

XYプロッタ用アプリケーションプログラムの開発

林 隆一*・片岡 新二**・山崎 格***

Ryuichi HAYASHI, Shinji KATAOKA and Itaru YAMASAKI
Development of Application Programs for XY-plotter

Abstract: A XY-plotter is placed in the Computer Center of Shimane University, but it is not used too much.

In this paper the application programs are described which were developed to heighten the utility value of the XY-plotter. On the occasion of developing, we wanted to make full use of the system subroutines made by the maker. So a reverse assembler was produced and their system subroutines were interpreted.

The developed programs are as follows:

- (1) the subroutine which automatically draws a graph on the orthogonal co-ordinates without considering at all the values for data.
- (2) the group of subroutines which draws a graph on the logarithmic co-ordinates.

1. はじめに

島根大学計算機センターの FACOM 270-20 には XYプロッタが設置されており、FORTRANの CALL¹⁾文で呼び出せる XYプロッタ用副プログラムが11種類用意されている。しかし、利用者は限られており、大部分は計算機センター室員が相談を受けプログラミングしているようである。この原因としては、メーカから提供された XYプロッタ用副プログラムを使いこなすには相当の経験があるということ、アプリケーションプログラムが不足していることが考えられる。

そこで、筆者等は XYプロッタの利用価値を高めることを意図して、XYプロッタ利用のためのアプリケーションプログラムを開発した。開発にあたっては、メーカ製の XYプロッタ用システムサブルーチン²⁾をベースにすることにしたが、システムプログラムは機械語の形で提供されるのがふつうで、その解説書をメーカに要求してもまずもらえないのが現状である。そのため、XYプロッタのアプリケーションプログラムを開発するに先立って、逆アセンブラを作成してメーカから提供されているシステムサブルーチンを解読した。

以下の章で各プログラムを詳述するが、その名称と大略の特徴は次の様である。

(1) 逆アセンブラ

命令語と数値語をできうる限り区別する。また番地指定を名前に表示するなど解読し易い形式で出力する。

(2) GRAPH 副プログラム

直交座標上に作図するプログラムで、与えられたデータを効果的に表示するような座標と目盛を自動的に刻む。

(3) LGSCAL副プログラム

与えられたデータの対数値とペンの移動の物理的長さとの関係を求める。

(4) LGAXIS 副プログラム

対数軸の目盛を刻み、指定された間隔で数値付けをし、軸ラベルを書く。

(5) LGLINE 副プログラム

与えられたデータから片対数あるいは両対数の曲線をプロットする。

2. 逆アセンブラ

今日の様に計算機を利用するものが増え、その内容も多種多様になってくると、メーカから提供されたシステムプログラムに満足しきれなくなって拡張・改良の必要

* 島根大学教育学部技術研究室

** 大阪市立南楼中学校

*** 新見市立熊谷中学校

S1=IBDC

0000	0000	S1;	BSS,	1:
0001	3931		STX.R,	*1,V1:
0002	5835		STD.R,	V2:
0003	5036		ST.R,	V3:
0004	F004		SKP.S,	M:
0005	F805		B.R,	P1:
0006	D090		SR.S,	16:
0007	7032		S.R,	V3:
0008	5031		ST.R,	V3:
0009	4036		L.R,	C2:
000A	F801		B.R,	P2:
000B	4035	P1;	L.R,	C3:
000C	502E	P2;	ST.R,	V5:
000D	40F2		L.R,	S1:
000E	5001		ST.R,	V6:
000F	3500		LX.L,	*1,0:
0010	0000	V6;		
0011	3903		STX.R,	*1,V7:
0012	F901		AX.S,	*1,1:
0013	3922		STX.R,	*1,V8:
0014	4400		L.L,	0:
0015	0000	V7;		
0016	5008		ST.R,	V9:
0017	3105		LX.S,	*1,5:
0018	4021		L.R,	V3:
0019	D8D0	P4;	SRAD.S,	16:
001A	8824		D.R,	C6:
001B	501E		ST.R,	V3:
001C	D810		SLD.S,	16:
001D	B01F		OR.R,	C7:
001E	5500		ST.L,	*1,0:
001F	0000	V9;		
0020	F9FF		AX.S,	*1,-1:
0021	F800		B.R,	P3:
0022	4017	P3;	L.R,	V3:

