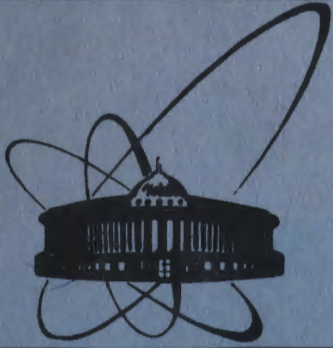


2/IV-84



Объединенный
институт
ядерных
исследований
Дубна

1685/84

P11-83-909

Х.Лайх, В.И.Приходько, К.Фогт

ГЕНЕРИРОВАНИЕ ДУГ ОКРУЖНОСТЕЙ

Направлено в журнал
"Rechentechnik/Daten Verarbeitung"

1983

1. ВВЕДЕНИЕ

Разработка эффективных алгоритмов построения кривых или отрезков кривых 1-го или 2-го порядков с помощью ЭВМ имеет большое значение при создании устройств управления плоттерами и графическими дисплеями. При этом целью всех алгоритмов является достижение минимального отклонения аппроксимирующей кривой от истинной, определенной математическим выражением, при минимальных вычислительных затратах.

Математическое описание кривой играет большую роль при выборе способа аппроксимации. В дальнейшем рассматривается параметрическое и непараметрическое описание кривых. Например, окружность описывается в параметрической форме уравнениями:

$$x = \sin t, \quad y = \cos t, \quad /1/$$

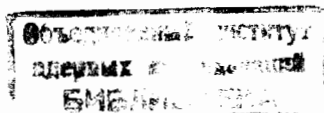
а в непараметрической форме -

$$x^2 + y^2 - r^2 = 0. \quad /2/$$

Генерирование кривых, заданных в параметрической форме, легко осуществляется с помощью цифровых дифференциальных анализаторов /ЦДА/. Однако при этом трудно получить хорошую точность, т.к. кривые зачастую получаются вырожденными/1/. Кроме того, затраты на техническую реализацию этого способа оказываются сравнительно высокими.

Принципы генерирования кривых параметрическим способом были детально исследованы Брезенхэмом/2/ и Метцгером/3/. Разработанные ими алгоритмы связаны, однако, со значительными затратами на вычисления. В дальнейшем параметрический способ генерирования кривых не получил широкого распространения и был практически полностью вытеснен непараметрическим способом.

Уже в 1965 году Брезенхэм/2/ представил простой и быстрый алгоритм генерирования прямых. Другие авторы/3/ усовершенствовали этот алгоритм и он сегодня наиболее часто используется при генерировании прямых в машинной графике. Основной принцип алгоритма Брезенхэм применил и для генерирования дуг окружностей/5/. При аппроксимации кривой этот алгоритм обходится только операциями сложения и вычитания, однако объем вычислений сравнительно велик.



