AGCM5

ごくらく AGCM5 体験

地球流体電脳倶楽部

平成 14 年 8 月 2 日 (AGCM5-3.4)

目 次

1　はじめに　1

2　準備　2
2.1 親ディレクトリ . . . 2
2.2 資源の ftp . . . 2
2.3 資源の解凍 . . . 3
2.4 ドキュメントのコンパイル . . . 4
2.4.1 agcm5 のドキュメント . . . 4
2.4.2 gtool のドキュメント . . . 5

3　インスト ール　6
3.1 sysmake のインストール . . . 6
3.1.1 Makedef ファイルの編集 . . . 6
3.1.2 インストール . . . 6
3.2 gt3-dcl5 のインストール . . . 6
3.2.1 Mkinclude の編集 . . . 7
3.2.2 コンパイル . . . 7
3.2.3 インストール . . . 7
3.2.4 さらに . . . 8
3.3 agcm5 のインストール . . . 8
3.3.1 Mkinclude の編集 . . . 8
3.3.2 ライブラリの作成 . . . 9
3.3.3 実行ファイルの作成 . . . 10

4　実行　11
4.1 軸・初期値・境界値ファイルの作成 . . . 11
4.1.1 軸ファイル . . . 11
4.1.2 初期値ファイル . . . 13
4.1.3 境界値ファイル . . . 15
4.2 NAMELIST 入力ファイルの作成 . . . 16
4.3 実行 . . . 18
4.3.1 実行 . . . 18
4.3.2 gtool 解析 . . . 19
4.3.3 再実行 . . . 20

5 標準実験：湿潤大気大循環 21
5.1 インストール 21
5.1.1 Mkinclude の編集 21
5.1.2 実行ファイルの作成 22
5.2 実行 22
5.2.1 初期値ファイル 22
5.2.2 境界値ファイル 22
5.2.3 NAMELIST 入力ファイルの作成 23
5.2.4 実行 24
5.2.5 再実行 25
5.3 gtool 解析 26
5.3.1 アニメーション 26
5.3.2 データ解析 27

6 おわりに 30

A 謝辞 31

第1章　はじめに

このドキュメントは, ワークステーションでの「誰でもできちゃう AGCM5 体験」のメモ (堀之内, 96/12/12) をもとに, 「ごくらく AGCM5 体験」として電脳ドキュメント化したものである. AGCM5 と周辺ライブラリをインストールするところから始めて, 簡単な例題プログラムを走らせ, その結果をグラフ化できるように, 手順がまとめられている.

この解説は, 読者が UNIX マシーンをそれなりに使える (コマンドがわかる) 状況にあることを前提としている. 処理系言語は FORTRAN77 を用いているが, FORTRAN90 への対応はいずれ改めて記述する予定である. とりあえず, NAMELIST 文に関する基礎知識があればわかりやすい. NAMELIST 入力機能は FORTRAN77 の規格にはないが, たいていのコンパイラがオプションとして備えている. FORTRAN90 では, これが規格に含まれる標準的な機能となっている. 岩波 FORTRAN 辞典 (1986) によると,

NAMELIST ある種の処理系言語において NAMELIST 文を識別する機能語. FORTRAN77 にはない. NAMELIST 文は名前の組を宣言し, 組名を付ける. READ 文ではこの組名を参照する. 入力は次のような形で書く.

名前 = 定数, 名前 = 定数, ...

これによって, 入力記録中の名前と組名中の名前とが一致した要素だけに対して, 値が入力される.

機種依存部分 (方言) がある場合もあるので, NAMELIST の機能そのものについてはそれぞれの FORTRAN のマニュアル類を参照のこと.

AGCM5 のインストールや実行は, 計算機の機種や OS, 各サイトの環境などに依存する部分が多いので, どこでもうまくいくようにドキュメントを用意することは難しい. とりあえず, ここでは, OS が SunOS 4.1.4 の SPARCstation(stupid??@kugi.kyoto-u...) の例を書いておくので, それぞれの状況にあわせてインストールを行なうこと. また, ユーザー名は `horinout`(堀之内) として書いてあるので, これも各自のユーザー名にあわせること.

96/12/12 堀之内 武
97/02/06 余田 成男

第2章　準備

資源としては, agcm5, gtool(gt3-dcl5), sysmake の 3 つが必要である. また, gtool では dcl-5 のサブルーチンを使っているので, これがないときには dcl-5 も必要である. 以下では, dcl-5 はすでにインストールされているとして, その手続きは省略する.

2.1　親ディレクトリ

まず, 作業をするディレクトリをつくる. ディレクトリ名は各自の好みでよい.

```
% mkdir ~/dennou
```

ここでは, ホームディレクトリのしたに dennou ディレクトリを作った.

2.2　資源の ftp

agcm5, gtool(gt3-dcl5), sysmake の 3 つの資源を近くの電脳サーバーより anonymous ftp する.

```
% cd ~/dennou
% ftp dennou-k
Name: anonymous
Password: 自分の email アドレス
230 Guest login ok, access restrictions apply.
ftp> cd saloon/dennou_develop/model/agcm5
ftp> dir
total 899
-rw-rw-r--  1 dennou   dennou        736 Nov 12 09:29 TEBIKI.agcm5
drwxrwxr-x  7 dennou   dennou        512 Nov 12 10:08 agcm5.940120
-rw-r--r--  1 dennou   dennou     901780 Nov 12 09:33 agcm5.940120.tar.gz
drwxrwxr-x  3 dennou   dennou        512 Nov 12 10:03 config_local
ftp> binary
ftp> get agcm5.940120.tar.gz
ftp> cd /saloon/dennou_develop/util/gtool
ftp> dir
total 621
drwxrwxr-x  2 60124    dennou        512 Jun 22 05:29 OLD
-rw-rw-rw-  1 60124    dennou       1619 Jun 22 05:28 TEBIKI.gtool
```

96/12/12 堀之内 武
97/02/06 余田 成男

```
drwxrwxr-x  4 dennou   dennou        512 Oct 18  1995 config_local
lrwxrwxrwx  1 60121    dennou         22 Oct 27 07:30 gt3-dcl5.tar.gz -> gt3-dcl5_960906.tar.gz
-rw-rw-r--  1 60124    dennou     166977 Jun 22 05:37 gt3-dcl5_960110.tar.gz
-rw-rw-r--  1 60124    dennou     163876 Jun 22 07:17 gt3-dcl5_960622.tar.gz
-rw-rw-r--  1 60124    dennou     259415 Sep  6 08:48 gt3-dcl5_960906.tar.gz
ftp> get gt3-dcl5_960906.tar.gz
ftp> cd ../sysmake
ftp> dir
total 12
drwxrwxr-x  2 60124    dennou        512 Jun 22 05:29 OLD
-rw-rw-r--  1 60124    dennou        726 Jun 22 07:39 TEBIKI.sysmake
drwxrwxr-x  3 dennou   dennou        512 Jun 22 05:29 sysmake
lrwxrwxrwx  1 root     1              21 Jun 22 05:29 sysmake.tar.gz -> sysmake_960622.tar.gz
-rw-rw-r--  1 60124    dennou       3645 Jun 22 07:53 sysmake_960622.tar.gz
-rw-rw-r--  1 60124    dennou       3939 Sep  4 08:10 sysmake_960904.tar.gz
ftp> get sysmake_960904.tar.gz
ftp> bye
```

これで必要な資源が ftp できたはずである.

```
% ls -l
total 1165
-rw-r--r--  1 horinout   901780 Dec 12 14:08 agcm5.940120.tar.gz
-rw-r--r--  1 horinout   259415 Dec 12 14:19 gt3-dcl5_960906.tar.gz
-rw-r--r--  1 horinout     3939 Dec 12 14:20 sysmake_960904.tar.gz
```

2.3 資源の解凍

ftp した tar ファイルは gzip で圧縮されているので, gzcat と tar で解凍する.

```
% gzcat agcm5.940120.tar.gz | tar xvf -
% gzcat gt3-dcl5_960906.tar.gz | tar xvf -
% gzcat sysmake_960904.tar.gz | tar xvf -
% rm *.gz
% du -s *
9495    agcm5.940120
1480    gt3-dcl5
25      sysmake
```

ここで, gzcat (ファイル名) の代りに, gunzip < (ファイル名) としても良い.

