地球流体電脳ライブラリ

GRPH2

(図形処理上位パッケージ)

地球流体電脳倶楽部

1995 年 6 月 9 日 (DCL-5.0)

目 次

1 概要 **1**
1.1 はじめに . . . 1
1.2 オートスケーリングルーチン . . . 3

2 GRPACK：上位コントロールルーチン **5**
2.1 概要 . . . 5
2.2 サブルーチンのリスト . . . 5
2.3 サブルーチンの説明：コントロール . . . 6
2.3.1 GROPN . . . 6
2.3.2 GRFRM . . . 6
2.3.3 GRFIG . . . 7
2.3.4 GRCLS . . . 7
2.4 サブルーチンの説明：正規化変換 . . . 7
2.4.1 GRSVPT . . . 8
2.4.2 GRSWND/GRSSIM/GRSMPL . . . 8
2.4.3 GRSTRN . . . 8
2.4.4 GRSTRF . . . 9

3 U[XYZ]PACK：直交座標軸ルーチン **10**
3.1 概要 . . . 10
3.2 座標軸の場所 . . . 11
3.3 座標軸の構成要素とその属性 . . . 11
3.4 LLA における属性の指定方法 . . . 12
3.5 サブルーチンのリスト . . . 15
3.6 サブルーチンの説明：上位応用ルーチン (ULA) . . . 16
3.6.1 UXAXDV/UYAXDV . . . 17
3.6.2 UXAXNM/UYAXNM . . . 17
3.6.3 UXAXLB/UYAXLB . . . 18
3.6.4 UXMTTL/UYMTTL . . . 19
3.6.5 UXSTTL/UYSTTL . . . 19
3.7 サブルーチンの説明：UZPACK . . . 20
3.7.1 UZ*p*GET/UZ*p*SET(UZ*p*STX) . . . 20
3.7.2 UZINIT . . . 22
3.7.3 UZFACT . . . 22

3.7.4 UZPSAV . . . 23
3.7.5 UZPRST . . . 23
3.8 サブルーチンの説明 : 下位応用ルーチン (LLA) . . . 24
3.8.1 UXPAXS/UYPAXS . . . 24
3.8.2 UXPTMK/UYPTMK . . . 25
3.8.3 UXPLBL/UYPLBL . . . 25
3.8.4 UXPNUM/UYPNUM . . . 26
3.8.5 UXPTTL/UYPTTL . . . 27
3.9 その他のサブルーチン ・関数のリスト . . . 27

4 ULPACK : 対数座標軸ルーチン 28
4.1 概要 . . . 28
4.2 サブルーチンのリスト . . . 29
4.3 サブルーチンの説明 . . . 29
4.3.1 ULXLOG/ULYLOG . . . 29
4.3.2 ULXSFM/ULYSFM . . . 30
4.3.3 ULSXBL/ULSYBL . . . 30
4.3.4 UL*p*GET/UL*p*SET(UL*p*STX) . . . 31

5 UCPACK : 日付に関する座標軸ルーチン 33
5.1 概要 . . . 33
5.2 サブルーチンのリスト . . . 34
5.3 サブルーチンの説明 . . . 34
5.3.1 UCXACL/UCYACL . . . 34
5.3.2 UCXADY/UCYADY . . . 34
5.3.3 UCXAMN/UCYAMN . . . 35
5.3.4 UCXAYR/UCYAYR . . . 36
5.3.5 UC*p*GET/UC*p*SET(UC*p*STX) . . . 36
5.3.6 その他の関数 . . . 37

6 U[UVH]PACK : 1 次元グラフルーチン 38
6.1 概要 . . . 38
6.2 未定義値 (RUNDEF) の解釈 . . . 39
6.3 サブルーチンのリスト . . . 40
6.4 サブルーチンの説明: 折れ線 . . . 42
6.4.1 UULIN/UULINZ . . . 42
6.5 サブルーチンの説明: マーカー . . . 42
6.5.1 UUMRK/UUMRKZ . . . 42
6.6 サブルーチンの説明: エラーバー . . . 43
6.6.1 UVERB/UVERBZ/UHERB/UHERBZ . . . 43
6.7 サブルーチンの説明: 差分 . . . 43
6.7.1 UVDIF/UVDIFZ/UHDIF/UHDIFZ . . . 43
6.8 サブルーチンの説明: 棒グラフ . . . 44
6.8.1 UVBRF/UVBRFZ/UHBRF/UHBRFZ . . . 44

6.8.2 UVBRA/UVBRAZ/UHBRA/UHBRAZ . . . 45
6.8.3 UVBRL/UVBRLZ/UHBRL/UHBRLZ . . . 45
6.9 サブルーチンの説明: 箱グラフ . . . 46
6.9.1 UVBXF/UVBXFZ/UHBXF/UHBXFZ . . . 46
6.9.2 UVBXA/UVBXAZ/UHBXA/UHBXAZ . . . 47
6.9.3 UVBXL/UVBXLZ/UHBXL/UHBXLZ . . . 47
6.10 サブルーチンの説明: パラメタ設定 . . . 48
6.10.1 UUSLNT/UUSLNI . . . 48
6.10.2 UUSMKT/UUSMKI/UUSMKS . . . 48
6.10.3 UUSERT/UUERI/UUSERS . . . 49
6.10.4 UUSBRS . . . 49
6.10.5 UUSARP . . . 49
6.10.6 UUSFRT/UUSFRI . . . 50
6.10.7 UUSIDV . . . 50

7 USPACK ：オートスケーリングルーチン 51
7.1 概要 . . . 51
7.2 基本構造 . . . 51
7.2.1 簡単描画ルーチン (USGRPH) . . . 51
7.2.2 スケーリング ・ルーチン . . . 52
7.2.3 座標軸ルーチン . . . 52
7.2.4 内部変数管理 . . . 53
7.3 USPACK による作画範囲 . . . 54
7.4 サブルーチンのリスト . . . 55
7.5 サブルーチンの説明 (簡単ルーチン) . . . 56
7.5.1 USGRPH . . . 56
7.6 サブルーチンの説明 (スケーリング ・ルーチン) . . . 57
7.6.1 USSPNT . . . 57
7.6.2 USPFIT . . . 57
7.7 サブルーチンの説明 (座標軸ルーチン) . . . 58
7.7.1 USDAXS . . . 58
7.7.2 USXAXS/USYAXS . . . 58
7.7.3 USSTTL . . . 58
7.8 サブルーチンの説明 (パラメータ管理) . . . 60
7.8.1 US*p*GET/US*p*SET(US*p*STX) . . . 60
7.8.2 USCGET/USCSET(USCSTX) . . . 62
7.8.3 USINIT . . . 63

8 UDPACK：等高線ルーチン 64
8.1 概要 . . . 64
8.2 サブルーチンの説明 . . . 66
8.2.1 UDCNTR . . . 66
8.2.2 UDCNTZ . . . 67
8.2.3 UDGCLA . . . 67

8.2.4 UDGCLB . . . 68
8.2.5 UDSCLV . . . 69
8.2.6 UDDCLV . . . 69
8.2.7 UDICLV . . . 70
8.2.8 RUDLEV . . . 70
8.2.9 UDSFMT . . . 70
8.2.10 UD*p*GET/UD*p*SET(UD*p*STX) . . . 71
8.2.11 その他のサブルーチン ・関数 . . . 72

9 UEPACK : トーン塗りつぶしルーチン 73
9.1 概要 . . . 73
9.2 サブルーチンの説明 . . . 75
9.2.1 UETONE . . . 75
9.2.2 UETONF . . . 76
9.2.3 UEGTLA . . . 76
9.2.4 UEGTLB . . . 77
9.2.5 UESTLV . . . 77
9.2.6 UESTLN . . . 78
9.2.7 UEITLV . . . 78
9.2.8 UE*p*GET/UE*p*SET(UE*p*STX) . . . 79
9.2.9 その他のサブルーチン . . . 80

10 UGPACK : ベクトル場表示ルーチン 81
10.1 概要 . . . 81
10.2 サブルーチンの説明 . . . 83
10.2.1 UGVECT . . . 83
10.2.2 UGSUT . . . 84
10.2.3 UG*p*GET/UG*p*SET(UG*p*STX) . . . 85
10.2.4 その他のサブルーチン ・関数 . . . 87

11 UWPACK : 格子点情報の管理ルーチン 88
11.1 概要 . . . 88
11.2 サブルーチンの説明 . . . 89
11.2.1 UWSGXA/UWSGYA . . . 89
11.2.2 UWSGXB/UWSGYB . . . 89
11.2.3 UWQGXZ/UWQGYZ . . . 90
11.2.4 RUWGX/RUWGY . . . 90
11.2.5 IUWGX/IUWGY . . . 91
11.2.6 UWQGXI/UWQGYI . . . 91
11.2.7 UWDFLT . . . 91

12 UMPACK : 地図投影ルーチン 93
12.1 概要 . . . 93
12.1.1 はじめに . . . 93
12.2 サブルーチンの説明:コントロール . . . 96

12.2.1 UMSCNT 96
12.2.2 UMSCWD 96
12.2.3 UMSPNT 97
12.2.4 UMPFIT 97
12.3 サブルーチンの説明:地図情報の描画 97
12.3.1 UMPGLB 97
12.3.2 UMPGRD 98
12.3.3 UMPLIM 98
12.3.4 UMPMAP 99
12.3.5 UM*p*GET/UM*p*SET(UM*p*STX) 99

13 UBPACK : ラベル付折線ルーチン **101**
13.1 概要 101

14 UFPACK : 矢印付線分ルーチン **102**
14.1 概要 102

謝辞 **103**

第 1 章

概要

1.1 はじめに

GRPH2 図形処理上位パッケージは, GRPH1 の折れ線や文字といった単純な図形要素を組み合せて, 座標軸や等高線図などを描くルーチンを集めたものである.

GRPH2 はレベル 2 のライブラリであり機種依存はしないが, GRPH1, MISC1, MATH1 が正しくインストールされている必要がある. (厳密には, UMPACK が使う地図情報データの収められたファイル名の決定方法はシステムに依存する可能性がある.) 座標系や座標変換, 各種出力プリミティブとその属性などに関する基本用語は 『GRPH1」マニュアルを参照のこと.

GRPH2 は描くべき図の種類によって, いくつかのサブパッケージにわかれている. これらのサブパッケージの数は徐々に増えているが, 現在のところ, 1 次元スカラー量のグラフ化, 2 次元スカラー量の表示, 2 次元ベクトル量の表示など, 以下のようなサブパッケージからなる.

- **GRPACK**

  改ページなどの制御とパラメタの初期化などをする.

- **UXPACK, UYPACK, UZPACK, ULPACK, UCPACK**

  座標軸を描画する. それぞれのサブパッケージの機能は次の通り.

  - **UXPACK/UYPACK :** 直交座標系の座標軸
  - **ULPACK :** 対数軸
  - **UCPACK :** 日付に関する座標軸
  - **UZPACK :** 座標軸のパラメータ管理

- **UUPACK/UVPACK/UHPACK**

  折れ線, マーカー, 棒グラフなど.

  - **UUPACK :** X 軸と Y 軸が対等なグラフ, およびパラメタ設定.
  - **UVPACK :** X 軸が独立変数で, Y 軸が従属変数になるようなグラフ.
  - **UHPACK :** Y が独立変数で, X 軸が従属変数になるようなグラフ.

- **USPACK**

オートスケーリングおよび簡単座標軸ルーチン.

- **UDPACK, UEPACK, UGPACK, UWPACK**

  2 次元量の図形描画. それぞれのサブパッケージの機能は次の通り.

  - **UDPACK :** 2 次元コンター
  - **UEPACK :** 2 次元トーン
  - **UGPACK :** 2 次元ベクトル場
  - **UWPACK :** 格子座標点データの管理

  このうち UDPACK, UEPACK は U 座標系で作画するので, GRPH1 がサポートするすべての正規化変換に対応する. いっぽう UGPACK は, 現在のバージョンでは V 座標系で作画するので, 直角直線座標系 (ITR= $1 \sim 4$) でしか利用できない.

- **UMPACK**

  地図情報 (海岸線や緯度経度線) の描画.

- **UFPACK, UBPACK**

  図形要素描画. それぞれのサブパッケージの機能は次の通り. なお dcl-ver.5 では, 以下の機能を持つサブルーチン群を GRPH1 レベルに取り込んだので, これらは将来的に消去される.

  - **UFPACK :** 矢印
  - **UBPACK :** ラベル付折れ線

GRPH2 の多くのパッケージは独自の内部変数管理ルーチンを xx$p$GET/xxpSET という名前で持っている. ここで xx は各パッケージ名の最初の 2 文字, p は, 変数のタイプ別に R:実数; I:整数; L:論理変数; C: 文字変数 を表わす. このマニュアルでは以後, タイプ別のエントリ名を総称して xx$p$GET/xxpSET などと書く.

1.2 オートスケーリングルーチン

GRPH2 のパッケージには USPACK, UMPACK などのように, 変換関数のパラメタ (ウィンドウなど) を自動的に決めるパッケージがいくつかある. これらのパッケージを使うと, 細かなパラメタを気にすることなく, 与えられたデータをグラフ化することができて便利であるが, システムが決めるパラメータが全ての目的に最適である訳ではない. パラメタの一部をユーザーが指定することで, 最終的な図の品質が各段に向上することも少なくない.

このように 「一部のパラメタをユーザーが指定する」ことを可能にするためには, システムが 「ユーザーがどのパラメタを陽に指定したか」ということを知る必要がある. そのために GRPH2 では, 欠損値と同様に 「未定義値」を導入する. この 「未定義値」は GLpGET/GLpSET によって管理される 'RUNDEF', 'IUNDEF' である.

具体的な手続きとしては,

1: 新しい図を描く前に自動決定可能なパラメタ全てに 「未定義値」を代入する.

2: オートスケーリングルーチンは, 「未定義値」になっているパラメタのみを自動的に決定する.

1 の 「未定義値」の代入は GRPACK の GRFRM, GRFIG を呼ぶことで行なわれる. したがて, ユーザーは SGRFM の代りに GRFRM を呼び, オートスケーリングルーチンを呼ぶ前に, GRSWND/SGSWND などによってパラメタを与えることで一部のパラメタを陽に指定することができる. GRSWND のように複数のパラメタを同時に指定するルーチンで, その中の一部のパラメタだけ陽に指定したい場合には, その他のパラメタに 'RUNDEF' のあらわす値を代入しておけばよい. オートスケーリングルーチンは, 指定されているパラメタによって若干異なる動作をする場合がある.

例えば, 全てをユーザーが指定する (オートスケーリングルーチンを使わない) 場合には,

```
1         CALL GROPN(IWS)
2         CALL GRFRM
3
4         CALL GRSTRN(ITR)
5         CALL GRSVPT(VXMIN, VXMAX, VYMIN, VYMAX)
6         CALL GRSWND(UXMIN, UXMAX, UYMIN, UYMAX)
7
8         CALL GRSTRF
9         ........
```

となるが, 描画したい点の情報から USPACK に適当なパラメタを決めさせたい場合は,

```
1         CALL GROPN(IWS)
2         CALL GRFRM
3
4         CALL GRSTRN(ITR)
5         CALL USSPNT(N, UPX, UPY)
6         CALL USPFIT
7
8         CALL GRSTRF
9         ........
10        CALL SGPLU(N, UPX, UPY)
11        ........
```

となる. つまり, GRSWND, GRSVPT などの代りに, これらを決定するための情報を USSPNT によって与える. 必要なパラメタは USPFIT が呼ばれることによって決定され, GRSTRF を呼ばれることによって変換関数が確定する. 一部のパラメタを陽に指定したい場合は, GRFRM と GRSTRF の間で GRSWND や GRSVPT などを使って指定すればよい. 例えば,

```
1          CALL GROPN(IWS)
2          CALL GRFRM
3
4          CALL GRSTRN(ITR)
5          CALL GRSWND(UXMIN, UXMAX, RUNDEF, RUNDEF)
6          CALL USSPNT(N, UPX, UPY)
7          CALL USPFIT
8
9          CALL GRSTRF
10         ........
11         CALL SGPLU(N, UPX, UPY)
12         ........
```

とすれば, Y 軸の最大最小値のみ自動的に決めることができる.

第 2 章

GRPACK : 上位コントロールルーチン

2.1 概要

GRPACK は GRPH2 の各種パッケージを使って新しい図を描くとき, これらのパッケージを初期化したり, 変換関数に関するパラメータを (実行時オプションの介入を許して) 設定するものである. 現在のところ SGPACK, UZPACK, UUPACK, USPACK, UMPACK の初期化をおこなっている.

GRPACK は基本的に GRPH1 の 「コントロール」および 「正規化変換」に関するルーチンに相当するものを用意している. たとえば GRPACK には, SGFRM に相当する GRFRM というルーチンがある. これは内部的に SGFRM などを呼ぶと同時に, 各種パッケージの初期化をおこなう. 初期化動作としては, GRPH2 でのサブパッケージが自動的に決める可能性のあるパラメタに, 「未定義」の値を代入したりする. (具体的には, GL*p*GET/GL*p*SET が管理する内部変数'RUNDEF', 'IUNDEF' が示す値を代入する.) また SGSTRF に相当する GRSTRF というルーチンがある. これは内部的には, 正規化変換に必要なパラメータがすべて設定されている (RUNDEF でない) ことをチェックした上で SGSTRF を呼ぶ.

さらに GRPACK で設定するパラメータについては, 実行時オプションの介入がおこなわれる. たとえば SGOPN に相当する GROPN というルーチンを使うと, 指定するワークステーション番号は実行時オプションで変更できる.

2.2 サブルーチンのリスト

以下のリストでは, 対応する GRPH1/SGPACK のルーチン名を括弧内に示す.

コントロール

GROPN (SGOPN)	出力装置のオープン.
GRFRM (SGFRM)	フレームの設定.
GRFIG	新たな図の設定.
GRCLS (SGCLS)	出力装置のクローズ.

正規化変換

GRSVPT (SGSVPT) ビューポートの設定.
GRSWND (SGSWND) ウインドウの設定.
GRSSIM (SGSSIM) 相似変換の設定.
GRSMPL (SGSMPL) 地図投影の極の設定.
GRSTRN (SGSTRN) 変換関数の番号による設定.
GRSTRF (SGSTRF) 変換関数の確定.

2.3 サブルーチンの説明 : コントロール

以下のルーチンは, 基本的には対応する GRPH1/SGPACK のコントロールに関するルーチン (GR を SG にかえたもの) と同等な動作をする (GRFIG のみ対応するルーチンはない). 詳しくは 『GRPH1」マニュアルを参照のこと.

2.3.1 GROPN

1. 機能

   図形出力装置をオープンする (初期化処理をおこなう).

2. 呼び出し方法

```
CALL GROPN(IWS)
```

3. パラメーターの説明

   IWS (I) ワークステーション番号.

4. 備考

   (a) GROPN は 内部変数 'IWS' を SGISTX で設定して SGINIT を実行する. したがって SGpGET/SGpSET の管理する内部変数 'IWS' は実行時パラメータによる介入が可能である. (以下, 単に 「内部変数」というのは SGpGET/SGpSET の管理する内部変数のことを指す.)

   (b) また, GROPN は SGpSTX によって内部変数'LCNTL' を .TRUE. にする.

2.3.2 GRFRM

1. 機能

   新しい作画領域を設定するとともに, 新しい図を描くために必要な初期化をする.

2. 呼び出し方法

```
CALL GRFRM
```

3. パラメーターの説明

   なし.

4. 備考

   (a) このルーチンは変換関数に関する内部変数をすべて 「未定義」にするとともに, UZPACK, USPACK の初期化をおこなったのち, SGFRM を呼ぶ.

2.3.3 GRFIG

1. 機能

   新しい図を描くために必要な初期化をする.

2. 呼び出し方法

```
CALL GRFIG
```

3. パラメーターの説明

   なし.

4. 備考

   (a) このルーチンは変換関数に関する内部変数をすべて 「未定義」にするとともに, UZPACK, USPACK の初期化をおこなう. 一つのフレーム内に複数の図を描くとき, 2 つ目以降の図を描き始める前に, そのつど, このルーチンを呼ばなければならない.

2.3.4 GRCLS

1. 機能

   図形出力装置をクローズする (終了処理をおこなう).

2. 呼び出し方法

```
CALL GRCLS
```

3. パラメーターの説明

   なし.

4. 備考

   (a) 内部で SGCLS を呼んでいる.

2.4 サブルーチンの説明： 正規化変換

以下のルーチンは, 基本的には対応する GRPH1/SGPACK の正規化変換に関するルーチン (GR を SG にかえたもの) と同等な動作をする. 詳しくは 「GRPH1」マニュアル参照.

以下にあらわれるパラメータは, すべて SGpSTX を用いて設定される. したがって, 実行時オプションによる介入が可能である.

2.4.1 GRSVPT

1. 機能

   正規化変換のためのビューポートを設定する.

2. 呼び出し方法

```
CALL GRSVPT(VXMIN,VXMAX,VYMIN,VYMAX)
```

3. パラメーターの説明

VXMIN,VXMAX,VYMIN,VYMAX	(R)	ビューポート.

4. 備考

   (a) なし.

2.4.2 GRSWND/GRSSIM/GRSMPL

1. 機能

   正規化変換のためのパラメータを設定する.

2. 呼び出し方法

```
CALL GRSWND(UXMIN,UXMAX,UYMIN,UYMAX)
CALL GRSSIM(SIMFAC,VXOFF,VYOFF)
CALL GRSMPL(PLX,PLY,PLROT)
```

3. パラメーターの説明

UXMIN,UXMAX,UYMIN,UYMAX	(R)	ウインドウ.
SIMFAC	(R)	相似変換のスケーリングファクター.
VXOFF,VYOFF	(R)	原点のオフセット.
PLX,PLY,PLROT	(R)	地図投影の際の回転角.

4. 備考

   (a) なし.

2.4.3 GRSTRN

1. 機能

   正規化変換の変換関数番号を指定する.

2. 呼び出し方法

```
CALL GRSTRN(ITR)
```

3. パラメーターの説明

ITR	(I)	変換関数番号.

4. 備考

   (a) なし.

2.4.4 GRSTRF

1. 機能

   正規化変換を確定させる.

2. 呼び出し方法

   ```
   CALL GRSTRF
   ```

3. パラメーターの説明

   なし.

4. 備考

   なし.

第 3 章

U[XYZ]PACK : 直交座標軸ルーチン

3.1 概要

UXPACK/UYPACK/UZPACK は直交座標系の座標軸をプロットするためのサブルーチンパッケージである. したがって, 変換関数番号としては ITR= 1 ∼ 4(直角直線座標系) の正規化変換に対してのみ使用できる. それ以外の変換関数番号を指定しても動作は保証されない.

このパッケージにもとづいて ULPACK (第 4章), UCPACK (第 5章), USPACK (第 7章) などの座標軸関連パッケージは構築されている.

UXPACK/UYPACK/UZPACK それぞれのパッケージの機能は次のようになっている.

UXPACK : X 軸をプロットする
UYPACK : Y 軸をプロットする
UZPACK : 内部変数の管理や初期化をおこなう

UXPACK, UYPACK はそれぞれ次のような 3 レベルのサブルーチン群からなる.

1. 基本ルーチン群 (Basic Routines) : 基本的な作画をおこなうサブルーチン. ユーザーは知る必要がない.
2. 下位応用ルーチン群 (Lower Level Applications) : 基本ルーチン群を使って書かれていて, 具体的な作画をおこなうルーチン群. (以下では LLA と略記する)
3. 上位応用ルーチン群 (Upper Level Applications) : LLA を組み合わせて書かれていて, 実際にユーザーが使用するルーチン群. (以下では ULA と略記する)

このほかに座標軸のオフセットあるいは座標系のリニアオフセットをコントロールするサブルーチン群がある. 簡単な座標軸の作画はほとんど ULA で実現できる. ULA で満足のゆく作画ができない場合, LLA を組み合わせればさらに高度な目的に応じた ULA が作成できるはずである. 以下の 「サブルーチンの説明」では, 上位応用ルーチン (ULA), UZPACK, 下位応用ルーチン (LLA) の順に説明する.

簡単な座標軸の作画をしようとするユーザーは, 「座標軸の場所」に関する次節の説明を読んだあと 3.6節の 「上位応用ルーチンの説明」に進んでもらえれば, おそらく期待に沿うような座標軸の描画ができるはずである.

3.2 座標軸の場所

このパッケージを使うためには, あらかじめ変換関数番号が ITR= 1 ∼ 4 (直角直線座標系) の正規化変換が設定されていなければならない (「GRPH1」のマニュアル参照). たとえば

```
CALL SGSVPT(VXMIN,VXMAX,VYMIN,VYMAX)
CALL SGSWND(UXMIN,UXMAX,UYMIN,UYMAX)
CALL SGSTRN(ITR)
CALL SGSTRF
```

が呼んであるとする. このとき LLA, ULA は次の場所に座標軸を設定できる.

