地球流体電脳ライブラリ

MATH2

(数学処理上位パッケージ)

地球流体電脳倶楽部

2018 年 07 月 20 日 (DCL-7.3.3)

目次

第1章　概要

MATH2 数学処理上位パッケージは地球流体の様々な現場で標準的に用いられる数値計算のための基本ツール群として計画されている. 残念ながら現状での整備はあまり進んではいない.

MATH2 に対応するパッケージには既存のソフトウェア, 例えば, IMSL [*1], NAG [*2], Numerical Recipes [*3] 等を参照すればソースコードとして手に入るようなものが多いのではあるが, その精度等の信頼度は必ずしも地球流体の研究者にとって十分なものとは言えない. また, これらのソフトウェアには PDS (Public Domain Softwares) ではないものが多いので自分たちのソフトウェアに組み込んで自由に配布することができないという難点がある.

MATH2 パッケージの目標として

- 差分の基本スキーム
- フーリエ変換
- 球面直交展開, 逆変換
- 座標変換
- 基本統計パッケージ
- 特種関数
- 固有値問題
- ...

を自力開発する, あるいは, PDS から精選する努力が現在なされている.

[*1] The IMSL libraries という老舗の数値計算ライブラリ. 日本では日本 IMSL 株式会社 (Tel 03-5689-7550) が扱っている. PC からスパコンまでほとんどすべての計算機で (金を出せば) 使うことができる. 東大センター, 京大センター には既に導入されているがソースを見ることはできない. PC 版を買えば PC 用のではあるけれどソースコードを見ることができる. FORTARAN のみならず C 版もあるのがミソ.

[*2] Numerical Algorithms Group の略. 英国 6 大学計算機センターではじめられた数値計算ライブラリ開発プロジェクトの名称. 現在では NAG 社 (The Numerical ALgorithms Group Limited) という会社が設立されていて配布してくれる. 東大センターには最近導入され, ソースコードも公開されている (村尾裕一他, 1991 : 東京大学大型計算機センターニュース, **23**, 73-81).

[*3] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., 1986 : *Numerical Recipes. The Art of Scientific Computing.* Cambridge University Press. 数値計算の基本ルーチンを集めた解説・公式集. Cambridge University Press はソースコードのフロッピーディスクも販売している. バグが多いので注意.

第2章　FFTLIB：高速フーリエ変換

2.1　概要

任意の長さのデータについて高速フーリエ変換をおこなうサブルーチンパッケージ. NCARの数値計算ライブラリより移植した.

サブルーチンの説明の中の「定義」の節では, 入出力パラメータの数学的解説に関して次のような表記法をとる: 処理する配列 X (長さ N) の第 i 番目 ($i = 1, \ldots, \mathrm{N}$) の配列要素について, 入力時の値を x_i (小文字), 出力時の値を X_i (大文字) と書く.

以下の7つのサブルーチン群の中で初期化をおこなうサブルーチン (サブルーチン名が I で終わる) は, そのサブルーチン群に属する変換ルーチンを用いる際, かならず最初に1回呼ばなければならない. ただしそれ以後は, 異なるデータ数を指定するときに限って初期化ルーチンを呼べばよい. なお, 初期化ルーチンが用いる作業領域は, 同じサブルーチン群に属する変換ルーチンを用いている間変更してはならない. (この作業領域には, 因数と三角関数表が格納されている).

2.2　サブルーチンのリスト

周期実数値データのフーリエ変換をおこなうサブルーチン群.

`RFFTI(N,WSAVE)`	初期化をおこなう.
`RFFTF(N,R,WSAVE)`	フーリエ順変換をおこなう.
`RFFTB(N,R,WSAVE)`	フーリエ逆変換をおこなう.

RFFTI, RFFTF, RFFTB の簡易型サブルーチン群.

`EZFFTI(N,WSAVE)`	初期化をおこなう.
`EZFFTF(N,R,A0,A,B,WSAVE)`	フーリエ順変換をおこなう.
`EZFFTB(N,R,A0,A,B,WSAVE)`	フーリエ逆変換をおこなう.

奇の周期データの SINE 変換をおこなうサブルーチン群.

`SINTI(N,WSAVE)`	初期化をおこなう.
`SINT(N,X,WSAVE)`	SINE 変換をおこなう.

偶の周期データの COSINE 変換をおこなうサブルーチン群.

`COSTI(N,WSAVE)`	初期化をおこなう.
`COST(N,X,WSAVE)`	COSINE 変換をおこなう.

奇数波数成分のみの SIN 変換をおこなうサブルーチン群.

`SINQI(N,WSAVE)`	初期化をおこなう.
`SINQF(N,X,WSAVE)`	SINE 順変換をおこなう.
`SINQB(N,X,WSAVE)`	SINE 逆変換をおこなう.

偶数波数成分のみの COSINE 変換をおこなうサブルーチン群.

`COSQI(N,WSAVE)`	初期化をおこなう.
`COSQF(N,X,WSAVE)`	COSINE 順変換をおこなう.
`COSQB(N,X,WSAVE)`	COSINE 逆変換をおこなう.

周期複素数データのフーリエ変換をおこなうサブルーチン群.

`CFFTI(N,WSAVE)`	初期化をおこなう.
`CFFTF(N,C,WSAVE)`	フーリエ順変換をおこなう.
`CFFTB(N,C,WSAVE)`	フーリエ逆変換をおこなう.

2.3 サブルーチンの説明

2.3.1 RFFTI/RFFTF/RFFTB

1. 機能

   周期実数値データのフーリエ変換をおこなう. `RFFTI` は初期化をおこなう; `RFFTF` はフーリエ順変換をおこなう; `RFFTB` はフーリエ逆変換をおこなう.

2. 定義

   N が偶数のとき $N' = N/2 - 1$, N が奇数のとき $N' = (N-1)/2$ とおく.

   順変換は次のように定義される.

$$R_1 = \sum_{i=1}^{N} r_i,$$

$$R_{2k} = \sum_{i=1}^{N} r_i \cos\frac{2\pi(i-1)k}{N}, \quad R_{2k+1} = -\sum_{i=1}^{N} r_i \sin\frac{2\pi(i-1)k}{N} \quad (k = 1, \ldots, N').$$

   ただし N が偶数のとき,

$$R_N = \sum_{i=1}^{N} (-1)^{i-1} r_i.$$

   逆変換は次のように定義される.

   N が偶数のとき,

$$R_i = r_1 + (-1)^{i-1} r_N + 2\sum_{k=1}^{N'} (r_{2k} \cos\frac{2\pi(i-1)k}{N} - r_{2k+1} \sin\frac{2\pi(i-1)k}{N}) \quad (i = 1, \ldots, N).$$

N が奇数のとき,

$$R_i = r_1 + 2\sum_{k=1}^{N'}(r_{2k}\cos\frac{2\pi(i-1)k}{N} - r_{2k+1}\sin\frac{2\pi(i-1)k}{N}) \quad (i=1,\ldots,N).$$

3. 呼び出し方法

```
CALL RFFTI(N,WSAVE)
CALL RFFTF(N,R,WSAVE)
CALL RFFTB(N,R,WSAVE)
```

4. パラメーターの説明

N	(I)	処理するデータの長さ.
WSAVE	(R)	作業用配列. 長さは少なくとも 2N+15 以上でなければならない.
R	(R)	処理する実数型配列. 入力パラメータでもあり出力パラメータでもある. (上記定義参照).

5. 備考

(a) この変換では正規化されない. つまり RFFTF, RFFTB を続けて呼ぶと, もとの N 倍の値が返される.

2.3.2 EZFFTI/EZFFTF/EZFFTB

1. 機能

RFFTI, RFFTF, RFFTB の簡易型サブルーチン. EZFFTI は初期化をおこなう; EZFFTF はフーリエ順変換をおこなう; EZFFTB はフーリエ逆変換をおこなう.

2. 定義

順変換は次のように定義される. (N が偶数のとき $N' = N/2-1$, N が奇数のとき $N' = (N-1)/2$ とおく.)

$$A_0 = \frac{1}{N}\sum_{i=1}^{N} r_i,$$

$$A_k = \frac{2}{N}\sum_{i=1}^{N} r_i\cos\frac{2\pi(i-1)k}{N}, \quad B_k = \frac{2}{N}\sum_{i=1}^{N} r_i\sin\frac{2\pi(i-1)k}{N} \quad (k=1,\ldots,N').$$

ただし N が偶数のとき,

$$A_{N/2} = \frac{1}{N}\sum_{i=1}^{N}(-1)^{i-1}r_i, \quad B_{N/2} = 0.$$

逆変換は次のように定義される. (N が偶数のとき $N' = N/2$, N が奇数のとき $N' = (N-1)/2$ とおく.)

$$R_i = a_0 + \sum_{k=1}^{N'}(a_k\cos\frac{2\pi(i-1)k}{N} + b_k\sin\frac{2\pi(i-1)k}{N}) \quad (i=1,\ldots,N).$$

3. 呼び出し方法

```
CALL EZFFTI(N,WSAVE)
CALL EZFFTF(N,R,A0,A,B,WSAVE)
CALL EZFFTB(N,R,A0,A,B,WSAVE)
```

4. パラメーターの説明

N	(I)	処理するデータの長さ.
WSAVE	(R)	作業用配列. 長さは少なくとも 3N+15 以上でなければならない.
R	(R)	処理する実数型配列. EZFFTF においては入力パラメータ, EZFFTB においては出力パラメータである.
A0	(R)	上記定義における A_0 および a_0.
A, B	(R)	N が偶数のとき N/2, N が奇数のとき (N − 1)/2 の長さの実数型配列 (上記定義参照).

