

らくらく dcpam5

地球流体電脳倶楽部

平成24年2月26日

目次

第1章	この文書について	1
第2章	設定ファイルを用いた実験設定の変更	2
2.1	解像度を変更するには	3
2.2	積分期間を変更するには	4
2.3	物理定数・惑星に関する定数を変更するには	5
2.4	出力設定を変更するには	6
2.5	リスタート計算を行うには	9
2.5.1	dcpam5でのリスタート計算の概要	9
2.5.2	リスタートファイルの出力のための設定	10
2.5.3	リスタート計算を行うための設定	10
第3章	ソースの変更・追加	13
3.1	一般的手順	13
3.2	初期値・海水面温度分布の変更	15
3.2.1	準備	15
3.2.2	作業用ディレクトリ作成	15
3.2.3	ソースプログラムの編集	16
3.2.4	実験実行用のディレクトリのセットアップ	17
3.2.5	実行ファイルの作成	18
3.2.6	実験の実行	19
3.2.7	最後に	19
3.3	出力する変数を増やす	20
3.3.1	準備	20
3.3.2	作業用ディレクトリ作成	20
3.3.3	ソースプログラムの編集	21
3.3.4	設定ファイルの編集	22
3.3.5	実験実行用のディレクトリのセットアップ	23
3.3.6	実行ファイルの作成	24
3.3.7	実験の実行	24
3.3.8	最後に	25
3.4	モジュールの変更・追加	25

3.4.1	準備	25
3.4.2	作業用ディレクトリ作成	26
3.4.3	モジュールプログラムの用意とプログラムの変更	27
3.4.4	設定ファイルの編集	27
3.4.5	実験実行用のディレクトリのセットアップ	27
3.4.6	実行ファイルの作成	28
3.4.7	実験の実行	29
3.4.8	最後に	29
第4章	鉛直 1 次元計算を行うには	30
4.1	はじめに	30
4.2	dcpam5 の鉛直 1 次元化の概要	30
4.3	コンパイル	31
4.4	鉛直 1 次元計算のための設定	31
4.4.1	格子点数の指定	31
4.4.2	力学過程の指定	32
4.4.3	緯度, 経度の指定	32
4.5	鉛直 1 次元計算の実行	32
第5章	軸対称 2 次元計算を行うには	33
5.1	はじめに	33
5.2	dcpam5 の軸対称 2 次元化の概要	33
5.3	コンパイル	34
5.4	軸対称 2 次元計算のための設定	34
5.4.1	格子点数の指定	34
5.5	軸対称 2 次元計算の実行	34
第6章	並列計算を行うには	35
6.1	はじめに	35
6.2	dcpam5 の MPI 並列化の概要	35
6.2.1	分割方法	35
6.2.2	入出力	36
6.3	コンパイル	37
6.3.1	必要なソフトウェアの準備	37
6.3.2	コンパイル時の注意	37
6.4	並列計算の実行	38
6.5	入出力データの分割と統合	39
6.5.1	入力データの分割	39
6.5.2	出力データの統合	39

付 録 A 使用上の注意とライセンス規定

1

第1章 この文書について

この文書は、地球流体電脳倶楽部で開発中の惑星大気モデル (Dennou-Club Planetary Atmospheric Model) のバージョン 5 である dcpam5 の使用方法について記すものである。入門的解説文書「らくらく dcpam5」¹ に比較して、より高度な使い方を解説する。

設定ファイルを用いた実験設定の変更, ソースプログラムを変更して実験する方法, 鉛直 1 次元・2 次元軸対称モデルの使用法, 並列計算について解説する。

¹http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/doc/tutorial/gokuraku/index.htm

第2章 設定ファイルを用いた実験設定の変更

この章では設定ファイル (NAMELIST ファイル) を用いた実験設定の変更方法について記す. NAMELIST 変数のリストは http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/doc/code_reference/htm/classes/NAMELIST.html に一覧があるので参照されたい.

設定ファイルを変更した後の実際の計算実行の方法については「らくらく dcpam5」http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/doc/gokuraku/ を参照されたい.

2.1 解像度を変更するには

解像度は, 設定ファイル (NAMELIST ファイル) に `&gridset_nml` を用いて設定する. 例えば設定ファイルの例である `dcpam_hs94_T21L20.conf`¹ には下のように設定されている.

```
&gridset_nml
  nmax = 21,           ! 最大全波数.
                       ! Maximum truncated wavenumber
  imax = 64,          ! 経度格子点数.
                       ! Number of grid points in longitude
  jmax = 32,          ! 緯度格子点数.
                       ! Number of grid points in latitude
  kmax = 20           ! 鉛直層数.
                       ! Number of vertical level
/
```

これは, T21L20 の解像度 (経度, 緯度, 鉛直方向の格子点数はそれぞれ 64, 32, 20) の設定である. これを T42L20 の解像度にするためには下のように設定する.

```
&gridset_nml
  nmax = 42,           ! 最大全波数.
                       ! Maximum truncated wavenumber
  imax = 128,         ! 経度格子点数.
                       ! Number of grid points in longitude
  jmax = 64,          ! 緯度格子点数.
                       ! Number of grid points in latitude
  kmax = 20           ! 鉛直層数.
                       ! Number of vertical level
/
```

なお, 鉛直解像度 (層数) を変更する場合には, `axeset_nml` の `Sigma` の値を編集し, 各層の配置も指定する必要がある.

¹http://www.gfd-dennou.org/www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/exp_setup_files/dcpam_hs94_T21L20.conf

2.2 積分期間を変更するには

積分期間は、設定ファイル (NAMELIST ファイル) に `×et_nml` を用いて設定する。dcpam では 2 つの方法のどちらかで積分期間を設定する。ひとつは積分する時間を設定する方法、もうひとつは積分開始と終了の時刻を指定する方法である。例えば、設定ファイルの例である `dcpam_hs94_T21L20.conf`² では、積分する時間を設定しており、下のように設定されている

```
&timeset_nml
      :
      IntegPeriodValue      = 10.0,
                              ! 計算終了時刻.
                              ! End time of calculation
      IntegPeriodUnit       = 'day',
                              ! 計算開始時刻の単位.
                              ! Unit of end time of calculation
      :
/
```

これは 10 日間積分することを表している。これを変更して積分時間を 20 日間にするためには下のように設定する。

```
&timeset_nml
      :
      IntegPeriodValue      = 20.0,
                              ! 計算終了時刻.
                              ! End time of calculation
      IntegPeriodUnit       = 'day',
                              ! 計算開始時刻の単位.
                              ! Unit of end time of calculation
      :
/
```