UXPACK については

'B'ottom – (UXMIN, UYMIN) と (UXMAX, UYMIN) を結ぶ線
'T'op – (UXMIN, UYMAX) と (UXMAX, UYMAX) を結ぶ線
'U'ser – (UXMIN, UYUSER) と (UXMAX, UYUSER) を結ぶ線

UYPACK については

'L'eft – (UXMIN, UYMIN) と (UXMIN, UYMAX) を結ぶ線
'R'ight – (UXMAX, UYMIN) と (UXMAX, UYMAX) を結ぶ線
'U'ser – (UXUSER, UYMIN) と (UXUSER, UYMAX) を結ぶ線

X 軸に関する'B', 'T', 'U', Y 軸に関する'L', 'R', 'U' は, 座標軸の構成要素 (次節参照) を作画する場所を示すためのパラメーター (CSIDE) として用いられる. UXUSER, UYUSER は内部変数'UXUSER', 'UYUSER' を設定して初めて有効となる (初期値は不定). 以下, 特に断らないかぎり単に 「内部変数」というのは UZpGET/UZpSET(3.7.1節参照) の管理する内部変数のことを指す. 座標軸を設定できるこれらの場所は, UXPACK については UXSAXZ または UXSAXS によって Y 方向に平行移動することができる; UYPACK については UYSAXZ または UYSAXS によって X 方向に平行移動することができる (3.8.1節参照).

3.3 座標軸の構成要素とその属性

LLA が作画する座標軸の構成要素には次のものがある (カッコ内はその構成要素を描くサブルーチン名).

1. 軸を示す線分 (UXPAXS/UYPAXS)
2. 目盛 (UXPTMK/UYPTMK)
3. ラベル
   - 文字列で指定 (UXPLBL/UYPLBL)
   - 数値で指定 (UXPNUM/UYPNUM)
4. タイトル (UXPTTL/UYPTTL)

それぞれの構成要素は次のような属性を持っている.

1. 軸を示す線分については
   a. 線分のラインインデクス <ISLCT>

2. 目盛については
   a. 目盛の長さ <ISLCT>
   b. 目盛を描く線のラインインデクス <ISLCT>
3. ラベルについては
   a. 文字の高さ <ISLCT>
   b. 文字列の回転角 <CSIDE>
   c. 文字列のセンタリングオプシ ョン <CSIDE>
   d. 文字列を描く線のラインインデクス <ISLCT>
4. タイトルについては
   a. 文字の高さ <ISLCT>
   b. 文字列の回転角 <CSIDE>
   c. 文字列のセンタリングオプシ ョン <引き数>
   d. 文字列を描く線のラインインデクス <ISLCT>

これらの属性の指定方法および<ISLCT>, <CSIDE>, <引き数>の意味については次節で述べる.

座標軸の設定を'U'ser 以外でおこなった場合, ラベルとタイトルは常にウインドウの外側に向かって描かれる. 'U'ser で設定をおこなった場合, ラベルとタイトルを軸のどちら側に書くかは内部変数'IFLAG' が決める (3.8.3節参照).

3.4 LLA における属性の指定方法

LLA の各サブルーチンには ISLCT, CSIDE という 2 つの引き数がある. この 2 つの引き数およびその他の引き数が各構成要素の属性を決める.

1. ISLCT は 0,1,2 の値をとりうる. 3.3節で述べた属性のうち<ISLCT>とあるものは, この引き数によって定まる内部変数を参照して決まる. ISLCT=0,1,2 の具体的な意味は,

   ISLCT=0 : ユーザーが定義した内部変数を使って作画する.

   ISLCT=1 : このパッケージがあらかじめ設定している内部変数を使って作画する. 小さめの目盛, ラベル, タイトルを描くときに用いる.

   ISLCT=2 : このパッケージがあらかじめ設定している内部変数を使って作画する. 大きめの目盛, ラベル, タイトルを描くときに用いる.

   たとえば, 軸を示す線分のラインインデクスについては, ISLCT=0,1,2 それぞれについて'INDEXT0', 'INDEXT1', 'INDEXT2' という内部変数があって (初期値は順に, 不定, 1, 3) これを参照して属性が決まる. 以下では以上の表現をまとめて次のように略記する. — 「内部変数'INDEXTi' で指定してある値を参照してラインインデクスの属性が決まる (初期値は, 不定, 1, 3)」. これらの内部変数は UZpGET/UZpSET によって参照/変更できる (3.7.1節参照).

2. CSIDE は構成要素を描く場所を指定するとともに, 描く場所に依存する属性を決める. 3.3節で述べた属性のうち<CSIDE>とあるものは, この引き数によって定まる内部変数を参照して決まる. たとえば, X 軸につけるタイトルの回転角については, CSIDE='B','T','U' それぞれについて'IROTCXB', 'IROTCXT', 'IROTCXU' という内部変数があって (初期値は順に, 0, 0, 0) これを参照して属性が決ま

る. 以下では以上の表現をまとめて次のように略記する. — 「内部変数'IROTCXs' で指定してある値を参照して, X 軸につけるタイトルの回転角に関する属性が決まる (初期値は, 0, 0, 0)」. また, Y 軸につけるタイトルの回転角については, CSIDE='L','R','U' それぞれについて'IROTCYL', 'IROTCYR', 'IROTCYU' という内部変数があって (初期値は順に, 1, 1, 1) これを参照して属性が決まる. 以下では以上の表現をまとめて次のように略記する. — 「内部変数'IROTCYs' で指定してある値を参照して, Y 軸につけるタイトルの回転角に関する属性が決まる (初期値は, 1, 1, 1)」. さらに'IROTCXs' と 'IROTCYs' をまとめて次のように略記する. — 「内部変数'IROTCzs' で指定してある値を参照して, 軸につけるラベルの回転角に関する属性が決まる (初期値は, 0, 0, 0, 1, 1, 1)」.

3. 3.3節で述べた属性のうち<引き数>とあるものは, 引き数として直接指定することによって決まる.

3.3節で述べた属性は具体的に次のように決まる.

1. 軸を示す線分については

   a. 線分のラインインデクス <ISLCT>
   内部変数'INDEXTi' で指定してあるラインインデクスを用いて線分を描く (初期値は, 不定, 1, 3).

2. 目盛については

   a. 目盛の長さ <ISLCT>
   内部変数'RSIZETi' で指定してある長さ (単位は V 座標系) の目盛を描く (初期値は, 不定, 0.007, 0.014).

   b. 目盛を描く線のラインインデクス <ISLCT>
   内部変数'INDEXTi' で指定してあるラインインデクスを用いて目盛を描く (初期値は, 不定, 1, 3). (注:目盛を描く向きは内部変数'INNER' が決める. 'INNER' が正のとき内側に向かって目盛を描く; 負のとき外側に向かって目盛を描く (初期値は+1).)

3. ラベルについては

   a. 文字の高さ <ISLCT>
   内部変数'RSIZELi' で指定してある高さ (単位は V 座標系) の文字を描く (初期値は, 不定, 0.021, 0.028).

   b. 文字列の回転角 <CSIDE>
   内部変数'IROTLzs' で指定してある回転角オプションを使ってラベルを描く (初期値は, 0, 0, 0, 0, 0, 0). 回転角オプションとは, ラベルを描くときの回転角を 90 度が 1 単位の整数値で指定するものである. (たとえば 90 度のとき 1, –90 度のとき–1 と指定する.)

   c. 文字列のセンタリングオプション <CSIDE>
   内部変数'ICENTzs' で指定してあるセンタリングオプションを使ってラベルを描く (初期値は, 0, 0, 0, +1, +1, +1). センタリングオプションとは, ラベルを描くときに左よせ (–1) するか, センタリング (0) するか, 右よせ (+1) するかを指定するものである.

   d. 文字列を描く線のラインインデクス <ISLCT>
   内部変数'INDEXLi' で指定してあるラインインデクスを用いて文字を描く (初期値は, 不定, 3, 3).

4. タイトルについては

a. 文字の高さ <ISLCT>
内部変数'RSIZECi' で指定してある高さ (単位は V 座標系) の文字を描く (初期値は, 不定, 0.028, 0.035).

b. 文字列の回転角 <CSIDE>
内部変数'IROTCzs' で指定してある回転角オプションを使ってラベルを描く (初期値は, 0, 0, 0, +1, +1, +1). 回転角オプションとは, タイトルを描くときの回転角を 90 度が 1 単位の整数値で指定するものである. (たとえば 90 度のとき 1, −90 度のとき−1 と指定する.)

c. 文字列のセンタリングオプション <引き数>
引き数で指定してあるセンタリングオプションを使ってタイトルを描く. センタリングオプションとは, ラベルを描くときに左よせ (−1) するか, センタリング (0) するか, 右よせ (+1) するかを指定するものである.

d. 文字列を描く線のラインインデクス <ISLCT>
内部変数'INDEXLi' で指定してあるラインインデクスを用いて文字を描く (初期値は, 不定, 3, 3).

3.5 サブルーチンのリスト

上位応用ルーチン (ULA)

UXAXDV(CSIDE,DX1,DX2)	目盛とラベルを描く間隔を指定して座標軸を描く.
UYAXDV(CSIDE,DY1,DY2)	
UXAXNM(CSIDE,UX1,N1,UX2,N2)	目盛とラベルを描く場所を指定して座標軸を描く.
UYAXNM(CSIDE,UY1,N1,UY2,N2)	
UXAXLB(CSIDE,UX1,N1,UX2,CH,NC,N2)	目盛とラベルを描く場所を指定して座標軸を描く.
UYAXLB(CSIDE,UY1,N1,UY2,CH,NC,N2)	
UXMTTL(CSIDE,CTTL,PX)	大きめのタイトルを描く.
UYMTTL(CSIDE,CTTL,PY)	
UXSTTL(CSIDE,CTTL,PX)	小さめのタイトルを描く.
UYSTTL(CSIDE,CTTL,PY)	

下位応用ルーチン (LLA)

UXPAXS(CSIDE,ISLCT)	軸を示す線分を描く.
UYPAXS(CSIDE,ISLCT)	
UXPTMK(CSIDE,ISLCT,UX,N)	目盛を描く.
UYPTMK(CSIDE,ISLCT,UY,N)	
UXPLBL(CSIDE,ISLCT,UX,CH,NC,N)	文字列で指定したラベルを描く.
UYPLBL(CSIDE,ISLCT,UY,CH,NC,N)	
UXPNUM(CSIDE,ISLCT,UX,N)	数値で指定したラベルを描く.
UYPNUM(CSIDE,ISLCT,UY,N)	
UXPTTL(CSIDE,ISLCT,CTTL,PX)	タイトルを描く.
UYPTTL(CSIDE,ISLCT,CTTL,PY)	

UZPACK

UZpGET(CP,IPARA)	内部変数を参照する.
UZpSET(CP,IPARA)	内部変数を変更する.
UZpSTX(CP,IPARA)	内部変数を変更する (実行時オプション による変更を許す).
UZINIT	内部変数を初期化する.
UZFACT(RFACT)	内部変数を何倍かする.
UZPSAV	内部変数をスクラ ッチファイルに書き出す.
UZPRST	スクラ ッチファ イルに書いた値で再設定する.

3.6 サブルーチンの説明 : 上位応用ルーチン (ULA)

このサブルーチンパッケージ UXPACK/UYPACK/UZPACK において, 実際にユーザーが使用するであろうルーチン群. 下位応用ルーチン (LLA) を組み合わせて書かれており, 座標軸の構成要素の属性は UZpGET/UZpSET によって参照/変更することができる. 上位応用ルーチン (ULA) に関する以下の説明では, 内部で呼ばれている下位応用ルーチン名がしばしば引用されるので, くわしくは 3.8節を参照のこと.

上位応用ルーチン (ULA) に含まれるサブルーチンとしては以下のものがある.

- 座標軸 (目盛とラベル) の描画ルーチン

  UXAXDV/UYAXDV ··· 目盛とラベルを描く間隔を指定して座標軸を描く.

  UXAXNM/UYAXNM ··· 目盛とラベルを描く場所を指定して座標軸を描く.

  UXAXLB/UYAXLB ··· 目盛とラベルを描く場所, つけるラベルを指定して座標軸を描く.

- タイトルの描画ルーチン

  UXMTTL/UYMTTL ··· 大きめのタイトルを描く.

  UXSTTL/UYSTTL ··· 小さめのタイトルを描く.

等間隔の目盛と数値ラベルを描くだけなら UXAXDV/UYAXDV を用いればよい. 不等間隔の目盛と数値ラベルを描きたければ UXAXNM/UYAXNM を用いればよい. 数値ラベルのフォーマットはユーザーが指定することも可能なので, 数値ラベル座標軸の描画はほぼこれらのルーチンで事足りる. それでも満足できない場合は, 目盛とラベルを描く位置さらには描くラベルを指定するルーチン UXAXLB/UYAXLB を用いればよい. また, タイトルは UXMTTL/UYMTTL, UXSTTL/UYSTTL によって, 描いたラベル等のオフセット量を気にしなくても次々と重ならないように描くことができる.

以下の説明で共通してあらわれる引数 CSIDE は, 座標軸の構成要素を描く場所を指定するパラメーターで, X 軸については 'B', 'T', 'U' が指定できる; Y 軸については 'L', 'R', 'U' が指定できる (くわしくは 3.2 節参照).

目盛は CSIDE すべての選択について描かれる. しかしラベルを描くかどうかは, 場所に依存した内部変数 'LABELzs' が決める. この内部変数が .TRUE. なら描く, .FALSE. なら描かない (初期値は, .TRUE., .FALSE., .TRUE., .TRUE., .FALSE., .TRUE.). つまり初期設定では, 下 ('B') の座標軸と左 ('L') の座標軸についてラベルを描くが, 上 ('T') の座標軸と右 ('R') の座標軸についてはラベルを描かない (ユーザーの座標軸はラベルを描く).

以下のルーチンのうち, UXAXLB/UYAXLB, UXAXNM/UYAXNM, UXAXDV/UYAXDV については, 内部変数'LOFFSET' が .TRUE. なら座標系をリニアオフセットする;.FALSE. ならしない (初期値は.FALSE.). つまり, 内部変数 'LOFFSET' が .TRUE. のとき次のように座標系の変換がおこなわれウインドウが再設定される (この設定はルーチン内でのみ有効であって, ルーチンを抜け出すとき最初の設定に戻される; つまり座標軸を描くときのみ有効である).

UXPACK については内部変数'XOFFSET', 'XFACT' を参照して次の変換をおこなう.

$$f(x) = \mathtt{XFACT} \times x + \mathtt{XOFFSET}$$

UYPACK については内部変数'YOFFSET', 'YFACT' を参照して次の変換をおこなう.

$$g(y) = \mathtt{YFACT} \times y + \mathtt{YOFFSET}$$

これはたとえば, 温度に関するデータがあってデータの単位はセ氏で与えられているが, 目盛は華氏で打ちたいというときなどに有効である.

3.6.1 UXAXDV/UYAXDV

1. 機能

   目盛とラベルを描く間隔を指定して座標軸を描く.

2. 呼び出し方法

```
CALL UXAXDV(CSIDE,DX1,DX2)
CALL UYAXDV(CSIDE,DY1,DY2)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を描く場所を指定する (3.2節参照).
DX1, DY1	(R)	小さめの目盛を描く間隔を指定する (U 座標系の値で指定する).
DX2, DY2	(R)	大きめの目盛およびラベルを描く間隔を指定する (U 座標系の値で指定する). DX2 および DY2 はそれぞれ DX1 および DY1 の整数倍でなければならない.

4. 備考

   (a) 目盛およびラベルは指定した間隔の整数倍の場所に描かれる.

   (b) UXAXDV/UYAXDV はそれぞれ UXPAXS/UYPAXS (ISLCT=2) で軸を描いている; UXPTMK/UYPTMK (ISLCT=1) で小さめの目盛を描いている; UXPTMK/UYPTMK (ISLCT=2) で大きめの目盛を描いている; UXPNUM/UYPNUM (ISLCT=1) でラベルを描いている.

   (c) このルーチンは目盛とラベルを描く場所を内部で生成して UXPTMK/UYPTMK などを呼んでいる. この作業用配列の制約から目盛を打つ最大数は 200 以下でなければならない.

   (d) ラベルのフォーマットは UXSFMT/UYSMFT (3.8.4節参照) によって変更できる.

3.6.2 UXAXNM/UYAXNM

1. 機能

   目盛とラベルを描く場所を指定して座標軸を描く.

2. 呼び出し方法

```
CALL UXAXNM(CSIDE,UX1,N1,UX2,N2)
CALL UYAXNM(CSIDE,UY1,N1,UY2,N2)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を描く場所を指定する (3.2節参照).
UX1, UY1	(R)	小さめの目盛を描く場所を格納した長さ N1 の配列 (U 座標系の値で指定する).
N1	(I)	配列 UX1, UY1 の長さ.
UX2, UY2	(R)	大きめの目盛およびラベルを描く場所を格納した長さ N2 の配列 (U 座標系の値で指定する).
N2	(I)	配列 UX2, UY2 の長さ.

4. 備考

(a) UXAXNM/UYAXNM はそれぞれ UXPAXS/UYPAXS (ISLCT=2) で軸を描いている; UXPTMK/UYPTMK (ISLCT=1) で小さめの目盛を描いている; UXPTMK/UYPTMK (ISLCT=2) で大きめの目盛を描いている; UXPNUM/UYPNUM (ISLCT=1) でラベルを描いている.

(b) N1 として 0 以下の値を指定すると UXPTMK/UYPTMK (ISLCT=1) を呼ばない. このとき UX1, UY1 にはダミーの変数を与えればよい.

(c) N2 として 0 以下の値を指定すると UXPTMK/UYPTMK (ISLCT=2), UXPNUM/UYPNUM (ISLCT=1) を呼ばない. このとき UX2, UY2 にはダミーの変数を与えればよい.

(d) ラベルのフォーマットは UXSFMT/UYSMFT (3.8.4節参照) によって変更できる.

3.6.3 UXAXLB/UYAXLB

1. 機能

目盛とラベルを描く場所を指定して座標軸を描く. 描くラベルも指定する.

2. 呼び出し方法

```
CALL UXAXLB(CSIDE,UX1,N1,UX2,CH,NC,N2)
CALL UYAXLB(CSIDE,UY1,N1,UY2,CH,NC,N2)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を描く場所を指定する (3.2節参照).
UX1, UY1	(R)	小さめの目盛を描く場所を格納した長さ N1 の配列 (U 座標系の値で指定する).
N1	(I)	配列 UX1, UY1 の長さ.
UX2, UY2	(R)	大きめの目盛およびラベルを描く場所を格納した長さ N2 の配列 (U 座標系の値で指定する).
CH	(C*(NC))	描くラベルを格納した文字長 NC 長さ N2 の文字型配列.
NC	(I)	配列 CH の文字長.
N2	(I)	配列 UX2, UY2, CH の長さ.

4. 備考

(a) UXAXLB/UYAXLB はそれぞれ UXPAXS/UYPAXS (ISLCT=2) で軸を描いている; UXPTMK/UYPTMK (ISLCT=1) で小さめの目盛を描いている; UXPTMK/UYPTMK (ISLCT=2) で大きめの目盛を描いている; UXPLBL/UYPLBL (ISLCT=1) でラベルを描いている.

(b) N1 として 0 以下の値を指定すると UXPTMK/UYPTMK (ISLCT=1) を呼ばない. このとき UX1, UY1 にはダミーの変数を与えればよい.

(c) N2 として 0 以下の値を指定すると UXPTMK/UYPTMK (ISLCT=2), UXPLBL/UYPLBL (ISLCT=1) を呼ばない. このとき UX2, UY2, CH, NC にはダミーの変数を与えればよい.

3.6.4 UXMTTL/UYMTTL

1. 機能

   大きめのタイトルを描く.

2. 呼び出し方法

```
CALL UXMTTL(CSIDE,CTTL,PX)
CALL UYMTTL(CSIDE,CTTL,PY)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を描く場所を指定する (3.2節参照).
CTTL	(C*(*))	描くタイトル.
PX, PY	(R)	-1 から$+1$ の間の実数値でタイトルを描く位置を指定する. -1 のとき左よせ, 0 のときセンタリング, $+1$ のとき右よせする.

4. 備考

   (a) このルーチンは UXPTTL/UYPTTL (ISLCT=2) を使ってタイトルを描いている.

3.6.5 UXSTTL/UYSTTL

1. 機能

   小さめのタイトルを描く.

2. 呼び出し方法

```
CALL UXSTTL(CSIDE,CTTL,PX)
CALL UYSTTL(CSIDE,CTTL,PY)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を描く場所を指定する (3.2節参照).
CTTL	(C*(*))	描くタイトル.
PX, PY	(R)	-1 から$+1$ の間の実数値でタイトルを描く位置を指定する. -1 のとき左よせ, 0 のときセンタリング, $+1$ のとき右よせする.

4. 備考

   (a) このルーチンは UXPTTL/UYPTTL (ISLCT=1) を使ってタイトルを描いている.

3.7 サブルーチンの説明 : UZPACK

内部変数の管理や初期化をおこなうサブルーチンパッケージ.

3.7.1 UZpGET/UZpSET(UZpSTX)

1. 機能

   UXPACK/UYPACK/UZPACK で使う内部変数を参照/変更する. (UZPSTX は実行時オプションによる変更を許す.)

2. 呼び出し方法

```
CALL UZpGET(CP,IPARA)
CALL UZpSET(CP,IPARA)
CALL UZpSTX(CP,IPARA)
```

3. パラメーターの説明

CP	(C*8)	内部変数の名前.
IPARA	(I,R,L)	内部変数の値.

以下に CP として指定できる名前のリストを記す. 'ROFFzs', 'INDEXTi' などの略記法については 3.4節を参照のこと.

'UXUSER' (R) Y 軸の作画ルーチンにおいて場所を示すパラメーターとして'U'ser を指定したときに用いられる X 座標値 (初期値は不定). 詳しくは 3.2節参照.

'UYUSER' (R) X 軸の作画ルーチンにおいて場所を示すパラメーターとして'U'ser を指定したときに用いられる Y 座標値 (初期値は不定). 詳しくは 3.2節参照.

'ROFFzs' (R) 最後に描いたラベルやタイトルの軸からの距離を保持している内部変数 (初期値は, 0, 0, 0, 0, 0, 0). 3.8.1節参照.

'ROFGzs' (R) 座標軸を本来描く場所からどれだけ平行移動して描くかを指定する内部変数 (初期値は, 0, 0, 0, 0, 0, 0). 3.8.1節参照.

'LABELzs' (L) ラベルを描くかどうかを指定する内部変数; .TREU. なら描く, .FALSE. なら描かない (初期値は, .TRUE., .FALSE., .TRUE., .TRUE., .FALSE., .TRUE.). 3.6.3, 3.6.2, 3.6.1節参照.

'IROTLzs' (I) ラベルの文字列の回転角を指定する内部変数; 回転角は 90 度を単位とする整数値で指定する (初期値は, 0, 0, 0, 0, 0, 0). 3.4, 3.8.3節参照.

'IROTCzs' (I) タイトルの文字列の回転角を指定する内部変数; 回転角は 90 度を単位とする整数値で指定する (初期値は, 0, 0, 0, +1, +1, +1). 3.4, 3.8.5節参照.

'ICENTzs' (I) ラベルの文字列のセンタリングオプションを指定する内部変数; −1 のとき左よせ, 0 のときセンタリング, +1 のとき右よせする (初期値は, 0, 0, 0, +1, +1, +1). 3.4, 3.8.3節参照.

'INDEXT*i*' (I) 座標軸および目盛を描く線分のラインインデクスを指定する内部変数 (初期値は, 不定, 1, 3). 3.4, 3.8.2節参照.

'INDEXL*i*' (I) ラベルおよびタイトルの文字を描く線分のラインインデクスを指定する内部変数 (初期値は, 不定, 3, 3). 3.4, 3.8.3節参照.

'RSIZET*i*' (R) 目盛の長さを指定する内部変数 (単位は V 座標系; 初期値は, 不定, 0.007, 0.014). 3.4, 3.8.2節参照.

'RSIZEL*i*' (R) ラベルの文字の高さを指定する内部変数 (単位は V 座標系; 初期値は, 不定, 0.021, 0.028). 3.4, 3.8.3節参照.

'RSIZEC*i*' (R) タイトルの文字の高さを指定する内部変数 (単位は V 座標系; 初期値は, 不定, 0.028, 0.035). 3.4, 3.8.5節参照.

'LOFFSET' (L) 座標系をリニアオフセットするかどうかを指定する内部変数. .TRUE. ならリニアオフセットする; .FALSE. ならしない (初期値は.FALSE.). 3.6節参照.

'XOFFSET' (R) 'LOFFSET' が.TRUE. のとき, X 軸のオフセット値を指定する内部変数 (初期値は 0.0). 3.6節参照.

'YOFFSET' (R) 'LOFFSET' が.TRUE. のとき, Y 軸のオフセット値を指定する内部変数 (初期値は 0.0). 3.6節参照.

'XFACT' (R) 'LOFFSET' が.TRUE. のとき, X 軸のファクター値を指定する内部変数 (初期値は 1.0). 3.6節参照.

'YFACT' (R) 'LOFFSET' が.TRUE. のとき, Y 軸のファクター値を指定する内部変数 (初期値は 1.0). 3.6節参照.

'PAD1' (R) 1 つ前に描いた座標軸の構成要素からどれだけ離してラベルを描くかを指定する内部変数 (初期値は 0.7; 単位は今描こうとしているラベルの高さを 1 とする).

'PAD2' (R) 座標軸のオフセット量を内部的に決める時に用いる内部変数 (初期値は 1.5; 単位は内部変数'RSIZEC2' の大きさを 1 とする).

'IFLAG' (I) 座標軸の設定を'U'ser でおこなった場合, ラベルとタイトルを軸のどちら側に描くかを指定する内部変数. 'IFLAG' が正のときラベルは UXPLBL については上側, UYPLBL については右側に描かれる. 負のときはその逆である. 初期値は–1. 3.8.3節参照.

'LBTWN' (L) ラベルを描く場所に関するオプションを指定する内部変数. 'LBTWN' が.TRUE. なら間に描く; .FALSE. ならその場所に描く (初期値は .FALSE.). 3.8.3節参照.

'RBTWN' (R) ラベルを間に描くとき ('LBTWN' が.TRUE. のとき), さらにその位置を詳しく指定する. –1.0 のとき左よせ, 0.0 のときセンタリング, +1.0 のとき右よせする (初期値は 0.0).

'LBOUND' (L) タイトルを描こうとして, 文字幅がビューポートの幅を越えてしまうようなとき, タイトルを描くかどうかを指定する. .TRUE. なら描かない, .FALSE. なら描く (初期値は.FALSE.).

'LBMSG' (L) 'LBOUND' が.TRUE. でタイトルを描くスペースがないとき, エラーメッセージを出力するかどうかを指定する. .TRUE. なら出力する, .FALSE. なら出力しない (初期値は.TRUE.).

'INNER' (I) 目盛を内側に向かって打つか, 外側に向かって打つかを指定する内部変数. 'INNER' が正のとき内側に向かって目盛を打つ; 負のとき外側に向かって目盛を打つ (初期値は+1).

'IUNDEF' (I) 値が不定であることを示す内部変数 (初期値は-999).

4. 備考

(a) 内部変数を管理するための下請けルーチンとして以下のものがある.

UZPQNP(NCP)	内部変数の総数 NCP を求める.
UZPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
UZPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
UZPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
UZPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) UZpGET は上に述べた UZPQID を呼んで内部変数の位置を求め, UZPQVL によって値を参照する; UZpSET は UZPQID を呼んで内部変数の位置を求め, UZPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは UZPQID が出力する.

(c) 内部変数の型に応じ, IPARA として適切な型の定数あるいは変数を指定すること.

3.7.2 UZINIT

1. 機能

内部変数 'ROFFzs', 'ROFGzs' をすべて 0 にセットする.

2. 呼び出し方法

