

# USERS GUIDE for SCALE-GM

Version 5.5.1

Team SCALE  
UGC working group

2024 年 4 月 12 日

# 目次

第 1 章	概要	2
1.1	はじめに	2
1.2	SCALE-GM における用語について	2
1.3	表記上の注意	3
第 2 章	クイックスタート	5
2.1	実行の準備	5
2.2	コンパイル	5
2.3	データベースの準備	6
2.3.1	既存のデータベースを使用する	6
2.3.2	新たにデータベースを作成する	6
2.4	実験の実行	9
2.4.1	テストケースの説明	9
2.4.2	計算実行: <code>scale-gm</code>	9
2.5	後処理過程: <code>ico2ll</code>	10
第 3 章	実験設定とモデル設定の変更方法	11
3.1	テストケースの変更方法	11
3.1.1	ディレクトリの準備	11
3.1.2	コンフィグレーションファイルの編集: <code>nhm_driver.cnf</code>	11
3.2	物理スキームの変更方法	15
3.3	MPI プロセスの変更方法	17
3.3.1	ディレクトリを準備する	17
3.3.2	Makefile を編集する	17
3.3.3	コンフィグレーションファイルを編集する: <code>nhm_driver.cnf</code>	17
3.4	水平格子間隔の変更方法	18
3.4.1	ディレクトリの準備	19
3.4.2	Makefile の編集	19
3.4.3	コンフィグレーションファイルの編集: <code>nhm_driver.cnf</code>	19
3.5	参考文献等	22
付 録 A	付録: 設定パラメータの説明	23

# 第1章 概要

## 1.1 はじめに

本書は“SCALE USERS GUIDE”の別冊として SCALE ライブラリパッケージに含まれる全球モデル SCALE-Global Model(GM) の利用方法を説明する。SCALE-GM は SCALE ライブラリを用いて構築された全球大気モデルのことを指す。SCALE-GM の実行手順について DCMIP2016 実験を用いて解説する。SCALE ライブラリの概要については“SCALE USERS GUIDE”の第1章を参照されたい。

SCALE-GM は、Nonhydrostatic ICosahedral Atmospheric Model (NICAM) として開発された正二十面体格子の力学コアをライブラリとして再構成したものである。現行の SCALE-GM は力学コアの理想実験を実行することができる。

ここで、SCALE-GM の力学コアについて簡潔に説明する。予報変数は、密度、運動量、全エネルギー（運動エネルギー＋内部エネルギー）、及び凝結物等のトレーサーである。音波は水平方向には陽解法で計算され、鉛直方向には陰解法で計算される (HEVI)。格子のトポロジーは正二十面体をベースにしており、水平方向には Arakawa A-grid の格子点配置が適用されている。したがって、全ての予報変数は六角形セルの中心に位置している。水平方向のオペレーターの離散化には有限体積法を用いている。鉛直方向には Lorenz タイプのスタaggerd 格子が利用されている。正二十面体格子の最適化にはバネ格子法 (Tomita et al. 2002) を用いて最適化されており、特定の場所に格子点を集めて局所的に高解像度化させるストレッチ格子 (Tomita et al. 2008) も利用できる。有限体積法による水平離散化においては、2 次精度の divergence と gradient が使用されている (Tomita et al. 2001)。トレーサー移流においては、線形再構築を用いた上流型の移流スキーム (Miura 2007) が用いられている。時間積分に関しては、3 段、もしくは 2 段のルンゲ・クッタスキームを用いることができる。また、計算安定性のための数値粘性として、4 次の高次粘性スキームが実装されている。

SCALE-GM の力学コアのさらなる詳細については、Tomita and Satoh (2004) や Satoh et al. (2008) を参照されたい。

## 1.2 SCALE-GM における用語について

- g-level (grid level): 正二十面体は正三角形で面が構成されているが、元の正二十面体から正三角形が何度分割されたか、その分割回数を示す整数である。したがって、g-level が大きいと格子点数が増え、水平格子間隔が小さくなることに相当する。g-level 1 は元の正二十面体を意味する。g-level 4 以上の格子を使用することを推奨する。およそその水平格子間隔は、g-level 4=480 km、5=240 km、6=120 km、7=60 km、8=30 km である。
- r-level (region level): 正二十面体において正三角形を 2 つ組み合わせた菱型のタイルを region と呼び、元の正二十面体から菱形が何度分割されたか、その分割回数を示す整数である。r-level が 0 の場合、10 個の菱形 (region) が存在する。1 つの region に含まれる格子点上の計算を 1 つの MPI プロセスが担当する。したがって、r-level=0 の場合、最大並列数は 10 プロセスであ

る。r-level 1 であれば 40 プロセス、r-level 2 であれば、160 プロセスと並列できるプロセス数が増加する。ただ、r-level を上げることは、Halo に含まれる格子点数を増やすことになるので、低い g-level（低解像度）において、高い r-level を使用することは避けること。

