

Министерство высшего и среднего специального образования
Р С Ф С Р

Ордена Трудового Красного Знамени
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ РАДИОФИЗИЧЕСКИЙ
ИНСТИТУТ

Препринт № 44

В.В.Бочаров, Л.П.Киселев, Г.Н.Пипчук

ИРФФИ
AMES PLOT - СИСТЕМА ВЫВОДА ИНФОРМАЦИИ
В ГРАФИЧЕСКОЙ ФОРМЕ

В ы п у с к 1

Общее описание

г.Горький,
1973

Описываемая система программирования предназначена для вывода информации в графической форме на двухкоординатные устройства отображения (графопостроители, дисплеи). Являясь в целом независимой от аппаратуры вывода, она может быть приспособлена к работе с любым устройством за счет изменения одного программного модуля.

Система реализована на машинах типа М-20 и представляет собой комплекс программных модулей (написанных на автокоде-ИПМ), которые оформлены как стандартные программы в системе ИС-2, 22.

Проброобразом для данной разработки послужила система **AMESPLOT** (США-UCSD) и в дальнейшем описываемую систему мы также будем называть **AMESPLOT**

The programming system described is intended for the output of information in a picture form to two coordinate image equipments (plotters, displays).

Being, as a whole, independent of the hardware of output, it may operate with any device due to the variation of one program module.

The system is realized at the computer of M-20 type and represents the complex of program modules (written at the autocode of the Institute of Applied Mathematics) that are used as the standard programs in the IS-2, 22 system. **AMESPLOT** system (USA-UCSD) is used as a prototype for the given development, and in future the system described will also be called as **AMESPLOT**.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ AMESPLOT	3-6
AMESPLOT -ПРОГРАММА	6
ОПЕРАТОРЫ СИСТЕМЫ AMESPLOT	7
ОШИБКИ	9
ПРИМЕР	10
ОРГАНИЗАЦИЯ РАБОТЫ С СИСТЕМОЙ AMESPLOT	16
ГРАФОПОСТРОИТЕЛЬ	17
ЛИТЕРАТУРА	18

В В Е Д Е Н И Е

Система **AMESPLOT** возникла, прежде всего, как система, ориентированная на обслуживание пользователей, занятых решением научно-технических задач.

Круг вопросов, решаемых ею, включает в себя построение графиков функций (одной переменной), заданных, как правило, таблично, идентификацию полученных на чертеже кривых и вывод сопроводительных текстов. Этим определяется наличие различных типов координатных осей, интерполяций, преобразований (сдвиг, поворот и т.д.), а также наборов маркеров, режимов проведения линии и средств вывода текстовой информации.

Однако, система не является жестко ориентированной на указанный круг задач и допускает расширение без изменения основных правил.

Описание системы построено следующим образом. Выпуск 1 содержит общие сведения о системе и рекомендуется для первоначального ознакомления с ней. Второй выпуск содержит описание основных операторов, в дальнейшие выпуски предполагается включать расширения системы по мере их появления.

Общее описание системы **AMESPLOT**

Один из основных принципов, положенных в основу системы, заключается в том, чтобы смысл операторов и последовательности их выполнения соответствовали обычным действиям человека при ручном построении чертежа. Основываясь на этом, представим себе наши действия при черчении с тем, чтобы на этом примере уяснить ее идеологию.

Для начала мы берем лист бумаги нужного формата

(PAGE)⁺), затем прикладываем в каком месте листа расположить чертеж (ORIGIN) и определяем его размеры (TITLE). Далее изображаем систему координат нужного типа (GRAPH , LGLG и др.) и приступаем к рисованию собственно кривой (CURVE).

Разумеется перед рисованием мы решаем, каким образом будем интерполировать наши числовые массивы (INTRP), как будем помечать исходные точки, если это необходимо (MARKR), а также будет ли линия сплошной, пунктирной или ее вообще не будет (STLIN).

Впрочем, так мы поступаем в тех случаях, когда нужно сделать что-то экзотичное, в обычных же ситуациях рука сама соединяет точки сплошной линией, использует линейную интерполяцию и т.д., т.е. мы действуем согласно собственным привычкам.

Система также имеет свои "привычки" и, если мы явно не прикажем ей пользоваться чем-либо, она будет использовать соответствующие стандартные средства (BEGRPL).