Fig. 1 Execution example of a reverse assembler

にせまられる。しかし、システムプログラムは公表されないのがたてまえになっているので、機械語からアセンブラ言語へ変換して出力する逆アセンブラを作成して、システムプログラムを解読しているようである。****

逆アセンブラ作成の際に最も問題となるのは、命令語と数値語（定数）をどのようにして区別するかということである。多くの逆アセンブラでは、すべてを命令語とみなして、命令語にあてはまらないものだけを数値語として区別している。しかし、このような方法ではほとんどの数値語は命令語として変換されてしまう。今回は、この問題を解決するために、置数命令、算術演算命令、論理演算命令の番地部で指定される番地は命令語とは考えられないので、その指定された番地を記憶しておき、そこを逆アセンブルするときには数値語として取り扱うという方法で命令語と数値語を区別した。

また、アセンブラ言語でプログラミングするときは、作業用番地の名前にしても、プログラム中の番地の名前にしても、その意味を示唆する適当な名前を使うことが多いが、逆アセンブラではそこまで望むわけにはいかな

い。しかし、その番地の性格はある程度判断できるので、その意味を表わす文字のあとに連番号をつけた名前によって番地を指定した。

これらの基本的な考えのもとに作成した逆アセンブラによって、2進10進変換のシステムサブルーチン（IBDC）を逆アセンブルした一部を Fig. 1 に示す。左側から相対番地（16進数）、機械語（16進数）、逆アセンブル結果である。番地指定の名前は、

プログラム・ポイント	P 1, P 2, ……
定数	C 1, C 2, ……
変数	V 1, V 2, ……
サブルーチンのエントリ・ポイント	S 1, S 2, ……

で表示している。

前述した基本的な操作のほか、次の様なことも考慮している。

逆アセンブルは最初から1語づつ順番に行なうので、

**** 逆アセンブラに関して個人的なコミュニケーションはあるが、公表された例^{3),4)}はあまり知らない。

定数を参照する命令がその定数よりも前にある場合は問題ないが、後にあるときはその定数をすでに命令語として逆アセンブルしているため (Fig. 2 (a) 参照)、定数に訂正しなければならない。この訂正を行なうために、最初から順番に一応逆アセンブルした後、再度最初からチェックをし、上の様な場合は定数に訂正している (Fig. 2 (b) 参照)。その際、命令語として逆アセンブルされていたものを定数に訂正したために他の部分に影響がでてくることもあり、命令語を定数に、定数を命令語に訂正する繰返しの操作も必要に応じて行なっている (Fig. 2 (c) 参照)。

<pre>V5; BSS, 1: AX. I, *3, 48: C7; L. R, V5: OR. R, C7:</pre>	→	<pre>V5; BSS, 1: AX. I, *3, 48: C7; DN, 48: L. R, V5: OR. R, C7:</pre>
(a)		(b)

<pre>V5; BSS, 1: AX. I, *3, 48: C7; L. R, V5: OR. R, C7:</pre>	→	<pre>V5; BSS, 1: DN, -127: C7; DN, 48: L. R, V5: OR. R, C7:</pre>
		(c)

Fig. 2 A method of correcting to constant from instruction code

3. XYプロッタ用アプリケーションプログラム

3.1 GRAPH副プログラム

XYプロッタを制御するときは、ペンの移動の物理的長さや計算機の数値との関係をまず定めなければならない (以下 *scaling* と呼ぶ)。メーカ製のXYプロッタ用各種副プログラムを組み合わせ直交座標上に作図するには、座標軸を描くための *scaling* と計算結果をプロットするための *scaling* を独立に行なわねばならない。これは利用者、特に初心者にとって非常に煩雑でプログラミング・ミスの主な原因となっている。また、*scaling* するためには当然計算結果の値は予測できなければならないが、予測できない場合もしばしばある。その際には、一度計算結果をラインプリンタ等へ出力してから、*scaling* したり最大軸目盛や最小軸目盛を決めるなどのXYプロッタ用データを設定し、その後XYプロッタに図を描く目的のみで再度同一処理を行なわなければならない。

