解説

PLOT-WSX の紹介

小西 龍一 *

1　はじめに

パソコンや EWS 上で図形処理を行う際、大型汎用計算機で所謂、標準となっているカルコンプ・インターフェースの FORTRAN プログラムを移植して、プログラムのデバックや開発をしたい事があります。できるだけプログラムを手直しせずに、カルコンプ互換のあるグラフィック・ライブラリが有れば便利です。
ここで紹介する PLOT-WSX は、そのようなカルコンプ・グラフィック・ライブラリと互換性のある FORTRAN 用のグラフィック・ライブラリです。計算サーバ（spp）で利用できます。

2　グラフィック仕様

2.1　座標系について

座標系は、下図のようにウィンドウの左下を原点とし、１つの画素（1 ドット）を１単位としています。表示可能な座標範囲はウィンドウの大きさに対応しています（図 1）。

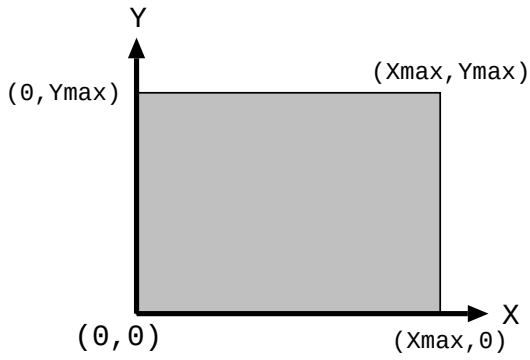


図 1. 座標系

2.2　色番号について

PLOT-WSX の色番号は、ペン番号のことで以下のように設定されています。NEWPEN ルーチンの引数で指定します（表 1）。

表 1. NEWPEN ルーチンの引数と色の対応

引数	色	引数	色	引数	色	引数	色
0	黒	1	青	2	赤	3	紫
4	緑	5	水色	6	黄	7	黒
8	灰色	9	暗青	10	暗赤	11	暗紫
12	暗緑	13	暗水色	14	暗黄	15	暗青

なお、ウィンドウの背景色は、白です。

2.3　ストロークファイルについて

PLOT-WSX は、ストロークファイルと呼ばれる中間プロットデータを作ります（任意に指定したファイル名）。プログラムを実行させてウィンドウにグラフィック表示させた後、図をプリンタに出力させる際に必要となるファイルです。
ストロークファイルは、PLOTS ルーチン（4.1.1 参照）の引数で指定します（基本ルーチンの仕様を参照）。PLOT-WSX は、作られたストロークファイルの中間プロットデータを読込んで、ポストスクリプトファイルを作ります（ファイル名：outfile.ps）。

3　使い方

3.1　プログラムの実行

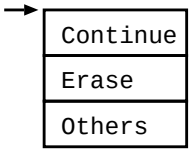
計算サーバ（spp）に login します。

```
spp[1] frt testplot.f -lplotwsx -lX11
```

ここで、testplot.f は FORTRAN プログラム、

* こにし りゅういち：京都大学 学術情報メディアセンター

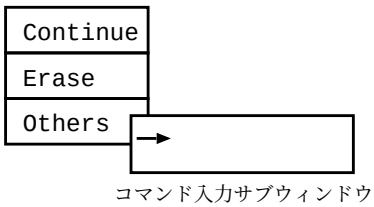
plotwsx は PLOT-WSX のライブラリ名です。
X11 は X 用ライブラリです。
コンパイルが終わり、次に実行させます。
spp[2] **./a.out**
実行が始まると、作画ウィンドウが自動的に立ち上ります。マウスを適当な所でリリースします。
PLOT-WSX では、 PLOT ルーチン（4.1.2 参照）の引数 IPENS を -13 ≦ IPENS ＜ 0 の値で呼ぶと、その時点で作画ウィンドウ上のマウスカーソルの形状がペン型からマウス型に変ります（IPENS ＞ 0 の時はペン型になったまま）。ここで、マウスの右ボタンをクリックすると下図のようなメニューが現れます（図 2）。



（マウスカーソルは右矢印になります）

図 2. メニューの選択

Continue は実行の継続を、 Erase は描画画面をクリアして再びコマンド待ち、 Others は、これ以外のコマンドの入力を意味します（後述）。
そして、マウスボタンを押したままマウスカーソルを指定する項目上に移動しボタンをリリースします。例えば、 Continue を選択すると描画され、再びコマンド待ちとなります。次に Erase で描画画面をクリアせず再び Continue を選択すると重ね書きされます。
Others を選択した時は、 Others の項目の右側にコマンド入力サブウィンドウが現れます（図 3）。



コマンド入力サブウィンドウ

図 3. コマンド入力サブウィンドウ

入力可能なコマンドは、 **q**（この時点でプログラムの実行を終了します）、 **s**（現在のプロットのパラメータの表示）、 **p**（s コマンドで表示されたパラメータの変更）があります。 s コマンド、 p コマンドの使い方の詳細はマニュアル（PLOT シリーズユーザー・マニュアル： MRI システムズ）をご覧ください。

3.2 サーチ・アドレスについて

PLOT-WSX では、 PLOT ルーチンの引数 IPENS を -13 ≦ IPENS ＜ 0 の値で呼ぶと、その時点で作画ウィンドウ上のマウスカーソルの形状がペン型からマウス型に変ることを述べましたが、同時にサーチ・アドレスも 1 つ増える仕様になっています。つまり、サーチ・アドレスとは、 1 回の Continue で描画できる単位であり、 PLOT ルーチンで原点移動（IPENS=-3）する毎に 1 つ増えていきます。
またプリンタに出力させる際の改ページの意味も持っています。ところが、汎用サーバシステム（kuma）で利用できるゼロックス X4024II では、 IPENS=-999 の時のみ改ページの意味を持っています。
センターでは、原点移動する度にバラバラに図が出力されないように PLOT-WSX に改良を加え IPENS=-999 で改ページさせています。つまり、ウィンドウ上での描画は PLOT-WSX の仕様と IPENS=-999 を生かし、出力は IPENS=-999 でのみ表示している 1 ページを出力させています。