5. 備考

(a) この変換では正規化される. つまり EZFFTF, EZFFTB を続けて呼ぶと, もとの値が返される.

2.3.3 SINTI/SINT

1. 機能

奇の周期データの SINE 変換をおこなう. SINTI は初期化をおこなう; SINT は SINE 変換をおこなう.

2. 定義

順変換は次のように定義される.

$$X_k = 2\sum_{i=1}^{N} x_i \sin\frac{\pi ik}{N+1}, \quad (k = 1, \ldots, N).$$

逆変換は順変換と同じである.

3. 呼び出し方法

```
CALL SINTI(N,WSAVE)
CALL SINT(N,R,WSAVE)
```

4. パラメーターの説明

N	(I)	処理するデータの長さ.
WSAVE	(R)	作業用配列. 長さは少なくとも 2.5N+15 以上でなければならない.
R	(R)	処理する実数型配列. 入力パラメータでもあり出力パラメータでもある (上記定義参照).

5. 備考

(a) SINT は逆変換でもある. また, この変換では正規化されない. つまり SINT を 2 回続けて呼ぶと, もとの 2(N + 1) 倍の値が返される.

2.3.4 COSTI/COST

1. 機能

偶の周期データの COSINE 変換をおこなう. COSTI は初期化をおこなう; COST は COSINE 変換をおこなう.

2. 定義

順変換は次のように定義される.

$$X_k = x_1 + (-1)^{k-1} x_N + 2 \sum_{i=2}^{N} x_i \cos \frac{\pi(i-1)(k-1)}{N-1}, \quad (k = 1, \ldots, N).$$

逆変換は順変換と同じである.

3. 呼び出し方法

`CALL COSTI(N,WSAVE)`

`CALL COST(N,R,WSAVE)`

4. パラメーターの説明

N	(I)	処理するデータの長さ.
WSAVE	(R)	作業用配列. 長さは少なくとも 3N+15 以上でなければならない.
R	(R)	処理する実数型配列. 入力パラメータでもあり出力パラメータでもある (上記定義参照).

5. 備考

(a) `COST` は逆変換でもある. また, この変換では正規化されない. つまり `COST` を 2 回続けて呼ぶと, もとの $2(\mathtt{N}-1)$ 倍の値が返される.

2.3.5 SINQI/SINQF/SINQB

1. 機能

奇数波数成分のみの SIN 変換をおこなう. `SINQI` は初期化をおこなう; `SINQF` は SINE 順変換をおこなう; `SINQB` は SINE 逆変換をおこなう.

2. 定義

順変換は次のように定義される.

$$X_k = (-1)^{k-1} x_N + 2 \sum_{i=1}^{N-1} r_i \sin \frac{\pi(2k-1)i}{2N} \quad (k = 1, \ldots, N).$$

逆変換は次のように定義される.

$$X_i = 4 \sum_{k=1}^{N} r_k \sin \frac{\pi(2k-1)i}{2N} \quad (i = 1, \ldots, N).$$

3. 呼び出し方法

`CALL SINQI(N,WSAVE)`

`CALL SINQF(N,R,WSAVE)`

`CALL SINQB(N,R,WSAVE)`

4. パラメーターの説明

N	(I)	処理するデータの長さ.
WSAVE	(R)	作業用配列. 長さは少なくとも 3N+15 以上でなければならない.
R	(R)	処理する実数型配列. 入力パラメータでもあり出力パラメータでもある (上記定義参照).

5. 備考

(a) この変換では正規化されない. つまり `SINQF`, `SINQB` を続けて呼ぶと, もとの 4N 倍の値が返される.

2.3.6 COSQI/COSQF/COSQB

1. 機能

   偶数波数成分のみの COSSINE 変換をおこなう. `COSQI` は初期化をおこなう; `COSQF` は COSINE 順変換をおこなう; `COSQB` は COSINE 逆変換をおこなう.

2. 定義

   順変換は次のように定義される.

$$X_k = x_1 + 2\sum_{i=2}^{N} r_i \cos\frac{\pi(2k-1)(i-1)}{2N} \quad (k = 1, \ldots, N).$$

   逆変換は次のように定義される.

$$X_i = 4\sum_{k=1}^{N} r_k \cos\frac{\pi(2k-1)(i-1)}{2N} \quad (i = 1, \ldots, N).$$

3. 呼び出し方法

```
CALL COSQI(N,WSAVE)
CALL COSQF(N,R,WSAVE)
CALL COSQB(N,R,WSAVE)
```

4. パラメーターの説明

`N`	`(I)`	処理するデータの長さ.
`WSAVE`	`(R)`	作業用配列. 長さは少なくとも 3N+15 以上でなければならない.
`R`	`(R)`	処理する実数型配列. 入力パラメータでもあり出力パラメータでもある (上記定義参照).

5. 備考

   (a) この変換では正規化されない. つまり `COSQF`, `COSQB` を続けて呼ぶと, もとの 4N 倍の値が返される.

2.3.7 CFFTI/CFFTF/CFFTB

1. 機能

   周期複素数データのフーリエ変換をおこなう. `CFFTI` は初期化をおこなう; `CFFTF` はフーリエ順変換をおこなう; `CFFTB` はフーリエ逆変換をおこなう.

2. 定義

   以下では $i = \sqrt{-1}$ とする.

   順変換は次のように定義される.

$$C_k = \sum_{j=1}^{N} c_j \exp(-i\frac{2\pi(j-1)(k-1)}{N}) \quad (k = 1, \ldots, N).$$

   逆変換は次のように定義される.

$$C_j = \sum_{k=1}^{N} c_k \exp(i\frac{2\pi(j-1)(k-1)}{N}) \quad (j = 1, \ldots, N).$$

3. 呼び出し方法

```
CALL CFFTI(N,WSAVE)
CALL CFFTF(N,C,WSAVE)
CALL CFFTB(N,C,WSAVE)
```

4. パラメーターの説明

N	(I)	処理するデータの長さ.
WSAVE	(R)	作業用配列. 長さは少なくとも 4N+15 以上でなければならない.
C	(C)	処理する複素数型配列. 入力パラメータでもあり出力パラメータでもある (上記定義参照).

5. 備考

(a) この変換では正規化されない. つまり CFFTF, CFFTB を続けて呼ぶと, もとの N 倍の値が返される.

2.3.8 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. ここではサブルーチン名をあげるにとどめる.

cfftb1.f	cfftf1.f	cffti1.f	cosqb1.f	cosqf1.f	ezfft1.f	passb.f	passb2.f
passb3.f	passb4.f	passb5.f	passf.f	passf2.f	passf3.f	passf4.f	passf5.f
pimach.f	radb2.f	radb3.f	radb4.f	radb5.f	radbg.f	radf2.f	radf3.f
radf4.f	radf5.f	radfg.f	rfftb1.f	rfftf1.f	rffti1.f	sint1.f	

第3章　ODELIB：常微分方程式 (ルンゲクッタ)

3.1　概要

このパッケージは連立常微分方程式の積分を行うためのものである．ここに含まれるルーチンは大きく 3 つのレベルにわかれる．最下層のレベルは, 特定のアルゴリズムに従って 1 ステップ積分するアルゴリズムルーチンで, その上に精度チェックをしながら 2 ステップ積分して適当な刻み幅を求めるステッパルーチンがあり, 最上位に実際に与えられた積分区間の積分を行うドライバルーチンがある．ユーザーは通常このドライバルーチンを使って積分を行う．この際, ユーザーは使用するアルゴリズムルーチン, または, ステッパルーチンを目的に応じて指定することができる．

なお, このライブラリは Numerical Recipes のアルゴリズムに基づくものである．

3.2　サブルーチンのリスト

アルゴリズムルーチン

`CALL ODRKG(N,FCN,T,DT,X,DX,XOUT,WORK)`	ルンゲ-クッタ-ギル．
`CALL ODRK4(N,FCN,T,DT,X,DX,XOUT,WORK)`	4 次精度のルンゲ-クッタ．
`CALL ODRK2(N,FCN,T,DT,X,DX,XOUT,WORK)`	2 次精度のルンゲ-クッタ．
`CALL ODRK1(N,FCN,T,DT,X,DX,XOUT,WORK)`	1 次精度のルンゲ-クッタ．

ステッパルーチン

`CALL ODRKGR(N,FCN,T,DT,EPSL,X,WORK)`	可変幅のきざみを求める．
`CALL ODRKGS(N,FCN,T,DT,EPSL,X,WORK)`	固定幅のきざみを求める．
`CALL ODRK4R(N,FCN,T,DT,EPSL,X,WORK)`	可変幅のきざみを求める．
`CALL ODRK4S(N,FCN,T,DT,EPSL,X,WORK)`	固定幅のきざみを求める．

ドライバルーチン

`CALL ODRKDU(N,ALGOR,FCN,T,TEND,ISTEP,X,WORK)`	固定刻幅で積分を行う．
`CALL ODRKDV(N,STEPER,FCN,T,TINT,DT,X,WORK)`	要求精度を満たすように積分を行う．

内部変数管理ルーチン

CALL OD*p*GET(CPARA,IPARA)	内部変数を参照する．
CALL OD*p*SET(CPARA,IPARA)	内部変数を設定する．

3.3 サブルーチンの説明

3.3.1 ODRKG/ODRK4/ODRK2/ODRK1

1. 機能

   特定のアルゴリズムに従い, 与えられたステップ幅で 1 ステップの積分を行う. 積分アルゴリズムは ODRKG がルンゲ-クッタ-ギル, ODRK4 が 4 次精度のルンゲ-クッタ, ODRK2 が 2 次精度のルンゲ-クッタ, ODRK1 が 1 次精度のルンゲ-クッタ (Euler) である.

