ＤＣＬマニュアル

稲津 將　(北海道大学大学院地球環境科学研究科大気海洋圏環境科学専攻)

平成 10 年　4 月　15 日

1　ＧＲＯＰＮ (ウインドー = オープン) の前の設定

1-1 SW#SET(内部変数、値)

# (型)	内部変数	内容	初期値
I (整数型)	'IWIDTH'	ウインドーの横サイズ	900
I (整数型)	'IHEIGHT'	ウインドーの縦サイズ	650
I (整数型)	'IPOSX'	ウインドーの位置 (x 座標)	-999(マウスで指定)
I (整数型)	'IPOSY'	ウインドーの位置 (y 座標)	-999(マウスで指定)
L (論理型)	'LWAIT'	自動 (マウスなしで) 改ページ (.false.)	.true.
L (論理型)	'LWAIT0'	自動 (マウスなしで) オープン (.false.)	.true.
L (論理型)	'LWAIT1'	自動 (マウスなしで) クローズ (.false.)	.true.
L (論理型)	'LCOROL'	ＰＳ＝ＦＩＬＥをカラーに (.true.)	.false.
L (論理型)	'LSEP'	ＰＳ＝ＦＩＬＥを１ページづつ分割 (.true.)	.false.
L (論理型)	'LPRINT'	作成したＰＳ＝ＦＩＬＥを直ちに印刷 (.true.)	.false.
C (文字型)	'TITLE'	Ｘウインドーのタイトル	'*'
C (文字型)	'FNAME'	ＰＳ＝ＦＩＬＥの名称	'dcl'

使用例

swlset('lwait',.true.) : たくさんのページにわたる画像でマウスクリックに手が疲れそうな時

swlset('lsep',.true.) : 部数が多くなりプリンターが受け付けないおそれがある時、動画 (アニメーション) 処理をする時

swlset('fname','*') : 現在実行中のファイル名をＰＳ＝ＦＩＬＥ名に使う時

補足

1.'IPOSX' の値は左端で'IPOSY' の値は上端で 1 となっています。

2. ＰＳ＝ＦＩＬＥの名称を指定しないと、多くの場合それでも構わないだろうが、実行の度に既存の dcl.ps を上書きする事になります。

1-2 SGPWSN : '1:X 2:PS 3:Tek' と出力先の表示を出す。

補足 次のようにコーディングするとよい。

call sgpwsn : 出力先の表示

read(5,*) iws : 出力番号の読み込み

call gropn(iws) : 次章参照

2　ＧＲＯＰＮ (ウインドー = オープン)

2 GROPN(出力番号 (I)) : 出力装置のオープンをする。

使用例

gropn(1) : 出力先をＸ = ＷＩＮＤＯＷにする時

gropn(2) : 出力先をＰＳ = ＦＩＬＥにする時

補足 この値を負にすると縦横反転する。

3　ループの中に入れないレイアウトに関する設定

3-1 SG# SET (内部変数、値) : レイアウトに関する細かい設定

# (型)	内部変数	内容	初期値
I (整数型)	'INDEX'	マージンに描く文字列の線の色と太さ	1
L (論理型)	'LCORNER'	コーナーマークを書く (.true.)	.true.
R (実数型)	'CORNER'	コーナーマークの長さ	0.01
L (論理型)	'LWIDE'	タイトルマージンをとる (.true.)	(.false.)
R (実数型)	'XPAD'	タイトルの左右のすき間	1.0
R (実数型)	'YPAD'	タイトルの上下のすき間	1.0
L (論理型)	'LFULL'	画面いっぱいに作画 (.true.), 最大内接正方形 (.false.)	.false.
L (論理型)	'LCLIP'	地図などをはみ出して書かない (.true.)	.false.

使用例

sglset('index',23) : マージンに描く文字列の色と太さの変更 (赤、太) する時

sglset('lcorner',.false.) : コーナマークを書かないようにした方がよい時

sglset('lfull',.true.) : 図を用紙いっぱいに書く時

sglset('lclip',.false.) : 図をはみ出して書く時

補足

1. 線の色と太さの指定については最後のページに記載しています。

2.'LFULL' は GRSVPT を深い関係にある。その関係についてはＧＲＳＶＰＴの章を参照しています。

3-2 SLSIZE(' 出力用紙'(C2)//' 縦横指定'(C1)) : 出力用紙の設定をする。

使用例

slsize('a4t') : Ａ４縦使い (このルーチンを省略した時の値)

slsize('b5y') : Ｂ５横使い

3-3 SLDIV(' 縦並びか横並びか'(C1), 横の分割数 (I), 縦の分割数 (I)) : １ページに複数の図を並べる。

使用例

sldiv('t',3,4) : 縦並びで横に３列、横に４列、計１２枚の図を１ページに出力する。

sldiv('y',2,1) : 横並びで横に２列、計２枚の図を１ページに出力する。**sldiv('y',1,1)** : １ページに図を１枚出力する。(このルーチンを省略した時の値)

