

CHAPTER 4: 描画関数 (線画)

4.1 ARC1

(1) 機能

中心、半径および方位角で指定される円弧を描く(GPSL準拠)

(2) 呼び出し形式

CALL ARC1(X,Y,R,ANGS,ANGE,LTY)

(X,Y) : [REAL*4] 中心のx,y座標

R : [REAL*4] 半径

ANGS : [REAL*4] 円弧の開始角(deg)

ANGE : [REAL*4] 円弧の終了角(deg)

LTY : [INTEGER*4] 線種を指定 =0;線種の変更は行なわない

(1:実線 2:破線 3:一点鎖線 4:二点鎖線 5:点線)

4.2 ARC2

(1) 機能

2点および半径で指定される円弧を描く(GPSL準拠)

(2) 呼び出し形式

CALL ARC2(XS,YS,XE,YE,R,LTY)

(XS,YS) : [REAL*4] 円弧の始点のx,y座標

(XE,YE) : [REAL*4] 円弧の終点のx,y座標

ABS(R) : [REAL*4] 半径

R > 0 : 時計回りに円弧を描く

R < 0 : 反時計回りに円弧を描く

LTY : [INTEGER*4] 線種を指定 =0;線種の変更は行なわない

(1:実線 2:破線 3:一点鎖線 4:二点鎖線 5:点線)

4.3 AROHD

(1) 機能

指定された線分に矢印を描く(GPSL準拠)

(2) 呼び出し形式

CALL AROHD(X0,Y0,X1,Y1,AL,AW,IC)

(X0,Y0) : [REAL*4] 矢印を描く線分の始点

(X1,Y1) : [REAL*4] 矢印を描く線分の終点 (矢印部分が付く位置)
 AL,AW : [REAL*4] 矢印の長さ・幅
 IC : [INTEGER*4]
 IC/10=0: 矢印部分のみ描画する
 1: 線分と矢印を描画する
 2: 線分の両側に矢印を描画する
 MOD(IC,10)=1: 白抜き三角形 2: 上書き三角形 3: 片側三角形
 4: 両側矢 5: 片側矢 6: 塗りつぶし三角形
 7: 片側塗りつぶし三角形

表示例が Figure 2 にある

(3) その他

- ・ 矢印の線種はPENX, NEWPEN等で指定しておく必要がある
- ・ 塗りつぶしの色はPENX, NEWPEN等で指定した色になる

4.4 ARROW

(1) 機能

矢印を描画する

(2) 呼び出し形式

CALL ARROW(X0,Y0,X1,Y1,DC)

(X0,Y0) : [REAL*4] 矢の先端 (矢印部分が付く位置)
 (X1,Y1) : [REAL*4] 矢の始点
 DC : [REAL*4] 矢と矢の軸の長さの比

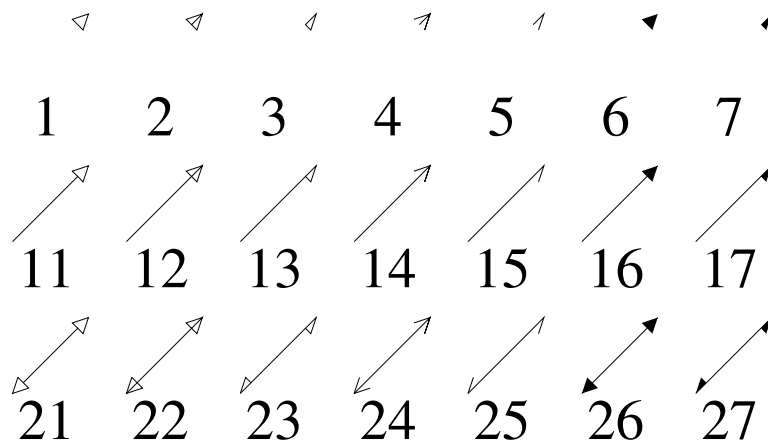


Figure 2 Sample of AROHD

(3) その他

- ・ 矢印の線種はPENX, NEWPEN等で指定しておく必要がある

4.5 AXIS

(1) 機能

線形の座標軸を描く(GPSL準拠)

(2) 呼び出し形式

CALL AXIS(X,Y,NBCD,N,ALEN,THETA,FIRST,DELTA)