2. 呼び出し方法

```
CALL ODRKG(N,FCN,T,DT,X,DX,XOUT,WORK)
CALL ODRK4(N,FCN,T,DT,X,DX,XOUT,WORK)
CALL ODRK2(N,FCN,T,DT,X,DX,XOUT,WORK)
CALL ODRK1(N,FCN,T,DT,X,DX,XOUT,WORK)
```

3. パラメーターの説明

N	(I)	被積分変数 (方程式) の数. (i)
FCN	手続き名	DX を計算するサブルーチン名.
T	(R)	独立変数 t の値. (i)
DT	(R)	積分ステップ幅. (i)
X	R(N)	被積分変数の $t = \mathtt{T}$ における値. (i)
DX	R(N)	被積分変数の $t = \mathtt{T}$ における微分値. (i) ODRKG の時は DX の値は保存されない.
XOUT	R(N)	被積分変数の $t = \mathtt{T} + \mathtt{DT}$ における値. (i) 実引数として X と同じ変数を指定してもよい.
WORK	R(N,M)	作業変数. M=3 (ODRK4) または 1 (その他のルーチン).

4. 備考

   (a) FCN は サブルーチンの形でユーザーが用意する. その形式は以下のとおり.

```
SUBROUTINE FCN(N,T,X,DX)
```

3.3.2 ODRKGR/ODRKGS/ODRK4R/ODRK4S

1. 機能

   ODRKG (ODRKGR, ODRKGS) または, ODRK4 (ODRK4R, ODRK4S) により 2 ステップの積分を行ない, 精度のチェックをして適当な DT を求める. ODRKGS と ODRK4S は与えられた DT で 2 ステップの積分をした後, 次のステップのための DT を求めてリターンするが, ODRKGR と ODRK4R は要求精度が満たされないとき DT を小さくして再計算を行う.

2. 呼び出し方法

```
CALL ODRKGR(N,FCN,T,DT,EPSL,X,WORK)
CALL ODRKGS(N,FCN,T,DT,EPSL,X,WORK)
CALL ODRK4R(N,FCN,T,DT,EPSL,X,WORK)
CALL ODRK4S(N,FCN,T,DT,EPSL,X,WORK)
```

3. パラメーターの説明

N	(I)	被積分変数 (方程式) の数. (i)
FCN	手続き名	DX を計算するサブルーチン名.
T	(R)	独立変数 t の値. 2 ステップ分の積分幅を加えて出力される. (i/o)
DT	(R)	積分ステップ幅. 適当な値に変更されて出力される. (i/o)
EPSL	(R)	要求精度. (i)
X	R(N)	被積分変数の $t=$ T における値を入力し, $t=$ T $+ 2 \times$ DT における値を出力する.
WORK	R(N,M)	作業変数. M=7 (ODRK4R), M=5 (ODRK4S, ODRKGR), M=3 (ODRKGS).

4. 備考

(a) ODRKGS, ODRK4S は ODRKGR, ODRK4R に比べて, 作業領域が小さくてすむが, 要求精度が満たされたくても, 再計算をしないので, 記憶領域が許す限り, ODRKGR, ODRK4R を推奨する.
なお ODRKGS, ODRK4S では, DT を決める際に, 小さめの安全率を使っているので, 方程式の性質が時間的に緩やかに変わる場合には有効である.

(b) FCN は サブルーチンの形でユーザーが用意する. その形式は以下のとおり.

```
SUBROUTINE FCN(N,T,X,DX)
```

3.3.3 ODRKDU

1. 機能

各アルゴリズムルーチンにより, 固定刻幅で積分を行う.

2. 呼び出し方法

```
CALL ODRKDU(N,ALGOR,FCN,T,TEND,ISTEP,X,WORK)
```

3. パラメーターの説明

N	(I)	被積分変数 (方程式) の数. (i)
ALGOR	手続き名	使用するアルゴリズムルーチン名 (i)
FCN	手続き名	DX を計算するサブルーチン名 (i)
T	(R)	積分を始める独立変数 t の値. RETURN 時には最後のステップ t (TEND) が出力される. (i/o)
TEND	(R)	積分を終了する独立変数 t の値. (i)
ISTEP	(I)	ステップ数. (i)
X	R(N)	被積分変数の $t=$ T における値を入力し, $t=$ TEND における値を出力する. (i/o)
WORK	R(N,M)	作業変数. M=5 (ODRK4), M=3 (その他).

4. 備考

(a) 使用するアルゴリズムルーチンは, 特に支障がない限り, ODRKG (Runge-Kutta-Gill) を推奨する.

(b) FCN は サブルーチンの形でユーザーが用意する. その形式は以下のとおり.

```
SUBROUTINE FCN(N,T,X,DX)
```

3.3.4 ODRKDV

1. 機能

   各ステッパルーチンにより, 要求精度を満たすように積分を行う.

2. 呼び出し方法

   `CALL ODRKDV(N,STEPER,FCN,T,TINT,DT,X,WORK)`

3. パラメーターの説明

`N`	`(I)`	被積分変数 (方程式) の数. (i)
`STEPER`	手続き名	使用するスッテパルーチン名.
`FCN`	手続き名	`DX` を計算するサブルーチン名 (i)
`T`	`(R)`	積分を始める独立変数 t の値. RETURN 時には最後のステップ t `(TEND)` が出力される. (i/o)
`TEND`	`(R)`	積分を終了する t の値. (i)
`DT`	`(I)`	ステップ幅の初期値. (i/o)
`X`	`R(N)`	被積分変数の $t =$ `T` における値を入力し, $t =$ `TEND` における値を出力する. (i/o)
`WORK`	`R(N,M)`	作業変数. M=7 (`ODRK4`), M=5 (その他).

4. 備考

   (a) 精度は `OD`p`GET/OD`p`SET` の管理する内部変数 `'EPSILON'` によって指定できる. 最初の呼び出し時に, 引数 `DT` に適当な値を代入しておくと要求精度を満たすように `DT` が設定されて返って来るので, これを, そのまま次回の呼び出し時に指定すればよい.

   (b) `ODRKDV` は `OD`p`GET/OD`p`SET` の管理する内部変数`'MAXSTEP'` で指定された回数だけ積分しても `TEND` に到達しないときは, エラーメッセージを出力して, 実行を中止する. `'MAXSTEP'` は `ODISET` で変更可能である.

   (c) 使用するステッパルーチンは, 特に支障がない限り, `ODRKGR` (Runge-Kutta-Gill 再計算) を推奨する.

   (d) `FCN` は サブルーチンの形でユーザーが用意する. その形式は以下のとおり.

   `SUBROUTINE FCN(N,T,X,DX)`

3.3.5 ODpGET/ODpSET

1. 機能

   `ODRKDV` で使用する内部変数を参照/設定する.

2. 呼び出し方法

   `CALL OD`p`GET(CPARA,IPARA)`

   `CALL OD`p`SET(CPARA,IPARA)`

3. パラメーターの説明

`CPARA`	`(C*8)`	内部変数の名前. (i)
`IPARA`	`(I,R)`	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.

'EPSILON'	(R)	要求精度.
'MAXSTEP'	(I)	最大積分回数.
'NSTEP'	(I)	実際に積分した回数.

4. 備考

(a) IPARA としては適切な型の定数または変数を指定すること.

第 4 章　SHTLIB：球面調和関数

4.1　概要

これは, スペクトル (球面調和関数) 変換を行なうサブルーチンパッケージであり, 球面調和関数展開の係数からグリッドデータ、およびその逆の変換を行なう. このパッケージは, データ解析を念頭において設計されており, 等間隔グリッドデータを扱えるという特長がある. また, スペクトルで与えられたデータの解析を念頭におき, 逆変換系のルーチンを充実させている. このパッケージの内部では FFTLIB のサブルーチンを用いている.

切断波数 M(三角切断) のスペクトル逆変換は, 以下のように表せる:

$$G(\lambda,\varphi) = \sum_{n=0}^{M}\sum_{m=-n}^{n} S_n^m P_n^m(\sin\varphi)e^{im\lambda}. \tag{4.1}$$

または, ルジャンドル逆変換:

$$W^m(\varphi) \equiv \sum_{n=|m|}^{M} S_n^m P_n^m(\sin\varphi) \tag{4.2}$$

を導入すると、

$$G(\lambda,\varphi) = \sum_{m=-M}^{M} W^m(\varphi)e^{im\lambda} \tag{4.3}$$

と, ルジャンドル逆変換とフーリエ逆変換の積として表される. ここに, λ: 経度, φ: 緯度である.