補足 SLDIV は後述の GRSVPT と深い関係にある。その対応表はＧＲＳＶＰＴのところにあります。

3-4 SLMGN(左辺 (R), 右辺 (R), 下辺 (R), 上辺 (R)) : 図のマージンを決める

使用例 **slmgn(0.1,0.1,0.1,0.1)** : 図全体の左右上下に 1 割のマージンをとる。

補足

1. マージンの値はもっとも長い辺に対する割合として定める。
2. マージンの部分に SLSTTL (後述) でタイトル等を書かせるので、それに見合うだけのマージンを確保する必要がある。
3.SLDIV と SLMGN を繰り返して呼ぶ事ができる。次のようにコーディングすると 1 ページをまず横に 2 分割し 1 割のマージンを確保して、その各々に対してをさらに横に 3 分割、縦に 3 分割し、 5 %のマージンを確保をして、その各々に図を書く。

call sldiv('t',2,1) : 1 ページを横に 2 分割する。
call slmgn(0.1,0.1,0.1,0.1) : 1 割のマージンをとる。
call sldiv('y',3,3) : さらに横に 3、縦に 3 分割する。
call slmgn(0.05,0.05,0.05,0.05) : 5 %のマージンをとる。

3-5 SLRAT(横の割合 (R), 縦の割合 (R)) : 縦横比を指定してはみ出した部分をマージンとする。

使用例 **slrat(1.0,1.0)** : 図を書かせる領域を正方形型にする。

補足 SLMGN と似ているが、マージンとは別に図の比を定めるのに用いる。このルーチンを用いるとやや図が小さくなる可能性がある。

3-6 UZFACT(文字の大きさの倍率 (R)) : 図に付け加える文字の大きさを一定の大きさ倍にする。

使用例 **uzfact(0.6)** : 全ての文字を 0.6 倍する

補足 : これをループの中に入れると 0.6 の n 乗倍されるので注意が必要です

4 ループの中でもよいレイアウトの設定

4-1 SLSTTL(' タイトル名'(C),' タイトルの位置'(C), 左右の位置 (R), 上下の位置 (R),1) : マージンにページのタイトルを書かせる。

使用例

slsttl('TITLE','t',0.0,0.0,1):トップマージンの中央に'TITLE' の文字を書かせる。
slsttl('# TIME # DATE # PAGE','b',1.0,1.0,1): ボトムマージンの左上に時間と日付、ページ数をかかせる。

補足

1. タイトルはマージンに書かせるため SLMGN でマージンを確保する必要がある。
2. タイトル名として日付、時間、ページ数を指定する事ができる。(使用例参照)

4-2 SLPWWR(線の色と太さ (I)) : 図を書く領域の枠を書く

使用例 **slpwwr(43)** : 図を書く領域のコーナマーク (青色に設定)

補足

1. 線の色と太さの指定については最後のページに記載しています。

2.SLPWVR(I),SLPVPR(I) もある。

4-3 SLPWWC(線の色と太さ (I), コーナマークの長さ (R)) : 図を書く領域のコーナーマークを書く

使用例 **slpwwc(43,0.1)** : 図を書く領域のもっとも長い辺を 1 とする時の割合で 1 割の長さのコーナーマークを書く。(緑色に設定)

補足

1. 線の色と太さの指定については最後のページに記載しています。
2. SLPWVC(I,R),SLPVPC(I,R) もある。

5 図の初期化

5-1 GRFRM :次の 2 つの用途があり、図を書く時には必ず一回は呼ぶ。

1. いちばん最初の図を書く。(初期化)
2. 次の図を書くため、図を書く領域を移動する。

補足 ＧＲＳＶＰＴ (後述) を用いて設定した枠内で図を書く。それ以外の場所にはみ出した部分は描かれない。

5-2 GRFIG : 同じ領域に次の図を書く。

補足

1. ＧＲＳＶＰＴ (後述) を用いて設定した枠内で図を書く。それ以外の場所にはみ出した部分は描かれない。
2. ＧＲＦＲＭと併用、もしくはＧＲＦＩＧを繰り返し用いる事で、同じ領域に複数の図を描く事ができる。次のようにコーディングするとよい。

call grfrm 初期化
********** ＧＲＳＶＰＴを使って、１つ目の図を書かせる。
call grfig 初期化するが次の領域に移動しない
********** ＧＲＳＶＰＴを使って、２つ目の図を書かせる。
call grfrm 初期化して次の領域に移動する。

6 図を書く領域の指定

6-1 GRSWND(横軸の最小値 (R), 横軸の最大値 (R), 縦軸の最小値 (R), 縦軸の最大値 (R)) : 横軸、縦軸の物理量の範囲指定をする。どのような物理量かは場合により異なる。分かりやすく例をあげて書くと、次のようになる。

1. 折れ線図 :$y = f(x)$ のような場合。通常は、横軸を x に、縦軸を yにとる。その時は、図を書かせる範囲として、x、yの最大値、最小値を設定する。これは、必ずしも定義域と値域に同じくなくてよい。ただし、ある程度写像の写す先の情報がないと、設定が難しい。
2. 等高線図、ベクトル図 :$z = f(x, y)$ のような場合。通常は、横軸を x に、縦軸を yにとる。その時は、図を書かせる範囲として、x、yの最大値、最小値を設定する。この場合は、写像の情報は不要である。
3. 地図投影図 :不要な事もある。描くべき経度と緯度の最大値、最小値を設定するとよい。