(X,Y) : [REAL*4] 軸の始点の座標

NBCD : [CHARACTER*N] 軸に付ける名称

N : [INTEGER*4] 軸に付ける名称の文字数

ALEN : [REAL*4] 軸の長さ

THETA : [REAL*4] 描く軸とX軸とのなす角度

FIRST : [REAL*4] 軸の最初の目盛数値

DELTA : [REAL*4] 軸の目盛間の増分値

(3) その他

- ・ 線種はLINSEL,PENX等の指定と無関係に実線である
- ・ 実際に表示する軸の長さはALEN*FCTRとなるので FACTORにより尺度因子を随時変更すること

4.6 BAR

(1) 機能

ハッチつきか、ハッチのない棒グラフを描く(GPSL準拠)

(2) 呼び出し形式

CALL BAR(X,Y,TH,H,W,SH,IH,N)

(X,Y) : [REAL*4] 棒グラフの左下の位置

TH : [REAL*4] 棒グラフがx軸となす角度

(W,H) : [REAL*4] 棒グラフの幅・高さ

SH : [REAL*4] ハッチする部分の高さ

IH : [INTEGER*4]

=1: ハッチングしない

2: 左下から右上にハッチングする

3: 右下から左上にハッチングする

>=4: 両方のハッチングをする

N : [INTEGER*4] 単位長さ(1m)当たりハッチの本数

(3) その他

- ・ 棒グラフおよびハッチの線種はPENX, LINSEL等で指定したものになる
- ただし、実線と破線のみ対応

4.7 CIRC1

(1) 機能

中心、半径で指定される円を描く(GPSL準拠)

(2) 呼び出し形式

CALL CIRC1(X,Y,R,LTY)

(X,Y) : [REAL*4] 中心のx,y座標

R : [REAL*4] 半径

LTY : [INTEGER*4] 線種を指定 =0;線種の変更は行なわない

(1:実線 2:破線 3:一点鎖線 4:二点鎖線 5:点線)

4.8 CIRC2

(1) 機能

中心、半径で指定される円を描く(GPSL準拠)

(2) 呼び出し形式

CALL CIRC2(X,Y,XA,YA,LTY)

(X,Y) : [REAL*4] 中心のx,y座標

(XA,YA) : [REAL*4] 円弧の点のx,y座標

LTY : [INTEGER*4] 線種を指定 =0;線種の変更は行なわない

(1:実線 2:破線 3:一点鎖線 4:二点鎖線 5:点線)

4.9 CIRC3

(1) 機能

2点および半径で指定される円を描く(GPSL準拠)

(2) 呼び出し形式

CALL CIRC3(X1,Y1,X2,Y2,R,LTY)

(X1,Y1) : [REAL*4] 円弧の点のx,y座標

(X2,Y2) : [REAL*4] 円弧の点のx,y座標

ABS(R) : [REAL*4] 半径

R > 0 : (X1,Y1)から(X2,Y2)に時計回りに円弧を描く

R < 0 : (X1,Y1)から(X2,Y2)に反時計回りに円弧を描く

LTY : [INTEGER*4] 線種を指定 =0;線種の変更は行なわない

(1:実線 2:破線 3:一点鎖線 4:二点鎖線 5:点線)

4.10 CIRCL

(1) 機能

円・円弧・螺旋を描く(GPSL準拠)

(2) 呼び出し形式

CALL CIRCL(X,Y,THO,THF,RO,RF,DI)

(X,Y) : [REAL*4] 始点のx,y座標

THO : [REAL*4] 始点を通る半径がx軸となす角度

THF : [REAL*4] 終点を通る半径がx軸となす角度

RO : [REAL*4] 始点における曲率半径

RF : [REAL*4] 終点における曲率半径

DI : [REAL*4]

=0.0 : 実線

=0.5 : 破線

その他: 線種を変更しない

4.11 CNTRL

(1) 機能

配列に設定されているデータ点間を一点鎖線で結ぶ(GPSL準拠)

(2) 呼び出し形式

CALL CNTRL(X,Y,N,INC)

N : [INTEGER*4] 点列の点の数

X,Y : [REAL*4; 配列N] 点列の各点の座標

INC : [INTEGER*4]