また, $P_n^m(\mu)$ は 2 に正規化されたルジャンドル陪関数で, 以下のように定義される:

$$P_n^m(\mu) \equiv \sqrt{(2n+1)\frac{(n-m)!}{(n+m)!}}\frac{1}{2^n n!}(1-\mu^2)^{m/2}\frac{d^{n+m}}{d\mu^{n+m}}(\mu^2-1)^n, \tag{4.4}$$

$$\int_{-1}^{1}\{P_n^m(\mu)\}^2 d\mu = 2. \tag{4.5}$$

また, スペクトル逆変換は以下のように表せる:

$$S_n^m = \frac{1}{4\pi}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2} G(\lambda,\varphi)P_n^m(\sin\varphi)e^{-im\lambda}\cos\varphi d\varphi d\lambda. \tag{4.6}$$

逆変換の場合と同様に, フーリエ正変換を,

$$W^m(\varphi) \equiv \frac{1}{2\pi}\int_0^{2\pi} G(\lambda,\varphi)e^{-im\lambda}d\lambda \tag{4.7}$$

と導入すると,

$$S_n^m = \frac{1}{2}\int_{-\pi/2}^{\pi/2} W^m(\varphi) P_n^m(\sin\varphi)\cos\varphi d\varphi \tag{4.8}$$

と, フーリエ正変換とルジャンドル正変換の積として表される.

$G(\lambda, \varphi)$ が実数であるとすると, S_n^m および $W^m(\varphi)$ は以下の関係を満たしている必要がある:

$$W^{-m}(\varphi) = \{W^m(\varphi)\}^* \tag{4.9}$$

$$S_n^{-m} = \{S_n^m\}^* \tag{4.10}$$

ここに, $\{\}^*$ は複素共役を表す. 従って, $W^m(\sin\varphi)$ および S_n^m は $m \geq 0$ の範囲だけを求めれば良い. さらに, 上の制約から, $W^0(\sin\varphi)$ および S_n^0 は実数である.

本ライブラリは, スペクトルデータ (S_n^m) $\rightarrow$ 等間隔緯度円上のウエーブデータ ($W^m(\varphi_j)$) $\rightarrow$ 等間隔格子点上のグリッドデータ ($G(\lambda_i, \varphi_j)$) の逆変換を (1∼ 3) 式に基づいて行うルーチン群, 等間隔格子点上のグリッドデータ ($G(\lambda_i, \varphi_j)$) $\rightarrow$ 等間隔緯度円上のウエーブデータ ($W^m(\varphi_j)$) $\rightarrow$ スペクトルデータ (S_n^m) の正変換を (6∼8) 式に基づいて行うルーチン群およびそして, その他の補助ルーチン群よりなっている.

ここに, 格子点の経度 λ_i, 緯度 φ_j は分割数 I, J によって以下のように定められるものとする:

$$\lambda_i = \frac{\pi i}{I}, \quad i = -I, -I+1, \cdots, 0, \cdots, I-1, I, \tag{4.11}$$

$$\varphi_j = \frac{\pi j}{2J}, \quad j = -J, -J+1, \cdots, 0, \cdots, J-1, J. \tag{4.12}$$

4.2 サブルーチンのリスト

SHTINT(MM,JM,IM,WORK)	初期化ルーチン
SHTNML(MM,N,M,LR,LI)	スペクトルデータの格納位置の計算
SHTLAP(MM,IND,A,B)	スペクトルデータへのラプラシアンの演算
SHTS2W(MM,JM,ISW,S,W,WORK)	スペクトルデータからウエーブデータへの変換
SHTW2G(MM,JM,IM,W,G,WORK)	ウエーブデータからグリッドデータへの変換
SHTS2G(MM,JM,IM,ISW,S,W,G,WORK)	スペクトルデータからグリッドデータへの変換
SHTG2W(MM,JM,IM,G,W,WORK)	グリッドデータからウエーブデータへの変換
SHTW2S(MM,JM,ISW,W,S,WORK)	ウエーブデータからスペクトルデータへの変換
SHTG2S(MM,JM,IM,ISW,G,W,S,WORK)	グリッドデータからスペクトルデータへの変換
SHTSWA(MM,JM,ISW,M1,M2,S,W,WORK)	SHTS2W の下位ルーチン (波数区間指定)
SHTWGA(MM,JM,IM,M1,M2,W,G,WORK)	SHTW2G の下位ルーチン (波数区間指定)
SHTSGA(MM,JM,IM,ISW,M1,M2,S,W,G,WORK)	SHTS2G の下位ルーチン (波数区間指定)
SHTSWM(MM,JM,M,ISW,S,WR,WI,WORK)	SHTS2W の下位ルーチン (ある波数成分のみ)
SHTWGM(MM,JM,IM,M,WR,WI,G,WORK)	SHTW2G の下位ルーチン (ある波数成分のみ)
SHTSGM(MM,JM,IM,M,ISW,S,WR,WI,G,WORK)	SHTS2G の下位ルーチン (ある波数成分のみ)
SHTSWZ(MM,JM,ISW,S,WZ,WORK)	SHTS2W の下位ルーチン (帯状成分のみ)
SHTWGZ(JM,IM,WZ,G)	SHTW2G の下位ルーチン (帯状成分のみ)
SHTSGZ(MM,JM,IM,ISW,S,WZ,G,WORK)	SHTS2G の下位ルーチン (帯状成分のみ)
SHTSWJ(MM,JM,ISW,J,M1,M2,S,WJ,WORK)	SHTS2W の下位ルーチン (緯度指定)
SHTWGJ(MM,IM,M1,M2,WJ,GJ,WORK)	SHTW2G の下位ルーチン (緯度指定)
SHTSGJ(MM,JM,IM,ISW,J,M1,M2,S,WJ,GJ,WORK)	SHTS2G の下位ルーチン (緯度指定)
SHTFUN(MM,JM,M,FUN,WORK)	ルジャンドル陪関数の計算
SHTLFW(MM,JM,M,ISW,WM,SM,WORK)	ルジャンドル正変換
SHTLBW(MM,JM,M,ISW,SM,WM,WORK)	ルジャンドル逆変換

4.3 サブルーチンの説明

4.3.1 SHTINT

1. 機能

   SHTLIB の初期化ルーチン．SHTLIB の他のサブルーチンを使用する前に必ず一度呼ばねばならない．

2. 定義

   切断波数 M, 東西分割数 I, 南北分割数 J については概要を参照.

3. 呼び出し方法

   `SHTINT(MM,JM,IM,WORK)`

4. パラメーターの説明

`MM`	(I)	入力．切断波数 (M).
`JM`	(I)	入力．南北分割数の 1/2(J)
`IM`	(I)	入力．東西分割数の 1/2(I)
`WORK`	(R)	出力．SHTLIB の他のルーチンで用いられる作業領域. 長さ `(JM+1)*(4*JM+5*MM+14)+(MM+1)*(MM+1)+MM+2+6*IM+15` の一次元配列.

5. 備考

   (a) `JM`≥`(MM+1)/2`, `IM`≥`MM+1` でなければばならない.

   (b) SHTLIB を使用している間, 配列 `WORK` の内容を変更してはならない.

4.3.2 SHTNML

1. 機能

   スペクトルデータの格納位置を求める.

2. 定義

   `SHTLIB` において, スペクトルデータ (S_n^m) は概要に述べた制限をもとに, 独立な $(M+1)^2$ 個の成分; $S_0^0, S_1^0, \cdots, S_M^0$, $\mathrm{Re}(S_1^1), \mathrm{Re}(S_2^1), \cdots, \mathrm{Re}(S_M^1)$, $\mathrm{Im}(S_1^1), \mathrm{Im}(S_2^1), \cdots, \mathrm{Im}(S_M^1)$, $\cdots, \mathrm{Re}(S_M^M), \mathrm{Im}(S_M^M)$ がこの順序で (長さ `(MM+1)**2` の配列に) 格納されている． ここに, Re( ) は実数部を, Im( ) は虚数部を表す．このサブルーチンは切断波数 M, S_n^m の全波数 n, および帯状波数 m から $\mathrm{Re}(S_n^m)$ と $\mathrm{Im}(S_n^m)$ の配列中の格納位置を求めるものである.

3. 呼び出し方法

   `SHTNML(MM,N,M,LR,LI)`

4. パラメーターの説明

`MM`	(I)	入力．切断波数 (M).
`N`	(I)	入力．全波数 (n)
`M`	(I)	入力．帯状波数 (m)
`LR`	(I)	出力．$\mathrm{Re}(S_n^m)$ の格納位置.
`LI`	(I)	出力．$\mathrm{Im}(S_n^m)$ の格納位置.

5. 備考

(a) $\mathrm{Im}(S_n^0)$ 成分は存在しないので, `M=0` の場合は `LI` には `LR` と同じ値が返される.

4.3.3 SHTLAP

1. 機能

   スペクトルデータに対してラプラシアンを演算する.