一応、数学的記述は次の通りである。

	Ω	Δ	G_1	G_2
折れ線図	$\Omega \subset R$	$\Delta \subset R$	Ω	Δ
等高線図	$\Omega \subset R^2$	$\Delta \subset R$	Ω_x	Ω_y
地図投影	$\Omega \subset S^2$	$\Delta \subset R$	λ (経度)	ϕ (緯度)
ベクトル図	$\Omega \subset R^2$	$\Delta \subset R^2$	Ω_x	Ω_y

但し、横軸に書く物理量 G_1、縦軸に書く物理量 G_2とし、函数 $f : \Omega \to \Delta$とする。また、$\Omega \subset R^2$の時、$\Omega = \Omega_x \times \Omega_y$とする。

6-2 GRSVPT(領域の左 (R), 領域の右 (R), 領域の下 (R), 領域の上 (R)) : 図を書く領域の指定をする。値は今までの中で設定した図を書く領域の最大辺を 1 とする割合で右下を基準点としている。$\mathrm{SGLSET('LFULL',.TRUE.)}$ と SLDIV と大きな関係にある。次の表にまとめてみた。ここにいう横縦分割比とは $\mathrm{SLDIV('T'}$, 横, 縦$)$ の横:縦であり、各横縦分割比に対する最大限の $\mathrm{GRSVPT(R_1,R_2,R_3,R_4)}$ の値を表にした。

SGLSET('LFULL',.TRUE.) のとき

横縦分割比	R_1	R_2	R_3	R_4	横縦分割比	R_1	R_2	R_3	R_4
1:1	0.0	1.0	0.0	0.721	-:-	–	–	–	–
1:2	0.0	1.0	0.0	0.360	2:1	0.0	0.692	0.0	1.0
1:3	0.0	1.0	0.0	0.240	3:1	0.0	0.461	0.0	1.0
1:4	0.0	1.0	0.0	0.180	4:1	0.0	0.346	0.0	1.0
1:5	0.0	1.0	0.0	0.144	5:1	0.0	0.277	0.0	1.0
1:6	0.0	1.0	0.0	0.120	6:1	0.0	0.230	0.0	1.0
1:7	0.0	1.0	0.0	0.103	7:1	0.0	0.198	0.0	1.0
2:3	0.0	1.0	0.0	0.481	3:2	0.0	0.923	0.0	1.0
2:5	0.0	1.0	0.0	0.288	5:2	0.0	0.553	0.0	1.0
2:7	0.0	1.0	0.0	0.206	7:2	0.0	0.395	0.0	1.0
3:4	0.0	1.0	0.0	0.541	4:3	0.0	1.0	0.0	0.961
3:5	0.0	1.0	0.0	0.432	5:3	0.0	0.830	0.0	1.0
3:7	0.0	1.0	0.0	0.309	7:3	0.0	0.593	0.0	1.0
4:5	0.0	1.0	0.0	0.577	5:4	0.0	1.0	0.0	0.901
4:7	0.0	1.0	0.0	0.412	7:4	0.0	0.791	0.0	1.0
5:6	0.0	1.0	0.0	0.601	6:5	0.0	1.0	0.0	0.865
5:7	0.0	1.0	0.0	0.515	7:5	0.0	0.988	0.0	1.0
6:7	0.0	1.0	0.0	0.618	7:6	0.0	1.0	0.0	0.831

SGLSET('LFULL',.FALSE.) のとき

最大限の $\mathrm{GRSVPT(R_1,R_2,R_3,R_4)}$ は常に $\mathrm{GRSVPT(0.0,1.0,0.0,1.0)}$ である。

補足

1. あまり詰めすぎないためにも、この表の値に対して 1 割程度は余白を考えておくべきである。
2. 大きいタイトルのためには (左右 0.15、上下 0.2)、小さいタイトルのためには (左右 0.10、上下 0.13)、目盛のためには (左右 0.04、上下 0.05) だけ余白をとればよい。

7　地図座標変換

7-1 SG#SET(内部変数、値)

# (型)	内部変数	内容	初期値
R (実数型)	'STLAT1'	何らかの角度 (45.0 が良い)	
R (実数型)	'STLAT2'	何らかの角度 (30.0 が良い)	

補足 : 後述の地図番号 20 番台の地図投影にのみ使う。

7-2 GRSTXY(経度最小 (R), 経度最大 (R), 緯度最小 (R), 緯度最大 (R)) : 作画する地図の緯度と経度の範囲を指定する。

使用例

grstxy(-180.0,180.0,-90.0,90.0) : 後述の地図番号 30 番以外の方法にて全球の地図投影する

grstxy(-180.0,180.0,0.0,90.0) : 後述の地図番号 30 番の方法で半球投影する

7-2 GRSSIM(拡大倍率 (R), 横方向平行移動 (R), 縦方向平行移動 (R)) : 地図のための相似変換を行なう。(章末の表参照)