```
CALL UZINIT
```

3. パラメーターの説明

なし.

4. 備考

(a) 普通 UZINIT は GRFRM 中に記述されているので, GRPACK (第 2章参照) を用いていればユーザーは UZINIT を陽に呼ぶ必要はない.

3.7.3 UZFACT

1. 機能

内部変数 'RSIZETi', 'RSIZELi', 'RSIZECi' $(i = 1, 2)$ をすべて何倍かする.

2. 呼び出し方法

```
CALL UZFACT(RFACT)
```

3. パラメーターの説明

RFACT (R) 何倍するかを指定する実数値.

4. 備考

(a) なし.

3.7.4 UZPSAV

1. 機能

   現在設定されている内部変数の値すべてをスクラッチファイルに書き出す.

2. 呼び出し方法

   ```
   CALL UZPSAV
   ```

3. パラメーターの説明

   なし.

4. 備考

   (a) スクラッチファイルは MISC1/MISCLIB の IUFOPN が返す装置番号に対してオープンされる.

3.7.5 UZPRST

1. 機能

   スクラッチファイルに書き出してあった値ですべての内部変数を再設定する.

2. 呼び出し方法

   ```
   CALL UZPRST
   ```

3. パラメーターの説明

   なし.

4. 備考

   (a) なし.

3.8 サブルーチンの説明： 下位応用ルーチン (LLA)

このサブルーチンパッケージ UXPACK/UYPACK/UZPACK において, 具体的な作画をおこなうルーチン群. 基本ルーチン (Basic Routines) を使って書かれており普通ユーザーが直接呼ぶことはなく, 上位応用ルーチン (ULA) を作成するのに用いられる.

以下の説明で共通してあらわれる引数 CSIDE は座標軸の構成要素を描く場所を指定するパラメーターで, X 軸については 'B', 'T', 'U' が指定できる; Y 軸については 'L', 'R', 'U' が指定できる (くわしくは 3.2 節参照). また, 引数 ISLCT は座標軸の構成要素の属性を指定するパラメーターで, 0, 1, 2 のうち 1 つが指定できる (くわしくは 3.4節参照).

3.8.1 UXPAXS/UYPAXS

1. 機能

   軸を示す線分を描く.

2. 呼び出し方法

```
CALL UXPAXS(CSIDE,ISLCT)
CALL UYPAXS(CSIDE,ISLCT)
```

3. パラメーターの説明

CSIDE	(C*1)	線分を描く座標軸の場所を指定する (3.2節参照).
ISLCT	(I)	描く軸の属性を指定する (3.4節参照).

4. 備考

   (a) 軸を示す線分のラインインデクスは内部変数'INDEXTi' が決める.

   (b) 構成要素を描く場所に依存する内部変数として'ROFFzs' がある. これは, 最後に描いたラベルやタイトルの軸からの距離を保持している内部変数である. この内部変数を初期化しなければ, ラベルやタイトルは次々とウインドウの外側に向かって重ならないよう描かれる.

   (c) 座標軸は, 本来描くべき場所から平行移動した場所に描くことができる (3.2節参照). この移動量は, 描く場所に依存する内部変数'ROFGzs' が管理する (初期値はすべて 0). この値は X/Y 軸それぞれについて, サブルーチン

```
UXSAXZ(CSIDE,ROFFX) / UYSAXZ(CSIDE,ROFFY)
```

   によって設定できる. またオフセット量を内部的に決定するルーチン

```
UXSAXS(CSIDE) / UYSAXS(CSIDE)
```

   によっても設定できる. オフセット量は内部変数'PAD2' にもとづいて計算される.

   (d) 内部変数'ROFFzs' と'ROFGzs' が等しいときに限って軸を示す線分を描く. これらの内部変数はサブルーチン UZINIT によって 0 にセットされる. 普通 UZINIT は GRFRM 中に記述されているので, GRPACK (第 2章参照) を用いていればユーザーは UZINIT を陽に呼ぶ必要はない.

3.8.2 UXPTMK/UYPTMK

1. 機能

   目盛を描く.

2. 呼び出し方法

```
CALL UXPTMK(CSIDE,ISLCT,UX,N)
CALL UYPTMK(CSIDE,ISLCT,UY,N)
```

3. パラメーターの説明

CSIDE	(C*1)	目盛を描く座標軸の場所を指定する (3.2節参照).
ISLCT	(I)	描く目盛の属性を指定する (3.4節参照).
UX, UY	(R)	目盛を描く場所を格納した長さ N の配列. (U 座標系の値で指定する.)
N	(I)	配列 UX, UY の長さ.

4. 備考

   (a) 目盛の長さは内部変数'RSIZETi' が決める.

   (b) 目盛を描く線のラインインデクスは内部変数'INDEXTi' が決める.

   (c) 目盛を内側に向かって打つか, 外側に向かって打つかは内部変数'INNER' が決める. 'INNER' が正のとき内側に向かって目盛を打つ; 負のとき外側に向かって目盛を打つ (初期値は+1).

3.8.3 UXPLBL/UYPLBL

1. 機能

   文字列で指定したラベルを描く.

2. 呼び出し方法

```
CALL UXPLBL(CSIDE,ISLCT,UX,CH,NC,N)
CALL UYPLBL(CSIDE,ISLCT,UY,CH,NC,N)
```

3. パラメーターの説明

CSIDE	(C*1)	ラベルを描く座標軸の場所を指定する (3.2節参照).
ISLCT	(I)	描くラベルの属性を指定する (3.4節参照).
UX, UY	(R)	ラベルを描く場所を格納した長さ N の配列.(U 座標系の値で指定する.)
CH	(C*(NC))	描くラベルを格納した文字長 NC 長さ N の文字型配列.
NC	(I)	配列 CH の文字長.
N	(I)	配列 UX, UY, CH の長さ.

4. 備考

   (a) このパッケージでは次の 2 通りのラベルの描きかたがある. UXPLBL/UYPLBL それぞれについて

   - UX(i)/UY(i) の場所に CH(i) のラベルを描く.

- UX(i)/UY(i) と UX(i+1)/UY(i+1) の間に CH(i) のラベルを描く.

2 つの方法のどちらを選ぶかは内部変数'LBTWN' が決める. 'LBTWN' が.TRUE. なら間に描く; .FALSE. ならその場所に描く (初期値は.FALSE.). ただし, ラベルを間に描くように指定していても, ラベルが 2 点の間隔より大きいときラベルは描かれない. またメッセージも出力されない.

(b) ラベルの文字の高さは内部変数'RSIZELi' が決める.

(c) ラベルの文字列の回転角は内部変数'IROTLzs' が決める.

(d) ラベルの文字列のセンタリングオプション は内部変数'ICENTzs' が決める.

(e) ラベルの文字を描く線分のラインインデクスは内部変数'INDEXLi' が決める.

(f) 文字列 CH(i) の文字長は LENC (『MATH1 」マニュアルの CHRLIB 参照) が決める. つまり後続のブランクは無視される.

(g) CSIDE として'U' 以外を指定した場合, ラベルは常にウインドウの外側に描かれる. CSIDE として'U' を指定した場合, ラベルの描かれる位置は内部変数'IFLAG' が決める. 'IFLAG' が正のときラベルは UXPLBL については上側, UYPLBL については右側に描かれる. 負のときはその逆である. 初期値は–1.

(h) 1 つ前に描いた座標軸の構成要素からどれだけ離してラベルを描くかは内部変数'PAD1' が決める. 初期値は 0.7. 単位は今描こうとしているラベルの高さを 1 とする.

3.8.4 UXPNUM/UYPNUM

1. 機能

数値で指定したラベルを描く.

2. 呼び出し方法

```
CALL UXPNUM(CSIDE,ISLCT,UX,N)
CALL UYPNUM(CSIDE,ISLCT,UY,N)
```

3. パラメーターの説明

CSIDE	(C*1)	ラベルを描く座標軸の場所を指定する (3.2節参照).
ISLCT	(I)	描くラベルの属性を指定する (3.4節参照).
UX, UY	(R)	ラベルを描く場所を格納した長さ N の配列.(U 座標系の値で指定する.)
N	(I)	配列 UX, UY の長さ.

4. 備考

(a) このルーチンは, サブルーチン CHVAL (『MISC1 」マニュアルの FMTLIB 参照) を使って UX, UY の値を文字化したあとで UXPLBL/UYPLBL を呼んでいる. (ラベルの属性については, UXPLBL/UYPLBL を参照のこと.) UXPLBL/UYPLBL に渡す文字型配列を内部で用意しているため UX, UY の長さ N は 40 以下でなければならない.

(b) サブルーチン CHVAL に渡すフォーマットは, UXPNUM についてはサブルーチン

```
UXSFMT(CFMT) / UXQFMT(CFMT)
```

で設定/参照することができる. UYPNUM についてはサブルーチン

UYSFMT(CFMT) / UYQFMT(CFMT)

で設定/参照することができる. CFMT は長さ 16 以下の文字型変数. たとえば CFMT として '(F6.1)' のように指定する. 初期値は CHVAL に固有のオプションである'B' (有効数字 3 ケタで文字化し, 小数点以下に続く後方の 0 は取り除く; さらに小数点が最下位ならそれも取り除く). CHVAL は自動的に文字を左詰めする. つまり実数値 1.2 を文字化するために'(F4.1)' としても'(F8.1)' としても CHVAL が返す値はどちらも'1.2' である.

3.8.5 UXPTTL/UYPTTL

1. 機能

   タイトルを描く.

2. 呼び出し方法

```
CALL UXPTTL(CSIDE,ISLCT,CTTL,PX)
CALL UYPTTL(CSIDE,ISLCT,CTTL,PY)
```

3. パラメーターの説明

CSIDE	(C*1)	タイトルを描く座標軸の場所を指定する (3.2節参照).
ISLCT	(I)	描くタイトルの属性を指定する (3.4節参照).
CTTL	(C*(*))	描くタイトル.
PX, PY	(R)	-1 から$+1$ の間の実数値でタイトルを描く位置を指定する. -1 のとき左よせ, 0 のときセンタリング, $+1$ のとき右よせする.

4. 備考

   (a) タイトルの文字の大きさは内部変数'RSIZECi' が決める.

   (b) タイトルの文字列の回転角は内部変数'IROTCzs' が決める.

   (c) タイトルの文字を描く線分のラインインデクスは内部変数'INDEXLi' が決める.

   (d) 内部変数'LBOUND' が.TRUE. でタイトルがウインドウの幅をこえるとタイトルを描かない; .FALSE. ならウインドウの幅をこえても描く (初期値は.FALSE.). また, 内部変数'LBOUND' が.TRUE. でタイトルがウインドウの幅をこえて描けないとき内部変数'LBMSG' が.TRUE. ならその旨メッセージを出力する;.FALSE. なら出力しない (初期値は.TRUE.).

3.9 その他のサブルーチン ・関数のリスト

このパッケージにはこのほかに以下の下位ルーチン ・関数がある. ここではそれらの名前をあげるにとどめる.

```
LUXCHK  UXPLBA  UXPLBB  UXPTMZ  UXSOFF  UXROFF
LUYCHK  UYPLBA  UYPLBB  UYPTMZ  UYSOFF  UYROFF
```

第 4 章

ULPACK：対数座標軸ルーチン

4.1 概要

ULPACK は対数座標軸をプロットするためのサブルーチンパッケージである. したがって, 変換関数番号としては ITR= $1 \sim 4$(直角直線座標系) の正規化変換に対してのみ使用できる. それ以外の変換関数番号を指定しても動作は保証されない.

このパッケージは直交座標軸をプロットするサブルーチンパッケージ UXPACK/UYPACK/UZPACK (第 3章参照) にもとづいて書かれている. また USPACK (第 7章参照) における対数座標軸の作画は, このパッケージを用いておこなわれている.

このパッケージでは, 作画ルーチンで与える引数や内部変数を変更することによって, いろいろなタイプの対数座標軸を描くことができる. まず, 作画ルーチン (ULXLOG/ULYLOG) で与える引数によって, 1 桁の範囲に何個のラベルを描くかが指定できる. また, ULpGET/ULpSET が管理する内部変数を変更することによって座標軸に描くラベルの書式を変更することができる.

以下の説明であらわれる引数 CSIDE は, 座標軸の構成要素を描く場所を指定するパラメーターで, X 軸については 'B', 'T', 'U' が指定できる; Y 軸については 'L', 'R', 'U' が指定できる (くわしくは 3.2節参照).

4.2 サブルーチンのリスト

ULXLOG(CSIDE,NLBL,NTICKS)	対数座標軸を描く.
ULYLOG(CSIDE,NLBL,NTICKS)	
ULXSFM(CFMT)	フォーマットを設定する.
ULYSFM(CFMT)	
ULSXBL(BL,NBL)	ラベルを描く位置を指定する.
ULSYBL(BL,NBL)	
ULpGET(CP,IPARA)	内部変数を参照する.
ULpSET(CP,IPARA)	内部変数を設定する.
ULpSTX(CP,IPARA)	内部変数を設定する. (実行時オプションによる変更を許す).

4.3 サブルーチンの説明

4.3.1 ULXLOG/ULYLOG

1. 機能

   対数座標軸を描く.

2. 呼び出し方法

```
CALL ULXLOG(CSIDE,NLBL,NTICKS)
CALL ULYLOG(CSIDE,NLBL,NTICKS)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を描く場所を指定する (3.2節参照).
NLBL	(I)	1 桁の範囲に描くラベルの数. 1 から 4 までの整数値を指定する. 1: 10^n のみにラベルを描く. 2: 10^n と $2{\cdot}10^n$ にラベルを描く. 3: 10^n, $2{\cdot}10^n$ と $5{\cdot}10^n$ にラベルを描く. 4: ユーザーが指定した場所にラベルを描く.
NTICKS	(I)	1 桁の範囲に描く目盛の数. $1 \leq$NTICKS≤ 9 の範囲で指定する. 9 より小さな値を指定した場合には, 目盛間隔の狭いところから省かれる.

4. 備考

   (a) ULXLOG/ULYLOG はそれぞれ UXPAXS/UYPAXS (ISLCT=2) で軸を描いている; UXPTMK/UYPTMK (ISLCT=1) で 10^n 以外の小さめの目盛を描いている; UXPTMK/UYPTMK (ISLCT=2) で 10^n の大きめの目盛を描いている; UXPLBL/UYPLBL (ISLCT=1) でラベルを描いている. したがって, これら座標軸の構成要素の属性は, UZpGET/UZpSET によって参照/変更することができる. くわしくは第 3章を参照のこと.

   (b) このルーチンは小さめの目盛を打つ場所を内部で生成して UXPTMK/UYPTMK を呼んでいる. この作業用配列の制約から目盛を打つ最大数は 200 以下でなければならない.

   (c) このルーチンはラベルを描く場所を内部で生成して UXPLBL/UYPLBL を呼んでいる. この作業用配列の制約から描くラベルの最大数は 50 以下でなければならない.

(d) NLBL として 4 を指定する場合は, あらかじめ ULSXBL/ULSYBL によってラベルを描く位置が指定されてなければならない.

(e) ラベルのフォーマットは, ULpGET/ULpSET が管理する内部変数'IXTYPE', 'IYTYPE' (それぞれ X 軸, Y 軸のフォーマット) で指定する (初期値はそれぞれ 1: すべてのラベルに指数部をつける).

4.3.2 ULXSFM/ULYSFM

1. 機能

   内部変数'IXTYPE'/'IYTYPE' が 3 または 4 のときに使用するフォーマットを設定する.

2. 呼び出し方法

```
CALL ULXSFM(CFMT)
CALL ULYSFM(CFMT)
```

3. 引数の説明

CFMT	(C*(*))	文字書式仕様あるいはサブルーチン CHVAL (「MISC1」のマニュアル FMTLIB 参照) が解釈するオプション.

4. 備考

   (a) サブルーチン CHVAL に渡すフォーマットは, ULXSFM/ULYSFM についてはサブルーチン

```
ULXQFM(CFMT) / ULYQFM(CFMT)
```

   で参照することができる. 初期値は CHVAL に固有のオプションである'B' (有効数字 3 ケタで文字化し, 小数点以下に続く後方の 0 は取り除く; さらに小数点が最下位ならそれも取り除く).

4.3.3 ULSXBL/ULSYBL

1. 機能

   ラベルを描く位置を指定する.

2. 呼び出し方法

```
CALL ULSXBL(BL,NBL)
CALL ULSYBL(BL,NBL)
```

3. パラメーターの説明

BL	(R)	1 桁の範囲に描くラベルの位置. 各要素の値は 1 以上で 10 より小さくなければならない. また BL(1) は 1 でなければならない.
NBL	(I)	配列 BL の長さ.

4. 備考

   (a) ULXLOG/ULYLOG において NLBL として 4 を指定する場合は, あらかじめ本ルーチンによってラベルを描く位置を指定しなければならない.

   (b) BL の長さは 9 以下でなければならない.

(c) ラベルを描く位置は, X/Y 軸についてサブルーチン

```
ULQXBL(BL,NBL) / ULQYBL(BL,NBL)
```

で参照することができる.

(d) 内部的に, ラベルを描く位置を得るルーチンとして ULXLBL/ULYLBL がある.

4.3.4 ULpGET/ULpSET(ULpSTX)

1. 機能

ULPACK サブルーチンパッケージで使用する内部変数を参照/設定する. (ULpSTX は実行時オプションによる変更を許す.)

2. 呼び出し方法

```
CALL ULpGET(CP,IPARA)
CALL ULpSET(CP,IPARA)
CALL ULpSTX(CP,IPARA)
```

3. パラメーターの説明

CP	(C*8)	内部変数の名前.
IPARA	(I)	内部変数の値.

以下に CP として指定できる名前のリストを記す.

'IXCHR'	(I)	X 軸に関する $2 \cdot 10^n$ という形のラベルで, '·' の位置に使う文字番号. DCL 文字番号 (『GRPH1』のマニュアル参照) で指定する. 初期値は 195 (·).
'IYCHR'	(I)	Y 軸に関する $2 \cdot 10^n$ という形のラベルで, '·' の位置に使う文字番号. DCL 文字番号 (『GRPH1』のマニュアル参照) で指定する. 初期値は 195 (·).
'IXTYPE'	(I)	X 軸に関するラベルのフォーマットを指定する. 1 から 4 の整数値が指定できる (初期値は 1). それぞれの意味は次項で説明.

'IYTYPE' (I) Y 軸に関するラベルのフォーマットを指定する. 1 から 4 の整数値が指定できる (初期値は 1).

1: 全てのラベルに指数部を付ける.

例 $1\cdot10^3$ $2\cdot10^3$ $5\cdot10^3$ $1\cdot10^4$

ただし, SGLGET/SGLSET (『GRPH1』のマニュアル参照) が管理する内部変数'LCNTL' が.FALSE. のときには$\cdot10^n$のかわりに En と表現する.

2: 10^n以外のラベルには指数部を付けない.

例 10^3 2 5 10^4

3: サブルーチン CHVAL (『MISC1』のマニュアル FMTLIB 参照) を使って全てのラベルを文字化する.

例 1000 2000 5000 10000

4: 10^nのラベルのみ CHVAL を使う. そのほかは 2 を指定したときと同じ.

例 1000 2 5 10000

4. 備考

(a) 内部変数を管理するための下請けルーチンとして以下のものがある.

ULPQNP(NCP)	内部変数の総数 NCP を求める.
ULPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
ULPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
ULPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
ULPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) ULpGET は上に述べた ULPQID を呼んで内部変数の位置を求め, ULPQVL によって値を参照する; ULpSET は ULPQID を呼んで内部変数の位置を求め, ULPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは ULPQID が出力する.

(c) 'IXTYPE'/'IYTYPE' が 3 または 4 のとき, CHVAL に渡すフォーマットはそれぞれ ULXSFM/ULYSFM で変更することができる.

第 5 章

UCPACK : 日付に関する座標軸ルーチン

5.1 概要

UCPACK は日付を目盛とする座標軸をプロットするためのサブルーチンパッケージである.

このパッケージは直交座標軸をプロットするサブルーチンパッケージ UXPACK/UYPACK/UZPACK (第3章参照) にもとづいて書かれている.

作画しようとする座標軸については, 日数を単位として正規化変換が設定してなければならない. また座標軸の作画は, ユーザー座標系 (UC) において 0 に相当する位置からおこなわれる. したがって, たとえば 1 月 1 日から 1 月 31 日まで 31 日分をビューポートいっぱいに割り当てたいときは, ウインドウの両端値を (0, 30) と指定する.

以下でサブルーチンに共通してあらわれる引数 CSIDE, JD0, ND について説明する.

CSIDE – 場所を指定する文字型の引数. X 軸に関しては'B'ottom, 'T'op, 'U'ser が指定できる. Y 軸に関しては'L'eft, 'R'ight, 'U'ser が指定できる. 詳しくは 3.2節を参照のこと.

JD0 – UC で 0 に相当する場所の日付. 上 5 桁以上が年, 3,4 桁が月, 1,2 桁が日の 1 整数で日付をあらわす. たとえば 1990 年 9 月 1 日は 19900901 と表現する.

ND – JD0 を 0 日目として作画する日数. たとえば 1990 年 9 月 1 日から 1990 年 9 月 30 日まで 30 日分を描きたいとすると JD0 を 19900901 として ND を 29 とすればよい.

5.2 サブルーチンのリスト

UCXACL(CSIDE,JD0,ND)	日 ・月 ・年に関する座標軸を描く.
UCYACL(CSIDE,JD0,ND)	
UCXADY(CSIDE,JD0,ND)	日に関する座標軸を描く.
UCYADY(CSIDE,JD0,ND)	
UCXAMN(CSIDE,JD0,ND)	月に関する座標軸を描く.
UCYAMN(CSIDE,JD0,ND)	
UCXAYR(CSIDE,JD0,ND)	年に関する座標軸を描く.
UCYAYR(CSIDE,JD0,ND)	
UCpGET(CP,IPARA)	内部変数を参照する.
UCpSET(CP,IPARA)	内部変数を設定する.
UCpSTX(CP,IPARA)	内部変数を設定する. (実行時オプションによる変更を許す).

5.3 サブルーチンの説明

5.3.1 UCXACL/UCYACL

1. 機能

   日 ・月 ・年に関する座標軸を描く.

2. 呼び出し方法

```
CALL UCXACL(CSIDE,JD0,ND)
CALL UCYACL(CSIDE,JD0,ND)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を書く場所を指定する.
JD0	(I)	UC で 0 に相当する場所の日付を指定する.
ND	(I)	作画する日数を指定する.

4. 備考

   (a) UCXACL は UCXADY, UCXAMN, UCXAYR (以下の節参照) を続けて呼んでいる.

   (b) UCYACL は UCYADY, UCYAMN, UCYAYR (以下の節参照) を続けて呼んでいる.

5.3.2 UCXADY/UCYADY

1. 機能

   日に関する座標軸を描く.

2. 呼び出し方法

```
CALL UCXADY(CSIDE,JD0,ND)
CALL UCYADY(CSIDE,JD0,ND)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を書く場所を指定する.
JD0	(I)	UC で 0 に相当する場所の日付を指定する.
ND	(I)	作画する日数を指定する.

