AGCM5

SMPACK 対応部分 解説

地球流体電脳倶楽部

1998 年 3 月 8 日 (AGCM5-3)

目 次

1 はじめに **1**

2 使用方法 **2**
2.1 SMPACK および ISPACK の入手 . . . 2
2.2 ISPACK および SMPACK のインストール . . . 2
2.3 AGCM5 ライブラリおよび実行ファイルの作成 . . . 2

3 変更点 **4**
3.1 ディレクトリ構成 . . . 4
3.2 新規作成したファイル . . . 4
3.3 変更したファイル . . . 4
3.4 廃止したファイル . . . 5

4 コード解説 **6**
4.1 SMPACK 関連の全体の流れ . . . 6
4.2 多層球関数変換 . . . 6
4.2.1 DSMINI . . . 7
4.2.2 DSMG2S . . . 7
4.2.3 DSMV2S . . . 8
4.2.4 DSMV1S . . . 8
4.2.5 DSMS2G . . . 9
4.2.6 DSMS2V . . . 9
4.3 1 層球関数変換 . . . 10
4.3.1 DS1INI . . . 11
4.3.2 DS1G2S . . . 11
4.3.3 DS1S2G . . . 11
4.3.4 DS1S1V . . . 12

A ベンチマークテスト **1**
A.1 比較実験の概要 . . . 1
A.2 数値の比較 . . . 1
A.3 CPU タイムの比較 . . . 1

B 謝辞 **3**

第 1 章

はじめに

AGCM5 パッケージを改良し, 球関数ライブラリ SMPACK を利用できるようにした. 改訂にあたっては, 従来の利用環境を完全に守った 「上位互換」パッケージとなるように留意した. 本ドキュメントでは, SMPACK を利用する方法およびコードの概略を述べる. なお, 解説は AGCM5 の構造を熟知しているユーザを念頭に行なっているため, 必要に応じて適宜他のドキュメントを参照していただきたい.

SMPACK は石岡圭一氏によって製作されたパッケージであり, 下位ルーチンとして同じく石岡圭一氏の ISPACK ライブラリ中の FTPACK および BSPACK を利用している. これらのライブラリの著作権は石岡圭一氏が所有しており, 配布制限などについては GPL に従わなければならない. また, 数値実験の結果等を論文に発表する際は, SMPACK を用いたことを明記しなくてはならない. 詳細はパッケージに附属している SMPACK 利用の手引きをを参照されたい.

SMPACK を利用して計算できるのは三角切断の 3 次元計算に限定されている. 3 次元セクターモデル計算や x-z, y-z 2 次元モデル計算では従来の球関数ライブラリを用いなければならない.

98/03/08 赤堀浩司

第 2 章

使用方法

2.1 SMPACK および ISPACK の入手

しかるべき場所から SMPACK および ISPACK を入手すること.

2.2 ISPACK および SMPACK のインストール

ISPACK のインストールは, パッケージの中に収められている 『ISPACK 利用の手引き」を参考にして行なう. ISPACK にはさまざまなサブルーチン群が収められているが, AGCM5 で用いるのは FTPACK と BSPACK である. `Mkinclude` ファイルの `DIRS` の項目ではこれら 2 つのパッケージを選択すればよい.

SMPACK のインストールは, パッケージの中に収められている 『SMPACK 利用の手引き」を参考にして行なう. 現在のところ SMPACK-1.0 には `Mkinclude` ファイルが用意されていないので, ISPACK の `Mkinclude` を参考にしてインストール作業を行なう.

2.3 AGCM5 ライブラリおよび実行ファイルの作成

AGCM5 の `Mkinclude` ファイルにおいて, 「spack smpack ライブラリのありか」の項目を編集する. また 『STDLIBSSM」の項目を実状に合せて編集する.