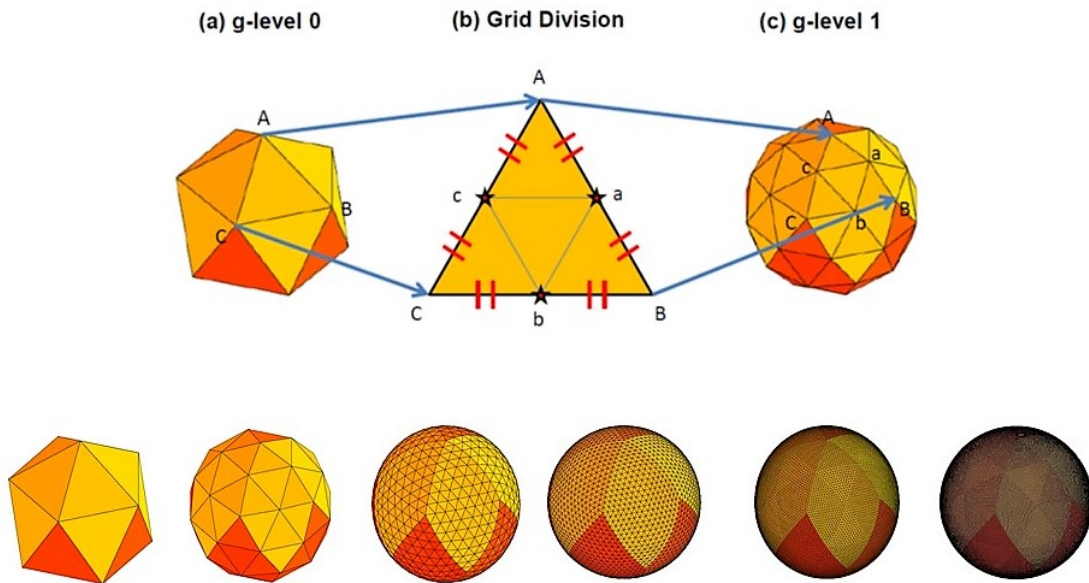


図 1.1: g-level の概念図

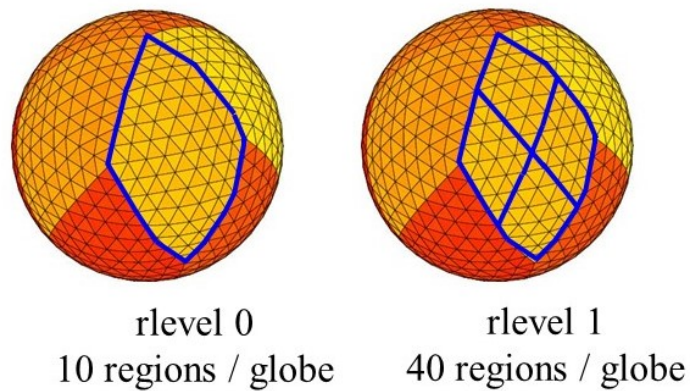


図 1.2: r-level の概念図

[要検討: ここに、g-level と r-level を説明するための適切な図を追加する必要がある: とりあず佐藤さんの発表資料より拝借!さらに HALO を説明する図もある方が良いでしょう]

### 1.3 表記上の注意

- 本書の説明は bash 環境を想定している。tcsh 環境においては、文中のコマンドの “export” を “setenv” に読み替える必要がある。(例. > setenv SCALE\_SYS "Linux64-gnu-mpi")

- “>” の記号は端末においてコマンドを実行することを意味する。
- 文中のゴシック体は標準出力の結果であることを意味する。

## 第2章 クイックスタート

### 2.1 実行の準備

ここでは SCALE-GM のコンパイルについて簡潔に説明する。依存関係のあるライブラリの準備やコンパイル前に必要な環境変数の設定については、“SCALE USERS GUIDE” の第 2 章の 2.1 節、2.2 節を参照されたい。SCALE-GM に必要とされるライブラリは、SCALE-RM と同じく、HDF5、NetCDF、および MPI である。

### 2.2 コンパイル

ソースコードの取得方法については“SCALE USERS GUIDE”の 2.3.1 節、マシン環境やコンパイラ等を指定する環境変数 (Makedef) の設定については“SCALE USERS GUIDE”の 2.3.2 節を参照されたい。Makedef ファイルは、SCALE-RM と共通のファイルを使用する。この Makedef ファイルの設定をコンパイル前に忘れないようにすること。これらの準備が整えば、src ディレクトリへ移動する。SCALE-GM では、scale-5.5.1/scale-gm/src の下でコンパイルを行い、scale-5.5.1/bin の下に各種実行バイナリが生成され、このバイナリファイルを用いて各種実験や後処理を行う。各テストケースディレクトリでも make できるが、作業効率やバイナリの取り違えを防ぐために、ここで説明する方法を推奨する。

```
> cd scale-5.5.1/scale-gm/src
```

make コマンドを用いてコンパイルを行う。

```
> make -j 4*1)
```

コンパイルが正常に終了したならば、以下のバイナリが scale-5.5.1/bin の下に生成される。

- scale-gm (SCALE-GM 本体の実行バイナリ)
- gm\_fio\_cat (fio フォーマットの cat コマンドツール)<sup>\*2)</sup>
- gm\_fio\_dump (fio フォーマットのファイルをダンプするツール)
- gm\_fio\_ico2ll (fio フォーマットの二十面体格子データを LatLon 格子データに変換するツール)
- gm\_fio\_sel (fio フォーマットの sel コマンドツール)

---

<sup>\*1)</sup> make コマンドの -j オプションはコンパイル時に使用する並列プロセス数を指定している。コンパイルにかかる時間を短縮するため、2 以上の数を指定することで並列コンパイルを行うことができる。SCALE-GM では、2 ~ 8 プロセスの指定を推奨する。

<sup>\*2)</sup> “fio” とは、ヘッダー情報付きバイナリをベースにした独自ファイルフォーマットである。

- `gm_mkhgrid` (バネ格子を適用した二十面体の水平格子を作成するツール)
- `gm_mkllmap` (LatLon の水平格子を作成するツール)
- `gm_mkmnginfo` (MPI プロセスのマネージメントファイルを作成するツール)
- `gm_mkrawgrid` (正二十面体の水平格子を作成するツール)
- `gm_mkvlayer` (鉛直格子を作成するツール)

## 2.3 データベースの準備

SCALE-GM の実行には、水平格子のデータが必要である。チュートリアルで使用する設定は、`g-level=5`、`r-level=0` で MPI プロセスは 10 を想定している。この設定のデータベースはソースコードの tarball に含まれているので、別途ダウンロードする必要はないが、これ以外の設定で実験を行う場合は以下に述べるいずれかの方法でデータベースを準備する必要がある。

### 2.3.1 既存のデータベースを使用する

いくつかの代表的な計算設定に対応するデータベースが<https://scale.riken.jp/ja/download/>に用意されているので、それをダウンロードして使用する。