ファイル名 agcm5.940120 は, 標準的な名前に戻しておけばよい.

```
% mv agcm5.940120 agcm5
```

2.4 ドキュメントのコンパイル

インストールを始める前に, それぞれのドキュメントをコンパイルしておく. ただし, sysmake にはドキュメントがない.

2.4.1 agcm5 のドキュメント

agcm5 の下の doc ディレクトリにドキュメントがある.

```
% cd agcm5/doc/
% ls -l
total 42
-rw-r--r--  1 horinout     2897 Mar  6 13:24 HISTORY.AGCM5.DOC
-rw-r--r--  1 horinout      516 Mar  6 13:24 Makefile
-rw-r--r--  1 horinout      654 Mar  6 13:24 Mkinclude
-rw-r--r--  1 horinout     4116 Mar  6 13:24 STATUS.AGCM5.DOC
-rw-r--r--  1 horinout    26204 Mar  6 13:24 TAGS
drwxr-xr-x 10 horinout      512 Mar  6 13:41 dai1bu/
drwxr-xr-x 10 horinout      512 Mar  6 13:41 dai2bu/
drwxr-xr-x  3 horinout      512 Mar  6 13:41 dai3bu/
drwxr-xr-x 12 horinout      512 Mar  6 13:41 dai4bu/
drwxr-xr-x  8 horinout      512 Mar  6 13:41 gokuraku/
drwxr-xr-x  8 horinout      512 Mar  6 13:41 guide/
```

HISTORY.AGCM5.DOC}にはドキュメンテーションに関する経緯が書いてあり, STATUS.AGCM5.DOC にはドキュメントの概略が書いてあるので参照のこと. dai1bu/ などのディレクトリの下には同名の TeX ファイルがあるので, jlatex をかける.

まず, Mkinclude を各サイトの状況にあわせて編集する. 具体的には, マクロ LPR, GUNZIP, MAKE, JLATEX, DVI2JPS, RM の定義を違うところだけ書き換える. もし, sysmake がすでにインストールされているのであれば, そのファイルをインクルードしてその中の定義を利用することも可能である.

make}でそれぞれのドキュメントの ps ファイルができる. ちゃんと ps ファイルが出来ているかを確認して出力する. プリンタ出力は make lpr とすればよい.

```
% make
Making gokuraku/
jlatex gokuraku.tex
(中略)
% ls -l */*.ps
-rw-r--r--  1 horinout   285414 Mar  6 13:30 dai1bu/dai1bu.ps
-rw-r--r--  1 horinout  1242088 Mar  6 13:31 dai2bu/dai2bu.ps
-rw-r--r--  1 horinout  2790150 Mar  6 13:32 dai3bu/dai3bu.ps
-rw-r--r--  1 horinout   949703 Mar  6 13:34 dai4bu/dai4bu.ps
-rw-r--r--  1 horinout   352904 Mar  6 13:29 gokuraku/gokuraku.ps
```

```
-rw-r--r--  1 horinout   413994 Mar  6 13:30 guide/guide.ps
% make lpr
Making gokuraku/
lpr -s gokuraku.ps
(中略)
% make clean
```

設定がわからず make に頼れない場合には, 自分で jlatex をかければよい. 例えば, 次のように foreach を使って一度にやる.

```
% ls */*.tex
dai1bu/dai1bu.tex       dai3bu/dai3bu.tex       gokuraku/gokuraku.tex
dai2bu/dai2bu.tex       dai4bu/dai4bu.tex       guide/guide.tex
% foreach dir ( [a-z]* )
foreach? cd $dir
foreach? jlatex $dir.tex
foreach? jlatex $dir.tex
foreach? jlatex $dir.tex
foreach? dvi2ps $dir.dvi >! $dir.ps
foreach? cd ..
foreach? end
```

2.4.2　gtool のドキュメント

こちらも gt3-dcl5 の下の doc ディレクトリにドキュメントがある. やはり Makefile がついているので, make コマンドでドキュメントが作れるはずである.

```
% cd ../../gt3-dcl5/doc
% make gtool.ps
```

ただし, make を使う場合には ../Mkinclude を適当に書き換えていなければ失敗する. また, コマンド makeindex がインストールされてない場合にも失敗する.

make に頼れない場合には, 自分で jlatex をかければよい.

```
% jlatex gtool.tex
% jlatex gtool.tex
% jlatex gtool.tex
% dvi2ps gtool.dvi >! gtool.ps
% lpr gtool.ps
```

なお, お掃除は,

```
% make clean
```

とする. このディレクトリには, TeX ファイル以外に「サンプルプログラム」(実行型コマンド) の使い方の例が howtoloo.doc(テキストファイル) にもあるので, これも出力しておくとよい.

第3章　インストール

インストールする順番は,

1. sysmake,　2. gtool,　3. agcm5

が良い. それぞれ, テキストファイル sysmake/INSTALL, gt3-dcl5/INSTALL, および, 「AGCM5 利用者の手引」(agcm5/doc/guide/guide.tex) を参照のこと.

3.1　sysmake のインストール

基本的には sysmake/INSTALL を参考にして, Makedef.XXX ファイルを書き換えるだけである. ここで, XXX は Sun, HITAC, SX3 のような機種名などである.

3.1.1　Makedef ファイルの編集

sysmake/Makedef.Sun を編集する. 特に注意すべきは X ライブラリのインストール場所である. サイトの状況に合わせて,

```
SYSXLIBDIR = /usr/X11R5/lib
```

とする. また,

```
INDEX           = ctags -wo
```

とする. これは, うち (stupid@kugi) では INDEX = etags -wx ではうまく行かないからである. SunOS 4.1.3 になる前はこれでうまく行ってたような気がするのだけど (堀之内 談).

3.1.2　インストール

インストールはファイルをコピーするだけである. /usr/local/lib の下の然るべき位置にインストールしたい場合は cp する. 個人で使う場合は何もしなくて良い.

3.2　gt3-dcl5 のインストール

gt3-dcl5/INSTALL を参考にして, Mkinclude の編集, コンパイル, インストール, の順に作業をする. そのあとで, タグファイルの作成とお掃除をやったほうがよい.

96/12/12 堀之内 武
97/02/07 余田 成男

3.2.1　Mkinclude の編集

前半のサイト依存部分を状況にあわせて編集する. システムの選択, コンパイルオプションの指定, gtool3 ライブラリおよびコマンドのインストール先の指定, および, dcl-5.0, sysmake のインストールされているディレクトリの指定などを行なう.

うち (stupid@kugi) での変更部分を列挙する.

```
SYSTEM          = Sun
GTOOLDIR = $(HOME)/dennou/gt3-dcl5
DESTDIR = $(GTOOLDIR)
BINDEST = $(DESTDIR)/bin
DCLLIBDIR = /usr/local/lib/dcl-5.0
SYSMAKEDIR = $(HOME)/dennou/sysmake
```

SYSTEM は Sun がデフォルトである. GTOOLDIR と SYSMAKEDIR は, 各自のディレクトリの事情に合わせること. 今の場合, 親ディレクトリ $(HOME)/dennou の下に gt3-dcl5, sysmake を置いている. BINDEST はデフォルトのままだが, 一般に, この bin ディレクトリは幾つもの FORTRAN 実行コマンドにより 10~20MB ぐらいまで膨れるので, ホームディレクトリではなく, データ用 (作業用) ディスクにこのディレクトリを切っておくことを強く奨める. DCLLIBDIR は, 環境変数 DCLDIR が設定されていれば, オリジナルのままでよい.

ついでに, $(BINDEST) にパスを通しておくとよい.

3.2.2　コンパイル

make で gtool ライブラリとそれを用いて FORTRAN コンパイルのためのシェルスクリプト gt3frt が作成される.

```
% make clean.all
% make
```

ここで, まず make clean.all とお掃除したのは, ftp した tar ファイルにはコンパイル済のものが含まれていることがあり, 最初にそれらを消しておいたほうがよいからである. コンパイルが正しくできたかどうか, 次の二点を確認しておくこと：

1. $(LIBRARY)(ふつうは gt3-dcl5/lib/libgtool3.a) が出来ている.
2. シェルスクリプト $(BINDIR)/gt3frt の中身が, サイトに適合した正しいものになっている.