Нарисовав некоторое количество кривых в построенной системе координат, мы можем мысленно изменить тип одной или обеих осей (DUBAX) с тем, чтобы совместить на одной картинке несколько различных графиков и вновь заняться рисованием.

При этом можно оставить некоторый участок чистым от построений, имея ввиду, например, расположить в нем текст.

Наконец, мы можем закончить построения в этой системе координат (ENDGR) и перейти к новой, расположенной в другом месте нашего листа, проделав действия, аналогичные предыдущим.

Обратим внимание на некоторые детали процесса построения чертежа.

Если мы неправильно выбрали оси, то может оказаться, что изображаемая кривая выходит за пределы не толь-

⁺ Здесь и далее в скобках стоят названия операторов системы, выполняющие действия аналогичные описываемым.

ко координатной системы, но и листа, естественно, что в сложившейся ситуации пропускаем часть кривой, не укладываемую в лист. Так же поступает и система, игнорируя всякие попытки построить что-либо вне листа. Далее, все свои действия мы выполняем в определенном порядке, который не определяется нами явно при ручном построении, но тем не менее определенно существует. При использовании автоматизированных методов рисования роль его становится более очевидно. В самом деле, не определив размеры листа бумаги, мы не можем указать на нем положение рисунка (без риска опростоволоситься), а не решив, в какой системе (полярной, линейной) мы будем рисовать график, нам попросту неизвестно, каким образом использовать данные для нанесения их на чертеж. Таким образом, для того, чтобы перейти к следующему этапу построений, нужно иметь некоторую совокупность сведений, необходимых для этого нового этапа. Кроме того, очевидно, что выполнив один этап и перейдя к следующему, нельзя изменить некоторые параметры, определенные на первом этапе.

Действительно, изменение размера листа после того, как мы уже изобразили координатную систему, может быть приравнено к смене листа бумаги при ручном рисовании, что довольно любопытно, в системе же это может привести к не менее занятным результатам. В связи с этим введено понятие уровня (этапности) и все средства системы могут использоваться только из разрешенных уровней.

Система может находиться на одном из трех уровней.

На первом уровне разрешено изменять размер листа и положение координатной системы, но запрещены какие бы то ни было операции с графопостроителем.

Перейдя на второй уровень, мы тем самым фиксируем выбранный значе. ия и может определять координатную систему. Здесь же мы уже можем производить некоторые построения, такие как вывод рамок, текста, поскольку определена точка, относительно которой производятся все построения.

На третьем уровне возможен вывод кривых, т. к. основные параметры к этому моменту уже определены.

Всякая попытка использования какого-либо оператора с запрещенного для него уровня игнорируется системой, а это может привести к тому, что весь остаток субчертежа пропадет.

Заканчивая общее описание системы **AMESPLOT**, определим некоторые понятия, используемые в дальнейшем.

Чертежом в системе мы будем называть совокупность всех координатных систем, попадающих на заданный лист.

Каждая такая система образует субчертеж. Поле субчертежа представляет собой прямоугольную область, в которой изображаются координатные оси (рисование возможно и вне поля субчертежа).

Субчертеж называется простым, если он составлен из одной пары координатных осей, называемых первичными, или сложным, в котором на первичные оси наложены и вторичные. Вторичные оси подменяют соответствующие первичные, но на субчертеже не изображаются.

В пределах каждого субчертежа фиксируется точка, которая называется физическим началом координат (ФНК) и все построения производятся относительно этой точки (ФНК совпадает с левым нижним углом поля субчертежа).

Символьческим началом координат (СНК) мы будем называть текущую точку, т.е. ту точку, которая в данный момент наносится на чертеж.

AMESPLOT - программа

Последовательность действий по выводу информации определяется цепочкой выполняемых операторов, которые составляют **AMESPLOT** — программу.

