-84-7

Эффективные алгоритмы подгонки окружности

После изучения известных методов подгонки окружности предложены два новых эффективных алгоритма, удовлетворяющие всем требованиям, предъявляемым при массовой обработке данных.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1983

Ososkov G.A., Chernov N.I.

P5-84-7

Effective Algorithms of Circle Fitting

After the investigation of known methods for circle fitting two new effective algorithms are proposed which meet all requirements of mass data handling.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Preprint of the Joint Institute for Nuclear Research, Dubna 1983

Перевод Панковой Н.С.