例えば、`scale-gm_database_gl06rl01pe40.tar.gz`<sup>\*3)</sup>をダウンロードした場合、40 個の boundary ファイル (水平格子データベース) と 41 個の llmap ファイル (LatLon 格子変換テーブル) が格納されている。

```
> tar -zxvf scale-gm_database_gl06rl01pe40.tar.gz
```

次にデータベースをそれぞれの格納場所に移動させる。

まず boundary ファイルは、`scale-5.5.1/scale-gm/test/data/grid/boundary` の下へ新たにディレクトリを作成して移動させる。いま、展開したデータベースのディレクトリにいることを想定すると下記のようなコマンド実行になる。

```
> mkdir scale-5.5.1/scale-gm/test/data/grid/boundary/gl06rl01pe40
> mv boundary_GL06RL01.* scale-5.5.1/scale-gm/test/data/grid/boundary/gl06rl01pe40/
```

残る llmap ファイルは、`scale-5.5.1/scale-gm/test/data/grid/llmap` の下へ新たにディレクトリを作成して移動させる。

```
> mkdir -p scale-5.5.1/scale-gm/test/data/grid/llmap/gl06/r101
> mv llmap.* scale-5.5.1/scale-gm/test/data/grid/boundary/gl06/r101/
```

### 2.3.2 新たにデータベースを作成する

[要検討: ここで言及するのがいいのか、後 (例えば 3 章) の方がいいのか]

データベースは必要に応じて以下の要領で作成することができる。

---

<sup>\*3)</sup> `g-level=6, r-level=1, 40` プロセスを使うことを想定

## 1. 水平格子データベースの作成

水平格子データベースの作成は以下の3つの手順からなり、この順番に実行する必要がある。以下のプロセスで作成されるデータベースは前節で述べた場所に保存される。<sup>\*4)</sup>

### (i) プロセス管理データベースの作成

```
> cd scale-5.5.1/scale-gm/test/framework/mkmnginfo/rl01pe40
> make jobshell
> make run
```

この際、以下のファイルを適切に編集する。

```
> vi nhm_driver.cnf
&mkmnginfo_cnf
  rlevel      = 1,                <-- rlevel
  prc_num     = 40,               <-- proess number
  output_fname = "rl01-prc40.info", <-- output filename
/
```

### (ii) 素の格子データベースの作成

```
> cd scale-5.5.1/scale-gm/test/framework/mkrawgrid/gl05rl00pe10
> make jobshell
> make run
```

この際、以下のファイルを適切に編集する。

```
> vi mkrawgrid.cnf
&ADMPARAM
  glevel      = 5,                <-- glevel
  rlevel      = 0,                <-- rlevel
  vlayer      = 1,                <-- vertical layer?
  rgnmngfname = "rl00-prc10.info", <-- input filename
/

&PARAM_MKGRD
  MKGRD_DOSPRING = .true.,        <-- バネ格子を使うかどうか
  MKGRD_OUT_BASENAME = "rawgrid_GL05RL00", <-- output filename
  MKGRD_spring_beta = 1.15D0,     <-- バネ格子を使った場合のバネの強さ
/
```

### (iii) 水平格子データベースの作成

```
> cd scale-5.5.1/scale-gm/test/framework/mkhgrid/gl05rl00pe10
```

---

<sup>\*4)</sup>例えば scale-5.5.1/scale-gm/test/data/grid/boundary

```
> make jobshell
> make run
```

この際、以下のファイルを適切に編集する。

```
> vi mkhgrid.cnf
&ADMPARAM
    glevel      = 5,                <-- glevel
    rlevel      = 0,                <-- rlevel
    vlayer      = 1,                <-- vertical layer
    rgnmngfname = "rl00-prc10.info", <-- input management filename
/

&PARAM_MKGRD
    MKGRD_DOPREROTATE = .false.,    <-- 回転するかどうか
    MKGRD_DOSTRETCH   = .false.,    <-- ストレッチするかどうか
    MKGRD_DOSHRINK    = .false.,    <-- 縮退させるかどうか
    MKGRD_DOROTATE    = .false.,    <-- 始めに回転させるかどうか
    MKGRD_IN_BASENAME = "rawgrid_GL05RL00", <-- input rawgrid filename
    MKGRD_OUT_BASENAME = "boundary_GL05RL00", <-- output boundary filename
/
```

## 2. LatLon 格子変換テーブルデータベースの作成

```
> cd scale-5.5.1/scale-gm/test/framework/mkllmap/gl05rl00pe10_t42
> make jobshell
> make run
```

この際、以下のファイルを適切に編集する。

```
> vi mkllmap.cnf
&ADMPARAM
    glevel      = 5,
    rlevel      = 0,
    vlayer      = 1,
    rgnmngfname = "rl00-prc10.info",
/

&GRDPARAM
    hgrid_io_mode = "ADVANCED",
    hgrid_fname   = "boundary_GL05RL00",
    VGRID_fname   = "NONE",
/

&LATLONPARAM
    latlon_type = "GAUSSIAN",    <-- ‘GAUSSIAN’ or ‘EQUIDIST’ (equal distance)
```

```

imax      = 128,          <-- 経度方向の格子数
jmax      = 64,          <-- 緯度方向の格子数
/

```

## 2.4 実験の実行

### 2.4.1 テストケースの説明

scale-5.5.1/scale-gm/test/case のディレクトリの下にいくつかの理想実験ケースが準備されている。例えば、DCMIP2016 の実験ケースについて表 1 にまとめた。テストケースの詳細については <https://www.earthsystemcog.org/projects/dcmip-2016/testcases> を参照するか、同 Web ページからダウンロードできる Test Case Document を参照されたい。これらの実験ケースディレクトリの下にはさらに格子間隔や MPI プロセス数ごとに異なるディレクトリが作成されている。自分の実行環境や目的に合わせてディレクトリを選択してほしい。また、各種ツールを用いて、任意の格子間隔や MPI プロセス数の設定を作成することも可能である。