2. 定義

   球面調和関数展開

$$G(\lambda,\varphi) = \sum_{n=0}^{M}\sum_{m=-n}^{n} A_n^m P_n^m(\sin\varphi)e^{im\lambda}. \tag{4.13}$$

   に対して、水平 Laplacian

$$\nabla^2 \equiv \frac{\partial^2}{\cos^2\varphi\partial\lambda^2} + \frac{\partial}{\cos\varphi\partial\varphi}\left(\cos\varphi\frac{\partial}{\partial\varphi}\right) \tag{4.14}$$

   を作用させると, 球面調和関数の性質から,

$$\nabla^2 G(\lambda,\varphi) = \sum_{n=0}^{M}\sum_{m=-n}^{n} -n(n+1)A_n^m P_n^m(\sin\varphi)e^{im\lambda}. \tag{4.15}$$

   となる. そこで,

$$B_n^m \equiv -n(n+1)A_n^m \tag{4.16}$$

   を導入すると,

$$\nabla^2 G(\lambda,\varphi) = \sum_{n=0}^{M}\sum_{m=-n}^{n} B_n^m P_n^m(\sin\varphi)e^{im\lambda}. \tag{4.17}$$

   と表せる. また, 逆に

$$\nabla^2 G(\lambda,\varphi) = \sum_{n=0}^{M}\sum_{m=-n}^{n} A_n^m P_n^m(\sin\varphi)e^{im\lambda}. \tag{4.18}$$

   であるとき,

$$B_n^m \equiv -\frac{1}{n(n+1)}A_n^m \tag{4.19}$$

   を導入すると,

$$G(\lambda,\varphi) = \sum_{n=0}^{M}\sum_{m=-n}^{n} B_n^m P_n^m(\sin\varphi)e^{im\lambda}. \tag{4.20}$$

   と表せる.

   本サブルーチンは, `IND=1` の場合は A_n^m から $B_n^m \equiv -n(n+1)A_n^m$ を, `IND=-1` の場合は A_n^m から $B_n^m \equiv -A_n^m/\{n(n+1)\}$ を計算するものである。

3. 呼び出し方法

   `SHTLAP(MM,IND,A,B)`

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
IND	(I)	入力. ラプラシアンの演算形式を指定する (上記定義を参照).
A	(R)	入力. A_n^m が格納されている配列 (長さおよび並び方は SHTNML を参照).
B	(R)	出力. B_n^m が格納る配列 (長さおよび並び方は SHTNML を参照).

5. 備考

(a) IND=-1 の場合, $B_0^0 = 0$ が代入される.

4.3.4 SHTS2W

1. 機能

スペクトル逆変換の前半部分であるルジャンドル逆変換を行う.

2. 定義

ISW=0 の場合, 通常のルジャンドル逆変換;

$$W^m(\varphi) \equiv \sum_{n=|m|}^{M} S_n^m P_n^m(\sin\varphi) \tag{4.21}$$

を行う.

ISW=1 の場合, 緯度微分のルジャンドル逆変換;

$$W^m(\varphi) \equiv \frac{d}{d\varphi} \sum_{n=|m|}^{M} S_n^m P_n^m(\sin\varphi) \tag{4.22}$$

を行う.

ISW=-1 の場合, 経度微分のルジャンドル逆変換;

$$W^m(\varphi) \equiv \frac{im}{\cos\varphi} \sum_{n=|m|}^{M} S_n^m P_n^m(\sin\varphi) \tag{4.23}$$

を行う.

3. 呼び出し方法

```
SHTS2W(MM,JM,ISW,S,W,WORK)
```

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
ISW	(I)	入力. 変換の種類の指定 (上記定義を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照.
W	(R)	出力. ウエーブデータ. 長さ (2*JM+1)*(2*MM+1) の配列 (並び方は以下の備考を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

(a) W(-JM:JM,-MM:MM) と宣言しておけば, W(J,M) (M>0) には $\mathrm{Re}(W^m(\varphi_j))$ が, W(J,-M) (M>0) には $\mathrm{Im}(W^m(\varphi_j))$ がそれぞれ格納され, W(J,0) には $(W^0(\varphi_j))$ が格納される.

4.3.5 SHTW2G

1. 機能

   スペクトル逆変換の後半部分であるフーリエ逆変換を行う.

2. 定義

   SHTS2W によって作成されたウエーブデータに対して, フーリエ逆変換;

$$G(\lambda,\varphi)=\sum_{m=-M}^{M}W^m(\varphi)e^{im\lambda} \tag{4.24}$$

   を行う.

3. 呼び出し方法

   SHTW2G(MM,JM,IM,W,G,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
IM	(I)	入力. 東西分割数の 1/2(I)
W	(R)	入力. ウエーブデータ. (長さおよび並び方は SHTS2W を参照).
G	(R)	出力. グリッドデータ 長さ (2*IM+1)*(2*JM+1) の配列 (並び方は以下の備考を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) G(-IM:IM,-JM:JM) と宣言しておけば, G(I,J) には $G(\lambda_i,\varphi_j)$ が格納される (λ_i,φ_j の定義については概要を参照).

   (b) M1=0, M2=MM と指定すれば SHTS2W を呼ぶのと全く同様になる.

4.3.6 SHTS2G

1. 機能

   SHTS2W, SHTW2G を連続して行うことにより, スペクトル逆変換を行う.

2. 定義

   ISW および配列 S, W, G の意味は SHTS2W および SHTW2G に同じである. 本サブルーチンは, SHTS2W, SHTW2G を連続して行うことにより,

   ISW=0 の場合, 通常のスペクトル逆変換;

$$G(\lambda,\varphi)=\sum_{n=0}^{M}\sum_{m=-n}^{n}S_n^mP_n^m(\sin\varphi)e^{im\lambda}. \tag{4.25}$$

   を行う.

ISW=1 の場合, 緯度微分のスペクトル逆変換;

$$G(\lambda, \varphi) = \frac{\partial}{\partial\varphi} \sum_{n=0}^{M} \sum_{m=-n}^{n} S_n^m P_n^m(\sin\varphi) e^{im\lambda}. \tag{4.26}$$

を行う.

ISW=-1 の場合, 経度微分のスペクトル逆変換;

$$G(\lambda, \varphi) = \frac{\partial}{\partial\lambda} \sum_{n=0}^{M} \sum_{m=-n}^{n} S_n^m P_n^m(\sin\varphi) e^{im\lambda}. \tag{4.27}$$

を行う.

3. 呼び出し方法

   SHTS2G(MM,JM,IM,ISW,S,W,G,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
IM	(I)	入力. 東西分割数の 1/2(I)
ISW	(I)	入力. 変換の種類の指定 (上記定義を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
W	(R)	出力. ウエーブデータ. (長さおよび並び方は SHTS2W を参照).
G	(R)	出力. グリッドデータ. (長さおよび並び方は SHTW2G を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) 本サブルーチンは, SHTS2W, SHTW2G を連続して呼ぶのと全く同様である.

   (b) ISW=1 と ISW=-1 の変換を両方用いることによって, スカラー場の勾配ベクトルを求めることができる.

4.3.7 SHTG2W

1. 機能

   スペクトル正変換の前半部分であるフーリエ正変換を行う.

2. 定義

   グリッドデータに対して, フーリエ正変換;

$$W^m(\varphi) \equiv \frac{1}{2\pi} \int_0^{2\pi} G(\lambda, \varphi) e^{-im\lambda} d\lambda \tag{4.28}$$

   を行う.

3. 呼び出し方法

   SHTG2W(MM,JM,IM,G,W,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
IM	(I)	入力. 東西分割数の 1/2(I)
G	(R)	入力. グリッドデータ (長さおよび並び方は SHTW2G を参照).
W	(R)	出力. ウエーブデータ (長さおよび並び方は SHTS2W を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

(a) グリッドデータ G が SHTW2G によって作成されたものである場合には, 上記の積分が aliasing なしに完全に評価される.

4.3.8 SHTW2S

1. 機能

スペクトル正変換の後半部分であるルジャンドル正変換を行う.

2. 定義

ISW=0 の場合, 通常のルジャンドル正変換;

$$S_n^m = \frac{1}{2}\int_{-\pi/2}^{\pi/2} W^m(\varphi)P_n^m(\sin\varphi)\cos\varphi d\varphi \tag{4.29}$$

を行う.

ISW=1 の場合, 緯度微分のルジャンドル正変換;

$$S_n^m = \frac{1}{2}\int_{-\pi/2}^{\pi/2} \frac{d}{\cos\varphi d\varphi}\left\{\cos\varphi W^m(\varphi)\right\} P_n^m(\sin\varphi)\cos\varphi d\varphi \tag{4.30}$$

を行う.

ISW=-1 の場合, 経度微分のルジャンドル正変換;

$$S_n^m = \frac{1}{2}\int_{-\pi/2}^{\pi/2} \frac{im}{\cos\varphi} W^m(\varphi)P_n^m(\sin\varphi)\cos\varphi d\varphi \tag{4.31}$$

を行う.

3. 呼び出し方法

SHTW2S(MM,JM,ISW,W,S,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
ISW	(I)	入力. 変換の種類の指定 (上記定義を参照)
W	(R)	入力. ウエーブデータ (長さおよび並び方は SHTS2W を参照).
S	(R)	出力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

(a) ISW=0 の場合, ウエーブデータ W が SHTS2W で ISW=0 として作成されたものである場合には, 上記の積分が aliasing なしに完全に評価される.

(b) ISW= ±1 の場合, ウエーブデータ W が SHTS2W で ISW= ±1 として作成されたものである場合には, 上記の積分が aliasing なしに完全に評価される.

4.3.9 SHTG2S

1. 機能

   SHTG2W, SHTW2S を連続して行うことにより, スペクトル正変換を行う.