もうひとつの方法である、積分開始と終了の時刻を使って積分期間を設定するには下のように指定する。

²http://www.gfd-dennou.org/www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/dcpam_hs94_T21L20.conf


```

&timeset_nml
      :
      cal_type           = 'noleap',
      InitialYear       = 1,
      InitialMonth      = 1,
      InitialDay        = 1,
      InitialHour       = 0,
      InitialMin        = 0,
      InitialSec        = 0.0d0,
      EndYear          = 1,
      EndMonth          = 1,
      EndDay            = 3,
      EndHour           = 0,
      EndMin            = 0,
      EndSec            = 0.0d0,
      :
/

```

cal_type は暦の種類を指定し、上記で指定している noleap はうるう年のない暦を表している。InitialYear, InitialMonth, ... は開始年, 月, ... を表し, EndYear, EndMonth, ... は終了年, 月, ... を表している。その他の指定については, gtool5 チュートリアル, DCCalCreate の項目 http://www.gfd-dennou.org/library/gtool/gtool5/gtool5_current/doc/tutorial/dc_calendar1.htm を参照されたい。

2.3 物理定数・惑星に関する定数を変更するには

惑星に関する定数は、設定ファイル (NAMELIST ファイル) に &constants_nml を用いて設定します。例えば、設定ファイルの例である dcpam_hs94_T21L20.conf³ では、下のように設定されている。

```

&constants_nml
      RPlanet          = 6.371e6,
                        ! $ a $ [m].
                        ! 惑星半径.

```

³http://www.gfd-dennou.org/www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/dcpam_hs94_T21L20.conf

```

                                ! Radius of planet
Omega      = 7.292e-5,
                                ! $ \Omega $ [s-1].
                                ! 回転角速度.
                                ! Angular velocity
Grav       = 9.8,
                                ! $ g $ [m s-2].
                                ! 重力加速度.
                                ! Gravitational acceleration
CpDry      = 1004.0,
                                ! $ C_p $ [J kg-1 K-1].
                                ! 乾燥大気の定圧比熱.
                                ! Specific heat of air at constant pressure
GasRDry    = 286.85714285
                                ! $ R $ [J kg-1 K-1].
                                ! 乾燥大気の気体定数.
                                ! Gas constant of air
/

```

これら上から順に、惑星半径、自転角速度、重力加速度、大気の定圧比熱、大気の気体定数を指定している。これらの指定を変更することで、値を変更することができる。

2.4 出力設定を変更するには

解析用のヒストリデータの出力に関する設定は `>ool_historyauto_nml` を編集することで変更する。例えば設定ファイルの例である `dcpam_hs94_T21L20.conf`⁴ には下のように設定されている。

```

! ヒストリデータ出力の全体設定
! Global settings about history data output
!
&gtool_historyauto_nml
  IntValue = 1.0,

```

⁴http://www.gfd-dennou.org/www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/exp_setup_files/dcpam_hs94_T21L20.conf

```

! ヒストリデータの出力間隔の数値.
! 負の値を与えると, 出力を抑止します.
!
! Numerical value for interval of history data output
! Negative values suppress output.

IntUnit = 'day',

! ヒストリデータの出力間隔の単位.
! Unit for interval of history data output

Precision = 'float', ! 単精度
! ヒストリデータの精度.
! Precision of history data

FilePrefix = '',

! ヒストリデータのファイル名の接頭詞.
! Prefixes of history data filenames

/
!
! ヒストリデータ出力の個別設定
! Individual settings about history data output
!
&gtool_historyauto_nml
  Name = 'U, V, Temp, Ps, QVap, Vor, Div, SigDot, DPiDt'
/
&gtool_historyauto_nml
  Name = 'Mass, KinEngy, IntEngy, PotEngy, LatEngy, TotEngy, Enstro',
  SpaceAverage = .true., .true., .true., .true., .true.
/

```

それぞれの設定項目について以下に記す.

IntValue

(実数型) 出力間隔の数値

IntUnit

(文字型) 出力間隔の単位. "sec", "min", "hour", "day", "month", "year" などが使用可能である. 使用可能な単位の詳細については, gtool5 ライブラリ: dc_date_types モジュール http://www.gfd-dennou.org/library/gtool/gtool5/gtool5_current/doc/code_reference/classes/dc_date_types.html の "Characters list for unit" を参照されたい.

Precision

(文字型) データの精度. "float" (単精度実数型), "double" (倍精度実数型), "int" (整数型) を指定可能

SpaceAverage

(論理型配列) 空間平均のフラグ. 配列の 1 番目, 2 番目, 3 番目が, 経度, 緯度, 高度 () に対応する.

Name を指定しない, もしくは空文字を与えた場合, それは全ての変数に対するデフォルト設定となる. その場合にのみ有効な項目として以下のものがある.

FilePrefix

(文字型) データのファイル名の接頭詞. 例えば "exp1-" と指定すれば, 変数 "U" の出力ファイル名は "exp1-U.nc" となる. また, "data01/" のようにスラッシュを含む文字列を指定することで, カレントディレクトリ以外の場所に出力するよう設定することも可能である.

出力時間間隔などは, これらの値を編集することで変更することができる.

より詳しい説明については, gtool5 チュートリアル: 多数のファイル出力を行うモデルでのデータ出力 – 設定可能な項目 http://www.gfd-dennou.org/library/gtool/gtool5/gtool5_current/doc/tutorial/gtauto_first.htm#label-7 を参照されたい.

また, さらに出力する変数を追加する場合には,

```
&gtool_historyauto_nml
  Name = 'U, V, Temp, Ps, QVap, Vor, Div, SigDot, DPiDt'
/
```

の Name の行に変数名を追加するか, 新たに下のように行を追加する.

```
&gtool_historyauto_nml
  Name = 'XXX'
/
```

2.5 リスタート計算を行うには

この節では, dcpam5 でのリスタート計算の方法について述べる. ここ, リスタートとは, ある期間積分した後で, その最後の状態から計算を再開することを指す⁵.

2.5.1 dcpam5 でのリスタート計算の概要

dcpam5 のリスタート計算は, 以下の手順により行う.

- リスタートファイルの指定,
 - 大気中の予報変数用のリスタートファイル,
 - 惑星表面・土壌中の変数用のリスタートファイル,
- 予備変数用のファイルの指定,
- 計算再開時刻の指定.

つまり, 再計算のためには, それ以前の計算において

- 大気中の予報変数用のリスタートファイル,
- 惑星表面・土壌中の変数用のリスタートファイル,
- 予備変数用のファイル,

を出力しておく必要がある.

また, 現在の dcpam5 においては, 計算条件によっては, 正確なリスタート計算のためには制限がある. 具体的には, 地球計算, 火星計算において, リスタートファイルの出力時刻が放射計算の時刻と一致している必要がある⁶. 正確なリスタート計算を行う場合には, リスタートファイルの出力タイミングに注意すること.