3.3 プリンタに出力

PLOT-WSX では、描画させた図をプリンタに出力させるためには、前述したストロークファイルを作る必要があります。 PLOTS ルーチンの引数を 1 に設定します（0 はストロークファイルを作らない）。 3.1 プログラムの実行で述べたように、プログラムを実行させます。実行が始まると、次の問い合わせがあります。

```
 PLEASE INPUT PLOT FILE NAME
  ( DEFAULT = RETURN KEY )
===>
konishi
```

任意のストロークファイル名を入力します（例： konishi）。必ず入力してください（30 文字以内）。

作画ウィンドウが自動的に立ち上ります。描画操作は、既に述べたとおりです。メニュー項目を選択して描画させます。
次に、ストロークファイルからポストスクリプトファイルを作ります。
spp[3] **plot2ps**
実行が始まると、作画ウィンドウが自動的に立ち上り、次の問い合わせがあります（図 4）。

```
出力先の指定
1.ファイル（outfile.ps）
2.デバイス
====>
```

図 4. 問合わせ：出力先の指定

1 を入力します。
再び問い合わせがあります（図 5）。

```
用紙のサイズ
1.A4 : 2.B4 : 3.A3
====>
```

図 5. 問合わせ：用紙のサイズ

1 を入力します。
再び問い合わせがあります（図 6）。

```
デフォルトでは、横768、縦1024が用紙に展開されます。
これに対するスケールファクターを入力してください。
====>
```

図 6. 問合わせ：スケールファクター

1 を入力します。最後の問い合わせがあります。

```
図形データのファイル名を入力して下さい.
（30 文字まで　：　終了ーリタンキー　）
===>
```

konishi

プログラムの実行の際に作られたストロークファイル名（例： konishi）を入力します。
作画ウィンドウが自動的に立ち上るので、メニュー項目を選択して描画させます。
サーチ・アドレスが変更する度にメッセージが出力されます（SEARCH ADDRES がかわります。）。
全ての描画が終了すると、再度ストロークファイル名を入力する問い合わせがあります。終了の場合はキーボードの Enter キーを押して終ります。
続いて別のストロークファイル名を入力したければ入力し、描画操作を繰返していきます。 Enter キーを押して終ると **outfile.ps** という名前でポストスクリプトファイルが出来上がります。
lpr コマンドでプリンタに出力させます。
spp[3] lpr -Plw outfile.ps

4　基本ルーチンと応用ルーチン仕様

4.1　基本ルーチン

PLOT-WSX で提供されている基本ルーチンについて説明します。

4.1.1　PLOTS

[機能]
PLOT-WSX ライブラリの初期化をします。
[呼出し方]
CALL PLOTS (I)

`I`　：ストロークファイルを作成するか否かのフラグ。
`= 0` ：ストロークファイルを作成しません。
`= 1` ：ストロークファイルを作成します。

[注]
このサブルーチンは他のサブルーチンより前に、最初に一度呼出す必要があります。 PLOTS を引数なしで使用した場合の動作は、保証されません。

4.1.2　PLOT

[機能]
現在点から指定した点（X,Y）まで直線を作図させたり、原点を変更させたりします。
[呼出し方]
CALL PLOT (X, Y, IPENS)

`X,Y`　：移動しようとする点の座標（単位は `FACTOR` 参照）。
`IPENS`：描画の状態。

= 2　描画状態で移動します。つまり現在位置から点（X,Y）まで直線をプロットします。

= 3　何もプロットせずに位置を移動します。

=12 描画状態で OFFSET へ（OFFSET 参照）。

=13 プロットせずに OFFSET へ。

=22 描画状態で移動後、その点を新しい原点とします。

=22 プロットせずに移動後、その点を新しい原点とします。

=92 絶対座標の（X,Y）まで描画状態で移動し、その点を原点にします。

=93 絶対座標の（X,Y）まで何もプロットせずに移動し、その点を原点にします。

（IPENS=92、93 以外は、点（X,Y）は相対座標で指定します）

-13 ≦ IPENS ＜ 0

点（X,Y）に移動後、その点を新しい原点（0,0）とし、サーチ・アドレスを 1 増やします。描画状態は上記正の数値の場合と同じです。

=-999

出力の際の改ページを意味します。作画ウィンドウではペンの動きは -3 と同じです。

=999

ペンの動きは -3 と同じです。作業を終了する際、必ず 999 としなければなりません。

[注]

CALL PLOT(X,Y,999) はすべてのプロット操作の終わりを示すもので、最後の 1 回だけ呼出さなければなりません。

引数 X、Y は暗黙の型宣言に従い、実数型 (REAL*4) です。1.0 が画面ドットの 1 個に対応しています。

4.1.3 NEWPEN

[機能]

ペン番号を指定します（線の太さではない）。

[呼出し方]

CALL NEWPEN (IPEN)

IPEN：0 ～ 15 の整数で、ペン番号（色の番号：表 1参照）を指定します。デフォルトは 1 です。

4.1.4 FACTOR

[機能]

図形全体の拡大、縮小を指定します。

[呼出し方]

CALL FACTOR (FACT)

FACT：尺度。1.0 が座標 1 カウント =1 ドットに対応します。デフォルトは 1.0 です。

汎用サーバシステム（kuma）で利用できるゼロックス X4024II では、プロット単位が CM（センチメートル）なので、プログラムを変更せずに実行させる際には FACT 値を 26.7 にする必要があります（1CM　26.7 ドット、5 使用例と実行例参照）。

[注]

FACTOR を一度呼出すと、次に FACTOR を CALL するまでこの尺度は他のサブルーチンに対して適用されます。もし、FACT=2.0 とすると、以降 FACT 値が変らない限り、寸法が原図の 2 倍に拡大されプロットされます。

4.1.5 SYMBOL

[機能]