3.2.3　インストール

make install でライブラリとコマンド群がインストールされる.

```
% make install
```

このとき, $(LIBDIR) と $(LIBDEST) が異なる場合には, ライブラリが後者にコピーされる. また, コマンド群は src/sample/*/ にあるが, これがコンパイルされて $(BINDEST) にコピーされる. この結果, $(BINDEST) は 12.2MB ほどの大きさになる.

3.2.4 さらに

インストールは終わったが, ここで, タグテーブルファイル作成とお掃除をしておくとよい.

emacs(mule) 等で複数のファイルにまたがるソースの検索を支援するためのタグテーブルファイル TAGS を用意しておくと便利である.

タグテーブル タグテーブルとは, いくつかのファイルにまたがったプログラムがどのようにファイルに分れているかを記述したものです. そこには, おのおののファイルとそのファイルにある関数名とその位置のリストがあります. 関連したファイルをまとめると, 1 つのコマンドですべてのファイルを検索したり置き換えたりすることが可能になります. ··· タグテーブルはタグテーブルファイルに記憶されます. タグテーブルファイルは TAGS といわれています. (共立出版「GNU Emacs マニュアル」, 1988; p.123)

通常は, すでに TAGS が出来ているので, make しなくてよいが,

```
% make tags
```

とすれば, TAGS を作ってくれる.

コンパイル作業により, ファイル使用量がかなり膨れ上がってるはずなので, ここでお掃除をやったほうがよい.

```
% make clean
```

これにより余分な中間ファイル (オブジェクトファイル や src/saple/*/ に残された実行型ファイルなど) が消される.

3.3 agcm5 のインスト ール

ここでは個人ですべての資源を持つことを想定して作業をすすめる. Mkinclude の編集, ライブラリの作成, 実行ファイルの作成, の順に行なう. AGCM5 利用者の手引 (agcm5/doc/guide/guide.tex) の第 1 章を参照のこと.

3.3.1 Mkinclude の編集

agcm5/src にある Mkinclude ファイルの内容は, 実験設定部分とサイト依存部分とに分れている. まず, サイト依存部分について述べ, つぎに, 傾圧不安定初期値問題を実験例として, 実験設定部分について述べる.

うち (stupid@kugi) での「サイト依存部分」の変更部分を中心に列挙する. システム (sysmake) の指定やパコンパイルオプションの指定はデフォルトのままで, パスの設定を状況にあわせて変更する.

```
DENNOUDIR       = $(HOME)/dennou
AGCMDIR         = $(DENNOUDIR)/agcm5
DESTDIR         = $(AGCMDIR)
LIBDEST         = $(DESTDIR)/lib
```

```
BINDEST         = $(DESTDIR)/bin
DCLLIBDIR = /usr/local/lib/dcl-5.0
SYSMAKEDIR = $(DENNOUDIR)/sysmake
GTOOLLIBDIR = $(DENNOUDIR)/gt3-dcl5/lib
GTOOLINCDIR = $(DENNOUDIR)/gt3-dcl5/include
```

つぎに, T10L5 傾圧不安定初期値問題用の実験設定を行なう. 「物理過程」はほとんど使わないことにする. 選択または変更したところを列挙する.

```
############## 分解能の指定 ########################
RESOLUTION  = t10l5

############## 実行ファイル名指定 agcm5.$(RESOLUTUON)$(MODELNAME)
MODELNAME       = dyn

############## nonstd/（非標準ルーチン）の指定 ###########
CUMLUS = nonstd/p2cum0.o        #       積雲パラメタリゼーションなし
DRYADJ =  nonstd/p2dad0.o       #       乾燥対流調節なし
SHLCNV =  nonstd/p2shl0.o       #       浅い積雲なし
RADIAT  =  nonstd/p2rad0.o       #       放射なし
VDFUS   =  nonstd/p2vdf0.o \
           nonstd/p2sfc0.o      #       拡散係数一定
GRVD    =  nonstd/p2gra0.o      #       重力波抵抗なし
```

MODELNAME は適当な名前でよいが, この実験は「力学」部分ばかりなので MODELNAME を dyn とした. これで, 実行ファイル名は agcm5.t10l5dyn となる. 非標準ルーチンの指定のうち, 鉛直拡散は拡散係数一定を選び, それ以外の物理過程については「なし」を選択した. また, 利用者自身があるルーチンを独自に作り変えた場合には,「利用者設定ルーチン」を適切なディレクトリに置き, この Mkinclude の中で指定することになるが, もちろんここでは見合わせる.

3.3.2　ライブラリの作成

ライブラリの作成は src ディレクトリで行なう. 余計なものが残ってた場合に備えて, 最初にお掃除をしておく.

```
% cd agcm5/src/
% make clean
```

さらに, 初回だけは以下の措置をしておこう：

```
% touch include/zcdim.F
```

この理由は次である. 分解能を指定するファイル include/zcdim.F が正しいものに更新されているかどうかは, ファイル include/zc$(RESOLUTUON).sel より新しいかどうかで判断される. ftp してきたばかりのときには日付が縮退していることがあるので, touch することで強制的に分解能指定ファイルの更新時刻等を現在の時刻にするほうが安全である.

ここで,

```
% make lib
```

とすると, ライブラリのコンパイルが行なわれ, $(LIBDEST)/libagcm5$(RESOLUTUON).a ができる. いまの例では, libagcm5t10l5.a というファイルが, $(HOME)/dennou/agcm5/lib というディレクトリの下にできる. モデルの分解能が変らない限り, ライブラリを再コンパイルする必要はない. 非標準ルーチンやユーザー定義ルーチンを取り替えて再実行する場合には, これらに対応するオブジェクトファイルが, 実行ファイルを作成する時に, (必要ならばコンパイルされ) 直接リンクされるからである.

3.3.3　実行ファイルの作成

メインプログラムなどをコンパイルし, 前節で作ったライブラリをリンクすることにより, 実行ファイルを作成する. src ディレクトリで,

```
% make gcm
```

とすれば, $(BINDEST)/agcm5.$(RESOLUTUON)$(MODELNAME) ができる. いまの例では, agcm5.t10l5dyn というファイルが, $(HOME)/dennou/agcm5/bin というディレクトリの下にできる.

該当する分解能のライブラリがない場合には, エラーとなるので前節のライブラリの作成に戻る. 鉛直 1 次元モデルなら make 1dim, 2 次元モデルなら make gcmyz または make gcmxz とする.

いつでも最後はお掃除である.

```
% make clean
```

第4章　実行

前章で実行ファイルができたわけであるが, 実際に agcm5 を走らせるためには初期値ファイルと境界値ファイルが必要である (後者は省略可). また, 出力結果のグラフを gtool で書くためには軸ファイルがいる. これも忘れないうちに作ってしまおう. さらに, agcm5 の `Mkinclude` で指定した以外の実験設定は NAMELIST 入力という機能を使って可読ファイルから読み込むので, `NAMELIST` 入力ファイルも作らなければならない.

ここまでお膳立てができたところで, コマンド実行 (時間積分) は,

`%（実行コマンド）<（NAMELIST 入力ファイル）`

という形式で行う. 初期値・境界値ファイル名を `NAMELIST` 入力ファイルに記述しておけば, 実行時にそれが参照され, 該当するファイルがオープンされて, データが読み込まれるという手順となる.