⁵実際には, リスタートファイルが作成されていれば, 前回の積分の途中からの再開も可能である.

⁶dcpam5 においては, 計算時間の節約のために, 放射計算はすべての時間ステップで行っているわけではなく, ある一定の時間間隔でのみ行う. この放射計算のタイミングと異なる時間ステップにおいては, 前回の放射計算の結果を使用して時間積分する. したがって, 放射計算のタイミングと異なるタイミングで計算が終了してしまうと, リスタート計算開始時に前回の放射計算結果を

2.5.2 リスタートファイルの出力のための設定

リスタート計算を行う場合に必要となるリスタートファイルは、下のように指定することで出力される⁷。

大気中の変数用のファイル、惑星表面・土壌中の変数用のファイルは、それぞれ、dcpam5 の計算において以下の namelist ブロックで指定されることで出力される。

- 大気中の予報変数用のリスタートファイル,

```
&restart_file_io_nml
...
  OutputFile = 'ファイル名'
...
/
```

- 惑星表面・土壌中の変数用のリスタートファイル,

```
&restart_surftemp_io_nml
...
  OutputFile = 'ファイル名'
...
/
```

2.5.3 リスタート計算を行うための設定

リスタート計算を行う場合には、設定ファイル (namelist ファイル) に下のように指定する。

持っていないため、(正確な) リスタート計算ができない。もちろん、この放射計算に関わる予備変数を保存しておけば、(正確な) リスタート計算が可能である。地球流体電脳倶楽部大気大循環モデル AGCM5 のデフォルト放射モデルを用いた計算においては、放射計算に関わる予備変数もファイルに書き出しており、常に (正確な) リスタート計算が可能である (ことになっている)。

⁷出力指定していない場合にも、計算の終了時にリスタートファイルが作られる。この時のファイル名は、大気中の変数用ファイルは rst.nc、惑星表面・土壌中の変数用ファイルは rst_sst.nc、AGCM5 のデフォルト放射モデルで用いる予備変数用のファイルは rst_rad.nc となる。このため、明示的に指定しなくてもリスタートすることは可能である。

- 計算再開時刻,

```
&timeset_nml
...
RestartTimeValue = XXX
RestartTimeUnit  = YYY
...
/
```

なお, このとき, InitialYear, InitialMonth, 等 Initial* は, リスタート時刻ではなく, 初回の計算の時刻を指定するため, ここでは設定を変更する必要はない.

- 大気中の予報変数用のリスタートファイル,

```
&restart_file_io_nml
...
InputFile = 'ファイル名'
...
/
```

- 惑星表面・土壌中の変数用のリスタートファイル,

```
&restart_surftemp_io_nml
...
InputFile = 'ファイル名'
...
/
```

namelist ブロック, timeset_nml, に指定する RestartTimeValue, RestartTimeUnit の値は, 実際には restart_file_io_nml の InputFile に指定されるファイルの中の変数 time のある値とするのが良い. 例えば,

```
&restart_file_io_nml
...
InputFile = 'input.nc'
...
/
```

であり,

```
% ncdump -v time input.nc
netcdf input {
    ...
    double time(time) ;
        time:long_name = "time" ;
        time:units = "sec" ;
    ...
    time = 0, 86400, 172800 ;
}
```

の場合に, 前回の計算の終了時からの再計算を行う場合には, 下のよう指定する.

```
&timeset_nml
    ...
    RestartTimeValue = 172800.0
    RestartTimeUnit = 'sec'
    ...
/
&restart_file_io_nml
    ...
    InputFile = 'input.nc'
    ...
/
&restart_surftemp_io_nml
    ...
    InputFile = 'input_surf.nc'
    ...
/
```

なお, RestartTimeValue に与える数値は, 倍精度で書いてもよい. また, ここでは惑星表面・土壌中の変数用のリスタートファイル名を input_surf.nc とした.

第3章 ソースを変更・追加して計算 を実行したい場合には

この章では、ソースプログラムを変更あるいは追加して実行ファイルを作成し計算を行う際のお勧めの方法について記す。

3.1 ソースを変更・追加する一般的手順

dcpam5 のソースプログラムを変更あるいは追加したい場合には、dcpam5 のオリジナルソースツリー内にあるファイルを直接編集したりツリー内にファイルを追加したりすることは避けて頂きたい。以下の手順のように、オリジナルソースツリーとは別のディレクトリを用意してその下で作業することを勧める。そのための手順は一般的に次のようになるだろう。

1. ライブラリとモジュールファイルの作成
オリジナルソースツリーで dcpam5 を configure, コンパイルしておく。
2. 実験用ディレクトリの作成
オリジナルソースツリーの外に実験用ディレクトリを作成する。
3. ソースプログラムの編集・追加
改変したいソースプログラムファイルをオリジナルソースからコピーし、編集する。あるいは新たに追加したいモジュールのソースプログラムを作成する。
4. 設定ファイル (namelist ファイル) の準備
実験用設定ファイルを作業ディレクトリに用意する。オリジナルソースの例をコピーし必要に応じて編集するのが簡単だろう。
5. 実験用のディレクトリのセットアップ
実行ファイル作成のための環境 (Makefile 等) を用意する。

6. 実行ファイルの作成

make コマンドで実行ファイルを作成する.

7. 実験の実行

作業ディレクトリは以下のような構造を想定している.

```
top-directory/  
  /exp-name1/  
    /src/           # 非標準ソースオリジナル  
      /main        # 実行プログラムソース置き場  
      /otherdir1   # モジュールソース置き場  
      ...  
    /conf          # 非標準設定ファイルオリジナル  
    /bin           # 実行ファイル置き場  
    /include       # モジュールファイル置き場  
    /data1         # データ置き場 1  
    /data2         # データ置き場 2  
    ...  
  /exp-name2/  
    ...
```

3.2 初期値・海水面温度分布などを変更するには

この節では、ソースプログラムを改変して計算実行する一例として初期値・海水面温度分布などを変更するための手順を記す。具体的な例題として水惑星実験 (APE:Aqua Planet Experiments) の海水面温度分布を変更してみる。実際に改造するソースプログラムは海水面温度データ作成プログラムソース ”src/main/sst_data.f90” であるが、適宜初期値データ作成プログラムソース ”src/main/init_data.f90” に置き換えることで初期値データの変更もできるだろう。

3.2.1 準備

インストールガイド (http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/INSTALL.htm) にしたがって、dcpam5 をコンパイルしてライブラリ (lib/libdcpam5.a) とモジュールファイル (include/*.mod 等) を作成しておく。