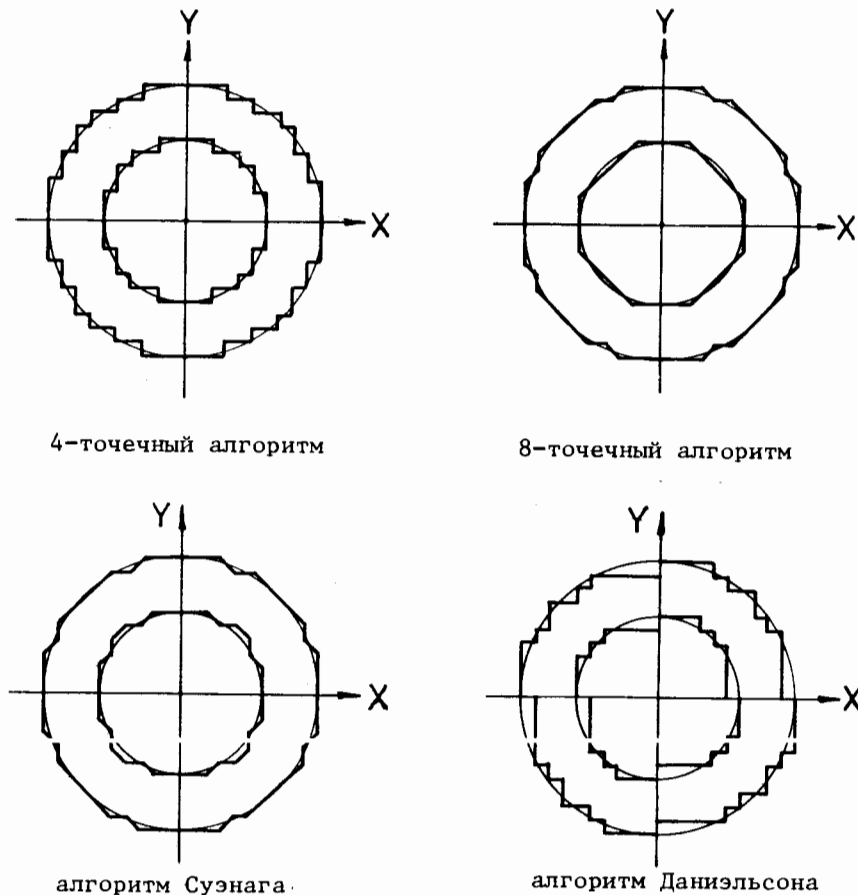


Рис. 1

Даниэльсон/1/ предложил алгоритмы на основе непараметрического метода для генерирования прямых и окружностей и сравнил их с методами на основе ЦДА. Он показал, что непараметрический способ позволяет генерировать замкнутые невырожденные окружности. Этот алгоритм использует инкрементные шаги либо в X-, либо в Y-направлениях. Шаги под углом 45° не допускаются.

Независимо друг от друга Йордан, Лэннон, Холм/4/ и Камаз/6/ предложили новый алгоритм, который они назвали 4-точечным и 8-точечным алгоритмами соответственно. Эти алгоритмы позволяют генерировать окружности с очень малыми отклонениями от истинной кривой. При этом 8-точечный алгоритм допускает шаги в X-, Y-направлениях и под углом 45° , что увеличивает симметрию аппроксимирующей кривой.

Суэнага, Камаз и Кобаяши представили в /7/ очень быстрый, улучшенный алгоритм. Изложенные в /7/ результаты свидетельствуют о преимуществе нового алгоритма по скорости, симметрии и объему вычислений по сравнению с названными выше алгоритмами. На рис.1 показано генерирование окружностей на основе ряда рассмотренных алгоритмов.

2. ПОСТРОЕНИЕ ОКРУЖНОСТЕЙ

В ОИЯИ разрабатывается интеллектуальный графический терминал на основе мультимикропроцессорной системы/8,9/. С целью быстрого генерирования примитивов, таких, как линии и окружности, графический процессор этого терминала построен на базе микропрограммируемого процессора на микропроцессорных секциях типа ИНТЕЛ-3000. В то же время отсюда вытекают определенные трудности в реализации генератора окружностей. Одна из них связана с тем, что для получения высокого быстродействия при построении окружностей допускаются только операции сложения, вычитания и сдвига. Другое условие связано с ограниченностью имеющейся в процессоре микропрограммной памяти. Исходя из этих условий был выбран рассмотренный ниже конкретный алгоритм.

Непараметрическое представление кривой в двумерном пространстве имеет форму:

$$f(x, y) = 0.$$

/3/

Значения функции для точек, не принадлежащих заданной кривой, отличны от нуля. Это изменение знака функции используется в качестве критерия выбора следующего шага аппроксимации. На рис.2 видно, что из заданной точки P_0 возможен переход в одну из восьми соседних точек $P_1 \dots P_8$, если шаг составляет один дискрет по обеим осям. Цель, поставленная при разработке алгоритма, состоит в том, чтобы найти такой критерий, который позволил бы в каждом шаге аппроксимации найти точку, имеющую наименьшее расстояние от истинной кривой. Путем задания направления генерирования дуги окружности возможные переходы из точки P_0 ограничиваются тремя направлениями /см.рис.3/. В табл.1 сведены различные критерии для определения следующего шага аппроксимации.

Вместо сравнения значений функции $f(x, y)$ в двух соседних точках, Суэнага и др. предлагали сравнивать значение функции в точке P_1 с определенной константой K . Таким путем значительно уменьшается объем вычислений, и генерирование происходит более быстро.