4.1　軸・初期値・境界値ファイルの作成

4.1.1　軸ファイル

軸ファイルがなくても agcm5 は走るが, 図は描けないので軸ファイルを作っておく. この部分は実験設定に依存しないので, 最初に片ずけることにする.

```
% cd agcm5/src/prep
% make axis
```

とすると,

```
*** WARNING (REWNML) ***  UNIT 5 MAY BE A STANDARD INPUT
```

というメッセージを吐いて実行が中断される. これは各層の σ 座標値の入力を求めているためである. `^D` を入力することでデフォルトの値が採用され, 軸ファイルが作られる. 実際, 次のようなファイル群である.

```
% ls -l GTAX*
-rw-r--r--  1 horinout     1064 Dec 12 21:42 GTAXLOC.CSIG5.M
-rw-r--r--  1 horinout     1060 Dec 12 21:42 GTAXLOC.CSIG5.P
-rw-r--r--  1 horinout     1104 Dec 12 21:42 GTAXLOC.GGLA16
-rw-r--r--  1 horinout     1172 Dec 12 21:42 GTAXLOC.GLON32
-rw-r--r--  1 horinout     1064 Dec 12 21:42 GTAXWGT.CSIG5.M
-rw-r--r--  1 horinout     1060 Dec 12 21:42 GTAXWGT.CSIG5.P
```

96/12/12 堀之内 武
97/02/14 余田 成男

```
-rw-r--r--  1 horinout     1104 Dec 12 21:42 GTAXWGT.GGLA16
-rw-r--r--  1 horinout     1172 Dec 12 21:42 GTAXWGT.GLON32
```

ここで, GTAXLOC.* は軸の位置のデータであり, GTAXWGT.* は平均を取るときの重みのデータである. 分解能の指定 (第 3.3.1 節) で RESOLUTION=t10l5 としたので, 鉛直に σ 座標で 5 層, 水平の緯度経度格子が 16×32 の分解能用の軸データができていることが見てとれる. これらの中身はバイナリーであるので, それを表示するためには gtool のコマンド gtshow を使う. (そのためには, gt3-dcl5/Mkinclude で指定した $(BINDEST) にパスを通しておくかなければならない; 第 3.2.1 節参照).

```
% gtshow GTAXLOC.GGLA16
###### Data No.    1 ###############
 IDFM            :             9010
 DSET            : AXLOC
 ITEM            : GGLA16
 FNUM            :                1
 DNUM            :                0
 TITLE           : latitude
 UNIT            : deg
 TIME            :                0
 UTIM            : SEC
 DATE            : 19900001000100
 TDUR            :                1
 AITM1           : GGLA16
 ASTR1           :                1
 AEND1           :               16
 ASTR2           :                1
 AEND2           :                1
 ASTR3           :                1
 AEND3           :                1
 DFMT            : UR4
 MISS            :   -0.9990000E+03
 DMIN            :   -0.9000000E+02
 DMAX            :    0.9000000E+02
 DIVS            :    0.1000000E+02
 DIVL            :    0.3000000E+02
 STYP            :                1
 CDATE           : 19961212 214236
 CSIGN           : horinout
 MDATE           : 19961212 214237
 MSIGN           : horinout
  1,  1,  1:  8.16506E+01   2,  1,  1:  7.08346E+01   3,  1,  1:  5.99549E+01
  4,  1,  1:  4.90607E+01   5,  1,  1:  3.81612E+01   6,  1,  1:  2.72592E+01
  7,  1,  1:  1.63559E+01   8,  1,  1:  5.45204E+00   9,  1,  1: -5.45204E+00
```

```
 10,  1,  1: -1.63559E+01  11,  1,  1: -2.72592E+01  12,  1,  1: -3.81612E+01
 13,  1,  1: -4.90607E+01  14,  1,  1: -5.99549E+01  15,  1,  1: -7.08346E+01
 16,  1,  1: -8.16506E+01
```

次に, これらの軸ファイルを所定のディレクトリに移す. gtool で軸ファイルの読み込み時にそれを探すのは, カレントディレクトリ, gt3-dcl5/Mkinclude の GTAXLOC, GTAXWGT で指定したディレクトリの順である. 指定を自分で変えていなければ, 後者は $(LIBDEST)/gt3 である. いまの場合, これは $(HOME)/dennou/gt3-dcl5/lib/gt3 というディレクトリである. はじめての場合には,

```
% mkdir ~/dennou/gt3-dcl5/lib/gt3
```

として, このディレクトリを作り,

```
% mv GTAX* ~/dennou/gt3-dcl5/lib/gt3
```

で, 軸ファイルを移す.

4.1.2　初期値ファイル

オリジナルのソースで用意されているのは, 等温静止大気に微小擾乱を加えた初期値データである. これを作るのは簡単で,

```
% cd agcm5/src/prep
% make init
```

とすると, FORTRAN プログラムがコンパイル・実行される. データの名前とファイル名を聞いてくるので適当に入力する:

```
 DATA NAME  ?
test
 FILE NAME  ?
/datadisk/horinout/TEST.INIT
```

まず, DATA NAME には初期値データの名称 test を入力した. 16 文字までの適当な名前をつける. 次に, FILE NAME で初期値データのファイル名を仮に /datadisk/horinout/TEST.INIT とした. モデル分解能が細かいときには, 当然大きなファイルができるのでパスに注意しなければならない. ホームディレクトリではなく, 適当なデータ用ディスク /datadisk に置く例である. その後, ネームリスト入力などを聞いてくるが, ^D を入力してデフォルトに委ねて構わない. 出来たファイルの中で, 初期値は U, V, T, PS, Q(東西風, 南北風, 温度, 地上気圧, 水蒸気量) の順でならんでいる.

長時間走らせて準平衡状態になってから解析するような問題では, このように等温静止大気を初期値とすれば良い. 一方, 初期値問題として何かを解こうとすれば, その問題にふさわしい初期場を作るために, prep/xminit.F を然るべく編集しなければならない.

ここでは, t10l5 のモデルで傾圧不安定の初期値問題を解くことにする. まず, ユーザー各自の資源を収めるディレクトリを src/usr/ の下につくる.

```
% cd agcm5/src/usr
% mkdir horinout
% cd horinout
% mkdir prep
% cd prep
```

次に, prep ディレクトリにある Makefile をコピーして編集する.

```
% cp ../../../prep/Makefile .
```

Makefile 中のパスを ../include から../../../include 等に変える.

あとは各自が勝手に編集すれば良いのだが, ここでは堀之内のプログラムをコピーして利用することにする.

```
% cp ~/dennou/agcm5/src/usr/horinout/prep/xmkinit00.F .
% cp ~/dennou/agcm5/src/usr/horinout/prep/barocli.F .
% cp ~/dennou/agcm5/src/usr/horinout/prep/ltinit.F .
```

これらに対応して, Makefile 中の OBJS_INIT を,

```
OBJS_INIT      = xmkinit00.o barocli.o ltinit.o
```

とし, ファイルの最後に,

```
xmkinit00.o: ../../../include/zcdim.F ../../../include/zhdim.F ../../../include/zccom.F
barocli.o: ../../../include/zcdim.F ../../../include/zhdim.F ../../../include/zccom.F
ltinit.o: ../../../include/zcdim.F ../../../include/zhdim.F
```

を加える.

あとは, 等温静止大気の場合と同じである.

```
% make init
```

として, 聞かれるままにデータの名前やファイル名などを入力すればよい:

```
 DATA NAME  ?
barocli
 FILE NAME  ?
/datadisk/horinout/TEST.INIT
```

さらなる質問は, やはり ^D でかわせばよい. これにより, 鉛直シアーがあり各層で剛体回転する基本帯状流に微小擾乱を加えた初期値データが得られる. FORTRAN プログラム barocli.F 中のパラメータ TB1, TB2, U01, U02 の値を変えれば, 傾圧性の異なる基本場が得られる.

やはり, 出来たファイルの中で, 初期値は U, V, T, PS, Q の順でならんでいる. これは, gtool のコマンド gtshow で確かめられる :

```
% cd /datadisk/horinout
% gtshow -head TEST.INIT
```

ここで, オプション -head を指定することにより, ヘッダーのみが表示される. さらに, gtcont で各種コンター図を描くこともできる：

```
% gtcont TEST.INIT
% gtcont y=5 TEST.INIT
% gtcont x=0 TEST.INIT
```

最初の例は, gtcont z=0 と等価で, 鉛直平均した初期場の水平面図が U, V, ··· の順に描かれる. 二番目のように gtcont y=5 とすれば, 北から 5 番目のグリッドにおける鉛直断面図が描かれる. また, gtcont x=0 とすれば, 東西平均した子午面断面図が描かれる (図 4.1). gtool コマンドについては, 姉妹編「ごくらく GTOOL3 体験」を参照のこと.