3.2.2 作業用ディレクトリ作成

まず dcpam5 ソースのトップディレクトリ (以下の例では dcpam5-YYYYMMDD とする) に移動しておく。次にソース変更作業と実験用ディレクトリを dcpam5 ソースツリー外部に作成する。ここでは dcpam5 ソースツリーの隣に dcpam5-exp/icitsst ディレクトリを作成し、その中で作業を行うことにする。

```
% mkdir -p ../dcpam5-exp/icitsst
```

作成した実験用ディレクトリに移り、その下に実験専用のソースファイル置き場設定ファイル (NAMELIST ファイル) 置き場を作成する。

```
% cd ../dcpam5-exp/icitsst
% mkdir -p src/main
% mkdir conf
```

初期値作成, 海水面温度データ作成のソースプログラムとモデル本体のソースプログラムを "src/main" ディレクトリにコピーする. 用いる設定ファイル (の元) を "conf" ディレクトリにコピーする.

```
% cp ../../dcpam5-YYYYMMDD/src/main/*.f90 src/main
% cp ../../dcpam5-YYYYMMDD/conf/dcpam_ape_T21L16.conf conf
% cp ../../dcpam5-YYYYMMDD/conf/init_data_T21L16.conf conf
% cp ../../dcpam5-YYYYMMDD/conf/sst_data_T21.conf conf
```

3.2.3 ソースプログラムの編集

コピーしたプログラムソースファイルを編集し, 作成したい初期値分布あるいは海水面温度分布をプログラムする. 例として, "src/main/sst_data.f90" を編集して, 海水面温度分布を $250 + 50 \sin^2 \varphi$ に変更してみよう (φ は緯度). まず, 緯度情報を用いるために先頭のモジュール使用宣言部部分に "gridset" モジュールの "imax, jmax" を用いるよう追加する (各行にある!以降の部分はコメントなので入力を省略しても構わない).

```
! 格子点設定
! Grid points settings
!
use gridset, only: imax, & ! 経度格子点数.
                        ! Number of grid points in longitude
&                        jmax ! 緯度格子点数.
                        ! Number of grid points in latitude
```

さらに "axeset" モジュールの "y_Lat" 変数を用いるよう追加する.

```
! 座標データ
! Axes data
!
use axeset, only: y_Lat
```

作業変数の宣言部にて DO loop 変数 i, j を宣言しておく.

```
integer :: i, j           ! Do loop variable
```

次に, 海水面温度を設定している箇所

```
! 地表面データの作成
! Generate surface data
!
call RestartSurfTempOutput( &
    & xy_SurfTemp )           ! (in)
```

を次のように変更する.

```
! 地表面データの作成
! Generate surface data
!
!!!call RestartSurfTempOutput( &
!!! & xy_SurfTemp )           ! (in)
!
do j=1,jmax
  do i=0,imax
    xy_SurfTemp(i,j) = 250.0d0 + 50.0d0*sin(y_Lat(j))**2
  end do
end do
```

3.2.4 実験実行用のディレクトリのセットアップ

さて, ファイルの準備が整ったら, 実験実行用のディレクトリをセットアップする. そのためには dcpam5 のソースツリートップに戻って”make expdir” を実行する. すると, 作業ディレクトリトップの名前と非標準ソースディレクトリ名並びに実験ディレクトリの名前をきかれるので, それらを入力する.

```

% cd ../../dcpam5-YYYYMMDD
% make expdir

Enter top directory name []: ../dcpam5-exp
Enter experimet directory name []: initsst

*** "../dcpam5-exp/initsst" is already exist ***

Directory in which non-standard files are prepared
[../dcpam5-exp/initsst/src]: Creating "../dcpam5-exp/initsst/Makefile"
Creating "../dcpam5-exp/initsst/src/Makefile" ... done.
Creating "../dcpam5-exp/initsst/src/main/Makefile" ... ls: ../dcpam5-exp
のようなファイルやディレクトリはありません
done.
Creating "../dcpam5-exp/initsst/Config.mk" ... done.
Creating "../dcpam5-exp/initsst/rules.make" ... done.

*** Setup of "../dcpam5-exp/initsst" is complete ***

```

すると, ../dcpam5-exp/initsst/ に Config.mk と rules.make ならびに src 以下の各サブディレクトリの Makefile が作成される。

3.2.5 実行ファイルの作成

実行ファイルを作成しよう。そのためには実験ディレクトリ ”../dcpam5-exp/initsst” に移って”make”を行う¹。

```

% cd ../dcpam5-exp/initsst
% make

```

コンパイルエラーが出てしまったら、先程編集したファイルを修正し、再び ”make”を行う。エラーがなくなるまでこの作業を繰り返す。

¹環境変数 FFLAGS を dcpam5 ライブラリを作成したときと同じ値にしておく必要があるかもしれない。

めでたくエラーがなくなり、実行ファイルができ上がったら

```
% make instal
```

を実行する。すると実行ファイルが "bin" ディレクトリにインストールされる。

3.2.6 実験の実行

実行の仕方はごらく dcpam の水惑星実験の手順と一緒である。まず初期値、海水面温度データを作成する。

```
% bin/sst_data -N=./conf/sst_data_T21.conf  
% bin/init_data -N=./conf/init_data_T21L16.conf
```

そして、実験を実行するには、

```
% bin/dcpam_main -N=./conf/dcpam_ape_T21L16.conf \  
>& dcpam_ape_T21L16.log &
```

といった具合である。

簡単な解析と可視化については、「ごらく dcpam5」の「簡単な解析・可視化」を参照のこと。

3.2.7 最後に

実験のために修正したファイルらは別の場所にコピー保存しておくことを勧める。

```
% cp src/main/sst_data.f90 [somewhere]  
% cp ...
```

3.3 出力する変数を増やすには

この節では、出力する変数を増やすためのソースプログラムを改変と実行ファイルを作成するための手順を記す。具体的な例題として Held and Schuarz (1994) のベンチマーク実験でを追加出力することを試みる。実際の改変は、

```
HistoryAutoAddVariable    ! 出力変数の定義
HistoryAutoPut            ! データ出力
```

の2項目を追加することになる。そのほかに実験設定ファイルにおいて、新たに出力したい変数名を追加する必要がある。

以下の例では、実際に改造するソースプログラムがモジュールソースファイル”src/held_suarez_1994/held_suarez_1994.f90”であるが、他のモジュールプログラムにおいても同様の手順で出力データの追加をしたプログラムを作成・実行できるだろう。

また、前章「設定ファイルを用いた実験設定の変更」での「出力設定を変更するには」の節、あるいは「らくらく dcpam5」の「実験条件の変更/出力設定の変更」https://www.gfd-dennou.org/GFD_Dennou_Club/dc-arch/takepiro/dcpam5-cvs/doc/gokuraku/changesetup.htm#label-4も参照されたい。