表 2.1: Corresponding test cases

SCALE-GM におけるテストケースの名称	実験内容
DCMIP2016-11	湿潤傾圧波理想実験
DCMIP2016-12	全球台風理想実験
DCMIP2016-13	全球スーパーセル理想実験

### 2.4.2 計算実行: scale-gm

計算機上で数値モデルを実行するためのジョブコマンドはシステムによって異なるが、SCALE-GM では計算機環境の違いを考慮してスクリプトを作成するシステムが用意されている。任意の実験ディレクトリ<sup>\*5)</sup>へ移動したあと、モデル実行スクリプトと後処理スクリプトを作成するために下記のコマンドを実行する。

```
> make jobshell
```

このコマンドによって “run.sh” と “ico2ll\_netcdf.sh” のスクリプトが作成される。モデルを実行するためには、下記のようにコマンドを実行する。

```
> make run
```

これで、実験の計算が開始される。DCMIP2016 実験においては、scale-gm を実行したとき、一番はじめに初期値の作成を行っているため、数値モデルの実行前に初期値を作成する手順はない。

要検討：1～3 の実験を gl04 or gl05 くらいで、PE5 あたりで実行したときのおおよその所要時間があると RM における UG の仕様と合わせることができる。また、正常に実行できた場合にどんなファイルが生成されるのか、どれが History ファイルで、どれがログファイルなのかくらいの説明があってもよいだろう。

<sup>\*5)</sup>例えば scale-5.5.1/scale-gm/test/case/DCMIP2016-11/gl06r101z30pe40

## 2.5 後処理過程: ico2ll

計算結果の描画や解析を容易にするために、実験の実行が終了したあと、もとの二十面体格子から LatLon 格子へ格子変換を行う。

後処理過程を実行するまえに、実施した実験設定に合わせて `ico2ll_netcdf.sh` の下記の箇所を編集する必要がある。

```
> vi ico2ll_netcdf.sh

[at Line 22]
# User Settings
# -----

glev=5          # g-level of original grid
case=161        # test case number
out_intev='day' # output interval (format: "1hr", "6hr", "day", "100s")
```

後処理過程のスキプの編集が完了すれば、下記のコマンドによって後処理過程を実行できる。

```
sh ico2ll_netcdf.sh
```

ico2ll によって作成される LatLon 格子 h データは、現行のスキプでは、netcdf フォーマットで出力されるようになっている。この場合、“`nicam.161.200.L30.interp_latlon.nc`” といったファイル名になる。また、スキプの設定を変更することで、GrADS 形式の単純バイナリフォーマットでデータ出力することもできる。

## 第3章 実験設定とモデル設定の変更方法

ここでは、実験設定やモデル設定を変更する際に必要な namelist ファイルの編集方法を説明する。ただし、各々の詳しい説明は控え、使い始めのユーザーに必要な項目のみに留める。

### 3.1 テストケースの変更方法

要検討：以下、XX を指定する、という記述がたくさん出てくるが、指定すると何が起るのか（どういった設定が有効になるのか）という説明も必要である。できる限り、追記お願い致します。ここでは新しくディレクトリを作成し、実験セットを自分で用意する方法を、DCMIP2016 で取り扱われた3つのテストケースの設定について説明する。それぞれのテストケースについて設定済みのコンフィグレーションファイルが `scale-5.5.1/scale-gm/test/case` の中に用意されている。まず第1歩としては、コンフィグレーションファイルを変更せずに実行することを勧める。以下の説明では、

#### 3.1.1 ディレクトリの準備

まず、実行したいテストケースのディレクトリへ移動する。例えば、全球台風理想実験を行いたいのであれば、`scale-5.5.1/scale-gm/test/case/DCMIP2016-12/`へ移動する。その後、新たに実行したい g-level や r-level に合わせたディレクトリを作成する。ここでは、`gl05rl00z30pe40` のディレクトリを新しく作成することを想定して以下の説明を進める。

```
> mkdir gl05rl00z30pe40
> cd    gl05rl00z30pe40
```

Makefile とコンフィグレーションファイルをテンプレートとして既存のディレクトリから新しいディレクトリへコピーする。例えば、DCMIP2016-11 からコピーする。

```
> cp ../../DCMIP2016-11/gl05rl00z30pe10/Makefile      ./
> cp ../../DCMIP2016-11/gl05rl00z30pe10/nhm_driver.cnf ./
```

#### 3.1.2 コンフィグレーションファイルの編集: `nhm_driver.cnf`

以下で、テストケース毎に namelist 中の設定しなければならない項目について説明する。

- 湿潤傾圧波理想実験 (DCMIP2016-11) の設定方法  
("←" のマークは編集が必要な項目を意味する)

```

> vi nhm_driver.cnf
* snip *

&RUNCONFPARAM
  RUNNAME          = 'DCMIP2016-11',          <--
  NDIFF_LOCATION   = 'IN_LARGE_STEP2',
  THUBURN_LIM      = .true.,
  RAIN_TYPE        = "WARM",
  AF_TYPE          = 'DCMIP',
/

* snip *

&DYCORETESTPARAM
  init_type        = 'Jablonowski-Moist',      <--
  test_case        = '1',                      <--
  chemtracer       = .true.,                   <--
  prs_rebuild      = .false.,
/

* snip *

&FORCING_DCMIP_PARAM
  SET_DCMIP2016_11 = .true.,                   <--
/

* snip *