AMESPLOT — программа может состоять из нескольких сегментов, расположенных текстуально не обязательно подряд. Связь между сегментами осуществляется через общий информационный массив **PS**. Сегмент **AMESPLOT** — программы представляет собой последовательность операторов системы, оформленную следующим образом:

а) для программ, написанных на языке Автокод или в машинных кодах, в виде обращения к стандартной программе:

28	16	28 + 1	7501	7610
	52	PS	0500	—
	77	—	—	—

последовательность AMESPLOT-операторов в машинном формате,

строка с кодом операции 77 является признаком окончания сегмента;

б) для программ, написанных на входном языке транслятора ТА-2М, в виде оператора неопиываемой процедуры cod^+ :

```
<сегмент AMESPLOT - программы>::cod('plot', <PS>,
  <последовательность AMESPLOT - операторов>, 'fin')
<последовательность AMESPLOT - операторов>::<AMESPLOT -
оператор>|
```

```
  <последовательность AMESPLOT - операторов>, <AMESPLOT -
оператор>
```

```
<PS>::<массив >
```

```
<AMESPLOT - оператор>::<begpl>|...|<endgr>.
```

Массив связи PS длиной 40₁₀(50₈) выделяется программистом. Изменять содержимое этого массива между работой динамически смежных сегментов нельзя, т.к. может нарушиться логическая связь между отдельными операторами.

Операторы системы AMESPLOT

AMESPLOT —оператор в машинном коде изображается согласно следующим правилам:

- 1) оператор начинается с ячейки, коп которой $\neq 0$ (коп идентифицирует оператор);

⁺) Следует помнить, что оператор процедуры может иметь не более 127 параметров.

- 2) информация о его параметрах (если таковые имеются) располагается в адресной части этой ячейки слева направо и (если параметров больше 3-х) сверху вниз, занимая адресные части следующих подряд идущих ячеек (с нулевыми КОПами), количество которых ограничивается форматом этого оператора или, если опущено некоторое количество последних (текстуально) параметров - первой ячейкой следующего оператора (т.е. первой ячейкой, КОП которой $\neq 0$);
- 3) информация об одном параметре занимает адрес и соответствующий ему признак π_i (т.е. все адреса модифицируются по РА).

ПРИМЕР

Для иллюстрации приведенных правил рассмотрим пару операторов из некоторой АМЕСПЛОТ -программы:

```
title (x, y, s1, n1, s2, n2, s3, n3)
cross
```

(оператор cross - без параметров).

В машинном виде это может выглядеть так:

π	04	x	y	s1	}	title
π	00	n1	s2	n2		
π	00	s3	n3	-		
π	00	A1	A2	A3		
π	05	-	-	-		cross

В этом случае четвертая ячейка будет пониматься как пустой оператор, расположенный* между операторами title и cross (этим приемом можно пользоваться для резервирования памяти внутри сегмента АМЕСПЛОТ -программы. Если $s3 = n3 = 0$, то запись оператора можно сократить т.о.:

π	04	x	y	s1	} title
π	00	n1	s2	n2	
π	05	-	-	-	

если же и $n1 = s2 - n2 = 0$, то можно не писать и вторую строку, однако в этих случаях между операторами **title** и **cross** не должно быть пустого оператора, т.к. в противном случае он будет понят как продолжение оператора **title** .

О ш и б к и

В системе предусмотрен довольно полный контроль выполнения тех ограничений, которые налагаются на операторы и их параметры. Нарушение этих ограничений фиксируются как ошибки и сообщения о них выдаются на АЦПУ в следующем виде:

NN оператор (**А А А А А А**) содержит ошибку $\begin{matrix} -TT...T \\ TT...T \\ \vdots \\ TT...T \end{matrix}$

где **NN** - двузначный порядковый номер оператора в текущем сегменте **AMESPLOT** -программы (первый оператор сегмента имеет номер 01),

А А А А А А - наименование этого оператора,

TT...T - текст, идентифицирующий зафиксированную ошибку (таких текстов может быть выдано несколько подряд).

К ошибочным ситуациям приводят следующие нарушения:

1. Выполнение оператора с запрещенного уровня
текст: нарушение уровня,
2. Нарушение ограничений, накладываемых на i -ый параметр
текст: ошибка в i параметре,

3. Неправильное использование оператора DUBAX текст: оси несовместны.

Если при выполнении оператора возникает ошибочная ситуация, то его действие эквивалентно действию пустого оператора (исключение составляет один оператор **VEGRL**). При анализе ошибок следует иметь в виду, что возникновение ошибочной ситуации при выполнении некоторого оператора может быть обусловлено другими, ранее выполнявшимися операторами.