そこで、筆者等は以下の点に留意したXYプロッタ用副プログラム **GRAPH** を開発した。

- (a) データの数値の大きさを一切考慮することなしに、自動的に直交座標上に図を描くことができる。
- (b) この副プログラムを利用する際には、副プログラムを呼び出す **FORTRAN** の **CALL** 文とその引数のみの1行の文で足りること。

次に、**FACOM 270**—**20**の命令語としては1語長命令 (**R, S**形式) と2語長命令 (**L, I**形式) があり、2語長命令の番地部 (2語目) は絶対番地をあらわす。しかし、システムサブルーチンでは、**location independent** でなければならないという性質から、一般に2語長命令の番地部の数値は実際に参照する番地を指定していない。更に、インデックス修飾で番地を指定する場合にも、インデックスレジスタの内容がわからないかぎり実際の番地を確定できない。従って、2語長命令とインデックス修飾のときには番地部の数値をそのまま10進数で出力している。

(c) オペレータに負担をかけないこと。

(d) 占有記憶容量をできる限り小さくすること。

(c), (d) を満足させるために、この副プログラムをアセンブラ言語でシステムサブルーチンとして作成し、2次記憶 (ドラム) に登録するようにした。GRAPH の仮引数の種類と並びは次のとおりである。

(XPOINT, YPOINT, X, Y, N, XBCD, YBCD, XN, YN, XL, YL)

XPOINT, YPOINT; 作図したいXYプロッタ上の位置を指定する引数で、左下端の x, y 座標を実数 (単位 mm) で表わす。1回目の呼び出しではXYプロッタの中心からの距離、2回目以降は前回の XPOINT, YPOINT からの距離である。

X, Y; x, y 座標のデータが格納されている実数型配列名を指定する。配列の大きさは (データの個数 + 2) 用意しておく。

N; プロットすべきデータの個数を整数で指定する。

XBCD, YBCD; x, y 軸に軸ラベルとして書きたい文字を **nH** 型文字定数で指定する。

XN, YN; XBCD, YBCD で指定した文字の数を整数で指定する。XN(YN) > 0 ならば目盛とラベルは軸の正の側に書き、XN(YN) < 0 ならば負の側に書く。

XL, YL; 描きたい図の大きさ、すなわち x 軸, y 軸の長さを実数で指定する。単位は mm である。

なお、定数、文字定数を指定する引数では、それらが格納されている変数または配列名を書いてもよい。

Fig. 3 はこの副プログラムの流れ図である。以下に

流れ図を詳述するが、データが同符号の場合には正だけを考えるものとする。

プログラムはまずデータの最大値、最小値をみつけ、ペン1mm当りのデータの大きさ(以下 DELTA と称す)を計算する。データが同符号ならば0から最大値、異符号ならば最小値から最大値までのデータの間隔を軸の長さで割ったものを仮の DELTA とする。

次に、軸目盛の刻み幅としては約20mmが適当であろうという考えのもとに、DELTA に20を掛けた値を $a \times 10^N$ ($0 < a < 10$, N は整数)の形に変形し、軸目盛の単位刻み幅当りのデータの大きさを次の様に決定する。

$0 < a < 1.5$ ならば	1×10^N
$1.5 \leq a < 3.5$ ♪	2×10^N
$3.5 \leq a < 7.5$ ♪	5×10^N
$7.5 \leq a < 10$ ♪	$1 \times 10^{N+1}$

軸目盛の刻み幅が決定すると、データのおさまる最大軸目盛および最小軸目盛を求める。ただし、同符号のデ

ータでは最小軸目盛を0とする。しかし、仮の DELTA を用いて最小軸目盛から最大軸目盛までを描いたのでは、与えられた軸長とはならないので、最小軸目盛と最大軸目盛の間隔を軸の長さで割ったものを真の DELTA とする。

更に、このままでは、データが同符号のときには最小軸目盛を0としているので、データをプロットした曲線が右端に片寄るおそれもある。これをさけるために、データの最小値の位置が2分した軸の右側になる場合には、データの最小値から適当な最小軸目盛を決定し、この値から DELTA を計算し直し、それを真の DELTA とする。

真の DELTA が決定すれば、軸を描き、データをプロットする。軸目盛の数値付けは、軸目盛の刻み幅の係数(1, 2あるいは5)を1目盛づつ加えながら最大軸目盛まで書き、最後に位取りを $\times 10^N$ の形で書く。