2. 定義

   ISW および配列 S, W, G の意味は SHTG2W および SHTW2S に同じである. 本サブルーチンは, SHTG2W, SHTW2S を連続して行うことにより,

   ISW=0 の場合, 通常のスペクトル正変換;

$$S_n^m = \frac{1}{4\pi}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2} G(\lambda,\varphi)P_n^m(\sin\varphi)e^{-im\lambda}\cos\varphi d\varphi d\lambda. \tag{4.32}$$

   を行う.

   ISW=1 の場合, 緯度微分のスペクトル正変換;

$$S_n^m = \frac{1}{4\pi}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2} \frac{\partial}{\cos\varphi\partial\varphi}\left\{\cos\varphi G(\lambda,\varphi)\right\}P_n^m(\sin\varphi)e^{-im\lambda}\cos\varphi d\varphi d\lambda. \tag{4.33}$$

   を行う.

   ISW=-1 の場合, 経度微分のスペクトル正変換;

$$S_n^m = \frac{1}{4\pi}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2} \frac{\partial}{\cos\varphi\partial\lambda}\left\{G(\lambda,\varphi)\right\}P_n^m(\sin\varphi)e^{-im\lambda}\cos\varphi d\varphi d\lambda. \tag{4.34}$$

   を行う.

3. 呼び出し方法

   SHTG2S(MM,JM,IM,ISW,G,W,S,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
IM	(I)	入力. 東西分割数の 1/2(I)
ISW	(I)	入力. 変換の種類の指定 (上記定義を参照)
G	(R)	入力. グリッドデータ. (長さおよび並び方は SHTW2G を参照).
W	(R)	出力. ウエーブデータ. (長さおよび並び方は SHTS2W を参照).
S	(R)	出力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) 本サブルーチンは, SHTG2W, SHTW2S を連続して呼ぶのと全く同様である.

   (b) ISW=1 と ISW=-1 の変換を両方用いることによって, ベクトル場の発散を求めることができる.

4.3.10 SHTSWA

1. 機能

   スペクトル逆変換の前半部分であるルジャンドル逆変換を指定された波数区間のみについて行う．

2. 定義

   スペクトルデータ S_n^m から，ウエーブデータ $W^m(\varphi)$ への変換を M1$\leq |m| \leq$M2 の波数範囲のみについて行う．変換式は SHTS2W を参照．

3. 呼び出し方法

   `SHTSWA(MM,JM,ISW,M1,M2,S,W,WORK)`

4. パラメーターの説明

MM	(I)	入力．切断波数 (M)．
JM	(I)	入力．南北分割数の $1/2(J)$
ISW	(I)	入力．変換の種類の指定 (SHTS2W を参照)
M1	(I)	入力．変換する波数区間の最小値 (上記定義を参照)
M2	(I)	入力．変換する波数区間の最大値 (上記定義を参照)
S	(R)	入力．スペクトルデータ (長さおよび並び方は SHTNML を参照)．
W	(R)	出力．ウエーブデータ (長さおよび並び方は SHTS2W を参照)．
WORK	(R)	SHTINT で初期化された作業領域．

5. 備考

   (a) W(-JM:JM,-MM:MM) と宣言されている場合，指定された波数区間の外すなわち W(J,M) (|M|<M1, |M|>M2) には 0 が代入される．

4.3.11 SHTWGA

1. 機能

   スペクトル逆変換の後半部分であるフーリエ逆変換を指定された波数区間のみについて行う．

2. 定義

   ウエーブデータ $W^m(\varphi)$ からグリッドデータ $G(\lambda,\varphi)$ への変換を M1$\leq |m| \leq$M2 の波数範囲のみについて行う．変換式は SHTW2G を参照．

3. 呼び出し方法

   `SHTWGA(MM,JM,IM,M1,M2,W,G,WORK)`

4. パラメーターの説明

MM	(I)	入力．切断波数 (M)．
JM	(I)	入力．南北分割数の $1/2(J)$
IM	(I)	入力．東西分割数の $1/2(I)$
M1	(I)	入力．変換する波数区間の最小値 (上記定義を参照)
M2	(I)	入力．変換する波数区間の最大値 (上記定義を参照)
W	(R)	入力．ウエーブデータ．(長さおよび並び方は SHTS2W を参照)．
G	(R)	出力．グリッドデータ (長さおよび並び方は SHTW2G を参照)．
WORK	(R)	SHTINT で初期化された作業領域．

5. 備考
   (a) `M1=0, M2=MM` と指定すれば `SHTW2G` を呼ぶのと全く同様になる.

4.3.12 SHTSGA

1. 機能

   SHTSWA, SHTWGA を連続して行うことにより, スペクトル逆変換を指定された波数区間のみについて行う.

2. 定義

   SHTSWA, SHTWGA を参照.

3. 呼び出し方法

   `SHTSGA(MM,JM,IM,ISW,M1,M2,S,W,G,WORK)`

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の $1/2(J)$
IM	(I)	入力. 東西分割数の $1/2(I)$
ISW	(I)	入力. 変換の種類の指定 (SHTS2G を参照)
M1	(I)	入力. 変換する波数区間の最小値 (SHTSWA を参照)
M2	(I)	入力. 変換する波数区間の最大値 (SHTSWA を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
W	(R)	出力. ウエーブデータ. (長さおよび並び方は SHTS2W を参照).
G	(R)	出力. グリッドデータ. (長さおよび並び方は SHTW2G を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考
   (a) 本サブルーチンは, SHTSWA, SHTWGA を連続して呼ぶのと全く同様である.

4.3.13 SHTSWM

1. 機能

   スペクトル逆変換の前半部分であるルジャンドル逆変換を指定された一つの波数成分のみについて行う.

2. 定義

   スペクトルデータ S_n^m から, ウエーブデータ $W^m(\varphi)$ への変換を指定された一つの波数成分 $m =$M(> 0) のみについて行う. 変換式は SHTS2W を参照.

3. 呼び出し方法

   `SHTSWM(MM,JM,M,ISW,S,WR,WI,WORK)`

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
M	(I)	入力. 変換する波数 (上記定義を参照)
ISW	(I)	入力. 変換の種類の指定 (SHTS2W を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WR	(R)	出力. $W^m(\varphi)$ の実数部分. 長さ 2*JM+1 の配列 (並び方は備考を参照).
WI	(R)	出力. $W^m(\varphi)$ の虚数部分. 長さ 2*JM+1 の配列 (並び方は備考を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考
   (a) WR(-JM:JM), WI(-JM:JM) と宣言されている場合, WR(J) には $\mathrm{Re}(W^m(\varphi_j))$ が, WI(J) には $\mathrm{Im}(W^m(\varphi_j))$ がそれぞれ格納される.
   (b) M=0(帯状成分) については SHTSWZ を用いること.

4.3.14 SHTWGM

1. 機能
   スペクトル逆変換の後半部分であるフーリエ逆変換を指定された一つの波数成分のみについて行う.
2. 定義
   ウエーブデータ $W^m(\varphi)$ からグリッドデータ $G(\lambda, \varphi)$ への変換を指定された一つの波数成分 $m =$M(> 0) のみについて行う. 変換式は SHTW2G を参照.
3. 呼び出し方法
   SHTWGM(MM,JM,IM,M,WR,WI,G,WORK)
4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
IM	(I)	入力. 東西分割数の 1/2(I)
M	(I)	入力. 変換する波数 (上記定義を参照)
WR	(R)	入力. $W^m(\varphi)$ の実数部分. (長さおよび並び方は SHTSWM を参照).
WI	(R)	入力. $W^m(\varphi)$ の虚数部分. (長さおよび並び方は SHTSWM を参照).
G	(R)	出力. グリッドデータ (長さおよび並び方は SHTW2G を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考
   (a) M=0(帯状成分) については SHTSWZ を用いること.

4.3.15 SHTSGM

1. 機能
   SHTSWM, SHTWGM を連続して行うことにより, スペクトル逆変換を指定された一つの波数成分のみについて行う.

2. 定義

   SHTSWM, SHTWGM を参照.

3. 呼び出し方法

   `SHTSGM(MM,JM,IM,M,ISW,S,WR,WI,G,WORK)`

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
IM	(I)	入力. 東西分割数の 1/2(I)
M	(I)	入力. 変換する波数 (SHTSWM を参照)
ISW	(I)	入力. 変換の種類の指定 (SHTS2G を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WR	(R)	出力. $W^m(\varphi)$ の実数部分. (長さおよび並び方は SHTSWM を参照).
WI	(R)	出力. $W^m(\varphi)$ の虚数部分. (長さおよび並び方は SHTSWM を参照).
G	(R)	出力. グリッドデータ (長さおよび並び方は SHTW2G を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) 本サブルーチンは, SHTSWM, SHTWGM を連続して呼ぶのと全く同様である.

4.3.16 SHTSWZ

1. 機能

   スペクトル逆変換の前半部分であるルジャンドル逆変換を帯状成分のみについて行う.

2. 定義

   スペクトルデータ S_n^m から, ウエーブデータ $W^m(\varphi)$ への変換を帯状成分 $m=0$ のみについて行う. 変換式は SHTS2W を参照.

3. 呼び出し方法

   `SHTSWZ(MM,JM,ISW,S,WZ,WORK)`

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
ISW	(I)	入力. 変換の種類の指定 (SHTS2W を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WZ	(R)	出力. $W^0(\varphi)$ が格納される. 長さ 2*JM+1 の配列 (並び方は備考を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) WZ(-JM:JM) と宣言されている場合, WZ(J) には $W^0(\varphi_j)$ が格納される.

