

SNPACK 使用の手引 (version 0.2)

石岡 圭一 (2005/07/01)

1 概要

1.1 パッケージについて

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格子点値、およびその逆の変換を行なうサブルーチンなどからなっている。

SNPACK は STPACK と SMPACK のハイブリッド的な後継パッケージとなっている。特長は、

- STPACK 並の省メモリ (切断波数の 2 乗オーダーまでの領域しか必要としない。ちなみに SMPACK では切断波数の 3 乗オーダーの領域が必要)。
- SMPACK 並の高速性 (SMPACK と同様に、多層モデルのように複数の変換を同時に実行する際にベクトル長を長くにとって高速化することが可能)。
- SMPACK と同様に入力保存される。

ということである。速度では以前の SMPACK に若干及ばないこともあるが (富士通 VX や NEC の SX4 等で計測したところでは SMPACK より SNPACK の方が最大で 1 割程遅い程度)、その省メモリ性を活用すれば、複数の変数をついてまとめるなどして SMPACK を凌駕することも可能はずである。

また、このパッケージは FTPACK と BSPACK の上位パッケージであり、これらのパッケージを内部で引用している。

なお、SNPACK は SMPACK の後継パッケージではあるが、配列中のデータの並べ方や正規化の仕方が異っているので、乗り換える場合は注意すること。

1.2 球面調和関数変換について

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:

$$g(\lambda, \varphi) = \sum_{n=0}^M \sum_{m=-n}^n s_n^m P_n^m(\sin \varphi) e^{im\lambda}. \quad (1)$$

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:

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

$$\int_{-1}^1 \{P_n^m(\mu)\}^2 d\mu = 2. \quad (3)$$

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

$$s_n^{-m} = \{s_n^m\}^* \quad (4)$$

ここに、 $\{\}^*$ は複素共役を表す。従って、 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、 s_n^0 は実数である。

展開係数に対する上の制約 (??) により、逆変換の式 (??) は以下のようにも表せる:

$$g(\lambda, \varphi) = \sum_{n=0}^M \left[a_n^0 P_n^0(\sin \varphi) + \sum_{m=1}^n \left\{ a_n^m P_n^m(\sin \varphi) \sqrt{2} \cos(m\lambda) - b_n^m P_n^m(\sin \varphi) \sqrt{2} \sin(m\lambda) \right\} \right]. \quad (5)$$

ここに、

$$a_n^0 = \text{Re}(s_n^0), \quad a_n^m = \sqrt{2} \text{Re}(s_n^m), \quad b_n^m = \sqrt{2} \text{Im}(s_n^m), \quad (m = 1, \dots, n; n = 0, \dots, M), \quad (6)$$

であり、随所にファクター $\sqrt{2}$ が掛かっているのは、正規化のためである。

また、スペクトル正変換は以下のように表せる:

$$s_n^m = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} g(\lambda, \varphi) P_n^m(\sin \varphi) e^{-im\lambda} \cos \varphi d\varphi d\lambda. \quad (7)$$

数値計算においては、上記の積分はそれぞれ離散近似される。フーリエ正変換の部分:

$$G^m(\varphi) = \frac{1}{2\pi} \int_0^{2\pi} g(\lambda, \varphi) e^{-im\lambda} d\lambda, \quad (8)$$

は経度方向の等間隔格子点上での値を用いた離散フーリエ正変換 (FTPACK マニュアルを参照) によって近似し、ルジャンドル正変換の部分:

$$s_n^m = \frac{1}{2} \int_{-\pi/2}^{\pi/2} G^m(\varphi) P_n^m(\sin \varphi) \cos \varphi d\varphi, \quad (9)$$

は、ガウス-ルジャンドル積分公式により:

$$s_n^m = \frac{1}{2} \sum_{j=1}^J w_j G^m(\varphi_j) P_n^m(\sin \varphi_j) \quad (10)$$

として近似する。ここに、 φ_j はガウス緯度と呼ばれる分点で、ルジャンドル多項式 $P_J(\sin \varphi)$ (ルジャンドル陪関数の定義式中で $m = 0$ とし、正規化係数 ($\sqrt{}$ の部分) を無くしたもの) の J 個の零点を小さい方から順に並べたものであり、 w_j は各分点に対応するガウシアンウェイトと呼ばれる重みで、

$$w_j \equiv \frac{2(1 - \mu_j^2)}{\{JP_{J-1}(\mu_j)\}^2} \quad (11)$$

で与えられる。ここに、 $\mu_j \equiv \sin \varphi_j$ である。ある条件のもとでは、この積分公式は完全な近似、すなわちもとの積分と同じ値を与える。