```

## 注意

- 湿潤傾圧波理想実験を行うには、“RUNNAME”を必ず“DCMIP2016-11”と指定すること。
- 同様に“init\_type”は、必ず“Jablonowski-Moist”と指定すること。
- “test\_case”は初期擾乱に関する設定であり、以下の1～6から選択できる。  
 case 1: 初期擾乱あり (指数関数型) / 湿潤実験  
 case 2: 初期擾乱あり (流線関数型) / 湿潤実験  
 case 3: 初期擾乱あり (指数関数型) / 乾燥実験  
 case 4: 初期擾乱あり (流線関数型) / 乾燥実験  
 case 5: 初期擾乱なし / 湿潤実験  
 case 6: 初期擾乱なし / 乾燥実験
- FORCING\_DCMIP\_PARAM は、必ず“SET\_DCMIP2016\_11 = .true.”と指定すること。
- NMHISD の項目にある“step”はヒストリーを出力する間隔を指定する。任意の値を指定できるが、このテストケースでは、通常12時間間隔に相当するステップ数を指定する。

- “NMHIST” の項目の “item” は、ヒストリー出力する変数を指定している。任意の設定が可能である。
- “CNSTPARAM” の項目の “small\_planet\_factor” は必ず “1” と指定すること。

## ○ 全球台風理想実験 (DCMIP2016-12) の設定方法

(“<--” のマークは編集が必要な項目を意味する)

```
> vi nhm_driver.cnf
* snip *

&RUNCONFPARAM
  RUNNAME          = 'DCMIP2016-12',          <--
  NDIFF_LOCATION   = 'IN_LARGE_STEP2',
  THUBURN_LIM      = .true.,
  RAIN_TYPE        = "WARM",
  AF_TYPE          = 'DCMIP',
/

* snip *

&DYCORETESTPARAM
  init_type        = 'Tropical-Cyclone',      <--
/

* snip *

&FORCING_DCMIP_PARAM
  SET_DCMIP2016_12 = .true.,                  <--
/

* snip *
```

### 注意

- 全球台風理想実験を行うには、“RUNNAME” を必ず “DCMIP2016-12” と指定すること。
- 同様に “init\_type” は、必ず “Tropical-Cyclone” と指定すること。
- FORCING\_DCMIP\_PARAM は、必ず “SET\_DCMIP2016\_12 = .true.” と指定すること。
- NMHIST の項目にある “step” はヒストリーを出力する間隔を指定する。任意の値を指定できるが、このテストケースでは、通常 1 時間間隔に相当するステップ数を指定する。
- “NMHIST” の項目の “item” は、ヒストリー出力する変数を指定している。任意の設定が可能である。

- “CNSTPARAM” の項目の “small\_planet\_factor” は必ず “1” と指定すること。

## ○ 全球スーパーセル理想実験 (DCMIP2016-13) の設定方法

(“<--” のマークは編集が必要な項目を意味する)

```
> vi nhm_driver.cnf
* snip *

&RUNCONFPARAM
  RUNNAME          = 'DCMIP2016-13',          <--
  NDIFF_LOCATION   = 'IN_LARGE_STEP2',
  THUBURN_LIM      = .true.,
  RAIN_TYPE        = "WARM",
  AF_TYPE          = 'DCMIP',
/

* snip *

&DYCORETESTPARAM
  init_type        = 'Supercell',              <--
  test_case        = '1',                      <--
/

* snip *

&FORCING_DCMIP_PARAM
  SET_DCMIP2016_13 = .true.,                  <--
/

* snip *
```

### 注意

- 全球スーパーセル理想実験を行うには、“RUNNAME” を必ず “DCMIP2016-13” と指定すること。
- 同様に “init\_type” は、必ず “Supercell” と指定すること。
- “test\_case” は初期擾乱に関する設定であり、以下の 1 2 から選択できる。  
case 1: 初期擾乱あり  
case 2: 初期擾乱なし
- FORCING\_DCMIP\_PARAM は、必ず “SET\_DCMIP2016\_13 = .true.” と指定すること。
- NMHISD の項目にある “step” はヒストリーを出力する間隔を指定する。任意の値を指定できるが、このテストケースでは、通常 2 分間隔に相当するステップ数を指定する。

- “NMHIST” の項目の “item” は、ヒストリー出力する変数を指定している。任意の設定が可能である。
- “CNSTPARAM” の項目の “small\_planet\_factor” は必ず “120” と指定すること。
- “CNSTPARAM” の項目の “earth\_angvel” は必ず “0” と指定すること。
- この実験を行うためには lapack を組み込んだ状態でコンパイルする必要がある。環境変数 SCALE\_ENABLE\_MATHLIB=T を設定し、lapack のライブラリを以下のような要領で指定する必要がある場合がある。export scale\_math\_libs=-L/usr/lib64 -llpack -blas。

上記の編集を行ったあとは、0.2.4 節の説明に沿って計算の実行ができる。

```
> make jobshell
> make run
```

## 3.2 物理スキームの変更方法

各テストケースについて、パッケージに準備されているコンフィグレーションファイルは、DCMIP2016 における標準仕様の物理スキームが設定されている。物理スキームの設定はユーザーが任意に変更することができる。ただし、物理スキームの全ての組み合わせが全てのテストケースにおいて正常に動作することは確かめられていないため、物理スキームを変更して実行する際は注意されたい。

### ○ kessler 雲微物理スキームの代わりに大規模凝結スキームを使用する

DCMIP2016 テストケースにおける標準仕様の雲微物理スキームは Kessler スキームである。この代わりに大規模凝結スキーム (Precipitation Scheme; Reed and Jablonowski 2012) を使用するには、namelist に “SET\_DCMIP2016\_LSC” のパラメータを True 設定で加える。下記に湿潤傾圧波理想実験の場合の例を示す。

(“<--” のマークは編集が必要な項目を意味する)

```
> vi nhm_driver.cnf
* snip *

&FORCING_DCMIP_PARAM
  SET_DCMIP2016_11 = .true.,
  SET_DCMIP2016_LSC = .true.,      <--
/

* snip *
```

### ○ 雲微物理スキームを使用しない

どんな雲微物理に関する物理モデルも使用せずに実験を行うには、namelist に “SET\_DCMIP2016\_DRY” のパラメータを True 設定で加える。下記に湿潤傾圧波理想実験の場合の例を示す。