分解能の指定」では次のような smpack 用のファイルを選択する.

```
RESOLUTION = t42l12sm
```

98/03/07 赤堀 浩司

希望するファイルがない場合は include/t42l12sm/zcdim.F 等を参考にファイルを作成すればよい. その際には SMPACK の仕様に従って

```
LMAX = MMAX,
NMAX = LMAX,
MINT = 1,
NMDIM = (MMAX+1)*(MMAX+1)
```

としなければならない[1].

ライブラリの作成は

```
% make libsm
```

によって行う.

実行ファイルの作成や初期値ファイルの作成は次のようにして行なう.

```
% make gcmsm
% make initsm
```

詳細は Makefile を解読されたい[2].

[1] SMPACK では定数 LMAX, NMAX, MINT を利用していないが, これらはスペクトル診断のサブルーチン SPCTLD (dynamics/ddiag.F) で利用するためこのようにあたえる必要がある. SPCTLD を書きかえる方法もあるが, SPCTLD は netCDF を利用した agcm5 において変更されているので, ここでも変更すると混乱すると判断した.

[2] 現在のところ Makefile は一部不完全なまま. ターゲット installsm は作ったものの動かない. ターゲット index, print, pack は何をするのか把握しておらず, なにもしていない.

第 3 章

変更点

3.1 ディレクトリ構成

SMPACK を利用するためのコードは, 専用のディレクトリ

```
src/dynamics_sm
src/admin_sm
src/util_sm
```

に収め, 変更したファイルを明確に認識できるようにしてある. これらはそれぞれ dynamics , admin, util の代替ディレクトリであり, 両者で同一のファイルはリンクを張って共有している.

3.2 新規作成したファイル

```
dynamics_sm/
dsm.F          DSMINI, DSMG2S, DSMV2S, DSMV1S, DSMS2G, DSMS2V, DSMGAU   球関数変換 (多層)
ds1.F          DS1INI, DS1G2S, DS1S2G, DS1S1V                           球関数変換 (1 層)
```

3.3 変更したファイル

```
admin_sm/
asetl.F     SETLAT   球関数を初期化. ガウス緯度 ・荷重の計算部分も変更
```

98/03/07 赤堀 浩司

dynamics_sm/		
ddifc.F	CORDIF	球関数部分の変更
dg2wd.F	TENG2W, GD2WD, GUV2VD	球関数部分の変更
dgeng.F	GENGD	球関数部分の変更
dphis.F	SETPHS	球関数部分の変更
dpixy.F	PIXPIY	球関数部分の変更
dsetc.F	DSETC	球関数の準備方法を変更
	GETNMO	新規作成
dstrt.F	STRTV	球関数部分の変更

3.4　廃止したファイル

dynamics_sm/	
dsphe.F	W2G, G2W, SPSTUP
util_sm/	
uspst.F	SETPNM, LGNDRE, GAUSS
usphe.F	SPW2G, SPG2W, SPW2Z, SPZ2W, GRADX
ufftr.F	FFT99X, RFFTIM, RFOURT, CFOURT

第 4 章

コード解説

4.1 SMPACK 関連の全体の流れ

まず, サブルーチン SETLAT (admin_sm/asetl.F) の中でDSMINI (dynamics_sm/dsm.F 新規作成) と DS1INI (dynamics_sm/ds1.F 新規作成) を呼び SMPACK を初期化している. この直後に DSMGAU (dynamics_sm/dsm.F 新規作成) を呼びガウス緯度 ・荷重を得ている[1].

次に, DSETC (dynamics_sm/dsetc.F) を呼び各種定数等を設定する. ここでは従来とは異なり球関数関連の設定は行なっていない. スペクトル添字順番を記憶する配列 NMO は, GETNMO (dynamics_sm/dsetc.F) から得ている。

格子⇔スペクトルの変換は, 多層の変数は DSM??? (dynamics_sm/dsm.F 新規作成) で, 1 層の変数は DS1??? (dynamics_sm/ds1.F 新規作成) で行なう. これらのルーチンの中では, 対応した SMPACK の変換ルーチンを呼んでいる.