Значение функции /2/ в соседней точке определяется выражением:

$$f(x \pm 1, y \pm 1) = f(x, y) \pm 2x \pm 2y + 2,$$

/4/

из которого можно вычислить значение функции в заданной точке, исходя из значения функции в предыдущей точке, при помощи опе-

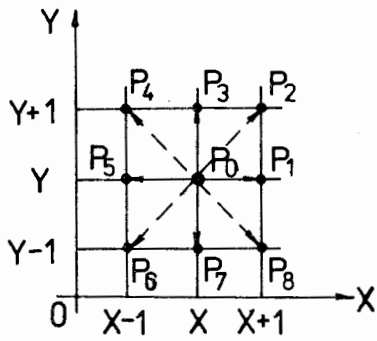


Рис. 2

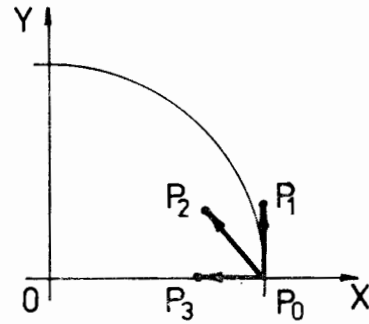


Рис. 3

Таблица 1

Алгоритм	Критерий
Алгоритм Даниэльсона	Если $f(P_0) \geq 0$, то переход в P_3 , в противном случае - в P_1 .
4-точечный алгоритм	Если $ f(P_1) \leq f(P_3) $, то переход в P_1 , в противном случае - в P_3 .
8-точечный алгоритм	Если $ f(P_1) \leq f(P_2) $, то переход в P_1 , в противном случае - в P_2 .
Алгоритм Суэнага и др.	Если $ f(P_1) < K$, то переход в P_1 , в противном случае - в P_2 .

раций сложения, вычитания и сдвига. При этом исходная точка аппроксимации должна лежать на окружности, т.е.

$$f(x_H, y_H) = 0. \quad /5/$$

Итак, алгоритм Суэнага и др. наилучшим образом удовлетворяет заданным требованиям.

В [7] показано, что в качестве константы K можно использовать радиус r . Окружность можно рассматривать как пересечение плоскости X, Y с параболоидом $Z = x^2 + y^2 - r^2$ /см.рис.4/. Алгоритм рассматривает изменение значений функции $f(P_i)$. Если значение функции меньше радиуса r , то осуществляется переход в P_1 . Чем дальше аппроксимирующая кривая отходит от истинной, тем больше значение $|f(P_i)|$ /см. рис.5/. Если превышает порог r , то

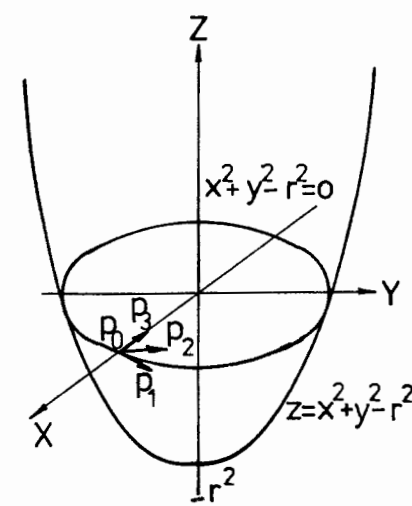


Рис. 4

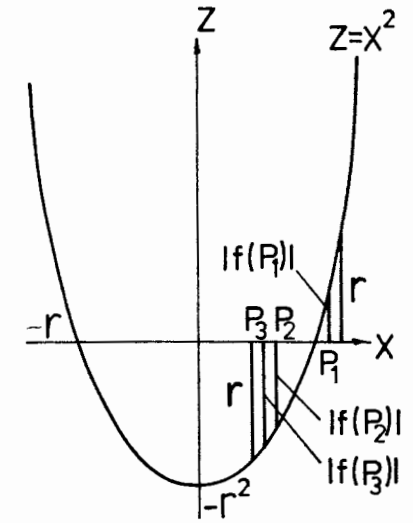


Рис. 5

необходимо сделать диагональный шаг в точку P_2 , приближаясь тем самым к истинной кривой.

3. ГЕНЕРАТОР ОКРУЖНОСТЕЙ

При построении окружности примем направление движения по кривой в положительном в математическом смысле направлении. Рассмотренный выше алгоритм в своей общей форме действителен только для одного октанта. При переходе от одного октанта к другому должны быть выполнены некоторые условия. В каждом октанта имеется определенная предпочтительная координата, которая изменяется в каждом шаге аппроксимации. Если, например, начальная точка расположена в первом октанте на оси X , то предпочтительной является Y -координата /см.рис.6/.

Когда значение функции $|f(P_1)|$ превышает порог, выполняется диагональный шаг в направлении x - и y -координатных осей /в первом октанте $y = +1, x = -1/$.