4.1.3　境界値ファイル

境界値ファイルがなくてもモデルは走る. この場合, 大気下端境界は全球が海面で覆われている仮定され, 海面温度 (SST) は初期値の最下層の温度がそのまま用いられる. /s08d1c

SST ファイルは make sst で作ることができる. 初期場と同様に NAMELIST 入力は ^D でデフォルトにすれば, 仮想的な緯度分布の東西に一様な SST 分布を与えるファイルができる.

```
% cd agcm5/src/prep
```

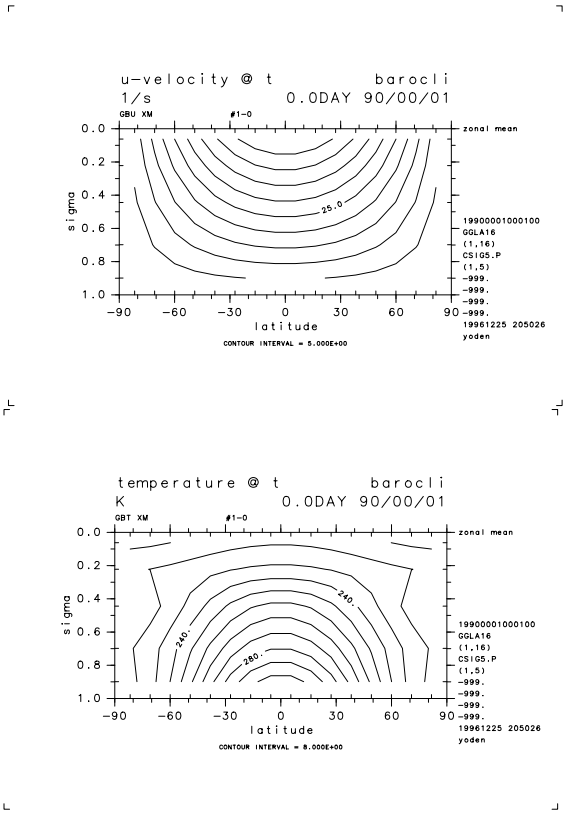

図 4.1: 初期値ファイルの gtcont 出力例より. 東西風 (左) と温度 (右).

```
% make sst
 DATASET NAME?
SST
 OUTPUT FILE NAME for SST OUTPUT ?
sst.data
^D
```

やはり, 出来た境界値ファイルは, gtool のコマンドで確かめられる：

```
% gtcont sst.data
% gtcurv x=1 sst.data
```

gtcont で水平分布のコンター図が描かれ, gtcurv で南北分布の折れ線図が描かれる.

4.2　NAMELIST 入力ファイルの作成

いよいよ最終段階である. ここで実験設定用の NAMELIST 入力ファイルさえ作れば, あとは走らせるだけである. NAMELIST 入力は, おもにデフォルトの設定値を変更するために使われる. 変更の必要のない項目は記述しなくてもよい. NAMELIST 入力機能そのものについては FORTRAN のマニュアル類を参照のこと.

NAMELIST 入力ファイルの例は agcm5/sample に収められているので, 適当なものをコピーして加工すると良い (例えば, agcm5/sample/sh.Sun/INDATA). ここではディレクトリ /datadisk/horinout/ に INDATA という名前の NAMELIST 入力ファイルを作ることにする.

```
% cd agcm5/sample/sh.Sun
% cp INDATA /datadisk/horinout/
```

以下は, 上記の傾圧不安定の初期値問題に対する NAMELIST 入力ファイル INDATA の例である：

```
% cd /datadisk/horinout/
% ls -l
total 141
-rw-r--r--  1 horinout     1858 Dec 13 02:11 INDATA
-rw-r--r--  1 horinout   141536 Dec 12 23:35 TEST.INIT
% cat INDATA
 &NMRUN  RUN='TEST01',     SIGN='horinouchi'          &END
 &NMINTG TUNIT='MIN',      DELT=90,
         INISTP=3,         TFIL=0.05             &END
 &NMTIME TUNIT='DAY',
         START=0,          END=40,
         INTRST=0                                &END
 &NMINIT FILE='TEST.INIT'                  &END
 &NMRSTR FILE='TEST.RST', DFMT='UR8'       &END
 &NMINIG FILE=' '                                &END
```

```
&NMRSTG FILE=' ',               DFMT='UR8'      &END
&NMDATE IDATEI=1996,12,12,0,0,0                  &END
&NMCALN OPERPT=F                                 &END
&NMDAYL TUNIT='HOUR',     DAYLEN=24              &END
&NMHDIF TUNIT='HOUR',     TEFOLD=48,
        ORDER=6                                  &END
&NMVDF0 DFMCON=0.01, DFHCON=0.01, DFECON=0.01      &END
&NMBFIL HGRSST=' ', HGRHC=' ',
        HGRALB=' ',      HGRBET=' ',
        HGRZ0=' ',       HGRZSD=' ',
        HIDSFC=' '                              &END
&NMPHIS HPHIS=' '                               &END

&NMDIAG FILE='CON',       UNIT='HOUR',
        INTDIA=6,         IDIAG=16,
        JDIAG=100,         KDIAG=-1,
        NCOLS=80                                 &END
&NMHISD FILE=' ','CON',   TUNIT='HOUR','DAY',
        INTOUT=24,2,       INTSUM=0,0,
        AVRTYP=' ',       DFORM='UR4',
        INTEND=0                                 &END
&NMMONI IINT=10,          JINT=100,
        KINT=100,           NCOLS=133             &END
&NMHIST ITEM='U',
        FILE='intg1/u.data',             &END
&NMHIST ITEM='V',
        FILE='intg1/v.data',             &END
&NMHIST ITEM='T',
        FILE='intg1/t.data',             &END
&NMHIST ITEM='PS',
        FILE='intg1/ps.data',            &END
&NMHIST ITEM='SIGDOT',
        FILE='intg1/sigdot.data',        &END
&NMHIST ITEM='VOR',
        FILE='intg1/vor.data',           &END
&NMHIST ITEM='DIV',
        FILE='intg1/div.data',           &END
```

ここで, 各行の & の前の空白は, Sun の場合には必要である (Convex では要らない). &NMHIST で指定したように, データは /datadisk/horinout の下の intg1 のディレクトリに納めることにする:

```
% mkdir intg1
```

intg1 としたのは, 時間積分を延長したくなったら別のディレクトリ intg2 を用意することを想定しているからである (第 4.3.3 節参照).

4.3 実行

4.3.1 実行

いよいよ実行である.

```
% cd /datadisk/horinout/
% ~/dennou/agcm5/bin/agcm5.t10l5test < INDATA
```

これで, INDATA が読み込まれて, agcm5.t10l5test が実行される. 以下は, 標準出力への出力結果の一部分である.

```
 &nmrun  run='TEST01          ', sign='horinouchi      '
 &end
 &nmintg  tunit='MIN ', delt=  120, inistp=  3, tfil=    5.0000000000000D-02
 &end
 &nmtime  tunit='DAY ', start=  0, end=  40, intrst=  0
 &end
 &nmfile  vmiss=   -999.00000000000
```

(中略)

```
############### CPU TIME SUMMARY #################
 SETUP              0.880000D+00   0.800000D+00
 TIMCON             0.293013D+01   0.291012D+01
 PHYSCS             0.114790D+03   0.114500D+03
 DYNMCS             0.259960D+03   0.259840D+03
 PADJST             0.355302D+02   0.353902D+02
 TFILT              0.751009D+01   0.749007D+01
 MISC               0.249985D+00   0.189987D+00
 HISTIN             0.929989D+01   0.927989D+01
 HISTOU             0.574001D+01   0.461004D+01
    TOTAL TIME =    0.437140D+03   0.435170D+03
```

これで, 無事時間積分が終わったわけである. この CPU TIME SUMMARY は, 左列のルーチンにかかった時間が秒単位で示されており, 中央列は usr + sys, 右列は usr での時間である. この実行例では, CPU time は合計で 437 秒かかっている.

intg1 の下にちゃんとデータが出来ているか確認しておく.

```
% ls -l intg1
total 2952
-rw-r--r--  1 horinout      451200 Dec 12 23:55 div.data
-rw-r--r--  1 horinout      123520 Dec 12 23:55 ps.data
-rw-r--r--  1 horinout      533120 Dec 12 23:55 sigdot.data
-rw-r--r--  1 horinout      451200 Dec 12 23:55 t.data
```