=1:データを連続して選択

n:データを(n-1)個おきに選択

4.12 CONTR

(1) 機能

等値線を描く

(2) 呼び出し形式

CALL CONTR(Z,LMAX,IMAX,JMAX,C1,C2,C3)

Z(LMAX,JMAX) : [REAL*4] 等値線を描くための値が入っている配列

IMAX,JMAX : [INTEGER*4] 上記配列の内、等値線を描く領域の大きさ

C1,C2,C3 : [REAL*4] 最初の等値線の値、最後の等値線の値、

および等値線の間隔

(3) その他

- ・ 環境設定には次のサブルーチンを使用する

SETDRW : 配列Zの図面への投影法の定義

SETDXDY: 可変格子の設定

SETT3D : 3次元座標変換の設定

SETSPV : 欠損値処理の選択

SETVAL : 等値線値記入の選択、文字の属性の指定

なお、設定された環境は、環境設定サブルーチンにより

再度定義され直すまで有効である

4.13 DASHL

(1) 機能

配列に設定されているデータ点間を破線で結ぶ(GPSL準拠)

(2) 呼び出し形式

CALL DASHL(X,Y,N,INC)

N : [INTEGER*4] 点列の点の数

X,Y : [REAL*4; 配列N] 点列の各点の座標

INC : [INTEGER*4]

=1:データを連続して選択

n:データを(n-1)個おきに選択

4.14 DASHP

(1) 機能

破線を引く(GPSL準拠)

(2) 呼び出し形式

CALL DASHP(X,Y,W)

(X,Y) : [REAL*4] ペンの移動先

W : [REAL*4] 破線の間隔

(3) その他

- ・ 破線の開始位置はPLOTで予め設定しておく必要がある

4.15 DRAWTXT

(1) 機能

横書きテキストを描画する

(2) 呼び出し形式

CALL DRAWTXT(CM,N,YS,YINT,YD,MODE)

CM : [CHARACTER*N] 描画するテキストの文字列

N : [INTEGER*4] 文字列の長さ

YS : [REAL*4] 文字列の高さ

YINT : [REAL*4] 行間 (文字列の高さを含む)

YD : [REAL*4] 段落前の間隔

MODE : [INTEGER*4]

=0: 両端揃え

=1: 左揃え

=2: 中央揃え

=3: 右揃え

(3) その他

- ・ テキストの描画範囲を SETPAGE で予め設定しておく必要がある
 - ・ 英字フォントはSETFNTで設定する
 - ・ 漢字、かな文字フォントはSETKFNTで設定する
- ただし、JISについては対応していないので、EUC,Shift-JISコードに変換する必要がある
- ・ 漢字、かなコード(EUC,JIS,Shift-JIS)はプログラムで自動認識するが、

SETKCODEで手動設定することもできる

なお、EUC,Shift-JISの半角カナのみを利用する場合、文字化けするので、必ずSETKCODEで手動設定すること

- ・ 漢字、かな文字の文字列の長さはEUC,Shift-JISの場合、1文字で2になるが、半角カナはEUCでは2に、Shift-JISでは1になる
- JISの場合はShift-JISと同じだが、IOS 2022に準拠したコードが入るのでFortran組み込み関数LENで文字列の長さを調べることを勧める
- ・ 文字の色はPENX, NEWPEN等で指定したものとなる

4.16 DSCALE

(1) 機能

座標軸およびTICKマークを描く

(2) 呼び出し形式

CALL DSCALE(X0,Y0,X1,Y1,H,D,N)

(X0,Y0) : [REAL*4] 座標軸の始点の座標位置

(X1,Y1) : [REAL*4] 座標軸の終点の座標位置

H : [REAL*4] TICKマークの高さ

>0の場合、座標軸の上方につける

<0の場合、座標軸の下方につける

D : [REAL*4] TICKマークを入れる間隔

N : [INTEGER*4]

ABS(N) : D=0の場合、座標軸に表示するTICKマークの本数

N<0 : 座標軸を表示しない

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.17 ELIPS

(1) 機能

楕円・楕円弧を描く(GPSL準拠)

(2) 呼び出し形式

CALL ELIPS(X,Y,RM,RN,TH,THO,THF,IP)