Предпочтительные координаты одинаковы в 1-м и 8-м, 2-м и 3-м, 4-м и 5-м, 6-м и 7-м октантах соответственно. Для нахождения поворотных точек, в которых изменяется наклон касательной, необходимо знать момент перехода из одного квадранта в другой /в этот момент изменяется знак неpreferred координаты/. При построении дуг окружностей важной проблемой является обнаружение начальной и конечной точек дуги. При этом в первую очередь необходимо определить октанты начальной x_H, y_H и конечной x_K, y_K точек/обе точки могут находиться и в одном октанте/. 5

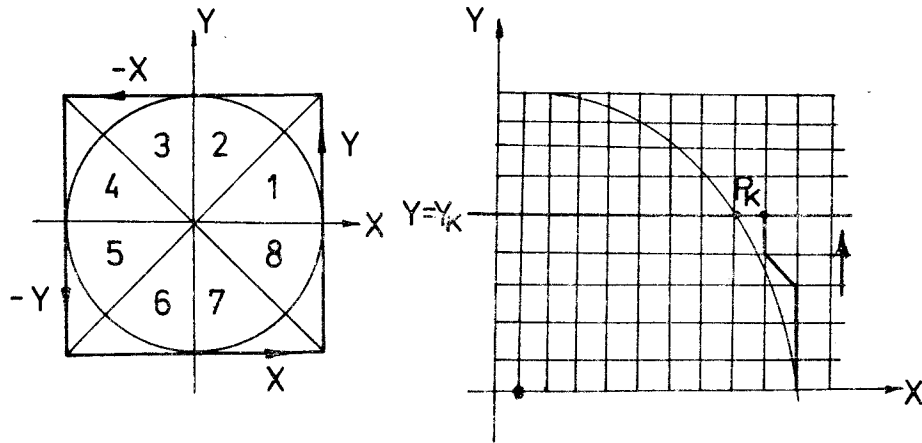


Рис. 6

Рис. 7

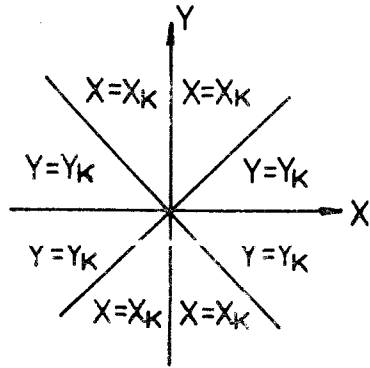


Рис. 8

Рис. 9. Блок-схема программы генерации дуг окружностей.

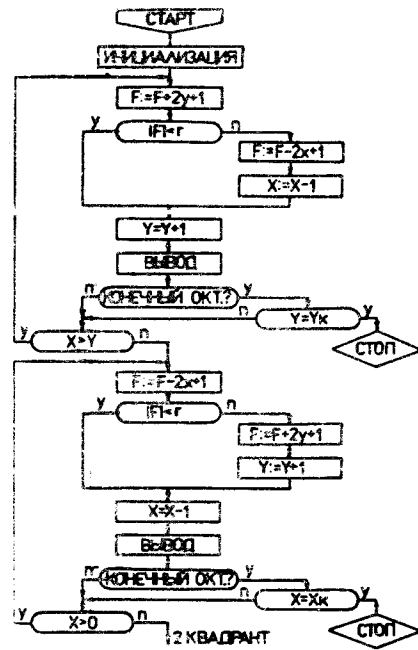
В начальной точке должно быть выполнено условие /5/. Процесс генерирования заканчивается, если значение предпочтительной координаты достигло значения соответствующей конечной координаты /рис.7/. В зависимости от октанта, в котором находится конечная точка, это может быть либо x -, либо y -координата /рис.8/; опера-

ция сравнения текущей координаты с конечной выполняется только для этого октанта.

Конечная точка не всегда находится на истинной кривой /рис.7/, однако нет необходимости в коррекции этой точки, поскольку ее максимальное отклонение от истинной всегда меньше одного дискрета, что при заданной разрешающей способности ЭЛТ незаметно для глаза.

На рис.9 показана блок-схема программы. Вычисление аппроксимирующей кривой происходит в относительных координатах, причем центр окружности или дуги может находиться в любой точке экрана. Для инициализации процесса генерации необходимо задать радиус r , координаты x и y начальной и конечной точек, а также октанты, в которых эти точки находятся.