7-3 GRSMPL(投影中心経度 (R), 投影中心緯度 (R), 回転角 (R)) : 地図のための回転変換を行なう。

7-4 GRSTRN(座標番号 (I)) : 座標系を決める。座標系の座標番号は以下の通り。

直交座標系	円錐座標系	円筒座標系	方位座標系
1:一様	10:円筒	20:正距円錐	30:正射
2:横軸対数	11:メルカトル	21:ランンベルト正積	31:ポーラステレオ
3:縦軸対数	12:モルワイデ	22:ランベルト正角	32:正距方位
4:両対数	13:ハンメル	23:ボンヌ	
5:極	14:エケルト		

使用例 : **sglset('lfull',.true.)**、**grsvpt(0.02,0.98,0.05,0.68)** の設定の後に各々の地図番号に対応して GRSSIM,GRSMPL を次のように設定すればよい。

地図番号 (GRSTRF)	相似変換 (GRSSIM)	回転変換 (GRSMPL)
11(メルカトル)	(0.15,0.0,0.0)	(経度,90.0,0.0)
12(モルワイデ)	(0.169,0.0,0.0)	(経度,90.0,0.0)
13(ハンメル)	(0.169,0.0,0.0)	(経度,90.0,0.0)
14(エケルト)	(0.172,0.0,0.0)	(経度,90.0,0.0)
20,21,22	(0.1,0.0,0.0)	(0.0,90.0,0.0)
23(ボンヌ)	(0.15,0.0,0.19)	(0.0,90.0,0.0)
30(正射)	(0.3,0.0,0.0)	(経度, 緯度, 回転角)
32(正距方位)	(0.1,0.0,0.0)	(経度, 緯度, 回転角)

補足

1.10 番台の地図投影で回転変換は投影する中心の座標の経度を設定する。

2.30 番台の地図投影で回転変換は投影する中心の座標の経度緯度と回転角を設定する。

8　ＧＲＳＴＲＦ (領域および変換の決定)

8 GRSTRF : 領域および変換の決定

9　軸に関する設定

9-1 UZ#SET(内部変数、値) : 軸に関する細かい設定

# (型)	内部変数	値	初期値
R (実数型)	'UXUSER'	縦軸の位置 (横座標=R)	不定
R (実数型)	'UYUSER'	横軸の位置 (縦座標=R)	不定
R (実数型)	'ROFF'//' 縦横'//' 位置'	ラベル、タイトルと軸との距離	0.0
R (実数型)	'ROFG'//' 縦横'//' 位置'	座標軸の平行移動	0.0
L (論理型)	'LABEL'//' 縦横'//' 位置'	ラベルを書く (.true.)	.false.
I (整数型)	'IROTL'//' 縦横'//' 位置'	ラベルの文字の回転 (90×I)	0.0
I (整数型)	'IROTC'//' 縦横'//' 位置'	タイトルの文字の回転 (90× I)	横軸 0.0 縦軸 1.0
I (整数型)	'ICENT'//' 縦横'//' 位置'	ラベルのセンタリング (-1:左 0:中 1:右)	横軸 0.0 縦軸 1.0
I (整数型)	'INDEXT'//' 大小'	座標軸、目盛の線の色と太さ	横軸 0.0 縦軸 1.0
I (整数型)	'INDEXL'//' 大小'	ラベル、タイトルの線の色と太さ	横軸 0.0 縦軸 1.0
R (実数型)	'RSIZET'//' 大小'	目盛の長さ	横軸 0.007, 縦軸 0.014
R (実数型)	'RSIZEL'//' 大小'	ラベルの文字の高さ	横軸 0.007, 縦軸 0.014
R (実数型)	'RSIZEC'//' 大小'	タイトルの文字の高さ	横軸 0.028, 縦軸 0.035
I (整数型)	'IFLAG'	軸のラベルの位置 (正の値:上右、負の値:左下)	
I (整数型)	'INNER'	目盛を内むけにするか否か (正の値:内、負の値:外)	
L (論理型)	'LOFFSET'	リニアオフセットする (.true.)(サンプル参照)	.false.
R (実数型)	'XOFFSET'	横軸の値の平行移動	0.0
R (実数型)	'YOFFSET'	縦軸の値の平行移動	0.0
R (実数型)	'XFACT'	横軸の値の拡大縮小	1.0
R (実数型)	'YFACT'	縦軸の値の拡大縮小	1.0

使用例

uzrset('uxuser',0.0) : $x = 0$(縦軸) をユーザ指定の軸とする。
uzrset('uyuser',0.0) : $y = 0$(横軸) をユーザ指定の軸とする。
uzrset('roffyl',0.1),uzrset('rofgyl',0.1) : 二本目の軸を書く時に指定する
uzlset('loffset',.true.) :二本目の軸を書くが目盛の値を異なる尺度にしたい時指定
uzrset('yoffset',273.15) :今まで 0 であったところを 273.15 にする。
uzrset('yfact',0.6) :今までの目盛を 0 を中心に 0.6 倍する。
uziset('indext1',23) :小さな目盛の色と太さの指定 (赤)
uziset('indext2',113) : 大きなタイトルの色と太さの指定 (紫)

補足

1. 線の色と太さの関する指定については、最後のページ参照。
2. 長さに関する指定は全て図を書く領域の最大辺を 1 とした時の比である。

3.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。
4.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t')、ユーザーによる指定 ('u') の 5 種類がある。' 縦横' との組み合わせに注意がいる。
5.' 大小' とは指定する目盛、ラベルが大きいものに対する指定 ('1') か、小さいものに対する指定か ('2') を示している。