(X,Y) : [REAL*4] 始点のx,y座標

RM,RN : [REAL*4] 楕円の長径,短径

TH : [REAL*4] 楕円の長径がx軸とのなす角度

THO : [REAL*4] 楕円弧の開始角(deg)

THF : [REAL*4] 楕円弧の終了角(deg)

IP : [INTEGER*4]

=2: 始点までペンを下げて移動

=3: 始点までペンを上げて移動

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.18 ELLIPS

(1) 機能

楕円・楕円弧を描く

(2) 呼び出し形式

CALL ELLIPS(X,Y,RM,RN,TH,THO,THF)

(X,Y) : [REAL*4] 中心のx,y座標

RM,RN : [REAL*4] 楕円の長径,短径

TH : [REAL*4] 楕円の長径がx軸とのなす角度

THO : [REAL*4] 楕円弧の開始角(deg)

THF : [REAL*4] 楕円弧の終了角(deg)

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.19 FLINE

(1) 機能

配列に設定されているデータ点間を直線または曲線で結ぶ(GPSL準拠)

(2) 呼び出し形式

CALL FLINE(X,Y,NPT,INC,LTP,ICOD)

X,Y : [REAL*4; 配列ABS(NPT)] 点列の各点の座標

NPT : [INTEGER*4]

ABS(NPT): 点列の点の数

>0: データ点間を直線で結ぶ

<0: データ点間をなめらかな曲線で結ぶ

INC : [INTEGER*4]

=1:データを連続して選択

n:データを(n-1)個おきに選択

LTP : [INTEGER*4]

=0:データ点間を実線で結ぶ

1:データ点間を実線で結び、その点にICODで指定した記号を描く

m:データ点間を実線で結び、(m-1)個毎にICODで指定した記号を描く

-1:データ点にICODで指定した記号を描く

-m:データ点の(m-1)個毎にICODで指定した記号を描く

ICOD : [INTEGER*4] センタシンボルに対応する 0-15 を指定する

4.20 GPL

(1) 機能

折れ線を描画する(KGRAF準拠)

(2) 呼び出し形式

CALL GPL(N,X,Y)

N : [INTEGER*4] 折れ線の点の数

X,Y : [REAL*4;配列N] 折れ線の各点の座標

(3) その他

・線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.21 GRID

(1) 機能

格子を描画する(GPSL準拠)

(2) 呼び出し形式

CALL GRID(X,Y,DX,DY,NX,NY)

X,Y : [REAL*4] 格子の左下端の座標

DX,DY : [REAL*4] x,y方向の格子の間隔

NX,NY : [INTEGER*4] x,y方向の格子の数

4.22 LINE

(1) 機能

配列に設定されているデータ点間を実線で結ぶ(GPSL準拠)

(2) 呼び出し形式

CALL LINE(X,Y,NPT,INC,LTP,ICOD)

X,Y : [REAL*4; 配列NPT] 点列の各点の座標

NPT : [INTEGER*4] 点列の点の数

IN : [INTEGER*4]

=1:データを連続して選択

n:データを(n-1)個おきに選択

LTP : [INTEGER*4]

=0:データ点間を実線で結ぶ

1:データ点間を実線で結び、その点にICODで指定した記号を描く

m:データ点間を実線で結び、(m-1)個毎にICODで指定した記号を描く

-1:データ点にICODで指定した記号を描く

-m:データ点の(m-1)個毎にICODで指定した記号を描く

ICOD : [INTEGER*4] センタシンボルに対応する 0-15 を指定する

4.23 LINE2

(1) 機能

直線を描画する

(2) 呼び出し形式

CALL LINE2(X0,Y0,X1,Y1)

(X0,Y0) : [REAL*4] 直線の始点

(X1,Y1) : [REAL*4] 直線の終点

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.24 NUMBER

(1) 機能

浮動小数点の数値を描く(GPSL準拠)

(2) 呼び出し形式

CALL NUMBER(X,Y,S,FPN,ARG,N)