4.2 多層球関数変換

多層用球関数変換ルーチン (dynamics_sm/dsm.F 新規作成) は次のようになっている.

```
SUBROUTINE DSMINI !" 球関数 (多層) を初期化
     CALL SMINIT
     RETURN
ENTRY DSMG2S          !" 格子→スペクトル
     CALL SMTG2S
```

98/03/07 赤堀 浩司

[1] わかりやすさを考えればメインプログラムの中で初期化させる方が良いが, その場合には初期値作成用ファイルや軸作成用ファイルも変更しなくてはいけなくなる.

```
      RETURN
ENTRY DSMV2S          !" 格子 (ベクトル) →スペクトル (発散 ・回転)
      CALL SMTV2S
      RETURN
ENTRY DSMV1S          !" 格子 (ベクトル) →スペクトル (発散)
      CALL SMTV1S
      RETURN
ENTRY DSMS2G          !" スペクトル→格子
      CALL SMTS2G
      RETURN
ENTRY DSMS2V          !" スペクトル (発散 ・回転) →格子 (ベクトル)
      CALL SMTS2V
      RETURN
ENTRY  DSMGAU         !" ガウス緯度 ・ガウス荷重の取得
      RETURN
```

以下に詳細を述べる. 説明中のパラメータは次のとおり.

```
      REAL      GDATA( IDIM,JDIM,KMAX )
      REAL      GDATB( IDIM,JDIM,KMAX )
      REAL      WDATA( NMDIM,KMAX )
      REAL      WDATB( NMDIM,KMAX )
      CHRACTER  HFUNC*4
```

4.2.1 DSMINI

1. 機能

   SMPACK のサブルーチン SMINIT を呼んで球関数変換の準備をする. その他 SMPACK の使用条件のチェックと地球半径ファクタの準備もする.

2. 呼び出し方法

   ```
   CALL DSMINI
   ```

3. 備考

   (a) 現在のところ `src/admin_sm/asetl.F` の中で呼んでいる.

4.2.2 DSMG2S

1. 機能

   SMPACK のサブルーチン SMTG2S を呼び, グリッドデータからスペクトルデータへの変換を行なう.

2. 定義

   SMPACK のドキュメントを参照のこと

3. 呼び出し方法

```
 CALL DSMG2S
O          ( WDATA    ,
I            GDATA    )
```

4.2.3 DSMV2S

1. 機能

   SMPACK のサブルーチン SMTV2S を呼び, 格子 (ベクトル) →スペクトル (発散 ・回転) への変換を行なう.

2. 定義

   入力の実空間ベクトルデータ (u, v) に対して, 発散 A および回転 B のスペクトルデータ WDATA および WDATB を求める. ここで A, B はそれぞれ

$$A \equiv \frac{1}{a}\left\{\frac{\partial u}{\cos\varphi\partial\lambda} + \frac{\partial(v\cos\varphi)}{\cos\varphi\partial\varphi}\right\} = \frac{1}{a(1-\mu^2)}\frac{\partial(u\cos\varphi)}{\partial\lambda} + \frac{1}{a}\frac{\partial(v\cos\varphi)}{\partial\mu} \tag{4.1}$$

$$B \equiv \frac{1}{a}\left\{\frac{\partial v}{\cos\varphi\partial\lambda} - \frac{\partial(u\cos\varphi)}{\cos\varphi\partial\varphi}\right\} = \frac{1}{a(1-\mu^2)}\frac{\partial(v\cos\varphi)}{\partial\lambda} - \frac{1}{a}\frac{\partial(u\cos\varphi)}{\partial\mu} \tag{4.2}$$

   と定義する. a は地球半径.

3. 呼び出し方法

```
 CALL DSMV2S
O          ( WDATA , WDATB,
I            GDATA , GDATB,
I            HFUNC  )
```

4. 備考

   (a) HFUNC の 1 文字目が空白のとき, 入力として (u, v) =(GDATA, GDATB) を用いる. 空白でないときは入力として $(u\cos\phi, v\cos\phi)$=(GDATA, GDATB) を用いる.