4.3.17 SHTWGZ

1. 機能

   スペクトル逆変換の後半部分であるフーリエ逆変換を帯状成分のみについて行う.

2. 定義

   ウエーブデータ $W^m(\varphi)$ からグリッドデータ $G(\lambda,\varphi)$ への変換を帯状成分 $m=0$ のみについて行う. 変換式は SHTW2G を参照.

3. 呼び出し方法

   SHTWGZ(JM,IM,WZ,G)

4. パラメーターの説明

JM	(I)	入力. 南北分割数の $1/2(J)$
IM	(I)	入力. 東西分割数の $1/2(I)$
WZ	(R)	入力. ウエーブデータ $(W^0(\varphi))$ (長さおよび並び方は SHTSWZ を参照).
G	(R)	出力. グリッドデータ (長さおよび並び方は SHTW2G を参照).

5. 備考

   (a) なし.

4.3.18 SHTSGZ

1. 機能

   SHTSWZ, SHTWGZ を連続して行うことにより, スペクトル逆変換を帯状成分のみについて行う.

2. 定義

   SHTSWZ, SHTWGZ を参照.

3. 呼び出し方法

   SHTSGZ(MM,JM,IM,ISW,S,WZ,G,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の $1/2(J)$
IM	(I)	入力. 東西分割数の $1/2(I)$
ISW	(I)	入力. 変換の種類の指定 (SHTS2G を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WZ	(R)	出力. ウエーブデータ $(W^0(\varphi))$ (長さおよび並び方は SHTSWZ を参照).
G	(R)	出力. グリッドデータ (長さおよび並び方は SHTW2G を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) 本サブルーチンは, SHTSWZ, SHTWGZ を連続して呼ぶのと全く同様である.

4.3.19 SHTSWZ

1. 機能

   スペクトル逆変換の前半部分であるルジャンドル逆変換を指定された一つの緯度円上で指定された波数区間のみについて行う.

2. 定義

   スペクトルデータ S_n^m から, ウエーブデータ $W^m(\varphi)$ 指定された一つの緯度円 φ_j 上で M1$\leq |m| \leq$M2 の波数範囲のみについて行う. 変換式は SHTS2W を参照.

3. 呼び出し方法

   SHTSWJ(MM,JM,ISW,J,M1,M2,S,WJ,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
ISW	(I)	入力. 変換の種類の指定 (SHTS2W を参照).
J	(I)	入力. 変換を行う緯度円の指定 (備考を参照).
M1	(I)	入力. 変換する波数区間の最小値 (上記定義を参照)
M2	(I)	入力. 変換する波数区間の最大値 (上記定義を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WJ	(R)	出力. $W^m(\varphi_j)$ が格納される. 長さ 2*MM+1 の配列 (並び方は備考を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) WJ(-MM:MM) と宣言しておけば, WJ(M) (M>0) には $\mathrm{Re}(W^m(\varphi_j))$ が, WJ(-M) (M>0) には $\mathrm{Im}(W^m(\varphi_j))$ がそれぞれ格納され, WJ(0) には $(W^0(\varphi_j))$ が格納される ($j =$J).

   (b) 指定された波数区間の外すなわち WJ(M) (|M|<M1, |M|>M2) には 0 が代入される.

4.3.20 SHTWGJ

1. 機能

   スペクトル逆変換の後半部分であるフーリエ逆変換を一つの緯度円上のウエーブデータに対して指定された波数区間のみについて行う.

2. 定義

   一つの緯度円 φ_j 上において, ウエーブデータ $W^m(\varphi_j)$ からグリッドデータ $G(\lambda,\varphi_j)$ への変換を指定された M1$\leq |m| \leq$M2 の波数範囲のみについて行う. 変換式は SHTW2G を参照.

3. 呼び出し方法

   SHTWGJ(MM,IM,M1,M2,WJ,GJ,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
IM	(I)	入力. 東西分割数の 1/2(I).
M1	(I)	入力. 変換する波数区間の最小値 (上記定義を参照).
M2	(I)	入力. 変換する波数区間の最大値 (上記定義を参照).
WJ	(R)	入力. $W^m(\varphi_j)$ (長さおよび並び方は SHTSWJ を参照).
GJ	(R)	出力. グリッドデータ. 長さ 2*IM+1 の配列 (並び方は備考を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) GJ(-IM:IM) と宣言しておけば, GJ(I) には $G(\lambda_i,\varphi_j)$ が格納される.

4.3.21 SHTSGJ

1. 機能

SHTSWJ, SHTWGJ を連続して行うことにより, スペクトル逆変換を指定された一つの緯度円上で指定された波数区間のみについて行う.

2. 定義

SHTSWJ, SHTWGJ を参照.

3. 呼び出し方法

SHTSGJ(MM,JM,IM,ISW,J,M1,M2,S,WJ,GJ,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
IM	(I)	入力. 東西分割数の 1/2(I)
ISW	(I)	入力. 変換の種類の指定 (SHTS2G を参照).
J	(I)	入力. 変換を行う緯度円の指定 (SHTSWJ を参照).
M1	(I)	入力. 変換する波数区間の最小値 (SHTSWJ を参照)
M2	(I)	入力. 変換する波数区間の最大値 (SHTSWJ を参照)
S	(R)	入力. スペクトルデータ (長さおよび並び方は SHTNML を参照).
WJ	(R)	出力. $W^m(\varphi_j)$ (長さおよび並び方は SHTSWJ を参照).
GJ	(R)	出力. グリッドデータ. (長さおよび並び方は SHTWGJ を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

(a) 本サブルーチンは, SHTSWJ, SHTWGJ を連続して呼ぶのと全く同様である.

4.3.22 SHTFUN

1. 機能

ルジャンドル陪関数を計算する.

2. 定義

指定された帯状波数 $0 \leq m \leq M$ のルジャンドル陪関数;

$$P_n^m(\sin\varphi) \quad (n = m, m+1, \cdots, M) \tag{4.35}$$

を求める (ルジャンドル陪関数の定義は概要を参照).

3. 呼び出し方法

SHTFUN(MM,JM,M,FUN,WORK)

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
M	(I)	入力. 求めるルジャンドル陪関数の帯状波数 (m).
FUN	(R)	$P_n^m(\sin\varphi)$ が格納される長さ (2*JM+1)*(MM-M+1) の配列 (並び方は備考を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

(a) FUN(-JM:JM,M:MM) と宣言されている場合, FUN(J,N) には $P_n^m(\sin\varphi_j)$ が格納される.

4.3.23 SHTLFW

1. 機能

   ルジャンドル正変換を行う.

2. 定義

   帯状波数 $m(\geq 0)$ の実ウエーブデータ $W^m(\varphi)$ に対して,

   ISW=0 の場合, 通常のルジャンドル正変換;

$$S_n^m = \frac{1}{2}\int_{-\pi/2}^{\pi/2} W^m(\varphi)P_n^m(\sin\varphi)\cos\varphi d\varphi \tag{4.36}$$

   を行う.

   ISW=1 の場合, 緯度微分のルジャンドル正変換;

$$S_n^m = \frac{1}{2}\int_{-\pi/2}^{\pi/2} \frac{d}{d\varphi}\left\{W^m(\varphi)\right\} P_n^m(\sin\varphi)\cos\varphi d\varphi \tag{4.37}$$

   を行う.

   ISW=-1 の場合, 経度微分のルジャンドル正変換;

$$S_n^m = \frac{1}{2}\int_{-\pi/2}^{\pi/2} \frac{-m}{\cos\varphi} W^m(\varphi)P_n^m(\sin\varphi)\cos\varphi d\varphi \tag{4.38}$$

   を行う.

3. 呼び出し方法

   `SHTLFW(MM,JM,M,ISW,WM,SM,WORK)`

4. パラメーターの説明

MM	(I)	入力. 切断波数 (M).
JM	(I)	入力. 南北分割数の 1/2(J)
ISW	(I)	入力. 変換の種類の指定 (上記定義を参照)
M	(I)	入力. 変換を行う帯状波数 (m).
WM	(R)	入力. ウエーブデータ. (長さおよび並び方は SHTLBW を参照).
SM	(R)	出力. スペクトルデータ. (長さおよび並び方は SHTLBW を参照).
WORK	(R)	SHTINT で初期化された作業領域.

5. 備考

   (a) ISW=0 の場合, ウエーブデータ WM が SHTLBW で ISW=0 として作成されたものである場合には, 上記の積分が aliasing なしに完全に評価される.

   (b) ISW= ±1 の場合, ウエーブデータ WM が SHTLBW で ISW= ±1 として作成されたものである場合には, 上記の積分が aliasing なしに完全に評価される.

   (c) M≥ 0 であること.

4.3.24 SHTLBW

1. 機能

   ルジャンドル逆変換を行う.

2. 定義

   帯状波数 $m(\geq 0)$ の実スペクトルデータ $S_n^m(n = m, m+1, \cdots, M)$ に対して,

   `ISW=0` の場合, 通常のルジャンドル逆変換;

$$W^m(\varphi) \equiv \sum_{n=|m|}^{M} S_n^m P_n^m(\sin\varphi) \tag{4.39}$$

   を行う.

   `ISW=1` の場合, 緯度微分のルジャンドル逆変換;

$$W^m(\varphi) \equiv \frac{d}{d\varphi} \sum_{n=|m|}^{M} S_n^m P_n^m(\sin\varphi) \tag{4.40}$$

   を行う.