9-2 U' 縦横'AXDX(' 位置', 小目盛の値の間隔 (R), 大目盛の値の間隔 (R)) : 等間隔の目盛を持つ軸を書く。

使用例

uxaxdx('b',0.2,1.0) :横軸を下部に書き、小さな目盛を 0.2 刻みに、大きな目盛を 1.0 刻みにする軸を書く。
uyaxdx('l',0.2,1.0) :縦軸を左端に書き、小さな目盛を 0.2 刻みに、大きな目盛を 1.0 刻みにする軸を書く。

補足

1.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。
2.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t')、ユーザーによる指定 ('u') の 5 種類がある。' 縦横' との組み合わせに注意がいる。

9-3 U' 縦横'AXLB(引数は補足参照) : 不等間隔の目盛を持つ軸を書く。

補足

1. このルーチンの引数は、U' 縦横'AXLB(' 位置', 小目盛の配列 (R), 小目盛の配列の大きさ (I), 大目盛の配列 (R),' ラベル'(C), ラベルの文字の長さ (I), 大目盛の配列の大きさ (I)) である。2.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。
3.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t')、ユーザーによる指定 ('u') の 5 種類がある。' 縦横' との組み合わせに注意がいる。

9-4 U' 縦横'STTL(' 位置', タイトル (C), タイトルの位置 (R)) : 小さいのタイトルを書かせる。

使用例 **uxsttl('b','inaz',0.0)** :下端の書いた横軸に'inaz' というタイトルを書く。

補足

1. タイトルの位置は左端 (-1.0)、センター (0.0)、右端 (1.0) である。

2.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。
3.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t')、ユーザーによる指定 ('u') の 5 種類がある。' 縦横' との組み合わせに注意がいる。

9-5 U' 縦横'MTTL(' 位置', タイトル (C), タイトルの位置 (R)) : 大きなのタイトルを書かせる。

使用例 **uxmttl('b','dcl',0.0)** :下端の書いた横軸に'dcl' というタイトルを書く。

補足

1. タイトルの位置は左端 (-1.0)、センター (0.0)、右端 (1.0) である。
2.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。
3.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t')、ユーザーによる指定 ('u') の 5 種類がある。' 縦横' との組み合わせに注意がいる。

9-6 UL' 縦横'LOG(' 位置', ラベルの表示 (I), 1 桁あたりの目盛数 (I)) : 対数軸を書かせる。

使用例 **ulxlog('b',1,9)** : 下端に 1 桁に 9 個目盛を置き、然るべきラベルを書いた対数軸 (横軸) を書く。

補足

1. ラベルの表示とは $1{:}10^n$,$2{:}10^n$ と 2.0×10^n, $3{:}10^n$ と 2.0×10^n と 5.0×10^n にラベルを置くことを意味する。
2.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。
3.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t')、ユーザーによる指定 ('u') の 5 種類がある。' 縦横' との組み合わせに注意がいる。

9-7 UC# SET(内部変数、値) : 日付に関する細かい設定をする。

# (型)	内部変数	値	初期値
I (整数型)	'NDAYS'	I 日毎に目盛を入れる	不定
I (整数型)	'NCHAR'	月名を I 文字まで書く	不定
L (論理型)	'LOWER'	月名の 2 文字以降を小文字にする (.true.)	.false.

使用例

uciset('ndays',10) : １０日間隔で日付軸に目盛をふる。
uciset('nchar',3) : 月の名称を 3 文字まで書く。JAN のように。

9-8 UC' 縦横'ACL(' 位置', 元旦からの日数 (I), 軸に書かせる日数 (I)) : 日付軸を書かせる

使用例 **ucxacl('b',19770121,300)** :1977 年 1 月 21 日から 300 日分書いた横軸を下端におく。

補足

1.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。
2.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t')、ユーザーによる指定 ('u') の 5 種類がある。' 縦横' との組み合わせに注意がいる。

10 離散的なデータの表示

10 UUMRKZ(引数は補足参照) : 離散的なデータの値の点に印をつける

補足

1. このルーチンの引数は、UUMRKZ(配列の大きさ (I), 横軸座標の配列 (R), 縦軸座標の配列 (R), 印の種類 (I), 印の線の色と太さ (I), 印の大きさ (R))
2. 印の種類、および印の色と太さに関しては最後のページを参照
3. 印の大きさは図を書く領域の最大辺を 1 とする時の比である。

11 連続データの表示

11-1 SG# SET (内部変数、値) : 連続データ表示に関する細かい設定をする。

# (型)	内部変数	内容	初期値
L (論理型)	'LCHAR'	ラベル付き折れ線 (.true.)	.false.
L (論理型)	'LROT'	一定の回転角でラベルを書く (.true.)	線分に沿って書く (.false.)
I (整数型)	'IROT'	ラベルの回転角 (90×I)	0.0
I (整数型)	'INDEXC'	ラベルのラインインデックスの指定	1