ПРИМЕР

Рассмотрим в качестве иллюстрации **AMESPLOT** -программу для вывода чертежа, изображенного на рисунке. Для наглядности запишем ее в виде билстинга, где левая часть будет представлять программу в машинных кодах, а правая в операторной форме. Условимся также идентификаторами в левой части билстинга обозначать адреса значений, а в правой, в позициях соответствующих параметров, по возможности ставить сами значения, исключая случаи, когда в качестве параметров выступают массивы числовых значений, некоторые из которых, впрочем, мы тоже укажем в пояснении к программе.

I6	* +I	750I	76I0		
52	PS	0500	-		
0I	x	y	-	I	BEGDL (I,I)
02	lx	ly	-	2	PAGE (290,240)
03	px	py	-	3	ORIGN (40,50)
04	dx	dy	S1	4	TITLE (90,I40,
00	0006	S2	0005		'TEST-I,'G'AXS-x',5,'AXS-y'..)
00	S3	0005	-		
I4	0000	-	-	5	FRAME (0)
I4	000I	-	-	6	FRAME (I)
05	-	-	-	7	CROSS
06	0002	βI	-	8	BLANK (2, βI)
I4	0004	-	-	9	FRAME (4)
I0	xH	xk	yH	I0	GRAPH(-I,0, I.5, -3.5,
					3.5)
00	yH	-	-		
24	000I	-	-	II	MARKR (I)
20	0004	α	-	I2	INTRP (4,3) α = 3
33	azq	fun	n	I3	CURVE (azq, fun, 20,
					0,0,3)
00	0000	0000	h		
07	0002	-	-	I4	RSBLK (2)
40	S4	00I0	xI	I5	MESS ('(проба)',10,820)
00	yI	-	-		
40	S5	00I4	x2	I6	MESS('ABCDEFGHLMNW',
					I4, 8, I5)

-	y2	-	-		
40	S6	00I3	x3	I?	MESS ('(TEKCT-TEKCT)', I3,I0,)
00	y3	-	-		
14	000C	f r	-	IF	FRAME (E, f z)
10	S'	0022	x4	IS	MESS ('(AMESPLOT) - сн- стема',20,I22,20)
00	y4	-	-		
10	S.	00I6	x5	20	MESS ('(HELP:MI _ I9'2)I,)', I6,I22,5)
00	y5	-	-		
52	-	-	-	2I	ENDGR
77	-	-	-		
16	*+1	750I	76I0		
52	ps	0500	-		
03	lx1	lyI	-	I	ORIGN (160,90)
04	dxI	dyI	S9	2	TITLE (I10,20,'TEST-2',6 'T',1,'F/T/-100-COS/T#T',17)
00	0006		000I		
00	SIO	002I	-		
14	000I	-	-	3	FRAME (I)
I2	xHI	TX	yHI	4	LGLG (0.0I,40,I,30)
00	Ty	-	-		
20	0002	m	-	5	INTRP (2,3)
33	xm	ym	n	6	CURVE (xm, ym, I5,0, 0,3)
00	0000	0000	h		
77	-	-	-		

Наша программа разбита на два сегмента (хотя это не обязательно), между которыми может располагаться, например, программа вычисления функции для второго субчертежа.

Оператор `VEGPL` выполняется первым, он заносит набор стандартных параметров в массив связи `PS` и располагает чертеж на расстоянии 1 мм по оси x и y от нулевой точки графопостроителя.

Следующая пара операторов устанавливает новые размеры чертежа и значение ФНК.

Оператором `TITLE` мы задаем длины горизонтальной и вертикальной осей, а также выводим заголовки осей и субчертежа. Система переводится на второй уровень.

Далее рисуем рамку чертежа и субчертежа, определяемого оператором `TITLE`.

Восьмой оператор резервирует область субчертежа с координатами (в мм) относительно ФНК

$$0 \leq x \leq 70, \quad 0 \leq y \leq 40$$

которые заданы массивом `B1`:

```
B1 : 0
B1+1 : 70
B1+2 : 0
B1+3 : 40
```

Закключаем в рамку эту область девятым оператором. Оператором `GRAPH` специфицируются координатные оси как линейные, причем горизонтальная ось соответствует интервалу изменения переменной от -1.5 до $+1.5$, а вертикальная от -3.5 до $+3.5$. Этот же оператор рисует оси с разметкой т.о., что точка пересечения совпадает с началом координат, поскольку раньше был выполнен оператор `CROSS` и нули обеих осей попадают на область субчертежа. Система переходит на уровень 3.