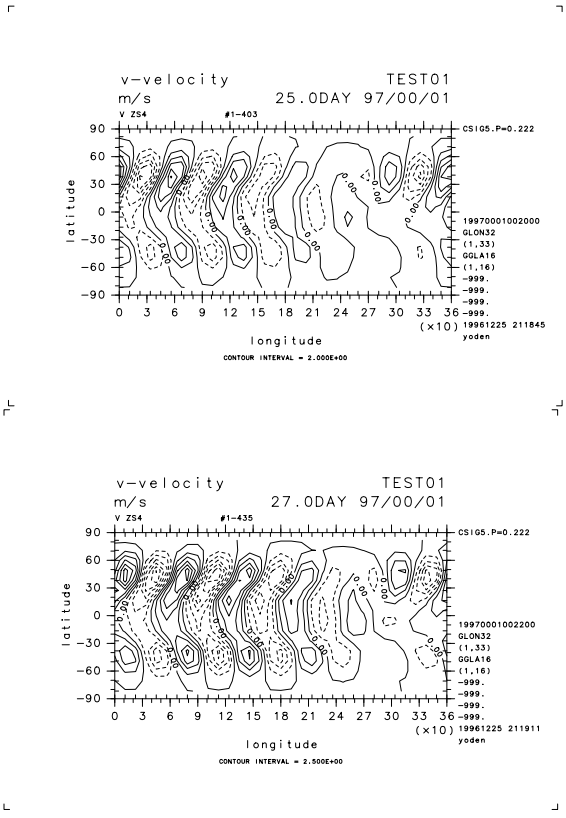

図 4.2: 出力結果のクイックルック (gtcont 出力) 例より. 南北風 V の水平面図.

```
-rw-r--r--  1 horinout      451200 Dec 12 23:55 u.data
-rw-r--r--  1 horinout      451200 Dec 12 23:55 v.data
-rw-r--r--  1 horinout      451200 Dec 12 23:55 vor.data
```

4.3.2 gtool 解析

出力結果を gtool のアニメーションで見てみる.

```
% cd intg1
% gtcont -sw:lalt=T -sw:lwait=F -greset z=4 v.data
```

ここで, `-sw:lalt=T -sw:lwait=F` は, dcl の swpack が管理する内部変数を置き換えるオプションで, `lalt` を `T`(rue), `lwait` を `F`(alse) と指定することにより, アニメーション表示が可能となる (1 ページの画面が比較的短時間で作画できる場合). 詳細は, 地球流体電脳ライブラリ GRPH1 第 7 章を参照のこと. また, `-greset` は, 風速のコンター間隔を結果の値に応じて変えるようにするためのオプションである. デフォルトでは, 風速のコンター間隔が 2.5m/s に固定されている. `z=4` で下から 4 レベル目を選択し, 南北風 V の水平面コンター図を描いている (図 4.2).

```
% gtcont -sw:lalt=T -sw:lwait=F str=20 step=2 x=0 u.data
```

この例では, `str` と `step` で, 時系列データの読み込の始めと間隔をそれぞれ指定している. また, `x=0` で帯状平均値の子午面分布のコンター図を描くようにしでいる. データは, 東西風 U である.

x=n などで, n が 0 のときは x 方向の平均値を, $n(>0)$ のときは n 番目のグリッドでの値を描くことになる.

また, ポストスクリプト (ps) ファイルを作るには,

```
% gtcont  str=28 end=28 x=0 wsn=2 u.data
```

などとして, wsn で dcl のワークステーション番号 2 を指定する.

4.3.3　再実行

intg1 の結果に続けて積分を行ないたい場合には, 別のディレクトリ intg2 を用意する：

```
% cd /datadisk/horinout/
% mkdir intg2
```

また, 再実行用の NAMELIST 入力ファイル INDATA2 を用意する. これと INDATA との diff をとった結果が次であるので, 該当部分を書き換えればよい.

```
% diff INDATA INDATA2
5c5
<          START=0,          END=40,
---
>          START=40,          END=80,
7,8c7,8
<  &NMINIT FILE='TEST.INIT'                 &END
<  &NMRSTR FILE='TEST.RST', DFMT='UR8'      &END
---
>  &NMINIT FILE='TEST.RST'                 &END
>  &NMRSTR FILE='TEST.RST2', DFMT='UR8'      &END
34c34
<          FILE='intg1/u.data',             &END
---
>          FILE='intg2/u.data',             &END
(中略)
46c46
<          FILE='intg1/div.data',           &END
---
>          FILE='intg2/div.data',           &END
```

あとは,

```
% ~/dennou/agcm5/bin/agcm5.t10l5test < INDATA2
```

で, 時間積分の延長 (再実行) が行なわれる.

第5章　標準実験：湿潤大気大循環

これまでは, ほとんどの物理過程を取り除いた「力学」モデルを構成し, 傾圧不安定初期値問題を解くという実例を述べた. しかし, これでは GCM とはいえないので, この章では放射や湿潤対流過程などを含む標準的な設定で, 準平衡状態を求める実験を行う. ワークステーションでそれほど時間をかけずに計算を行うために, 分解能は前章と同じにしたが, もっと高い分解能で計算すればより現実的な実験となるであろう.

モデルの設定としては, 基本的に各物理過程の標準ルーチンを使用する. ただし, 重力波抵抗だけは入れないことにする. 自分でそれなりの設定をしない限り重力波抵抗はゼロになってしまい, 余計に計算時間がかかるだけだからである.

これまでの二つの章でのべたインストールと実行を繰り返すことになる. 同じライブラリを使ったりするので, いくつかの手続きをスキップする. 今回の場合, 実行ファイルを作り直して, 初期値・境界値を与え, 実験設定をして実行することになる.

5.1　インスト ール

分解能はこれまでと同じとするので, ライブラリを新たに作り直す必要はない. `Mkinclude` ファイルを編集し直して, 実行ファイルを再コンパイルするだけでよい.

5.1.1　Mkinclude の編集

第 3.3.1 節と同様に `/agcm5/src` ディレクトリの `Mkinclude` ファイルを編集する. 「サイト依存部分」は同じなので, ここで書き換えるのは, 「実験設定部分」のおもに非標準ルーチンの指定部分だけである. 以下に, 選択または変更したところを列挙する.

```
############## 実行ファイル名指定 agcm5.$(RESOLUTUON)$(MODELNAME) #####
MODELNAME       = moist         # 適当な名前をつけましょう.

############### nonstd/（非標準ルーチン）の指定 ############
###### 積雲パラメタリゼーション #####
CUMLUS  =                       # 標準    Kuo 方式

###### 乾燥対流調節 ######
DRYADJ  =                       # 標準   あり

###### 浅い積雲対流 ######
```

97/02/20 堀之内 武
97/03/07 余田 成男

```
SHLCNV  =                       # 標準  Tiedtke

###### 重力波抵抗 ######
GRVD    =  nonstd/p2gra0.o      #       重力波抵抗なし

###### 放射 ######
RADIAT  =                       # 標準

###### 鉛直拡散 ######
VDFUS   =                       # 標準
```

5.1.2　実行ファイルの作成

Mkinclude ファイルの書き換えが終わったら, やはり /agcm5/src ディレクトリで実行ファイルを再コンパイルする.

```
% make gcm
% make clean
```

これで, $(HOME)/dennou/agcm5/bin というディレクトリの下に, agcm5.t10l5moist というファイルができているはずである.

5.2　実行

分解能が同じなので, 軸ファイルもつくり直す必要がない. 湿潤大気大循環実験用に初期値ファイル・境界値ファイルを作り直して, NAMELIST 入力ファイルを再編集し, 実行すればよい.