(X,Y) : [REAL*4] 文字を書き出す左下の位置

=(999.0,999.0)の場合カレントポイントを変えない

=(999.0,999.0)->カレントポイントを変えず上付文字を書く

=(999.0,-999.0)->カレントポイントを変えず下付文字を書く

S : [REAL*4] 文字の高さ

FPN : [REAL*4] 浮動小数点の数値

ARG : [REAL*4] 文字列がX軸とのなす角度

N : [INTEGER*4] 数値の桁数を指定

>0: 小数点以下桁まで描く

=0: 小数点以上の数値と小数点を描く

<0: 整数部|N|桁以上の数値を描く

(3) その他

- ・ N>=0の場合、表示する数字が10を越えると自動的に指数表示する
- ・ 文字フォントはSETFNTで設定する
- ・ 文字の色はPENX, NEWPEN等で指定したものとなる
- ・ 環境設定には次のサブルーチンを使用する

SETNUM : 整数表示、指数表示の設定

なお、設定された環境は、環境設定サブルーチンにより

再度定義され直すまで有効である

4.25 NUMBER3

(1) 機能

浮動小数点の数値を描く

(2) 呼び出し形式

CALL NUMBER3(X,Y,Z,S,FPN,ARG,N)

ペンの移動先XX,YYは以下のように変換される

$$XX=X03D+(X-X03D)*(1-(Y-Y03D)/YM3D)*X3D+(Y-Y03D)*X2D$$

$$YY=Y03D+(Y-Y03D)*(1-(X-X03D)/XM3D)*Y3D+(X-X03D)*Y2D$$

$$+Z*(1-(Y-Y03D)/YM3D)*X3D$$

S : [REAL*4] 文字の高さ

FPN : [REAL*4] 浮動小数点の数値

ARG : [REAL*4] 文字列がX軸とのなす角度

N : [INTEGER*4] 数値の桁数を指定

>0: 小数点以下桁まで描く

=0: 小数点以上の数値と小数点を描く

<0: 整数部IN桁以上の数値を描く

(3) その他

- ・ N>=0の場合、表示する数字が10を越えると自動的に指数表示する
- ・ 文字フォントはSETFNTで設定する
- ・ 文字の色はPENX, NEWPEN等で指定したものとなる
- ・ 環境設定には次のサブルーチンを使用する

SETT3D : 3次元座標変換の設定

SETNUM : 整数表示、指数表示の設定

なお、設定された環境は、環境設定サブルーチンにより再度定義され直すまで有効である

4.26 PLOT

(1) 機能

直線を引く・原点の移動(GPSL準拠)
ページ替え・P Sファイル終了の宣言

(2) 呼び出し形式

CALL PLOT(X,Y,NS)

(X,Y) : [REAL*4] ペンの移動先

NS : [INTEGER*4]

= 2: ペンを下げて移動

3: ペンを上げて移動

-2: ペンを下げて移動し、移動先に原点を移す

-3: ペンを上げて移動し、移動先に原点を移す

5: ページ替え

999: P Sファイル終了の宣言

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

- ・ ページを替えると原点移動は初期化される

4.27 PLOT1

(1) 機能

折れ線を描く (GPSL準拠)

(2) 呼び出し形式

CALL PLOT1(X,Y,N)

N : [INTEGER*4] 折れ線の点の数

X,Y : [REAL*4; 配列N] 折れ線の各点の座標

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.28 PLOT3

(1) 機能

直線を引く

(2) 呼び出し形式

CALL PLOT3(X,Y,Z,NS)

ペンの移動先XX,YYは以下のように変換される

$$XX=X03D+(X-X03D)*(1-(Y-Y03D)/YM3D)*X3D+(Y-Y03D)*X2D$$

$$YY=Y03D+(Y-Y03D)*(1-(X-X03D)/XM3D)*Y3D+(X-X03D)*Y2D \\ +Z*(1-(Y-Y03D)/YM3D)*X3D$$

NS : [INTEGER*4]

=2: ペンを下げて移動

3: ペンを上げて移動

5: ページをかえる

999: 終了を宣言する

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する
- ・ 環境設定には次のサブルーチンを使用する

SETT3D : 3次元座標変換の設定

なお、設定された環境は、環境設定サブルーチンにより

再度定義され直すまで有効である

4.29 RECT

(1) 機能

長方形を描く(GPSL準拠)

(2) 呼び出し形式

CALL RECT(X0,Y0,DH,DW,TH,NP)