Операторами `MARKR`, `INTRP` задаем соответственно тип маркера и интерполяцию. Эти средства используются оператором `CURVE`, который выводит кривую, заданную массивами аргументов и значений функции в 20 узлах. Мас-

сивы расположены в нулевом кубе. Каждая третья опорная точка (начиная с первой) маркируется.

Следующим оператором разрешаем рисование в заблокированной ранее области и далее располагаем в ней три строки текста операторами MESS .

Восемнадцатый оператор рисует рамку, охватывающую прямоугольную область, заданную массивом $\{z\}$, который имеет ту же структуру, что и массив $\{1\}$ из восьмого оператора, в которой также выдается текст.

И наконец, последним оператором первого сегмента является оператор перехода на уровень 1 - ENDGR . Этот оператор использован нами для перехода к построению нового субчертежа и заданию соответствующих параметров, которые не могут быть определены на третьем уровне. Оператор ENDGR в массиве связи P5 изменяет только значение уровня, остальные параметры имеют те же значения, что и перед выполнением оператора ENDGR . Сегмент заканчивается признаком конца (команда СТОП).

Второй сегмент продолжает нашу программу. Оператором ORIGN задаем новое значение ФНК. TITLE определяет область субчертежа, выводит заголовки и переводят систему на второй уровень. Заметим, что в заголовке вертикальной оси присутствует символ = , который на чертеже заменен пробелом (см. MESS). Заклучив субчертеж в рамку, описываем координатные оси. Несмотря на то, что в свое время выполнялся оператор CROSS , точка пересечения осей совпадает с ФНК.

Пятым оператором мы меняем интерполяцию, а шестым выводим кривую, определяемую содержимым массивов x_m и y_m . Поскольку тип маркера изменен не был, а маркировка требуется, то рисуется тот же маркер, что и на предыдущем субчертеже.

Проиллюстрируем правила записи AMESLOT -программы на входном языке транслятора TA - 2M на том же примере. Первый сегмент нашей программы может выглядеть так:

1. cod ('plot', ps, 'begpr', 1, 1, 'page', 290, 210, 'origin', 40, 30, 'title', 90, 40, 51, 6, 51, '0', 1, 5, 51, '0', 2, 5);
2. cod ('frame', 0, 'frame', 1, 'cross', 'blank', 2, 81);
3. cod ('frame', 4, 'graph', a, 1, 5, 6, 3, 5); ...
4. cod ('mess', 58, 15, 122, 5, 'endgz', 'fin');

Каждый из записанных здесь операторов процедуры `cod` не является сегментом `AMESPLOT` — программы.

Так первый оператор `cod` породит в готовой программе последовательность команд, аналогичную первым восьми командам, изображенным на биллистинге, а четвертый — последним четырьмя из первого сегмента. Т.о., если между этими операторами не будут расположены другие операторы АЛГОЛИА, то после трансляции будет получен правильный сегмент `AMESPLOT` — программы. Укажем на некоторые детали приведенной выше записи:

1) `ps`, `s1`, `s1`, `58` — описаны в АЛГОЛ-программе как массивы,

2) `s1[1]` — (если массив описан как `array s1[1:n]`) — содержит заголовок субчертежа,

`s1[2]` — заголовок оси x ,

`s1[3]` — заголовок оси y , что указано с помощью идентификаторов.

3) первый и третий параметры оператора `'graph'` не могут быть заданы числами, т.к. их пришлось бы изобразить числами со знаком, а такая конструкция запрещена синтаксисом процедур-кодов, и поэтому в позициях, соответствующих этим параметрам, стоят идентификаторы простой переменной или массива.

Второй сегмент можно записать т.о.:

```
cod ('plot', ps, 'origin', 160, 90, 'title', 110, 80, 59, 6,
    59, '0', 1, 1, 59, '0', 2, 17, 'frame', 1,
    'lglg', 0, 01, 40, 1, 35, 'intzp', 2, 3,
    'curve', xm, ym, 15, 3, 'fin');
```


или имея ввиду вышесказанное:

```
cod ('plot', ps);  
cod ('origin', 160, 90);  
cod ('curve', xm, ym, 15, 3);  
cod ('fin');
```

И наконец, последнее замечание, которое касается записи оператора CURVE : из него удалены параметры, указывающие номер куба, в котором располагаются массивы, поскольку номер куба считается заранее известным и равным 1.