英数字、特殊文字をプロットする時、あるいは、シンボル・テーブル（図 14.）のコードで指定されている記号をプロットする時に用います。

[呼出し方]

① **CALL SYMBOL** (X, Y, HEIGHT, IBCD, ANGLE, NCHAR)

② **CALL SYMBOL** (X, Y, HEIGHT, ISYM, ANGLE, ICODE)

①の場合

X,Y　：最初にプロットする文字の左下端の座標。

HEIGHT：プロットする文字の高さ。

IBCD　：文字の入っている変数の名前。

ANGLE ：プロットする文字と X 軸のなす角度、反時計方向が正（単位：度）。

NCHAR ：プロットする文字数。

＞ 0 ：データは IBCD の左端からつめて入れなければなりません。

= 0 ： IBCD の右端の 1 文字だけプロットします。

[注]

字幅は、文字間の空白も含めて高さと同じです。座標 X または Y を 999.0 にすると前回 SYMBOL のプロットし終わった場所から連続して文字をプロットできます。一回の呼出しで指定可能な文字数は 132 以下に制限されています。

②の場合

X,Y　：最初にプロットする文字の左下端の座標。

HEIGHT：プロットする記号の高さ。

ISYM　：シンボル・テーブルのコード番号（整数）（0 ≦ ISYM ≦ 127）。

ANGLE ：プロットする文字と X 軸のなす角度、反時計方向が正（単位：度）。

ICODE ：移動状態

= -1　点（X,Y）まで移動し、指定コードの記号をプロットします。

≦ -2　点（X,Y）まで移動し、つまり現在位置より点（X,Y）まで直線を描き、そこで指定コードの記号をプロットします。

[注]

0 ≦ ISYM ≦ 13 の時（センター・シンボル）には、X、Y は記号の左下端ではなく、記号の中心を示しています。

4.1.6 NUMBER

[機能]

REAL 型の数字を 10 進数でプロットします。

[呼出し方]

CALL NUMBER (X, Y, HEIGHT,RNUM, ANGLE, NDEC)

X,Y　：最初にプロットする数字（又は小数点）の左下端の座標。

HEIGHT：プロットする数字の高さ。

RNUM　：数字の入っている変数の名前か REAL 型の定数。

ANGLE ： X 軸のなす角度、反時計方向が正（単位：度）。

NDEC　：小数点以下の桁数、いずれも端数は四捨五入。

≦ -1：整数部 NDEC 桁より大きい部分の数字をプロットします。

= -1 ：整数部のみプロットします。

= 0 ：整数部と小数点をプロットします。

＞ 0 ：整数部と小数点以下 NDEC 桁までプロットします。

[注]

X、Y、HEIGHT、ANGLE に関しては SYMBOL と同じです。

4.1.7 SCALE

[機能]

配列 ANAME に格納されているデータのスケーリング・パラメータ、つまり最小値 MIN（最大値 MAX）と増分Δを求めます。

[呼出し方]

CALL SCALE (ANAME,WIDTH,N,INTV)

ANAME ：スケールするデータが格納されている配列の名前。

WIDTH ：データをスケールする領域の長さ。

N　：配列 ANAME の中に入っているデータ数。

INTV　：配列 ANAME の中に入っているデータ間隔。

= ± 1 ：スケールするデータは、配列 ANAME の中に連続して格納されています。
通常 1 次元の DIMENSION をもつ配列からデータを取出す時に用います。 -1 の時は最大値を出発値とします。

＞ 1 ：スケールするデータは配列 ANAME 中に INTV おきに格納されています。
通常多次元の DIMENSION をもつ配列からデータを取出す時に用います。 出発値として最適の最小値を用います。 また、単位長さ当りの増分として 正のΔをとります。

＜ -1 ：出発値として最適の最大値を用います。また、 単位長さ当りの増分として負のΔをとります。

[注]

スケールした結果は ANAME（N*INTV+1）に最小値 MIN が、 ANAME（N*INTV+INTV+1）

には増分Δ =(MAX-MIN)/WIDTH が格納され、LINE、 AXIS 等でこの 2 つの値を使うことができます。

4.1.8 OFFSET

[機能]

PLOT によって使われる OFFSET の要素を変更することにより、座標変換を行います。

[呼出し方]

CALL OFFSET (XOFF,XFCT,YOFF,YFCT)

XOFF ：基準点の X 座標。

XFCT ： X 軸に対する縮尺率。

YOFF ：基準点の Y 座標。

YFCT ： Y 軸に対する縮尺率。

[注]

PLOT の引数 IPENS が± 12、± 13 の場合、与えられた座標（X,Y）はマトリックス形式で表示すると

$$\begin{Bmatrix} X1 \\ Y1 \end{Bmatrix} = \begin{Bmatrix} \frac{1}{XFCT} & 0 \\ 0 & \frac{1}{YFCT} \end{Bmatrix} \begin{Bmatrix} X - XOFF \\ Y - YOFF \end{Bmatrix}$$

つまり

$$X1 = \frac{X - XOFF}{XFCT} \quad , \quad Y1 = \frac{Y - YOFF}{YFCT}$$

と変換された点（X1,Y1）に移動します。

OFFSET を一度も呼出さない場合には、デフォルトとして XOFF、 YOFF は 0.0 に、 XFCT、 YFCT は 1.0 にそれぞれ設定されています。

4.1.9 AXIS

[機能]

指定した点より目盛りとタイトルの付いた軸を描きます。

[呼出し方]

CALL AXIS (X, Y, TITLE, NCAHR,
LENGTH, ANGLE, FIRSTV, DELTA)