(“<--” のマークは編集が必要な項目を意味する)

```

> vi nhm_driver.cnf
* snip *

&FORCING_DCMIP_PARAM
  SET_DCMIP2016_11 = .true.,
  SET_DCMIP2016_DRY = .true.,      <--
/

* snip *

```

### ○ George Bryan の大気境界層スキームを使用する

DCMIP2016 テストケースにおける標準仕様の大気境界層スキームは、RJ12 スキーム (Reed and Jablonowski 2012) である。この代わりに George Bryan の大気境界層スキームを使用するには、namelist に “SM\_PBL\_Bryan” のパラメータを True 設定で加える。この設定は、全球台風理想実験についてのみ有効である。以下に設定例を示す。

(“<--” のマークは編集が必要な項目を意味する)

```

> vi nhm_driver.cnf
* snip *

&FORCING_DCMIP_PARAM
  SET_DCMIP2016_12 = .true.,
  SM_PBL_Bryan      = .true.,      <--
/

* snip *

```

### ○ 全ての物理モデルを使用しない

全ての物理モデルを使用せずに実験を行うには、namelist の “RUNCONFPARAM” の項目における “AF\_TYPE” のパラメータに “NONE” を指定する。下記に湿潤傾圧波理想実験の場合の例を示す。

(“<--” のマークは編集が必要な項目を意味する)

```

> vi nhm_driver.cnf
* snip *

&RUNCONFPARAM
  RUNNAME          = 'DCMIP2016-11',
  NDIFF_LOCATION   = 'IN_LARGE_STEP2',
  THUBURN_LIM      = .true.,
  RAIN_TYPE        = "WARM",
  AF_TYPE          = 'NONE',      <--
/

* snip *

```

### 3.3 MPI プロセスの変更方法

並列計算に用いる MPI プロセス数を増やすことで、数値モデルの計算実行にかかる所要時間を短縮することができる。以下では、湿潤傾圧波理想実験の g-level 5 において、40 プロセスを使用して実行するための変更を例にして説明する。

10 プロセスで実行していた設定を 40 プロセスに増加させるためには、r-level 0 の並列数の上限が 10 プロセスなので、r-level を 0 から 1 へ上げなければならない。

#### 3.3.1 ディレクトリを準備する

ここでは、scale-5.5.1/scale-gm/test/case/DCMIP2016-11/ のディレクトリにいることを想定している。

```
> mkdir gl05r101z30pe40 <-- r-level is 1
> cd gl05r101z30pe40/
```

Makefile とコンフィグレーションファイルを既存のディレクトリから新規ディレクトリへコピーする。

```
> cp ../gl05r100z30pe10/Makefile ./
> cp ../gl05r100z30pe10/nhm_driver.cnf ./
```

#### 3.3.2 Makefile を編集する

(“<--” のマークは編集が必要な項目を意味する)

17 行目から 21 行目のパラメータを下記のように変更する。

```
> vi Makefile
glevel = 5
rlevel = 1 <--
nmpi = 40 <--
zlayer = 30
vgrid = vgrid30_stretch_30km_dcmip2016.dat
```

#### 3.3.3 コンフィグレーションファイルを編集する: nhm\_driver.cnf

(“<--” のマークは編集が必要な項目を意味する)

```
> vi nhm_driver.cnf
* snip *
&ADMPARAM
  glevel      = 5,
  rlevel      = 1, <--
  vlayer      = 30,
  rgnmngfname = "rl01-prc40.info", <--
```

```

/

&GRDPARAM
  hgrid_io_mode = "ADVANCED",
  hgrid_fname   = "boundary_GL05RL01", <--
  VGRID_fname   = "vgrid30_stretch_30km_dcmip2016.dat",
  vgrid_scheme  = "LINEAR",
/

* snip *

&RESTARTPARAM
  input_io_mode   = 'IDEAL',
  output_io_mode  = 'ADVANCED',
  output_basename = 'restart_all_GL05RL01z30', <--
  restart_layername = 'ZSALL32_DCMIP16',
/

```

上記の編集を行ったあとは、0.2.4 節の説明に沿って計算の実行ができる。

```

> make jobshell
> make run

```

## 注意

- MPI プロセス数を増やせば必ずしも計算の所要時間が短くなるわけではない。使用する計算機に適したプロセス数を指定することが肝要である。
- MPI プロセス数を変更したり、r-level を上げたりした場合、その設定に対応する水平グリッドファイル (boundary ファイルと lmap ファイル) やリージョン管理ファイル (region manage file) が database に存在するか必ず確認すること。これらがなければ、0.2.3 節のデータベースの準備の説明にしたがって SCALE の Web ページからデータベースをダウンロードするか、自分でツールを使って作成する必要がある。

## 3.4 水平格子間隔の変更方法

ここでは g-level 6 (約 120 km 格子間隔)、MPI プロセスは 40 プロセスを使用する設定に変更する場合を例にして説明する。テストケースは湿潤傾圧波理想実験を想定する。水平格子間隔を変更した場合は、格子に関する設定以外に、数値積分間隔 (DTL)、最大タイムステップ数 (LSTEP\_MAX)、数値粘性フィルターに関する各パラメータ、そしてヒストリー出力間隔といったパラメータの変更が追加で必要になる。

### 3.4.1 ディレクトリの準備

ここでは、はじめに `scale-5.5.1/scale-gm/test/case/DCMIP2016-11/` のディレクトリにいることを想定する。当該ディレクトリに下記のコマンドで新たにディレクトリを作成し、移動する。

```
> mkdir gl06r101z30pe40 <-- g-level is 6, and r-level is 1
> cd gl06r101z30pe40/
```

テンプレートとして利用するため、Makefile とコンフィグレーションファイルを既存のディレクトリからコピーする。

```
> cp ../gl05r100z30pe10/Makefile ./
> cp ../gl05r100z30pe10/nhm_driver.cnf ./
```

### 3.4.2 Makefile の編集

(“<--” のマークは編集が必要な項目を意味する)