4. 備考

(a) 何日ごとに目盛と日付を描くかは, 関数 NUCDAY を通して内部変数'NDAY' が決める. 'NDAY' が不定なら 0, 1, 2, 5, 10, 15 の中から適当な値が選ばれる (0 以下のときは軸を描くだけで目盛と日付は描かない); 陽に与えられていればその値を使う (初期値は不定).

(b) 目盛と日付を描くための情報を保持するために内部で配列を用意している. その配列の大きさに関する制限から, 描きうる目盛と日付は最大 100 個である.

(c) UCXADY/UCYADY はそれぞれ UXPAXS/UYPAXS (ISLCT=2) で軸を描いている; UXPTMK/UYPTMK (ISLCT=1) で目盛を描いている; UXPLBL/UYPLBL (ISLCT=1) で日のラベルを描いている. したがって, これら座標軸の構成要素の属性は, UZpGET/UZpSET によって参照/変更することができる. ただし, 'ICENTzs' は 0 とする; 'IROTLzs' は'IROTCzs' を参照して用いる; 'LBTWN' は .FALSE. とする. くわしくは第 3章を参照のこと.

5.3.3 UCXAMN/UCYAMN

1. 機能

月に関する座標軸を描く.

2. 呼び出し方法

```
CALL UCXAMN(CSIDE,JD0,ND)
CALL UCYAMN(CSIDE,JD0,ND)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を書く場所を指定する.
JD0	(I)	UC で 0 に相当する場所の日付を指定する.
ND	(I)	作画する日数を指定する.

4. 備考

(a) 月名を何文字描くかは, 関数 NUCCHR を通して内部変数'NCHAR' が決める. 'NCHAR' が不定なら 0, 1, 2, 3, 9 の中から適当な値が選ばれる (0 のときは軸を描くだけで目盛と月名は描かない); 陽に与えられていればその値を使う (初期値は不定). 'NCHAR' の値が負のときはその絶対値の長さの月名を書き, さらに 2 文字目以降を小文字にする.

(b) 目盛と月名を描くための情報を保持するために内部で配列を用意している. その配列の大きさに関する制限から, 描きうる目盛と月名は最大 50 個である.

(c) UCXAMN/UCYAMN はそれぞれ UXPAXS/UYPAXS (ISLCT=2) で軸を描いている; UXPTMK/UYPTMK (ISLCT=2) で目盛を描いている; UXPLBL/UYPLBL (ISLCT=2) で月のラベルを描いている. したがって, これら座標軸の構成要素の属性は, UZpGET/UZpSET によって参照/変更することができる. ただし, 'ICENTzs' は 0 とする; 'IROTLzs' は'IROTCzs' を参照して用いる; 'LBTWN' は .TRUE. とする. くわしくは第 3章を参照のこと.

5.3.4 UCXAYR/UCYAYR

1. 機能

   年に関する座標軸を描く.

2. 呼び出し方法

```
CALL UCXAYR(CSIDE,JD0,ND)
CALL UCYAYR(CSIDE,JD0,ND)
```

3. パラメーターの説明

CSIDE	(C*1)	座標軸を書く場所を指定する.
JD0	(I)	UC で 0 に相当する場所の日付を指定する.
ND	(I)	作画する日数を指定する.

4. 備考

   (a) 目盛と年を表す数字を描くための情報を保持するために内部で配列を用意している. その配列の大きさに関する制限から, 描きうる目盛と数字は最大 50 個である.

   (b) UCXAYR/UCYAYR はそれぞれ UXPAXS/UYPAXS (ISLCT=2) で軸を描いている; UXPTMK/UYPTMK (ISLCT=2) で目盛を描いている; UXPLBL/UYPLBL (ISLCT=2) で年のラベルを描いている. したがって, これら座標軸の構成要素の属性は, UZpGET/UZpSET によって参照/変更することができる. ただし, 'ICENTzs' は 0 とする; 'IROTLzs' は'IROTCzs' を参照して用いる; 'LBTWN' は .TRUE. とする; 'RSIZET2' は 1.5 倍している. くわしくは第 3章を参照のこと.

5.3.5 UCpGET/UCpSET(UCpSTX)

1. 機能

   UCPACK で使う内部変数を参照/変更する. (UCpSTX は実行時オプションによる変更を許す.)

2. 呼び出し方法

```
CALL UCpGET(CP,IPARA)
CALL UCpSET(CP,IPARA)
CALL UCpSTX(CP,IPARA)
```

3. パラメーターの説明

CP	(C*8)	内部変数の名前.
IPARA	(I,R,L)	内部変数の値.

   以下に CP として指定できる名前のリストを記す.

'NDAY' (I) UCXADY/UCYADY を使うとき, 何日ごとに目盛と日付を描くかを指定する. 'NDAY' が不定なら 0, 1, 2, 5, 10, 15 の中から適当な値が選ばれる (0 以下のときは軸を描くだけで目盛と日付は描かない); 陽に与えられていればその値を使う (初期値は不定). これは関数 NUCDAY が決める.

'NCHAR' (I) UCXAMN/UCYAMN を使うとき, 月名を何文字描くかを指定する. 'NCHAR' が不定なら 0, 1, 2, 3, 9 の中から適当な値が選ばれる (0 のときは軸を描くだけで目盛と月名は描かない); 陽に与えられていればその値を使う (初期値は不定). これは関数 NUCCHR が決める.

'LOWER' (L) UCXADY/UCYADY で描く月名の 2 文字目以下を小文字にするかどうかを指定する. .TRUE. なら小文字にする; .FALSE. なら大文字のままにする (初期値は.FALSE).

'IUNDEF' (I) 値が不定であることを示す内部変数 (初期値は-999).

4. 備考

(a) 内部変数を管理するための下請けルーチンとして以下のものがある.

UCPQNP(NCP)	内部変数の総数 NCP を求める.
UCPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
UCPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
UCPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
UCPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) UCpGET は上に述べた UCPQID を呼んで内部変数の位置を求め, UCPQVL によって値を参照する; UCpSET は UCPQID を呼んで内部変数の位置を求め, UCPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは UCPQID が出力する.

(c) 内部変数の型に応じ, IPARA として適切な型の定数あるいは変数を指定すること.

5.3.6 その他の関数

このパッケージにはこのほかに以下の下位関数がある. ここではその名前をあげるにとどめる.

NUCCHR NUCDAY

第 6 章

U[UVH]PACK：1 次元グラフルーチン

6.1 概要

UUPACK/UVPACK/UHPACK は直交座標系で折れ線やマーカー, 棒グラフなどを描くためのサブルーチンパッケージである. したがって, 変換関数番号としては ITR= $1 \sim 4$(直角直線座標系) の正規化変換に対してのみ使用できる. それ以外の変換関数番号を指定しても動作は保証されない.

USPACK (第 7章) の折れ線は, このパッケージのルーチンを使用して描かれている.

折れ線を描くルーチン (UULIN, UULINZ) およびマーカー列を描くルーチン (UUMRK, UUMRKZ) に関しては, 次節で述べる未定義値を引数に指定しなければ, GRPH1 における対応するルーチン (SGPLU, SGPLZU および SGPMU, SGPMZU) と同じ動作をする. (ただし, ラインインデクス等のパラメータは独立である.)

UUPACK/UVPACK/UHPACK それぞれのパッケージの機能は次のようになっている.

UUPACK : X 軸と Y 軸が対等なグラフ要素 (折れ線など) およびパラメタ管理.
UVPACK : X 軸が独立変数で Y 軸が従属変数となるようなグラフ要素.
UHPACK : Y 軸が独立変数で X 軸が従属変数となるようなグラフ要素.

これらのサブルーチン群は, 変数の数によって次の 4 つのタイプに分類される.

	独立変数の数	従属変数の数
TYPE 1	N	N
TYPE 2	N	$N \times 2$
TYPE 3	$N + 1$	N
TYPE 4	$N + 1$	$N \times 2$

ここで, 独立変数として指定する配列の数は常に 1 つで, その長さが N または $N + 1$ である. 従属変数は 1 つの配列の長さは常に N であるが, 2 つの配列を指定するもの $(N \times 2)$ がある.

TYPE 3 と TYPE 4 のルーチンはこのパッケージでは 「箱グラフ」と呼んでいる. これは, 指定した独立変数の 「位置」にある幅の棒を描く棒グラフに対して, 指定した独立変数の 「区間」を埋める箱を描くものである. この中の U[VH]BXL, U[VH]BXLZ は階段状のグラフを描くものである.

6.2 未定義値 (RUNDEF) の解釈

UUPACK/UVPACK/UHPACK のサブルーチンは, 独立変数または従属変数に未定義値 (RUNDEF) を指定することができる. この値が指定された時の解釈は次の通り.

独立変数が RUNDEF の時: 独立変数に関して等間隔のデータであると解釈される. 独立変数の定義域は USIDV(UMIN, UMAX) で指定される. この範囲が指定されていない場合は, ウインドウ全体を独立変数の定義域とみなす. TYPE1 および TYPE2 のルーチンでは定義域を N-1 で, また, TYPE3 および TYPE4 のルーチンでは定義域を N で均等に分割する.

従属変数が RUNDEF の時: 内部変数 UREF (初期値=0) の値が指定されたものとみなす. ただし, 従属変数に RUNDEF が指定できるのは, 2 つの従属変数をもつ TYPE2, TYPE4 のルーチンのみである.

従属変数の未定義値は, 棒グラフのように棒の上下の位置を指定するようなサブルーチンで, 常に座標軸から指定された位置までの棒を描くような場合を想定している.

6.3 サブルーチンのリスト

折れ線 (TYPE1)

UULIN (N,X,Y)	折れ線を描く.
UULINZ(N,X,Y,ITYPE,INDEX)	属性を指定して折れ線を描く.

マーカー (TYPE1)

UUMRK (N,X,Y)	マーカー列を描く.
UUMRKZ(N,X,Y,ITYPE,INDEX,RSIZE)	属性を指定してマーカー列を描く.

エラーバー (TYPE2)

UVERB (N,X,Y1,Y2)	エラーバーを描く.
UHERB (N,X1,X2,Y)	エラーバーを描く.
UVEFBZ(N,X,Y1,Y2,ITYPE,INDEX,RSIZE)	属性を指定してエラーバーを描く.
UHEFBZ(N,X1,X2,Y,ITYPE,INDEX,RSIZE)	属性を指定してエラーバーを描く.

差分 (TYPE2)

UVDIF (N,X,Y1,Y2)	2 本の線の間をトーンパターンで塗る.
UHDIF (N,X1,X2,Y)	2 本の線の間をトーンパターンで塗る.
UVDIFZ(N,X,Y1,Y2,ITP1,ITP2)	2 本の線の間を指定されたトーンパターンで塗る.
UHDIFZ(N,X1,X2,Y,ITP1,ITP2)	2 本の線の間を指定されたトーンパターンで塗る.

棒グラフ (TYPE2, TYPE1)

UVBRF (N,X,Y1,Y2)	棒グラフの枠を描く.
UHBRF (N,X1,X2,Y)	棒グラフの枠を描く.
UVBRA (N,X,Y1,Y2)	棒グラフの内部領域を塗る.
UHBRA (N,X1,X2,Y)	棒グラフの内部領域を塗る.
UVBRL (N,X,Y)	棒グラフを線で結ぶ.
UHBRL (N,X,Y)	棒グラフを線で結ぶ.
UVBRFZ(N,X,Y1,Y2,ITYPE,INDEX,RSIZE)	属性を指定して棒グラフの枠を描く.
UHBRFZ(N,X1,X2,Y,ITYPE,INDEX,RSIZE)	属性を指定して棒グラフの枠を描く.
UVBRAZ(N,X,Y1,Y2,ITP1,ITP2,RSIZE)	属性を指定して棒グラフの内部領域を塗る.
UHBRAZ(N,X1,X2,Y,ITP1,ITP2,RSIZE)	属性を指定して棒グラフの内部領域を塗る.
UVBRLZ(N,X,Y,ITYPE,INDEX,RSIZE)	属性を指定して棒グラフを線で結ぶ.
UHBRLZ(N,X,Y,ITYPE,INDEX,RSIZE)	属性を指定して棒グラフを線で結ぶ.

箱グラフ (TYPE4, TYPE3)

UVBXF (N,X,Y1,Y2)	箱グラフの枠を描く.
UHBXF (N,X1,X2,Y)	箱グラフの枠を描く.
UVBXA (N,X,Y1,Y2)	箱グラフの内部領域を塗る.
UHBXA (N,X1,X2,Y)	箱グラフの内部領域を塗る.
UVBXL (N,X,Y)	階段状のグラフを描く.
UHBXL (N,X,Y)	階段状のグラフを描く.
UVBXFZ(N,X,Y1,Y2,ITYPE,INDEX)	属性を指定して箱グラフの枠を描く.
UHBXFZ(N,X1,X2,Y,ITYPE,INDEX)	属性を指定して箱グラフの枠を描く.
UVBXAZ(N,X,Y1,Y2,ITP1,ITP2)	属性を指定して箱グラフの内部領域を塗る.
UHBXAZ(N,X1,X2,Y,ITP1,ITP2)	属性を指定して箱グラフの内部領域を塗る.
UVBXLZ(N,X,Y,ITYPE,INDEX)	属性を指定して階段状のグラフを描く.
UHBXLZ(N,X,Y,ITYPE,INDEX)	属性を指定して階段状のグラフを描く.

パラメタ設定

UUSLNT(ITYPE)	折れ線のラインタイプを指定する.
	UULIN,U[VH]BRL,U[VH]BXL 共通.
UUSLNI(INDEX)	折れ線のラインインデクスを指定する.
	UULIN,U[VH]BRL,U[VH]BXL 共通.
UUSMKT(ITYPE)	マーカータイプを指定する.
UUSMKI(INDEX)	マーカーのラインインデクスを指定する.
UUSMKS(RSIZE)	マーカーの大きさを指定する.
UUSEBT(ITYPE)	エラーバーのラインタイプを指定する.
UUSEBI(INDEX)	エラーバーのラインインデクスを指定する.
UUSEBS(RSIZE)	エラーバーの横幅を指定する.
UUSBRS(RSIZE)	棒グラフの横幅を指定する.
UUSARP(ITP1,ITP2)	領域を塗るトーンパターンをを指定する.
	U[VH]DIF,U[VH]BRA,U[VH]BXA 共通.
UUSFRT(ITYPE)	枠のラインタイプを指定する.
	U[VH]BRF,U[VH]BXF 共通.
UUSFRI(INDEX)	枠のラインインデクスを指定する.
	U[VH]BRF,U[VH]BXF 共通.
UUSIDV(UMIN,UMAX)	独立変数の定義域を指定する.

6.4 サブルーチンの説明: 折れ線

6.4.1 UULIN/UULINZ

1. 機能

   折れ線を描く.

2. 呼び出し方法

```
CALL UULIN (N,X,Y)
CALL UULINZ(N,X,Y,ITYPE,INDEX)
```

3. パラメーターの説明

N	(I)	配列 X, Y の長さ.
X, Y	(R(N))	折れ線の座標値.
ITYPE	(I)	折れ線のラインタイプ.
INDEX	(I)	折れ線のラインインデクス.

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N-1 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) UULIN が描く折れ線の属性は UUSLNT, UUSLNI で指定する.

6.5 サブルーチンの説明: マーカー

6.5.1 UUMRK/UUMRKZ

1. 機能

   マーカー列を描く.

2. 呼び出し方法

```
CALL UUMRK (N,X,Y)
CALL UUMRKZ(N,X,Y,ITYPE,INDEX,RSIZE)
```

3. パラメーターの説明

N	(I)	配列 X, Y の長さ.
X, Y	(R(N))	マーカーの座標値.
ITYPE	(I)	マーカーのタイプ.
INDEX	(I)	マーカーのラインインデクス.
RSIZE	(R)	マーカーの大きさ.

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N-1 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) UUMRK が描くマーカーの属性は UUSMKT, UUSMKI UUSMKS で指定する.

6.6 サブルーチンの説明: エラーバー

6.6.1 UVERB/UVERBZ/UHERB/UHERBZ

1. 機能

   エラーバーを描く.

2. 呼び出し方法

```
CALL UVERB (N,X,Y1,Y2)
CALL UHERB (N,X1,X2,Y)
CALL UVERBZ(N,X,Y1,Y2,ITYPE,INDEX,RSIZE)
CALL UHERBZ(N,X1,X2,Y,ITYPE,INDEX,RSIZE)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N))	独立変数.
X1,X2,Y1,Y2	(R(N))	エラーバーの両端の座標値.
ITYPE	(I)	エラーバーのラインタイプ.
INDEX	(I)	エラーバーのラインインデクス.
RSIZE	(R)	エラーバーの幅.

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N-1 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) X1, Y1 または X2, Y2 に RUNDEF を指定すると, 内部変数 'UREF' の値が指定されたものとみなす. 'UREF' の初期値は 0 である.

   (c) UVERB, UHERB が描くエラーバーの属性は UUSEBT, UUSEBI, USSEBS で指定する.

6.7 サブルーチンの説明: 差分

6.7.1 UVDIF/UVDIFZ/UHDIF/UHDIFZ

1. 機能

   2 本の折れ線の間をトーンパターンで塗る.

2. 呼び出し方法

```
CALL UVDIF (N,X,Y1,Y2)
CALL UHDIF (N,X1,X2,Y)
CALL UVDIFZ(N,X,Y1,Y2,ITP1,ITP2)
CALL UHDIFZ(N,X1,X2,Y,ITP1,ITP2)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N))	独立変数.
X1,X2,Y1,Y2	(R(N))	従属変数.
ITP1,ITP2	(I)	トーンパターン番号. X1 < X2 または Y1 < Y2 の時 ITP1, X1 > X2 または Y1 > Y2 の時 ITP2 が使われる.

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N-1 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) X1, Y1 または X2, Y2 に RUNDEF を指定すると, 内部変数 'UREF' の値が指定されたものとみなす. 'UREF' の初期値は 0 である.

   (c) UVDIF, UHDIF が塗るトーンパターンは UUSARP で指定する.

6.8 サブルーチンの説明: 棒グラフ

6.8.1 UVBRF/UVBRFZ/UHBRF/UHBRFZ

1. 機能

   棒グラフの枠を描く.

2. 呼び出し方法

```
CALL UVBRF (N,X,Y1,Y2)
CALL UHBRF (N,X1,X2,Y)
CALL UVBRFZ(N,X,Y1,Y2,ITYPE,INDEX,RSIZE)
CALL UHBRFZ(N,X1,X2,Y,ITYPE,INDEX,RSIZE)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N))	独立変数.
X1,X2,Y1,Y2	(R)	棒の両端の座標値.
ITYPE	(I)	枠のラインタイプ.
INDEX	(I)	枠のラインインデクス.
RSIZE	(R)	棒の幅

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N-1 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) X1, Y1 または X2, Y2 に RUNDEF を指定すると, 内部変数 'UREF' の値が指定されたものとみなす. 'UREF' の初期値は 0 である.

   (c) UVBRF, UHBRF が描く枠の属性は UUSFRT, UUSFRI で指定する. 棒の幅は UUSBRS で指定する.

6.8.2 UVBRA/UVBRAZ/UHBRA/UHBRAZ

1. 機能

   棒グラフの内部領域を塗る.

2. 呼び出し方法

```
CALL UVBRA (N,X,Y1,Y2)
CALL UHBRA (N,X1,X2,Y)
CALL UVBRAZ(N,X,Y1,Y2,ITP1,ITP2,RSIZE)
CALL UHBRAZ(N,X1,X2,Y,ITP1,ITP2,RSIZE)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N))	独立変数.
X1,X2,Y1,Y2	(R(N))	棒の両端の座標値.
ITP1,ITP2	(I)	トーンパターン番号. X1 < X2 または Y1 < Y2 の時 ITP1, X1 > X2 または Y1 > Y2 の時 ITP2 が使われる.
RSIZE	(R)	棒の幅

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N-1 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) X1, Y1 または X2, Y2 に RUNDEF を指定すると, 内部変数 'UREF' の値が指定されたものとみなす. 'UREF' の初期値は 0 である.

   (c) UVBRA, UHBRA が塗るトーンパターンは UUSARP で指定する. 棒の幅は UUSBRS で指定する.

6.8.3 UVBRL/UVBRLZ/UHBRL/UHBRLZ

1. 機能

   棒グラフを線で結ぶ.

2. 呼び出し方法

```
CALL UVBRL (N,X,YD)
CALL UHBRL (N,XD,Y)
CALL UVBRLZ(N,X,YD,ITYPE,INDEX,RSIZE)
CALL UHBRLZ(N,XD,Y,ITYPE,INDEX,RSIZE)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N))	独立変数
XD,YD	(R(N))	棒の端の値.
ITYPE	(I)	ラインタイプ.
INDEX	(I)	ラインインデクス.
RSIZE	(R)	棒の幅

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N-1 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) X1, Y1 または X2, Y2 に RUNDEF を指定すると, 内部変数 'UREF' の値が指定されたものとみなす. 'UREF' の初期値は 0 である.

   (c) UVBRF, UHBRF が描く線の属性は UUSFRT, UUSFRI で指定する. 棒の幅は UUSBRS で指定する.

6.9 サブルーチンの説明: 箱グラフ

6.9.1 UVBXF/UVBXFZ/UHBXF/UHBXFZ

1. 機能

   箱グラフの枠を描く.

2. 呼び出し方法

```
CALL UVBXF (N,X,Y1,Y2)
CALL UHBXF (N,X1,X2,Y)
CALL UVBXFZ(N,X,Y1,Y2,ITYPE,INDEX)
CALL UHBXFZ(N,X1,X2,Y,ITYPE,INDEX)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N+1))	独立変数.
X1,X2,Y1,Y2	(R(N))	箱の両端の座標値.
ITYPE	(I)	枠のラインタイプ.
INDEX	(I)	枠のラインインデクス.

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) X1, Y1 または X2, Y2 に RUNDEF を指定すると, 内部変数 'UREF' の値が指定されたものとみなす. 'UREF' の初期値は 0 である.

   (c) UVBXF, UHBXF が描く枠の属性は UUSFRT, UUSFRI で指定する.

6.9.2 UVBXA/UVBXAZ/UHBXA/UHBXAZ

1. 機能

   箱グラフの内部領域を塗る.

2. 呼び出し方法

```
CALL UVBXA (N,X,Y1,Y2)
CALL UHBXA (N,X1,X2,Y)
CALL UVBXAZ(N,X,Y1,Y2,ITP1,ITP2)
CALL UHBXAZ(N,X1,X2,Y,ITP1,ITP2)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N+1))	独立変数.
X1,X2,Y1,Y2	(R(N))	箱の両端の座標値.
ITP1,ITP2	(I)	トーンパターン番号. X1 < X2 または Y1 < Y2 の時 ITP1, X1 > X2 または Y1 > Y2 の時 ITP2 が使われる.

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) X1, Y1 または X2, Y2 に RUNDEF を指定すると, 内部変数 'UREF' の値が指定されたものとみなす. 'UREF' の初期値は 0 である.

   (c) UVBXA, UHBXA が塗るトーンパターンは UUSARP で指定する.

6.9.3 UVBXL/UVBXLZ/UHBXL/UHBXLZ

1. 機能

   階段状のグラフを描く.

2. 呼び出し方法

```
CALL UVBXL (N,X,YD)
CALL UHBXL (N,XD,Y)
CALL UVBXLZ(N,X,YD,ITYPE,INDEX)
CALL UHBXLZ(N,XD,Y,ITYPE,INDEX)
```

3. パラメーターの説明

N	(I)	配列の長さ.
X, Y	(R(N+1))	独立変数.
XD,YD	(R(N))	従属変数.
ITYPE	(I)	ラインタイプ.
INDEX	(I)	ラインインデクス.
RSIZE	(R)	箱の幅

4. 備考

   (a) X または Y に RUNDEF を指定すると, UUSIDV で指定した定義域を N 等分した点が指定されたものとみなす. 定義域が指定されていなければ, ウインドウ全体を定義域とする.

   (b) UVBXF, UHBXF が描く線の属性は UUSFRT, UUSFRI で指定する.

6.10 サブルーチンの説明: パラメタ設定

6.10.1 UUSLNT/UUSLNI

1. 機能

   折れ線の属性を指定する.

2. 呼び出し方法

```
CALL UUSLNT(ITYPE)
CALL UUSLNI(INDEX)
```

3. パラメーターの説明

ITYPE	(I)	折れ線のラインタイプ.
INDEX	(I)	折れ線のラインインデクス.

4. 備考

   (a) これらのルーチンで設定する属性は UULIN,U[VH]BRL,U[VH]BXL で使われる.

6.10.2 UUSMKT/UUSMKI/UUSMKS

1. 機能

   マーカーの属性を指定する.

2. 呼び出し方法

```
CALL UUSMKT(ITYPE)
CALL UUSMKI(INDEX)
CALL UUSMKS(RSIZE)
```

3. パラメーターの説明

ITYPE	(I)	マーカーのラインタイプ.
INDEX	(I)	マーカーのラインインデクス.
RSIZE	(R)	マーカーの大きさ.

4. 備考

   (a) なし

6.10.3 UUSERT/UUERI/UUSERS

1. 機能

   エラーバーの属性を指定する.

2. 呼び出し方法

```
UUSEBT(ITYPE)
UUSEBI(INDEX)
UUSEBS(RSIZE)
```

3. パラメーターの説明

   ITYPE (I) エラーバーのラインタイプ.
   INDEX (I) エラーバーのラインインデクス.
   RSIZE (R) エラーバーの横幅.