X,Y ：座標軸をかき始める出発点の座標。

TITLE ：軸の名称の入っている変数名。

NCHAR ： TITLE の文字数。

＞ 0 ：軸に対して反時計廻りに目盛りと軸の名称を描きます。

＜ 0 ：軸に対して時計廻りに目盛りと軸の名称を描きます。

LENGTH：軸の長さ（REAL 型）。

ANGLE ：軸と X 軸のなす角度、反時計方向が正（単位：度）。

FIRSTV：軸の最初の目盛りの値、通常 SCALE で求めた最小値 MIN （最大値 MAX）を使用しますが、別に与えることもできます。

DELTA ：軸に目盛りを付ける時、単位長さ当りの目盛りの増分（目盛り / 長さ）。
通常 SCALE で求めた増分Δの値を用いますが、別に与えることもできます。

[注]

軸の名称は軸の中央部に、軸に対して平行にプロットされます。また、目盛りの数値は小数点以下第 1 位まで表示されます。

4.1.10 LINE

[機能]

2 つの配列 X および Y に格納されている座標点を直線で結びます。

[呼出し方]

CALL LINE (X,Y,N,INTV,LTYPE,ISYM)

X ：データの X 座標とスケーリング・パラメータを格納している配列名。

Y ：データの Y 座標とスケーリング・パラメータを格納している配列名。

N ：プロットするデータの数（X と Y は同じデータ数でなければなりません）。

INTV ：プロットするデータの配列に入っている間隔。

= 1 ：プロットするデータは配列の中に連続して格納されています。
通常 1 次元の DIMENSION をもつ配列からデータを取出す時に用います。

＞ 1 ：プロットするデータは配列中に INTV おきに格納されています。
通常多次元の DIMENSION をもつ配列からデータを取出す時に用います。

出発値として最適の最小値をとります。また、単位長さ当りの増分として正のΔをとります。

＜ -1 ：出発値として最適の最大値をとります。また、単位長さ当りの増分として負のΔをとります。

LTYPE ：線のむすび方

= 0 ：実線でデータの各点を結びます。

= 1 ：実線でデータの各点を結び、各点に指定した記号をプロットします。

≧ 2 ：実線でデータの各点を結び、LTYPE おき、つまり（X(1), Y(1)）、（X(LTYPE+1), Y(LTYPE+1)）・・・に記号をプロットします。

＜ 0 ：データの各点は結ばず、記号のみプロットします。絶対値の意味は正の場合と同じです。

ISYM ：プロットする記号の種類（シンボル・テーブル参照）。

[注]

このサブルーチンは、原点を次の位置にあるとみなすので、注意が必要です。

X 座標 =MINX（Δ X が正の時）、または MAXX（Δ X が負の時）

Y 座標 =MINY（Δ Y が正の時）、または MAXY（Δ Y が負の時）

ここで MINX、Δ X、MINY、Δ Y は配列 X 及び Y を SCALE で計算して求めた各々のスケーリング・パラメータです。

また、データを LINE でプロットする場合、i 番目の点（X(i), Y(i)）の原点からの距離は次のとおりです。

$$X = \frac{X(i) - MINX}{\Delta X} \quad or \quad \left[= \frac{MAXX - X(i)}{\Delta X} \right]$$

$$Y = \frac{Y(i) - MINY}{\Delta Y} \quad or \quad \left[= \frac{MAXY - Y(i)}{\Delta Y} \right]$$

SCALE を用いて、あらかじめ MINX、Δ X、MINY、Δ Y に相当するスケーリング・パラメータを求めていない時は、ユーザーが与えなくてはなりません。その場合、データの最小値（または最大値）と単位長さ当りの増分（(最大値 - 最小値)/ 作図したい幅）を求めて設定します。

また、データ X、Y の DIMENSION は、（N*INTV+INTV+1）個とっておかなければなりません。

4.1.11 WHERE

[機能]

現在の位置と尺度を求めます。

[呼出し方]

CALL WHERE (X, Y, FACT)

X ：現在の位置の X 座標を格納します（出力）。

Y ：現在の位置の Y 座標を格納します（出力）。

FACT ：前の FACTOR で与えた尺度を格納します（出力）。

[注]

変数 X、Y、FACT は全て REAL 型の変数です。X、Y は最新に原点移動を行った点からの FACT 倍した値です。

4.1.12 CTOI

[機能]

CHARACTER 型変数内の文字列を整数型変数に代入します。

[呼出し方]

CALL CTOI (IC, C, ILEN)

IC ：整数型配列

C ：文字型変数、もしくはリテラル定数

ILEN ：文字数

[注]

IC は整数型配列として宣言されていて、その大きさは、

ILEN が 4 の倍数の場合は、ILEN/4

ILEN が 4 の倍数でない場合は、 ILEN/4+1
より大きくなければなりません。

4.1.13 XWPARM

[機能]
作画ウィンドウの領域を指定します。
[呼出し方]
CALL XWPARM (I1, I2, I3, I4)
X ディスプレイ上の 2 点をドット座標で指定します（図 7）。

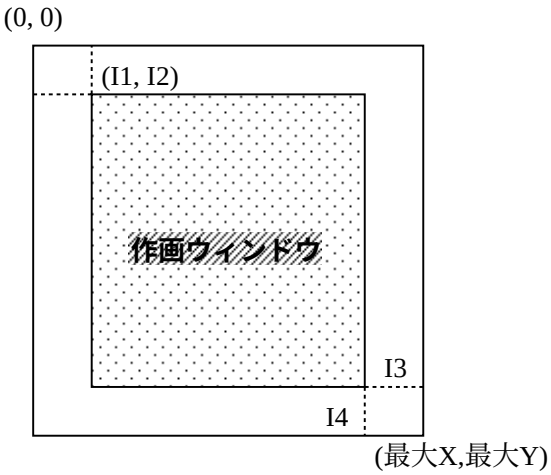


図 7. ウィンドウの領域指定

I1　　：左上点の X 座標。
I2　　：左上点の Y 座標。
I3　　：右下点の最大 X までのドット数。
I4　　：右下点の最大 Y までのドット数。

[注]
（最大 X, 最大 Y）は EWS 等に依存します。また、ドット座標系は PLOT-WSX の座標系（左下隅が（0,0））と異なります。 PLOTS の前に呼んでください。