本ライブラリは、スペクトルデータ (s_n^m (a_n^m, b_n^m で表示)) \rightarrow 格子点上のグリッドデータ ($g(\lambda_i, \varphi_j)$) の逆変換を行うルーチン群、等間隔格子点上のグリッドデータ ($g(\lambda_i, \varphi_j)$) \rightarrow スペクトルデータ (s_n^m (a_n^m, b_n^m で表示)) の正変換を行うルーチン群、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 は全周を等間隔に I -分割した経度で、緯度 φ_j は上述の J 個のガウス緯度である。

また、SNPACK は多層用のルーチンであるので、 K を層の数とすると、逆変換は各高度レベル z_k に対応する $s_n^m(z_k)$ からグリッドデータ $g(\lambda_i, \varphi_j, z_k)$ への変換、正変換は $g(\lambda_i, \varphi_j, z_k)$ から $s_n^m(z_k)$ への変換となる。

以下のサブルーチンの説明において、

MM: 切断波数 M
 IM: 東西格子点数 I
 JM: ガウス緯度の個数 J
 KM: 並行して行う変換の個数 (または層の数) K
 N: 全波数 n
 M: 帯状波数 m
 I: 東西格子点の番号 i
 J: ガウス緯度の番号 j
 K: 層の番号 k

なる対応関係がある。なお、SMPACK からの乗り換えユーザは、SMPACK と SNPACK の以下の相異点に注意すること。

- スペクトルデータの並べ方の違い。
- スペクトルデータの正規化の違い。
- ガウス緯度の並べ方の違い。

2 サブルーチンのリスト

SNNM2L スペクトルデータの格納位置の計算
 SNL2NM SNNM2L の逆演算
 SNINIT 初期化 (一層用)
 SNKINI 初期化 (多層化)
 SNTS2G スペクトルデータからグリッドデータへの変換
 SNTG2S グリッドデータからスペクトルデータへの変換
 SNTSOG スペクトルデータからグリッドデータへの変換 (並列版)
 SNTGOS グリッドデータからスペクトルデータへの変換 (並列版)

3 サブルーチンの説明

3.1 SNNM2L

1. 機能全波数と帯状波数からスペクトルデータの格納位置を計算する。
2. 定義

SNPACK において、スペクトルデータ (a_n^m, b_n^m) は概要に述べた制限をもとに、独立な $(M+1)^2$ 個の成分を長さ $(M+1)^2$ の配列に格納して扱う。

格納順は、切断波数に依存しない普遍的な並びになるように以下のような順序になっている:

$$\begin{array}{ccccccc}
& & & a_0^0, & & & n = 0 \text{ 成分} \\
& & b_1^1, & a_1^0, & a_1^1, & & n = 1 \text{ 成分} \\
& b_2^2, & b_2^1, & a_2^0, & a_2^1, & a_2^2, & n = 2 \text{ 成分} \\
& & \vdots & \vdots & \vdots & & \\
& b_n^n, \dots, b_n^1 & & a_n^0 & a_n^1, \dots, a_n^n & & n = n \text{ 成分} \\
& & \vdots & \vdots & \vdots & & \\
b_M^M, \dots, b_M^1 & & a_M^0 & a_M^1, \dots, a_M^M & & & n = M \text{ 成分}
\end{array}$$

配列には、この図の左上から右下への順番に格納されている。

このサブルーチンは、 a_n^m, b_n^m の全波数 n 、および帯状波数 m から a_n^m, b_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

SNNM2L(N,M,L)

4. パラメーターの説明

- N (I) 入力. 全波数
- M (I) 入力. 帯状波数 (備考参照)
- L (I) 出力. スペクトルデータの格納位置

5. 備考

M > 0 なら $m = M$, $n = N$ として a_n^m の格納位置を, M < 0 なら $m = -M$, $n = N$ として b_n^m の格納位置を返す。

3.2 SNL2NM

1. 機能 SNNM2L の逆演算, すなわち, スペクトルデータの格納位置から全波数と帯状波数を求める。

2. 定義

SNNM2L の項を参照

3. 呼び出し方法

SNL2NM(L,N,M)

4. パラメーターの説明

- L (I) 入力. スペクトルデータの格納位置
- N (I) 出力. 全波数
- M (I) 出力. 帯状波数

5. 備考

M の正負についての意味づけは SNNM2L と同じである。

3.3 SNINIT

1. 機能 SNPACK の初期化ルーチン. 変換ルーチン SNT???および多層用初期化ルーチンで用いられる配列 IT,T,Y,IP,P,R,IA,A の初期化を行う.