11-2 SGSPLC('折れ線の説明'(C)) : 折れ線の説明文字の設定

使用例 **sgsplc('japan')** 折れ線に'japan' の説明をつける。

補足 : SGLSET('LCHAR',.TRUE.) としなければ、このルーチンは動かない。

11-3 SGSPLS(文字の大きさ (R)) : 折れ線につける説明文字の大きさの設定

使用例 **sgspls(0.02)** 文字の大きさを設定する。図の大きさの 2 %にする。(省略するとこの値になる)

補足

1. SGLSET('LCHAR',.TRUE.) としなければ、このルーチンは動かない。
2. 文字の大きさは図を描く領域の最大辺を 1 とする比で定める。

11-4 UULINZ(引数は補足参照) : 折れ線を書かせる。

補足

1. このルーチンの引数は (配列の大きさ (I), 横軸座標の配列 (R), 縦軸座標の配列 (R), 線の種類 (I), 線の色と太さ (I)) である。
2. 線の種類とは 1:実線、2:破線、3:点線、4: 1 点鎖線を表す。
3. 線の色と太さに関しては最後のページを参照

11-5 SGPLZV(引数は UULINZ と同じ) : 軸とは関係無く図での位置を基準にして折れ線を書く。

補足 図での位置とは、図を描く領域の最大辺を 1 、右下を原点とした時の座標で表す。

12 等値線、トーンの表示

12-1 UD# SET(内部変数、値) : 等値線図に関する細かい設定

# (型)	内部変数	内容	初期値
I (整数型)	'INDXMJ'	計曲線の線の色と太さ	3
I (整数型)	'INDXMN'	主曲線の線の色と太さ	1
L (論理型)	'LABEL'	計曲線にラベルを書く (.true.)	.true.
L (論理型)	'LDASH'	負のコンターを別のラインで書く (.true.)	.true.
I (整数型)	'IDASH'	負のコンターの線の色と太さ	3
I (整数型)	'ICYCLE'	計曲線を主曲線に付き I 本の割合で書く	2
I (整数型)	'NLEV'	コンターレベルの自動設定時のコンター数	12
R (実数型)	'RSIZEL'	コンターの文字の大きさ	0.02
L (論理型)	'LMSG'	グラフ下部へのメッセージの表示 (.true.)	.true.
R (実数型)	'XTTL'	グラフ下部へ表示されるメッセージの位置 (左$-1 \leq R \leq 1$右)	0.0

使用例

udiset('indxmj',23) : 計曲線 (大きな目盛の曲線) の色と太さの指定 (赤)

udlset('ldash',.true.),**udiset('idash',33)**:負の値のコンターを別の線で書く。(省略してもそうなる)

udlset('lmsg',.false.) :グラフの下部に書かれるメッセージを省略する。

補足

1. 計曲線とは大きな間隔で等値線を計る線であり、主曲線とは小さな間隔で等値線を計る線である。

2. 文字の大きさは図を描く領域の最大辺を 1 とする時の比である。
3. 線の色と太さに関しては最後のページを参照

12-2 UE# SET(内部変数、値) : トーンに関する細かい設定

# (型)	内部変数	内容	初期値
I (整数型)	'ICYCLE'	計曲線を主曲線に付き I 本の割合で書く	2
I (整数型)	'NLEV'	コンターレベルの自動設定時のコンター数	12

補足 UE# SET は UD# SET と対にして書くとよい。

12-3 UEGTLA(ここから (R), ここまで (R), この間隔 (R)) : トーンを塗り分ける間隔を指定する。
使用例 : **uegtla(100.0,1000.0,50.0)**: 100.0 から 1000.0 の値の間に 50.0 くらいの間隔でトーンを塗り分ける。

12-4 UESTLV(ここから (R), ここまで (R), トーンの色 (I)) : トーンの色を指定する。
使用例 : **uestlv(100.0,150.0,2999)**: 100.0 から 150.0 の値の間に 2999 の色 (赤) のトーンを塗る。
補足 : トーンの色は最後のページ参照。(色番号 (2 桁) + トーン番号 (3 桁))

12-5 UETONE(引数は補足参照) : トーンを書く。
補足 このルーチンの引数は (2 次元配列 (R), 1 次元目の配列の大きさ (I), 1 次元目の配列の大きさ (I), 2 次元目の配列の大きさ (I)) である。

12-6 UDGCLA(ここから (R), ここまで (R), この間隔 (R)) : コンターをひく事を指定する。
使用例 :**udgcla(100.0,1000.0,50.0)**: 100.0 から 1000.0 の値の間に 50.0 くらいの間隔で等値線を書く。

12-7 UDSCLV(引数は補足参照) : 等値線を 1 本だけ描く。
使用例 **udsclv(100.0,23,2,'tropo',0.02)** : 100.0 の値のところに 23(赤) の色と太さで破線であるような等値線を'tropo' なる大きさ 0.05 のラベルを加えてかく。
補足
1. このルーチンの引数は (等値線をひく値 (R), 線の色と太さ (I), 線の種類 (I), ラベル (C), ラベルの文字の大きさ (R)) である。
2. 線の種類は 1:実線、2:破線、3:点線、4: 1 点鎖線で表す。
3. 線の色と太さに関しては最後のページを参照
4. コンターラベルの文字の大きさは図を描く領域を 1 とした時の比である。