4.2.4 DSMV1S

1. 機能

   SMPACK のサブルーチン SMTV1S を呼び, 格子 (ベクトル) →スペクトル (発散) への変換を行なう.

2. 定義

   入力の実空間ベクトルデータ (u, v) に対して, 発散 A のスペクトル WDATA を求める. ここで A は

$$A \equiv \frac{1}{a}\left\{\frac{\partial u}{\cos\varphi\partial\lambda} + \frac{\partial(v\cos\varphi)}{\cos\varphi\partial\varphi}\right\} = \frac{1}{a(1-\mu^2)}\frac{\partial(u\cos\varphi)}{\partial\lambda} + \frac{1}{a}\frac{\partial(v\cos\varphi)}{\partial\mu} \tag{4.3}$$

   と定義する. a は地球半径.

3. 呼び出し方法

```
 CALL DSMV1S
M          ( WDATA ,
I            GDATA , GDATB,
I            HFUNC  )
```

4. 備考

   (a) HFUNC の 1 文字目が空白のとき, 入力として (u, v) =(GDATA, GDATB) を用いる. ただし, 空白の場合の演算は AGCM のソースコードでは行なっていない. 空白でないときは入力として $(u\cos\phi, v\cos\phi)$=(GDATA, GDATB) を用いる.

   (b) HFUNC の 4 文字目が S のとき, (出力の WDATA の値) = (入力の WDATA の値) − (変換で得た値) となる. S 以外のときは変換で得た値を出力する. ただし, S 以外の演算は AGCM のソースコードでは行なっていない.

4.2.5 DSMS2G

1. 機能

   SMPACK のサブルーチン SMTS2G を呼び, スペクトル→格子の変換を行なう.

2. 定義

   SMPACK のドキュメントを参照のこと

3. 呼び出し方法

```
 CALL DSMS2G
O          ( GDATA ,
I            WDATA  )
```

4.2.6 DSMS2V

1. 機能

   SMPACK のサブルーチン SMTS2V を呼び, スペクトル (発散 ・回転) →格子 (ベクトル) の変換を行なう.

2. 定義

入力の 2 つのスペクトルデータ WDATA (発散に対応) および DATB (渦度に対応) の逆変換 A, B に対して, 実空間のベクトル (u, v) を出力する. ここで u, v はそれぞれ

$$u \equiv a\left\{\frac{1}{\cos\varphi}\frac{\partial(\nabla^{-2}A)}{\partial\lambda} - \frac{\partial(\nabla^{-2}B)}{\partial\varphi}\right\} = \frac{1}{a\cos\varphi}\left\{\frac{\partial(a^2\nabla^{-2}A)}{\partial\lambda} - (1-\mu^2)\frac{\partial(a^2\nabla^{-2}B)}{\partial\mu}\right\} \tag{4.4}$$

$$v \equiv a\left\{\frac{\partial(\nabla^{-2}A)}{\partial\varphi} + \frac{1}{\cos\varphi}\frac{\partial(\nabla^{-2}B)}{\partial\lambda}\right\} = \frac{1}{a\cos\varphi}\left\{(1-\mu^2)\frac{\partial(a^2\nabla^{-2}A)}{\partial\mu} + \frac{\partial(a^2\nabla^{-2}B)}{\partial\lambda}\right\} \tag{4.5}$$

と定義する. また ∇^2 演算子を次のように定義する.

$$\nabla^2 \equiv \frac{\partial^2}{\cos^2\varphi\partial\lambda^2} + \frac{\partial}{\cos\varphi\partial\varphi}\left(\cos\varphi\frac{\partial}{\partial\varphi}\right) = \frac{1}{1-\mu^2}\frac{\partial^2}{\partial\lambda^2} + \frac{\partial}{\partial\mu}\left\{(1-\mu^2)\frac{\partial}{\partial\mu}\right\} \tag{4.6}$$

3. 呼び出し方法

```
 CALL DSMS2V
O         ( GDATA  , GDATB,
I           WDATA  , WDATB,
I           HFUNC           )
```

4. 備考

(a) HFUNC の 1 文字目が空白のとき (GDATA, GDATB)= (u, v) を出力する. 空白でないときは (GDATA,GDATB)= $(u\cos\varphi, v\cos\varphi)$ を出力する. ただし, 空白でない場合の演算は, AGCM のソースコードでは行なっていない.