ОРГАНИЗАЦИЯ РАБОТЫ С СИСТЕМОЙ AMESPLOT

Для своей работы система требует:

- 1) расширенное РП,
- 2) АЦПУ,
- 3) графопостроитель или устройство вывода на перфоленту,
- 4) а при использовании системы ИС-2 еще и вторую четверть МБ-О, на которую специальной картой переписываются программы системы AMESPLOT.

Расширение РП интерпретирующей системы необходимо постольку, поскольку AMESPLOT реализована как набор стандартных программ в ИС, которые вызываются для выполнения на РП. Конкретное значение длины РП зависит от операторов, исполняемых программистом, и может быть вычислено точно с помощью таблицы длин СП (выпуск 2, приложение 2). В качестве максимального может быть задано РП длиной 1400_g ячеек.

В том случае, когда программа пишется в машинных кодах, программист должен при распределении памяти учитывать требования системы обычным образом. Для программ, написанных на АЛГОЛЕ, выделение памяти под РП достигается использованием предварительного комментария для исполнения (который располагается в программе перед пер-

бым begin), например:

```
comment to 006^00MOZ;
```

Это указание резервирует МОЗУ, начиная с ячейки 6000, но не расширяет РП ИС, что в свою очередь достигается выполнением оператора.

```
cod ('is-zp', '6000').
```

Графопостроитель

Графопостроитель ДРП-3м представляет собой аналоговое двухкоординатное устройство отображения информации, которому придан цифро-аналоговый преобразователь УП-7. Он (ДРП) позволяет выводить чертежи, размеры которых не превышают $800 \times 800 \text{ мм}^2$. Пишущий элемент способен перемещаться в поднятом или опущенном состоянии из текущей точки (т.е. той точки, в которой он находится в данный момент) в точку, координаты которой поданы на устройство управления, по прямой, соединяющей эти точки.

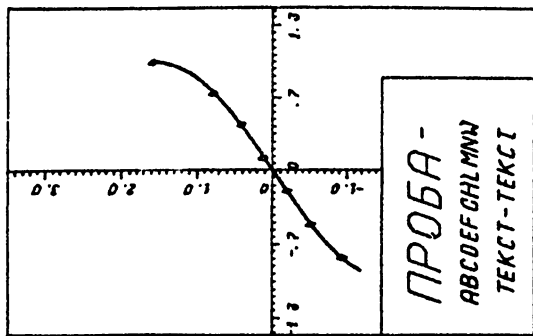
ДРП-3м может работать в двух режимах:

- 1) непосредственно от ЭВМ,
- 2) автономно вводя данные с перфоленты (на которую в этом случае выведены результаты работы AMES PLOT -программы).

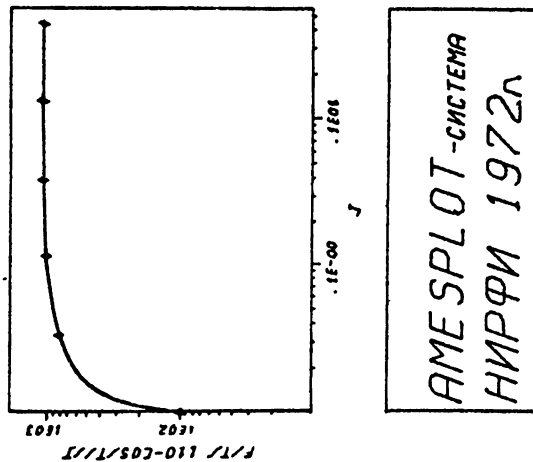
Реальные технические характеристики ДРП-3м таковы, что улучшение качества рисунков достигается за счет уменьшения максимальных размеров чертежа, и в силу этого за максимальные нами приняты размеры $400 \text{ мм} \times 400 \text{ мм}$.

Разрешающая способность устройства ограничивается в этом случае конструкцией пишущего элемента и практически равна $\sim 0.5 \text{ мм}$.

TEST-1



TEST-2



АКС-Х

ЛИТЕРАТУРА

1. Harschsohn I.
AMESPLOT—a higher level data
plotting software system.
Communs ACM, 1970, 13, n.9,
546-555.
2. Наймарк Б.М.
Программная система высшего уровня AMESPLOT
для построения графической информации.
Э.И., В.Т. 1971, № 2.
3. Баяковский Ю.М., Михайлова Т.Н.
Автокод для М-220, БЭСМ-4.