2. 定義

3. 呼び出し方法

SNINIT(MM,IM,JM,IT,T,Y,IP,P,R,IA,A)

4. パラメーターの説明

MM	(I)	入力. 切断波数
IM	(I)	入力. 東西格子点数
JM	(I)	入力. 南北格子点数
IT	(I(5))	出力. SNT???で用いられる配列
T	(D(IM*2))	出力. SNT???で用いられる配列
Y	(D(JM*2))	出力. SNT???で用いられる配列
IP	(I(((MM+1)/2+MM+1)*2))	出力. SNT???で用いられる配列
P	(D(((MM+1)/2+MM+1)*JM))	出力. SNT???で用いられる配列
R	(D(((MM+1)/2*2+3)*(MM/2+1)))	出力. SNT???で用いられる配列
IA	(I((MM+1)*(MM+1)*4))	出力. SNT???で用いられる配列
A	(D((MM+1)*(MM+1)*6))	出力. SNT???で用いられる配列

5. 備考

- MM は 2 以上の整数, JM および IM はそれぞれ, 2 以上の偶数でなければならない.
- Y(JM/2,4) と宣言されている場合, $Y(J,1): \sin(\varphi_{J/2+j})$, $Y(J,2): \frac{1}{2}w_{J/2+j}$, $Y(J,3): \cos(\varphi_{J/2+j})$, $Y(J,4): 1/\cos(\varphi_{J/2+j})$, が格納される.

3.4 SNKINI

1. 機能 SNPACK の初期化ルーチン. 多層の場合 (KM>1) に変換ルーチン SNT???で用いられる配列 IPK,PK,RK の初期化を行う.

2. 定義

3. 呼び出し方法

SNKINI(MM,JM,KM,IP,P,R,IPK,PK,RK)

4. パラメーターの説明

MM	(I)	入力. 切断波数
JM	(I)	入力. 南北格子点数
KM	(I)	入力. 並行して行う変換の個数 (または層の数)
IP	(I(((MM+1)/2+MM+1)*2))	入力. SNINIT で初期化された配列
P	(D(((MM+1)/2+MM+1)*JM))	入力. SNINIT で初期化された配列
R	(D(((MM+1)/2*2+3)*(MM/2+1)))	入力. SNINIT で初期化された配列
IPK	(I(KM*((MM+1)/2+MM+1)*2))	出力. SNT???で用いられる配列
PK	(D(KM*((MM+1)/2+MM+1)*JM))	出力. SNT???で用いられる配列
RK	(D(KM*((MM+1)/2*2+3)*(MM/2+1)))	出力. SNT???で用いられる配列

5. 備考

後述の変換ルーチン SNT???を一層 (KM= 1) でしか使わない場合には IP,P,R を IPK,PK,RK としてそのまま使えば良いので、このサブルーチン呼び出す必要はない。

3.5 SNTS2G

1. 機能

スペクトルデータからグリッドデータへの変換を行う。

2. 定義

スペクトル逆変換 (概要を参照) によりスペクトルデータ ($a_n^m(z_k), b_n^m(z_k)$) から格子点上のグリッドデータ ($g(\lambda_i, \varphi_j, z_k)$) を求める。

3. 呼び出し方法

SNTS2G(MM, IM, ID, JM, JD, KM, S, G, IT, T, Y, IPK, PK, RK, IA, A, Q, WS, WW, IPOW, IFLAG)