4.1.14 SLCHARC

[機能]
サブルーチン①SYMBOL で指定した小文字の英字を、大文字に変換して表示するか小文字のまま表示するかの指定を行います。
[呼出し方]
CALL SLCHARC (IFLG)

IFLG　：指定フラグ。
= 0　：SYMBOL で指定した小文字の英字を大文字に変換して表示（デフォルト）。
= 1　：SYMBOL で指定した小文字の英字を小文字のまま表示。

4.1.15 DFNFNT

[機能]
KANJI ルーチンで使用するフォントファイルをオープンします。 JIS 第一、 JIS 第二水準漢字を含む文字を描くことができます。
[呼出し方]
CALL DFNFNT (IUNIT, CDIS)

IUNIT　：フォントファイルのファイル番号。
CDIS　：フォントファイルのあるディレクトリ（文字型、半角のブランクを'' で指定。5 使用例と実行例参照）。

[注]
漢字ルーチンを使用する前に一度だけ呼出してください。また、 FORTRAN プログラムは EUC で作成してください。

4.1.16 KANJI

[機能]
JIS 第一、 JIS 第二水準漢字を含む文字列を表示します。
[呼出し方]
CALL KANJI (X, Y, HEIGHT, ANGLE, CHRS, NN)

X,Y　：文字列の左下端の座標。
HEIGHT　：プロットする文字の高さ。
ANGLE　：プロットする文字列と X 軸のなす角度。反時計方向が正（単位：度）。
CHRS　：プロットする文字列（文字型）。
NN　：プロットする文字数（ASCII 文字も漢字も各 1 文字として数えます）。

[注]

字の縦横比は 10： 10、字の幅と字間の比は 10：3 になっています。フォントは基本的に 8 × 8 で作られているので、高さを 8 の倍数にすると比較的きれいにプロットできます。
座標 X または Y を 999.0 にすると前回の KANJI、SYMBOL の終わった場所から連続して文字をプロットできます。また、 KANJI ルーチンを呼出す前に必ず DFNFNT ルーチンを呼出す必要があります。

4.1.17 SENDTITLE

[機能]
描画ウィンドウのタイトルの設定をします。
[呼出し方]
CALL SENDTITLE (C, NCHAR)

C　　：タイトルに使う文字列（文字型変数）。
NCHAR　：文字列のバイト数。

[注]
描画ウィンドウのタイトルバーに表示される文字列を変更します。 PLOT-WSX では、デフォルトは「PLOT-WSX」です。

4.1.18 PLFLUS

[機能]
作画ウィンドウを強制的にリフレッシュ（実際の描画実行）します。
[呼出し方]
CALL PLFLUS

4.1.19 EPAINT

[機能]
指定した色で描かれた閉領域を指定した色で塗潰します。
[呼出し方]
CALL EPAINT (X, Y, IC, ICB)

X,Y　　：始点（塗潰し）の X、 Y 座標。
IC　　：塗潰しをする色番号（ペン番号）。
ICB　　：境界線の色番号（ペン番号）。

以上、基本ルーチンについて紹介しましたが、詳細はマニュアルを参照してください。

4.2 応用ルーチン

PLOT-WSX では、基本ルーチン群を使った便利な応用ルーチン群があります。代表的なものを載せます（表 2）。

表 2. 応用ルーチン群

ARROW	配列に格納されているデータを結び、先端に矢印を描く.
AXISB	AXIS と同様で、扱うデータによって単位が自動的に決定され、標号が軸のタイトルとして描かれる.
AXISC	AXIS と同様だが、目盛間に月名を描く.
BAR	棒を描き、ハッチングもできる.
CIRCL	円、円弧、ら線を描く.
CNTRL	配列 X、 Y に格納されているデータを 1 点鎖線で結ぶ.
CURVX	X の多項式 $Y = C1X^{E1} + C2X^{E2} + C3X^{E3} + C4X^{E4}$ の曲線を描く.
CURVY	Y の多項式 $Y = C1Y^{E1} + C2Y^{E2} + C3Y^{E3} + C4Y^{E4}$ の曲線を描く.
DASHL	配列に格納されているデータを破線で結ぶ.
DASHP	ペンの現在位置から指定された座標まで破線を描く.
DIMEN	製図用の寸法線、寸法補助線、寸法を描く.
ELIPS	楕円を描く.
FIT	与えられた 3 点を結び、点 1、点 3 を結ぶ線分の垂直 2 等分線上に頂点がある片双曲線を描く.
FLINE	配列に格納されているデータを直線、または放物線カーブで結ぶ.
GRID	格子を描く.
LABEL	指定座標点間に注釈を描く.
LBAXS	常用対数軸を描き、扱うデータにより単位が自動的に決定され、標号が軸のタイトルに描かれる.
LGAXS	常用対数軸を描く.