5.2.1　初期値ファイル

長期間積分した後の準平衡状態を求めるのだから, 初期値は特に凝る必要がないであろう. そこで, 初期値として src/prep ディレクトリに用意されている静止等温大気に微小温度擾乱を加えたものを用いることにする. 初期値ファイルの作り方は, 第 4.1.2 節のはじめに述べた通りである：

```
% cd prep
% make init
 DATA NAME  ?
rest
 FILE NAME  ?
/datadisk/horinout/Rest.init
^D
```

5.2.2　境界値ファイル

境界条件としては, 地表面は全球海面とし, そこでの温度分布は気候学的 SST の分布に近い値を用いる. これも src/prep ディレクトリに用意されているものを使う：

```
% make sst
 DATASET NAME?
SST
 OUTPUT FILE NAME for SST OUTPUT ?
/datadisk/horinout/SST.data
^D
```

5.2.3 NAMELIST 入力ファイルの作成

NAMELIST 入力ファイルを再編集する. 以下の例で行っているのは, おもに入出力のファイルに関する設定だけあり, モデルの物理にかかわる設定の変更は何もしてない.

```
% cd /datadisk/horinout
% ls -l
total XXX
-rw-r--r--  1 horinout     1994 Mar  6 22:43 INDATA
-rw-r--r--  1 horinout   182432 Mar  6 22:14 Rest.init
-rw-r--r--  1 horinout     3088 Mar  6 22:15 SST.data
drwxr-xr-x  2 horinout      512 Mar  7 09:25 intg1/
% cat INDATA
 &NMRUN  RUN='AP.KUO.T10L5',    SIGN='horinouchi' &END
 &NMINTG TUNIT='MIN',      DELT=60,
         INISTP=3,         TFIL=0.05              &END
 &NMTIME TUNIT='DAY',
         START=0,          END=50,
         INTRST=0                                 &END
 &NMINIT FILE='Rest.init'                         &END
 &NMRSTR FILE='Rst.day50', DFMT='UR8'             &END
 &NMDATE IDATEI=0,1,1,0,0,0                       &END
 &NMBFIL HGRSST='SST.data',                       &END

 &NMDIAG FILE='CON',       UNIT='HOUR',
         INTDIA=24,        IDIAG=100,
         JDIAG=100,         KDIAG=-1,
         NCOLS=80                                 &END
 &NMHISD FILE=' ','CON',   TUNIT='HOUR','DAY',
         INTOUT=24,10,     INTSUM=0,0,
         AVRTYP=' ',       DFORM='UR4',
         INTEND=0                                 &END
 &NMMONI ISTR=1,           IINT=10,
         JSTR=4,           JINT=100,
         KSTR=1,           KINT=100,
         NCOLS=133                                &END
 &NMHIST ITEM='U',      FILE='intg1/u.data',      &END
```

```
 &NMHIST ITEM='V',      FILE='intg1/v.data',      &END
 &NMHIST ITEM='T',      FILE='intg1/t.data',      &END
 &NMHIST ITEM='PS',     FILE='intg1/ps.data',     &END
 &NMHIST ITEM='SIGDOT', FILE='intg1/sigdot.data', &END
 &NMHIST ITEM='Q',      FILE='intg1/q.data',      &END
 &NMHIST ITEM='RAIN',   FILE='intg1/rain.data',   &END
 &NMHIST ITEM='SENS',   FILE='intg1/sens.data',   &END
 &NMHIST ITEM='EVAP',   FILE='intg1/evap.data',   &END
 &NMHIST ITEM='QCNDC',  FILE='intg1/qcndc.data',  &END
 &NMHIST ITEM='VOR',   TUNIT='HOUR', INTOUT=48, INTSUM=0
                        FILE='intg1/vor.data',    &END
 &NMHIST ITEM='DIV',   TUNIT='HOUR', INTOUT=48, INTSUM=0
                        FILE='intg1/div.data',    &END
 &NMHIST ITEM='OLR',   TUNIT='HOUR', INTOUT=48, INTSUM=0
                        FILE='intg1/olr.data',    &END
 &NMHIST ITEM='QRADL', TUNIT='HOUR', INTOUT=48, INTSUM=0
                        FILE='intg1/qradl.data',  &END
```

ここで, ランの名前 AP.KUO.T10L5 は, 「Aqua-Planet で, 積雲パラメタリゼーションが KUO スキームの T10 切断 5 層のモデル」にちなんでつけたものである. 結果の出力は, 風速 (U,V) や温度 (T), 地表気圧 (PS) といった量以外に, 降水量 (RAIN), 顕熱・潜熱フラックス (SENS,EVAP), 積雲による加熱 (QCNDC), 長波放射 (OLR) なども出すようにしている. ほとんどの物理量は 1 日間隔で出力するが (ヒストリーデフォルト設定 &NMHISD の行を参照), 渦度 (VOR) など一部のファイルはサンプリング間隔を粗くして 2 日としている.

5.2.4　実行

実行方法は前章の例と同様である.

```
% cd /datadisk/horinout/
% mkdir intg1
% ~/dennou/agcm5/bin/agcm5.t10l5moist < INDATA | & tee std.out
```

これで, INDATA が読み込まれて, agcm5.t10l5moist が実行される. ここではコマンド tee を使って標準出力をファイル std.out にも落としている. 実行時間は Sun ワークステーション SS5 で以下の通りであった.

```
% tail -11 std.out
 ############### CPU TIME SUMMARY #################
 SETUP              0.276667D+01   0.271667D+01
 TIMCON             0.540115D+01   0.535109D+01
 PHYSCS             0.488534D+03   0.488317D+03
 DYNMCS             0.584149D+03   0.584066D+03
 PADJST             0.134199D+03   0.134165D+03
 TFILT              0.209024D+02   0.208690D+02
 MISC               0.350037D+00   0.300049D+00
```

```
 HISTIN             0.265144D+02   0.264976D+02
 HISTOU             0.983353D+01   0.911698D+01
    TOTAL TIME =    0.127353D+04   0.127187D+04
```

CPU 時間の合計は 21 分ちょっとであり, 要したメモリーは 8MB 弱であった. 各項を見ると, 力学過程 (DYNMCS) と物理過程 (PHYSCS) に多くの時間を要していることがわかる.

5.2.5 再実行

前小節の実行がうまくいっていることを確認すれば, 第 4.3.3 節参考にして, さらに時間積分を続けよう.

```
% cd /datadisk/horinout/
% mkdir intg2
```

また, 再実行用の NAMELIST 入力ファイル INDATA2 を用意する.

```
% diff INDATA INDATA2
5c5
<          START=0,          END=50,
---
>          START=50,          END=200,
7,8c7,8
<  &NMINIT FILE='Rest.init'                          &END
<  &NMRSTR FILE='Rst.day50', DFMT='UR8'              &END
---
>  &NMINIT FILE='Rst.day50'                          &END
>  &NMRSTR FILE='Rst.day200', DFMT='UR8'             &END
24,33c24,33
<  &NMHIST ITEM='U',      FILE='intg1/u.data',      &END
(中略)
<  &NMHIST ITEM='QCNDC',  FILE='intg1/qcndc.data',  &END
---
>  &NMHIST ITEM='U',      FILE='intg2/u.data',      &END
(中略)
>  &NMHIST ITEM='QCNDC',  FILE='intg2/qcndc.data',  &END
(以下略)
```

あとは,

```
% ~/dennou/agcm5/bin/agcm5.t10l5test < INDATA2
% ~/dennou/agcm5/bin/agcm5.t10l5moist < INDATA2 | & tee std.out2
```

で, 時間積分の延長 (再実行) が行なわれる. できたファイルを繋げるには次のようにすれば良い.

```
% cd intg1
% foreach file ( *data )
```

```
foreach? gtsel -apnd ../intg2/$file out:$file
foreach? end
```

これで, intg1/ の各ファイルの後ろに intg2/ の同名のファイルが付く. gtsel は gtool コマンドのひとつで, -apnd オプションを使っている.

5.3 gtool解析

AGCM5 の標準的な設定で湿潤大気大循環モデルを 200 日間走らせることができた. 得られた結果を gtool を用いて見ておこう.

5.3.1 アニメーション

まず, 帯状平均量の変遷を gtcont アニメーションで見てみよう.

```
% cd /datadisk/horinout/intg1
% gtcont -sw:lalt=t -sw:lwait=f str=2 step=2 x=0 t.data \
    cont=2.5,10 range=200,300 color=40 cpat=20999,2000
```

この例では, 帯状平均 (x=0) した温度 (\verbt.data!) の子午面コンター図を 2 日目 (str=2) から 2 日毎 (step=2) にアニメーション表示 (-sw:lalt=t -sw:lwait=f) している. 2 行目では等高線図でカラートーンを貼る指定をいろいろ行っている. コンター間隔はマイナーラインが 2.5K でメジャーラインが 10K(cont=2.5,10) とし, 表示する範囲は 200K から 300K まで (range=200,300) である. また, トーンの数は 40 (color=40) で, カラーマップの番号 20999 から 2000 番おき (cpat=20999,2000) の色をつけている.

トーンを以下のように陽に指定することもできる.

```
% gtcont -sw:lalt=t -sw:lwait=f str=2 step=2 x=0 t.data \
    tone=-999,240,270 pat=9999,0,8999
```

この例ではコンターはデフォルトに任せ, トーンは 270K 以下と 290K 以上とに貼っている. tone の最初の引数 -999 はデフォルト ($-\infty$) で, これを省略して , からはじめても良い.