4. パラメーターの説明

MM	(I)	入力. 切断波数
IM	(I)	入力. 東西格子点数
ID	(I)	入力. G の第 1 次元目の寸法
JM	(I)	入力. 南北格子点数
JD	(I)	入力. G の第 2 次元目の寸法
KM	(I)	入力. 並行して行う変換の個数 (または層の数)
S	(D((MM+1)*(MM+1), KM))	入力. $a_n^m(z_k), b_n^m(z_k)$ が格納されている配列
G	(D(ID, JD, KM))	出力. $g(\lambda_i, \varphi_j, z_k)$ が格納される配列
IT	(I(5))	入力. SNINIT で初期化された配列
T	(D(IM*2))	入力. SNINIT で初期化された配列
IP	(I(((MM+1)/2+MM+1)*2))	入力. SNINIT で初期化された配列
P	(D(((MM+1)/2+MM+1)*JM))	入力. SNINIT で初期化された配列
R	(D((MM+1)/2*2+3)*(MM/2+1))	入力. SNINIT で初期化された配列
IPK	(I(KM*((MM+1)/2+MM+1)*2))	入力. SNKINI で初期化された配列
PK	(D(KM*((MM+1)/2+MM+1)*JM))	入力. SNKINI で初期化された配列
RK	(D(KM*((MM+1)/2*2+3)*(MM/2+1)))	入力. SNKINI で初期化された配列
IA	(I((MM+1)*(MM+1)*4))	入力. SNINIT で初期化された配列
A	(D((MM+1)*(MM+1)*6))	入力. SNINIT で初期化された配列
Q	(D(KM*((MM+1)/2+MM+1)*JM))	作業領域
WS	(D(大きさの決め方は備考参照))	作業領域
WW	(D(大きさの決め方は備考参照))	作業領域
IPOW	(I)	作用させる $1/\cos \varphi$ の次数 (備考参照)
IFLAG	(I)	変換の種類を決めるフラグ (備考参照)

5. 備考

- ID \geq IM, JD \geq JM でなければならない。またベクトル計算機においてはバンク競合を避けるために、ID,JD はできれば奇数にとるのがよい。

- $G(ID, JD, KM)$ と宣言されている場合, $G(I, J, K)$ には $g(\lambda_i, \varphi_j, z_k)$ が格納される.
- $ID > IM$ の場合は, $G(I, J, K)$ ($I > IM$) には $G(1, J, K)$ と同じ値が代入される. また, $JD > JM$ の場合は, $G(I, J, K)$ ($J > JM$) には $G(I, JM, K)$ と同じ値が代入される.
- $MM \geq IM/2$ の場合には, $m \leq IM/2 - 1$ までの成分が使われる.
- 作業領域 WS, WW の大きさは, ともに
 $KM * \text{MAX}(((MM+1)/2 * 2 + 3) * (MM/2 + 2) * 2, JD * ((MM+1)/2 + MM + 1) * 2, JD * IM)$ 以上の大きさであること. または, さらに単純には, やや余裕をもって, $KM * \text{MAX}((MM+4) * (MM+3), JD * 3 * (MM+1), JD * IM)$ としておいてもよい. ただし, Fortran90 など動的に領域を確保するのでない限り, PARAMETER 文中でこのような値を MM, JD, IM から自動的に設定できるようにするのは難しい (MAX のような関数が使えないため). しかし, 実際的な場合を考えると, 以下のように簡単に設定できる場合が多い筈である. ($MM \geq 3$ としておく).
- 逆変換 正変換 で元のスペクトルデータへの復元が保証される条件
 $(JD \geq JM \geq MM + 1, IM \geq 2 * MM + 2 \text{ (IM は偶数より)})$ が満たされている場合. $(IM + MM + 1) * JD * KM$ なる大きさにとればよい.
- 2 次の非線形項からのエリアジングを除く条件 $(JD \geq JM \geq 3 * MM / 2D0, IM \geq 3 * MM + 1)$ が満たされている場合. $(IM + 2) * JD * KM$ なる大きさにとればよい.
- IFLAG は逆変換の種類を示し, IPOW はその際に同時に作用させる $1/\cos \varphi$ の次数を示す. $IPOW = l$ と書いておけば, IFLAG の値によって,

- IFLAG = 0: 通常の逆変換.

$$g(\lambda, \varphi) = \frac{1}{\cos^l \varphi} \sum_{n=0}^M \sum_{m=-n}^n s_n^m P_n^m(\sin \varphi) e^{im\lambda}. \quad (12)$$

- IFLAG = 1: 緯度微分を作用させた逆変換.

$$g(\lambda, \varphi) = \frac{1}{\cos^l \varphi} \sum_{n=0}^M \sum_{m=-n}^n s_n^m \cos \varphi \frac{\partial}{\partial \varphi} P_n^m(\sin \varphi) e^{im\lambda}. \quad (13)$$

- IFLAG = -1: 経度微分を作用させた逆変換.

$$g(\lambda, \varphi) = \frac{1}{\cos^l \varphi} \sum_{n=0}^M \sum_{m=-n}^n s_n^m P_n^m(\sin \varphi) \frac{\partial}{\partial \lambda} e^{im\lambda}. \quad (14)$$

- IFLAG = 2: $\sin \varphi$ を作用させた逆変換.

$$g(\lambda, \varphi) = \frac{1}{\cos^l \varphi} \sum_{n=0}^M \sum_{m=-n}^n s_n^m \sin \varphi P_n^m(\sin \varphi) e^{im\lambda}. \quad (15)$$

従って, 勾配を求めるためにこのルーチンを使う場合は, $IFLAG = \pm 1, IPOW = 1$ として呼出せばよい.