データのプロットにあたっては、軸目盛が0をもつな

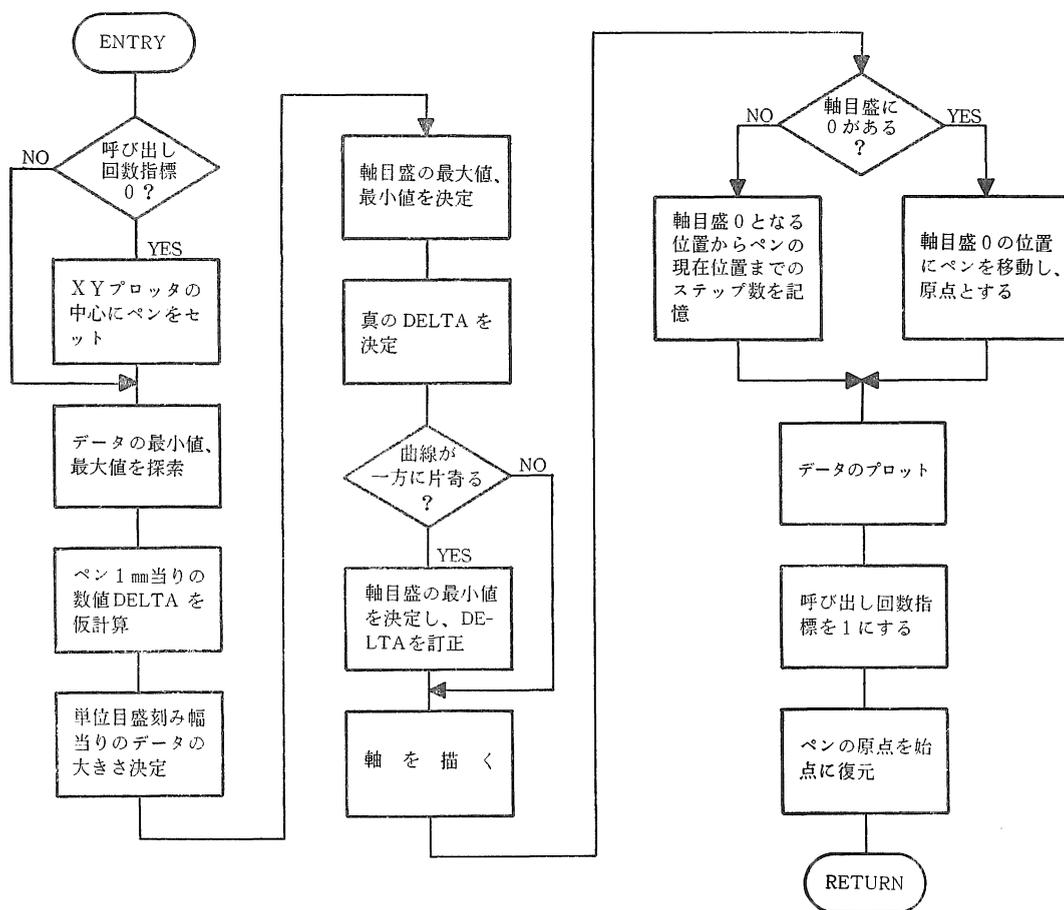


Fig. 3 Flow chart of GRAPH subroutine

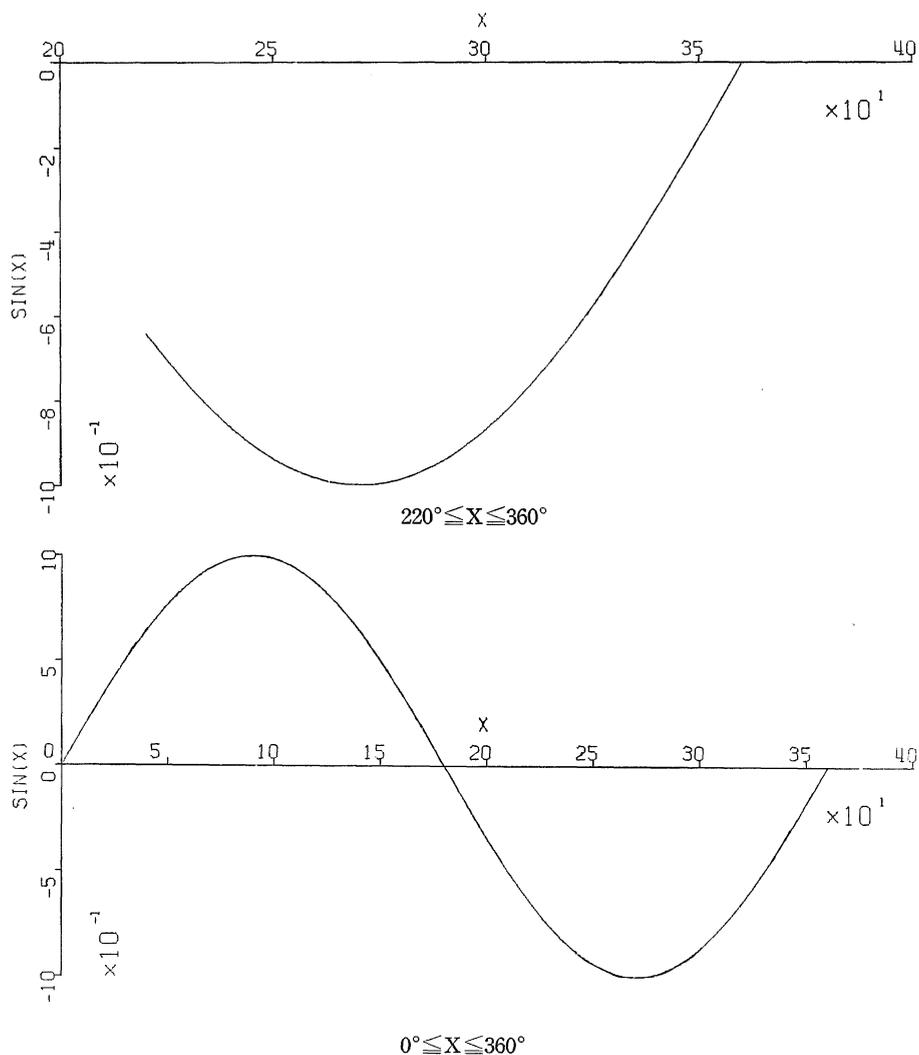


Fig. 4 Exemplar of plotting a sine curve by GRAPH subroutine

らば、ペンをそこまで移動した後その座標を原点としてプロットすればよい。一方、軸目盛に0がない場合（すなわち、曲線が片寄る場合）には、軸目盛が0となる位置を求めてそこをペンの原点にしなければならないが、その点までのペンの移動量が非常に大きくなることもありうる。そのため、ペンを移動させず、軸目盛が0となる位置から現在のペンの位置までのステップ数をペンの現在位置記憶番地に入れて、データをプロットする。

呼び出し回数指標を設けたのは次の理由からである。同一の JOB で何度もこの副プログラムを呼び出す際の XPOINT, YPOINT の指定には、前回のこれらの点からの距離を用いるのが合理的である。しかし、1回目はどこかに基準を設けなければならない。その基準を XY

プロットの中心にしたため、1回目の呼び出しと2回目以降との区別が必要である。*****

データのプロットが終了すれば、ペンの原点を XPPOINT, YPOINT に返し次の呼び出しに備える。

Fig. 4 に GRAPH 副プログラムを利用して描いた正弦曲線の作図例を示す。

3.2 対数用副プログラム

メーカーから提供された XYプロッタ用副プログラムでは、SCALE, AXIS, LINE 等のベーシック副プログラム¹⁾を組み合わせて直交座標上に作図する。常用対数座標上

***** GRAPH をセグメンテーションしたときは、呼び出し指標0で主記憶にロードされるので、常にXYプロットの中心からの距離を指定しなければならない。

に作図したいときに、これらの副プログラムと同じ働きをさせることを目的として開発したのが LGSCAL, LGAXIS, LGLINE 副プログラムである。これらの対数用副プログラムも 3.1 で述べた理由によりシステムサブルーチンとして作成した。