各応用ルーチンの引数等の詳細は、マニュアルを参照ください。

5　使用例と実行例

次に、PLOT-WSX の基本ルーチンを使ってサンプル・プログラムを作り、実行させます。

[例 1.] ファイル名： demo.f

```
C     PLOT-WSX ルーチンを使用する際、最初に CALL
      CALL PLOTS(0)
      CHARACTER*80 C80
      CHARACTER*5 CKEY
      EQUIVALENCE(C80(1:5),CKEY)
      IPEN = 7
      ICOLOR = 5
      OPEN(10,FILE='DEMO.INP',STATUS='OLD')
  100 READ(10,'(A)',END=900) C80
      IF(CKEY.EQ.'INIT ') THEN
        CALL XWPARM(240,140,320,140)
        CALL PLOTS(0)
        CALL NEWPEN(IPEN)
        CALL PLOT(0.,0.,3)
        CALL PLOT(860.,  0., 2)
        CALL PLOT(860., 750., 2)
        CALL PLOT(  0., 750., 2)
        CALL PLOT(  0.,   0., 2)
        CALL EPAINT( 200., 200., ICOLOR, IPEN)
      ELSE IF(CKEY.EQ.'TITL ') THEN
        CALL SENDTITLE(C80(6:24)//CHAR(0),20)
      ELSE IF(CKEY.EQ.'LINE ') THEN
        READ(C80(6:),'(5I5)') IX1,IY1,IX2,IY2,ICOL
        IF(ICOL.EQ.1) ICOL = 7
        IF(ICOL.NE.IPEN) THEN
          IPEN=ICOL
          CALL NEWPEN(IPEN)
        ENDIF
        X = IX1
        Y = IY1 + 80.- 42.
        CALL PLOT(X,Y,3)
        X = IX2
        Y = IY2 + 80.- 42.
        CALL PLOT(X,Y,2)
      ELSE IF(CKEY.EQ.'END  ') THEN
        CALL PLFLUS
        GO TO 900
      ELSE IF(CKEY.EQ.'FACT ') THEN
        READ(C80(6:),'(2I5)') IX1,IY1
        X = FLOAT(IX1)/FLOAT(IY1)
        CALL FACTOR(X)
      ELSE IF(CKEY.EQ.'FLUSH') THEN
        CALL PLFLUS
      ENDIF
      GO TO 100
  900 CALL PLOT(0.,0.,-3)
      CALL PLOT(0.,0.,999)
      STOP
      END
```

プログラム中の読込みデータ（DEMO.INP）は

```
spp[11]> more DEMO.INP
TITL  PLOT-WSX DEMO
INIT
FACT    15   10
LINE    91    9   91    8    1
LINE    91    8   92    8    1
LINE    92    8   93    9    1
LINE    93    9   93   10    1
LINE    93   10   92   10    1
LINE    92   10   91   10    1
LINE    91   10   91   11    1
LINE    91   11   91   11    1
LINE    91   11   91   12    1
LINE    91   12   92   12    1
LINE    92   12   93   11    1
LINE    94   12   94    9    1
LINE    94    9   94    8    1
LINE    94    8   95    8    1
LINE    95    8   96    9    1
LINE    96    9   96   12    1
LINE    97    8   97   12    1
LINE    97   12   98   12    1
LINE    98   12   98   11    1
```

図 8. 読込みデータ

のような文字と座標点からなるデータです。

3.1 プログラムの実行に従って実行させます。

spp[1] **frt demo.f -lplotwsx -lX11**

spp[2] **./a.out**

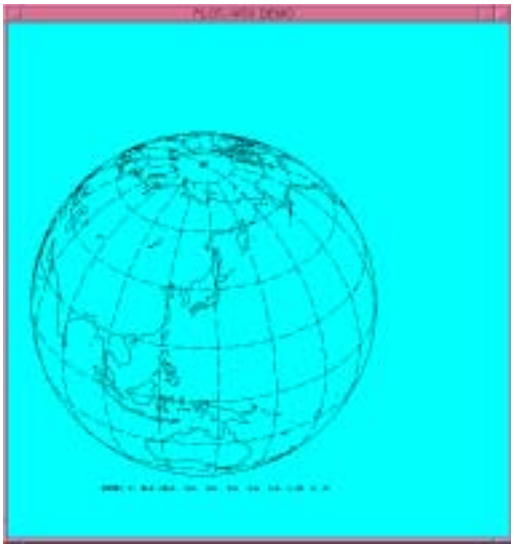


図 9. 実行結果

./a.out を実行させると、作画ウィンドウが自動的

に立ち上がり描画が始まります（図 9）。
[例 1.] ファイル名： konishi.f
次は、KPLOT を使って描いたプログラムと PLOT-WSX ライブラリを使って描いたプログラムとを比較してみます。
【KPLOT の場合】

```
      real*4    XXX(5)
      real*4    YYY(5)
      real*4    ZYPP,ZXPP,ZFACT
      data      XXX/4.5, 21.5, 21.5, 4.5, 4.5/
      data      YYY/1.5, 1.5, 18.5, 18.5, 1.5/
      character*6  TGAM, TPIN
      character*4  TISTEP
      character*10 TTIME
      character*15 JTIME/' (Tue/Nov/1997)'/
      character*4  TAKEDA(8), KANJI(7)
      data TAKEDA/'215a','3a6e', '2127', '4267',
     +            '373f', '4240', '4f3a', '215b'/
      data KANJI/'3441', '3b7a', '256b', '213d',
     +           '2541', '2573', '4e63'/
      IPLOT=0
      call plots
  100 データを読込む
      call factor(0.7)
      call plot(2.,2.,-3)
      call plot(XXX(1), YYY(1), 3)
      call plot(XXX(2), YYY(2), 2)
      call plot(XXX(3), YYY(3), 2)
      call plot(XXX(4), YYY(4), 2)
      call plot(XXX(5), YYY(5), 2)
      call plot(13.,10.,-3)
      call courie(-8.,  9.,0.3,'GAM=',0.,4)
      call courie(999.0,999.0,0.3,TGAM,0., 6)
      call courie(999.0,999.0,0.3,',PIN=',0.,5)
      call courie(999.0,999.0,0.3,TPIN,0., 6)
      call courie(999.0,999.0,0.3,',ISTEP=', 0.,7)
      call courie(999.0,999.0,0.3,TISTEP,0., 4)
      call courie(999.0,999.0,0.3,',TIME=',0.,6)
      call courie(999.0,999.0,0.3,TTIME,0.,10)
      call courie(999.0,999.0,0.4,JTIME,0.,15)
      call pkanji(-4.,-9.5, 0.4, KANJI,0., 7)
      call pkanji(999.0, 999.0, 0.4, TAKEDA, 0., 8)
      :
      データをプロットする
      IPLOT= IPLOT+1
      :
      if (IPLOT.gt.0) then
        call plot(0.,0.,-999)
        goto 100
      else
        call plot(0.0,0.0,999)
        stop
      endif
```

spp[1] **setenv fu17 konishi.data**
spp[2] **frt konishi.f -lkplotpr**
spp[3] **./a.out**
spp[4] **ghostview fort.99**
ghostview で実行結果（fort.99）を見ると