3.6 SNTG2S

1. 機能

グリッドデータからスペクトルデータへの変換を行う.

2. 定義

スペクトル正変換 (概要を参照) により格子点上のグリッドデータ ($g(\lambda_i, \varphi_j, z_k)$) からスペクトルデータ ($a_n^m(z_k), b_n^m(z_k)$) を求める.

3. 呼び出し方法

SNTG2S(MM, IM, ID, JM, JD, KM, G, S, IT, T, Y, IPK, PK, RK, IA, A, Q, WS, WW, IPOW, IFLAG)

4. パラメーターの説明

MM	(I)	入力. 切断波数
IM	(I)	入力. 東西格子点数
ID	(I)	入力. G の第 1 次元目の寸法
JM	(I)	入力. 南北格子点数
JD	(I)	入力. G の第 2 次元目の寸法
KM	(I)	入力. 並行して行う変換の個数 (または層の数)
G	(D(ID, JD, KM))	入力. $g(\lambda_i, \varphi_j, z_k)$ が格納されている配列
S	(D((MM+1)*(MM+1), KM))	出力. $a_n^m(z_k), b_n^m(z_k)$ が格納される配列
IT	(I(5))	入力. SNINIT で初期化された配列
T	(D(IM*2))	入力. SNINIT で初期化された配列
IP	(I(((MM+1)/2+MM+1)*2))	入力. SNINIT で初期化された配列
P	(D(((MM+1)/2+MM+1)*JM))	入力. SNINIT で初期化された配列
R	(D(((MM+1)/2*2+3)*(MM/2+1)))	入力. SNINIT で初期化された配列
IPK	(I(KM*(MM+1)/2+MM+1)*2)	入力. SNKINI で初期化された配列
PK	(D(KM*((MM+1)/2+MM+1)*JM))	入力. SNKINI で初期化された配列
RK	(D(KM*((MM+1)/2*2+3)*(MM/2+1)))	入力. SNKINI で初期化された配列
IA	(I((MM+1)*(MM+1)*4))	入力. SNINIT で初期化された配列
A	(D((MM+1)*(MM+1)*6))	入力. SNINIT で初期化された配列
Q	(D((KM*(MM+1)/2+MM+1)*JM))	作業領域
WS	(D(大きさの決め方は備考参照))	作業領域
WW	(D(大きさの決め方は備考参照))	作業領域
IPOW	(I)	作用させる $1/\cos \varphi$ の次数 (備考参照)
IFLAG	(I)	変換の種類を決めるフラグ (備考参照)

5. 備考

- ID, JD の設定などについては SNTS2G の備考を参照.
- G(ID, JD, KM) と宣言されている場合, G(I, J, K) には $g(\lambda_i, \varphi_j, z_k)$ が格納されていること. ID>IM の場合は, G(I, J, K) (I>IM) には何が入っていてもよい. また, JD>JM の場合は, G(I, J, K) (J>JM) には何が入っていてもよい.
- $MM \geq IM/2$ の場合には, $m \geq IM/2$ の部分には零が代入される.
- 作業領域 WS, WW の大きさの定め方は SNTS2G の備考を参照.
- IFLAG は正変換の種類を示し, IPOW はその際に同時に作用させる $1/\cos \varphi$ の次数を示す. IPOW = l と書いておけば, IFLAG の値によって,

- IFLAG = 0: 通常の正変換.

$$s_n^m = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} \frac{g(\lambda, \varphi)}{\cos^l \varphi} P_n^m(\sin \varphi) e^{-im\lambda} \cos \varphi d\varphi d\lambda. \quad (16)$$

- IFLAG = 1: 緯度微分を作用させた正変換.

$$s_n^m = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} \frac{\partial}{\cos \varphi \partial \varphi} \left(\cos^2 \varphi \frac{g(\lambda, \varphi)}{\cos^l \varphi} \right) P_n^m(\sin \varphi) e^{-im\lambda} \cos \varphi d\varphi d\lambda. \quad (17)$$