4. 備考

   (a) なし

6.10.4 UUSBRS

1. 機能

   棒グラフの横巾を指定する.

2. 呼び出し方法

```
UUSBRS(RSIZE)
```

3. パラメーターの説明

   RSIZE (R) 棒グラフの横幅.

4. 備考

   (a) なし

6.10.5 UUSARP

1. 機能

   内部領域を塗るトーンパターンを指定する.

2. 呼び出し方法

```
UUSARP(ITP1,ITP2)
```

3. パラメーターの説明

   ITP1, ITP2 (I) トーンパターン番号.

4. 備考

   (a) ここで設定する番号は U[VH]DIF,U[VH]BRA,U[VH]BXA で使われる.

6.10.6 UUSFRT/UUSFRI

1. 機能

   枠の属性を指定する.

2. 呼び出し方法

```
CALL UUSFRT(ITYPE)
CALL UUSFRI(INDEX)
```

3. パラメーターの説明

   ITYPE (I) 枠のラインタイプ.
   INDEX (I) 枠のラインインデクス.

4. 備考

   (a) これらのルーチンで設定する属性は U[VH]BRF,U[VH]BXF で使われる

6.10.7 UUSIDV

1. 機能

   独立変数の定義域を指定する.

2. 呼び出し方法

```
CALL UUSSIDV(UMIN, UMAX)
```

3. パラメーターの説明

   UMIN, UMAX (R) 定義域の最大最小値.

4. 備考

   (a) なし.

第 7 章

USPACK ：オートスケーリングルーチン

7.1 概要

USPACK はデータを簡単にグラフ化することを主目的としたいわゆるオート ・スケーリング ・ルーチンのパッケージである. このパッケージの主な機能は,

1. データの最大最小値を調べて, 適切な正規化変換パラメタを求める.
2. 現在設定されている正規化変換に対して, 適切な目盛及びラベル間隔を計算して座標軸を描く.

ことである. 2 番目の座標軸を描くルーチンは, 独立に呼ぶことができるので, 簡単に座標軸を描くユーティリティーをしても使える. コンターを描く UDPACK などと組み合せて使うと便利である.

なお, 実際の折れ線グラフは UUPACK, 座標軸やタイトルは UXPACK/UYPACK などに依存している.

7.2 基本構造

USPACK は与えられたデータに対して, 適当な正規化変換のパラメタを求める 「スケーリング ・ルーチン」と, 設定された正規化変換に対して, 適当な目盛り間隔などを求めて座標軸を描く 「座標軸ルーチン」がある. スケーリング ・ルーチンと座標軸ルーチンは基本的に独立に動作し, どちらか一方だけを使うことができる.
さらに, これらを組み合わせて簡便にグラフを描くための 「簡単ルーチン (USGRPH)」からなる.

7.2.1 簡単描画ルーチン (USGRPH)

とにかく簡単にデータをグラフ化するためのルーチンで, 1 行で折れ線グラフまで描いてくれる. これらを使ったプログラムは次のようになる.

```
1         CALL GROPN(IWS)    ! デバイスオープン
2         CALL GRFRM         ! フレームを用意
3         CALL GRSTRN(ITR)   ! 変換関数番号の指定（省略可, 省略値 1）
4
5         CALL USSTTL(....)  ! タイトル設定（省略可）
6         CALL USGRPH(N,X,Y) ! グラフ描画
7
8         CALL GRCLS         ! デバイスのクローズ
```

GRFRM と USGRPH の間で, 変換関数に関するパラメタを GRPACK や SGPACK で設定したり, その他のパラメタを USpSET により設定することができる. また, USSTTL により座標軸のタイトルなどを指定することもできる.

これらのルーチンは, とにかく簡便にデータをグラフ化するには便利である. しかし, 複数の折れ線を一つの図に入れたり, 他のパッケージと組み合わせて使うなどの応用をするにはスケーリング ・ルーチンと座標軸ルーチンを直接使う方がよい. 上記のプログラムをこれらの基本的なルーチンを使って書き直すと次のようになる.

```
1           CALL GROPN(IWS)     ! デバイスをオープン
2           CALL GRFRM          ! フレームを用意
3           CALL GRSTRN(ITR)    ! 変換関数番号の指定（省略可）
4
5           CALL USSPNT(N,X,Y) ! データ範囲の指定
6           CALL USPFIT         ! 変換関数パラメタ設定
7           CALL GRSTRF         ! 変換関数確定
8
9           CALL USSTTL(....)   ! タイトル設定（省略可）
10          CALL USDAXS         ! 座標軸描画
11
12          CALL UULIN (N,X,Y) ! 折れ線描画
13
14          CALL GRCLS          ! デバイスのクローズ
```

7.2.2 スケーリング ・ルーチン

スケーリング ・ルーチンは与えられたデータから適当な正規化変換のパラメタを求めるもので, GRSWND, GRSVPT の代わりをするルーチンである. 変換関数の確定は行わないので, GRSTRF により関数の確定をする必要がある.

正規化変換のパラメタを決めるためには, これから描こうとする折れ線などのデータ範囲を知らなければならない. USSPNT はそのデータを指定するもので, USPFIT を呼ぶ前に何回でも呼ぶことができる. 例えば複数の折れ線を描くためには, それらのデータをすべて与えなければならない. USPFIT はこれらのデータの最大最小値を切りの良い数値に丸めてウインドウを決定する. なお, GRSWND/SGSWND または SGpGET/SGpSET などにより, ウインドウをユーザーが陽に指定した場合は, これらの値が優先される.

ビューポートは, USPACK が描く座標軸関係の文字がすべて描けるように, 最大作画領域から文字の大きさ (UZpGET/UZpSET の管理する内部変数'RSIZEL1') に比例した幅のマージンをとった大きさに設定される. したがって, UZFACT 等によりラベルの文字の大きさを変えたり, SGpGET/SGpSET の管理する内部変数 'LFULL' を.TRUE. にしたりすると, グラフの大きさも変る. また, ビューポートも GRSVPT/SGSVPT または SGpSET, SGpSTX により陽に指定されている場合は, これらの値が優先される.

7.2.3 座標軸ルーチン

USDAXS は現在設定されている変換関数に対して, 目盛間隔やラベルの間隔を適切に設定して座標軸を描くものである. 座標軸を描く位置は USCSET の内部変数 CXSIDE/CYSIDE で指定され, タイトルなどは USSTTL または USCSET で指定することができる. なお, USDAXS は基本的に X–軸と Y–軸を独立に処理しており, しかも, ラベル間隔の決めるときにラベルと軸の角度 (平行か垂直) により異なった処理を行う. 通常は X–軸, y–軸ともにラベルを水平 (x–軸と並行) にするようになっているので, X–軸と Y–軸のデータ範囲が同じであっても, ラベル間隔が異なることがある. それが気持ち悪いときには, USLSET で'LMATCH' を .TRUE. に

すると, 実際のラベルの向きに関わらず, ラベルが軸と平行に書かれるものと仮定してラベル間隔を求めるようになるので, X–軸と Y–軸のラベル間隔を揃えることができる.

USDAXS は X–軸と Y–軸の両方を描くが, 片方ずつ描画する USXAXS, USYAXS もある.

7.2.4 内部変数管理

USPACK 固有のパラメタは USpGET/USpSET によって管理されているが, タイトルをまとめて指定するルーチン USSTTL も用意されている. このうち USINIT は, GRFRM GRFIG から呼ばれるので, 通常ユーザーが呼ぶ必要はない.

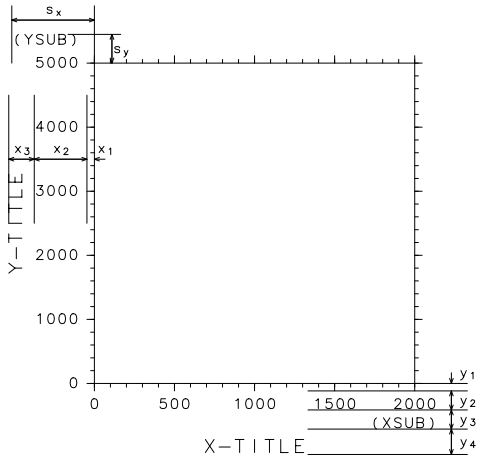
なお, USPACK は UUPACK や UXPACK/UYPACK のルーチンを使って座標軸や折れ線を描くので, これらのパッケージの内部変数に影響される. 各制御項目とそれを制御するルーチン名及び内部変数名は以下のとおり.

制御項目	設定ルーチン	内部変数名
座標軸の最大最小値	* SGpSET/SGSWND	UXMIN/UXMAX/UYMIN/UYMAX
ビューポート	* SGpSET/SGSVPT	VXMIN/VXMAX/VYMIN/VYMAX
座標軸の向き	USLSET	LXINV/LYINV
オフセット値	USRSET	XOFF/YOFF
ファクター値	USRSET	XFAC/YFAC
目盛間隔	USRSET	DXT/DYT
座標軸の位置	USCSET	CXSIDE/CYSIDE
目盛の向き	*UZISET	INNER
ラベル間隔	USRSET	DXL/DYL
ラベルの向き	*UZISET	IROTLzs
ラベルのセンタリングオプション	*UZISET	ICENTzs
タイトル文字	USCSET/USSTTL	CXTTL/CYTTL
サブラベルの単位	USCSET	CXUNIT/CYUNIT
サブラベルの括弧	USCSET	CBLKT
サブラベルの位置	USCSET	CXSPOS/CYSPOS
タイトル, サブラベル, ラベルのフラグ	*UZLSET	LABELzs
文字の大きさ	*UZFACT	引数
折れ線の属性	*UUSLNI/UUSLNT	引数

7.3 USPACK による作画範囲

USPACK による最大作画範囲を下の図に示す. これらの範囲は内部変数によって変化するが, 初期状態では目盛を内側に描く $(x_1 = y_1 = 0)$ ので, 左マージンが $x_2 + x_3 = 0.1463$, 右マージンが $\mathtt{RSIXEL1} \times 2 = 0.042$, 上マージンが $s_y = 0.0537$, 下マージンが $y_2 + y_3 + y_4 = 0.119$ あればタイトルまで含めて全て作画範囲に入る.

なお初期状態で上下左右とも 0.2 のマージンを取る.



uscnf.f:page1

$$
\begin{aligned}
x_1 &= \mathtt{RSIZET2} = 0.014 \\
x_2 &= (\mathtt{MXDGTY} + \mathtt{PAD1}) \times \mathtt{RSIZEL1} = 0.0987 \\
x_3 &= (1 + \mathtt{PAD1}) \times \mathtt{RSIZEC1} = 0.0476
\end{aligned}
$$

$$
\begin{aligned}
y_1 &= \mathtt{RSIZET2} = 0.014 \\
y_2 &= (1 + \mathtt{PAD1}) \times \mathtt{RSIZEL1} = 0.0357 \\
y_3 &= (1 + \mathtt{PAD1}) \times \mathtt{RSIZEL1} = 0.0357 \\
y_4 &= (1 + \mathtt{PAD1}) \times \mathtt{RSIZEC1} = 0.0476
\end{aligned}
$$

$$
\begin{aligned}
s_x &= \mathtt{RSIZET2} + (\mathtt{MXDGTSY} + \mathtt{PAD1}) \times \mathtt{RSIZEL1} = 0.1547 \\
s_y &= \mathtt{SOFFYLT} + (1 + \mathtt{PAD1}) \times \mathtt{RSIZEL1} = 0.0537
\end{aligned}
$$

7.4 サブルーチンのリスト

簡単ルーチン

USGRPH(N,X,Y)	X 座標値, Y 座標値を与えて, 折れ線グラフを描く.

スケーリング ・ルーチン

USSPNT(N,X,Y)	ウインドウの範囲に含めたいデータを指定する.
USPFIT	変換パラメータの決定.

座標軸ルーチン

USDAXS	座標軸描画.
USXAXS(CXSIDE)	X 座標軸を描画する.
USYAXS(CYSIDE)	Y 座標軸を描画する.
USSTTL(CXTTL, CXUNIT, CYTTL, CYUNIT)	タイトル設定

パラメタ管理

USpGET(CP, IPARA)	内部変数参照.
USpSET(CP, IPARA)	内部変数設定.
USpSTX(CP, IPARA)	内部変数設定 (実行時オプションによる変更可).
USINIT	内部変数初期化

7.5 サブルーチンの説明 (簡単ルーチン)

7.5.1 USGRPH

1. 機能

   適当なスケーリングを行って座標軸とグラフを描く.

2. 呼び出し方法

```
CALL USGRPH(N, X, Y)
```

3. 引数の説明

N	(I)	X,Y のデータ数. (i)
X	(R(N))	折れ線グラフの X 座標 (i)
Y	(R(N))	折れ線グラフの Y 座標 (i)

4. 備考

   (a) このルーチンは USSXEX, USSYEX, USAXIS と UULIN を呼ぶ.

   (b) X または Y に RUNDEF を指定した時は等間隔のデータが指定されたものとみなす.

7.6 サブルーチンの説明 (スケーリング ・ルーチン)

7.6.1 USSPNT

1. 機能

   ウインドウ範囲に含めたいデータを指定する.

2. 呼び出し方法

```
CALL USSPNT(N, X, Y)
```

3. 引数の説明

N	(I)	X,Y のデータ数. (i)
X	(R(N))	X 座標 (i)
Y	(R(N))	Y 座標 (i)

4. 備考

   (a) USSTRF を呼ぶ前に何度でも呼ぶことができる. (複数の折れ線グラフを描く場合など.)

   (b) X または Y に RUNDEF を指定した時には, その座標値に関しては何もしない.

7.6.2 USPFIT

1. 機能

   USSPNT で設定されたデータの最大最小値を, きりのよい数値に丸めて, 適当な正規化変換のパラメタ (ウインドウとビューポート) を決める.

2. 呼び出し方法

```
CALL USPFIT
```

3. 備考

   (a) このルーチンは正規化変換のパラメタを設定するだけで, 実際の変換関数の確定は行わないので, USPFIT の後に GRSTRF を呼んで変換関数を確定する必要がある.

   (b) SGSVPT または SGRSET によりビューポートが指定されていない場合には, ビューポート (グラフの描画範囲) は WS ウインドウから, ラベルの文字の大きさ RSIZEL1 に比例する幅のマージンをとって設定される.

   (c) GRSWND/SGSWND または SGpSET, SGpSTX により座標軸の最大最小値があらかじめ設定されているときには, USSPNT により設定されている値は無視され, 丸めも行われない.

7.7 サブルーチンの説明 (座標軸ルーチン)

7.7.1 USDAXS

1. 機能

   デフォルトの座標軸を描く. タイトルが指定されていればタイトルも書く.

2. 呼び出し方法

```
CALL USDAXS
```

3. 備考

   (a) 座標軸を描く位置は, USCSET の内部変数 CXSIDE/CYSIDE により指定される. 初期値は'BT' と'LR' である.

   (b) タイトルは CXSIDE/CYSIDE の 1 文字目に指定された場所に書かれる.

   (c) タイトル及び単位は USSTTL または USCSET で指定する.

7.7.2 USXAXS/USYAXS

1. 機能

   現在設定されている正規化変換に対して適当な目盛とラベル間隔を計算し, 指定された場所に座標軸を一本描く.

2. 呼び出し方法

```
CALL USXAXS(CXSIDE)
CALL USYAXS(CYSIDE)
```

3. 引数の説明

```
CXSIDE  (C*1)   座標軸の位置 ('B'/'T'/'U'). (i)
CYSIDE  (C*1)   座標軸の位置 ('L'/'R'/'U'). (i)
```

4. 備考

   (a) 設定されている正規化変換番号に従って, 一様座標軸か対数座標軸を描く.

   (b) 一様座標軸の場合, ウインドの最大最小値が USPFIT により丸めてあれば, 座標軸の両端にほぼ間違いなく目盛がつく.

   (c) タイトルは書かれないので, U[XY]STTL などを使って書くこと.

7.7.3 USSTTL

1. 機能

   USDAXS で書く座標軸のタイトル及びサブラベルの中の単位を指定する.

2. 呼び出し方法

```
CALL USSTTL(CXTTL, CXUNIT, CYTTL, CYUNIT)
```

3. 引数の説明

CXTTL	(C*(*))	X 座標軸のタイトル (i)
CXUNIT	(C*(*))	X 座標軸の単位 (i)
CYTTL	(C*(*))	Y 座標軸のタイトル (i)
CYUNIT	(C*(*))	Y 座標軸の単位 (i)

4. 備考

(a) このルーチンは USCSTX を 4 回呼んで, 内部変数 CXTTL, CXUNIT, CYTTL, CYUNIT を設定するのと等価である.

(b) タイトルを付けたくないときには, 1 文字以上のブランクを指定する.

7.8 サブルーチンの説明 (パラメータ管理)

7.8.1 USpGET/USpSET(USpSTX)

1. 機能

   実数, 整数, 論理内部変数を参照/設定する. (USpSTX は実行時オプションによる変更が可能.)

2. 呼び出し方法

```
CALL USpGET(CP, IPARA)
CALL USpSET(CP, IPARA)
CALL USpSTX(CP, IPARA)
```

3. 引数

CP	(C*(*))	パラメータ名 (i)
IPARA	(I/R/L)	パラメータ値 (i/o)

4. パラメータリスト

IRESET	(I)	USINIT で自動的にパラメータをリセットすることを指定する変数. 初期値は 1. 0: スケーリングパラメタなど (* 印のついた変数) のリセットのみ行なう. 1: IRESET=0 に加えてタイトルのリセットをする. 2: USpSET のすべてのパラメタをリセットする. USINIT は通常 GRFRM から呼ばれる.
XDTMIN*, XDTMAX*	(R)	USSPNT で設定された X 座標の最大最小値.
YDTMIN*, YDTMAX*	(R)	USSPNT で設定された Y 座標の最大最小値.
LXINV	(L)	.TRUE. の時, X 軸の左右を逆にする.
LYINV	(L)	.TRUE. の時, Y 軸の上下を逆にする.
LMATCH	(L)	.TRUE. の時, X 軸と Y 軸のラベル間隔及び文字間隔を同じにする.
RMRGN	(R)	マージンの幅. 単位は文字幅. (9.524)
XFAC*	(R)	X 座標のファクター値 (999.)
YFAC*	(R)	Y 座標のファクター値 (999.)
XOFF*	(R)	一様 X 座標のオフセット値 (999.)
YOFF*	(R)	一様 Y 座標のオフセット値 (999.)
DXT*	(R)	一様 X 座標の目盛の間隔 (999.)
DYT*	(R)	一様 Y 座標の目盛の間隔 (999.)
DXL*	(R)	一様 X 座標のラベルの間隔 (999.)
DYL*	(R)	一様 Y 座標のラベルの間隔 (999.)

TFACT	(R)	一様座標の目盛間隔の目安 (2) これらの値に文字幅 (RSIZEL1) を掛けた幅よりも小さな間隔で, 切りのよい値を DUT として設定する.
MXDGTX	(I)	一様 X 座標のラベルの最大桁数 (4)
MXDGTY	(I)	一様 Y 座標のラベルの最大桁数 (4) ラベルの桁数がこれらの値よりも大きくなるとき, UOFF, UFAC を設定してこれらの桁数以内におさまるように試みる.
NBLANK1	(I)	ラベルが座標軸と平行な場合の, ラベル間の最小空白数. (一様座標) (1)
NBLANK2	(I)	ラベルが座標軸と直角な場合の, ラベル間の最小空白数. (一様座標) (2)
NLBLX	(I)	対数 X 座標のラベル形式 (999)
NLBLY	(I)	対数 Y 座標のラベル形式 (999)
NTICKSX	(I)	対数 X 座標の一桁の間の目盛の数 (999)
NTICKSY	(I)	対数 Y 座標の一桁の間の目盛の数 (999)
ITYPEX	(I)	対数 X 座標のタイプ (999)
ITYPEY	(I)	対数 Y 座標のタイプ (999)
MXDGTSX	(I)	X 軸のサブラベルが水平方向の場合, MXSDGTX で指定された文字数のラベルの範囲を越えないようにサブラベルを描く.
MXDGTSY	(I)	Y 軸のサブラベルが垂直方向の場合, MXDGTSY で指定された文字数のラベルの範囲を越えないようにサブラベルを描く.
LPRTCT	(L)	サブラベルを他方の軸のラベル領域にまで描かないことを指定する. これが .TRUE. の時には MXDGTSX/MXDGTSY の指定は無視される.
SOFFX*sp**	(R)	X 軸のサブラベルのオフセット値. *sp*='TR', 'TL', 'BR', 'BL', 'UR', 'UL'.
SOFFY*sp**	(R)	X 軸のサブラベルのオフセット値. *sp*='RT', 'RB', 'LT', 'LB', 'UT', 'UB'.
ROFFX*s**	(R)	Y 軸のサブラベルが X 軸のラベル領域に侵入したときの X 軸のオフセット値. *s*= 'T', 'B'.
ROFFY*s**	(R)	X 軸のサブラベルが Y 軸のラベル領域に侵入したときの Y 軸のオフセット値. *s*= 'L', 'R'.

* がついた変数は常にリセットされる.

5. 備考

(a) 内部変数を管理するための下請けルーチンとして以下のものがある.

USPQNP(NCP)	内部変数の総数 NCP を求める.
USPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
USPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
USPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
USPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) US*p*GET は上に述べた USPQID を呼んで内部変数の位置を求め, USPQVL によって値を参照する; US*p*SET は USPQID を呼んで内部変数の位置を求め, USPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは USPQID が出力する.

(c) ITYPE としては適切な型の定数または変数を指定すること.

7.8.2 USCGET/USCSET(USCSTX)

1. 機能

   文字パラメータを参照/設定する. (USCSTX は実行時オプション による変更可能.)

2. 呼び出し方法

```
CALL USCGET(CP, CPARA)
CALL USCSET(CP, CPARA)
CALL USCSTX(CP, CPARA)
```

3. 引数

CP	(C*(*))	パラメータ名 (i)
CPARA	(C*(*))	パラメータ値 (i/o)

4. パラメータリスト

CXTTL	X 軸のタイトル (32 文字以内)
CYTTL	Y 軸のタイトル (32 文字以内)
CXSIDE	X 軸を書く位置. 'T'(Top), 'B'(Bottom), 'U'(User) のうち 2 文字まで指定できる. 2 つの位置のうち最初に指定した位置にサブラベルとタイトルが書かれる. (注 1)
CYSIDE	Y を描く位置. 'R'(Right), 'L'(Left), 'U'(User) のうち 2 文字まで指定できる. 2 つの位置のうち最初に指定した位置にサブラベルとタイトルが書かれる. (注 1)
CXSPOS	X 軸のサブラベルを書く位置 'R' または 'L'
CYSPOS	Y 軸のサブラベルを書く位置 'T' または 'B'
CXUNIT	X 軸のサブラベルに書く単位 (32 文字以内)
CYUNIT	Y 軸のサブラベルに書く単位 (32 文字以内)
CBLKT	サブラベルに書く括弧. 初期値は'()'. ブランクを指定すると括弧は書かない.

   (注 1) 内部変数 CXSIDE, CYSIDE で指定される最初の位置の LABELzs が.FALSE. の時には, サブラベル, タイトルともに描かない.

5. 備考

   (a) 内部変数を管理するための下請けルーチンとして以下のものがある.

USCQNP(NCP)	内部変数の総数 NCP を求める.
USCQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
USCQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
USCQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
USCSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) USCGET は上に述べた USCQID を呼んで内部変数の位置を求め, USCQVL によって値を参照する; USCSET は USCQID を呼んで内部変数の位置を求め, USCSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは USCQID が出力する.

(c) ITYPE としては適切な型の定数または変数を指定すること.

7.8.3 USINIT

1. 機能

   サブラベルに関するパラメータ (SOFFX, SOFFY, ROFFX, ROFFY) を初期値に戻す. さらに US*p*GET/US*p*SET の管理する内部変数 IRESET の値に応じて, その他の内部変数を初期化する.

2. 呼び出し方法

```
CALL USINIT
```

3. 引数

   なし.

4. 備考

   (a) 通常このルーチンは GRFRM GRFIG から呼ばれるので, ユーザーが直接呼ぶ必要はない.

第 8 章

UDPACK：等高線ルーチン

8.1 概要

UDPACK は 2 次元等高線図を作画するサブルーチンパッケージである. 等高線は U 座標系で作画されるので, すべての正規化変換に対応している.

このパッケージにおいてラベルつきのコンターラインは, GRPH1 の SGpGET/SGpSET が管理する内部変数 'LCHAR' を .TRUE. として, 線分を描く下位ルーチンが描いている. なおこのパッケージでは, 等高線図を描くだけで座標軸などの外枠は描かない.

UDPACK においてユーザーが使用するサブルーチンとしては以下のようなものがある.

- 作画ルーチン

  UDCNTR ··· 2 次元等高線図を作画する (内部で用意した作業領域を使う).

  UDCNTZ ··· 2 次元等高線図を作画する (ユーザーが作業領域を与える).

- コンターレベルに関するルーチン

  UDGCLA ··· 最小値 ・最大値 ・きざみ値で生成する.

  UDGCLB ··· 格子点値を与える配列ときざみ値で生成する.

  UDICLV ··· コンターレベルを無効にする.

  UDSCLV ··· 1 本のコンターレベルを生成する.

  UDDCLV ··· 1 本のコンターレベルを削除する.