4.3 1 層球関数変換

1 層用球関数変換ルーチン (dynamics_sm/ds1.F 新規作成) は次のようになっている.

```
SUBROUTINE DS1INI      !" 球関数 ( 1 層) を初期化
      CALL SMINIT
      RETURN
ENTRY DS1G2S             !" 格子→スペクトル
      CALL SMTG2S
      RETURN
ENTRY DS1S2G             !" スペクトル→格子
      CALL SMTS2G
      RETURN
ENTRY DS1S1V             !" スペクトル→格子 (ベクトル)
      CALL SMTS1V
      RETURN
```

以下に詳細を述べる. 説明中のパラメータは次のとおり.

```
REAL      GDATA( IDIM,JDIM )
REAL      GDATB( IDIM,JDIM )
REAL      WDATA( NMDIM )
CHRACTER  HFUNC*4
```

4.3.1 DS1INI

1. 機能

   SMPACK のサブルーチン SMINIT を呼んで球関数変換の準備をする.

2. 呼び出し方法

```
      CALL DS1INI
```

3. 備考

   (a) 現在のところ src/admin_sm/asetl.F の中で呼んでいる.

4.3.2 DS1G2S

1. 機能

   SMPACK のサブルーチン SMTG2S を呼び, グリッドデータからスペクトルデータへの変換を行なう.

2. 定義

   SMPACK のドキュメントを参照のこと

3. 呼び出し方法

```
      CALL DS1G2S
     O         ( WDATA    ,
     I           GDATA    )
```

4.3.3 DS1S2G

1. 機能

   SMPACK のサブルーチン SMTS2G を呼び, スペクトル→格子の変換を行なう.

2. 定義

   SMPACK のドキュメントを参照のこと

3. 呼び出し方法

```
 CALL DS1S2G
O         ( GDATA ,
I           WDATA  )
```

4.3.4 DS1S1V

1. 機能

   SMPACK のサブルーチン SMTS1V を呼び, スペクトル→格子 (ベクトル) の変換を行なう.

2. 定義

   入力の 1 つのスペクトルデータ WDATA の逆変換 A に対して, 実空間のベクトル (u, v) を次のように定義する.

$$(u, v) \equiv \left(\frac{1}{\cos\varphi} \frac{\partial A}{\partial \lambda}, \frac{\partial A}{\partial \varphi} \right) \tag{4.7}$$

3. 呼び出し方法

```
 CALL DS1S1V
O         ( GDATA , GDATB,
I           WDATA , WDATB,
I           HFUNC          )
```

4. 備考

   (a) HFUNC の 1 文字目が空白のとき, 出力は (GDATA,GDATB)= (u, v) となる. ただし, AGCM5 コードではこのような演算を行なっていない. HFUNC の 1 文字目が空白ではないとき, 出力は (GDATA,GDATB)= $(u\cos\varphi, v\cos\varphi)$ となる.

Appendix A

ベンチマークテスト

A.1 比較実験の概要

精度および演算速度を, 従来の球関数と SMPACK とで比較した.

計算に用いたのは, 傾圧不安定の初期値問題 (ごくらく AGCM5 体験, 第 4 章) である. 解像度は T42L12. タイムステップは 30 分. 高低気圧システムが非線型段階にある 15 日から 17 日の 2 日間 (約 100 ステップ) について, 同じ 15 日目の倍精度リスタートデータから積分を開始し, 17 日目での倍精度データの値および CPU 時間を比較した.

使用した計算機は, Fujitsu VPP500 (京大大型計算機センター) である. 計算は 1CPU で行ない (すなわち並列化はしていない), ベクトル化の指示レベルはセンターのデフォルトとした.