以下に各プログラムの要点を述べ、仮引数の種類と並びを示す。

3.2.1 LGSCAL 副プログラム

この副プログラムは scaling のための副プログラム SCALE に対応するもので、対数用 DELTA (3.1参照) を計算する。

仮引数は次のとおりである。

(X, S, N, K, MIN, MAX)

X; データの格納されている実数型配列名を指定する。

S; 実数で軸の長さを指定する。単位は mm である。

N; プロットすべきデータの個数を整数で指定する。

K; プロットすべきデータの間隔を整数で指定する。

MIN; 実数で軸の始点における値、すなわち 1 番始めに付けられる目盛の数値を指定する。ただし、この数値は $\log_{10} \text{MIN}$ が整数となる値でなければならない。

MAX; 実数で軸の終点に付ける目盛の数値を指定する。ただし、 $\log_{10} \text{MAX}$ は整数でなければならない。

LGSCAL 副プログラムは、データの最小値をみつけて X ($N * K + 1$) に格納し、

$$(\log_{10} \text{MAX} - \log_{10} \text{MIN}) / S$$

を対数用 DELTA として X ($N * K + K + 1$) に格納する。したがって、配列の大きさは X ($N * K + 2 K$) 用意する必要がある。

3.2.2 LGAXIS 副プログラム

この副プログラムは軸を描く副プログラム AXIS に対応するもので、対数軸の目盛を刻み、指定された間隔で数値付けをし、軸ラベルを書く。

仮引数は次のとおりである。

(XPOINT, YPOINT, BCD, NBCD, S, THETA, MIN, INTVAL, X, N, K)

XPOINT, YPOINT; 実数で軸の始点の x, y 座標 (ペンの原点からの距離) を指定する。単位は mm である。

BCD; 軸ラベルとして書きたい文字を nH 型文字定数で指定する。

NBCD; 軸ラベルの文字数を整数で指定する。NBCD > 0 ならば目盛とラベルは軸の正の側に書き、NBCD < 0 ならば負の側に書く。

S; LGSCAL の S と同じ。

THETA; 実数で軸の +x 方向に対する回転角を指定する。単位は度で +x 方向を 0° とし、反時計回転の方向を正とする。

MIN; LGSCAL の MIN と同じ。

INTVAL; 軸目盛の数値付けをする間隔を整数で指定する。ただし、 $0 < \text{INTVAL} \leq 10$ である。

X; LGSCAL の X と同じ。

N; // N //

K; // K //

3.2.3 LGLINE 副プログラム

LINE 副プログラムに対応するもので、連続して与えられた座標の対数を取りプロットする。引数により片対数、両対数のどちらでも描くことができる。

仮引数は次のとおりである。

(X, Y, N, K, J, L, LOG)

X, Y; x, y 座標のデータが格納されている実数型配列名を指定する。

N; LGSCAL の N と同じ。

K; // K //

J; マークをプロットすべきデータの 間隔を整数で指定する。(詳細は参考文献(1)を参照)

L; 整数でマークの種類を指定する。(詳細は参考文献(1)を参照)

LOG; 整数で対数の種類を指定する。

LOG < 0 y 軸片対数

LOG = 0 両対数

LOG > 0 x 軸片対数

3.2.4 使用法

これらの対数用副プログラムの使用にあたっては、次の点に注意することが必要である。

(a) 原則としては、LGSCAL を引用した後、LGAXIS, LGLINE を呼び出すが、データの最小値と対数用 DELTA をあらかじめ X ($N * K + 1$) と X ($N * K + K + 1$) に格納しておけば、LGSCAL は使わなくてよい。