(X0,Y0) : [REAL*4] 長方形の左下の位置

(DW,DH) : [REAL*4] X,Y軸方向の増分

TH : [REAL*4] 長方形の底辺とX軸とのなす角度

NP : [INTEGER*4]

=2 : (X0,Y0)までペンを下げて移動する

その他 : (X0,Y0)までペンを上げて移動する

(3) その他

・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.30 SMOOT

(1) 機能

点列間をなめらかな曲線または直線で結ぶ(GPSL準拠)

(2) 呼び出し形式

CALL SMOOT(X,Y,NS)

(X,Y) : [REAL*4] ペンの移動先

NS : [INTEGER*4]

=0: 曲線を描き始める最初の点を指定する

-1: 曲線を描き始める最初の点を指定する

-2: ペンを下げて移動し、なめらかな曲線で結ぶ

-3: ペンを上げて移動

<-24: 曲線の最終点であることを指定し、一連の作業を終了する

2: ペンを下げて移動し、直線で結ぶ

3: ペンを上げて移動

(3) その他

・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する

4.31 STREAM

(1) 機能

流跡線を描画する

(2) 呼び出し形式

CALL STREAM(U,V,WK,LMAX,IMAX,JMAX)

U(LMAX,JMAX) : [REAL*4] x成分の値が入っている配列

V(LMAX,JMAX) : [REAL*4] y成分の値が入っている配列

WK(LMAX,JMAX) : [REAL*4] work配列

IMAX,JMAX : [INTEGER*4] 上記配列の内、流跡線を描画する領域の大きさ

(3) その他

- ・ 環境設定には次のサブルーチンを使用する

SETDRW : 配列Zの図面への投影法の定義

SETDXDY: 可変格子の設定

SETT3D : 3次元座標変換の設定

SETSTRM: 流跡線の間隔および矢印の間隔・大きさの指定

(2方向の実際の格子幅が同じなら、特に設定する必要はない)

なお、設定された環境は、環境設定サブルーチンにより

再度定義され直すまで有効である

4.32 SYMBOL

(1) 機能

文字を描く(GPSL準拠)

(2) 呼び出し形式

CALL SYMBOL(X,Y,S,C,ARG,N)

(X,Y) : [REAL*4] 文字を書き出す左下の位置

=(999.0,999.0)->カレントポイントを変えない

=(-999.0,999.0)->カレントポイントを変えず上付文字を書く

=(999.0,-999.0)->カレントポイントを変えず下付文字を書く

S : [REAL*4] 文字の高さ

C : [CHARACTER] 文字列

ARG : [REAL*4] 文字列がX軸とのなす角度

N : [INTEGER*4] 文字列の長さ

N<0の場合

(X,Y) : [REAL*4] センターシンボルの中心位置

S : [REAL*4] センターシンボルの高さ

C : [INTEGER*4] (0-15)でセンターシンボルの種類を指定

N=-1 : (X,Y)までペンを上げて移動

-2 : (X,Y)までペンを下げて移動

センターシンボルの表示例が Figure 3 にある

(3) その他

- ・ 英字フォントはSETFNTで設定する
 - ・ 漢字、かな文字フォントはSETKFNTで設定する
 - ・ 漢字、かなコード(EUC,JIS,Shift-JIS)はプログラムで自動認識するが、SETKCODEで手動設定することもできる
- なお、EUC,Shift-JISの半角カナのみを利用する場合、文字化けするので、必ずSETKCODEで手動設定すること
- ・ 漢字、かな文字の文字列の長さはEUC,Shift-JISの場合、1文字で2になるが、半角カナはEUCでは2に、Shift-JISでは1になる
 - JISの場合はShift-JISと同じだが、IOS 2022に準拠したコードが入るのでFortran組み込み関数LENで文字列の長さを調べることを勧める
 - ・ 漢字、かなフォントを縦書きに指定した場合、文字列に英字が入ると正常に表示しない
 - また、上付(下付)文字も正常に表示しない
 - ・ 文字の色はPENX, NEWPEN等で指定したものとなる

4.33 SYMBOL3

(1) 機能

文字を描く(GPSL準拠)