A.2 数値の比較

2 つの実験で得たデータの差を SMPACK によるデータで規格化して誤差と定義し, 最下層および第 8 層において大きさを見積った. 評価した変数は, 渦度, 発散, 温度, 気圧および比湿. その結果, 誤差はどの変数も同程度であり, 最下層では約 $1E-10$, 第 8 層では約 $5E-11$ であった. したがって, 両者の結果は十分な精度で一致しているといえる.

A.3 CPU タイムの比較

この 2 日間の積分に要した CPU 時間 (秒) は次のとおり:

98/03/07 赤堀 浩司

	SMPACK	オリジナル
力学過程	0.144917d+02	0.375584d+02
全時間	0.269713d+02	0.499214d+02

力学過程に要する積分時間は, オリジナルの球関数を用いた場合に比べ 38.6% にまで短縮されている.

力学過程のうち球関数ルーチンに関する CPU タイム (秒) は次のとおり:

多層

DSMG2S	0.384058d+01
DSMV2S	0.109984d+01
DSMV1S	0.210828d+01
DSMS2G	0.231807d+01
DSMS2V	0.227790d+01

一層

DS1G2S	0.353146d+00
DS1S1V	0.234311d+00
DS1S2G	0.129030d+00

Appendix B

謝辞

資源

agcm5

プログラム製作

沼口敦

プログラム製作協力

保坂征宏, 竹広真一, 石渡正樹, 堀之内武

文章

保坂征宏, 竹広真一, 石渡正樹, 堀之内武, 余田成男

版権と引用

本資源の版権は地球流体電脳倶楽部に属する. 地球流体電脳倶楽部の英語名は'GFD-DENNOU Club'である.

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. 詳細は心得を参照されたい. 地球流体電脳倶楽部の定める規定は地球流体電脳倶楽部サーバー (dennou) 上の

```
ftp://dennou-k.gaia.h.kyoto-u.ac.jp/saloon/dennou/To.Users
```

というファイルに収められている. 電脳サーバーは次の 4 箇所におかれている.

```
dennou-h.ees.hokudai.ac.jp
dennou-t.ms.u-tokyo.ac.jp
dennou-k.gaia.h.kyoto-u.ac.jp
dennou-q.geo.kyushu-u.ac.jp
```

93/07/09 林祥介
97/03/11 余田成男

agmc5 を用いて作成された著作物等には, 英語では例えば “The calculations were performed by agcm5, GFD-DENNOU Club.” 日本語では例えば 「計算は地球流体電脳倶楽部の agcm5 をもちいた.」のように明記していただければ幸いである. 「著者」に対応するものとしては地球流体電脳倶楽部 (GFD-DENNOU Club) をあげていただければ幸いである.

沿革

agcm5 は, 気象庁数値予報課の全球現業予報モデル (1985 年に住明正・増田耕一両氏が気象庁から東大理学部地球物理学科気象研究室に移植した) を参考に, 沼口敦氏が開発した大気大循環モデルである. agcm5 の 5 は氏の開発の第 5 版を意味する数字であり, 1988 年 の作業開始以来, 5 回にわたる大きな変更を経たものである. この間, 地球流体電脳ライブラリ (dcl) との整合性をも考慮し, 大学等での研究・教育用に利用できる形に整理がなされてきた. 残念ながら, agcm5 は現実大気のシミュレーションモデルとしてのチューンがなされていないモデルであり, 若干の水過程と若干の放射過程を含むものの, その実体は単なる球面プリミティブモデルといったほうが正しい.

国立環境研究所, 並びに, 東京大学気候システム研究センターでは, 沼口敦氏が製作の中心となり agcm5 をベースとして気候シミュレーションモデル, CCSR/NIES モデルが作成された.

地球流体電脳倶楽部版 agcm5 は沼口氏によって提供された資源に, マニュアルの整備, インストール道具の整備, バグフィックスを行なったものである.