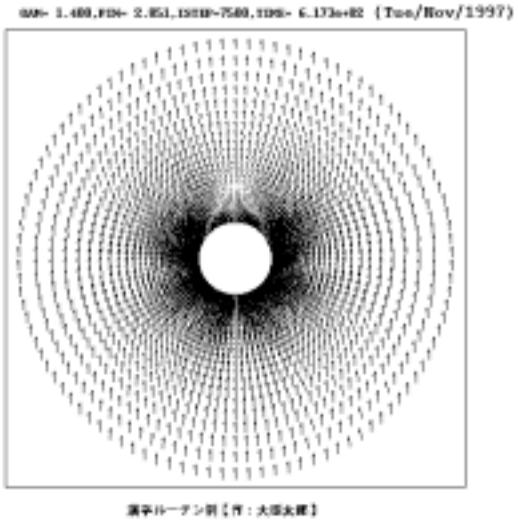


図 10. KPLOT での実行結果

となります。
【PLOT-WSX の場合】

```
      REAL*4    XXX(5)
      REAL*4    YYY(5)
      REAL*4    ZYPP,ZXPP,ZFACT
      DATA      XXX/4.5, 21.5, 21.5, 4.5, 4.5/
      DATA      YYY/1.5, 1.5, 18.5, 18.5, 1.5/
      CHARACTER*6  TGAM, TPIN
      CHARACTER*4  TISTEP
      CHARACTER*10 TTIME
      CHARACTER*16 JTIME/' (Mon/July/2003)'/
      IPLOT=0
      FACT= 26.7
      CALL PLOT(1)
  100 データを読込む
      CALL FACTOR(FACT*0.7)
      CALL PLOT(2.,2.,-3)
      CALL PLOT(XXX(1), YYY(1), 3)
      CALL PLOT(XXX(2), YYY(2), 2)
      CALL PLOT(XXX(3), YYY(3), 2)
      CALL PLOT(XXX(4), YYY(4), 2)
      CALL PLOT(XXX(5), YYY(5), 2)
      CALL PLOT(13.,10.,-3)
      CALL SLCHARC(1)
      CALL SYMBOL(-8.,9.,0.3,'GAM=',0.,4)
```

```
      CALL SYMBOL(999.0,999.0,0.3,TGAM,0., 6)
      CALL SYMBOL(999.0, 999.0, 0.3, ',PIN=',0., 5)
      CALL SYMBOL(999.0, 999.0, 0.3,TPIN,0.,6)
      CALL SYMBOL(999.0, 999.0, 0.3,',ISTEP=',0.,7)
      CALL SYMBOL(999.0, 999.0, 0.3,TISTEP,0., 4)
      CALL SYMBOL(999.0, 999.0, 0.3,',TIME=',0.,6)
      CALL SYMBOL(999.0, 999.0, 0.3,TTIME,0.,10)
      CALL SYMBOL(999.0, 999.0, 0.3,JTIME,0.,16)
      CALL DFNFNT(99,' ')
      CALL KANJI(-4.,-9.5, 0.4, 0.,
     +    '漢字ルーチン例【作：大型太郎】',15)
      :
      データをプロットする
      IPLOT= IPLOT+1
      :
      if (IPLOT.GT.0) then
        CALL PLOT(0.,0.,-999)
        GOTO 100
      ELSE
        CALL PLOT(0.0,0.0,999)
        STOP
      ENDIF
```