			+	×					Y
0	1	2	3	4	5	6	7	8	9
					—				
10	11	12	13	14	15				

Figure 3 Sample of center symbol

(2) 呼び出し形式

CALL SYMBOL3(X,Y,Z,S,C,ARG,N)

ペンの移動先XX,YYは以下のように変換される

$$XX=X03D+(X-X03D)*(1-(Y-Y03D)/YM3D)*X3D+(Y-Y03D)*X2D$$

$$YY=Y03D+(Y-Y03D)*(1-(X-X03D)/XM3D)*Y3D+(X-X03D)*Y2D \\ +Z*(1-(Y-Y03D)/YM3D)*X3D$$

S : [REAL*4] 文字の高さ

C : [CHARACTER] 文字列

ARG : [REAL*4] 文字列がX軸とのなす角度

N : [INTEGER*4] 文字列の長さ

N<0の場合

C : [INTEGER*4] (0-15)でセンターシンボルの種類を指定

N=-1 : (X,Y)までペンを上げて移動

-2 : (X,Y)までペンを下げて移動

(3) その他

- ・ 英字フォントはSETFNTで設定する
- ・ 漢字、かな文字フォントはSETKFNTで設定する
- ・ 漢字、かな文字についての注意事項はSYMBOLを参照すること
- ・ 文字の色はPENX, NEWPEN等で指定したものとなる
- ・ 環境設定には次のサブルーチンを使用する

SETT3D : 3次元座標変換の設定

なお、設定された環境は、環境設定サブルーチンにより

再度定義され直すまで有効である

4.34 UVHPlot

(1) 機能

配列(U,V)が与えられたとき、水平風ベクトル／矢羽を描く

(2) 呼び出し形式

UVHPlot(U,V,DNS,LMAX,IMAX,JMAX,CD,ISP)

U(LMAX,JMAX) : [REAL*4] 風のX成分の値が入っている配列

V(LMAX,JMAX) : [REAL*4] 風のY成分の値が入っている配列

DNS(LMAX,JMAX) : [REAL*4] 風成分を密度で割る必要がある場合、
密度の値が入っている配列を入力する

DNS = 0.0 : 風成分を密度で割る必要がない場合
 IMAX, JMAX : [INTEGER*4] 上記配列の内、塗りつぶしを行なう
 領域の大きさ
 CD : [REAL*4] X方向1格子間のベクトルの大きさ、
 または1つの全矢羽に対する風速
 CD = 0.0 : 自動設定する
 ISP : [INTEGER*4]
 ISP > 0 : 水平風ベクトルを描く格子間隔
 ISP = 0 : 水平風ベクトルを描く格子間隔を自動設定する
 ISP < 0 : 矢羽を格子間隔 ABS(ISP) で描く

(3) その他

- ・ このサブルーチンを呼び出す前に次のどちらかの投影法を定義しておく必要がある
 - SETDRW : 配列U, Vのx-y図面への投影法の定義
 - SETDRW2 : 配列U, Vの心射(r-θ)図面への投影法の定義
 - SETDRW3 : 配列U, Vの心射(θ-r)図面への投影法の定義
 - ・ 矢羽の大きさは SETUVPL で指定しない限り、自動設定される
 - ・ 凡例の位置は SETUVPL で指定しない限り、図の右下に表示される
表示させないためには、SETUVPL で文字 0.0 で設定すること
 - ・ 環境設定には次のサブルーチンを使用する
 - SETDXDY : 可変格子の設定
 - SETUVPL : 水平風ベクトル/矢羽の色・大きさ、凡例の位置
- なお、設定された環境は、環境設定サブルーチンにより再度定義され直すまで有効である

4.35 YABANE

(1) 機能

矢羽を描く

(2) 呼び出し形式

CALL YABANE(X, Y, ANG, MET, FL)

(X, Y) : [REAL*4] 矢羽を描く線分の始点

ANG : [REAL*4] 風向 (360度方位)

MET : [INTEGER*4]

= 1: 半矢羽

= 2: 全矢羽

= 10: ペナント (旗)

FL : [REAL*4] 1つの全矢羽の長さ

(3) その他

- ・ 線種はLINSEL, GSLN, GSLWSC, GSPLCI, NEWPEN, PENX等で指定する
- ・ ペナントの塗りつぶしの色は線の色と同じになる