Согласно условию /5/, начальное значение функции равно нулю. Предпочтительная координата изменяется в каждом шаге аппроксимации и полученное значение функции сравнивается с порогом r . Если это значение превышает порог, то требуется заново вычислить значение функции в точке P_2 . Необходимость такой операции связана с тем, что в нормальном случае вычисляется значение функции в текущей точке в направлении предпочтительной координаты, а при диагональном шаге точка P_2 становится новой исходной точкой аппроксимации. При переходе в другой октант меняется выражение для определения критерия выбора следующего шага.



Описанный алгоритм был протестирован с помощью программы, написанной на языке PL/M. Затем на основе этой программы была разработана микропрограмма для графического процессора, имеющего разрядность 16 бит, что определяет максимальный радиус окружности. При разработке микропрограммы возникли проблемы, связанные с обработкой отрицательных чисел и реализацией операции сдвига, поскольку процессорные элементы I-3002 не имеют операций вычитания и сдвига влево и не позволяют манипулировать знаковым разрядом. Эти проблемы были решены путем использования специальных флагов для организации выполнения указанных операций с помощью имеющегося набора инструкций. В первом варианте алгоритма операции с флагами выполнялись на внутренних регистрах секций I-3002. В дальнейшем с целью увеличения быстродействия генератора указанные флаги были реализованы схемным путем; введены также соответствующие команды для установки, сброса и опроса этих флагов в процессе выполнения микропрограммы, что значительно облегчило реализацию алгоритма. В табл.2 сопоставлены характеристики обоих вариантов генератора дуг окружностей.

В приложении приведен фрагмент микропрограммы /вариант с дополнительной аппаратурой/, а на рис.10 - некоторые примеры построения окружностей на основе описанного алгоритма.

Таблица 2

Вариант алгоритма	Время построения дуги окружности /мкс/	
	Дуга длиной 64 точки или окружность с радиусом 10	Дуга длиной 1024 точки или окружность с радиусом 160
Реализация только на основе команд I-3002.	830	13500
Реализация алгоритма с использованием дополнительной аппаратуры	240	4900

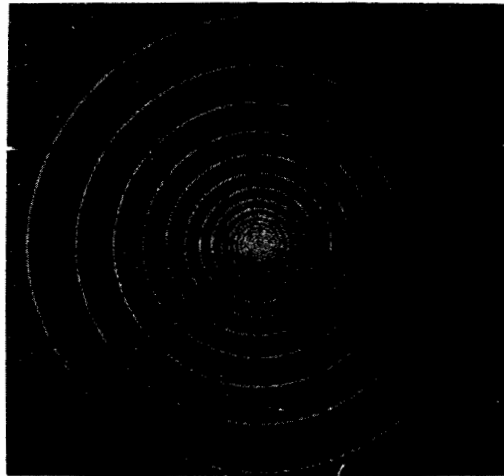
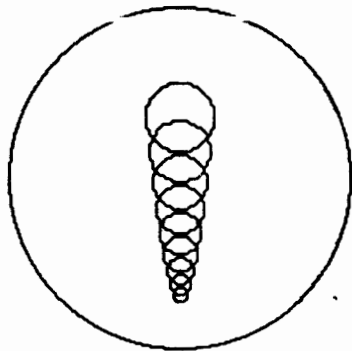


Рис.10. Примеры построения дуг окружностей: а/ на графопостроителе, б/ на экране дисплея.

4. ЗАКЛЮЧЕНИЕ

На основе анализа известных алгоритмов построения окружностей выбран наиболее эффективный алгоритм Суэнага и др., который был реализован в виде микропрограммы графического процессора, построенного на микропроцессорных секциях.

Разработанный генератор позволяет строить дуги окружностей любой длины и обеспечивает хорошую симметрию и минимальное отклонение от истинной кривой.