3.3.1 準備

インストールガイド (http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/INSTALL.htm) にしたがって、dcpam5をコンパイルしてライブラリ (lib/libdcpam5.a) とモジュールファイル (include/*.mod 等) を作成しておく。

3.3.2 作業用ディレクトリ作成

まず dcpam5 ソースのトップディレクトリ (以下の例では dcpam5-YYYYMMDD とする) に移動しておく。次にソース変更作業と実験用ディレクトリを dcpam5 ソースツリー外部に作成する。ここでは dcpam5 ソースツリーの隣に dcpam5-exp/addoutput ディレクトリを作成し、その中で作業を行うことにする。


```
% mkdir -p ../dcpam5-exp/addoutput
```

作成した実験用ディレクトリに移り, その下に実験専用のソースファイル置き場設定ファイル (NAMELIST ファイル) 置き場を作成する.

```
% cd ../dcpam5-exp/addoutput
% mkdir -p src/main
% mkdir src/held_suarez_1994
% mkdir conf
```

初期値作成のソースプログラムとモデル本体のソースプログラムを "src/main" ディレクトリにコピーする. 用いる設定ファイル(の元)を "conf" ディレクトリにコピーする.

```
% cp ../../dcpam5-YYYYMMDD/src/main/dcpam_main.f90 src/main
% cp ../../dcpam5-YYYYMMDD/src/main/init_data.f90 src/main
% cp ../../dcpam5-cvs/src/held_suarez_1994/held_suarez_1994.f90 \
  src/held_suarez_1994
% cp ../../dcpam5-YYYYMMDD/conf/dcpam_hs94_T21L20.conf conf
% cp ../../dcpam5-YYYYMMDD/conf/init_data_hs94_T21L20.conf conf
```

3.3.3 ソースプログラムの編集

コピーしたプログラムソースファイルを編集し出力する変数を追加する. 例として, "src/held_suarez_1994/held_suarez_1994.f90" を編集して, 温位 $\theta = T(p_0/p)^{\kappa}$ を出力に追加してみよう

まず, ヒストリデータ出力のためのへの変数登録箇所において温位を出力変数として追加する. 'subroutine Hs94Init' 中の

```
call HistoryAutoAddVariable( 'TempEQHS94', &
  & (/ 'lon ', 'lat ', 'sig ', 'time' /), &
```

```
& 'equilibrium temperature', 'K' )
```

の下の行(507 行目あたり)に”HistoryAddVariable” の項目を次のように追加する.

```
call HistoryAutoAddVariable( 'PTemp' , &
  & (/ 'lon ', 'lat ', 'sig ', 'time' /), &
  & 'potential temperature', 'K' )
```

次に変数の出力を追加する. 'subroutine HS94Forcing' 中のヒストリーデータ出力の箇所

```
! ヒストリデータ出力
! History data output
!
call HistoryAutoPut( TimeN, 'DUDtHS94', xyz_DUDt )
call HistoryAutoPut( TimeN, 'DVDtHS94', xyz_DVDt )
call HistoryAutoPut( TimeN, 'DTempDtHS94', xyz_DTempDt )
call HistoryAutoPut( TimeN, 'TempEQHS94', xyz_TempEQ )
```

の下に, 次の行を追加する.

```
call HistoryAutoPut( TimeN, 'PTemp', xyz_Temp*(1.0d5/xyz_Press)**Kappa)
```

3.3.4 設定ファイルの編集

実験用設定ファイルに, 新たに出力したい変数名を追加する. ./conf/dcpam_hs94_T21L20.conf の最後の方の「ヒストリデータ出力の個別設定」のリストに ”PTemp” を追加する.

```
!
! ヒストリデータ出力の個別設定
```

```

! Individual settings about history data output
!
&gtool_historyauto_nml
  Name = 'U, V, Temp, Ps, QVap, SigDot, OMG, TempEQHS94, PTemp'
/

```

3.3.5 実験実行用のディレクトリのセットアップ

さて、ファイルの準備が整ったら、実験実行用のディレクトリをセットアップする。そのためには dcpam5 のソースツリートップに戻って”make expdir” を実行する。すると、作業ディレクトリトップの名前と非標準ソースディレクトリ名並びに実験ディレクトリの名前をきかれるので、それらを入力する。

```

% cd ../../dcpam5-YYYYMMDD
% make expdir
sh ./setup_expdir_nonstd.sh

***** Setup a directory for a experiment *****

Enter top directory name []: ../dcpam5-exp
Enter experimet directory name []: addoutput

*** "../dcpam5-exp/addoutput" is already exist ***

Directory in which non-standard files are prepared
[../dcpam5-exp/addoutput/src]: Creating "../dcpam5-exp/addoutput/Makefile"
Creating "../dcpam5-exp/addoutput/src/Makefile" ... done.
Creating "../dcpam5-exp/addoutput/src/main/Makefile" ... ls: ../dcpam5-exp/ad
のようなファイルやディレクトリはありません
done.
Creating "../dcpam5-exp/addoutput/src/held_suarez_1994/Makefile" ... ls: ../d
のようなファイルやディレクトリはありません
done.
Creating "../dcpam5-exp/addoutput/Config.mk" ... done.
Creating "../dcpam5-exp/addoutput/rules.make" ... done.

```

```
*** Setup of "../dcpam5-exp/addoutput" is complete ***
```

すると, ../dcpam5-exp/addoutput/ に Config.mk と rules.make ならびに src 以下の各サブディレクトリの Makefile が作成される.

3.3.6 実行ファイルの作成

実行ファイルを作成しよう. そのためには実験ディレクトリ "../dcpam5-exp/addoutput" に移って"make" を行う².

```
% cd ../dcpam5-exp/addoutput
% make
```

コンパイルエラーが出てしまったら, 先程編集したファイルを修正し, 再び"make" を行う. エラーがなくなるまでこの作業を繰り返す.

めでたくエラーがなくなり, 実行ファイルができ上がったら

```
% make instal
```

を実行する. すると実行ファイルが"bin" ディレクトリにインストールされる.

3.3.7 実験の実行

実行の仕方はごらく dcpam の Held and Suarez (1994) 実験の手順と一緒にである. まず初期値データを作成する.

²環境変数 FFLAGS を dcpam5 ライブラリを作成したときと同じ値にしておく必要があるかもしれない.

```
% bin/init_data -N=./conf/init_data_hs94_T21L20.conf
```

そして, 実験を実行するには,

```
% bin/dcpam_main -N=./conf/dcpam_hs94_T21L20.conf \  
>& dcpam_hs94_T21L20.log &
```

といった具合である. "PTemp.nc" が作成されていたら成功である.