spp[1] **setenv fu17 konishi.data**

spp[2] **frt konishi.f -lplotwsx -lX11**

spp[3] **./a.out**

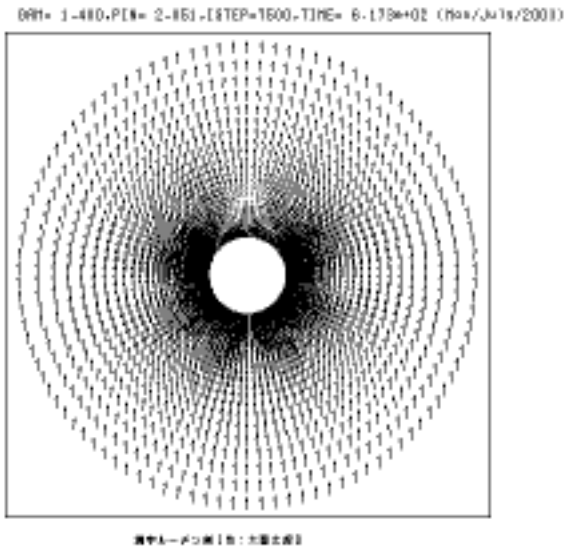


図 11. PLOT-WSX での実行結果

./a.out を実行させると、作画ウィンドウが自動的に立ち上がり描画が始まります（図 11）。

更に、作画ウィンドウ上で、メニューの選択により **Erase** ⇒ **Continue** を続けると、次の描画が始まります（図 12）。

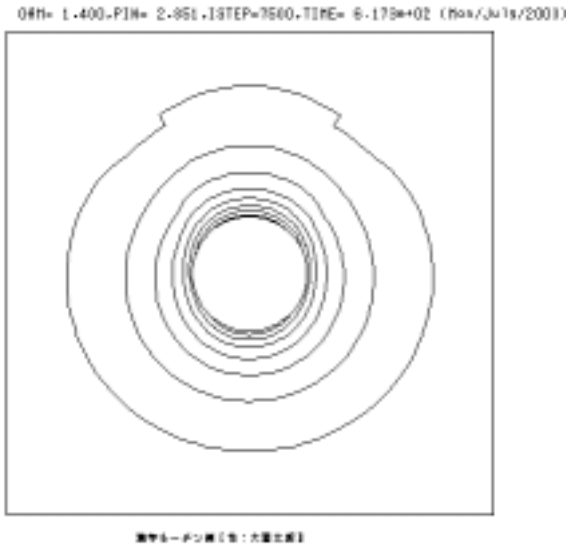


図 12. 描画の継続

Erase せずに Continue して行くと、図は重なり合って描画されます（図 13）。

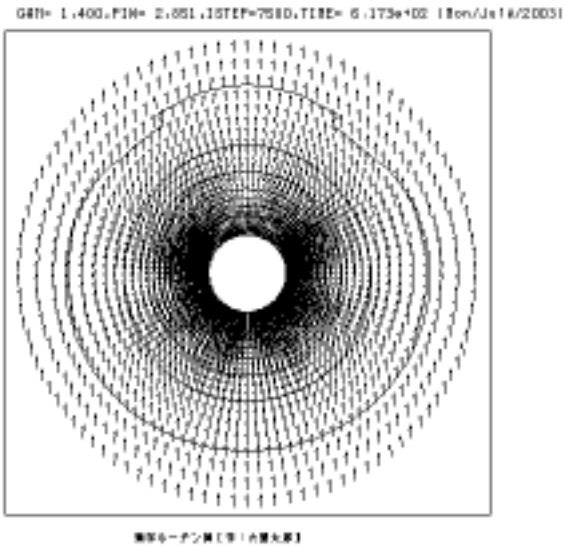


図 13. 描画の重なり

最後に、ポストスクリプト・ファイルを作り、プリンタに出力して見ましょう。

spp[4] **plot2ps**

実行が始まると、作画ウィンドウが自動的に立ち上ります。作画ウィンドウ上で、メニューの選択により図が正常に描画されているか確認します。

確認が終わったら、 lpr コマンドでプリンタに出力

させます。

spp[5] **lpr -Plw outfile.ps**

6　利用上の注意点

PLOT-WSX を利用する際には、以下の点に注意してください。

1）特有のエントリー名を内部で使用しているため、重複する場合があります。

2）以下の COMMON ブロック名を使用しているため、重複しないように注意してください。

```
      COMMON /G_GPSLMM/M(15)
     *      ,XORG,YORG,XNOW,YNOW,FACT,XOFF,
     *       YOFF,XFCT,YFCT,XSYM,YSYM,GPSLC,
     *       GPSLS,SMALL,IPEN,XMIN,XMAX,YMIN,
     *       YMAX,XSTP,YSTP,GPSLZ0,GPSLZ1,
     *       GPSLZ2,GPSLZ3,GPSLZ4,GPSLZ5,
     *       GPSLZ6,GPSLZ7,LUNCON,IPRINT,LUNRD
      COMMON /G_GPSLMN/ CLS(4)
      COMMON /G_FLGMAX/ MAXXXX,MAXYYY
      COMMON /G_GPMETA/IFM,IFFL,IMU11,IMU12,
     *       INQFG,IFEHC,ICHFLG
      COMMON /G_GPFNAM/METANM
      COMMON /kplot/idx,idy
```

3）EPAINT ルーチンで色指定で図を塗潰しても、出力時には線図形となります（濃淡なし）。

4）CTOI ルーチンは、本来はパソコン版用のサブルーチン（パソコンで動く FORTRAN では、文字型と整数型での文字の格納方法が異なっており、 SYMBOL ルーチンに文字型変数を渡すと 4 文字単位で逆転表示されてしまいます。例えば、'ABCD' は'DCBA' と表示されます。CTOI ルーチンはそれを防ぐためのものです。）ですが、使っても問題ありません。マニュアルに記されているサンプル・プログラムも正常に実行されます。

5）次に、②SYMBOL ルーチンで使うシンボル・テーブル表をプログラムと一緒に載せます。

```
      CHARACTER*4 TITL(1)
      DIMENSION ITITL(1)
      DATA TITL/'0   '/
C
      CALL XWPARM(400, 400, 250, 250)
      CALL SENDTITLE('MRI SYMBOL SAMPLE',17)
      CALL PLOTS(0)
C
      DO 10 J=0,7
      DO 10 I=0,15
       X=I*35.+30.
       Y=45.*J+40.
       II=MOD(I,7)+1
       CALL NEWPEN(1)
       CALL SYMBOL(X,Y,10.,J*16+I,0.,-1)
       CALL NEWPEN(7)
       RNUM=J*16+I
       IF(I.EQ.0.AND.J.EQ.0) THEN
          CALL CTOI(ITITL,TITL,4)
          CALL SYMBOL(X,Y-30.,8.,ITITL,0.,1)
         GO TO 10
       ENDIF
       IF(J.GE.1) CALL NUMBER(X,Y-12.,8.,RNUM,0.,-1)
       IF(J.EQ.0) CALL NUMBER(X,Y-30.,8.,RNUM,0.,-1)
   10 CONTINUE
      CALL PLOT(0.,0.,-3)
      CALL PLOT(0.,0.,999)
      STOP
      END
```

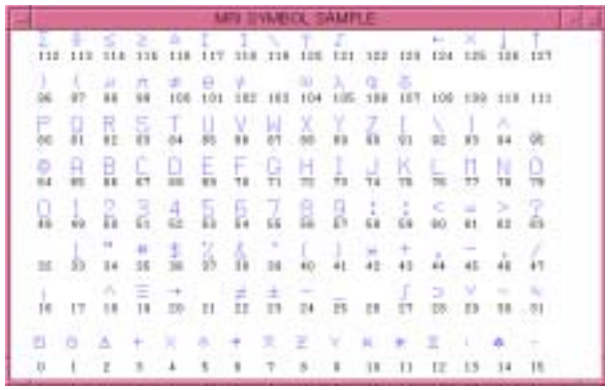


図 14. シンボル・テーブル

7　おわりに

PLOT-WSX について簡単に紹介してきましたが、如何でしたでしょうか？ X4024II、 KPLOT 同様使って頂ければ幸いです。

最後に、コンピューティング掛のスタッフの皆さん、また快くプログラム、データを提供して頂いたプログラム指導員の武田先生に感謝致します。有難うございました。