Приложение

GENERATION OF CIRCULAR ARCS (FIRST OCTANT)

```

001B E0 04 68 OKT1:CALL (OUTPUT);
001C BF 80 46 M10: JOF (SIGNFLAG,M11);.JUMP, IF SIGN NEGATIVE
                                .SIGN POSITIVE
001D B5 00 40 LBA (RY); .LOAD Y-COORDINATE
001E B4 04 FF STA (RZ); .RZ := Y
001F B5 00 FF ALR (RZ); .RZ, AC := 2Y
0020 B5 06 9F ADR (RFCTN); .RFCTN := F + 2Y
0021 B4 06 80 INR (RFCTN); .RFCTN := F + 2Y + 1
0022 C0 02 C0 JMP (M12);
                                .SIGN NEGATIVE
0023 B5 00 40 M11: LDA (RY); .LOAD Y-COORDINATE
0024 B4 04 FF STA (RZ); .RZ := Y
0025 B5 06 FF ADR (RZ); .RZ := 2Y
0026 B4 06 E0 INR (RZ);
0027 B5 00 E0 LDA (RZ); .AC := 2Y + 1
0028 B5 06 9F ADR (RFCTN); .RFCTN := (-F) + (2Y+1)
0029 8E 05 60 JOC (CARRY,M13); .JUMP, IF RESULT >= 0
002A C0 02 C0 JMP (M12);
002B 87 C0 00 M13: SFLAG (SIGNFLAG,0);.RESET SIGN
002C E0 03 C8 M12: CALL (CRITERION); .F < R ?
002D 9F A0 8A JNF (SUBFLAG,M15); .JUMP ON CRITERION
002E BF 80 74 JOF (SIGNFLAG,M14);.JUMP, IF SIGN NEGATIVE
                                .SIGN POSITIVE
002F B5 00 20 LDA (RX); .LOAD X-COORDINATE
0030 B4 04 FF STA (RZ); .RZ := X
0031 B5 06 FF ADR (RZ); .RZ := 2X
0032 B5 0E E0 INVR (RZ);
0033 B4 06 E0 INR (RZ);
0034 B4 06 E0 INR (RZ); .RZ := -2X + 1
0035 B5 00 E0 LDA (RZ);
0036 B5 06 9F ADR (RFCTN); .RFCTN := (-2X+1) + F
0037 8E 08 40 JOC (CARRY,M18); .JUMP, IF FCTN > 0
0038 87 C0 01 SFLAG (SIGNFLAG,1);.SIGN NEGATIVE
0039 C0 04 20 JMP (M18);
                                .SIGN NEGATIVE
003A B5 00 20 M14: LDA (RX); .LOAD X-COORDINATE

```

```

003B B4 04 FF      STA (RZ);          .RZ := X
003C B5 06 FF      ADR (RZ);
003D B5 0E E0      INVR (RZ);          .RZ := /2X
003E B4 06 E0      INR (RZ);
003F B4 06 E0      INR (RZ);          .RZ := -2X + 1
0040 B5 00 E0      LDA (RZ);
0041 B5 06 9F      ADR (RFCTN);        .RFCTN := (-F) + (-2X+1)
0042 B5 00 20      M18: LDA (RX);
0043 B5 03 FF      DCRA;
0044 B4 04 3F      STA (RX);          .RX := X - 1
0045 B4 06 40      M15: INR (RY);        .RY := Y + 1
0046 E0 04 68      CALL (OUTPUT);
0047 B4 09 40      JOC (NOGP15,M17); .JUMP, IF NO CIRC. ARC
0048 B5 0A BF      TZR (RDIF);        .TEST "LAST OCTANT?"
0049 C7 05 10      JNC (CARRY,M19);   .JUMP, IF LAST OCT.
                                .CHANGE OCTANT
004A B5 00 40      M17: LDA (RY);        .LOAD Y-COORDINATE
004B B4 04 FF      STA (RZ);
004C B5 00 20      LDA (RX);          .LOAD X-COORDINATE
004D B4 03 E0      CMTA;             .AC := -X
004E B5 06 FF      ADR (RZ);          .RZ := Y - X
004F B5 0A BF      JOC (CARRY,M16);   .JUMP, IF Y > X
0050 C0 01 C0      JMP (M10);         .JUMP TO BEGIN
                                .STOP-CRITERION
0051 E0 03 68      M19: CALL (EPOSY);
0052 B5 0A FF      TZR (RZ);
0053 BE 09 40      JOC (CARRY,M17);   .Y (<) YEND
0054 C0 06 00      JMP (STOP);        .Y = YEND
0055 B4 0B 20      M16: JOC (NOGP15,M20); .JUMP, IF NO CIRC. ARC
0056 B5 00 A0      LDA (RDIF);        .OCT := OCT - 1
0057 B5 03 FF      DCRA;
0058 B4 04 BF      STA (RDIF);        .END FIRST OCTANT