Makefile の 17 行目から 21 行目を下記のように編集する。

```
> vi Makefile
glevel = 6      <--
rlevel = 1      <--
nmpi   = 40     <--
zlayer = 30
vgrid  = vgrid30_stretch_30km_dcmip2016.dat
```

### 3.4.3 コンフィグレーションファイルの編集: nhm\_driver.cnf

数値積分間隔や数値粘性のパラメータ設定に関しては、実験が完走するように微調整が必要になる場合があるが、まずは以下に示すガイドラインにしたがって設定する。

- 数値積分間隔 (DTL) に対しては、  
g-level を 1 上げるごとに、DTL を元の 1/2 に設定することを推奨する。
- 数値粘性フィルターに対しては、  
g-level を 1 上げるごとに、各係数の値を元の 1/8 に設定することを推奨する。

このガイドラインにしたがって、設定を編集すると下記ようになる。

(“<--” のマークは編集が必要な項目を意味する)

```
> vi nhm_driver.cnf
* snip *
&ADMPARAM
    glevel      = 6,          <--
```

```

    rlevel      = 1,                <--
    vlayer      = 30,
    rgnmngfname = "rl01-prc40.info", <--
/

&GRDPARAM
    hgrid_io_mode = "ADVANCED",
    hgrid_fname   = "boundary_GL06RL01", <--
    VGRID_fname   = "vgrid30_stretch_30km_dcmip2016.dat",
    vgrid_scheme  = "LINEAR",
/

&TIMEPARAM
    DTL          = 300.D0,          <--
    INTEG_TYPE    = "RK3",
    LSTEP_MAX     = 4320,           <--
    start_date    = 0000,1,1,0,0,0
/

* snip *

&RESTARTPARAM
    input_io_mode   = 'IDEAL',
    output_io_mode  = 'ADVANCED',
    output_basename = 'restart_all_GL06RL01z30', <--
    restart_layername = 'ZSALL32_DCMIP16',
/

* snip *

&NUMFILTERPARAM
    lap_order_hdiff = 2,
    hdiff_type      = 'NONLINEAR1',
    Kh_coef_maxlim  = 1.500D+16,    <--
    Kh_coef_minlim  = 1.500D+15,    <--
    ZD_hdiff_nl     = 20000.D0,
    divdamp_type    = 'DIRECT',
    lap_order_divdamp = 2,
    alpha_d         = 1.50D15,      <--
    gamma_h_lap1    = 0.0D0,
    ZD              = 40000.D0,
    alpha_r         = 0.0D0,
/

```

```

* snip *

&NMHISD
  output_io_mode   = 'ADVANCED' ,
  histall_fname    = 'history'   ,
  hist3D_layername = 'ZSDEF30_DCMIP16',
  NO_VINTRPL       = .false.     ,
  output_type      = 'SNAPSHOT'  ,
  step             = 288          ,    <--
  doout_step0      = .true.      ,
/

```

上記の編集を行ったあとは、0.2.4 節の説明に沿って計算の実行ができる。

```

> make jobshell
> make run

```

要検討：DCMIP では鉛直層設定が取り決められており、変更の余地がなかったので記述していなかったが、モデルのユーザズガイドとしては、鉛直層数の設定変更や鉛直層設定の作成方法についても記述があるべきだろうと思う。

### 3.5 参考文献等

Tomita,H., Tsugawa,M., Satoh,M., Goto, K.(2001) : Shallow water model on a modified icosahedral geodesic grid by using spring dynamics. J. Comp. Phys., 174, 579-613

Tomita, H., Satoh, M., Goto, K.(2002) : An optimization of icosahedral grid by using spring dynamics. J. Comp. Phys., 183, 307-331

Tomita, H. and Satoh, M. (2004) : A new dynamical framework of nonhydrostatic global model using the icosahedral grid. Fluid Dyn. Res., 34, 357-400.

Miura,H., Satoh, M., Nasuno, T., Noda, A.T., Oouchi, K. (2007) : A Madden-Julian Oscillation event simulated using a global cloud-resolving model. Science, 318, 1763-1765.

Miura, H., 2007: An Upwind-Biased Conservative Advection Scheme for Spherical Hexagonal-Pentagonal Grids. Mon. Wea. Rev., 135, 4038-4044.

Satoh, M., T. Matsuno, H. Tomita, H. Miura, T. Nasuno, S. Iga, (2008) : Nonhydrostatic Icosahedral Atmospheric Model (NICAM) for global cloud resolving simulations. Journal of Computational Physics, the special issue on Predicting Weather, Climate and Extreme events, 227, 3486-3514, doi:10.1016/j.jcp.2007.02.006.

Tomita, H. (2008) : A stretched grid on a sphere by new grid transformation. J. Meteor. Soc. Japan, 86A, 107-119.

NICAM のホームページ : <http://nicam.jp/>

## 付 録 A      付録: 設定パラメータの説明

「設定例」のカラムは、g-level 5 の湿潤傾圧波理想実験の場合を示す。

表 A.1: ADMPARAM (モデル管理パラメータ群)

パラメータ	設定例	kind	説明
glevel	5	int	g-level の値
rlevel	0	int	r-level の値
vlayer	30	int	鉛直層数
rgnmngfname	"rl00-prc10.info"	char	リージョン管理ファイルの名前

表 A.2: GRDPARAM (格子設定パラメータ群)

パラメータ	設定例	kind	説明
hgrid_io_mode	"ADVANCED"	char	水平格子ファイルの IO モード
hgrid_fname	"boundary_GL05RL00"	char	水平格子ファイルの名前
VGRID_fname	"vgrid30_stretch_30km_dcmip2016.dat"	char	鉛直格子ファイルの名前
vgrid_scheme	"LINEAR"	char	鉛直格子ファイルの IO モード

表 A.3: TIMEPARAM (時間積分設定パラメータ群)