簡単な解析と可視化については, 「らくらく dcpam5」の「簡単な解析・可視化」を参照のこと.

3.3.8 最後に

実験のために修正したファイルらは別の場所にコピー保存しておくことを勧める.

```
% cp src/held_suarez_1994/held_suarez_1994.f90 [somewhere]  
% cp conf/dcpam_hs94_T21L20 [somewhere]  
% cp ...
```

3.4 モジュールを変更・追加するには

この節では, モジュールを変更・追加するための一般的な手順を記す. モジュールを変更する場合に関しては先の節「出力する変数を増やすには」も参考にされたい.

3.4.1 準備

インストールガイド (http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/INSTALL.htm) にしたがって, dcpam5 をコンパイルしてライブラリ (lib/libdcpam5.a) とモジュールファイル (include/*.mod 等) を作成しておく.

3.4.2 作業用ディレクトリ作成

まず dcpam5 ソースのトップディレクトリ (以下の例では dcpam5-YYYYMMDD とする) に移動しておく。次にソース変更作業と実験用ディレクトリを dcpam5 ソースツリー外部に作成する。ここでは dcpam5 ソースツリーの隣に dcpam5-exp/addmodule ディレクトリを作成し、その中で作業を行うことにする。

```
% mkdir -p ../dcpam5-exp/addmodule
```

作成した実験用ディレクトリに移り、その下に実験専用のソースファイル置き場を作成する。実行プログラムソースファイル用のディレクトリ "src/main" に加えて、新たなモジュール、あるいは変更するモジュールのソースファイル置き場として適当な名前のディレクトリを "src" 以下に作成する (以下の例では foo とする)。さらに設定ファイル (NAMELIST ファイル) 置き場を作成する。

```
% cd ../dcpam5-exp/addoutput
% mkdir -p src/main
% mkdir src/foo
% mkdir conf
```

実行ファイルプログラムを "src/main" ディレクトリにコピーする。モジュールを変更したい場合には、そのソースプログラムをコピーする (以下の例では bar.f90 としている)。また、用いる設定ファイル (の元) を "conf" ディレクトリにコピーする (以下の例では xxx.conf としている。設定ファイル (の元) は一般に実行用と初期値作成用ならびに地表面温度作成用の 3 種類が必要となる)。

```
% cp ../../dcpam5-YYYYMMDD/src/main/*_main.f90 src/main
% cp ../../dcpam5-YYYYMMDD/src/foo/bar.f90 src/foo
% cp ../../dcpam5-YYYYMMDD/conf/xxx.conf conf
```

3.4.3 モジュールプログラムの用意とプログラムの変更

モジュールを変更する場合には、コピーしたモジュールソースプログラム ("src/foo/bar.f90") を編集し変更する。

新しいモジュールを用意する場合には、そのソースファイルを "src/foo" ディレクトリにおく。モジュール名とファイル名の拡張子を取り除いた名前はおなじにしておかねばならないことに注意されたい。そして新しいモジュールをプログラムで使うべく、他のプログラムソースファイルも編集する必要があるだろう。そのようなファイルも "src" 以下に適当なディレクトリを作成し格納しておく (もちろん foo ディレクトリに並べておいても良い)。

3.4.4 設定ファイルの編集

新たに追加したモジュールにおいて NAMELIST 変数を追加した場合には実験用設定ファイルを適宜変更する。

3.4.5 実験実行用のディレクトリのセットアップ

さて、ファイルの準備が整ったら、実験実行用のディレクトリをセットアップする。そのためには dcpam5 のソースツリートップに戻って "make expdir" を実行する。すると、作業ディレクトリトップの名前と非標準ソースディレクトリ名並びに実験ディレクトリの名前をきかれるので、それらを入力する。

```
% cd ../../dcpam5-YYYYMMDD
% make expdir
sh ./setup_expdir_nonstd.sh

***** Setup a directory for a experiment *****

Enter top directory name []: ../dcpam5-exp
Enter experimet directory name []: addmodule

*** "../dcpam5-exp/addoutput" is already exist ***
```

```

Directory in which non-standard files are prepared
[../dcpam5-exp/addoutput/src]: Creating "../dcpam5-exp/addoutput/Makefile" ... done.
Creating "../dcpam5-exp/addoutput/src/Makefile" ... done.
Creating "../dcpam5-exp/addoutput/src/main/Makefile" ... ls: ../dcpam5-
のようなファイルやディレクトリはありません
done.
Creating "../dcpam5-exp/addoutput/src/held_suarez_1994/Makefile" ... ls
のようなファイルやディレクトリはありません
done.
Creating "../dcpam5-exp/addoutput/Config.mk" ... done.
Creating "../dcpam5-exp/addoutput/rules.make" ... done.

*** Setup of "../dcpam5-exp/addoutput" is complete ***

```

すると, ../dcpam5-exp/addmodule/ に Config.mk と rules.make ならびに src 以下の各サブディレクトリの Makefile が作成される。

3.4.6 実行ファイルの作成

実行ファイルを作成しよう。そのためには実験ディレクトリ ”../dcpam5-exp/addmodule” に移って”make”を行う³。

```

% cd ../dcpam5-exp/addoutput
% make

```

コンパイルエラーが出てしまったら、先程編集したファイルを修正し、再び ”make”を行う。エラーがなくなるまでこの作業を繰り返す。

めでたくエラーがなくなり、実行ファイルができ上がったら

```

% make instal

```

³環境変数 FFLAGS を dcpam5 ライブラリを作成したときと同じ値にしておく必要があるかもしれない。

を実行する。すると実行ファイルが "bin" ディレクトリにインストールされる。

3.4.7 実験の実行

実行の仕方はごらく dcpam での各実験の手順とおなじである。まず初期値データを作成する。

```
% bin/init_data -N=./conf/init_data...conf
```

必要ならば海水面温度分布データを作成する。

```
% bin/sst_data -N=./conf/sst_data...conf
```

そして、実験を実行するには、

```
% bin/dcpam_main -N=./conf/dcpam_...conf  
>& dcpam_....log &
```

といった具合である。

簡単な解析と可視化については、「ごらく dcpam5」の「簡単な解析・可視化」を参照のこと。

3.4.8 最後に

実験のために修正したファイルらは別の場所にコピー保存しておくことを勧める。

第4章 鉛直 1 次元計算を行うには

4.1 はじめに

dcpam5 は 3 次元モデルであるが、設定を変更することで鉛直 1 次元計算も可能である。この章では、dcpam5 を用いて鉛直 1 次元計算の実行方法について述べる。