12-8 UDCNTR(引数は補足参照) : 等値線を書く。
補足 このルーチンの引数は (2 次元配列 (R), 1 次元目の配列の大きさ (I), 1 次元目の配列の大きさ (I), 2 次元目の配列の大きさ (I)) である。

12-9 UDCNTZ(引数は補足参照) : たくさんのデータに対して等値線を書く。
補足 このルーチンの引数は (2 次元配列 (R), 1 次元目の配列の大きさ (I), 1 次元目の配列の大きさ (I), 2 次

元目の配列の大きさ (I), 作業配列 (I), 作業配列の大きさ (なるべく大きく) (I)) である。

13　ベクトル図の表示

13-1 UG# SET (内部変数、値) : ベクトル図の表示に関する細かい設定

# (型)	内部変数	値	初期値
I (整数型)	'INDEX'	ベクトルの矢の線の色と太さ	1
I (整数型)	'ICENT'	ベクトルを書く位置 (-1; 左:1; 右)	0
L (論理型)	'LMSG'	グラフ下部にメッセージを付す (.true.)	.false).
L (論理型)	'LMISSP'	欠損値ベクトルに印を付す (.true.)	.false.
I (整数型)	'ITYPE1'	欠損値ベクトルに対する印の種類	不定
L (論理型)	'LSMALL'	ベクトルの大きさがある値以下を欠損値扱いする (.true.)	.false.
R (実数型)	'RSMALL'	'LSMALL' を設定した時のそのある値 ！	0.001
I (整数型)	'ITYPE2'	上記設定に於ける印の種類	1
R (実数型)	'RSIZEM'	上記設定に於ける印の大きさ	0.01
L (論理型)	'LUNIT'	単位ベクトルを描く (.true.)	.false.
L (論理型)	'LUMSG'	単位ベクトルのスケールの意味でのタイトルを書く (.true.)	.true.
R (実数型)	'UXUNIT'	単位ベクトルのＸ方向の長さ (普通の量として)	不定
R (実数型)	'UYUNIT'	単位ベクトルのＹ方向の長さ (普通の量として)	不定
R (実数型)	'VXUNIT'	単位ベクトルのＸ方向の長さ	不定
R (実数型)	'VYUNIT'	単位ベクトルのＹ方向の長さ	不定
R (実数型)	'VXULOC'	単位ベクトルの始点のＸ座標	不定
R (実数型)	'VYULOC'	単位ベクトルの始点のＹ座標	不定
R (実数型)	'RSIZEUT'	単位ベクトルにつけるタイトルの文字の大きさ	0.05
I (整数型)	'IUINDX'	単位ベクトルにつけるタイトルの線の色と太さ	3
I (整数型)	'IUTXRO'	単位ベクトルにつけるＸ軸タイトルの回転角	0
I (整数型)	'IUTYRO'	単位ベクトルにつけるＹ軸タイトルの回転角	90

使用例

ugiset('index',23) : ベクトルの矢の線の色と太さを変更する (赤)

uglset('lmsg',.true.) : 図の下にベクトルの長さに関するメッセージを付す。

uglset('lunit',.true.) : 単位ベクトルを描く。

ugiset('iuindx',23) : 単位ベクトルの線の色と太さを変更する (赤)

補足

1. 印の種類、および線の色と太さに関しては最後のページ参照

2.'IUTXRO','IUTYRO' は文字が１文字以外の時は指定できない。

3. 座標の位置の指定については、図を描く領域の最大辺を１とし、左下を原点とするような座標に対して行なう。Ｘ座標、Ｙ座標とは画面に対して横であるか、縦であるかという事にすぎない。

4. ベクトルの長さ、文字の大きさの指定は図を描く領域の最大辺を１としたときの比で表す。

13-2 UGSUT('ベクトルの位置'(C),'タイトル'(C)) : 単位ベクトルにつけるタイトルの設定

使用例

ugsut('x','x-axis') :横方向の単位ベクトルに対して'x-axis' というタイトルをつける。

ugsut('y','y-axis') :縦方向の単位ベクトルに対して'y-axis' というタイトルをつける。

13-4 UGVECT(引数は補足参照) : ベクトル図を書く。

補足 : このルーチンの引数は (X成分を表す配列 (R), 配列の 1 次元目の大きさ (I), Y 成分を表す配列 (R), 配列の 1 次元目の大きさ (R), 配列の 1 次元目の大きさ (R), 配列の 2 次元目の大きさ (R)) である。

14 等値線、トーン、ベクトル図表示のあとに

14-1 UDICLV : コンターを無効にする。

補足 : 等値線を書くルーチン (UDCNTR など) をループの中に入れた場合に必ず呼ぶ必要がある。

14-2 UDDCLV(消去するコンターの値 (R)) : あるコンターを無効にする。

14-3 UEITLV : トーンを無効にする。

補足 : トーンを描くルーチン (UETONE など) をループの中に入れた場合に必ず呼ぶ必要がある。

15 文字の表示

15-1 SG# SET (内部変数、値) : 文字の表示に関する細かい設定

# (型)	内部変数	値	初期値
I (整数型)	'IFONT'	フォント番号 (1:丸文字,2:普通)	1
L (論理型)	'LCNTL'	制御文字の有効化 (.true.)	.true.
R (実数型)	'SMALL'	上付け (下付け) 文字の文字の大きさ	0.7(文字の大きさからの比)
R (実数型)	'SHIFT'	上付け (下付け) 文字をはなす距離	0.3(文字の大きさからの比)
I (整数型)	'ISUP'	「¯」(上付け文字のはじめに置く制御文字) の番号	124
I (整数型)	'ISUB'	「_」(下付け文字のはじめに置く制御文字) の番号	95
I (整数型)	'IRST'	「゜」(制御文字の終了を示す制御文字) の番号	34