   `ISW=-1` の場合, 経度微分のルジャンドル逆変換;

$$W^m(\varphi) \equiv \frac{m}{\cos\varphi} \sum_{n=|m|}^{M} S_n^m P_n^m(\sin\varphi) \tag{4.41}$$

   を行う.

3. 呼び出し方法

   `SHTLBW(MM,JM,M,ISW,SM,WM,WORK)`

4. パラメーターの説明

`MM`	(I)	入力. 切断波数 (M).
`JM`	(I)	入力. 南北分割数の 1/2(J)
`ISW`	(I)	入力. 変換の種類の指定 (上記定義を参照)
`M`	(I)	入力. 変換を行う帯状波数 (m).
`SM`	(R)	入力. スペクトルデータ. 長さ `MM-M+1` の配列 (並び方は備考を参照).
`WM`	(R)	出力. ウエーブデータ. 長さ `(2*JM+1)` の配列 (並び方は備考を参照).
`WORK`	(R)	`SHTINT` で初期化された作業領域.

5. 備考

   (a) `SM(M:MM)` と宣言した場合, `SM(N)` には S_n^m を格納すれば良い. また, `WM(-JM:JM)` と宣言した場合, `WM(J)` には $W^m(\sin\varphi_j)$ が格納される.

   (b) `M`≥ 0 であること.

第5章　VSTLIB：ベクトルデータの統計処理

5.1　概要

1 種類または 2 種類のベクトルデータを連続的に処理して各種統計量を求めるサブルーチンパッケージ.

このパッケージは GL*p*GET/GL*p*SET (MATH1 参照) の管理する内部変数 LMISS を.TRUE. とすると欠損値処理をおこなう.

5.2　サブルーチンのリスト

1 種類のベクトルデータを処理するサブルーチン.

VS1INT(WZ,NW,IX)	初期化をおこなう.
VS1DIN(WZ,NW,IX,X)	ベクトルデータを読み込む.
VS1OUT(WZ,NW,IX)	結果を求める.

2 種類のベクトルデータを処理するサブルーチン.

VS2INT(WZ,NW,IX,IY)	初期化をおこなう.
VS2DIN(WZ,NW,IX,IY,X,Y)	ベクトルデータを読み込む.
VS2OUT(WZ,NW,IX,IY)	結果を求める.

5.3　サブルーチンの説明

5.3.1　VS1INT/VS1DIN/VS1OUT

1. 機能

   1 種類のベクトルデータを連続的に読み込んで平均と分散を求める. VS1INT は初期化をおこなう. VS1DIN はデータを読み込む. VS1OUT は結果を求める.

2. 呼び出し方法

   ```
   CALL VS1INT(WZ,NW,IX)
   CALL VS1DIN(WZ,NW,IX,X)
   CALL VS1OUT(WZ,NW,IX)
   ```

3. パラメーターの説明

WZ (R) 大きさが (IX,2) の 2 次元配列. VS1DIN を呼んでいる間は作業領域として用いられ, VS1OUT を呼ぶと
WZ(i,1) には平均が,
WZ(i,2) には分散が,
返される.

NW (I) 長さが (IX) の 1 次元配列. 何個のベクトルデータを処理したかを保持している.

IX (I) WZ の第 1 次元の寸法, および NW, X の長さ.

X (R) 読み込むベクトルデータ. 長さ IX の配列.

4. 備考
   (a) なし.

5.3.2 VS2INT/VS2DIN/VS2OUT

1. 機能

   2 種類のベクトルデータを連続的に読み込んで平均と分散を求める. VS2INT は初期化をおこなう. VS2DIN はデータを読み込む. VS2OUT は結果を求める.

2. 呼び出し方法

```
CALL VS2INT(WZ,NW,IX,IY)
CALL VS2DIN(WZ,NW,IX,IY,X,Y)
CALL VS2OUT(WZ,NW,IX,IY)
```

3. パラメーターの説明

WZ (R) 大きさが (IX,IY,5) の 3 次元配列. VS2DIN を呼んでいる間は作業領域として用いられ, VS2OUT を呼ぶと
WZ(i,j,1) には X の平均が,
WZ(i,j,2) には Y の平均が,
WZ(i,j,3) には X の分散が,
WZ(i,j,4) には Y の分散が,
WZ(i,j,5) には X と Y の共分散が,
返される.

NW (I) 大きさが (IX,IY) の 2 次元配列. 何個のベクトルデータを処理したかを保持している.

IX (I) WZ の第 1 次元の寸法, NW の第 1 次元の寸法, および X の長さ.

IY (I) WZ の第 2 次元の寸法, NW の第 2 次元の寸法, および Y の長さ.

X (R) 読み込むベクトルデータ. 長さ IX の配列.

Y (R) 読み込むベクトルデータ. 長さ IY の配列.

4. 備考
   (a) なし.

第6章　INTRLIB：補間

6.1　概要

補間をするサブルーチンパッケージ.

6.2　サブルーチンのリスト

VRINTR(RX,N,JX)	実数型配列の補間をする.
VCINTR(RX,N,JX)	複素数型配列の補間をする.

6.3　サブルーチンの説明

6.3.1　VRINTR

1. 機能

   実数型配列 (r_i) において GLpGET/GLpSET (MATH1 参照) が管理する内部変数'RMISS' と等しい実数値の部分を $r_{i+1} = r_i + r_c$ (ただし r_c は実定数) となるように補間する (線形補間).

2. 呼び出し方法

   CALL VRINTR(RX,N,JX)

3. パラメーターの説明

RX	(R)	処理する実数型の配列. 入力パラメータでもあり, 出力パラメータでもある.
N	(I)	処理する配列要素の個数.
JX	(I)	処理する配列要素の間隔.

4. 備考

   (a) 処理するデータ列の始めまたは終りの部分に'RMISS' であらわされる実数値がある場合, その部分の処理はおこなわれない.

6.3.2　VCINTR

1. 機能

   複素数型配列 (z_i) において実部および虚部の両方が GLpGET/GLpSET (MATH1 参照) が管理する内部変数'RMISS' と等しい実数値の部分を $z_{i+1} = z_c z_i$ (ただし z_c は複素定数) となるように補間する.

2. 呼び出し方法

   CALL VCINTR(CX,N,JX)

3. パラメーターの説明

`CX`	(C)	処理する複素数型の配列. 入力パラメータでもあり, 出力パラメータでもある.
`N`	(I)	処理する配列要素の個数.
`JX`	(I)	処理する配列要素の間隔.

4. 備考

(a) 処理するデータ列の始めまたは終りの部分に実部・虚部とも`'RMISS'`であらわされる複素数値がある場合, その部分の処理はおこなわれない.

第7章　RNMLIB：移動平均

7.1　概要

移動平均を計算するサブルーチンパッケージ.

7.2　サブルーチンのリスト

`VRRNM(RX,RY,N,JX,JY,NB)`	移動平均を計算する (`GLpGET/GL`p`SET` の管理する内部変数`'LMISS'` が`.TRUE.` なら欠損値処理をする; `.FALSE.` なら欠損値処理をしない).
`VRRNM0(RX,RY,N,JX,JY,NB)`	移動平均を計算する (欠損値処理をしない).
`VRRNM1(RX,RY,N,JX,JY,NB)`	移動平均を計算する (欠損値処理をする).

7.3　サブルーチンの説明

7.3.1　VRRNM/VRRNM0/VRRNM1

1. 機能

   移動平均を計算する. `VRRNM` は `GLpGET/GL`p`SET` の管理する内部変数`'LMISS'` が`.TRUE.` なら欠損値処理をする; `.FALSE.` なら欠損値処理をしない. `VRRNM0` は欠損値処理をしない. `VRRNM1` は欠損値処理をする. 欠損値は `GLpGET/GL`p`SET` の管理する内部変数`'RMISS'` で決まる.

2. 呼び出し方法

```
CALL VRRNM(RX,RY,N,JX,JY,NB)
CALL VRRNM0(RX,RY,N,JX,JY,NB)
CALL VRRNM1(RX,RY,N,JX,JY,NB)
```

3. パラメーターの説明

`RX`	`(R)`	移動平均を計算する実数型の配列.
`RY`	`(R)`	計算結果を納める実数型の配列.
`N`	`(I)`	処理する配列要素の個数.
`JX`, `JY`	`(I)`	配列 `RX`, `RY` において, 処理する配列要素の間隔.
`NB`	`(I)`	移動平均をとるデータ長.

4. 備考

   (a) `NB` は 1 以上 `N` 以下, かつ奇数でなければならない.

   (b) `RX`, `RY` において, 処理する第 i 番めの要素を RX_i, RY_i, また $M = (\mathtt{NB} - 1)/2$ とおくと, 移動平均の結果は, $i = M + 1$ から $\mathtt{N} - M$ までについては, $RY_i = (\sum_{i=-M}^{M} RX_i)/\mathtt{NB}$, それ以外については欠損値 (`GLRGET/GLRSET` の管理する内部変数`'RMISS'` で決まる) が代入される. 欠損値処理をす

るような設定になっているときは, 欠損値でないデータについて平均値を計算する. このとき上式の分母にあらわれる NB は欠損値ではないデータの個数となる. また, 移動平均を計算する配列要素すべてが欠損値のときは欠損値が返される.