- IFLAG= -1: 経度微分を作用させた正変換.

$$s_n^m = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} \frac{\partial}{\partial \lambda} \left(\frac{g(\lambda, \varphi)}{\cos^l \varphi} \right) P_n^m(\sin \varphi) e^{-im\lambda} \cos \varphi d\varphi d\lambda. \quad (18)$$

- IFLAG= 2: $\sin \varphi$ を作用させた正変換.

$$s_n^m = \frac{1}{4\pi} \int_0^{2\pi} \int_{-\pi/2}^{\pi/2} \sin \varphi \frac{g(\lambda, \varphi)}{\cos^l \varphi} P_n^m(\sin \varphi) e^{-im\lambda} \cos \varphi d\varphi d\lambda. \quad (19)$$

従って、ベクトル場の発散を求めるためにこのルーチンを使う場合は、IFLAG= ± 1 , IPOW= 1 として呼出せば良い。

3.7 SNTSOG

1. 機能

スペクトルデータからグリッドデータへの変換を行う (並列版 = SNTS2G を OpenMP で安直に並列化したもの)。

2. 定義

スペクトル逆変換 (概要を参照) によりスペクトルデータ ($a_n^m(z_k), b_n^m(z_k)$) から格子点上のグリッドデータ ($g(\lambda_i, \varphi_j, z_k)$) を求める。

3. 呼び出し方法

SNTSOG(MM, IM, ID, JM, KM, S, G, IT, T, Y, IPK, PK, RK, IA, A, Q, WS, WW, WV, IPOW, IFLAG)

4. パラメーターの説明 (SNTS2G と異なるものだけ述べる)

G	(D(ID, JM, KM))	出力. $g(\lambda_i, \varphi_j, z_k)$ が格納されている配列
WS	(D(KM*(IM+MM+1)*3*JM/2))	作業領域
WW	(D(KM*(IM+MM+1)*3*JM/2))	作業領域
WV	(D(大きさの決め方は備考参照))	作業領域

5. 備考

- IM \geq 2*MM+2 かつ JM \geq MM+1 であること.
- ID \geq IM でなければならない. またベクトル計算機においてはバンク競合またはキャッシュミス为了避免のために, ID はできれば奇数にとるのがよい.
- G(ID, JM, KM) と宣言されている場合, G(I, J, K) には $g(\lambda_i, \varphi_j, z_k)$ が格納される.
- ID>IM の場合は, G(I, J, K) (I>IM) には G(1, J, K) と同じ値が代入される.
- 作業領域 WV の大きさは, KM*(MM+4)*(MM+3)*NP 以上の大きさであること. ここに, NP は OpenMP で利用しうる最大のスレッド数である.

3.8 SNTGOS

1. 機能

グリッドデータからスペクトルデータへの変換を行う (並列版 = SNTG2S を OpenMP で安直に並列化したもの)。

2. 定義

スペクトル正変換 (概要を参照) により格子点上のグリッドデータ ($g(\lambda_i, \varphi_j, z_k)$) からスペクトルデータ ($a_n^m(z_k), b_n^m(z_k)$) を求める.

3. 呼び出し方法

SNTGOS(MM, IM, ID, JM, KM, G, S, IT, T, Y, IPK, PK, RK, IA, A, Q, WS, WW, WV, IPOW, IFLAG)

4. パラメーターの説明 (SNTG2S と異なるものだけ述べる)

G	(D(ID, JM, KM))	入力. $g(\lambda_i, \varphi_j, z_k)$ を格納される配列
WS	(D(KM*(IM+MM+1)*3*JM/2))	作業領域
WW	(D(KM*(IM+MM+1)*3*JM/2))	作業領域
WV	(D(大きさの決め方は備考参照))	作業領域

5. 備考

- $IM \geq 2*MM+2$ かつ $JM \geq MM+1$ であること.
- $ID \geq IM$ でなければならない. またベクトル計算機においてはバンク競合またはキャッシュミス为了避免のために, ID はできれば奇数にとるのがよい.
- $G(ID, JM, KM)$ と宣言されている場合, $G(I, J, K)$ には $g(\lambda_i, \varphi_j, z_k)$ が格納されていること. $ID > IM$ の場合は, $G(I, J, K)$ ($I > IM$) には何が入っていてもよい.
- 作業領域 WV の大きさは, $KM*(MM+4)*(MM+3)*NP$ 以上の大きさであること. ここに, NP は OpenMP で利用しうる最大のスレッド数である.