4.2 dcpam5 の鉛直 1 次元化の概要

鉛直 1 次元計算の実行方法について述べる前に、dcpam5 での鉛直 1 次元化の概要について簡単に説明しておく。

dcpam5 の鉛直 1 次元化は、緯度、経度方向の格子点数をそれぞれ 1 にし、移流を計算しないように設定することで実現している¹。移流計算以外は 3 次元計算の時に用いていたモジュールをそのまま用いている。

このため、例えば短波放射計算のように惑星上の緯度、経度に依存するような計算のために、計算する 1 次元カラムが位置する緯度、経度を指定する必要があり、その位置における適当な日変化、季節変化が計算される²。

また、例えば海表面温度など、外部データを読み込む際には、モデルに合わせて緯度、経度方向の格子点数がそれぞれ 1 のデータを用意する必要がある。

¹鉛直一次元計算においては、水平方向の格子点数が 1 であるため、スペクトル変換法を用いた力学過程モジュールでは計算できない。このため、スペクトル変換法を用いた力学過程を用いないモジュールを選択する必要がある。(yot, 2011/09/30)

²もちろん、日変化、季節変化をなくす設定をすれば話は別。

4.3 コンパイル

鉛直 1 次元計算を行うためのコンパイル方法は、3 次元モデルのコンパイル方法と全く同じである。コンパイルの基本的な方法の詳細は、「dcpam5 インストールガイド」(http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/INSTALL.htm) を参照すること。

4.4 鉛直 1 次元計算のための設定

鉛直 1 次元計算の特有の設定は、水平格子点数、力学過程、鉛直 1 次元カラムの緯度、経度の指定である。それぞれ、下に示すように指定する。

4.4.1 格子点数の指定

既に述べたように、鉛直 1 次元計算は、緯度、経度方向の格子点数をそれぞれ 1 にすることで実現している。緯度、経度方向の格子点数は、`gridset_nml` namelist ブロックにより、下のように指定する。

```
&gridset_nml
  ...
  imax   = 1,           ! 経度格子点数.
                        ! Number of grid points in longitude
  jmax   = 1,           ! 緯度格子点数.
                        ! Number of grid points in latitude
  kmax   = 16,          ! 鉛直層数.
                        ! Number of vertical level
  kslmax = 9            ! 地下の鉛直層数.
                        ! Number of subsurface vertical level
/
```

4.4.2 力学過程の指定

既に述べたように、鉛直 1 次元計算では、スペクトル変換法を用いた力学過程モジュールを使えないため、別のモジュールを選択する³。異なるモジュールは、下のよう指定する。

```
&dcpam_main_nml
...
  DynMode                = 'NoHorAdv',
...
/
```

4.4.3 緯度、経度の指定

既に述べたように、鉛直 1 次元計算では、そのカラムの緯度、経度を指定する必要がある。緯度、経度方向の格子点数は、axeset_nml namelist ブロックにより、下のよう指定する。

```
&axeset_nml
  LonInDeg = 0.0d0,    ! 経度 (degree)
  LatInDeg = 0.0d0    ! 緯度 (degree)
/
```

4.5 鉛直 1 次元計算の実行

鉛直 1 次元計算の実行方法は、3 次元モデルの実行方法と全く同じである。詳細は、「らくらく dcpam5」(http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/doc/tutorial/gokuraku/) を参照すること。

³この「別のモジュール」は、与えられた物理過程による時間変化率を用いて時間積分する。

第5章 軸対称 2 次元計算を行うには

5.1 はじめに

dcpam5 は、軸対称 2 次元計算に用いることができる。この章では、dcpam5 を用いた軸対称 2 次元計算の実行方法について述べる。

5.2 dcpam5 の軸対称 2 次元化の概要

軸対称 2 次元計算のための準備と実行方法について述べる前に、dcpam5 での軸対称 2 次元化の概要について簡単に説明しておく。

軸対称 2 次元化は、下のふたつの方法によって実装している。

- 移流計算におけるスペクトル変換に spml の wa_zonal_module モジュールを用いる、
- 経度方向の格子点数を 1 にする。

移流計算において用いる spml のスペクトル変換モジュールは、コンパイル時にプリプロセッサオプションで指定することによって選択する。経度方向の格子点数は、計算実行時の設定ファイル (namelist ファイル) で指定する。

なお、移流計算におけるスペクトル変換以外は 3 次元計算の時に用いていたモジュールをそのまま用いている。このとき、計算される子午面は、経度 0° における子午面として扱われ、例えば短波放射計算のように惑星上の経度 (地方時) に依存するような計算においては、経度 0° における日変化、季節変化が計算される¹。

¹もちろん、日変化、季節変化をなくす設定をすれば話は別。

5.3 コンパイル

コンパイルの基本的な方法は逐次版と同じであり, 詳細は, 「dcpam5 インストールガイド」(http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/INSTALL.htm) を参照すること. ただし, 下の点に注意すること.

- 環境変数の FFLAGS に `-DAXISYMMETRY` または `-DAXISYMMETRY_SJPACK` を指定する.

5.4 軸対称 2次元計算のための設定

軸対称 2次元計算の特有の設定は, 格子点数の指定である. 下に示すように指定する.

5.4.1 格子点数の指定

既に述べたように, 軸対称 2次元計算は, 経度方向の格子点数を 1 にすることで実現している. 経度方向の格子点数は, `gridset_nml` namelist ブロックにより, 下のよう

```
&gridset_nml
  ...
  imax      = 1           ! 経度格子点数.
                        ! Number of grid points in longitude
  ...
/
```

5.5 軸対称 2次元計算の実行

軸対称 2次元計算の実行方法は, 3次元モデルの実行方法と全く同じである. 詳細は, 「らくらく dcpam5」(http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/doc/tutorial/gokuraku/) を参照すること.

第6章 並列計算を行うには

6.1 はじめに

dcpam5 は, MPI (Message Passing Interface) を用いて並列化されている¹. この章では, dcpam5 を用いた並列計算の実行方法について述べる.

6.2 dcpam5 の MPI 並列化の概要

並列計算のための準備と実行方法について述べる前に, dcpam5 での MPI 並列実装の概要について簡単に説明しておく.

6.2.1 分割方法

dcpam5 の MPI 並列化においては, 全球の格子点を緯度方向に分割する². つまり, 各 MPI プロセスは, ある緯度帯の経度-緯度(帯)-高度の 3 次元データを保持しており, 必要に応じて MPI ライブラリを用いて通信を行う. 緯度方向の分割方法は, 現在の dcpam5 が移流計算に用いている ispack の MPI 並列化の方法に従っている. 例えば, T42 の水平解像度で, 4 並列で計算する場合, 各プロセスは以下の緯度

¹dcpam5 の移流計算においてスペクトル変換に用いている ispack が OpenMP を用いて並列化されているため, 移流計算部分は OpenMP での並列計算が可能である. しかし, 他の部分は OpenMP 並列に対応していないため, OpenMP での並列計算は実用的ではないだろう.