使用例

sglset('lcntl',.false.) : 制御文字を無効にする。

sgrset('small',0.3) : 上付け (下付け) 文字の文字の大きさを普通の文字の 0.3 倍にする。(おすすめ !)

sgrset('shift',0.1) : 上付け (下付け) 文字をはなす距離を 0.1 文字分とする。(おすすめ !)

補足 : 'ISUP"ISUB"IRST' を表す制御文字を用いて上付け文字、下付け文字を書ける。'LCNTR' で制御文字を有効化すると制御文字そのものを書く事ができない。

15-2 CSGI(文字番号 (I)):ＤＣＬ文字函数 :ＤＣＬ文字函数の番号に相当するものを返す。最後のページ参照

補足 :CHARACTER CSGI*1 の宣言をする必要がある。

15-3 SGTXZU (引数は補足参照) : 指定した座標の位置に文字を書く。

使用例 **sgtxzu(0.1,0.1,'stclim',0.02,0,0,23)** : (0.1,0.1) のところに'stclim' の文字の中心ががその座標に一致するように 23 (赤) の色と太さでかく。

補足

1. このルーチンの引数は (横軸座標 (R), 縦軸座標 (R), 書くべき文字 (C), 文字の大きさ (R) 文字を傾ける角度 (I), 文字の位置 (I), 文字の線の色と太さ (I)) である。
2. 文字の大きさは図を書く領域の最大辺を 1 とする時の比で表す。
3. 文字を傾ける角度は度数法で表す。反時計回りが正である。
4. 文字の位置は、左 (-1)、中 (0)、右 (1) となる。
5. 文字の線の色と太さは最後のページを参照

6.SGTXZV(引数は SGTXZU とおなじ) で図を書く領域の最大辺を 1 とし左下を原点とするような座標で描く事ができる。

16 矢印の表示

16-1 SG# SET (内部変数、値) : 矢印の表示に関する細かい設定

# (型)	内部変数	値	初期値
L (論理型)	'LPROP'	矢印を線の値に比例させる (.true.)	.true.
L (論理型)	'LVARW'	矢印を地図にへばらせる。(.true.)	.true.

補足 : 地図上にベクトル図を書かせる場合にはこのルーチンを利用する。

16-2 SGLAZU(引数は補足参照) : 指定した座標の位置に矢印を書く。

使用例 **sglazu(0.3,0.2,0.31,0.21,1,21)** : (0.3,0.2) から (0.31,0.21) へ矢印を 1 (実線) で 21 (赤) のような線で書く。

補足

1. このルーチンの引数は (始点の横軸座標 (R), 始点の縦軸座標 (R), 終点の横軸座標 (R), 終点の縦軸座標 (R), 矢印の線の種類 (I), 矢印の線の色と太さ (I)) である。
2. 矢印の線の種類は 1:実線、2:破線、3:点線、4: 1 点鎖線を表す。
3. 矢印の線の色と太さは最後のページ参照
4. 地図にベクトルを書き入れたい時は、始点と終点の (経度, 緯度) をとればよい。

17 完全簡単ルーチン

 # 0 USGRPH(配列の大きさ (I), 横軸座標を与える配列 (R), 縦軸座標を与える配列 (R))

: 函数 $y = f(x)$ を折れ線で書かせるもっとも単純なルーチン

 # 1-1 USSPNT(配列の大きさ (I), 横軸座標を与える配列 (R), 縦軸座標を与える配列 (R))

: 作画の範囲指定

 # 1-2 USPFIT : 切りのよい物理量の範囲の設定

 # 1-3 USCSET (内部変数、値) :

# (型)	内部変数	値	初期値
C (文字型)	'CXTTL'	横軸のタイトル	不定
C (文字型)	'CYTTL'	縦軸のタイトル	不定
C (文字型)	'C'//' 縦横'//'TTL'	' 位置' 軸の位置	不定

補足

1.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。

2.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t') をあらわす。

 # 1-4 USSTTL (' 横軸タイトル',' 横軸ラベル',' 縦軸タイトル',' 縦軸ラベル') : タイトルを書かせる。

 # 1-5 US' 縦横'AXS (' 位置') : 座標軸を 1 つ書かせる。

補足

1.' 縦横' とは指定が縦軸に関するもの ('y') か、横軸に関するもの ('x') かをさす。

2.' 位置' とは指定する軸の位置を定めるもので、左 ('l')、右 ('r')、下 ('b')、上 ('t') をあらわす。

 # 1-6 USDAXS : 座標軸を勝手に描かせる。

18 グラフィックの終了

0 GRCLS : グラフィックを終了する。