```

ЛИТЕРАТУРА

1. Danielsson P.E. IEEE Transactions on Computers, 1970, 19, 9, p.p.783-793.
2. Bresenham J.E. IBM Systems Journal, 1965, 4, 1, p.p.25-30.
3. Metzger R.A. Computer Generated Graphic Segments in a Ras-

- ter Display. Spring Joint Computer Conference, AFIPS Conf. Proc., 1969, p.p.161-172.
4. Jordan B.W., Lennon W.J., Holm B.D. IEEE Transactions on Computers, 1973, 22, 12, p.p. 1052-1060.
5. Bresenham J.E. Communications of the ACM, 1977, 20, 2, p.p.100-106.
6. Kamae T., Kosugi M., Hoshino T. Transactions IECE Japan, 1973, 56, 7, p.p.401-408.
7. Suenaga Y., Kamae T., Kabayashi T. IEEE Transactions on Computer, 1979, 28, 10, p.p. 728-736.
8. Лайх Х., Левчановский Ф.В., Приходько В.И. ОИЯИ, P11-83-385, Дубна, 1983.
9. Лайх Х., Левчановский Ф.В., Приходько В.И. ОИЯИ, P10-83-7, Дубна, 1983.

Рукопись поступила в издательский отдел
29 декабря 1983 года

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

	Труды VI Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1978 /2 тома/	7 р. 40 к.
	Труды VII Всесоюзного совещания по ускорителям заряженных частиц, Дубна, 1980 /2 тома/	8 р. 00 к.
D11-80-13	Труды рабочего совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике, Дубна, 1979	3 р. 50 к.
D4-80-271	Труды Международной конференции по проблемам нескольких тел в ядерной физике. Дубна, 1979.	3 р. 00 к.
D4-80-385	Труды Международной школы по структуре ядра. Алушта, 1980.	5 р. 00 к.
D2-81-543	Труды VI Международного совещания по проблемам квантовой теории поля. Алушта, 1981	2 р. 50 к.
D10,11-81-622	Труды Международного совещания по проблемам математического моделирования в ядерно-физических исследованиях. Дубна, 1980	2 р. 50 к.
D1,2-81-728	Труды VI Международного семинара по проблемам физики высоких энергий. Дубна, 1981.	3 р. 60 к.
D17-81-758	Труды II Международного симпозиума по избранным проблемам статистической механики. Дубна, 1981.	5 р. 40 к.
D1,2-82-27	Труды Международного симпозиума по поляризационным явлениям в физике высоких энергий. Дубна, 1981.	3 р. 20 к.
P18-82-117	Труды IV совещания по использованию новых ядерно-физических методов для решения научно-технических и народнохозяйственных задач. Дубна, 1981.	3 р. 80 к.
D2-82-568	Труды совещания по исследованиям в области релятивистской ядерной физики. Дубна, 1982.	1 р. 75 к.
D9-82-664	Труды совещания по коллективным методам ускорения. Дубна, 1982.	3 р. 30 к.
D3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р. 00 к.
D2,4-83-179	Труды XV Международной школы молодых ученых по физике высоких энергий. Дубна, 1982.	4 р. 80 к.
	Труды УШ Всесоюзного совещания по ускорителям заряженных частиц. Протвино, 1982 /2 тома/	11 р. 40 к.
D11-83-511	Труды совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике. Дубна, 1982.	2 р. 50 к.
D7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р. 55 к.
D2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р. 00 к.

Заказы на упомянутые книги могут быть направлены по адресу:
101000 Москва, Главпочтамт, п/я 79
Издательский отдел Объединенного института ядерных исследований

Лайх.Х., Приходько В.И., Фогт К.
Генерирование дуг окружностей

P11-83-909

На основе анализа известных алгоритмов построения окружностей выбран наиболее эффективный алгоритм Суэнага и др., который был реализован в виде микропрограммы графического процессора, построенного на микропроцессорных секциях. Разработанный генератор позволяет строить дуги окружностей любой длины и обеспечивает хорошую симметрию и минимальное отклонение от истинной кривой.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1983

Leich H., Prikhodko V.I., Voigt K.
Generation of Circular Arcs

P11-83-909

Based on the analysis of various algorithms for the generation of circles Suenaga's algorithm, which has been selected as the most effective one, has been realized as a microprogram for a graphics processor, which in its turn is built of bit-slice elements. The implementation of the generator permits to draw circular arcs of arbitrary length, while providing high symmetry paired with minimal deviation from the ideal curve.

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 О.С.Виноградовой