- その他のルーチン

  UDSFMT ··· コンターレベルのフォーマットを指定する.

  UDpGET/UDpSET··· 内部変数を参照/変更する.

格子点座標に関する情報は UWPACK (第 11節参照) が管理している.

- 格子点座標の設定に関するルーチン

  UWSGXA/UWSGYA ··· 各座標値で設定する.

UWSGXB/UWSGYB ··· 最小値と最大値で設定する.

2 次元データをとにかくコンタリングしたい (quick look) というときには, 単に UDCNTR (または UDCNTZ) を呼ぶだけでよい. このときは, 現在設定されているウインドウいっぱいに等間隔の格子点を設定し, コンターレベルも自動的に生成する.

もしも格子点が不等間隔の場合は, UDCNTR を呼ぶ前に UWSGXA (X 方向)/ UWSGYA (Y 方向) を呼んで格子点の座標値をあらかじめ指定すればよい. あるいはウインドウのある一部分に等間隔の格子点を設定したい場合は, UWSGXB (X 方向)/ UWSGYB (Y 方向) をもちい最小値と最大値を指定して格子点を設定すればよい.

またコンターレベルをあらかじめ指定したいときは, UDCNTR(または UDCNTZ) を呼ぶ前に UDGCLA または UDGCLB を呼べばよい. ただしこの場合は, 等間隔のコンターレベルしか生成されない. もしも, 不等間隔のコンターレベルを指定したい場合や, 特定のコンターレベルを追加したい場合は, UDSCLV で 1 本 1 本のコンターレベルを生成してやればよい. (1 本のコンターレベルを無効にするには UDDCLV を呼べばよい; すべてのコンターレベルを無効にするには UDICLV を呼べばよい.)

内部的には, UDBLK1, UDBLK2, UDBLK3 という名前の共通ブロックを使用している. ユーザーは同じ名前の外部手続き名・共通ブロック名を用いてはならない.

8.2 サブルーチンの説明

8.2.1 UDCNTR

1. 機能

   2 次元等高線図を描く (内部で用意した作業領域を使う).

2. 呼び出し方法

```
CALL UDCNTR(Z,MX,NX,NY)
```

3. パラメーターの説明

Z	(R)	MX×NY の 2 次元配列. 作画には NX×NY の部分を使う.
MX	(I)	配列 Z の第 1 次元整合寸法.
NX	(I)	作画に使う配列 Z の第 1 次元寸法.
NY	(I)	作画に使う配列 Z の第 2 次元寸法.

4. 備考

   (a) 作業配列の大きさの制約から配列 Z の大きさは 1 語のビット長が NB のとき

   $$(\mathtt{NX}+2)\times(\mathtt{NY}+2)\times 2/\mathtt{NB}+1\leq 1000$$

   でなければならない. NB は GL*p*GET/GL*p*SET の管理する内部変数'NBITSPW' を参照する. 普通この値は 32 である (『MATH1」のマニュアル参照). もしもこの制限を越える大きさの配列を用いて作画したいときは, 作業領域をユーザーが指定するサブルーチン UDCNTZ (次節参照) を用いるとよい.

   (b) X 方向の格子点の座標が UWSGXA (11.2.1節参照) または UWSGXB (11.2.2節参照) によって設定されていない場合, SGQWND でウインドウ情報を取得しウインドウの UXMIN, UXMAX を NX−1 等分するような格子点を設定する.

   (c) Y 方向の格子点の座標が UWSGYA (11.2.1節参照) または UWSGYB (11.2.2節参照) によって設定されていない場合, SGQWND でウインドウ情報を取得しウインドウの UYMIN, UYMAX を NY−1 等分するような格子点を設定する.

   (d) コンターを引くレベルの値が UDGCLA, UDGCLB または UDSCLV によって設定されていない場合, 次のサブルーチンを内部で呼んで約 NLEV 本のコンターレベルを設定する.

```
CALL UDGCLB(Z,MX,NX,NY,-REAL(NLEV))
```

   NLEV は UD*p*GET/UD*p*SET の管理する内部変数'NLEV' が参照する値である (初期値は 12). (以下, 特に断らない限り, 単に 「内部変数」というのは UD*p*GET/UD*p*SET の管理する内部変数のことをさす.)

   (e) コンターを引くアルゴリズムの制約から, 格子点の値はどのコンターレベルの値とも等しくてはならない. もしもそのような格子点値があると内部で微小量が修正される (その際メッセージが出力される).

   (f) GL*p*GET/GL*p*SET の管理する内部変数'LMISS' が .TRUE. なら欠損値処理をおこなう. (その際, 参照する欠損値は GL*p*GET/GL*p*SET の管理する内部変数'RMISS' である.) .FALSE. なら欠損値処理をおこなわない. くわしくは 『MATH1」のマニュアル参照.

(g) コンタリングの前にすべての格子点値のチェックをおこなう. 次の 2 つの場合コンタリングはおこなわずその旨メッセージが出力される. さらに内部変数'LMSG' が.TRUE. のときグラフの下部マージンにもメッセージを書く. .FALSE. のときメッセージを書かない(初期値は.TRUE.). 2 つの場合とは:

- すべての格子点が欠損値のとき (missing field).
- すべての格子点が同一な値のとき (constant field).

(h) 内部変数'LMSG' が.TRUE. で引くべきコンターレベルの数が 2 本以上あるとき, 実数型の関数 RUDLEV(1) (8.2.8節参照) を呼んで第 1 コンターレベルの値 (CZL1) と第 2 コンターレベルの値 (CZL2) の差 (CZL2–CZL1) をグラフの下部マージンに書き出す. これらのメッセージは, 地図投影変換以外が設定されているとき, UZPACK が管理する情報を使って書く. したがって, このような場合は, まず座標軸を描いた後で UDCNTR を呼ぶべきである. いっぽう, 地図投影変換が設定されているとき, 下部マージンの真中に書く.

8.2.2 UDCNTZ

1. 機能

   2 次元等高線図を描く (ユーザーが作業領域を与える).

2. 呼び出し方法

```
CALL UDCNTZ(Z,MX,NX,NY,IBR,NBR3)
```

3. パラメーターの説明

Z	(R)	MX×NY の 2 次元配列. 作画には NX×NY の部分を使う.
MX	(I)	配列 Z の第 1 次元整合寸法.
NX	(I)	作画に使う配列 Z の第 1 次元寸法.
NY	(I)	作画に使う配列 Z の第 2 次元寸法.
IBR	(I)	作業領域として用いられる配列.
NBR3	(I)	配列 IBR の長さ.

4. 備考

(a) このサブルーチンを使用する際の注意事項は, 配列の大きさに関する次の注意点を除いて UDCNTR (前節参照) と基本的に同じである.

(b) 作業配列の大きさは次の条件を満たすように決めてやればよい. 1 語のビット長が NB のとき

$$(\mathtt{NX}+2)\times(\mathtt{NY}+2)\times 2/\mathtt{NB}+1 \le \mathtt{NBR3}/3$$

NB は GLpGET/GLpSET の管理する内部変数'NBITSPW' を参照する. ふつうこの値は 32 である (『MATH1』のマニュアル参照).

8.2.3 UDGCLA

1. 機能

   UDCNTR (または UDCNTZ) で使うコンターレベル値を最小値・最大値およびきざみ値またはおよそのレベル数で設定する.

2. 呼び出し方法

```
CALL UDGCLA(XMIN,XMAX,DX)
```

3. パラメーターの説明

XMIN, XMAX	(R)	コンターレベルの最小値と最大値. 必ずしもきりのよい値でなくてもよい.
DX	(R)	DX >0 のとき DX をきざみ幅とする. DX $=0$ のとき内部変数'NLEV' を参照して, 約 NLEV 本のコンターレベルを生成する. DX <0 のとき約 INT(ABS(DX)) 本のコンターレベルを生成する.

4. 備考

(a) XMIN $\neq$ XMAX でなければならない.

(b) コンターレベルは DX が決めるきざみ幅の整数倍になるように設定される.

(c) コンターにつけるラベルは, サブルーチン UDLABL が生成する. このとき使用するフォーマットは, サブルーチン UDQFMT/UDSFMT で参照/変更できる (8.2.9節参照).

(d) コンターには major line と minor line の 2 種類がある. major line とはラベル付きのコンターで, minor line とはラベルなしのコンターである. (ただし, 内部変数'LABEL' を.FALSE. にすれば major line もラベルなしとすることができる. 初期値は.TRUE.) major line のラインインデクスは内部変数'INDXMJ' が決める (初期値は 3). major line に付けるラベルの文字高は内部変数'RSIZEL' が決める (単位は V 座標系の値; 初期値は不定. このとき RSIZEL は UZpGET/UZpSET の管理する内部変数'RSIZEL1' を参照する). minor line のラインインデクスは内部変数'INDXMN' が決める (初期値は 1).

(e) 内部変数'LDASH' が.TRUE. のとき, コンターレベルが負のコンターラインは内部変数'IDASH' を参照してそのラインタイプで描かれる ('LDASH' の初期値は.TRUE., 'IDASH' の初期値は 3: 破線). それ以外のコンターラインは内部変数'ISOLID' を参照してそのラインタイプで描かれる ('ISOLID' の初期値は 1:実線).

(f) minor line を major line の間に何本引くかは内部変数'ICYCLE' (初期値は 2) が決める. すなわち'ICYCLE' で参照する値に 1 本の割合で major line を引き, 残りは minor line を引く. major line のコンターレベルは DX で決まるきざみ幅の'ICYCLE' 倍になるように設定される.

8.2.4 UDGCLB

1. 機能

UDCNTR (または UDCNTZ) で使うコンターレベル値を格子点値を与える配列ときざみ値またはおよそのレベル数で設定する.

2. 呼び出し方法

```
CALL UDGCLB(Z,MX,NX,NY,DX)
```

3. パラメーターの説明

Z, MX, NX, NY	UDCNTR (8.2.1節) 参照.
DX	UDGCLA (8.2.3節) 参照.

4. 備考

   (a) このルーチンは Z の格子点値の最小値 ･最大値をチェックして, 内部で UDGCLA を呼んでいる.

8.2.5 UDSCLV

1. 機能

   コンターラインの属性をすべて指定して 1 本のコンターレベルを設定する.

2. 呼び出し方法

```
CALL UDSCLV(ZLEV,INDX,ITYP,CLV,HL)
```

3. パラメーターの説明

ZLEV	(R)	コンターレベルの値.
INDX	(I)	コンターラインのラインインデクス.
ITYP	(I)	コンターラインのラインタイプ.
CLV	(C*(*))	コンターラインにつけるラベル.
HL	(R)	ラベルの高さ (単位は V 座標系). HL ≤ 0 のときラベルをつけない.

4. 備考

   (a) UDGCLA, UDGCLB は内部でこのルーチンを呼んでいる. 補助線を入れたいときなど UDGCLA, UDGCLB を呼んだ後でこのルーチンを呼べば, 引くべきコンターレベルを 1 本追加できる.

   (b) ラベル (CLV) の長さは 8 以下でなければならない.

   (c) 現在設定されているコンターレベルの総本数は

```
CALL UDQCLN(NLEV)
```

   によって参照できる.

   (d) 第 NL 番目のコンターレベルの属性は

```
CALL UDQCLV(ZLEV,INDX,ITYP,CLV,HL,NL)
```

   によって参照できる. コンターレベルに関する情報は UDSCLV によって順に追加されるので, UDQCLV が参照するコンターレベルの値は必ずしも昇順にならんでいるとは限らない.

8.2.6 UDDCLV

1. 機能

   あるコンターレベルを削除する.

2. 呼び出し方法

```
CALL UDDCLV(ZLEV)
```

3. パラメーターの説明

ZLEV	(R)	削除するコンターレベルの値.

4. 備考

   (a) UDGCLA, UDGCLB によって既に設定されているコンターレベルから特定のコンターレベルを削除したいときこのルーチンを呼べばよい.

8.2.7 UDICLV

1. 機能

   UDGCLA, UDGCLB, UDSCLV で設定したコンターレベルを無効にする.

2. 呼び出し方法

   ```
   CALL UDICLV
   ```

3. パラメーターの説明

   なし

4. 備考

   (a) UDGCLA, UDGCLB, UDSCLV によって既に設定されているコンターレベルを無効にして UDCNTR を呼びたいとき, このルーチンを呼ぶ必要がある.

8.2.8 RUDLEV

1. 機能

   UDGCLA, UDGCLB, UDSCLV で設定したコンターレベルの間隔を求める.

2. 呼び出し方法

   ```
   RUDLEV(NLEV)
   ```

3. パラメーターの説明

NLEV	(I)	何番目のコンター間隔を調べるか指定する.
RUDVAL	(R)	コンター間隔を与える実数型関数値.

4. 備考

   (a) 内部的には UDQCLV (8.2.5節参照) によって第 NLEV 番目と NLEV+1 番目のコンターレベルの値を調べ, その差 (第 NLEV+1 番目から NLEV 番目を引いた値) を返す . UDQCLV が参照するコンターレベルの値は必ずしも昇順にならんでいるとは限らないことに注意すること.

   (b) コンターレベルは 2 本以上設定されていなければならない. また NLEV は 1 以上で, 設定されているコンターレベルの総数より小さくなければならない . これらの条件が満たされないとき RUDVAL は警告メッセージを出して, 0 を返す.

8.2.9 UDSFMT

1. 機能

   コンターラベルのフォーマットを指定する.

2. 呼び出し方法

   ```
   CALL UDSFMT(CFMT)
   ```

3. パラメーターの説明

CFMT (C*(*)) 指定するフォーマット (長さは 16 文字以下).

4. 備考

(a) たとえばコンターラベルをすべて整数型にしたいとき

```
CALL UDSFMT('(I8)')
```

とすればよい (文字は左詰めされるのでラベルが 8 文字以下なら, たとえば'(I12)' でも効果は同じ). 初期値はサブルーチン CHVAL (『MISC1」のマニュアル FMTLIB 参照) が解釈するオプション'D' (有効数字 3 ケタでほぼ 4–5 文字の表現) である.

(b) 現在設定されているフォーマットは

```
CALL UDQFMT(CFMT)
```

によって参照できる.

8.2.10 UDpGET/UDpSET(UDpSTX)

1. 機能

UDPACK サブルーチンパッケージで使用する内部変数を参照/変更する. (UDpSTX は実行時オプションによる変更を許す.)

2. 呼び出し方法

```
CALL UDpGET(CP,IPARA)
CALL UDpSET(CP,IPARA)
CALL UDpSTX(CP,IPARA)
```

3. パラメーターの説明

CP	(C*8)	内部変数の名前.
IPARA	(I,R,L)	内部変数の値.

以下に CP として指定できる名前のリストを記す.

'INDXMJ'	(I)	major line のラインインデクス (初期値は 3).
'INDXMN'	(I)	minor line のラインインデクス (初期値は 1).
'LABEL'	(L)	major line にラベルをつけるかどうかの指定をする. .TRUE. ならつける; .FALSE. ならつけない (初期値は.TRUE.).
'LDASH'	(L)	負のコンターラインを異なるラインタイプで描くかどうかを指定する. .TRUE. のとき負のコンターラインを内部変数'IDASH' の参照するラインタイプで描く; .FALSE. のときすべてのコンターラインを内部変数'ISOLID' の参照するラインタイプで描く (初期値は .TRUE.).
'IDASH'	(I)	'LDASH' が.TRUE. のとき, 負のコンターレベルに使うラインタイプ (初期値は 3:破線).
'ISOLID'	(I)	'LDASH' が.TRUE. のとき 0 以上のコンターレベルに, または 'LDASH' が.FALSE. のときすべてのコンターレベルに使うラインタイプ (初期値は 1:実線).

'ICYCLE' (I) minor line を major line の間に何本引くかを指定する. すなわち 'ICYCLE' で参照する値に 1 本の割合で major line を引き, 残りは minor line を引く (初期値は 2).

'NLEV' (I) コンターレベルを自動的に生成するとき, この内部変数を参照して約 NLEV 本のコンターレベルを生成する (初期値は 12).

'RSIZEL' (R) コンターラベルの文字高 (単位は V 座標系; 初期値は不定). この内部変数が不定のとき UZRGET/UZRSET の管理する内部変数'RSIZEL1' を参照する (3.7節参照).

'LMSG' (L) グラフの下部マージンにメッセージを書くかどうかを指定する. .TRUE. なら書く; .FALSE. なら書かない (初期値は .TRUE.).

'RSIZET' (R) グラフの下部マージンに書くメッセージの文字高 (単位は V 座標系; 初期値は不定). この内部変数が不定のとき UZpGET/UZpSET の管理する内部変数'RSIZEL1' を参照する (3.7節参照).

'XTTL' (R) グラフの下部マージンに書くメッセージの位置を決める内部変数. -1 から$+1$ までの実数値をとりうる. -1 のとき左よせ, 0 のときセンタリング, $+1$ のとき右よせする (初期値は 0).

'IUNDEF' (I) 値が不定であることを示す内部変数 (初期値は-999).

4. 備考

(a) 内部変数を管理するための下請けルーチンとして以下のものがある.

UDPQNP(NCP)	内部変数の総数 NCP を求める.
UDPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
UDPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
UDPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
UDPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) UDpGET は上に述べた UDPQID を呼んで内部変数の位置を求め, UDPQVL によって値を参照する; UDpSET は UDPQID を呼んで内部変数の位置を求め, UDPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは UDPQID が出力する.

(c) ITYPE としては適切な型の定数または変数を指定すること.

8.2.11 その他のサブルーチン・関数

このパッケージにはこのほかに以下の下位ルーチン・関数がある. ここではその名前をあげるにとどめる.

```
UDSCLZ  UDQCLZ  UDLINE  UDUXUY  UDGRDN  RUDVAL
UDBSET  UDBCLR  LUDCHK  UDISET  UDLABL
```

第 9 章

UEPACK : トーン塗りつぶしルーチン

9.1 概要

UEPACK は 2 次元等値線図のぬりわけをおこなうサブルーチンパッケージである. ぬりわけは U 座標系でおこなわれるので, すべての正規化変換に対応している.

このパッケージにおいて, 等値線図のぬりわけは 『GRPH1」のトーンプリミティブを用いておこなわれている (『GRPH1」のマニュアル参照). なおこのパッケージでは, ぬりわけをおこなうだけで等値線そのものや座標軸の外枠などは描かない.

UEPACK においてユーザーが使用するサブルーチンとしては以下のようなものがある.

- 作画ルーチン

  UETONE ··· トーンによる 2 次元等値線図のぬりわけをおこなう.

  UETONF ··· イメージ描画機能により 2 次元等値線図のぬりわけをおこなう.

- ぬりわけるレベルに関するルーチン

  UEGTLA ··· 最小値 ・最大値 ・きざみ値でぬりわけるレベル生成する.

  UEGTLB ··· 格子点値を与える配列ときざみ値でぬりわけるレベルを生成する.

  UESTLV ··· ぬりわけるレベルとパターンを 1 レベルごと指定する.

  UESTLN ··· ぬりわけるレベルとパターンを配列で複数レベル指定する.

  UEITLV ··· ぬりわけるレベルとパターンをすべて無効にする.

- その他のルーチン

  UEpGET/UEpSET··· 内部変数を参照/変更する.

格子点座標に関する情報は UWPACK (第 11節参照) が管理している.

- 格子点座標の設定に関するルーチン

  UWSGXA/UWSGYA ··· 各座標値で設定する.

  UWSGXB/UWSGYB ··· 最小値と最大値で設定する.

2 次元データをとにかくぬりわけたい (quick look) というときには, 単に UETONE を呼ぶだけでよい. このときは, 現在設定されているウインドウいっぱいに等間隔の格子点を設定し, 負の領域に斜線のハッチをつける.

ぬりわけるレベル値の設定を UDPACK と同様にしておこないたいときは, UE*p*GET/UE*p*SET の管理する内部変数 'LTONE' を .TRUE. として UETONE を呼べば, UDPACK が自動的に生成するきざみ値で (厳密には UDPACK と同じアルゴリズムにもとづいてレベル値を設定し) ぬりわける. さらに, UDGCLA に対応して UEGTLA, UDGCLB に対応して UEGTLB を呼べばコンターレベルとぬりわけるレベルの値が同一に設定される.

もしも格子点が不等間隔の場合は, UETONE を呼ぶ前に UWSGXA (X 方向)/ UWSGYA (Y 方向) を呼んで格子点の座標値をあらかじめ指定すればよい. あるいはウインドウのある一部分に等間隔の格子点を設定したい場合は, UWSGXB (X 方向)/ UWSGYB (Y 方向) をもちい最小値と最大値を指定して格子点を設定すればよい.

上に述べたように, ぬりわけるレベルをあらかじめ指定したいときは, UETONE を呼ぶ前に UEGTLA または UEGTLB を呼べばよい. ただしこの場合は, 等間隔のトーンレベルしか生成されない. もしも, 不等間隔のトーンレベルとパターンを指定したい場合や, 特定のトーンレベルとパターンを追加したい場合は, UESTLV または UESTLN を呼べばよい. (すべてのレベルを無効にするには UEITLV を呼べばよい.) なお, ぬりわけは指定したレベルの順におこなう. ぬりわけるトーンパターンについては, 「GRPH1」SGPACK の付録を参照のこと.

このパッケージが用いている 「GRPH1」のトーンプリミティブは出力装置の能力に応じて hard fill と soft fill を切替えることができる. SG*p*GET/SG*p*SET が管理する内部変数'LSOFTF' (「GRPH1」のマニュアル参照) が.TRUE. のときは soft fill をおこなう; .FALSE. のときは hard fill をおこなう (初期値は.TRUE.). ただし, hard fill を指定してもその能力がない場合は soft fill となる. なお soft fill はフィルエリアサブプリミティブ (「GRPH1」フィルエリアサブプリミティブの節参照) の下位ルーチンを用いている.

また, 色を用いたべた塗り (下位 3 桁が 999 であるようなトーンパターン番号) による塗りわけをおこなうとき, 色番号をドットなどのパターン番号として読みかえることができる. SG*p*GET/SG*p*SET の管理する内部変数'LCL2TN' (GRPH1/SGPACK 参照) が.TRUE. のときは読みかえる; .FALSE. のときは読みかえない (初期値は.FALSE.). ただし, 色を用いた塗りわけができないような装置では, 'LCL2TN' が.FALSE. と設定されていても, パターン番号による読みかえをおこなおうとする.

すでに描かれた図形の上に soft fill をおこなっても, トーンパターンが上書きされるだけで, それまでに描かれた図形が (べた塗りをしない限り) 消えてしまうことはない. しかしながら, 出力装置によっては hard fill によるぬりわけによって先に描かれた図形が消えてしまうことがあるので, 座標軸や等高線図と一緒に UEPACK を用いるときは描く順番に注意すること.

内部的には, UEBLK1, UEBLK2 という名前の共通ブロックを使用している. ユーザーは同じ名前の外部手続き名 ・共通ブロック名を用いてはならない.

9.2 サブルーチンの説明

9.2.1 UETONE

1. 機能

   トーンによる 2 次元等値線図のぬりわけをおこなう.

2. 呼び出し方法

```
CALL UETONE(Z,MX,NX,NY)
```

3. パラメーターの説明

Z	(R)	MX×NY の 2 次元配列. 作画には NX×NY の部分を使う.
MX	(I)	配列 Z の第 1 次元整合寸法.
NX	(I)	作画に使う配列 Z の第 1 次元寸法.
NY	(I)	作画に使う配列 Z の第 2 次元寸法.

4. 備考

   (a) X 方向の格子点の座標が UWSGXA (11.2.1節参照) または UWSGXB (11.2.2節参照) によって設定されていない場合, SGQWND でウインドウ情報を取得しウインドウの UXMIN, UXMAX を NX−1 等分するような格子点を設定する.

   (b) Y 方向の格子点の座標が UWSGYA (11.2.1節参照) または UWSGYB (11.2.2節参照) によって設定されていない場合, SGQWND でウインドウ情報を取得しウインドウの UYMIN, UYMAX を NY−1 等分するような格子点を設定する.

   (c) ぬりわけをおこなうレベルの値が UEGTLA, UEGTLB, UESTLV または UESTLN によって設定されていない場合, 次のようにぬりわけるレベル値とトーンパターンを設定する. UELGET/UELSET の管理する内部変数'LTONE' が .TRUE. のとき (以下, 特に断らない場合, 単に 「内部変数」というのは UE*p*GET/UE*p*SET の管理する内部変数のことをさす), 次のサブルーチンを内部で呼んで約 NLEV 本のトーンレベルを設定する.

```
CALL UEGTLB(Z,MX,NX,NY,-REAL(NLEV))
```

   NLEV は内部変数'NLEV' が参照する値である (初期値は 12); 内部変数'LTONE' が .FALSE. のとき, 内部変数'RLEV' が参照する値以下の領域を内部変数'IPAT' が参照する値のトーンパターンでぬりわける ('RLEV', IPAT' の初期値はそれぞれ 0.0, 1201:右上がりの斜線). なお 'LTONE' の初期値は .FALSE. である.

   (d) GL*p*GET/GL*p*SET の管理する内部変数'LMISS' が .TRUE. なら欠損値処理をおこなう. つまり欠損値のまわりの格子点はぬりわけない. (その際, 参照する欠損値は GL*p*GET/GL*p*SET の管理する内部変数'RMISS' である.) .FALSE. なら欠損値処理をおこなわない. くわしくは 「MATH1」のマニュアル参照.