(b) $\log_{10} 1 = 0$ であることから、LGLINE を呼び出す前に、対数軸方向のペンの原点を次の位置に設定しておかねばならない。

MIN = 1 対数軸の始点の位置。

MIN ≠ 1 対数軸の始点から

$-(\log_{10} \text{MIN} / \text{DELTA})$ だけ移動した位置。

Fig. 5 に $x=y^2$ を両対数として描くプログラムと作図例を示す。

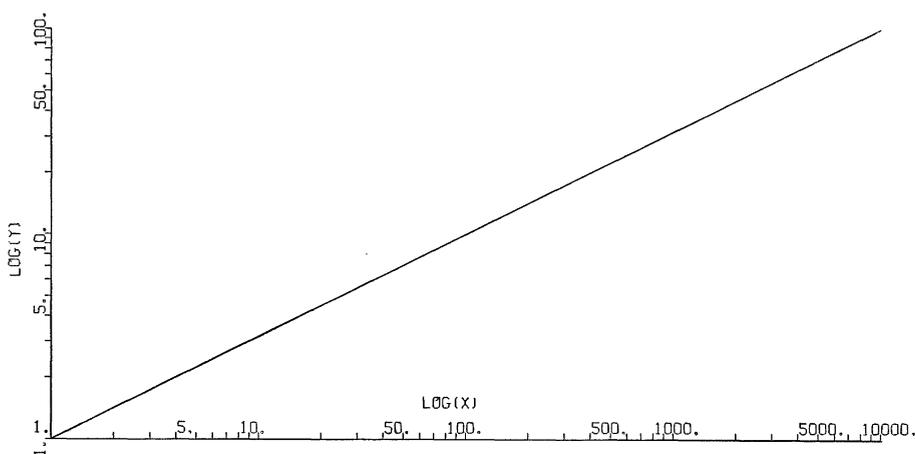
```

DIMENSION X(102),Y(102)
DO 10 I=1,100
  FI=I
  Y(I)=FI
  X(I)=Y(I)**2
10 CONTINUE
CALL SET
CALL PLOT(0.0,0.0,2)
CALL LGSCAL(X,180.0,100,1,1.,10000.)
CALL LGSCAL(Y,90.0,100,1,1.,100.)

CALL LGAXIS(0.0,0.0,6HLOG(Y),6,90.0,90.0,1.0,5,Y,100,1)
CALL LGAXIS(0.0,0.0,6HLOG(X),6,180.0,0.0,1.0,5,X,100,1)
CALL LGLINE(X,Y,100,1,0,42,0)
STOP 9
END

```

(a) A program example



(b) An example of plotting by Fig. 5(a)

Fig. 5 An example of plotting $x=y^2$ by LGSCAL, LGAXIS and LGLINE subroutine

4. おわりに

逆アセンブラの作成にあたっては、逆アセンブル結果を読み易くすることと留意して、番地を名前で表わしたり、名前の種類によって命令語と数値語をできる限り区別した。しかし、2語長命令とかインデックス修飾のある場合には完全には逆アセンブルできなかった。完全な逆アセンブルを求めるならば、プログラムの流れをいかにして計算機に理解させるかという現在未解決の意味論を考慮しなければならない。逆アセンブラの実用的な見地からは、今回の程度の逆アセンブルが可能であれば十分利用できる。というのは、いかに計算機で完全な逆アセンブルをしても最終的には人間がプログラムの流れを解釈せねばならない。したがって、実用上からはプログラムの流れを追っていく途中で簡単に誤りを訂正できればいいわけで、今回の程度の逆アセンブルの誤りの訂正はそれほど困難ではなかった。

XYプロッタ用アプリケーションプログラムは1976年8月より島根大学計算機センターで公開している。

GRAPH副プログラムは、筆者等の要求を満足させる

ものが作成できたが、軸目盛の刻み幅、位取りの文字の大きさと位置が固定であるなどの不便さが指摘されている

対数用副プログラムをベーシック副プログラム群としたのは、利用者はXYプロッタを積極的に利用しようとする者に限られるであろうので、利用者によくの自由度を残しておく方がベターであるという考えからであった。しかしながら、使いにくいという指摘もあるので、GRAPH副プログラムと同様に自動的に作図する副プログラムも検討している。

最後に、終始有益な御助言を頂いた島根大学教育学部野坂弥蔵教授、プログラムのデバッグなどに御協力頂いた島根大学計算機センター堀江昭彦氏に謝意を表わします。

参考文献

- 1) FCOM 270—20/30 FORTRAN 仕様書，富士通(1970)。
- 2) FACOM 270—20/30 FASP 仕様書，富士通(1970)。
- 3) 浜屋進：ソースリスタ“PAM—45”，沼津高専研究報告，No. 8, pp. 33~39 (1973)。
- 4) 森口繁一：システムあばき事始め，bit, Vol. 3, No. 13, 共立出版，pp. 82~87 (1971)。