帯状流も見てみよう.

```
% gtcont -sw:lalt=t -sw:lwait=f str=2 step=2 x=0 u.data \
    cont=2.5,10 range=-25,60 color=34 cpat=20999,2000
% gtcont -sw:lalt=t -sw:lwait=f str=2 step=2 x=0 u.data tone=,0 pat=14
```

徐々に亜熱帯ジェットが形成していく様がわかるはずである. 実行結果が同じかどうか確認するために, 200 日目の帯状平均した温度と帯状流の子午面図を図 5.1 に示しておく.

ところで, 雨はどう降っているのだろうか.

```
% gtcont -sw:lalt=t -sw:lwait=f rain.data \
    tone=50,200,800 pat=98999,5999,2999
```

激しい降水は低緯度に多いことが見てとれる. その他の物理量も見ておこう.

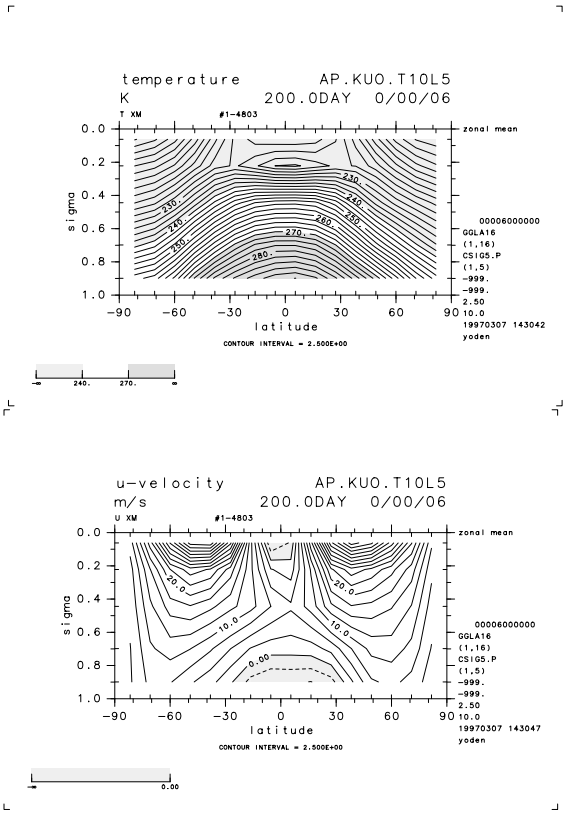

図 5.1: 湿潤大気大循環の 200 日積分結果の gtcont 出力例より. 温度 (左) と帯状風 (右).

```
% gtcont -sw:lalt=t -sw:lwait=f vor.data z=1
% gtcont -sw:lalt=t -sw:lwait=f vor.data z=4
% gtcont -sw:lalt=t -sw:lwait=f div.data z=1
% gtcont -sw:lalt=t -sw:lwait=f div.data z=4
```

渦度擾乱は主に中緯度に, 発散擾乱は主に低緯度にできているのがわかる.

5.3.2 データ解析

gtsel, gtseq を使って各種物理量の時系列を作って眺めよう.

```
% gtsel x=0  rain.data && gtseq out:zm_rain_seq.data
% gtcont -exch -greset              zm_rain_seq.data
% gtsel x=0 z=4 u.data && gtseq out:zm_u_z4_seq.data
% gtcont -exch                      zm_u_z4_seq.data
% gtsel str=100 y=4 z=3 v.data && gtseq out:y4_v_z3_seq.data
% gtcont -greset                            y4_v_z3_seq.data
% gtsel str=50  y=8   olr.data && gtseq out:y8_olr_seq.data
% gtcont -greset                            y8_olr_seq.data
% gtsel x=0 y=0 olr.data && gtseq out:global_olr_seq.data
% gtcurv y=1                         global_olr_seq.data
```

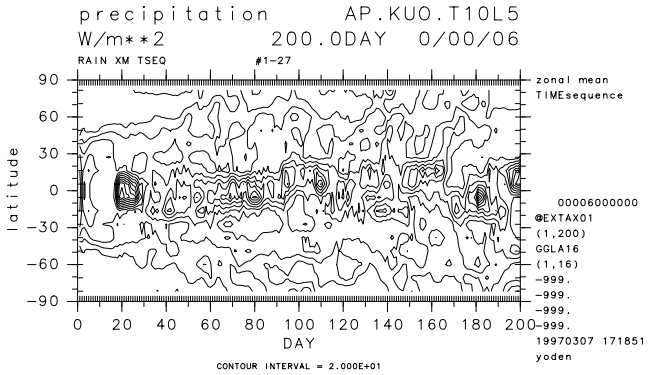

図 5.2: 湿潤大気大循環モデルで得られた帯状平均降水量の緯度-時間断面図.

切り方によって帯状平均量の時間発展やホフメラーダイアグラムが描けるはずである. 図 5.2 は帯状平均降水量の緯度-時間断面図である. ITCZ が形成され, 不規則に時間変動しているのがわかる (なぜ堀之内君の結果と微妙に違うのだろう? あなたの結果もこれと違っていますか?). 最後の例は, OLR の全球平均値を計算して, その時間変化を折れ線で図示するものである. `gtcurv` では本来 `y=1` と指定しなくてもうまく行くべきであろうが, 現状ではこれが必要である.

最後に「ごくらく GTOOL3 体験」で作った, 擾乱による南北熱フラックス ($\overline{T'v'}$) を計算する GTOOL コマンドスクリプトを順にかけてみる.

```
% ls -l eddy_heat_flux
-rwxr-xr-x  1 yoden         272 Mar  3 11:53 eddy_heat_flux*
% cat eddy_heat_flux
### GTOOL3 application script: eddy_heat_flux
### computation of eddy heat flux
%gtedy out:"tedy.data" t.data
gtedy out:"vedy.data" v.data
gtmlt out:"teve.data" tedy.data vedy.data
rm -f tedy.data vedy.data
gtsel x=0 out:"e_h_flux.data" teve.data
rm -f teve.data
% eddy_heat_flux
```

`e_h_flux.data` は, 擾乱による熱フラックス $\overline{T'v'}$ の緯度- 高度 2 次元データが 200 日分の時系列データとなったものである. この緯度-高度分布を等値線図でみたり, `gtseq` コマンドを通していろいろな時系列図を作ってみよう:

```
% gtcont cont=5 str=20 end=40  title:"eddy heat flux" unit:"Km/s" e_h_flux.data
% gtseq out:"e_h_flux_seq.data" e_h_flux.data
```

```
% gtcont y=1 cont=5 title:"eddy heat flux" unit:"Km/s" -exch e_h_flux_seq.data
% gtcurv x=4 y=1 title:"eddy heat flux" unit:"Km/s"          e_h_flux_seq.data
```

第6章　おわりに

以上で, AGCM5 に関連する資源の取得から実行・解析までのごくらく体験ができたはずである. おそらく全部のインストールを行なっても半日も要しなかったはずである. もっとも, GCM を走らせたというよりは, マニュアル通りに走らされたという印象かも知れない. だから, これで一人前, というわけでは全然ない. AGCM5 の使い方の第一歩を概観した程度である.

引き続いて, モデルの分解能はそのままにいろいろな実験を行ないたければ, NAMELIST 入力ファイルを変更して実行を繰り返せばよい. NAMELIST 入力ファイルの概要は, 「AGCM5 利用者の手引」第 3 章に書かれている. また, その詳細は, 「AGCM5 第 4 部 コードリファレンス」第 1 章にあるので, 参照すること.

これでどんどん実験ができる, というわけでもない. まず, エラーが出ずに計算が終了したとしても, それが全て正しい結果であるという保証はない. NAMELIST 入力ファイルの記述に誤りがあるかもしれないし, まだ見つかっていないバグが潜んでいるかもしれない. 自信をもって正しいというためには, 詳細な解析をおこなって, それぞれの結果が気象学の知識と矛盾しない, という確認が必要である. とくに, 研究目的の利用では, 新たな結果が基本的支配法則に従っていることの確認が必須であろう. 各計算過程をブラックボックスとして扱うのではなく, 個々のプログラムソースに戻って内容を把握しておくことも大切なことである.

97/02/14 余田 成男

第 A 章　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
```

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

93/07/09 林祥介
97/03/11 余田成男

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.