   (e) ぬりわけをおこなう前にすべての格子点値のチェックをおこなう. 次の 2 つの場合ぬりわけはおこなわずその旨メッセージが出力される. 2 つの場合とは:

   - すべての格子点が欠損値のとき (missing field).
   - すべての格子点が同一な値のとき (constant field).

9.2.2 UETONF

1. 機能

   イメージデータ描画機能を使って, 2 次元等値線図のぬりわけをおこなう.

2. 呼び出し方法

```
CALL UETONF(Z,MX,NX,NY)
```

3. パラメーターの説明

Z	(R)	MX×NY の 2 次元配列. 作画には NX×NY の部分を使う.
MX	(I)	配列 Z の第 1 次元整合寸法.
NX	(I)	作画に使う配列 Z の第 1 次元寸法.
NY	(I)	作画に使う配列 Z の第 2 次元寸法.

4. 備考

   (a) UETONE とほぼ互換性があるが, 指定できるトーンパターンはべた塗りだけである.

   (b) イメージ描画機能のないデバイスでは使用できない.

   (c) ピクセル毎に周囲 4 点のデータから線形内挿して値 (Z) を求め色を決める. したがって, 色の境界は一般に曲線となる.

9.2.3 UEGTLA

1. 機能

   UDTONE で使うトーンレベル値を最小値 ・最大値およびきざみ値またはおよそのレベル数で設定する.

2. 呼び出し方法

```
CALL UEGTLA(XMIN,XMAX,DX)
```

3. パラメーターの説明

XMIN, XMAX	(R)	トーンレベルの最小値と最大値. 必ずしもきりのよい値でなくてもよい.
DX	(R)	DX > 0 のとき DX をきざみ幅とする. DX $= 0$ のとき内部変数'NLEV' を参照して, 約 NLEV 本のトーンレベルを生成する. DX < 0 のとき約 INT(ABS(DX)) 本のトーンレベルを生成する.

4. 備考

   (a) このルーチンは UDGCLA (8.2.3節参照) と同じアルゴリズムにもとづいてレベル値を生成している. アルゴリズムを同一にするため, UDPACK で用いられる内部変数'NLEV', 'ICYCLE' と同じものが, UEPACK でも用意されている. UDPACK でこれらの値を変えたときは, UEPACK でも変更してやらなければならない.

   (b) XMIN $\neq$ XMAX でなければならない.

(c) トーンレベルは DX が決めるきざみ幅の整数倍になるように設定される.

(d) トーンパターンは内部変数'ICOLOR1' と'ICOLOR2' が示す色番号 (初期値は 15, 94) の間をトーンレベルの数で分割し, それに 'ITPAT' をパターン番号とする (初期値は 999, べた塗り) トーンパターン番号を生成する.

9.2.4 UEGTLB

1. 機能

   UETONE で使うコンターレベル値を格子点値を与える配列ときざみ値またはおよそのレベル数で設定する.

2. 呼び出し方法

```
CALL UEGTLB(Z,MX,NX,NY,DX)
```

3. パラメーターの説明

Z, MX, NX, NY	UETONE (9.2.1節) 参照.
DX	UEGTLA (9.2.3節) 参照.

4. 備考

   (a) このルーチンは Z の格子点値の最小値・最大値をチェックして, 内部で UEGTLA を呼んでいる.

9.2.5 UESTLV

1. 機能

   ぬりわけるレベルとパターンを 1 レベルごと指定する.

2. 呼び出し方法

```
CALL UESTLV(TLEV1,TLEV2,IPAT)
```

3. パラメーターの説明

TLEV1	(R)	ぬりわけるレベルの下限値.
TLEV2	(R)	ぬりわけるレベルの上限値.
IPAT	(I)	トーンパターン番号 (「GRPH1」のマニュアル参照).

4. 備考

   (a) ある値 (ZLEV) 以下の部分すべてをぬりたいときは, TLEV1 として GLpGET/GLpSET の管理する 'RMISS' の値を指定し, TLEV2 として ZLEV を指定すればよい. 逆に, ある値 (ZLEV) 以上の部分をすべてぬりたいときは, TLEV1 として ZLEV を指定し, TLEV2 として'RMISS' の値を指定すればよい.

   (b) TLEV1 $<$ TLEV2 でなければならない.

   (c) 設定するトーンレベルは 100 以下でなければならない.

   (d) 現在設定されているトーンレベルの数は

```
CALL UEQNTL(NTON)
```

によって参照できる.

(e) 第 ITON 番目のトーンレベルの属性は

```
CALL UEQTLV(TLEV1,TLEV2,IPAT,ITON)
```

によって参照できる.

9.2.6 UESTLN

1. 機能

   ぬりわけるレベルとパターンを配列で複数レベル指定する.

2. 呼び出し方法

```
CALL UESTLN(TLEVN,IPATN,NTON)
```

3. パラメーターの説明

TLEVN	(R)	ぬりわけるレベルの値を格納する配列. 大きさ NTON+1.
IPATN	(I)	トーンパターン番号を格納する配列. 大きさ NTON. (トーンパターン番号については 「GRPH1」のマニュアル参照).
NTON	(I)	配列 IPATN の大きさ.

4. 備考

   (a) UESTLV との関連で説明すると, TLEV1 として TLEVN(i), TLEV2 として TLEVN($i+1$), IPAT として IPATN(i) ($i = 1$, NTON) を指定したのと同等である. したがって, TLEVN(i) < TLEVN($i+1$) でなければならない. くわしくは 9.2.5節参照.

   (b) UESTLV と同じように, 現在設定されているトーンレベルの数は UEQNTL で, トーンレベルの属性は UEQTLV によって参照できる.

9.2.7 UEITLV

1. 機能

   UESTLV または UESTLN で設定したトーンレベルを無効にする.

2. 呼び出し方法

```
CALL UEITLV
```

3. パラメーターの説明

   なし

4. 備考

   (a) UESTLV または UESTLN によって既に設定されているトーンレベルを無効にして UETONE を呼びたいとき, このルーチンを呼ぶ必要がある.

9.2.8 UEpGET/UEpSET(UEpSTX)

1. 機能

   UEPACK サブルーチンパッケージで使用する内部変数を参照/変更する. (UEpSTX は実行時オプションによる変更を許す.)

2. 呼び出し方法

```
CALL UEpGET(CP,IPARA)
CALL UEpSET(CP,IPARA)
CALL UEpSTX(CP,IPARA)
```

3. パラメーターの説明

CP	(C*8)	内部変数の名前.
IPARA	(I,R,L)	内部変数の値.

   以下に CP として指定できる名前のリストを記す.

'LTONE'	(L)	ぬりわけるトーンレベルをユーザーが指定しなかったとき, UEGTLB を使ってトーンレベルとトーンパターンを生成するかどうかを指定する. .TRUE. なら UEGTLB をつかう; .FALSE. なら'RLEV'(下記)以下をぬるような設定をする (初期値は.FALSE.).
'RLEV'	(R)	'LTONE' が.FALSE. のとき, ぬりわける領域の境界値(初期値は0.0). ここで指定した値以下の領域をぬる.
'IPAT'	(I)	'LTONE' が.FALSE. のとき, 内部変数'RLEV' 以下の領域をぬるトーンパターン (初期値は 1201:右上がりの斜線).
'ICOLOR1'	(I)	'LTONE' が.TRUE. のとき用いる色番号の下限値 (初期値は 15).
'ICOLOR2'	(I)	'LTONE' が.TRUE. のとき用いる色番号の上限値 (初期値は 94).
'ITPAT'	(I)	'LTONE' が.TRUE. のとき用いるパターン番号 (初期値は 999, べた塗り).

4. 備考

   (a) 内部変数を管理するための下請けルーチンとして以下のものがある.

UEPQNP(NCP)	内部変数の総数 NCP を求める.
UEPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
UEPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
UEPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
UEPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

   (b) UEpGET は上に述べた UEPQID を呼んで内部変数の位置を求め, UEPQVL によって値を参照する; UEpSET は UEPQID を呼んで内部変数の位置を求め, UEPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは UEPQID が出力する.

   (c) 上記内部変数のほかに, UDGCLA, UDGCLB とアルゴリズムの同一性を保つために, UDPACK で用いられる内部変数'NLEV', 'ICYCLE' と同じものが, UEPACK でも用意されている. UDPACK でこれらの値を変えたときは, UEPACK でも変更してやらなければならない.

   (d) ITYPE としては適切な型の定数または変数を指定すること.

9.2.9 その他のサブルーチン

このパッケージにはこのほかに以下の下位ルーチンがある. ここではその名前をあげるにとどめる.

```
UESTLZ  UEQTLZ  UEAREA  UEAINT
```

第 10 章

UGPACK：ベクトル場表示ルーチン

10.1 概要

UGPACK は 2 次元ベクトル場を作画するサブルーチンパッケージである. 現在のところ, ベクトルの長さのスケーリング等を V 座標系でおこない, ベクトルを V 座標系で描いているため, 直角直線座標系 (変換関数番号 ITR= $1 \sim 4$) にしか対応していない.

このパッケージにおいて, ベクトルは GRPH1 のアローサブプリミティブをもちいて描かれる. なおこのパッケージでは, ベクトル場を描くだけで座標軸などの外枠は描かない.

UGPACK においてユーザーが使用するサブルーチンとしては以下のようなものがある.

- 作画ルーチン

  UGVECT $\cdots$ 2 次元ベクトル場を作画する.

- その他のルーチン

  UGSUT $\cdots$ ユニットベクトルのタイトルを設定する.

  UGpGET/UGpSET$\cdots$ 内部変数を参照/変更する.

格子点座標に関する情報は UWPACK (第 11節参照) が管理している.

- 格子点座標の設定に関するルーチン

  UWSGXA/UWSGYA $\cdots$ 各座標値で設定する.

  UWSGXB/UWSGYB $\cdots$ 最小値と最大値で設定する.

2 次元格子点上で与えられたベクトルデータをとにかく描きたい (quick look) というときには, 単に UGVECT を呼ぶだけでよい. このときは, 現在設定されているウインドウいっぱいに等間隔の格子点を設定し, ベクトルの長さも自動的スケーリングされる.

もしも格子点が不等間隔の場合は, UGVECT を呼ぶ前に UWSGXA (X 方向)/ UWSGYA (Y 方向) を呼んで格子点の座標値をあらかじめ指定すればよい. あるいはウインドウのある一部分に等間隔の格子点を設定したい場合は, UWSGXB (X 方向)/ UWSGYB (Y 方向) をもちい最小値と最大値を指定して格子点を設定すればよい.

また内部変数 (以下, 特に断らないかぎり単に 「内部変数」というのは UGpGET/UGpSET の管理する内部変数のことを指す) を設定してやることによってベクトルのスケーリングファクターを変えたり, ユニットベクトルを描いたりすることができる.

10.2 サブルーチンの説明

10.2.1 UGVECT

1. 機能

   2 次元ベクトル場を描く.

2. 呼び出し方法

```
CALL UGVECT(U,MU,V,MV,NX,NY)
```

3. パラメーターの説明

U	(R)	ベクトルの X 成分を与える MU×NY の 2 次元配列. 作画には NX×NY の部分を使う.
MU	(I)	配列 U の第 1 次元整合寸法.
V	(R)	ベクトルの Y 成分を与える MV×NY の 2 次元配列. 作画には NX×NY の部分を使う.
MV	(I)	配列 V の第 1 次元整合寸法.
NX	(I)	作画に使う配列 U, V の第 1 次元寸法.
NY	(I)	作画に使う配列 U, V の第 2 次元寸法.

4. 備考

   (a) ベクトルの各成分の単位は V 座標系における単位とする. ただし普通は, V 座標系の単位に変換した配列 U, V を与える必要はなく, スケーリングファクターを与えてやればよい. このスケーリングファクターは, 内部変数'LNRMAL' が.TRUE.(初期値) なら内部的に決定される; .FALSE. なら内部変数'XFACT1', 'YFACT1' (初期値はそれぞれ 1.0) を参照する. したがって, スケーリングファクターを陽に指定したければ, 'LNRMAL' を.FALSE. として'XFACT1', 'YFACT1' を設定してやればよい. さらに, 'LNRMAL' が.TRUE. のとき, 内部変数'LEQRAT' が.TRUE. ならば X と Y のスケーリングファクターを同じにする (初期値は .TRUE.).

   (b) ベクトルは UFPACK (第 14章参照) のなかの UFLNZV によって描かれる. したがってベクトルの形は UFPGET/UFPSET の管理する内部変数によって決まる. ただし, ベクトルを描く線分のラインインデクスは内部変数'INDEX' が決める (初期値は 3). また, 内部変数'ICENT' によってベクトルの始点を格子点に対してどこにとるか指定できる. 'ICENT' が+1 ならベクトルの終点が格子点と一致するように描く; 0 ならベクトルの中点が格子点と一致するように描く; −1 ならベクトルの始点が格子点と一致するように描く (初期値は 0).

   (c) X 方向の格子点の座標が UWSGXA または UWSGXB によって設定されていない場合, SGQWND でウインドウ情報を取得しウインドウの UXMIN, UXMAX を NX−1 等分するような格子点を設定する.

   (d) Y 方向の格子点の座標が UWSGYA または UWSGYB によって設定されていない場合, SGQWND でウインドウ情報を取得しウインドウの UYMIN, UYMAX を NY−1 等分するような格子点を設定する.

   (e) GL*p*GET/GL*p*SET(『MATH1』のマニュアル参照) の管理する内部変数'LMISS' が.TRUE. なら欠損値処理をおこなう; (その際, 参照する欠損値は GL*p*GET/GL*p*SET の管理する内部変数'RMISS' である) .FALSE. なら欠損値処理をおこなわない. ここでいう欠損値処理とは, ベクトルの X または Y 成分のうち少なくとも一方が欠損値 (以下では, このようなベクトルを欠損値ベクトルと呼ぶ) のときにおこなわれる処理のことである. 具体的には, 内部変数'LMISSP' が.TRUE. の

ときその格子点にマークを描く (描くマークのマーカータイプ, マーカーインデクス, マーカーサイズは内部変数'ITYPE1', 'INDEX', 'RSIZEM' を参照する. 初期値はそれぞれ 5, 3, 0.01.); .FALSE. なら, なにも描かない (初期値は.FALSE.). また, 内部変数'LSMALL' が.TRUE. なら内部変数'RSMALL' が決める値以下のベクトル長のベクトルについても上と同様な処理をおこなう. ただし描くマーカーのマーカータイプは, 内部変数'ITYPE2' を参照する. ('LSMALL', RSMALL, ITYPE2 の初期値はそれぞれ.FALSE., 0.001, 1.)

(f) 作画をおこなう前にすべてのベクトル値のチェックをおこなう. 次の 2 つの場合作画はおこなわずその旨メッセージが出力される. さらに内部変数'LMSG' が.TRUE. のときグラフの下部マージンにもメッセージを書く; .FALSE. のときメッセージを書かない (初期値は.TRUE.). 2 つの場合とは:

- すべてのベクトルが欠損値ベクトルのとき (missing field).
- すべてのベクトルがゼロベクトルのとき (zero field).

(g) 内部変数'LUNIT' が.TRUE. ならユニットベクトルを描く; .FALSE. なら描かない (初期値は.FALSE.). ユニットベクトルは, 内部変数'VXULOC', 'VYILOC' (V 座標系における x,y 座標値: 初期値は不定) を陽に指定しなければビューポートの右下隅から内部変数'VXUOFF', 'VYUOFF' で決まる量だけはなれた位置に描かれる ('VXUOFF', 'VYUOFF' の初期値は 0.02, 0). またユニットベクトルの長さは, 内部変数 VXUNIT, VYUNIT で決まる (単位は V 座標系; 初期値は 0.05, 0.05). ただし内部変数 UXUNIT, UYUNIT が陽に設定されていればそれらの値を用いる (初期値は不定). UXUNIT, UYUNIT は VXUNIT, VYUNIT と違ってユニットベクトルが示す次元量の単位で指定する. さらに'LUNIT' が.TRUE. のとき, 内部変数'LUMSG' が.TRUE. ならユニットベクトルの大きさを図の下部マージンに書く (初期値は.TRUE.); 'LUNIT' が.FALSE. のとき, 内部変数'LMSG' が.TRUE. ならスケーリングファクターを図の下部マージンに書く (初期値は.TRUE.). これらのメッセージは UXPTTL (3.8.5節参照) を使って書く. したがって, まず座標軸を描いた後で UGVECT を呼ぶべきである.

10.2.2 UGSUT

1. 機能

   ユニットベクトルにつけるタイトルを設定する.

2. 呼び出し方法

```
CALL UGSUT(CSIDE,CTTL)
```

3. パラメーターの説明

CSIDE	(C*1)	タイトルをつける位置を指定する. 'X' (X 方向のベクトル), 'Y' (Y 方向のベクトル) が指定できる.
CTTL	(C*(*))	つけるタイトル.

4. 備考

   (a) 設定できるタイトルの数は最大 10 個, またタイトルの長さは最大 32 文字である.

   (b) タイトル文字のラインインデクスは, 内部変数'IUINDX' が決める (初期値は 3). また, タイトルの回転角は X および Y 方向についてそれぞれ内部変数'IUTXRO', 'IUTYRO' が決める (初期値はそれぞれ 0, 90:単位は度).

10.2.3 UGpGET/UGpSET(UGpSTX)

1. 機能

   UGPACK サブルーチンパッケージで使用する内部変数を参照/変更する. (UGpSTX は実行時オプションによる変更を許す.)

2. 呼び出し方法

```
CALL UGpGET(CP,IPARA)
CALL UGpSET(CP,IPARA)
CALL UGpSTX(CP,IPARA)
```

3. パラメーターの説明

CP	(C*8)	内部変数の名前.
IPARA	(I,R,L)	内部変数の値.

以下に CP として指定できる名前のリストを記す.

'INDEX' (I) ベクトルを描く線分のラインインデクス. および'LMISSP', 'LSMALL' が.TRUE. のときに描くマーカーのラインインデクス (初期値は 3).

'LNRMAL' (L) 内部的にスケーリングファクターを決めるかどうかを指定する. .TRUE. のとき内部的に決める; .FALSE. のとき内部変数'XFACT1', 'YFACT1 を参照する (初期値は.TRUE.).

'LEQRAT' (L) 内部的にスケーリングファクターを決めるとき ('LNRNAL' が.TRUE. のとき) X 方向と Y 方向のスケーリングファクターを同じにするかどうかを指定する. .TRUE. なら同じにする (初期値は.TRUE.).

'XFACT1' (R) 'LNRNAL' が.FALSE. のときに参照する X 方向のスケーリングファクター (初期値は 1.0).

'YFACT1' (R) 'LNRNAL' が.FALSE. のときに参照する Y 方向のスケーリングファクター (初期値は 1.0).

'XFACT2' (R) 実際に用いられる X 方向のスケーリングファクター (初期値は不定). ユーザーが指定しても意味がない.

'YFACT2' (R) 実際に用いられる Y 方向のスケーリングファクター (初期値は不定). ユーザーが指定しても意味がない.

'LMSG' (L) グラフの下部マージンにスケーリングファクターなどのタイトルを書くかどうかを指定する. .TRUE. なら書く;.FALSE. なら書かない (初期値は.TRUE.).

'ICENT' (I) ベクトルの始点を格子点に対してどこにとるか指定する. +1 ならベクトルの終点が格子点と一致するように描く; 0 ならベクトルの中点が格子点と一致するように描く; −1 ならベクトルの始点が格子点と一致するように描く (初期値は 0).

'LMISSP' (L) 欠損値ベクトルのとき, その格子点にマーカーを描くかどうか (欠損値処理をおこなうかどうか) を指定する. .TRUE. なら描く;.FALSE. なら描かない (初期値は.FALSE.).

'ITYPE1' (I) 欠損値処理をおこなうとき, その格子点に描くマーカーのマーカータイプ (初期値は 5).

'LSMALL' (L) ベクトルの大きさが内部変数'RSMALL' で決まる値以下のとき欠損値処理と同じように, その格子点にマーカーを描くかどうか指定する. .TRUE. なら描く;.FALSE. なら描かない (初期値は.FALSE.).

'RSMALL' (R) 内部変数'LSMALL' が参照するベクトルの大きさのしきい値 (初期値は 0.001).

'ITYPE2' (I) 内部変数'LSMALL' が.TRUE. のとき, その格子点に描くマーカーのマーカータイプ (初期値は 1).

'RSIZEM' (R) 'LMISSP', 'LSMALL' が.TRUE. のときに描くマーカーのマーカーサイズ (初期値は 0.01).

'RSIZET' (R) グラフの下部マージンに書くタイトルの文字高 (単位は V 座標系; 初期値は不定). この内部変数が不定のとき UZpGET/UZpSET の管理する内部変数'RSIZEL1' を参照する (3.7節参照).

'XTTL' (R) グラフの下部マージンに書くタイトルの位置を決める内部変数. -1 から$+1$ までの実数値をとりうる. -1 のとき左よせ, 0 のときセンタリング, $+1$ のとき右よせする (初期値は 0).

'LUNIT' (L) ユニットベクトルを描くかどうかを指定する. .TRUE. なら描く; .FALSE. なら描かない (初期値は.FALSE.).

'LUMSG' (L) ユニットベクトルを描くときグラフの下部マージンにユニットベクトルの単位を書くかどうかを指定する. .TRUE. なら書く;.FALSE. なら書かない (初期値は.TRUE.).

'VXULOC' (R) ユニットベクトルの始点の X 座標 (単位は V 座標系; 初期値は不定).

'VYULOC' (R) ユニットベクトルの始点の Y 座標 (単位は V 座標系; 初期値は不定).

'VXUOFF' (R) 'VXULOC', 'VYULOC' が陽に指定されていないとき, ユニットベクトルをビューポートの右下隅から X 方向にどれだけ離すかを与えるオフセット値 (単位は V 座標系; 初期値は 0.02).

'VYUOFF' (R) 'VXULOC', 'VYULOC' が陽に指定されていないとき, ユニットベクトルをビューポートの右下隅から Y 方向にどれだけ離すかを与えるオフセット値 (単位は V 座標系; 初期値は 0.0).

'UXUNIT' (R) X 方向のユニットベクトルの長さ (単位は次元量; 初期値は不定).

'UYUNIT' (R) Y 方向のユニットベクトルの長さ (単位は次元量; 初期値は不定).

'VXUNIT' (R) X 方向のユニットベクトルの長さ (単位は V 座標系; 初期値は 0.05).

'VYUNIT' (R) Y 方向のユニットベクトルの長さ (単位は V 座標系; 初期値は 0.05).

'IUNTTL' (I) 設定されている, ユニットベクトルにつけるタイトルの数. ユーザーは値を設定してはいけない.

'RSIZEUT' (R) ユニットベクトルにつけるタイトルの文字高 (単位は V 座標系; 初期値は不定). この内部変数が不定のとき UZpGET/UZpSET の管理する内部変数'RSIZEL1' を参照する (3.7節参照).

'IUINDX' (I) ユニットベクトルにつけるタイトルのラインインデクス (初期値は 3).

'IUTXRO' (I) X 方向のユニットベクトルにつけるタイトルの回転角 (単位は度; 初期値は 0).

'IUTYRO' (I) Y 方向のユニットベクトルにつけるタイトルの回転角 (単位は度; 初期値は 90). タイトルの長さが 1 のときに限って'IUTYRO' は 90 以外の値をとることが許される.

'IUNDEF' (I) 値が不定であることを示す内部変数 (初期値は-999).

4. 備考

(a) 内部変数を管理するための下請けルーチンとして以下のものがある.

UGPQNP(NCP)	内部変数の総数 NCP を求める.
UGPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
UGPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
UGPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
UGPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) UGpGET は上に述べた UGPQID を呼んで内部変数の位置を求め, UGPQVL によって値を参照する; UGpSET は UGPQID を呼んで内部変数の位置を求め, UGPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは UGPQID が出力する.

(c) ITYPE としては適切な型の定数または変数を指定すること.

10.2.4 その他のサブルーチン ・関数

このパッケージにはこのほかに以下の下位ルーチン ・関数がある. ここではその名前をあげるにとどめる.

UGVXVY UGUNIT UGDUT

第 11 章

UWPACK : 格子点情報の管理ルーチン

11.1 概要

UWPACK は 2 次元の格子点座標に関する情報を管理するサブルーチンパッケージである.

UWPACK には以下のサブルーチンおよび関数がある (以下のルーチン名で 5 文字目が X であるものは X 軸に関するルーチンで, Y であるものは Y 軸に関するルーチンである).

UWSGXA/UWSGYA ··· 座標値で格子点を設定する.

UWQGXA/UWQGYA ··· 格子点の座標値を参照する.

UWSGXB/UWSGYB ··· 最小値 ・最大値および格子数で格子点を設定する.

UWQGXB/UWQGYB ··· 格子点の最小値 ・最大値および格子数を参照する.

UWSGXZ/UWSGYZ ··· 格子点が定義されているかどうかを指定する.

UWQGXZ/UWQGYZ ··· 格子点が定義されているかどうかを調べる.

UWDFLT ··· 座標値が設定されていない時に等間隔格子を設定する.

RUWGX/RUWGY ··· 格子点の座標値を参照する.

IUWGX/IUWGY ··· 格子点番号を参照する.

UWQGXI/UWQGYI ··· 格子点番号と位置を参照する.