²ここでは実空間の分割についてのみ述べる. 現在の dcpam5 では, 移流の計算にスペクトル法を用いており, 波数空間のデータの保有方法・分割方法については別途説明が必要であるが, ここでは省略する.

帯のデータを保持する^{3,4}.

```
process 0 : -20.9° ≤ φ ≤ 20.9°,
process 1 : -43.3° ≤ φ ≤ -23.7°, 23.7° ≤ φ ≤ 43.3°,
process 2 : -65.6° ≤ φ ≤ -46.0°, 46.0° ≤ φ ≤ 65.6°,
process 3 : -87.9° ≤ φ ≤ -68.4°, 68.4° ≤ φ ≤ 87.9°.
```

分割方法の詳細については, ispack の文書を参照すること.

6.2.2 入出力

現在の dcpam5 においては, 入出力は各プロセスごとに行っている. したがって, 入力データは各プロセス用に準備する必要がある. 同様に, 出力データも各プロセスごとに別のファイルに分割されているため, 必要に応じてそれらのデータを統合する必要がある.

この時, dcpam5 の入出力ファイル名には MPI のプロセス番号を含めており, ファイル名は *_rank000000.nc, *_rank000001.nc, *_rank000002.nc, ... の書式となる⁵.

ただし, 設定ファイル (実行時の namelist ファイル) でのファイル名の指定には, プロセス番号 _rank000000, _rank000001, _rank000002, ... の部分は含めず, 例えば, 初期値ファイル・リスタートファイルの名前は, 設定ファイルにおいて下のよう

³実際に保持される格子点の緯度は下のようになる.

```
process 0 : 1.4°S/N, 4.2°S/N, 7.0°S/N, 9.8°S/N, 12.6°S/N, 15.3°S/N, 18.1°S/N,
20.9°S/N,
process 1 : 23.7°S/N, 26.5°S/N, 29.3°S/N, 32.1°S/N, 34.9°S/N, 37.7°S/N, 40.5°S/N,
43.3°S/N,
process 2 : 46.0°S/N, 48.9°S/N, 51.6°S/N, 54.4°S/N, 57.2°S/N, 60.0°S/N, 62.8°S/N,
65.6°S/N,
process 3 : 68.4°S/N, 71.2°S/N, 73.9°S/N, 76.7°S/N, 79.5°S/N, 82.3°S/N, 85.1°S/N,
87.9°S/N.
```

⁴プロセス番号は 0 から始まる. これは MPI の決まり.

⁵この書式は gtool の決まり.


```
&restart_file_io_nml
  OutputFile = 'init_T21L20.nc'
/
```

このとき、それぞれのプロセスにおいて、初期値ファイル・リスタートファイルの名前は、`init_T21L20_rank000000.nc`, `init_T21L20_rank000001.nc`, `init_T21L20_rank000002.nc`, ... と解釈される。

6.3 コンパイル

6.3.1 必要なソフトウェアの準備

dcpam5 の並列計算のためには、以下のライブラリが必要である、

- MPI ライブラリ,
- MPI コンパイラでコンパイルした `ispack`,
- MPI コンパイラでコンパイルした `gtool5`,
- MPI コンパイラでコンパイルした `spml`.

MPI ライブラリのコンパイル、および `ispack`, `gtool5`, `spml` の MPI コンパイラを用いたコンパイルの詳細は、各ライブラリの文書を参照すること。

6.3.2 コンパイル時の注意

コンパイルの基本的な方法は逐次版と同じであり、詳細は、「dcpam5 インストールガイド」(http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/INSTALL.htm) を参照すること。ただし、下の点に注意すること。

- コンパイラとして MPI コンパイラ (例えば `mpif90`) を用いる,
- MPI コンパイラでコンパイルした `ispack`, `gtool5`, `spml` を用いる,
- `configure` のオプションに `--enable-mpi` を指定する。

6.4 並列計算の実行

計算の実行の手順は、逐次版と同じく、

- 初期値の準備,
- 実験用データの準備 (例えば、海表面温度、地形、オゾンの分布のデータのことを意味する),
- 実験の実行,

である。ただし、既に述べたように、入力ファイルはプロセスごとに分割されている必要がある。ここでは、まず Held and Suarez (1994) が提案した力学コア実験を例として取り上げ、実行方法について述べる⁶。

基本的な方法は逐次版と同じであり、「らくらく dcpam5」(http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5_latest/doc/tutorial/gokuraku/) の項目を参照すること。Held and Suarez (1994) の実験においては、初期値を用意し、実行すればよい。初期値は下のように用意する。

```
% mpiexec -n N ./init_data -N=init_data_hs94_T21L20.nml
```

ここで、N はプロセス数である。これにより、初期値ファイル `init_T21L20_rank000000.nc`, `init_T21L20_rank000001.nc`, `init_T21L20_rank000002.nc`, ... が生成される。

次に、下のように実行する。

```
% mpiexec -n N ./dcpam_main -N=dcpam_hs94_T21L20.nml
```

なお、dcpam5 で用意してある初期値生成プログラム、`init_data`、や、惑星表面温度データ生成プログラム、`sst_data`、を用いて入力ファイルを用意する際には、上記のように、それぞれを `mpiexec` を用いて実行することで、プロセスごとのデータファイルを生成することができる。その他のデータファイルに関しては、別途プロセスごとに分割する必要がある。

⁶MPI を用いて並列化されたプログラムの実行方法の一般的な説明・詳細な説明については MPI ライブラリの文書を参照すること。ここで示す方法がどの程度一般的であるかはわからない (yot, 2011/09/30)。

また, 実行により得られる結果は, 上記のように, プロセスごとに分割されている. 必要に応じて統合する必要がある.

6.5 入出力データの分割と統合

入出力データの分割と統合のためにプログラムを用意している. ただし, これらのプログラムは, 今のところ dcpam5 には付属していない. (今のところ) 以下の場所からダウンロードすることができる.

```
http://www.gfd-dennou.org/arch/dcpam/Clipboard/2011-09-14\_yot\_dcpam5-mpi-utils/
```

6.5.1 入力データの分割

入力データの分割のためにプログラム (`util_split`) を用意している. 使い方については, 当該プログラムに付属する README を参照すること.

6.5.2 出力データの統合

出力データの統合のためにプログラム (`util_merge`) を用意している. 使い方については, 当該プログラムに付属する README を参照すること.

付 録 A

使用上の注意とライセンス 規定

CREDITS¹ を参照ください.

¹<http://www.gfd-dennou.org/library/dcpam/dcpam5/dcpam5.latest/CREDITS>