パラメータ	設定例	kind	説明
DTL	600.D0	real	数値積分時間間隔 (s)
INTEG_TYPE	"RK3"	char	数値積分スキームの種類
LSTEP_MAX	2160	int	時間積分ステップ数
start_date	0000,1,1,0,0,0	int (array)	初期値の年、月、日、時、分、秒

表 A.4: RUNCONFPARAM (モデル内共通設定パラメータ群)

パラメータ	設定例	kind	説明
RUNNAME	'DCMIP2016-11'	char	テストケースの名前
NDIFF_LOCATION	'IN_LARGE_STEP2'	char	数値フィルターの設定
THUBURN_LIM	.true.	logical	リミッターの使用フラグ
EIN_TYPE	'SIMPLE'	char	内部エネルギーの評価手法
RAIN_TYPE	'WARM'	char	降雨評価の手法
CHEM_TYPE	'PASSIVE'	char	化学トレーサーのタイプ
AF_TYPE	'DCMIP2016'	char	外部強制 (物理) の種類

表 A.5: CHEMVARPARAM (化学トレーサー設定パラメータ群)

パラメータ	設定例	kind	説明
CHEM_TRC_vmax	2	int	化学トレーサーの最大種類数

表 A.6: BSSTATEPARAM (ベーシックステート (reference) 設定パラメータ群)

パラメータ	設定例	kind	説明
ref_type	'NOBASE'	char	ベーシックステートのタイプ

表 A.7: RESTARTPARAM (初期値/再計算 設定パラメータ群)

パラメータ	設定例	kind	説明
input_io_mode	'IDEAL'	char	入力ファイルの IO モード (初期値ファイル用)
output_io_mode	'ADVANCED'	char	出力ファイルの IO モード (リスタートファイル用)
output_basename	'restart_all_GL05RL00z30'	char	出力ファイルの名前
restart_layername	'ZSALL32_DCMIP16'	char	リスタートファイル向け鉛直層情報ファイルの名前

表 A.8: DYCORETESTPARAM (力学コアテストケース設定パラメータ群)

パラメータ	設定例	kind	説明
init_type	'Jablonski-Moist'	char	テストケース名
test_case	'1'	char	テストケース番号 (DCMIP のテストケース番号ではない)
chemtracer	.true.	logical	化学トレーサーのスイッチ
prs_rebuild	.false.	logical	初期値の気圧を再計算するスイッチ

表 A.9: FORCING\_PARAM (外部強制 (物理過程) 設定パラメータ群)

パラメータ	設定例	kind	説明
NEGATIVE_FIXER	.true.	logical	トレーサーに対する負値修正のスイッチ
UPDATE_TOT_DENS	.false.	logical	全密度の修正に対するスイッチ

表 A.10: FORCING\_DCMIP\_PARAM (DCMIP2016 物理過程設定パラメータ群)

パラメータ	設定例	kind	説明
SET_DCMIP2016_11	.true.	logical	DCMIP2016 湿潤傾圧波理想実験向け物理セットのスイッチ (排他的利用)
SET_DCMIP2016_12	.false.	logical	DCMIP2016 全球台風理想実験向け物理セットのスイッチ (排他的利用)
SET_DCMIP2016_13	.false.	logical	DCMIP2016 全球スーパーセル理想実験向け物理セットのスイッチ (排他的利用)

表 A.11: PARAM\_CONST (物理定数パラメータ群)

パラメータ	設定例	kind	説明
earth_radius	6.37122D+6	real	地球の半径 (m)
earth_angvel	7.292D-5	real	地球自転の角速度 (s-1)
small_planet_factor	1.D0	real	小惑星係数 (X)
earth_gravity	9.80616D0	real	重力加速度 (m s-2)
gas_cnst	287.0D0	real	乾燥空気の理想気体定数 (J kg-1 K-1)
specific_heat_pre	1004.5D0	real	定圧比熱 (J kg-1 K-1)

表 A.12: NUMFILTERPARAM (数値フィルター設定パラメータ群)

パラメータ	設定例	kind	説明
lap_order_hdiff	2	int	水平拡散スキームのオーダー
hdiff_type	'NONLINEAR1'	char	水平拡散のタイプ
Kh_coef_maxlim	1.200D+17	real	最大値 of Kh 係数 (for Non-Linear)
Kh_coef_minlim	1.200D+16	real	最小値 of Kh 係数 (for Non-Linear)
ZD_hdiff_nl	20000.D0	real	水平 2 次元ダンピング層の下端高度 (for Non-Linear)
divdamp_type	'DIRECT'	char	拡散型ダンピングのタイプ
lap_order_divdamp	2	int	拡散型ダンピングのオーダー
alpha_d	1.20D16	real	拡散型ダンピング係数の指定値

表 A.13: EMBUDGETPARAM (予報変数収支モニター設定パラメータ群)

パラメータ	設定例	kind	説明
MNT_ON	.true.	logical	モニタリングのスイッチ
MNT_INTV	72	int	モニタリング間隔 (steps)

表 A.14: NMHISD (ヒストリー出力共通設定パラメータ群)

パラメータ	設定例	kind	説明
output_io_mode	'ADVANCED'	char	ヒストリー出力ファイルの IO モード
histall_fname	'history'	char	ヒストリー出力ファイルの名前
hist3D_layername	'ZSDEF30_DCMIP16'	char	ヒストリー出力ファイル向け鉛直層情報ファイルの名前
NO_VINTRPL	.false.	logical	ヒストリー出力時の鉛直内挿のスイッチ
output_type	'SNAPSHOT'	char	ヒストリー出力値の設定 (snapshot or average)
step	72	int	ヒストリー出力間隔 (steps)
doout_step0	.true.	logical	初期値を出力するスイッチ

表 A.15: NMHIST (ヒストリー出力アイテム設定パラメータ群)

パラメータ	設定例	kind	説明
item	'm1_u'	char	ヒストリー出力変数のモデル内の名前
file	'u'	char	ヒストリー出力変数の出力ファイル内の名前
ktype	'3D'	char	出力変数の次元