これらは, UDPACK(第 8章参照), UEPACK (第 9章参照), UGPACK (第 10章参照) の中で用いられており, 格子点に関する情報を共有している. たとえば, 等高線図の描画 (UDPACK) とトーンによる塗るわけ (UEPACK) を同じ格子点構造を持った配列に対して同時におこないたいとき, 格子点に関する情報を UWPACK によって最初に一度だけ設定すれば, UDPACK と UEPACK はその格子点情報をもとに作画をおこなう.

内部的には, UWBLKX, UWBLKY という名前の共通ブロックにおいて格子点情報が管理されている. ユーザーは同じ名前の外部手続き名 ・共通ブロック名を用いてはならない.

11.2 サブルーチンの説明

11.2.1 UWSGXA/UWSGYA

1. 機能

   格子点配列の格子点座標を各座標値で設定する.

2. 呼び出し方法

```
CALL UWSGXA(XP,NX)
CALL UWSGYA(YP,NY)
```

3. パラメーターの説明

XP	(R)	長さ NX の配列. X 座標を U 座標系の値で指定する.
YP	(R)	長さ NY の配列. Y 座標を U 座標系の値で指定する.
NX	(I)	配列 XP の長さ.
NY	(I)	配列 YP の長さ.

4. 備考

   (a) このサブルーチンを呼ぶと UWQGXZ/UWQGYZ の参照する値が.TRUE. になる.

   (b) 作業用配列の大きさの制約から NX, NY は 400 以下でなければならない. また, NX, NY は 2 以上でなければならない.

   (c) 現在設定されている格子点の座標値は

```
CALL UWQGXA(XP,NX)
CALL UWQGYA(YP,NY)
```

   によって参照できる.

11.2.2 UWSGXB/UWSGYB

1. 機能

   格子点配列の格子点座標を最小値と最大値および格子点数で設定する.

2. 呼び出し方法

```
CALL UWSGXB(UXMIN,UXMAX,NX)
CALL UWSGYB(UYMIN,UYMAX,NY)
```

3. パラメーターの説明

UXMIN, UXMAX	(R)	U 座標系における X 座標の最小値と最大値.
UYMIN, UYMAX	(R)	U 座標系における Y 座標の最小値と最大値.
NX	(I)	X 方向の格子点数. つまり UXMIN から UXMAX までを NX−1 等分するような格子点座標を設定する.
NY	(I)	Y 方向の格子点数. つまり UYMIN から UYMAX までを NY−1 等分するような格子点座標を設定する.

4. 備考

(a) このサブルーチンを呼ぶと UWQGXZ/UWQGYZ の参照する値が .TRUE. になる.

(b) UWSGXA/UWSGYA とは違って, 作業配列を用いないので NX, NY に制限はない. ただし, NX, NY は 2 以上でなければならない.

(c) 現在設定されている格子点座標の最小値と最大値および格子点数は

```
CALL UWQGXB(UXMIN,UXMAX,NX)
CALL UWQGYB(UYMIN,UYMAX,NY)
```

によって参照できる.

11.2.3 UWQGXZ/UWQGYZ

1. 機能

   格子点が定義されているかどうかを調べる.

2. 呼び出し方法

```
CALL UWQGXZ(LSETX)
CALL UWQGYZ(LSETY)
```

3. パラメーターの説明

LSETX, LSETY	(L)	格子点が定義されているかどうかを調べる論理変数. 定義されていれば .TRUE., 定義されていなければ .FALSE. が返される.

4. 備考

(a) 現在設定されている格子点を無効にするには

```
CALL UWSGXZ(.FALSE.)
CALL UWSGYZ(.FALSE.)
```

としてやれば, 格子点は定義されていない状態になったとみなされる.

11.2.4 RUWGX/RUWGY

1. 機能

   格子点の座標値を返す.

2. 呼び出し方法

```
RUWGX(IX)
RUWGY(IY)
```

3. パラメーターの説明

IX, IY	(I)	格子点の番号.
RUWGX, RUWGY	(R)	格子点の U 座標系の座標値.

4. 備考

(a) この関数は UWSGXA/UWSGYA, UWSGXB/UWSGYB のどちらの方法で格子点の設定をした場合でも, 同様に使用できる.

11.2.5 IUWGX/IUWGY

1. 機能

   座標値に最も近い格子番号を返す.

2. 呼び出し方法

```
IUWGX(UX)
IUWGY(UY)
```

3. パラメーターの説明

UX, UY	(R)	座標値.
IUWGX, IUWGY	(I)	格子点番号.

4. 備考

   (a) これらの関数を使う前に UWIGXI/UWIGYI を呼んで初期化する必要がある.

   (b) この関数は UWSGXA/UWSGYA, UWSGXB/UWSGYB のどちらの方法で格子点の設定をした場合でも, 同様に使用できる.

11.2.6 UWQGXI/UWQGYI

1. 機能

   与えられた座標値よりも小さい最大の格子点番号を返す. 次のグリッドとの間の位置も返す.

2. 呼び出し方法

```
CALL UWQGXI(UX,IUX,FRACX)
CALL UWQGYI(UY,IUY,FRACY)
```

3. パラメータの説明

UX, UY	(R)	座標値.
IUX, IUY	(I)	格子点番号.
FRACX, FRACY	(R)	次の格子点との間での位置. FRACX/FRACY=0 の時 UX/UY は格子点上, FRACX/FRACY=1 の時 UX/UY は次の格子点上にある.

4. 備考

   (a) これらの関数を使う前に UWIGXI/UWIGYI を呼んで初期化する必要がある.

   (b) この関数は UWSGXA/UWSGYA, UWSGXB/UWSGYB のどちらの方法で格子点の設定をした場合でも, 同様に使用できる.

11.2.7 UWDFLT

1. 機能

   グリッドの設定がされていない時に, 等間隔グリッドを設定する.

2. 呼び出し方法

```
CALL UWDFLT(NX,NY)
```

3. パラメータの説明

NX, NY (I) グリッド数.

4. 備考

(a) なし.

第 12 章

UMPACK : 地図投影ルーチン

12.1 概要

12.1.1 はじめに

UMPACK は地図情報 (海岸線や緯度経度線) を描くための描画ルーチンと地図投影変換のために必要なパラメータを適当に決めるコントロールルーチンからなる.

UMPACK においてユーザーが使用するサブルーチンとしては以下のようなものがある.

- 描画ルーチン

  UMPMAP ··· 各種地図情報を描く.

  UMPGLB ··· 緯度経度線と地図の輪郭を描く.

  UMPGRD ··· 緯度経度線を描く.

  UMPLIM ··· 地図の輪郭を描く.

- コントロールルーチン

  UMSCNT ··· 球面と投影面の 「接点」を指定する.

  UMSCWD ··· 円形のウィンドウを指定する.

  UMSPNT ··· 地図に含める点を指定する.

  UMPFIT ··· 変換パラメタを適当に決める.

  UMpGET/UMpSET··· 内部変数を参照/変更する.

UMPFIT は GRFRM などによって未定義状態にされ, その後ユーザーが設定しなかった変換関数に関するパラメタを適当に決定する. これらのパラメタで変換関数を確定するには, GRSTRF を呼ばなければならない.

地図投影に必要なパラメータがすべて設定されているとき, UMPFIT はなにもしない. いっぽう, 地図投影に必要なパラメータが 1 つでも設定されていないとき, UMPFIT はすでに設定された情報から, 適切なパラメタを決定しようと試みる. UMPFIT の動作は大きく分けて以下の 2 段階からなる.

1: 地図投影の極 (PLX, PLY, PLROT) を決める.

2: 指定された点, および範囲が全て入るように, 相似変換パラメタ (SIMFAC, VXOFF, VYOFF) を調節する.

すなわち, まず地図投影の中心となるべき点を定め, 後は拡大 / 縮小および平行移動で必要な範囲を図に収める.

この地図投影の中心となるべき点のことをパッケージでは 「接点」と呼ぶ. 円筒図法および円錐図法においては, 球と投影面は線で接しているが, この接線と中央経線の交点を 「接点」とする. 方位図法における 「接点」は地図投影の極と同義である. これは地図学上の正確な言葉ではないが, 通常もっとも歪みの少ない中心となるべき点を表わす全ての図法共通の用語として 「接点」という言葉を使うことにする.

UMPFIT の第一段階は, この 「接点」を求めることである. このために使われる情報の優先順位は以下の通りである.

1: SGSMPL による極の指定.

2: UMSCNT による接点の指定.

3: UMSCWD による円形ウィンドウの指定.

4: SGSWND による矩形ウィンドウの指定.

5: UMSPNT による描画点指定.

6: デフォルト値 (PLX=0, PLY=90)

優先順位の高い情報が指定されていれば, それ以下の情報は無視される.

第二段階の描画範囲決定は, UMSCWD による円形ウィンドウ, SGSWND による矩形ウィンドウ, UMSPNT によって指定された点, 全てが描画範囲に入るように行なわれ, これらの間に優先順位はない. これらの描画範囲を決定する情報が指定されていなかった場合は, 全球が指定されたものと見なす. ただし, UMLSET によって内部変数'LGLOBE' が .TRUE. に設定された場合は, ウィンドウなどの情報は全て無視され全球が描画される. このとき, 無限遠を含む以下の図法に関しては T 座標系の次の範囲を 「全球」と見なす.

メルカトル図法:	南緯 $75°$ から北緯 $75°$
ポーラーステレオ図法:	赤道から北極
ランベルト正角円錐図法:	赤道から北極

なお, ビューポートのパラメタ VXMIN, VXMAX, VYMIN, VYMAX の一部または全部が指定されていなかった場合には, まず, 最大描画範囲に指定された範囲が最大内接するようにスケーリングされ, さらに指定された範囲が描画できる最小のビューポートを設定する.

UMPACK では同じ結果を得る複数のコーディングが可能であるが, 標準的な使用法を以下に示す.

- 矩形ウィンド (円錐図法)

```
1         CALL GRFRM
2         CALL GRSTRN(20)
3         CALL GRSWND(UXMIN, UXMAX, UYMIN, UYMAX)
4         CALL UMPFIT
5         CALL GRSTRF
6
```

- 円形ウィンド (方位図法)

```
1         CALL GRFRM
2         CALL GRSTRN(31)
3         CALL UMSCWD(UXC, UYC, R)
4         CALL UMPFIT
5         CALL GRSTRF
6
```

UXC, UYC は地図の中心になる緯度経度で, この点を中心に半径 R (角度) の範囲が描画される.

- 全球 (円筒図法)

```
1         CALL GRFRM
2         CALL GRSTRN(12)
3         CALL UMSCNT(UXC, 0., 0.)
4         CALL UMLSET('LGLOBE', .TRUE.)
5         CALL UMPFIT
6         CALL GRSTRF
7
```

UXC は中央経線となる経度.

- 全球 (横軸法)

```
1         CALL GRFRM
2         CALL GRSTRN(12)
3         CALL UMSCNT(UXC, UYC, 90.)
4         CALL UMLSET('LGLOBE', .TRUE.)
5         CALL UMPFIT
6         CALL GRSTRF
7
```

12.2 サブルーチンの説明:コントロール

12.2.1 UMSCNT

1. 機能

   投影面の 「接点」を指定する.

2. 呼び出し方法

```
CALL UMSCNT(UXC, UYC, ROT)
```

3. パラメーターの説明

UXC,UYC	(R)	接点の経度, 緯度.
ROT	(R)	中央の経線の傾き. (時計周り)

4. 備考

   (a) この情報は UMPFIT が使う.

   (b) 通常の円筒図法 (正軸法) では, UYC=0, ROT=0 である. そうでないとき斜軸法となる.

   (c) 円錐図法の時, 標準緯度 (STLAT1) が指定されていなければ, UYC が標準緯度となる. 標準緯度が指定されている時は, その標準緯度に対応する円錐を設定し, それが (UXC, UYC) で接するように, 円錐を傾ける. すなわち, 斜軸法になる.

   (d) 方位図法では, (UXC,UYC,ROT) は, 地図投影の極 (PLX,PLY,PLROT) と同じ意味になる.

12.2.2 UMSCWD

1. 機能

   円形のウィンドウを設定する.

2. 呼び出し方法

```
CALL UMSCWD(UXC, UYC, R)
```

3. パラメーターの説明

UXC,UYC	(R)	中心の経度, 緯度.
R	(R)	ウィンドウ半径. (角度)

4. 備考

   (a) この情報は UMPFIT が使う.

   (b) この円の範囲が入るようにスケーリングファクタが決められる.

   (c) 円筒図法, 円錐図法では UMSCNT と異なり, UYC が 0 でなくても, 陽に指定されない限り, 正軸法が採用される.

   (d) 円錐図法の時, 標準緯度 (STLAT1) が指定されていなければ, UYC が標準緯度となる.

12.2.3 UMSPNT

1. 機能

   地図に含める点を指定する.

2. 呼び出し方法

   ```
   CALL UMSPNT(N, UXP, UYP)
   ```

3. パラメーターの説明

N	(I)	点の数
UXP,UYP	(R(N))	点の経度, 緯度.

4. 備考

   (a) このルーチンは何度でも呼ぶことができる.

   (b) N の合計は 64 までである.

   (c) この情報から 「接点」を求める時には, 点の重心を 「接点」とする.

12.2.4 UMPFIT

1. 機能

   地図投影の変換関数のパラメタを適切に決める.

2. 呼び出し方法

   ```
   CALL UMPFIT
   ```

3. パラメーターの説明

   なし.

4. 備考

   (a) このルーチンは, 与えられた情報から変換関数のパラメタを適切に決めて設定するが, 確定はしない. (SGSTRF は呼ばない.)

12.3 サブルーチンの説明:地図情報の描画

12.3.1 UMPGLB

1. 機能

   地図の境界線 (縁) と緯度線, 経度線を描く.

2. 呼び出し方法

   ```
   CALL UMPGLB
   ```

3. パラメーターの説明

なし.

4. 備考

(a) これは次の UMPGRD と UMPLIM を連続して呼んでいる.

12.3.2 UMPGRD

1. 機能

緯度線 ・経度線を描く.

2. 呼び出し方法

```
CALL UMPGRD
```

3. パラメーターの説明

なし.

4. 備考

(a) 描く緯度線 ・経度線の種類としては major line と minor line の 2 種類がある.

(b) major line を描くかどうかは UMLGET/UMLSET が管理する内部変数'LGRIDMJ' が決める (初期値は .TRUE.. (以下, 特に断らない限り 「内部変数」とは UMpGET/UMpSET が管理する内部変数のことをいう.) 描くその間隔は'DGRIDMJ' が決める. また描く線分の属性は'INDEXMJ', 'ITYPEMJ' が決める (初期値はそれぞれ 3, 1). さらに子午線に関しては, 内部変数'DGRPLMJ' によって極から何度はいったところから描くかが指定できる (初期値は 0).

(c) minor line を描くかどうかは内部変数'LGRIDMN' が決める (初期値は .TRUE.. 描くその間隔は 'DGRIDMN' が決める. また描く線分の属性は'INDEXMN', 'ITYPEMN' が決める (初期値はそれぞれ 1, 3). さらに子午線に関しては, 内部変数'DGRPLMN' によって極から何度はいったところから描くかが指定できる (初期値は 0).

(d) 'DGRIDMJ' と 'DGRIDMN' は GRFRM を呼んだタイミングで 「不定」に設定される. これらの値が不定のとき, 地図の大きさから適当な値が設定される.

12.3.3 UMPLIM

1. 機能

地図の境界線 (縁) を描く.

2. 呼び出し方法

```
CALL UMPLIM
```

3. パラメーターの説明

なし.

4. 備考

(a) 地図境界を描く線分のラインインデクスは内部変数'INDEXBND' が決める (初期値は 3). ラインタイプはつねに 1 (実線) である.

12.3.4 UMPMAP

1. 機能

   各種地図情報を描く.

2. 呼び出し方法

```
CALL UMPMAP(CDSN)
```

3. パラメーターの説明

CDSN	(C*(*))	地図情報ファイル名.

4. 備考

   (a) CDSN で指定したファイル名が直接用いられるとは限らない. 実際に読み込まれるファイル名は UMQFNM が決める. UMQFNM の標準的な仕様では, まずカレントディレクトリを探す. 次に GLCGET/GLCSET が管理する内部変数'DUPATH' のさすディレクトリを探す. 最後に内部変数 'DSPATH' のさすディレクトリを探す.

   (b) 現在のところ次のような地図情報ファイルが用意されている.

'coast_world'	:	世界の海岸線.
'border_world'	:	世界の国境.
'plate_world'	:	世界のプレート境界.
'state_usa'	:	アメリカの州境.
'coast_japan'	:	日本の海岸線.
'pref_japan'	:	日本の県境.

12.3.5 UMpGET/UMpSET(UMpSTX)

1. 機能

   UMPACK サブルーチンパッケージで使用する内部変数を参照/変更する. (UMpSTX は実行時オプションによる変更を許す.)

2. 呼び出し方法

```
CALL UMpGET(CP,IPARA)
CALL UMpSET(CP,IPARA)
CALL UMpSTX(CP,IPARA)
```

3. パラメーターの説明

CP	(C*8)	内部変数の名前.
IPARA	(I,R,L)	内部変数の値.

   以下に CP として指定できる名前のリストを記す.

'LGRIDMJ' (L) UMPGRD が緯度線・経度線を描くとき major line を描くかどうかを指定する. .TRUE. なら描く; .FALSE. なら描かない (初期値は.TRUE.).

'LGRIDMN' (L) UMPGRD が緯度線・経度線を描くとき minor line を描くかどうかを指定する. .TRUE. なら描く; .FALSE. なら描かない (初期値は.TRUE.).

'INDEXMJ' (I) UMPGRD が描く緯度線・経度線の major line のラインインデクス (初期値は 3).

'INDEXMN' (I) UMPGRD が描く緯度線・経度線の minor line のラインインデクス (初期値は 1).

'ITYPEMJ' (I) UMPGRD が描く緯度線・経度線の major line のラインタイプ (初期値は 1).

'ITYPEMN' (I) UMPGRD が描く緯度線・経度線の minor line のラインタイプ (初期値は 3).

'DGRIDMJ' (I) UMPGRD が描く緯度線・経度線の major line の間隔 (単位は度; 初期値は 30).

'DGRIDMN' (I) UMPGRD が描く緯度線・経度線の minor line の間隔 (単位は度; 初期値は 10).

'DGRPLMJ' (I) UMPGRD が描く経度線の major line を極からどれだけはいったところから描くかを指定する. (単位は度; 初期値は 0).

'DGRPLMN' (I) UMPGRD が描く経度線の minor line を極からどれだけはいったところから描くかを指定する. (単位は度; 初期値は 0).

'INDEXBND'(I) UMPLIM が描く地図境界線のラインインデクス (初期値は 3).

'MAXBND' (I) UMPLIM が描く地図境界線を描くときいくつぐらいの線分に分割して描くかを設定する (初期値は 300).

'INDEXOUT'(I) 海岸線などの地図情報を描く線分のラインインデクス (初期値は 1).

'ITYPEOUT'(I) 海岸線などの地図情報を描く線分のラインタイプ (初期値は 1).

'LGLOBE' (L) UMSTRF が変換パラメータを決定するとき, 全球を描くようにするかどうかを指定する. .TRUE. なら全球を描く; .FALSE. なら全球を描かない (初期値は.FALSE.).

4. 備考

(a) 内部変数を管理するための下請けルーチンとして以下のものがある.

UMPQNP(NCP)	内部変数の総数 NCP を求める.
UMPQID(CP,IDX)	内部変数 CP の位置 IDX を求める.
UMPQCP(IDX,CP)	IDX の位置にある内部変数の名前 CP を参照する.
UMPQVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を参照する.
UMPSVL(IDX,IPARA)	IDX の位置にある内部変数の値 IPARA を変更する.

(b) UMpGET は上に述べた UMPQID を呼んで内部変数の位置を求め, UMPQVL によって値を参照する; UMpSET は UMPQID を呼んで内部変数の位置を求め, UMPSVL によって値を設定する. したがって指定した内部変数名が見つからないとき, エラーメッセージは UMPQID が出力する.

(c) ITYPE としては適切な型の定数または変数を指定すること.

第 13 章

UBPACK : ラベル付折線ルーチン

13.1 概要

UBPACK はラベル付きの折れ線を描くサブルーチンパッケージである.

このパッケージの実現していた機能は, 現在, GRPH1 のポリラインプリミティブがすべて実現しており, このパッケージは将来的には削除される.

以下にはサブルーチンのリストをあげるにとどめる. なお, 対応する GRPH1/SGPACK のルーチン名を括弧内に示す.

UBPLZU (SGPLZU)	U 座標系でラベル付き折れ線を描く (属性も設定).
UBPLZV (SGPLZV)	V 座標系でラベル付き折れ線を描く (属性も設定).
UBPLU (SGPLU)	U 座標系でラベル付き折れ線を描く.
UBPLV (SGPLV)	V 座標系でラベル付き折れ線を描く.
UBSPLT/UBQPLT (SGSPLT/SGQPLT)	ラインタイプを設定/参照する.
UBSPLI/UBQPLI (SGSPLI/SGQPLI)	ラインインデクスを設定/参照する.
UBSPLC/UBQPLC (SGSPLC/SGQPLC)	ラベルの文字列を設定/参照する.
UBSPLS/UBQPLS (SGSPLS/SGQPLS)	文字列の高さを設定/参照する.
UBPGET/UBPSET (SGPGET/SGPSET)	内部変数を参照/設定する.

UBPACK と SGPACK のポリラインプリミティブとの間で仕様が異なる点は以下の通り.

- UBPLU, UBPLV は呼ばれるたびに設定されている文字列の最後の文字の文字番号がふやされたが, SGPLU, SGPLV は設定されている文字列を変更しない. ただし, SGMPLC というルーチンが呼ばれると, 設定されている文字列の最後の文字の文字番号がふやされる.
- UBPGET/UBPSET の管理していた内部変数 'INDEX' は SGpGET/SGpSET では 'INDEXC' として管理されている.

第 14 章

UFPACK：矢印付線分ルーチン

14.1 概要

UFPACK は矢印付き線分 (線分の終点から対称な 2 本の線分をつけくわえる) を描くサブルーチンパッケージである.

このパッケージの実現していた機能は, 現在, GRPH1 のアローサブプリミティブがすべて実現しており, このパッケージは将来的には削除される.

以下にはサブルーチンのリストをあげるにとどめる. なお, 対応する GRPH1/SGPACK のルーチン名を括弧内に示す.

UFLNZU (SGLAZU)	U 座標系で矢印付き線分を描く (属性も設定).
UFLNZV (SGLAZV)	V 座標系で矢印付き線分を描く (属性も設定).
UFLNU (SGLAU)	U 座標系で矢印付き線分を描く.
UFLNV (SGLAV)	V 座標系で矢印付き線分を描く.
UFSLNI/UFQLNI (SGSLAI/SGQLAI)	ラインインデクスを設定/参照する.
UFPGET/UFPSET (SGPGET/SGPSET)	内部変数を参照/設定する.

UFPACK と SGPACK のアローサブプリミティブとの間で仕様が異なる点は, UFPACK がラインインデクスしか属性を持たないのに対して, アローサブプリミティブはラインインデクスとラインタイプの 2 つの属性を持つことである.

謝辞

資源	**dcl-5**
プログラム製作	石岡圭一, 酒井敏, 佐藤薫, 塩谷雅人, 沼口敦
プログラム作成協力	石渡正樹, 竹広真一, 中島健介, 西憲敬, 林祥介, 保坂征宏, 余田成男
デバッグ協力	石田明生, 石渡正樹, 市川香, 沖大幹, 沖理子, 桜井隆博, 佐藤正樹, 竹広真一, 中島健介, 沼口敦, 根田昌典, 野村真佐子, 藤尾伸三, 保坂征宏
文章	酒井敏, 塩谷雅人, 余田成男
文章協力	石渡正樹, 佐藤薫, 佐藤正樹, 竹広真一, 中島健介, 西憲敬, 沼口敦, 林祥介, 保坂征宏
データソース	NCAR [フォント情報, 地図情報]
参考資源	Layout [二木徹], XYGRAPH [佐藤亨], Numerical Recipes [Willam H. Press et al.], NCARG [NCAR], FFTPACK [Paul Swarztrauber]

版権

本資源の版権は地球流体電脳倶楽部に属する. 資源の利用にあたっては地球流体電脳倶楽部の定める規定にしたがっていただきたい. 原則として, 教育的目的の場合には自由に使用・改変して良いものとしている. 地球流体電脳倶楽部が定める規定の詳細は地球流体電脳倶楽部サーバー

```
dennou.gaia.h.kyoto-u.ac.jp
```

上の

```
~ftp/saloon/dennou/To.Users
```

というファイルに収められているので参照されたい. (このファイルは電脳ライブラリを収めたファイルシステムのトップディレクトリにも置かれている.) 上記のアクセスができない場合は, 電子メールにて地球流体電脳倶楽部管理グループ

```
dennou_admin@gaia.h.kyoto-u.ac.jp
```

へ連絡されたい.

引用について

「地球流体電脳ライブラリ」の著作者は「地球流体電脳倶楽部」, その英語名は 'GFD-DENNOU Library', 'GFD-DENNOU Club' である. 地球流体電脳ライブラリを用いて作成された著作物等には, 英語では例えば "The figures were produced by GFD-DENNOU Library." 日本語では例えば「図の作成には地球流体電脳ライブラリをもちいた.」のように引用していただければ幸いである.